B2B e-commerce
with WebSphere Commerce Business Edition V5.4
Patterns for e-business Series

Selecting application and runtime patterns for B2B e-commerce
Design and development guidelines
B2B e-commerce examples

Bill Moore
Andrew McDougall
Erik Rolshausen
Stephen Smith
Bernd Stehle

ibm.com/redbooks
B2B e-commerce with WebSphere Commerce Business Edition V5.4 Patterns for e-business Series

February 2003
Note: Before using this information and the product it supports, read the information in “Notices” on page xi.

First Edition (February 2003)

This edition applies to Versions 5.4 of WebSphere Commerce Business Edition and WebSphere Commerce Professional Edition.

© Copyright International Business Machines Corporation 2003. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Contents

Notices .................................................. xi
Trademarks .............................................. xii

Preface .................................................... xiii
The team that wrote this redbook. ........................................ xiv
Become a published author ............................................. xv
Comments welcome .............................................. xvi

Chapter 1. Introduction ...................................... 1
1.1 Our objectives ........................................ 2
1.2 Our focus ............................................. 4
1.3 How to use this guide .................................... 6

Chapter 2. Applying patterns ................................. 9
2.1 Architectural enablement tools .......................... 10
  2.1.1 Patterns for e-business ............................ 10
  2.1.2 Use of patterns with a simple example .............. 12
2.2 Pattern application for a B2B commerce site ......... 30
  2.3 Process and methods used in our redbook chapters ...
     2.3.1 Our starting point .................................. 32
     2.3.2 Pattern usage ........................................ 32
     2.3.3 Example proposal ..................................... 32
     2.3.4 We get the job ........................................ 32
  2.4 The Sell-Side Hub composite pattern ................. 33
     2.4.1 Introduction to e-Marketplaces .................... 33
     2.4.2 Fit for our imaginary business problem ............ 34
     2.4.3 More general fit for business problems .......... 35

Chapter 3. Requirements .................................... 37
3.1 About requirements ..................................... 38
  3.1.1 Definition of key terms ............................. 38
  3.1.2 The importance of requirements .................... 39
  3.1.3 How to gather requirements ......................... 39
3.2 The requirements of our customer ....................... 40
  3.2.1 Assumptions about the customer situation ............ 40
  3.2.2 Short business profile ................................ 41
  3.2.3 Business goals ........................................ 41
  3.2.4 Existing environment ................................ 41
  3.2.5 Functional requirements ............................ 43
Index ................................................................. 349
Notices

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

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area.

Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which 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:

<table>
<thead>
<tr>
<th>IBM®</th>
<th>Lotus Notes®</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIX®</td>
<td>Lotus QuickPlace ™</td>
</tr>
<tr>
<td>Approach®</td>
<td>Lotus Sametime™</td>
</tr>
<tr>
<td>CICS®</td>
<td>Net.Commerce™</td>
</tr>
<tr>
<td>DB2®</td>
<td>Notes®</td>
</tr>
<tr>
<td>DB2 Universal Database™</td>
<td>pSeries™</td>
</tr>
<tr>
<td>Domino™</td>
<td>QuickPlace™</td>
</tr>
<tr>
<td>eServer™</td>
<td>RS/6000®</td>
</tr>
<tr>
<td>Everyplace™</td>
<td>Sametime®</td>
</tr>
</tbody>
</table>

The following terms are trademarks of other companies:

- ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

This IBM Redbook is part of the Patterns for e-business series; in it, we examine how Composite patterns can be used when developing e-commerce Web sites using WebSphere Commerce Business Edition. In particular, we consider the Sell-Side Hub composite pattern and describe how it can assist architects and designers who have been asked to develop an e-Marketplace using WebSphere Commerce Business Edition.

We provide a Patterns for e-business approach to designing a B2B solution based on WebSphere Commerce Business Edition, and focus on the Sell-Side Hub composite pattern. This redbook is intended for architects, designers and developers.

We approach the problem by developing a technical architecture based on requirements that we expect are common to e-Marketplaces that are Sell-Side Hubs. In developing this architecture, we envisage an imaginary customer, with realistic requirements, and then follow a representative project development life cycle for this customer. The steps documented in the redbook include the following.

- Requirements gathering
- Understanding the business problem
- System overview
- Operational overview
- Detailed design
- Implementation
- Implementation samples
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, Raleigh Center.

**Bill Moore** is a WebSphere Specialist at the International Technical Support Organization, Raleigh Center. He writes extensively and teaches IBM classes on WebSphere and related topics. Before joining the ITSO, Bill was a Senior AIM Consultant at the IBM Transarc lab in Sydney, Australia. He has 18 years of application development experience on a wide range of computing platforms and using many different coding languages. He holds a Master of Arts degree in English from the University of Waikato, in Hamilton, New Zealand. His current areas of expertise include application development tools, object-oriented programming and design, and e-business application development.

**Andrew I. McDougall** is an IT Architect working for ITS in IBM Australia. He joined IBM in June 2001 and has over ten years of experience in IT development. He holds a degree in Business and Commerce Management from Lincoln University in New Zealand. His areas of expertise include e-business solution design and WebSphere Commerce implementation on multi-platforms. He is an IBM certified for e-business solution designer.

**Erik Rolshausen** is a Senior Architect at SerCon GmbH, a German subsidiary of IBM Global Services. He has four years of experience in the e-business field and has held a number of technical roles as well as project coordinating roles. Erik holds a degree in Business Administration and Engineering from Karlsruhe University of Applied Sciences in Germany. He has worked in many different customer situations and has a wide range of experience in Internet applications, especially Lotus Domino, IBM WebSphere Commerce and IBM Net.Commerce.

**Stephen Smith** is a Senior IT Architect with 16 years of experience in the IT industry. He works with the RAPM group within AMS, IBM Global Services Australia, and has been with IBM Global Services Australia since 1997. His areas of expertise include architecting Web infrastructure and application solutions for large corporate environments.

**Bernd Stehle** is an e-business Consultant at Sercon Consulting and Services in Ulm, Germany, a subsidiary of IBM. He has three years of experience in application design, GUI development, consulting and building Web applications using WebSphere Portal and WebSphere Commerce. His areas of expertise include e-commerce, Web portals and front-end programming as well as teaching classes on Lotus Notes for end users. Bernd holds a degree in Business Administration and Engineering from the University of Applied Sciences of Neu-Ulm, Germany.
Thanks to the following people for their contributions to this project:

Carmelo Porciuncula  
(IBM Australia)

Imad Albazz  
Afshan Ally  
Bob Fraser  
Ron Laudadio  
Jonathan Lin  
Erik Neathery  
John Rothwell  
Sam Wong  
(IBM Canada)

Jonathan Adams  
(IBM UK)

**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 Internet note to:
  redbook@us.ibm.com

- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HZ8 Building 662
  P.O. Box 12195
  Research Triangle Park, NC 27709-2195
Chapter 1. Introduction

This chapter describes our objectives, our focus and who can benefit from the content of this book. We outline the use of patterns and how to apply them and introduce an example of applying a Sell-Side Hub composite pattern using WebSphere Commerce Business Edition.

This chapter covers the following:

- Our objectives
- Our focus
- How to use this guide
1.1 Our objectives

The reuse of architectural assets from Patterns for e-business can save weeks of effort, reduce costs and add to the intellectual capital of a development organization. However, when the solution architect is faced with a complex business problem and disparate customer requirements, applying a pattern may be difficult. Moreover, when the application platform has already been mandated by the customer, or by the sales process, the application of patterns may be further complicated.

Patterns for e-business are a group of reusable assets that leverage the experience of IBM architects to create solutions for e-business problems. The Patterns for e-business can be broken down into the following elements:

- Business patterns that identify the interaction between users, businesses and data to create simple functionality.
- Integration patterns that connect other Business patterns together to create complex functionality.
- Composite patterns that are combinations of Business and Integration patterns for advanced applications.
- Custom designs that are developed to solve specific e-business problems for a single customer organization.

Depending on the customer requirements, the Application and Runtime patterns describe the application and runtime environments needed to develop the solution. The product mappings and guidelines for the design, development and deployment of the solution are assets for supporting the implementation of the solution. Figure 1-1 illustrates how the assets of the pattern can be used to translate customer requirements into an application.
WebSphere Commerce Business Edition is a solution for selling in the Business-to-Business (B2B) and Business-to-Consumer (B2C) environments, with extensions beyond the traditional Web storefront. It includes advanced capabilities that enable organizations to build and maintain meaningful relationships with customers and trading partners. The application reduces the complexity by managing business processes and integrating them with external and back-end business applications. In addition, Business Edition allows customers and trading partners to collaborate with virtual teaming technology.

Our objective in this guide is to discuss the use of a Sell-Side Hub composite pattern to assist in architecting a B2B solution using Business Edition. We do this by providing an overview of e-business patterns, applying the Sell-Side Hub composite pattern to a set of customer requirements and describing the development of a sample Sell-Side Hub functionality with Business Edition.
1.2 Our focus

This guide is intended for solution architects and system designers tasked with developing a technical architecture based on a Sell-Side Hub composite pattern using Business Edition. This guide may also be useful for other technical professionals developing technical architectures, using the Patterns for e-business on different application server platforms. In this book, we focus specifically on applying e-business patterns and do not cover server configuration, optimization and maintenance issues. See the redbook *WebSphere Commerce Suite V5.4 Handbook, Architecture and Integration Guide*, SG24-6567 for details on server configuration and management.

Our primary focus is applying the Sell-Side Hub composite pattern for B2B applications using Business Edition. The Composite patterns combine Business patterns and Integration patterns to create complex, advanced e-business applications. A solution design composed of these multiple building blocks is only considered a Composite pattern when it has been used repeatedly across a wide range of industries. The Sell-Side Hub composite pattern is part of a larger subset of Composite patterns that include trading exchanges and Buy-Side Hubs. This subset is called e-Marketplaces.

e-Marketplaces are trading exchanges that facilitate and promote buying and selling among trading partners within an industry. These solutions represent some of the complex e-business applications that Business Edition is designed to address. Three distinct types of e-Marketplaces have been defined by the Patterns for e-business:

- Trading Exchange allows buyers and sellers to trade goods and services on a public site.
- Sell-Side Hub is an e-Marketplace owned by the seller, who uses it to sell goods and services to prospective buyers across the Internet either through a browser, a pervasive device, or an interface with a customer's ERP or Buy-Side Hub system.
- Buy-Side Hub is an e-Marketplace owned by the buyer of the goods or services, who uses it for procurement and to solicit the best deals from prospective customers across the Internet, either through a browser, a pervasive device, or an interface with a customer's ERP or Sell-Side Hub system.

All three types of e-Marketplaces are composed of multiple pattern building blocks. Each building block pattern is either mandatory or optional for defining the type of e-Marketplace. For the Sell-Side Hub composite pattern, the building block patterns are defined as follows.
- Access Integration patterns provide a unified customer interface and are mandatory.

- Self-Service business patterns allow users to browse through a catalog and place an order with the hub; they are mandatory.

- Information Aggregation business patterns are used to create a catalog from the supplier’s product files, pricing files, and marketing collateral; they are mandatory.

- Application Integration patterns integrate the Business patterns that make up the Sell-Side Hub; they are mandatory.

- Collaboration patterns enable dynamic trading mechanisms, including auctions, and are optional.

- Extended Enterprise business patterns, including payment gateways, are optional.

Figure 1-2 shows a view of the patterns that are optional or mandatory in the Sell-Side Hub composite pattern.
As we discuss, a set of complex customer business requirements does not necessarily fit neatly into these discrete definitions of e-Marketplace patterns. Therefore, it is necessary to match the customer’s set of requirements against the various types of e-business patterns and look for a best fit. The process of selecting a core e-business pattern and adding components of subsidiary patterns is therefore a primary focus of this redbook.

In this redbook, we discuss the application of patterns and provide an example of matching a set of complex customer requirements against the patterns to produce a variety of architectural assets. Finally, we describe the development of four examples of Sell-Side Hub functionality using Business Edition.

1.3 How to use this guide

The chapters in this guide are organized around the following general topics.

► Overview of applying Patterns for e-business
  
  Chapter 2, “Applying patterns” on page 9
  
  This chapter provides an overview of e-business patterns and how they are related. It explains how patterns can be applied during the process of implementing a Business-to-Business commerce site. There is a discussion of the Sell-Side Hub composite pattern and an explanation of how we illustrated these topics by a worked example.

► Applying the Sell-Side Hub composite pattern includes:
  
  – Chapter 3, “Requirements” on page 37
  – Chapter 4, “Understanding the business problem” on page 55
  – Chapter 5, “Proposing a solution” on page 71
  – Chapter 6, “Operational overview” on page 155
  – Chapter 7, “Detailed design” on page 169
  – Chapter 8, “Implementation guidelines” on page 195

An example of applying the Sell-Side Hub composite pattern is covered in these chapters. Chapter 3, “Requirements” on page 37 describes how to gather requirements for a customer and discusses the selection of an Application pattern based on the requirements. Chapter 5, “Proposing a solution” on page 71 describes a system proposal for our customer and expresses this using a Runtime pattern. Chapter 6, “Operational overview” on page 155 discusses the way in which the functional and non-functional operational aspects of the system can be matched to a Product mapping. Chapter 7, “Detailed design” on page 169 provides details and gives guidance as to how to take the high-level overviews and express these as
more detailed technical design documents including use cases, user interaction, interface design, storyboards, and component interactions. Chapter 8, “Implementation guidelines” on page 195 discusses how to develop and implement the solution, including a gap analysis for identifying customizations, as well as guidelines for runtime implementation, development and testing.

- Worked examples include the following.
  - Chapter 9, “Procurement integration” on page 239
  - Chapter 10, “Customer care” on page 273
  - Chapter 11, “User identity integration” on page 303


This book covers a wide range of patterns topics. The book can be read from cover to cover and you are encouraged to do so. However, if you have a specific architectural or development need, you may wish to choose one of the following paths, depending on your interests:

1. Understanding Patterns for e-business
   - a. Review the short introduction to patterns in this chapter.

2. Creating a technical architecture based on a pattern
   - a. Read Chapter 3, “Requirements” on page 37 for information on gathering customer requirements.
   - b. Read Chapter 4, “Understanding the business problem” on page 55 for information on creating a solution using a Composite pattern.
   - c. Read Chapter 5, “Proposing a solution” on page 71 for information on creating a high-level architecture using an application pattern.
   - d. Read Chapter 6, “Operational overview” on page 155 for information on product mappings.

3. Creating technical design specifications and runtime implementation
   - a. Read Chapter 7, “Detailed design” on page 169 for information on creating detailed design specifications using the patterns design guidelines.
b. Read Chapter 8, “Implementation guidelines” on page 195 for information on developing and implementing the solution using the patterns implementation guidelines.

4. Developing Sell-Side Hub functionality using Business Edition
   a. Read Chapter 9, “Procurement integration” on page 239 for an example of Procurement integration with Buy-Side Hub’s using Business Edition.
   b. Read Chapter 10, “Customer care” on page 273 for an example of using Sametime and Quickplace for Customer Service.
   c. Read Chapter 11, “User identity integration” on page 303 for an example of integrating the Business Edition Member subsystem with a third-party access control system.
Chapter 2. Applying patterns

This chapter provides:

- An overview of Patterns for e-business.
- A discussion of how patterns can be applied during the process of implementing a B2B commerce site.
- A high-level overview of the process and methods we will use throughout this redbook to obtain requirements, construct a proposal, design the solution and implement the system.
- A discussion of the Sell-Side Hub composite pattern and its alignment with our business problem and other common business problems. In particular, we discuss whether the Sell-Side Hub composite pattern is an accurate description of a system that satisfies our relatively common requirements and hence whether we would implement it for our imaginary customer.

**Note:** In writing this chapter, we also considered the use of reference architectures and e-commerce business models. We concluded that there is some overlap between these ideas and Patterns for e-business, but that the scope of our redbook did not warrant a detailed discussion of these ideas. See Appendix A, “E-commerce business models” on page 311 for some details on e-commerce business models.
2.1 Architectural enablement tools

This section provides an overview discussion of Patterns for e-business, the use of reference architectures and WebSphere Commerce scenarios and a discussion of where these items are related, overlap or conflict.

2.1.1 Patterns for e-business

When experienced IT architects go to work each day and face a business problem, they may not realize it, but they use patterns. They have seen the business problem before and have devised a solution to solve the problem before as well. In recognizing the problem pattern and in remembering the previous solution, they have chosen a Runtime pattern to solve a recognized Business pattern. We would hope that the Runtime pattern selection is based on a solution that was proven to work well for the business problem at hand. If the architect were to share the patterns, others would appreciate a standard notation and terminology so they could be in a position to digest the content immediately rather than have to interpret the message that the material is trying to deliver.

The IBM Patterns for e-business have the advantage of being proven. Over 20,000 successful projects have been delivered, taking advantage of the patterns approach found at the IBM Patterns for e-business Web site (http://ibm.com/developerworks/patterns/). The IBM Patterns for e-business also provide a standardized terminology and notation that allows for easy communication.

Patterns for e-business overview

The following overview has been taken directly from the IBM Patterns for e-business Web site.

- Patterns for e-business are a group of reusable assets that can help speed the process of developing Web-based applications. This site breaks down these reusable assets into the following elements.
- Business patterns identify the interaction between users, businesses, and data. Business patterns are used to create simple, end-to-end e-business applications.
- Integration patterns connect other Business patterns together to create applications with advanced functionality. Integration patterns are used to combine Business patterns in advanced e-business applications.
- Composite patterns are combinations of Business patterns and Integration patterns that have themselves become commonly used types of e-business applications. Composite patterns are advanced e-business applications.
Custom designs are similar to Composite patterns, as they combine Business patterns and Integration patterns to form an advanced, end-to-end solution. These solutions, however, have not been implemented to the extent of Composite patterns, but are instead developed to solve the e-business problems of one specific company, or perhaps several enterprises with similar problems.

Application and Runtime patterns are driven by the customer's requirements and describe the shape of applications and the supporting runtime needed to build the e-business application.

Product mappings to populate the solution are based on proven implementations.

Guidelines for the design, development, deployment, and management of e-business applications also come into play.

The Patterns leverage the experience of IBM architects to create solutions quickly, whether for a small local business or a large multinational enterprise. As shown in the following figure, customer requirements are quickly translated through the different levels of Patterns assets to identify a final solution design and product mapping appropriate for the application being developed. Figure 2-1 shows a representation of the major parts of the Patterns for e-business process.
2.1.2 Use of patterns with a simple example

Chapter 3, “Requirements” on page 37, Chapter 4, “Understanding the business problem” on page 55 and Chapter 5, “Proposing a solution” on page 71 provide a worked example of pattern usage for a relatively complex business problem.

This section discusses how to apply the IBM Patterns for e-business material to assist in the definition of a solution using a simple business problem as an example. We used this same approach when working with the more complex business problem detailed in Chapter 3, “Requirements” on page 37. We have included the simple example here so that you can get a summary and roadmap of our approach, but you may choose to proceed directly to the detailed steps starting with Chapter 3, “Requirements” on page 37, if this is your preference.
Determining the Business patterns

The first step is to determine the Business pattern or patterns that are applicable. Before starting, it is necessary to have an appreciation of the current environment, including some knowledge of:

- Business goals and drivers for the organization
- Business goals, drivers and requirements for the specific initiative
- IT goals for the organization
- IT goals and non-functional requirements for the initiative

Your goals do not all have to be precisely defined, but you do need a good understanding of what it is you are going to design.

There are two broad approaches that may be used to kick off the process for making the determination of a Business pattern. Approach 1 on page 13 is more suitable for smaller problems or if there is already a good appreciation of what the solution will look like from an operational perspective. Approach 2 on page 14 is more suitable for complex problems or if there is less of an appreciation of the indicative solution from an operational perspective.

In either case, we need an appreciation of what the solution will look like from an operational perspective.


The two approaches we used to determine Business patterns in this redbook are simple, and are consistent with the processes of the Patterns for e-business site. We have provided more details of how we used the process, in an effort to help users of patterns more easily learn and understand the process.

1. Fast path based on advanced knowledge
   
   We could start by workshopping an overview diagram to identify users and functional nodes within the solution. We could look at the material that is presented on the IBM Patterns for e-business Web site and perform a look and feel match against the pattern guidance material.

   The architect may at this time move directly to a Composite pattern, if knowledge of the business need matches a Composite pattern description. It is still, in most cases, necessary to review the individual Business patterns.
and choose appropriate Application patterns. Quite simply, the selection of Application patterns to form a Composite pattern cannot be automated since an individual initiative will have different IT goals and non-functional requirements that will influence the Application pattern selection.

2. Standard approach for a custom design with multiple Business and Integration patterns

The standard approach is the approach we used in our redbook; all the steps we took are documented in Chapter 3, “Requirements” on page 37.

The starting point with this approach is to produce a high-level business description of the solution in order to find the actors and the interaction between actors and functional systems. This description is a high-level use case. If business requirements have already been documented for the solution to be designed, the task is obviously somewhat simplified and can, in some cases, already have been performed.

The following is an example of such a business description that has been extracted in a simplified form from Chapter 3, “Requirements” on page 37:
The user will use a browser to connect to the solution via the Internet. He or she will be able to browse and select products for purchase from the product catalog. Products selected for purchase are added to the shopping cart. The user can then commit the contents of the shopping cart to an order. The order is transferred to the fulfillment system.

Solution overview diagram

Whatever the method of familiarization, once we have described the solution, we can develop a solution overview diagram. The following symbols are provided in Patterns for e-business: A Strategy for Reuse, to allow a common notation to be used for this purpose.

- A rectangle: this represents a new or modified business function.
- A rectangle with two vertical lines: this represents an existing system that interfaces with the solution, but is not modified by it. This system is one of the actors.
- A picture or other icon: these represent other actors within the system.
- Connectors that link the other three symbols.
- The first step in drawing the solution overview diagram is to identify the key business functions and actors in the high-level business description. The underlined items are the actors in the solution and the items in bold are the business functions. In this example, they are:
  - User
  - Browser
  - Product catalog
- **Search** (referred to as *Product Search and Selection*)
- **Order**
- **Fulfillment** system

Figure 2-2 shows how the diagram looks after this first step: adding all actors and functional components.

The next step is to walk through all the processes described in the business description and draw the appropriate connectors detailed as interaction in the high-level business description between components. Figure 2-3 shows this view of the solution overview diagram.
Choosing the appropriate Business patterns

From the four basic Business patterns detailed in the IBM Patterns for e-business material, we now select those that are applicable to the problem at hand. Read the descriptive text for each of the Business patterns and choose those relevant.

The four patterns are described very briefly.

**The Self-Service business pattern**

Also known as the User-to-Business pattern, Self-Service addresses the general case of internal and external users interacting with enterprise transactions and data. See Figure 2-4.

![Figure 2-4 Schematic view of Self-Service business pattern](image)

- Relating to our example:
  - In our example, browsing the catalog, selecting product and committing the order are all Self-Service.

**Note:** When developing the solution overview diagram in Chapter 3, “Requirements” on page 37, it was difficult to distinguish between the Information Aggregation business pattern and the Self-Service business pattern.

The determination was made on the difference in the way the data is handled. Information aggregation was chosen since the user is using tools to handle the data directly, whereas in Self-Service, the user interacts with a business process that handles the data.

In this case, we used the focus the IBM Patterns for e-business provide on multiple sources of information being aggregated into one and complex transformation rules used in the aggregation as being the discriminators and chose the Self-Service business pattern for the user to catalog access.

**The Collaboration business pattern**

Sometimes called User-to-User, the Collaboration business pattern addresses the interactions and collaborations between users. This pattern can be observed
in solutions that support small or extended teams who need to work together in order to achieve a joint goal. See Figure 2-5.

Figure 2-5  Schematic view of the Collaboration business pattern

- Relating to our example:
  
  Our example has no use for the Collaboration business pattern.

**The Information Aggregation business pattern**

The Information Aggregation business pattern, also known as User-to-Data, can be observed in e-business solutions that allow users to access and manipulate data that is often aggregated from multiple sources. This Business pattern captures the process of taking large volumes of data, text, images, video and so on and using tools to extract useful information from them. See Figure 2-6.
Relating to our example:

We have no need for the Information Aggregation business pattern.

**Note:** Population of an initial product catalog would likely involve the Information Aggregation pattern, but this is outside the scope of our example since we have assumed that a catalog already exists.

**The Extended Enterprise business pattern**
The Extended Enterprise business pattern (Business-to-Business or B2B) addresses the interactions and collaborations between business processes in separate enterprises. This pattern can be observed in solutions that implement programmatic interfaces to connect inter-enterprise applications. See Figure 2-7.
Relating to our example:

In the simple solution we are dealing with as an example, use of the Extended Enterprise business pattern is not observed.

**Solution overview diagram including Business patterns**

Having identified the Business patterns that apply to our solution, we can now add them to the solution overview diagram. As observed, there is only one relevant Business pattern that is applicable in our simple example. That is the Self-Service business pattern used for browsing and selecting products and placing an order. See Figure 2-8 for the new solution overview.
Choosing the Integration patterns

There are two styles of Integration pattern detailed in the IBM Patterns for e-business material. These are:

- Access Integration
- Application Integration

**The Access Integration pattern**

The Access Integration pattern describes those recurring designs that enable access to one or more Business patterns. In particular, this pattern enables access from multiple channels (devices) and integrates the common services. Figure 2-9 shows an overview of the Access Integration pattern.
Relating to our example:

As our problem is relatively simple, we have no need for an Access Integration pattern.

**The Application Integration pattern**

The Application Integration pattern brings together multiple applications and information sources without the user directly invoking them. This pattern is most effectively applied when developmental efforts require the seamless execution of multiple applications and access to their respective data in order to automate a complex, new business function.

Wherever one of our subsystems accesses a backend system or another Business pattern, we are dealing with an Application Integration pattern. Figure 2-10 shows an overview of Application Integration pattern.
Relating to our example:

Application Integration is required to link the order and catalog components to the pre-existing fulfillment system.

Solution overview diagram including Integration patterns

Now that we have identified the different Integration patterns of the system, we can add them to the solution overview diagram. Figure 2-11 shows the result.
Composite patterns

When Business patterns and Integration patterns are combined to assemble solutions that perform complex business functions, certain recurring pattern combinations begin to emerge. These Composite patterns could be thought of as super patterns. Composite patterns represent commonly occurring combinations of Business patterns and Integration patterns, and typically solve major portions of functionality within a solution.

Our example business problem is extremely simple. If it were more complex, we would now determine whether the combination of Business and Integration patterns is itself a recurring pattern.

**Note:** It is questionable that Composite patterns add further value to the solution until product selection is being made. Product vendors have seen the existence of a recurring set of requirements for a particular business type and hence have developed a recurring solution to fit. The identification of Composite patterns is likely to assist in determining the fit for such a package.

As detailed in 1., “Fast path based on advanced knowledge” on page 13, it is still necessary in most cases to review the individual Business patterns and choose appropriate Application patterns. Quite simply, the selection of Application patterns to form a Composite pattern cannot be automated since each individual initiative will have different IT goals and non-functional requirements that will influence the Application pattern selection.

There are four Composite pattern types listed at the IBM Patterns for e-business Web site. This list is not extensive. More Composite patterns will be identified as IBM Patterns for e-business usage matures. There may, as in the case of the e-Marketplaces, be multiple sub-types of Composite patterns constructed for the specific business problem type with which they deal. The following table lists the four Composite patterns currently identified.
Table 2-1  Composite pattern types

<table>
<thead>
<tr>
<th>Composite pattern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronic commerce</td>
<td>Electronic commerce combines the Self-Service and Information Aggregation business patterns with the Application Integration pattern to facilitate user interaction. Additionally, an electronic commerce site can integrate supply chain management functionality into the solution through the Extended Enterprise pattern, or e-mail confirmation of orders to customers through the Collaboration pattern.</td>
</tr>
<tr>
<td>e-Marketplaces</td>
<td>e-Marketplaces are trading exchanges that facilitate and promote buying, selling and business communities among trading partners within certain industries. These solutions represent some of the most comprehensive and complex e-business applications that exist today.</td>
</tr>
<tr>
<td>Portals</td>
<td>Though designed to facilitate many variations of similar functionality, a portal solution typically aggregates multiple information sources and applications to provide a single, seamless and personalized access to users. The Composite pattern for portal applications is made up of an Access Integration pattern that facilitates functions such as single sign-on, multiple device support and personalization, plus at least one other Business pattern.</td>
</tr>
<tr>
<td>Account access</td>
<td>Account access solutions provide customers around-the-clock access to their account information. They also allow users to inquire, update, and delete information on their individual accounts. Many applications fall under this category of solutions, ranging from trading applications provided by online brokerages to account manager functions provided by utilities such as telephone companies.</td>
</tr>
</tbody>
</table>

Note: When using the IBM Patterns for e-business Web site, it looks as if it is possible to find the matching Composite pattern just by looking at the type of the Business and Integration patterns. Unfortunately, this is not the case, because the same Business and Integration patterns are present in many different Composite patterns. To find the right Composite pattern, we instead need to match up the textual description of the different Composite patterns to the requirements or perceived requirements of our customer.
Application patterns

Now that we have the Business patterns and Integration patterns selected, we choose the Application pattern that will satisfy each of the Business and Integration patterns.

**Note:** It is easy to agonize over choosing a perfect fit here. It is not that crucial. You are likely to have only an understanding of the most important goals, drivers and requirements. You will use a normal method-based approach to ensure the design is sound before anything is implemented.

Take guidance from the material, but use your own experience and judgement. Some of the characteristics given for Application patterns are at an enterprise level, other characteristics at an individual initiative level. As an example, if you read that something will not scale, you still need to consider whether this means that it will not scale for the one application integration point you are dealing with, or that it will not scale if used indiscriminately throughout a large corporate environment.

**Business and IT drivers, goals and requirements**

As stated in “Determining the Business patterns” on page 13, we need to have an appreciation of the most important business and IT drivers, goals and requirements both for the initiative we are designing for and the organization as a whole. We also need an appreciation of the existing environment with which the finished solution will integrate.

You can get some of this material from IT guidelines, policies and standards of the organization, and from any project requirement, charter or feasibility documents that have been produced. You can get other material by interviewing subject matter experts and executive sponsors within the organization.

We use this information to perform a match against the characteristics of the Application patterns so that a selection can be made.

The following business drivers have been produced to demonstrate Application pattern selection. These drivers are for the system as a whole and also for the organization for which the solution is being designed.

The business drivers are:

- Reduce costs by reducing the staff workload associated with servicing client orders
- Increase customer satisfaction by providing a Web-centered access channel allowing client organizations to have Self-Service capability
- Decrease end-to-end response order time by providing Self-Service capability
- Realize past investments by integrating with the existing order system.

The following IT drivers have been produced to demonstrate Application pattern selection.

The IT drivers are as follows.
- Must be able to scale to support additional products and users
- Easily extended for other purposes
- Integration with existing systems
- Ease of maintenance
- Buy before build

**Comparison against Application pattern characteristics**

Using the characteristics of the Application patterns as presented on the IBM Patterns for e-business Web site against the goals, drivers and requirements, we choose appropriate Application patterns. In our simple example shown in Figure 2-12, we only have the one Business pattern for which the selection of an Application pattern is required.

![Figure 2-12  Self-Service business pattern for product search and selection and order](image)

**Application pattern selection discussion**

The characteristic, easily extended for other purposes, has been matched since we may need to provide for additional access channels which are not provided for at present.
All functionality on the back-end systems hints towards the Router application pattern.

The Router pattern also provides isolation of function from the delivery channel. This enables functionality to change at some time in the future with little impact on the delivery channel. More importantly, the core functionality and data for our example initiative are in the back-end system.

The Decomposition and Agent application patterns were rejected since they provide for additional functionality not required for our business problem.

The first thought is to reject the Router pattern because the characteristics of the Web site indicate that it will not assist to improve organizational efficiency. This statement appears to have no relevance for our simple usage.

As such, the Router application pattern is seen as a suitable pattern for our solution to satisfy Self-Service access to product configuration, product search and selection and order functionality.

Figure 2-13  Self-Service::Router application pattern

**Runtime pattern**

Figure 2-14 shows the Runtime pattern as applicable to the BASE application. From the diagram, it can be seen that the integration server (Router) is implemented on the same node as the application server rather than in a separate node.
Application Integration patterns

Now we need to determine how we will link in with the existing back-end system and what Application Integration patterns will be used to link the components of our solution together.

We follow the same approach to find the Application Integration patterns we require as we used to find the Application patterns. We match our known requirements against the characteristics of the patterns to make a selection. For this simple example, the characteristics we require are:

- Non-transactional
- Not real time
- Process rather than data interface

Through comparing our needs against the characteristics on the IBM Patterns for e-business Web site, we will choose a Direct Connection application pattern as shown in Figure 2-15.
Product mapping

If we had multiple Business patterns making up the solution, we would need to join the resulting Runtime patterns together to make a total solution. As our example is a simple initiative that is solved in one Business pattern, we move directly to mapping products to the Runtime pattern. Figure 2-16 shows the selected products in our mapping.

Note: This is a sample mapping, so we have not selected and specified product versions. When implementing a real solution, we would select specific product versions to ensure that the products are supported and work together.
There is also additional information available at the IBM Patterns for e-business Web site, such as guidelines and other reference material that instructs the user on how to design, develop, and manage an e-business application.

### 2.2 Pattern application for a B2B commerce site

The patterns do not provide enough information from which to fully define a B2B commerce solution, nor do they endeavour to do so. The patterns are designed to meet 80% of most common customer requirements.

The IBM Patterns for e-business also make no attempt to replace a delivery methodology such as IBM's Global Method or Rational Unified Process. They are designed to operate alongside such a methodology.

So what do they provide us with?

The IBM Patterns for e-business significantly shortcuts arriving at a view of the architecture for most of a solution. The IBM Patterns for e-business also lessen risk by enabling the user to design a solution that is constructed on proven architectural patterns.

Following the process down to the formation of product specified runtime, and using the guidelines and best practices available, patterns provide a specification...
of the operational architecture. This may not be complete, but it certainly provides a suitable level of understanding for the construction of a proposal. Indeed, the runtime diagram produced is suitable for direct inclusion with a proposal. The logical nodes identified to allow choosing Business patterns provide a head start in identifying components within the proposal. Some of the more technical components are identified as the Runtime patterns are defined.

However, when the IBM Patterns for e-business are combined with a suitable methodology, the material can be used to provide a technical framework that can be used right through the project. They provide a natural grouping of major subsystems that is useful right through from business analysis to testing. As an example, the Business patterns identified for the project are suitable for grouping functional tests. The Application Integration patterns are suitable for grouping interface tests for integration testing.

The Composite patterns are very useful for those defining a more complex solution where the business need matches a Composite pattern description. Once the Composite pattern has been identified, it is possible to research package-based solutions that have the possibility of satisfying a large proportion of the solution in total.

For an e-commerce business need that aligns well with a Composite pattern, it is possible, with experience, to choose the Composite pattern directly from high-level requirements. This allows package investigation to commence very early in the project’s life and can run in parallel with the full architectural definition for the solution.

One benefit of IBM Patterns for e-business that should not be overlooked when producing a B2B or any other solution is the promotion of a standardized notation and terminology that is offered. Simply from an ease of communications point of view, in an organization that provides e-business solutions, using a standard terminology and notation is likely to save time, effort and hence reduce cost in the long run. As an example, if an architect talks about a population, a multi-step information aggregation-based solution for a section of an initiatives solution, all those familiar with the IBM Patterns for e-business can immediately picture how that area of the solution is structured without going through a detailed explanation.

2.3 Process and methods used in our redbook chapters

In producing worked examples throughout this document, we could have chosen to deliver artifacts that followed guidance from a particular methodology. We chose not to. It is likely that there would have been intellectual property and licensing considerations that may have hindered the release of this redbook if we
had chosen to follow one methodology. As such, we made a decision to provide sample deliverables that are typical without focusing on any specific methodology. Patterns for e-business can be used in conjunction with various design and development methodologies.

2.3.1 Our starting point

We start in Chapter 3, “Requirements” on page 37 with an imaginary request for a proposal that an imaginary business, BASE Pty. Ltd, has invited us to address.

The BASE organization has its own IT department. Along with the request for a proposal, we have imagined BASE Pty. Ltd’s IT standards, guidelines and policies, a baseline architecture for the existing IT environment, business goals and requirements for the system to be developed. The requirements also include non-functional requirements.

We present the material developed for the BASE organization, and also discuss why formal requirements are so important and how to capture them.

2.3.2 Pattern usage

This redbook provides a worked example to illustrate the use of IBM Patterns for e-business in solving a realistic business problem. We have employed the use of the IBM Patterns for e-business to work from requirement, through to an operational view of the architecture for a relatively complex problem where the solution is not known in advance.

2.3.3 Example proposal

An example proposal is constructed as a response to the request for a proposal from the BASE organization.

The proposal is constructed after the Patterns for e-business material has helped us define a solution and demonstrates the leverage from the IBM Patterns for e-business.

2.3.4 We get the job

Following from the proposal, we assume that we have gotten the job. The remainder of the book continues with the design, build and implementation activities for the solution to satisfy the BASE organization’s business problem.
2.4 The Sell-Side Hub composite pattern

The e-Marketplace, Sell-Side Hub composite pattern is the Composite pattern that was recognized in Chapter 4, “Understanding the business problem” on page 55 as a match for the e-commerce business our fictitious company, BASE Pty. Ltd wishes to undertake.

2.4.1 Introduction to e-Marketplaces

The definition of e-Marketplaces in this section is as described on the IBM Patterns for e-business Web site.

e-Marketplaces are trading exchanges that facilitate and promote buying, selling and business communities among trading partners within certain industries. These solutions represent some of the most comprehensive and complex e-business applications that exist today. There are three types of e-Marketplaces:

- Trading Exchange
- Sell-Side Hub
- Buy-Side Hub

In a Sell-Side Hub, the seller owns the e-Marketplace and uses it as a vehicle to sell goods and services to prospective buyers across the Web. As shown in Figure 2-17 on page 34, the Sell-Side Hub composite pattern includes the following patterns:

- An Access Integration pattern that helps provide a unified customer interface
- A Self-Service business pattern that allows users to browse through a catalog, create an order and place an order with the hub
- An Information Aggregation business pattern that is used to create the e-Marketplace catalog from the multiple sources of suppliers' product files, pricing files and advertising literature, etc.
- Application Integration patterns that integrate the Business patterns which are a part of the Sell-Side Hub

In addition to these basic functions, there can be several variations on this pattern. These include:

- Adding a Collaboration business pattern that allows for and enables auctions, reverse auctions and other collaborative buying functions
- Adding an Information Aggregation business pattern that will help integrate and present a unified catalog that combines raw catalog data with expert
advice, product comparisons and recommendations that can be pulled off public Internet sites

- Adding an Extended Enterprise business pattern that will integrate the Sell-Side Hub with external service providers such as a financial institution to handle credit processing or shipping company to handle the physical delivery of goods

![Sell-Side Hub composite pattern](image)

Figure 2-17  Sell-Side Hub composite pattern

### 2.4.2 Fit for our imaginary business problem

In our example, we start from the assumption that a Sell-Side Hub is the best fit for BASE. This is because the description of the Sell-Side Hub is an excellent match for the description of what our fictitious company, BASE Pty. Ltd is attempting to achieve.

BASE Pty. Ltd wishes to own an e-Marketplace and use it as a vehicle to sell goods to prospective buyers across the Web.

BASE has an existing access integration infrastructure in the way of a Web security solution that offers single sign-on to users of BASE online applications.
The solution will supply Self-Service functionality, allowing users to browse through a catalog, search and select products and create orders.

The solution makes use of existing disparate legacy sources of data to construct the catalog.

There is a collaboration component to the solution that allows Customer Service Representatives (CSRs) to interact with users experiencing difficulty. Another collaboration component provides the ability for customers who wish to have someone approve an order before it is submitted contacted for that approval.

From this simple description of what is required to satisfy the BASE business need, the alignment with the description of the Sell-Side Hub is obvious. There is no need for the Extended Enterprise business pattern, which is optional. Therefore, the e-Marketplace, Sell-Side Hub composite pattern is a good fit for the business BASE wishes to enter.

Given the above, we decide to implement a Sell-Side Hub to solve BASE’s online order capture needs.

**Note:** This preliminary match between the high-level description of BASE’s requirements and the Sell-Side Hub composite pattern allows us a starting point for our architecture and design process. As we work through the process of applying the Patterns for e-business, we will continue to refine and test our decision in order to verify that the choice of a Sell-Side Hub composite pattern is correct.

### 2.4.3 More general fit for business problems

In coming up with the requirements for our imaginary customer, BASE Pty. Ltd, the authors of this redbook used requirements they had been asked to satisfy when constructing e-commerce solutions for real customers. As such, the requirements, while not complete, are typical for such a business need. As the Sell-Side Hub satisfies our imaginary customer’s needs, it will satisfy a real customer’s needs.
Chapter 3. Requirements

This chapter describes our imaginary customer and the requirements they have for their e-business solution. We provide the following.

1. A refined set of business requirements that need to be satisfied by the solution required by our customer (Section 3.2.5, “Functional requirements” on page 43).

2. Non-functional requirements that will assist in constructing the more physical elements of the solution (Section 3.2.6, “Non-functional requirements” on page 47).

3. Principles, policies, standards and guidelines that will be followed in order to design a solution suitable for implementation in the customer’s corporate environment (Section 3.2.7, “Standards and policies” on page 50).

4. A view of the current IT environment with which the solution will integrate (Section 3.2.4, “Existing environment” on page 41).
3.1 About requirements

In this section, we briefly describe

- What requirements are,
- The necessity of requirements
- The gathering of requirements

3.1.1 Definition of key terms

The terms that are used when referring to the requirements phase include the following.

**Functional requirements**
Functional requirements describe in detail what the solution is expected to do in terms of functionality. The customer wants the system to satisfy certain business needs of their organization. The customer's wish list is the list of requirements.

**Non-functional requirements**
Non-functional requirements are those aspects of a business system that define the functionality of a system indirectly. They are general attributes the solution must satisfy.

The most important non-functional requirements are performance, availability, security, systems management and constraints.

Like functional requirements, non-functional requirements have to be measurable.

**Principles, policies, standards and guidelines**
These are general rules that an organization adheres to when using or deploying IT and business assets.

**Requirements and specifications**
There is a difference between requirements and specifications: requirements are what the customer wants to see in a solution, whereas specifications define what a system will actually do once it is built. In an ideal world, the specifications should align completely to the requirements, but sometimes, there are issues that prevent the fulfillment of all requirements, for example when the designated product does not provide a certain function that is needed to met the specification.
3.1.2 The importance of requirements

Why is the definition of requirements so important for any project?
Without a proper set of requirements, a team of architects would not be able to
design a solution that fits the customer's needs. Furthermore, without a
specification document that derives from those requirements, it would not be
possible to determine, in a methodical way, whether the produced solution meets
the specification or not.

3.1.3 How to gather requirements

The process of gathering functional requirements differs from the process of
gathering non-functional requirements.

Gathering functional requirements
Normally, functional requirements are gathered through one or more workshops
with the customer. Usually, your designated project manager and an IT architect
will work together with a group of the customer's employees that will consist of
their project manager and some people from the IT and other departments that
are to be involved in the project.

We have assumed that our sample project has reached a phase where basic
requirements are already stated, a high-level proposal has already been written
by the IBM sales team, and the product has already been chosen by the
customer. See 3.2.1, “Assumptions about the customer situation” on page 40 for
more details. Given these assumptions, we would proceed to requirements
gathering by contacting the IBM sales team to get all information they have about
the customer's situation, and then elaborate on any open issue during
workshops with the customer.

Note: Sometimes, not all of the requirements can be met by the solution.
Sometimes, they can be met, but only by spending an excessive amount of
money and/or effort. Therefore, it is helpful to have the customer establish
priorities on their requirements. This way, if there are indeed problems in
delivering the solution with all the requested features, you know at least which
are the most important features to your customer and can concentrate on those.

Gathering non-functional requirements
Non-functional requirements can be gathered through different approaches:
- Let the customer fill out questionnaires
- Hold interviews
Hold one or more workshops
Research existing documents

You might first want to ask the relevant customer people (IT manager, customer's project manager, other business and IT contact persons) to fill out a questionnaire and then hold a workshop in which you elaborate on those questions and their respective answers. By filling out the questionnaire, the customer knows what you will be talking about in the workshop, and they will make sure to have the right people to ask close at hand.

You can also try and discuss these topics as you go in the workshops that define the functional requirements. But if you have not started the gathering of the non-functional requirements with a questionnaire, you risk the customer not being prepared for the questions you will ask.

Note: It is often difficult for non-IT people to understand the implications of some of the non-functional requirements. For instance, people, when first asked, tend to want a full system availability of twenty-four hours, seven days a week. But such a high availability - without any possibility for maintenance or downtime - will make the solution much more expensive. In a workshop, you have the chance to point out these implications right from the start and save yourself and your client a lot of time and effort afterwards.

3.2 The requirements of our customer

To illustrate the use of the patterns, we drew up an imaginary customer, BASE Pty. Ltd. Of course, this customer is in many ways an ideal customer which does not exist as such in reality, but, given the short time available to write this redbook, we had to make some assumptions in our favor.

3.2.1 Assumptions about the customer situation

The following is already in place for BASE Pty. Ltd.

1. IBM has sold the benefits of Java 2 Platform, Enterprise Edition (J2EE) to BASE and convinced BASE to adopt IBM WebSphere Application Server as their standard runtime platform.

2. IBM has constructed a high-level proposal detailing the competitive advantage to be gained by BASE in providing an online selling capability to their business. This proposal sold the use of IBM WebSphere Commerce Business Edition as the J2EE compliant solution that will operate on BASE's standard runtime platform.

3. BASE has an existing Web presence.
4. BASE has existing customer and staff identity management strategies, processes and systems in place.

5. The BASE organization has existing online security processes and an enabling infrastructure that provides authentication and coarse-grained authorization capability to online applications.

6. BASE has invited us to respond to a request for proposal for the design and development of their online store. The requirements of this online store will be documented in this chapter.

3.2.2 Short business profile

BASE Pty. Ltd is a company that is located in England and services laboratories and health service organizations throughout Europe. It is in the business of manufacturing:

- Low cost, high-volume consumable items, for example disposable syringes and test tubes
- A limited range of complex analytical equipment, configured and built to order

The company was founded in 1982. It has a future goal of expanding its geographic coverage from Europe to world-wide to have a truly international presence.

3.2.3 Business goals

With their newly projected e-business solution, BASE plans to:

- Reduce costs by reducing the staff workload associated with servicing client orders. This should be achieved using an Internet-based provisioning system.
- Increase customer satisfaction by providing a Web-centered access channel, allowing client organizations to have Self-Service capability to place orders and review order status via the Internet.

3.2.4 Existing environment

This section describes the existing environment at BASE.

Business perspective

- All functionality to be enabled by the solution is currently in place for manual access channels, that is, the solution will offer functionality currently available in the form of paper-based forms and manual processes.
- The BASE organization has multiple fulfillment centers located throughout Europe to enable speedy servicing of customer requests.
IT perspective
An overview of the current IT environment of BASE can be seen in Figure 3-1. It will be described in detail in Chapter 6, “Operational overview” on page 155.

- BASE Pty. Ltd has an existing product inventory and a separate pricing system.
- The BASE organization employs the use of WebSphere MQ, providing inter-system communications between corporate systems.
- Detailed information about the products and the matching pictures for the printed catalog are stored in a product management system (PMS). This system has been used since for producing printed catalogs.
- BASE already has an array of highly available Web servers that host their Web presence and provide public access to a part of their technical documentation (the technical documentation repository). There is also a secure access area which holds documentation of a competitive, confidential nature that may only be viewed by BASE customers. This repository is also fed from the PMS.
- A corporate customer identity system contains identity information for customers and customer representatives BASE deals with. The company also has an additional identity system held within the corporate network that contains employee details.
- The BASE organization has an existing authentication, authorization and registration capability used by the technical documentation repository.
3.2.5 Functional requirements

In this section, we discuss the functional requirements of the solution for BASE Pty. Ltd. For easy reference, the requirements are structured into several subsystems: Catalog, Member, Order, Back-end integration, Customer care and Marketing.

Catalog subsystem

- Users that have not logged in (so-called anonymous users) will be able to browse the catalog without being able to see the prices. This is done to attract potential customers. This way, they can easily study the range of products BASE has to offer. If they are interested, they can get into a business contact with BASE Pty. Ltd.
- A catalog search capability will be provided to the users to allow them to find products without having to browse the catalog.
- Authenticated users will have the ability to browse a catalog. A catalog for perusal can be selected based on:
  - A contract
  - A combination of contracts
  - The catalog in full

Figure 3-1  The IT environment of BASE before the deployment of the projected solution
Customers can negotiate with BASE for certain conditions for the purchase of certain products. The outcome of these negotiations are captured in a contract between BASE and their customer.

- The provided solution will interface with a custom-built external product configurator application for the configuration of complex analytical equipment that is configured on a per-use basis.
- BASE is planning to connect their catalog system to one or several of their suppliers in the future.

**Member subsystem**

- New users will be able to self-register (that is, apply for a user ID) in order to gain access to the ordering system if the organization they belong to is an existing customer with BASE Pty. Ltd.
- New registration requests will be validated by a user organization representative before individual user access is provided. This validating role is called the buyer administrator.
- A client organization's buyer administrator will be validated by a BASE employee before access is provided.
- The solution will make use of the single sign-on capability provided by the existing BASE corporate solution. Customer representatives who have been supplied with user ID and password to access the technical documentation repository will be able to use these credentials to connect to the online shop system.
- The system will also use the user identity information from the existing corporate user identity repositories.
- Because BASE is dealing with customers all across Europe, the site will support multiple languages and multiple currencies. These attributes will be stored in the user's profile. Changes will become effective the next time the user logs on. The first release will only contain the English and German language.

Table 3-1 gives a summary of the different user types that will deal with the system.

<table>
<thead>
<tr>
<th>User type</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anonymous user</td>
<td>May only browse the catalog, may self-register</td>
</tr>
<tr>
<td>Buyer</td>
<td>Standard authenticated user, may place orders for his/her organization</td>
</tr>
</tbody>
</table>
Order subsystem

The solution will provide client organizations with the ability to place an order through the Internet. It will service:

- Requests from users from client organizations wishing to place an online order via Web browser
- Requests from compatible client organization systems that will interoperate with it to place an online order (using a machine interface)

Users must authenticate in order to place an order.

Users will have the ability to place standing orders. Standing orders are orders with consistently the same shopping cart which will reoccur over time, following a given schedule.

Users will have the ability to create template orders. Template orders are based on previous orders that can be modified to satisfy the current need. They can then be named and saved for further use.

Users will be able to use the bulk order form. Bulk order forms let users who already know what they need bypass the catalog browsing. This is done via a Web page with a list of entry fields where users enter several product stock keeping units (SKUs) at the same time. By submitting the bulk order form, all products are put into the user's shopping cart.

Order approval functions will be as follows:

- Approval must be provided by a client organization's buyer approver for orders over a pre-defined monetary value.
- The buyer approver will have the ability to configure the trigger level for approval of orders for their company.
- The system will notify the buyer approver as well as the buyer:
  - Every time an approval is necessary
  - Every time an approval has been given
  - Every time an order has been rejected by the buyer approver
Order status functions will be as follows:

- Users will be able to view the status of current and past orders. This also includes orders placed through other channels.
- Authorization control will provide the ability for a standard buyer to see all orders placed by that user.
- Buyer approvers will have the ability to access all orders placed by their organization, regardless of the channel that was used for placing the order.

Machine interface features will be as follows:

- The functionality offered by the machine interface will be limited to placing an order and checking the status of a current or past order.
- The following data will be delivered to the client organizations using the cXML protocol:
  - Product information
  - Pricing information
  - Order request
  - Order status request
  - Error information for the configuration and operation of the machine interface
- Future requirements may include integrating with the OCI protocol.

Orders are paid by invoice. There will be no online payment facility.

Back-end integration subsystem

- Pricing information will be retrieved from the pricing system within the BASE organization.
- Inventory information will be retrieved from the inventory system within the BASE organization.
- Fulfillment capability and order status information will be retrieved from the fulfillment system within the BASE organization.
- Detailed product information and product images will be retrieved from the product management system within the BASE organization.

Customer care subsystem

- Users will be able to interoperate with BASE Customer Service Representatives (CSRs) via text-based chat for support and trouble shooting.
- CSRs will be able to view user interaction with the system to assist users.
- Only non-anonymous users will be able to use the customer care facility.
CSR viewing of user interaction will only be granted on a per occurrence permission from the user.

CSRs will have the ability to perform all interactions on behalf of users.

Marketing subsystem

- The system will produce periodic marketing reports. These reports will provide information allowing targeted advertising based on customer usage and other customer activity.
- The frequency of the report generation will be configurable.

3.2.6 Non-functional requirements

In this section, we discuss the non-functional requirements of the solution for BASE Pty. Ltd. The non-functional requirements are structured into these subsystems: Capacity and performance, Availability, Security, Systems Management, Scalability, Maintainability, Manageability, System usability, Data integrity and failure recovery.

Capacity and performance

- The total user base is 10,000.
- In normal operating conditions, the number of concurrent users will be 20.
- In abnormal operating conditions (for example, near the end of the fiscal year), the concurrent user base will be 100.
- The estimated Internet connection speed of the users is 128 kbit/sec.
- Response times:
  - 5 second read response times in normal operating conditions
  - 10 second read response times in normal operating conditions
  - 8 second write response times in abnormal operating conditions
  - 16 second write response times in abnormal operating conditions
- Product displays expected in one day: 1500.
- The ratio between dynamic versus static pages is an estimated 20 to 1.
- Average page size: 30 KB.
- Average static file size: 10 KB.
Availability

- The complete system will be available during normal business hours across European time zones.
- The Web servers will operate 24 hours a day, 7 days a week.
- The Web servers will have a 99.5% availability to ensure a meaningful error message is provided to the users if the shop application is unavailable.

Security

- Authentication will be controlled by the corporate security system that is already in place.
- The solution will provide single sign-on capabilities with the existing technical documentation repository server.
- There will be interactive users as well as batch users (that is, external systems using the machine interface of the solution).
- User sessions will time-out after 30 minutes of inactivity. This figure will be configurable.
- A single user will not be able to use multiple concurrent connections to the system.
- User accounts will be locked after five invalid login attempts. Unlocking will have to be done by the user’s buyer administrator.
- Invalid login attempts will be logged for auditing purposes.
- Because of the competitive nature of the transactions between BASE Pty. Ltd and its customers, Secure Socket Layer (SSL) encryption will be used throughout the user session, starting with the login screen.
- The encrypted SSL transactions will use a key length of 128 bit or shorter, if the end user’s browser does not support a higher key length.
- An anonymous user’s browsing session will not be encrypted since this information is open to anyone anyway. This will save computing power for the system.

Systems management

- Interfaces from existing systems providing pricing information, product information, inventory, fulfillment capability and order status information will be batch-based.
- There is a weekly backup process in place on Saturday night which requires the application servers to be brought down. The downtime will be between 2:00 p.m. and 4:00 p.m.
The messaging and integration between enterprise applications is monitored by means of the IBM Tivoli Monitoring suite.

System management problems and issues are tracked in a custom-built problem tracking database.

Logging
- Errors must be captured in log files for troubleshooting.
- The system must be capable of capturing all interaction between components and users for contract tracing of error conditions.
- The level of information must be configurable to obtain a balance of everything for error resolution and key elements only for normal operation.
- All updates must be captured for auditing purposes.

Scalability
The system will be able to scale to support additional products in the catalog as well as additional concurrent users and additional geographies.

Maintainability
The solution will be compliant with industry standard technologies.

Manageability
The frequency of batch updates to and from interfacing systems will be configurable for each interface.

System usability
The solution will support the following Web browsers (cookies need to be enabled):
- Internet Explorer 5.01 and above
- Netscape Communicator 4.7 and above
- Mozilla 1.0 and above
- Opera 5 and above

Data integrity and failure recovery
- Mean Time To Recovery (MTTR) for normal failure condition will be no greater than one hour (backup restore time).
- MTTR for disaster condition will be no greater than 12 hours (setting up new machine, installing software, restoring backup).
- Mean Time Between Failure (MTBF) for normal failure condition during normal hours of operation will be no less than 200 hours for the Web servers.

Note: It will be less than that for the overall solution.
In the following sections, every notable standard within your customer’s organization should be recorded. We will only provide a rough overview here.

Business structure standards
- The solution will currently only support European users, with an option to expand this to worldwide users. Therefore, availability is most important during extend European business hours.
- The system will support multiple languages and multiple currencies.
- The solution will be hosted in the computer center of the headquarters of BASE Pty. Ltd. No second hosting location is planned at the moment.

Platform standards
BASE Pty. Ltd uses the following systems in their IT production environment:
- IBM zSeries Host Server
- IBM pSeries Servers on AIX 5.1
- Intel Pentium III based servers on Windows 2000 Server
- Intel Pentium III and Pentium II based desktop computers as clients, using Windows NT4.0 SP6a and Internet Explorer 5.5 SP2 as a Web browser

Note: The MTBF can be calculated as follows:

\[ \text{Availability} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}} \]

or

\[ \text{MTBF} = \frac{A \times \text{MTTR}}{1 - A} \]

The projected solution has an availability of 99.5% (A=0.995) for the Web servers and a MTTR of 1h in normal failure conditions.

This gives us: \( x = 199 \) h, which means: to achieve the demanded 99.5% availability, the system must be able to run for at least 200h with only 1h downtime or approx. 2400h with 12h downtime in disaster condition.

When referring to availability, it is popular to use the “nines” such as “three nines” for an availability of 99.9%. However, few people realize the implication of this. In the example above, 99.9% availability would mean 41 days with only 1h downtime, 99.99% would mean more than one year with 1h downtime.
Database: IBM UDB DB2
WebSphere MQ for inter-host communication
IBM WebSphere Application Server is the platform of choice for future applications
IBM Tivoli Management Suite is used as the systems management platform

Only IBM pSeries servers with AIX 5.1 or higher are allowed in network segments other than the intranet.

Application standards
The following standards apply for any application within the BASE Pty. Ltd IT environment:

- ERP system: BASE Pty. Ltd is not using a single ERP system. Instead, the functions of an ERP system are made up of different applications, for example pricing, inventory, fulfillment, etc.
- Product management system: custom-built software running on Windows 2000 Server using C++ as development language and IBM UDB DB2 as database.
- Product inventory implemented as a custom-built DB2 application on a zSeries server.
- Pricing system: custom-built J2EE application implemented on an AIX based WebSphere system.
- Technical documentation repository: custom-built J2EE document server application implemented on a AIX based WebSphere system.
- Authentication and authorization: IBM Tivoli Access Manager for e-business.
- External product configurator: custom-built J2EE software on an AIX-based WebSphere system.
- Development: Rational ClearCase is used as a source code version control system.

Technical standards
- Network standard: 100 megabit Ethernet
- Firewall topology standards at BASE Pty. Ltd: the segments of the network consist of the following.
  - De-militarized zone (DMZ)
  - Application zone
  - Database zone
  - BASE Intranet
Note: In a real-life customer situation, one would also assess which technical standards are in place regarding:

- Access control and security
- Development languages
- Electronic Data Interchange (EDI)
- Electronic Fund Transfer
- Public Key Infrastructure (PKI)
- User interface
- EAI layer/architecture

and the like. Because this is not relevant in the context of this redbook, we will not discuss it any further.

3.2.8 Operational preparedness

Operational preparedness describes to what degree the organization is able to cope with the changes the new solution brings, from a business perspective.

Catalog data

- The catalog data and its structure is stored in the product management system (PMS).
- Part of the data also comes from the inventory system and the pricing system in XML format.
- The products are grouped in five levels of categories. The hierarchical form of the categories follows a tree structure.

Site content (images, HTML)

- Marketing information and product images are stored in the PMS. This system has been used previously to produce printed catalogs.
- BASE uses a unique identifier (the stock keeping unit, SKU) to assign the product data of the PMS to the data of the inventory and the pricing system. A new part is being added to the PMS by the BASE developers to enable XML output of the data.

Process

Note: In a real-life customer situation, one would also elaborate on the processes of:

- How and how often the catalog data is updated
- Accepting and processing customer orders
- Changing and canceling customer orders
- How the fulfilment system works
and the like. Because this is not relevant in the context of this redbook, we will not discuss it any further.

**IT infrastructure**
It is important to understand and document the IT infrastructure to be used. This includes the following.

*Development and test environments*
- Development will be implemented on Windows 2000-based servers and workstations, mainly at the customer’s site.
- Deployment and testing will be implemented on an AIX-based server at the customer’s site.

*Hosting*
The solution will be hosted at the computer center of the main site of BASE Pty. Ltd, England.

*Staging*
**Note:** WebSphere Commerce supports a staging environment where changes to the code or the data can be evaluated before they are actually deployed to the production system.

A staging environment is not planned at this time.

*Maintenance and support*
BASE Pty. Ltd has personnel with network, server and database administration skills to maintain and administer the system later on.

*Customer characteristics and attitude*
Within your team, you should also try to capture the overall customer characteristics and attitude to the solution. This includes getting answers to intangible questions such as:
- How involved is the customer?
- Are the people in favor of your proposed solution?
Using the detailed requirements we discussed in Chapter 3, “Requirements” on page 37, we now move on and start architecting a solution using the Patterns for e-business. We follow the process discussed in Patterns for e-business: A Strategy for Reuse, by Jonathan Adams, Srinivas Koushik, Guru Vasudeva, George Galambos, published by IBM Press, ISBN: 1-931182-02-7. This chapter discusses the following steps.

1. Identifying actors and functional components of the business problem and their interaction (Section 4.2, “High-level business description” on page 57)
2. Developing a solution overview diagram (Section 4.3, “Solution overview diagram” on page 58)
3. Assigning Business patterns to the solution overview diagram (Section 4.4, “Choosing the appropriate Business patterns” on page 60)
4. Assigning Integration patterns to the solution overview diagram (Section 4.5, “Choosing the Integration patterns” on page 66)
5. Determining the type of Composite pattern that is applicable here (Section 4.6, “Finding the appropriate Composite pattern” on page 69)
4.1 Determining the Business patterns

Now that we have collected our customer's detailed requirements, we can start determining which are the Business patterns to apply to this case.

4.1.1 Introduction

This chapter will discuss the conceptual level of the solution. As can be seen in Figure 4-1, this part of the architecture consists of the development of the conceptual model and the outline of the custom solution.

Inputs for these concepts are the requirements of the customer. The output of this chapter will be the suitable Composite patterns, Business patterns and Integration patterns for the solution.

![Figure 4-1 Process overview](image-url)
4.2 High-level business description

Note: To determine the Business patterns, we used the fast path approach of the IBM Patterns for e-business. This approach usually only captures about 80% of the solution and should therefore be backed up by a methodical approach. The following description only takes into account the standard buyer user, not any of the other user types mentioned earlier in this chapter. Describing their interaction with the system would not add anything to the high-level solution overview diagram we are about to draw, and describing their interaction with the system would add too much complexity to the example. We will therefore disregard this in this chapter.

In 3.2, “The requirements of our customer” on page 40, we discussed the requirements of our imaginary customer, BASE Pty. Ltd. We will now start from this basis to develop a solution overview diagram. The underlined items are the actors in the solution and the items in bold are the business functions. To do this, we must first give a high-level business description of the solution to define the actors and the interaction between them.

The human user will use a browser to connect to the solution via the Internet. If the user does not already have a user ID and a password, he or she may register with the site. This registration will trigger an approval notification to a higher-level user at his organization.

Once approved, the user will be able to log in to the site. He or she will be able to browse the product catalog and use the technical documentation repository which is referred to by the catalog pages. The catalog gets its data from various systems out of the BASE Pty. Ltd back end:

- Product management system (detailed descriptions and images),
- Periodical updates on the inventory of products,
- Pricing.

The technical documentation repository also gets its data from the product management system.

Apart from just browsing the catalog, the user can also use the search function within the catalog to find the products he or she needs. If the user plans to buy a more sophisticated, configurable product, he or she may configure it by using the product configurator.

At any point in the buying process, the user may want to use the customer care chat function to connect to the sales personnel. Issues like the availability of a certain product can be discussed there.
After choosing or configuring products, the user can add these to the shopping cart. He or she can then transform this into an order and close the order. The system checks whether an order approval is necessary. If it is, the buyer approver receives a notification via e-mail. When the order is approved, it is transferred to the fulfillment system.

Apart from the human user, there is also a machine user that can interact with the projected system.

4.3 Solution overview diagram

Once we have described the solution in this way, we can develop the solution overview diagram for the case. We will use the following symbols in the diagram.

- A rectangle: this represents a new or modified business function.
- A rectangle with two vertical lines: this represents an existing system that interfaces with the solution, but is not modified by it. This system is one of the actors.
- A picture or other icon: this represents other actors within the system.
- Connectors that link the other three symbols.

The first step in drawing the solution overview diagram is to identify the key business functions in the high-level business description above:

- Human user
- Browser
- Registration (referred to as Customer registration and preferences)
- Notification
- Product catalog
- Technical documentation repository
- Product management system
- Inventory
- Pricing
- Search (referred to as Product search and selection)
- Product configurator
- Customer care chat (referred to as Customer assistance)
- Order (this contains the submittal of the shopping cart as well as access to order status updates)
- Notification
- Fulfillment system
- Machine user

Figure 4-2 shows how the diagram looks after adding all actors and functional components to it.
For Figure 4-3, we walked through all the processes described in the business description and drew the appropriate connectors between the components.

**Note:** As you might notice in Figure 4-3, the customer assistance is not linked to anything other than the Internet, and yet, later on, it will be accessible from every place during the buying process. That is because it is our understanding that customer assistance works like a hub that is enabling the collaboration functionalities. It enables CSRs and users to interact, but it will not interact with other parts of the system itself. Please refer to Chapter 10, “Customer care” on page 273 for more details.

Figure 4-2  Solution overview diagram, Step 1: actors and functional components
4.4 Choosing the appropriate Business patterns

In the Patterns for e-business, four basic Business patterns have been identified:

- Self-Service
- Collaboration
- Information Aggregation
- Extended Enterprise
We will discuss these Business patterns now in detail and see how each of them relates to the solution we are building for BASE Pty. Ltd.

4.4.1 The Self-Service business pattern

In a Self-Service business pattern, a direct interaction between individuals and a business can be observed. Individuals (later on simply called users) can be customers, employees and business partners. Business is everything from small companies, public authorities, to large enterprises. Therefore, the Self-Service pattern is also referred to as the user-to-business pattern.

Schematic view

Figure 4-4 shows a schematic view of the Self-Service business pattern: the user interacts directly with the business systems.

Relating to our example

In our example, we can think of Self-Service as the user selecting a product from the catalog and creating an order. But many more actions can be regarded as Self-Service:

- Selecting a product from the catalog and creating an order
- Using the product configurator to configure a complex product
- Looking up the status of previous orders
- Registering with the system and updating one’s account information

4.4.2 The Collaboration business pattern

A Collaboration business pattern describes interactions between users. A collaboration system acts as a hub for users to work together to achieve a joint goal. Therefore, the collaboration pattern is also referred to as the user-to-user pattern.

Schematic view

Figure 4-5 shows a schematic view of the Collaboration business pattern: the collaboration hub enables users to directly interact with one another.
4.4.3 The Information Aggregation business pattern

The Information Aggregation business pattern can be observed whenever users are using tools to extract information out of large volumes of data, often data from multiple sources. Therefore, the Information Aggregation pattern is also referred to as the user-to-data pattern.

Schematic view

Figure 4-6 shows a schematic view of the Information Aggregation business pattern: the aggregator hub enables users to directly interact with the data.
Chapter 4. Understanding the business problem

4.4.4 The Extended Enterprise business pattern

The Extended Enterprise business pattern can be observed when one business process interacts with another business process in a different company. Therefore, the Extended Enterprise pattern is also referred to as the business-to-business pattern.
**Schematic view**

Figure 4-7 shows a schematic view of the Extended Enterprise business pattern. A business connects to other businesses through a set of business rules.

![Schematic view of the Extended Enterprise business pattern](image)

**Relating to our example**

In the BASE solution, the Extended Enterprise business pattern cannot be observed.

**Note:** One might assume that the machine interface of the BASE solution is an Extended Enterprise pattern. Basically, there is a system talking to another system, but in this case, there is actually a user at a browser performing interactions with the procurement system.

This connection does not meet the criteria of an Extended Enterprise business pattern, because the procurement system of the client organization is actually using the Self-Service infrastructure that is built for the human user. Therefore, this is not a process in its own right, but simply a different implementation of the Self-Service process.

**Extending the solution overview diagram**

Having identified the Business patterns that apply to our solution, we can now add them to the solution overview diagram. They are as follows.

1. Information Aggregation: the product catalog and the technical documentation repository getting data from the back-end systems.
2. Self-Service: using the external product configurator; searching and/or selecting a product from the catalog; placing an order.
3. Collaboration: notifying the user of approvals and other workflow events; giving customer assistance to the user via text-based chat.

4. Self-Service: user registering him/herself or updating his/her account.

Figure 4-8 shows the result.
4.5 Choosing the Integration patterns

In the Patterns for e-business, two Integration patterns have been identified:

- Access Integration
- Application Integration

We will discuss these Integration patterns in detail.

4.5.1 The Access Integration pattern

Whenever a certain solution design allows user access to Business patterns, the Access Integration pattern can be observed. This pattern integrates the required services to support a consistent user interface even across multiple access channels (devices).

Schematic view

Figure 4-9 shows a schematic view of the Access Integration pattern: users access applications and data with the help of access integration services.

Relating to our example

In our example, Access Integration is responsible for identifying the user based on BASE’s customer and employee directories. It provides single sign-on to every function of the site.
4.5.2 The Application Integration pattern

Application Integration patterns can be observed between multiple Business patterns, to integrate the Business patterns, or within a single Business pattern, to integrate applications with data.

**Schematic view**

Figure 4-10 shows a schematic view of the Application Integration pattern. Applications make use of application integration to access other applications or other data.

![Figure 4-10 A schematic view of the Application Integration pattern](image)

**Relating to our example**

Application Integration will combine the Business patterns to make them work together as one solution.

Whenever one of our subsystems accesses a back-end system or another Business pattern, we are dealing with an Application Integration pattern:

1. Product management system to product catalog
2. Product management system to technical documentation repository
3. Inventory to product catalog
4. Pricing to product catalog
5. Order with fulfillment
6. Product catalog with product search and selection
7. Product search and selection with order
8. Product catalog to product configurator
9. Product search and selection with product configurator
10. Order to user notification
11. Order with customer registration and preferences
12. Customer registration and preferences to user notification

4.5.3 Extending the solution overview diagram

Now that we have identified the different Integration patterns of the system, we can add them to the solution overview diagram. Figure 4-11 shows the result. The numbered connectors between the components show where Application Integration patterns can be observed.

Figure 4-11  Solution overview diagram, Step 4: the Access Integration pattern and the Application Integration patterns (numbers 1 to 12) have been added.
4.6 Finding the appropriate Composite pattern

Now that we have all the pieces together, we can draw conclusions about which Composite pattern we are dealing with.

**Note:** From the information given at the patterns Web site, it looks as if it is possible to find the matching Composite pattern just by looking at the type of the Business and Integration patterns. Unfortunately, this is not the case, because the same Business and Integration patterns are present in many different Composite patterns. To find the right Composite pattern, we instead had to match up the textual description of the different Composite patterns to the requirements of our customer.

Comparing the requirements of our customer to the textual descriptions of the different Composite patterns from the patterns Web site, we concluded that the Composite pattern that can be seen as a solution for BASE’s business problem is the Sell-Side Hub composite pattern.

4.6.1 Sell-Side Hub composite pattern: a definition


In a Sell-Side Hub, the seller owns the e-Marketplace and uses it as a vehicle to sell goods and services to prospective buyers across the Web. The Sell-Side Hub composite pattern, shown in Figure 4-12 on page 70, includes the following:

- An Access Integration pattern that helps provide a unified customer interface.
- A Self-Service business pattern that allows users to browse through a catalog, create an order, and place an order with the hub.
- An Information Aggregation business pattern that is used to create the e-Marketplace catalog from the multiple sources of suppliers’ product files, pricing files, and advertising literature.
- Application Integration patterns to integrate the Business patterns that are part of the Sell-Side Hub.
In addition to these basic functions, there can be several variations on this pattern, including the following:

- A Collaboration business pattern can be added to enable auctions, reverse auctions, and other collaborative buying functions.
- An Information Aggregation business pattern can be added to help integrate and present a unified catalog that combines raw catalog data with expert advice, product comparisons, and recommendations pulled off the public Internet site.
- An Extended Enterprise business pattern can be added to integrate the Sell-Side Hub with external service providers, such as a financial institution to handle credit processing, or a shipping company to handle the physical delivery of goods.
Proposing a solution

This chapter describes how to leverage the Patterns for e-business to produce both a proposal and a more detailed specification of an architectural definition to solve a business problem.

Rather than simply list the tools, techniques and methods applied to arrive at an architectural definition, this chapter provides worked examples for a solution to satisfy the need of our imaginary business, BASE Pty. Ltd., whose requirements are documented in Chapter 3, “Requirements” on page 37.

The chapter is comprised of two main sections.

► The first section, 5.1, “Our starting point” on page 72 provides an introduction to our starting point in this chapter, including details of what has been provided from other chapters and what assumptions we have started with.

► To add a further touch of realism to the examples, the second section of this chapter, 5.5, “Constructing the proposal” on page 129 is worked around deliverables for a solution outline that is produced in the context of a business proposal.

This chapter does not produce a fully defined runtime architecture. This level of elaboration is delivered in Chapter 6, “Operational overview” on page 155.
5.1 Our starting point

As detailed in Figure 5-1, we enter this chapter part way through defining a solution to satisfy BASE Pty. Ltd’s ordering problem. This section details what has been provided from Chapter 3, “Requirements” on page 37.

The shaded area on the left in Figure 5-1 contains tasks that have already been performed. This early analysis based on the BASE’s business requirements drove the selection of the Business patterns that will be required to construct a solution to meet BASE’s requirements.

It was also recognized that the collection of Business patterns aligned with the Sell-Side Hub composite pattern.

Figure 5-1 also provides alignment against a typical methodology project phase. We can see from this perspective that we are entering the project phase where a specification view of the solution will be established.

5.2 Business pattern selection

In working through the conceptual phase of defining a solution in Chapter 3, “Requirements” on page 37, Figure 5-2 was created to provide a view of the
functional components required along with the major actors using the functional components identified.

This view can be thought of as an overview of the solution we are to develop. It shows all of the main areas of the solution.

This overview also allows the Business patterns that our solution is constructed from to be recognized.

![Image of Business and Integration patterns in our solution for BASE]

Figure 5-2  Business and Integration patterns in our solution for BASE

Once the Business and Integration patterns were identified, they were compared against Composite patterns for a possible match. The Sell-Side Hub composite pattern was seen as a solution for BASE’s business problem.

Figure 5-3 provides a view of the Sell-Side Hub composite pattern showing the individual Business and Integration patterns from which it is constructed. From the the IBM Patterns for e-business Web site at [http://www.ibm.com/developer/patterns/](http://www.ibm.com/developer/patterns/) we find a description that explains that in a Sell-Side Hub composite pattern, the seller owns the e-Marketplace and
uses it as a vehicle to sell goods and services to prospective buyers across the Web.

Figure 5-3  e-Marketplace Sell-Side Hub composite pattern

As shown in Figure 5-3, the Sell-Side Hub composite pattern includes the following Business and Integration patterns:

- **Access Integration** pattern, which helps provide a unified customer interface
- **Self-Service** business pattern, which allows users to browse through a catalog, create an order and place an order with the hub
- **Information Aggregation** business pattern, which is used to create the e-marketplace catalog from the multiple sources of suppliers’ product files, pricing files and advertising literature, etc.
- **Application Integration** patterns, which integrate the Business patterns that are a part of the Sell-Side Hub
In addition to these basic functions, there can be several variations on this pattern. These include:

- Adding a Collaboration business pattern that allows for and enables auctions, reverse auctions and other collaborative buying functions. In our case, there is a need for this Business pattern for the approval of orders and CSR support.

- Adding an Information Aggregation business pattern that will help integrate and present a unified catalog that can combine raw catalog data with expert advice, product comparisons and recommendations that can be pulled off public Internet sites. We use this Business pattern to allow our catalog to be constructed from data available from existing backend systems used within the BASE environment. There is also an expectation within BASE Pty. Ltd that there may be a future need to integrate with one of their suppliers product management systems.

- Adding an Extended Enterprise business pattern that will integrate the Sell-Side Hub with external service providers such as a financial institution to handle credit processing or a shipping company to handle the physical delivery of goods. We do not have a requirement for the Extended Enterprise business pattern as BASE does not integrate with external organizations for this purpose. Nor is this external integration seen as a future possibility.

### 5.3 Architectural artifacts

Chapter 3, “Requirements” on page 37, also provides us with:

- A refined set of business requirements that need to be satisfied by the solution required by BASE Pty. Ltd.

- Non-functional requirements that will assist in construction of the more physical elements of the solution.

- Principles, policies, standards and guidelines that will be followed in order to design a solution suitable for implementation in the BASE Pty. Ltd corporate environment.

- A view of the current IT environment with which the solution will integrate.

### 5.4 Constructing a logical view of the architecture

From a Patterns for e-business perspective, we are now at the phase in our project where we need to determine the Application patterns required to deliver a solution that satisfies the business need.
We will match our business and IT drivers against the business and IT characteristics of the Application patterns to arrive at selections that are relevant for BASE’s ordering system.

Although a great deal of the selection will be made from the business and IT characteristics, we also require a view of some of the more important business and non-functional requirements.

Once each of the Application patterns is selected, we will produce a Runtime pattern to satisfy the Application pattern. We will then combine the Runtime patterns to provide a single Runtime pattern. From this single Runtime pattern, we can specify nodes and network characteristics.

From the view of a typical methodology, this equates the logical view of the operational architecture.

Note: A user would not normally document their analysis of pattern usage to this extent. This chapter section, however, has been constructed to try and document the thought process that would take place to allow the pattern material to be used.

5.4.1 Select Application and Runtime patterns

When dealing with patterns, the focus is on reuse. In following this theme, much of the descriptive text and many of the images in this document section have been extracted from the IBM Patterns for e-business Web site at http://www.ibm.com/developer/patterns/.

It must be remembered that an Application pattern is a high-level view of the principal layout of the application. It is void of geographic and product detail.

As an example, we may choose an Application pattern that shows three quite separate tiers. This does not mean that there will necessarily be three nodes implemented to satisfy the pattern. It is possible, but unlikely, that the three logical tiers are implemented on a single node.

We will use Figure 5-2 on page 73 as a reference as we move through each of the individual Business patterns and discuss the associated Application patterns in order to make a selection. The relevant Business pattern will be highlighted in the section in which it is discussed.

The characteristics of the Application patterns will be matched against the requirements and goals of the solution we are to deliver.
A corresponding Runtime pattern will then be designed for each Application pattern.

**Online order capture application business drivers**
The following business drivers have been produced from both the requirement material delivered with the request for proposal and additional meetings with BASE business staff as required. These drivers enable us to match against the characteristics of the Application patterns. This obviously assists in selecting a pattern to match our need. These drivers are for the system as a whole and at times for the BASE organization.

The business drivers are as follows.

- Reduce costs by reducing the staff workload associated with servicing client orders.
- Increase customer satisfaction by providing a Web centered access channel allowing client organizations to have Self-Service capability.
- Decrease end-to-end response order time by providing Self-Service capability.
- Maintain customer satisfaction by maintaining a unified customer view across lines of business.
- Increase customer satisfaction by providing integration capability for customer procurement systems.
- CSRs will be able to provide online assistance to users.
- Delegation support will be provided to customers for some areas of functionality, allowing lesser users to perform some functionality, but not others, without approval.
- Maintain customer satisfaction by integrating with existing single sign-on infrastructure.
- Integrate with existing product, order, fulfilment and pricing systems.

**Online order capture application IT drivers**
As with the business drivers above, the following IT drivers have been produced from both the material delivered with the request for proposal, in this case the non-functional requirements and additional meetings with BASE business staff as required. They are also used to match against the characteristics of the Application patterns in order to arrive at a suitable selection.

The IT drivers:

- Must be able to scale to support additional products and users
- Are easily extended for other purposes
5.4.2 Self-Service business pattern for product configuration, product search and selection and order functionality

In this section, we choose an Application pattern to satisfy the Self-Service business patterns identified earlier in the project selected to satisfy the need for BASE's customers to have Self-Service access to product configuration, product search and selection and order.

![Self-Service application pattern for product configuration, product search and selection and order](image)
Find a match for the business drivers

Table 5-1 provides a mapping of our business drivers against the characteristics of the Application patterns as listed on the IBM Patterns for e-business Web site for the Self-Service business pattern.

A “✓” symbolizes that the Application pattern has the characteristic shown in the ‘Application pattern characteristics’ column. This information has been taken directly from the IBM Patterns for e-business Web site.

The shaded boxes indicate a match against our business driver.

There are also additional BASE drivers that are not explicitly referenced in the characteristics of the Application patterns on the IBM Patterns for e-business Web site. The descriptive text accompanying the Application pattern on the Web site does, however, allow for a match against these additional BASE drivers. If shading is shown without “✓”, the match was performed against the descriptive text.

Table 5-1 Business drivers for product configuration, product search and selection and order

<table>
<thead>
<tr>
<th>Business drivers for BASE Pty. Ltd</th>
<th>Application pattern characteristics</th>
<th>Stand-Alone</th>
<th>Directly Integrated</th>
<th>As-Is Host</th>
<th>Customized Presentation</th>
<th>Router</th>
<th>Decomposition</th>
<th>Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to market</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reduce costs by reducing the staff workload associated with servicing client orders</td>
<td>Improve organizational efficiency</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decrease end-to-end response order time by providing Self-Service capability</td>
<td>Reduce latency of business events</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Easy to adapt during mergers and acquisitions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
Find a match for the IT drivers
As we matched the business drivers, we now repeat the process using Table 5-2 for the IT drivers.

Table 5-2  IT drivers for product configuration, product search and selection and order
When first performing the match of business driver against characteristics of the Application patterns, it would appear that “Integrate across multiple delivery channels” and “Unified customer view across lines of business” provide a match. These drivers are not actually required to cater to the requirements as specified, as the functionality to provide for these drivers is not being implemented by the solution we build. This is already in place and implemented by existing systems.

The “easily extended for other purposes” may result in the need for additional access channels that are not provided for now. As such, a match has been made against “Integrate across multiple delivery channels” but not “Unified customer view across lines of business”.

**Application pattern selection discussion**

<table>
<thead>
<tr>
<th>IT Drivers</th>
<th>Application pattern characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Stand-Alone</td>
</tr>
<tr>
<td>Minimize total cost of ownership</td>
<td>✓</td>
</tr>
<tr>
<td>Leverage existing skills</td>
<td>✓</td>
</tr>
<tr>
<td>Integration with existing systems</td>
<td>Leverage legacy investment</td>
</tr>
<tr>
<td>Integration with existing systems</td>
<td>Backend application integration</td>
</tr>
<tr>
<td>Ease of maintenance</td>
<td>Minimize enterprise complexity</td>
</tr>
<tr>
<td>Must be able to scale to support additional products and users</td>
<td>Scalability</td>
</tr>
<tr>
<td>Easily extended for other purposes</td>
<td></td>
</tr>
</tbody>
</table>
The core functionality and/or data for the initiative being developed to satisfy this Self-Service business pattern is located in the back-end systems, which would indicate the Router application pattern may be a good fit. The Router application pattern also provides isolation of function from the delivery channel. This enables functionality to change at some time in the future with little impact on the delivery channel.

An area that discounts the router channel is the fact that it uses a synchronous connection to the back-end, which would not meet our needs. Reading further however, it appears that it does support asynchronous communications to the back-end as well.

So, let's look at the Decomposition and Agent application patterns. The Agent and Decomposition application patterns were rejected because they provide for additional functionality not required for our business problem.

As such, we will use the Router application pattern as shown in Figure 5-5 for our solution to satisfy Self-Service access to product configuration, product search and selection and order functionality.

![Figure 5-5 Self-Service::Router application pattern](image)

**Self-Service::Router runtime pattern**

Figure 5-6 shows the Runtime pattern as applicable to the BASE application. From the diagram, it can be seen that the integration server is implemented on the same node as the application server rather than in a separate node.
5.4.3 Find a match for the registration Self-Service business pattern

The second Self-Service business pattern identified is an online application that is already in place to allow customers and their representatives to register for applications offered via the Internet.
Find a match for the business drivers

As for the earlier Self-Service pattern, Table 5-3 provides a mapping of our business drivers against the characteristics of the Application patterns as listed on the IBM Patterns for e-business Web site for the Self-Service business pattern.

A ✓ symbolizes that the Application pattern has the characteristic shown in the Application pattern characteristics column. This information has been taken directly from the IBM Patterns for e-business Web site.

The shaded boxes indicate a match against our business driver.

There are also additional BASE drivers that are not explicitly referenced in the characteristics of the Application patterns on the IBM Patterns for e-business Web site. The descriptive text accompanying the Application pattern on the Web site does, however, allow for a match against these additional BASE drivers. If shading is shown without ✓ then the match was performed against the descriptive text.
Table 5-3  Business drivers for registration

<table>
<thead>
<tr>
<th>Business drivers for BASE Pty. Ltd</th>
<th>Application pattern characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Stand-Alone</td>
</tr>
<tr>
<td>Time to market</td>
<td>✓</td>
</tr>
<tr>
<td>Improve organizational efficiency</td>
<td>✓</td>
</tr>
<tr>
<td>Decrease end-to-end response order time by providing Self-Service capability</td>
<td>Reduce latency of business events</td>
</tr>
<tr>
<td>Easy to adapt during mergers and acquisitions</td>
<td>✓</td>
</tr>
<tr>
<td>Integrate across multiple delivery channels</td>
<td>✓</td>
</tr>
<tr>
<td>Unified customer view across lines of business</td>
<td>✓</td>
</tr>
<tr>
<td>Support effective cross selling</td>
<td>✓</td>
</tr>
<tr>
<td>Mass customization</td>
<td></td>
</tr>
</tbody>
</table>

Find a match for the IT drivers
As we matched the business drivers, we now repeat the process using Table 5-4 for the IT drivers.
It would appear from the guidance on the IBM Patterns for e-business Web site that the Router application pattern would be the best fit. However, many of the characteristics of the patterns apply to an enterprise rather than an application view.

In this case, the Stand-Alone Single Channel application pattern is actually a good fit. The registration application is constructed on top of LDAP compliant directory technology from IBM, SecureWay Directory. It is actually the use of the directory that provides an integration point for multiple access channels if required.

<table>
<thead>
<tr>
<th>IT Drivers</th>
<th>Application pattern characteristics</th>
<th>Stand-Alone</th>
<th>Directly Integrated</th>
<th>As-Is Host</th>
<th>Customized Presentation</th>
<th>Router</th>
<th>Decomposition</th>
<th>Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimize application complexity</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimize total cost of ownership</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Leverage existing skills</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Integration with existing systems</td>
<td>Leverage legacy investment</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Minimize enterprise complexity</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Ease of maintenance</td>
<td>Maintainability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Must be able to scale to support additional products and users</td>
<td>Scalability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Easily extended for other purposes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Users connect directly to the registration system when there is a need to register for access to an application. As such, the Stand-Alone Single Channel application pattern as shown in Figure 5-8 has been chosen for this area of the solution.

![Figure 5-8 Self-Service::Stand-Alone Single Channel application pattern](image)

The Runtime pattern to satisfy this application is shown in Figure 5-9. Again, we have separated the application server from the Web server since this is how the existing system is implemented. The registration application is physically located in the application zone.

![Figure 5-9 Self-Service::Stand-Alone Single Channel runtime pattern](image)

### 5.4.4 Collaboration business pattern for customer assistance

In this section, we choose Application patterns to satisfy our collaboration needs for the support of an online customer experiencing difficulty. In this case, the user can request support from a CSR.
Find a match for the business drivers

Table 5-5 provides a mapping of our business drivers against the characteristics of the Application patterns as listed on the IBM Patterns for e-business Web site for the Collaboration business pattern.

A ✓ means that the Application pattern has the characteristic shown in the Application pattern characteristics column. This information has been taken directly from the IBM Patterns for e-business Web site.

The shaded boxes indicate a match against our business driver.

There are also additional BASE drivers that are not explicitly referenced in the characteristics of the Application patterns on the IBM Patterns for e-business Web site. The descriptive text accompanying the Application pattern on the Web site does, however, allow for a match against these additional BASE drivers. If shading is shown without ✓ then the match was performed against the descriptive text.
Find a match for the IT drivers
As we matched the business drivers, we now repeat the process in Table 5-6 for the IT drivers.

<table>
<thead>
<tr>
<th>Business drivers for BASE Pty. Ltd</th>
<th>Application pattern characteristics</th>
<th>Point-to-point</th>
<th>Store and Retrieve</th>
<th>Directed Collaboration</th>
<th>Managed collaboration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time to market</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Reduce costs by reducing the staff workload associated with servicing client orders</td>
<td>Improve organizational efficiency</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Decrease end-to-end response order time by providing Self-Service capability</td>
<td>Reduce latency of business events</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Maintain customer satisfaction by providing online assistance to users when required</td>
<td>Require instantaneous collaboration</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Require referred collaboration</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Require workflow collaboration</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Many users</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
**Table 5-6  IT drivers for customer assistance**

<table>
<thead>
<tr>
<th>IT Drivers</th>
<th>Application Pattern</th>
<th>Point-to-point</th>
<th>Store and Retrieve</th>
<th>Directed Collaboration</th>
<th>Managed collaboration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leverage existing skills</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Network addressing independence</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Managed service</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Ease of maintenance</td>
<td>Maintainability</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Complex data types</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Significant network bandwidth</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

**Application pattern selection discussion**

In the case of a CSR providing support for an online customer experiencing difficulty, the user has no knowledge of the CSR user name or address. The interaction is controlled by a server since customers can only seek support from CSRs that are currently available to support their customers. With the agreement of the user during the collaborative session with the CSR, some of the customer’s information will be made available to the CSR.

The most suitable Application pattern for our CSR related collaboration requirements is the Directed Collaboration application pattern as shown in Figure 5-10.
The Runtime pattern shown in Figure 5-12 depicts the runtime environment to enable customer to CSR collaboration sessions. The shared data in this scenario is the conversation between the CSR and the customer representative. Order information is also shared to allow the CSR to provide assistance to the customer.
5.4.5 Collaboration business pattern for notification

This form of collaboration is required to allow for the need for an approver to approve an order that has exceeded a predetermined monetary value.
Chapter 5. Proposing a solution

Find a match for the business drivers

Table 5-7 provides a mapping of our business drivers against the characteristics of the Application patterns as listed on the IBM Patterns for e-business Web site for the Collaboration business pattern.

A ✓ means that the Application pattern has the characteristic shown in the Application pattern characteristics column. This information has been taken directly from the IBM Patterns for e-business Web site.

The shaded boxes indicate a match against our business driver.

There are also additional BASE drivers that are not explicitly referenced in the characteristics of the Application patterns on the IBM Patterns for e-business Web site. The descriptive text accompanying the Application pattern on the Web site does, however, allow for a match against these additional BASE drivers. If shading is shown without ✓ then the match was performed against the descriptive text.
Table 5-7  Business drivers for notification

<table>
<thead>
<tr>
<th>Business drivers for BASE Pty. Ltd</th>
<th>Application pattern characteristics</th>
<th>Point-to-point</th>
<th>Store and Retrieve</th>
<th>Directed Collaboration</th>
<th>Managed collaboration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to market</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Reduce costs by reducing the staff workload associated with servicing client orders</td>
<td>Improve organizational efficiency</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Decrease end-to-end response order time by providing Self-Service capability</td>
<td>Reduce latency of business events</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Easy to adapt during mergers and acquisitions</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Require instantaneous collaboration</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Require referred collaboration</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Delegation support will be provided to customers for some areas of functionality allowing lessor users to perform some functionality, but not others without approval</td>
<td>Require workflow collaboration</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Many users</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Find a match for the IT drivers
As we matched the business drivers, we now repeat the process in Table 5-8 for the IT drivers.

Table 5-8  IT drivers for notification

<table>
<thead>
<tr>
<th>IT Drivers</th>
<th>Application Pattern</th>
<th>Point-to-point</th>
<th>Store and Retrieve</th>
<th>Directed Collaboration</th>
<th>Managed collaboration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leverage existing skills</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Network addressing independence</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Managed service</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Ease of maintenance</td>
<td>Maintainability</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Complex data types</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Significant network bandwidth</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Application pattern selection discussion
This form of collaboration is required to allow for the need for an approver to approve an order that has exceeded a predetermined monetary value. The order placement and perhaps the monetary value of the order itself is the trigger. This has a workflow element to it as the approver is not required until the order is placed and the order will not proceed until approved. In this scenario, there is no direct collaboration between users; the interaction between users is indirect, the direct interaction being between the server and user environments.

The Managed Collaboration application pattern provides for BASE’s notification collaboration requirements, as shown in Figure 5-14.
Figure 5-14  Collaboration::Managed Collaboration application pattern

Figure 5-15 shows the Runtime pattern required to deal with the approval style of collaboration required by the solution we develop for BASE Pty. Ltd.

5.4.6 Information Aggregation pattern for catalog population

In this section, we choose an Application pattern to satisfy the Information Aggregation pattern required to populate the catalog from the disparate sources of information already within the BASE organization’s environment. These
disparate sources are the product management system, price and inventory systems.

![Figure 5-16 Information Aggregation pattern for catalog population](image)

**Find a match for the business drivers**

Table 5-9 provides a mapping of our business drivers against the characteristics of the Application patterns as listed on the IBM Patterns for e-business Web site for the Aggregation business pattern.

A ✓ means that the Application pattern has the characteristic shown in the Application pattern characteristics column. This information has been taken directly from the IBM Patterns for e-business Web site.

The shaded boxes indicate a match against our business driver.

There are also additional BASE drivers that are not explicitly referenced in the characteristics of the Application patterns on the IBM Patterns for e-business Web site. The descriptive text accompanying the Application pattern on the Web site does, however, allow for a match against these additional BASE drivers. If
shading is shown without ✓ then the match was performed against the descriptive text.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduce costs by reducing the staff workload associated with servicing client orders</td>
<td>Improve organizational efficiency</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>Decrease end-to-end response order time by providing Self-Service capability</td>
<td>Reduce latency of business events</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td></td>
<td>Distill meaningful information from vast amounts of structured data</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
</tr>
<tr>
<td>Integrate with existing product, order, fulfilment and pricing systems</td>
<td>Extensive reconciliation, transformation and restructuring of structured data</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
</tr>
<tr>
<td></td>
<td>Provide easier access to vast amount of unstructured data through indexing and categorization</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
</tr>
</tbody>
</table>
Find a match for the IT drivers
As we matched the business drivers, we now repeat the process in Table 5-10 for the IT drivers.

<table>
<thead>
<tr>
<th>Business drivers for BASE Pty. Ltd</th>
<th>Application pattern characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Population - single step</td>
</tr>
<tr>
<td></td>
<td>Population - multi step</td>
</tr>
<tr>
<td></td>
<td>Population - crawling and discovery</td>
</tr>
<tr>
<td></td>
<td>Population - summarization</td>
</tr>
<tr>
<td></td>
<td>Information access - read only</td>
</tr>
</tbody>
</table>

Enhance access to unstructured data through summarization

Provide access to distilled information and drill-through capability

✓
Application pattern selection discussion

We have relatively complex rules that need to be applied to transform our data from the individual sources we have in place to the single target satisfying our catalog view that is required to support BASE’s requirements.

Each of the inventory, product management and pricing systems needs to provide information to the ordering system at a rate that is appropriate for the typical rate of change experienced by that source system. We can expect the “in stock” count from the inventory system to update continuously throughout the day as orders are placed and fulfilled. The business is competitive, so price information could change throughout the day. Product descriptions, etc. would seldom change.

The information in the source systems is not in the same XML format expected by WebSphere Commerce Business Edition. Therefore, relevant information provided by source systems must be transformed before it can be loaded into WebSphere Commerce Business Edition.
The following words, extensive reconciliation, transformation and restructuring of structured data, are used to describe a key characteristic of the Multi Step application pattern. It is a perfect fit.

As such, the Population-Multi Step application pattern as shown in Figure 5-17 is a suitable solution for our business problem.

![Figure 5-17 Information Aggregation::Population-Multi Step application pattern](image)

There are three sources of information required to satisfy this aggregation need to build the catalog and one for the document management system. As such, the Runtime pattern detailed in Figure 5-16 shows the Information Aggregation::Population-Multi Step runtime pattern for the catalog with three sources.
This is a critical pattern within the solution since there is a need to reuse the BASE legacy systems. Unlike other areas of the solution, we know this area is going to be difficult. It is not going to fall out of the box. As such, a little more time has been spent on this pattern to ensure we do indeed have a feasible solution to form part of the proposal.

It must be noted that this is unlikely to be the final solution. It is just “a solution” so that we have something to discuss with BASE when the question is asked.

A once off, source data population is performed by running a query on each source system and writing the results to a file. The file is transferred to the transformation environment by tape or FTP and loaded via SQL script to the transformation database. Stored procedures are invoked and transform the data so it is in the format expected by WebSphere Commerce. The Loader is used to perform the first population of catalog information from this transformation database to the WebSphere Commerce catalog.
The above population could be repeated if required to recover from a disaster situation or on a regular basis as a refresh to ensure all data is in synch with the back-end systems.

In normal operation, source data will be extracted from the source systems by way of batch runs since the requirements for the online order capture solution do not warrant real time updates. A batch process will run on each of the inventory, price and product systems at regular intervals to determine what has changed since the last run. The interval is independently configurable for each system. The changed data is transferred via WebSphere MQ to the database server.

The data is read from the queue and stored in the transformation database. A stored trigger is invoked on storage and the data transformed to the format expected by WebSphere Commerce. Once data is transformed, the messaging system is used to update WebSphere Commerce catalog with the created, deleted or modified data.

5.4.7 Information aggregation for technical documentation repository population

We now discuss the second Aggregation pattern within the solution, the aggregation of data to the technical documentation repository. Remember, this is a system that is already in place in the operational environment. However, as the system is seen as part of the overall solution, it is being presented here.
Find a match for the business drivers

Table 5-11 provides a mapping of our business drivers against the characteristics of the Application patterns as listed on the IBM Patterns for e-business Web site for the Information Aggregation business pattern.

A ✓ means that the Application pattern has the characteristic shown in the Application pattern characteristics column. This information has been taken directly from the IBM Patterns for e-business Web site.

The shaded boxes indicate a match against our business driver.

There are also additional BASE drivers that are not explicitly referenced in the characteristics of the Application patterns on the IBM Patterns for e-business Web site. The descriptive text accompanying the Application pattern on the Web site does, however, allow for a match against these additional BASE drivers. If
shading is shown without ✓ then the match was performed against the descriptive text.

Table 5-11  Business drivers for technical documentation repository population

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduce costs by reducing the staff workload associated with servicing client orders</td>
<td>Improve organizational efficiency</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Decrease end-to-end response order time by providing Self-Service capability</td>
<td>Reduce latency of business events</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Distill meaningful information from vast amounts of structured data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrate with existing product, order, fulfilment and pricing systems</td>
<td>Extensive reconciliation, transformation and restructuring of structured data</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provide easier access to vast amount of unstructured data through indexing and categorization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Find a match for the IT drivers**

As we matched the business drivers, we now repeat the process in Table 5-12 for the IT drivers.

<table>
<thead>
<tr>
<th>Business drivers for BASE Pty. Ltd</th>
<th>Application pattern characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Population - single step</td>
</tr>
<tr>
<td></td>
<td>Population - multi step</td>
</tr>
<tr>
<td></td>
<td>Population - crawling and discovery</td>
</tr>
<tr>
<td></td>
<td>Population - summarization</td>
</tr>
<tr>
<td></td>
<td>Information access - read only</td>
</tr>
<tr>
<td>Enhance access to unstructured data through summarization</td>
<td>✔</td>
</tr>
<tr>
<td>Provide access to distilled information and drill-through capability</td>
<td>✔</td>
</tr>
</tbody>
</table>
Table 5-12  IT drivers for technical documentation repository population

<table>
<thead>
<tr>
<th>IT Drivers</th>
<th>Application Pattern</th>
<th>Population - single step</th>
<th>Population - multi step</th>
<th>Population - crawling and discovery</th>
<th>Population - summarization</th>
<th>Information access - read only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrate with existing product, order, fulfilment and pricing systems</td>
<td>Minimize total cost of ownership</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Integrate with existing product, order, fulfilment and pricing systems</td>
<td>Promote consistency of operational data</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Ease of maintenance</td>
<td>Maintainability</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Application pattern selection discussion

Once again, we have relatively complex rules that need to be applied to transform our data from the individual source we have in place to the single target since the data is not in the same format expected by the technical documentation repository. Therefore, relevant information must be transformed before it can be loaded into the technical documentation repository.

The product descriptions and associated technical documentation seldom change.

The following words extensive reconciliation, transformation and restructuring of structured data are used to describe a key characteristic of the Multi Step application pattern. Although reconciliation is not terribly relevant in this scenario, the other characteristics of the pattern are a good fit.

As such, the Population-Multi Step application pattern shown earlier in Figure 5-17 is a suitable solution for this business problem as well.
Figure 5-20 is obviously similar to the Runtime pattern shown in Figure 5-17 on page 101, this time with a single source used to populate the document management system. This existing system’s Runtime pattern was adopted for the earlier catalog aggregation solution. The source system for the population of data to the technical documentation repository is also the product management system.

5.4.8 Access Integration pattern for single sign-on

In this section, we choose an Application pattern for the Access Integration pattern required to satisfy single sign-on.

The shaded are within Figure 5-21 details the area of responsibility for the Access Integration, Single Sign-On pattern.
Application pattern selection discussion

From the material delivered in Chapter 3, “Requirements” on page 37, we know that we have a requirement to integrate with the existing corporate single sign-on infrastructure.

There are two Single Sign-On application patterns discussed in the Application pattern section of the IBM Patterns for e-business Web site. One offering extends security into the back-end applications, the other not. As there is no requirement to provide the extended level of security for the solution being constructed to satisfy BASE Pty. Ltd’s business requirements, the simple Application pattern shown in Figure 5-22 has been selected.
The existing infrastructure is based on the International Business Machines Corporation’s Tivoli suite of products, in particular, the WebSeal reverse proxy security server. Figure 5-24 provides a view of the interconnection of components to allow single sign-on in the BASE environment. This pattern is titled Access Integration::Web single sign-on::heterogeneous application server as it is designed to allow single sign-on across differing application server technologies.

Because the ordering system requirements dictate high levels of availability to the Web server so that intelligent error messages are received in the event of the application being unavailable, we require more than one Web server and authentication server.
What makes this particularly difficult is the fact that we have two user stores that will contain information for our users. One holds information for the customers, the other for employees. Again, as with the population of the catalog, this is an area of the solution that is quite complex. As such, we discuss it in more detail than the other patterns. We do not need to choose a final solution, but we do need to know that there is a feasible one.

Please refer to Chapter 11, “User identity integration” on page 303 to obtain more detail on the single sign-on solution.

5.4.9 Application Integration patterns

In this section, we choose the Application patterns to satisfy each of the Application Integration patterns identified during the conceptual phase of the project. These application integration points are depicted in Figure 5-24 by the numbered arrows.
Find a match for the business drivers

The Application Integration patterns are the patterns that operate to link all of the other patterns together. The numbered connectors in Figure 5-24 are the integration points within our application.

Table 5-13 shows the characteristics required from each of the specific application integration points we must design.
Table 5-13  Application integration points

<table>
<thead>
<tr>
<th>#</th>
<th>Source Component</th>
<th>Target Component</th>
<th>Integration Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Product management system</td>
<td>Product catalog</td>
<td>- This information is required to form the catalog within the online order system.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- This is a process integration rather than data integration point.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- The information is propagated on a periodical basis as changes are rare and are coordinated.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Non-transactional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Background</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Low frequency</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Initiated by batch process</td>
</tr>
<tr>
<td>2</td>
<td>Product management system</td>
<td>Technical documentati on repository</td>
<td>- This interface is already in place in the production environment to provide the product related information that is required by the online document server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- This is a process integration rather than data integration point.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- The information is propagated on a periodical basis as changes are rare and are coordinated.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Non-transactional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Background</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Low frequency</td>
</tr>
<tr>
<td>3</td>
<td>Inventory</td>
<td>Product catalog</td>
<td>- Information is required to form the availability component of the catalog for the online order system.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- This is a process integration rather than data integration point.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- The information is propagated on a periodical basis. This is acceptable from a business perspective as most items are high volume items. A large amount of stock is available at all times. The inventory updates are initiated by the order back-end system, not the online system to be developed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Non-transactional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Background</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Low frequency</td>
</tr>
</tbody>
</table>
## Pricing Product Catalog
- Information is required to form the pricing component of the catalog.
- This is a process integration rather than a data integration point.
- The information is propagated on a periodic basis as changes are rare and are coordinated.
- Non-transactional
- Background
- Low frequency

## Order Fulfillment
- This interface is to allow the online ordering system to place an order with the back-end fulfillment system.
- This is a process integration rather than a data integration point.
- Non-transactional
- Background
- Require guaranteed delivery for place order as it is imperative that an order not be lost.
- Place order functionality is already available via an existing ordering application working into the mainframe back-end. There is an expectation that this will be reused.
- The existing interfaces are available on an WebSphere MQ Integrator bus.

## Fulfillment Order
- This interface allows the online ordering system to receive status updates from the back-end fulfillment system on orders that have been placed with the back-end.
- This is a process integration rather than a data integration point.
- Non-transactional
- Background
- Order status update notification is already available to an existing ordering application working into the mainframe back-end. There is an expectation that this existing interface will be reused.
- The existing interfaces are available on an WebSphere MQ Integrator bus.

<table>
<thead>
<tr>
<th>#</th>
<th>Source Component</th>
<th>Target Component</th>
<th>Integration Characteristics</th>
</tr>
</thead>
</table>
| 4 | Pricing          | Product catalog  | - Information is required to form the pricing component of the catalog.  
- This is a process integration rather than data integration point.  
- The information is propagated on a periodical basis as changes are rare and are coordinated.  
- Non-transactional  
- Background  
- Low frequency |
| 5a | Order            | Fulfillment      | - This interface is to allow the online ordering system to place an order with the back-end fulfillment system.  
- This is a process integration rather than data integration point  
- Non-transactional  
- Background  
- Require guaranteed delivery for place order as it is imperative that an order not be lost.  
- Place order functionality is already available via an existing ordering application working into the mainframe back-end. There is an expectation that this will be reused.  
- The existing interfaces are available on an WebSphere MQ Integrator bus. |
| 5b | Fulfillment      | Order            | - This interface allows the online ordering system to receive status updates from the back-end fulfillment system on orders that have been placed with the back-end.  
- This is a process integration rather than data integration point  
- Non-transactional  
- Background  
- Order status update notification is already available to an existing ordering application working into the mainframe back-end. There is an expectation that this existing interface will be reused.  
- The existing interfaces are available on an WebSphere MQ Integrator bus. |
<table>
<thead>
<tr>
<th>#</th>
<th>Source Component</th>
<th>Target Component</th>
<th>Integration Characteristics</th>
</tr>
</thead>
</table>
| 6  | Product search and selection | Product catalog      | - This integration point is to allow the product and search component to have access to information available from the catalog.  
   |                                        |                       |   - Information  
   |                                        |                       |   - Non-transactional  
   |                                        |                       |   - Real-time  
   |                                        |                       |   - Provided by WebSphere Commerce |
| 7  | Order                  | Product search and selection | - This integration point is to allow the order component to have access to information available from the catalog.  
  |                                          |                       |   - Information  
  |                                          |                       |   - Non-transactional  
  |                                          |                       |   - Real-time  
  |                                          |                       |   - Provided by WebSphere Commerce |
| 8  | Product catalog        | Product configurator  | - Allows the product catalog to invoke the configurator to allow the user to configure an item to purchase.  
  |                                          |                       |   - Functional  
  |                                          |                       |   - Non-transactional  
  |                                          |                       |   - Real time  
  |                                          |                       |   - Pre built from WebSphere Commerce |
| 9  | Product search and selection | Product configurator  | - Allows the product search and selection component to invoke the configurator to allow the user to configure an item to purchase.  
  |                                          |                       |   - Process  
  |                                          |                       |   - Non-transactional  
  |                                          |                       |   - Real time  
  |                                          |                       |   - Pre built from WebSphere Commerce |
| 10 | Order                  | User notification     | - This integration point is to allow the order component to place notify users that a user under their responsibility has placed an order that requires approval.  
  |                                          |                       |   - Non-transactional  
  |                                          |                       |   - Background  
  |                                          |                       |   - Process  
  |                                          |                       |   - Low frequency  

<table>
<thead>
<tr>
<th>#</th>
<th>Source Component</th>
<th>Target Component</th>
<th>Integration Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Order</td>
<td>Customer registration and preferences</td>
<td>- This integration point is provided to allow the order component to access information from</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>the user stores to determine such things as the delivery address for the order.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Non-transactional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Real-time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Process</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Low frequency</td>
</tr>
<tr>
<td>12</td>
<td>Customer registration</td>
<td>User notification</td>
<td>- This point is to allow the nominated representative for a company to be nominated</td>
</tr>
<tr>
<td></td>
<td>and preferences</td>
<td></td>
<td>that he or she must approve a registration request.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Is also used to notify staff in the case that a company is not already registered as a</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>customer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Non-transactional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Process</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Background</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Low frequency</td>
</tr>
</tbody>
</table>
Integration pattern selection discussion for catalog propagation

As shown in Figure 5-25, integration points 1, 3 and 4 are Application Integration patterns within an Aggregation pattern. From the characteristics detailed and the selection made in the aggregation section of this chapter, any style of batch sourcing of information with a point to point delivery would be acceptable. We selected the Direct Connection application pattern shown in Figure 5-26.
WebSphere MQ is used within the BASE environment, offering flexibility not provided by other point to point styles of interface. As such, the Runtime patterns are designed around the use of WebSphere MQ.

Figure 5-27 provide a view of the Direct Connection runtime pattern to satisfy the Aggregation business pattern for integration points 1, 3 and 4.

If the complexity of data and functional integration becomes too complex within the BASE environment, the point to point style of interface chosen here could prove to be a poor selection. In this circumstance, a more sophisticated pattern implemented with a more sophisticated product such as WebSphere MQ Integrator would be a better choice. However, given the existing use of WebSphere MQ and the still relatively simple nature of integration needs within the BASE environment, this selection is seen as a suitable solution.
Integration pattern selection for technical documentation repository

Figure 5-28 shows the integration points required for the technical documentation repository aggregation.

Integration point 2 is identical to those already discussed to satisfy the catalog aggregation needs. Once again, we chose the Direct Connection application pattern shown in Figure 5-26. Therefore, the runtime shown in Figure 5-29 is almost identical, except that the target environment is in the application zone, since this is where the data store for the technical documentation repository is located.
Integration pattern selection: product and order Self-Service pattern

Figure 5-30 shows the integration points required to deliver Self-Service capability for order and product requirements.
Integration points 6 and 7 are similar in nature. They are both integration points within the Self-Service business pattern operating between new components of the solution.

These integration points do not require a great deal of flexibility as there are no future requirements for change in this area of the solution. A Direct Connection application pattern was also chosen for this integration need. In this case, we know we are using the WebSphere Commerce Business Edition product and the runtime for these integration points is provided by WebSphere Commerce Business Edition. As such, there is no Runtime pattern to show here.

Integration points 5a and 5b again are similar in nature in that there is no need for a real time style of connection.

Transactional capability is not required for any of these integration points. Guaranteed delivery is, however, required for 5a.

Unlike previous direct connections where the backend system initiated the conversation, communication is initiated by the new application server.

---

**Figure 5-31  Application Integration::Direct Connection application pattern**

Once again, a Direct Connection pattern is selected that will be implemented with WebSphere MQ. Figure 5-32 shows the runtime view of this pattern.
Integration points 8 and 9 appear to be direct connections because of their simple requirements.

Again, we use the fact that we are using WebSphere Commerce Business Edition into consideration. These points are similar to the previous ones in that
the runtime is provided by a standard mechanism used by WebSphere Commerce Business Edition to integrate with an external configurator. This takes advantage of URL rewriting to pass information backwards and forwards between the configurator and WebSphere Commerce Business Edition, allowing both applications to interoperate to satisfy a single business process.

Therefore, the runtime solution for these integration points does not add to the solution. There is no additional Runtime pattern for these integration points.

Integration patterns required for notification
Integration point 10 is used to allow the order component to place a notification with users.

This is also a direct connect style of integration. Again, this is provided by WebSphere Commerce Business Edition. It is internal to the product and does not add anything to the runtime.
Integration point 12 is used to allow the order component to place a notification with users when users within the same organization require access to the order capture application. Again, a Direct Connection application pattern is selected.

Application integration points 10 and 12 are satisfied by the runtime shown in Figure 5-35, where the application server mails notification of some event to the user. A mail gateway allows the mail to pass from the BASE environment out to the Internet.

**Figure 5-35  Direct connection for notification runtime**

**Integration patterns required for user order time preferences**
Application integration point 11 provides the order component with the ability to check against the user details captured at registration time in order to get information on order preferences, such as delivery location.
Figure 5-36  Integration pattern required for user preferences

This is again a direct connection provided by the WebSphere Commerce Business Edition product and does not add to the runtime.

**Combined application pattern**

Figure 5-37 provides a view of all the Application patterns selected to satisfy the solution.
The runtime environment in the BASE organization will become more complex as additional applications are developed that take advantage of the functionality and information available from the legacy (backend) systems. If this point to point style of interconnection between systems were continued, the environment would become too complex to maintain. If the environment did continue to grow, an alternative such as WebSphere MQ Integrator would need to be considered for systems integration. The use of WebSphere MQ as a point to point means of communication is suitable for an environment the size of BASE Pty. Ltd.

We now look at where the Composite pattern that has been identified earlier fits into the combined application pattern view. From Figure 5-38, we can see that many of the components that are included are from the existing environment.
If we were in the position of needing to find a product, that is, if we did not know in advance that the runtime solution was based on WebSphere Commerce Business Edition, this method of identification of the Composite pattern would allow us to shop for a package that provided good coverage against our needs.

Using the Composite pattern as a guide, we can now match where WebSphere Commerce Business Edition provides a solution for the business problem. WebSphere Commerce Business Edition will obviously not be responsible for any existing component within the BASE environment.
From Figure 5-39, it is evident that the WebSphere Commerce package can provide for nearly all of the new components identified when the Business patterns were selected in Chapter 4, “Understanding the business problem” on page 55.

This does not mean that the solution will simply fall out of the box. There will be extensive configuration, programmatic extension of WebSphere Commerce capabilities and the construction and configuration of interfaces to existing systems. The catalog population is one such area that is significant in development effort.

This information, along with the complete Runtime pattern shown in Figure 5-40, is used in Chapter 6, “Operational overview” on page 155 to perform product mapping against the logical view of the operational architecture.

**Complete Runtime pattern**

Figure 5-40 shows the combined Runtime patterns constructed from the individual application Runtime patterns. This is a conceptual view of the
operational architecture. Chapter 6, “Operational overview” on page 155 adds product mapping to this combined runtime view.

The operational architecture will be further defined in Chapter 6, “Operational overview” on page 155. This view is suitable for the proposal we are producing.

5.5 Constructing the proposal

This section illustrates how it is possible to leverage IBM’s Patterns for e-business when constructing a proposal to provide a solution to satisfy a business need.
5.5.1 Setting the scene

In constructing the example deliverables for our proposal, we start at a point that is focused on the delivery of a solution. The following is already in place for BASE Pty. Ltd.

1. IBM has sold the benefits of J2EE to BASE and convinced BASE to adopt WebSphere Application Server as its standard runtime platform.

2. IBM has constructed a high-level proposal detailing the competitive advantage to be gained by BASE in providing an online order capture capability for their business. This proposal sold the use of WebSphere Commerce Business Edition as the J2EE compliant solution for this space that will operate on BASE’s standard WebSphere Application Server runtime platform.

3. BASE has an existing Web presence.

4. BASE has existing customer and staff identity management strategies, processes and systems in place.

5. The BASE organization has existing online security processes and enabling infrastructure that provides authentication and coarse-grained authorization capability to online applications.

6. BASE has invited us to respond to a request for proposal for the design and development of its online store as per the requirements detailed in Chapter 3, “Requirements” on page 37.

Hence, we develop a project proposal for the design and construction of an IT solution for BASE Pty. Ltd to satisfy their business need.

In putting together a project proposal to meet BASE’s objectives for a new initiative, the proposal needs to establish BASE’s trust in our capability to satisfy their objectives. In doing so, we do not only provide a solution. Our project proposal will provide BASE with:

- Information that shows we have an appreciation of their business, including information specific to their business and the market in which they operate.
- Details of the BASE environment with which our solution will integrate.
- An appreciation of the risk associated with the design, build, implementation and operation of the solution.
- Cost information that will enable BASE to make a business decision on preceding with the solution,
Details of what we will deliver to BASE as part of the solution.

- A high-level design of the solution based on BASE’s standards, guidelines and policy for the implementation and operation of IT systems within its environment.

We will not deliver:

- A complete architectural design that is suitable for input into the build design phase. The architectural design is part of what the proposal is being produced to address.
- Cost information for the components that are not within our realm of responsibility.
- Information for costs associated with implementation or ongoing operation of the solution once it is delivered.

The rest of this document section provides example deliverables based on our business requirements which would form sections of a proposal.

## 5.5.2 High-level summary

The first step in a proposal is to provide a high-level summary. The summary has a number of key features which we detail in this section.

### Set the scene with the executive sponsors

This will establish what the business system means from a business perspective and will target business people and executives as the audience.

This section would be constructed around BASE’s business, BASE’s competitors, BASE’s business problems and BASE Pty. Ltd’s customer base. As an example:

- Many of BASE Pty. Ltd’s competitors are offering a Self-Service capability, allowing their customers to place orders with BASE via the Web. BASE’s customers expect the same level of service from the BASE organization.
- Many BASE customers have procurement systems that are capable of interfacing directly with the solution detailed in this proposal. There is an expectation that enabling the Business-to-Business, Self-Service style of operation offered by this solution will realize a cost saving not only to the BASE organization, but to BASE’s customers as well. This is likely to result in increased levels of customer satisfaction and hence loyalty and revenue.
Features and benefits
Features and benefits of the solution must be stated. It is not enough to simply satisfy requirements.

The following are examples of features and benefits which could be specified for the solution outlined in the proposal.

- A package-based solution is likely to result in reduced future costs for enhancement and maintenance.
- The change is likely to be implemented more quickly due to the standardized approach the platform encourages.
- Self-Service is likely to increase customer satisfaction, which may result in an increase in revenue.
- It is possible to simplify the implementation environment by utilizing the platform as a single solution for all order capture activities, including those performed by BASE staff directly against the legacy systems.
- There is integration with the corporate single sign-on infrastructure and legacy platforms to allow BASE to leverage from previous investments.
- Details can be given as to how the total solution is simplified from the development, deployment, operations and maintenance perspectives by adopting a package-based solution to solve the business problem.
- Details can be provided as to the standards compliance and the benefits derived from that compliance.

5.5.3 Overview
The next step in the proposal is a general overview that is suitable for the full audience. This includes a number of parts which we detail in this section.

Key or innovative aspects of the solution
Identify the key or novel areas of the solution and discuss them in turn. The Business patterns have provided the grouping on which to base much of this discussion whilst some of the discussion will also be based on the enabling technologies chosen which were identified when placing the product against the Runtime patterns. If using IBM products, the patterns Web site provides a head start on product selection for many of the Runtime patterns.

We will use the Information Aggregation pattern chosen for catalog population as an example of where Business patterns provided an area of focus for this section, and follow this with a statement about some of the enabling technologies.
Catalog population - Legacy system reuse
BASE has invested heavily in past years to ensure that the organization has the IT foundation necessary to operate in an international market. This solution proposes to leverage from that previous investment by using the product management system, along with pricing, inventory and fulfillment systems currently used within the BASE environment to build the online catalog offered to users of the proposed system.

This reuse enables orders to be viewed and managed from either the existing systems or the new order capture system. The source of the order is transparent to both staff and customers.

This also allows the proposed system to leverage from existing fulfilment processes and the enabling implementations.

Loose coupling of legacy systems
The solution uses a loose form of coupling to the legacy systems in order to source data for updating the catalog. This protect both the legacy systems from the order capture system and the order capture system from the legacy systems. This means that if one side of the coupling experiences operational difficulties, the experience is not noticeable to the other side of the coupling.

Enabling technologies
The solution includes the following enabling technologies.

1. WebSphere Application Server, Advanced Edition
   WebSphere Application Server, Advanced Edition is an industry proven application server platform that is a foundation component of the solution. It conforms to BASE’s IT standards and offers capabilities that ensure it will provide that foundation well into the future if the solution is required to scale to meet increased customer demand.

2. WebSphere Commerce Business Edition
   Much of the solution is provided out of the box by the WebSphere Commerce Business Edition product. This results in a solution that is easier to develop, implement, maintain and operate.

3. WebSphere MQ
   The WebSphere MQ product provides an abstraction layer that hides all details of one side of an interface from the other. This abstraction reduces development costs as each side can build to the interface without having to take the other side’s processes or technologies into account.

   This feature not only reduces development costs, but would indeed allow for the technology, processes or even the system itself on one side of the interface to change with no impact to the other side.

Chapter 5. Proposing a solution
133
5.5.4 Solution

The proposed solution has been designed as an e-Marketplace to capture orders for BASE goods from BASE’s customers across the Web.

Customers or their representatives can self-register to gain access to the online order functionality provided by the online order capture solution. Once registration has been finalized, users can place an order with BASE in two ways: one, as a user of the online order capture system from any location with Internet access, and two, by placing an order request from the customer’s compatible procurement system.

Not only does the solution provide for the requirements detailed within the BASE request for proposal, but it provides a solid foundation for more demanding requirements BASE may have in the future. As an example, the solution can be linked to systems BASE’s sellers use to automate much of BASE’s purchasing. Additional access channels can be added to allow customers to have access from pervasive devices such as WAP-enabled cellular telephones. The system can be expanded to form a single order entry system used within the BASE organization.

Figure 5-42 provides an overview of the solution components proposed for online order capture.
The solution is comprised of the following major high-level components:

- Product and catalog
- Order management
- Customer care
- Customer access and control
- Systems integration.

**Note:** In a completed proposal, a written description should be provided for these components. We do not include this in our redbook.

Figure 5-42 provides a more detailed view of the major solution components in the BASE environment.
5.5.5 Risks

The documentation of risks allows the customer to appreciate any such risks that may hinder the delivery of deliverables to the BASE organization. These risks are typically risks that BASE and we as a service provider share and need to work on together throughout the project to ensure that any adverse impact on cost, scope or timing is not imposed. The following is a typical risk that may be applicable for this project.

Legacy systems integration

BASE has many changes planned for the legacy fulfillment system which are scheduled during the proposed build stage for this initiative. If the schedules for the two initiatives are not aligned, it is possible either or both of the initiatives may suffer delay and cause a possible increase in costs.
5.5.6 Dependencies

In this section of the proposal, we list any of the dependencies we have back on the BASE organization. The following are examples of such dependencies for this project.

Availability of BASE expertise
Subject matter experts will be made available on an as-requested basis to provide information necessary to complete the design for a solution to satisfy BASE’s requirements.

A business owner of the online order capture system within the BASE organization will be available to resolve conflicts in the event of differing opinions between subject matter experts.

Formal acceptance of deliverables
BASE Pty. Ltd must provide formal acceptance by way of an executive sign-off of deliverables that are major dependencies for the next set of deliverables or project phase.

5.5.7 Requirements

The business requirements should be stated back to the customer in the proposal. Some readers, especially those at an executive level, are unlikely to read back through the request for proposal document that the proposal is constructed to satisfy.

Table 5-14 shows a list of requirements that have been extracted from the request for proposal document introduced in Chapter 3, “Requirements” on page 37. The requirements are numbered and are categorized and ordered as delivered by BASE Pty. Ltd for easy reference. Table 5-14 provides an indication of the solution components that will address each functional requirement.
<table>
<thead>
<tr>
<th>Requirement number</th>
<th>Requirement</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Catalog subsystem</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Users that have not logged in (so-called <em>anonymous users</em>) will be able to browse the catalog without being able to see the prices. This is done to attract potential customers. This way, they can easily study the range of products BASE has to offer. If they are interested, they can get into a business contact with BASE Pty. Ltd</td>
<td>Product and catalog, Customer access and control</td>
</tr>
<tr>
<td>2</td>
<td>Catalog search capability will be provided to the users to allow them to find products without having to browse the catalog</td>
<td>Product and catalog</td>
</tr>
<tr>
<td>3</td>
<td>Authenticated users will have the ability to browse a catalog. Catalog for perusal can be selected based on: - A contract, - A combination of contracts, or - The catalog in full. Customers can negotiate with BASE for certain conditions for the purchase of certain products. The outcome of these negotiations are captured in a contract between BASE and their customer</td>
<td>Product and catalog</td>
</tr>
<tr>
<td>4</td>
<td>The provided solution will interface with a custom-built external product configurator application for the configuration of complex analytical equipment that is configured on a per use basis</td>
<td>Systems integration</td>
</tr>
<tr>
<td>5</td>
<td>BASE is planning to connect their catalog system to one of their suppliers in the future</td>
<td>Product and catalog, Systems integration</td>
</tr>
<tr>
<td><strong>Member subsystem</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>New users will be able to <em>self register</em> (i.e., apply for a user ID) in order to gain access to the ordering system if the organization they belong to is an existing customer with BASE Pty. Ltd</td>
<td>Customer access and control</td>
</tr>
<tr>
<td>Requirement number</td>
<td>Requirement</td>
<td>Components</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>7</td>
<td>New registration requests will be validated by a user organization representative before individual user access is provided. This validating role is called the buyer administrator</td>
<td>Customer access and control</td>
</tr>
<tr>
<td>8</td>
<td>A client organization's buyer administrator will be validated by a BASE employee before access is provided.</td>
<td>Customer access and control</td>
</tr>
<tr>
<td>9</td>
<td>The solution will make use of the single sign-on capability provided by the existing BASE corporate solution. Customer representatives who have been supplied with user ID and password to access the technical documentation repository will be able to use these credentials to connect to the online shop system</td>
<td>Customer access and control, Systems integration</td>
</tr>
<tr>
<td>10</td>
<td>The system will also use the user identity information from the existing corporate user identity repositories</td>
<td>Customer access and control, Systems integration</td>
</tr>
<tr>
<td>11</td>
<td>Because BASE is dealing with customers all across Europe, the site will support multiple languages and multiple currencies. These attributes will be stored in the user's profile. Changes will become effective the next time the user logs on. The first release will only contain English and German language</td>
<td>Customer access and control.</td>
</tr>
</tbody>
</table>

**Order subsystem**

<table>
<thead>
<tr>
<th>Requirement number</th>
<th>Requirement</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>The solution will provide client organizations with the ability to place an order through the Internet. It will service both - Requests from users from client organizations wishing to place an online order via Web browser and - Requests from compatible client organization systems that will interoperate with it to place an online order (using a machine interface)</td>
<td>Customer access and control</td>
</tr>
<tr>
<td>13</td>
<td>Users must authenticate in order to place an order.</td>
<td>Customer access and control</td>
</tr>
<tr>
<td>Requirement number</td>
<td>Requirement</td>
<td>Components</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>------------</td>
</tr>
<tr>
<td>14</td>
<td>Users will have the ability to place standing orders. Standing orders are orders with always the same shopping cart that will reoccur over time, following a given schedule</td>
<td>Order management</td>
</tr>
<tr>
<td>15</td>
<td>Users will have the ability to create template orders. Template orders are based on previous orders that can be modified to satisfy the current need. They then can be named and saved for further usage</td>
<td>Order management</td>
</tr>
<tr>
<td>16</td>
<td>Users will be able to use the bulk order form. Bulk order forms let users who already know what they need bypass the catalog browsing. This is done by a Web page with a list of entry fields where users enter several product stock keeping units (SKUs) into at once. By submitting the bulk order form, all products are put into the user’s shopping cart</td>
<td>Order management</td>
</tr>
</tbody>
</table>
| 17                 | Order approval functions will be as follows:  
- Approval must be provided by a client organization's buyer approver for orders over a pre-defined monetary value.  
- The buyer approver will have the ability to configure the trigger level for approval of orders for their company.  
- The system will notify the buyer approver as well as the buyer every time  
  - An approval is necessary  
  - An approval has been given  
  - An order has been rejected by the buyer approver | Order management, Customer access and control |
| 18                 | Order status functions will be as follows:  
- Users will be able to view the status of current and past orders. This also includes orders placed through other channels.  
- Authorization control will provide the ability for a standard buyer to see all orders placed by that user.  
- Buyer approvers will have the ability to access all orders placed by their organization, regardless of the channel that was used for placing the order. | Order management, Customer access and control |
Chapter 5. Proposing a solution

<table>
<thead>
<tr>
<th>Requirement number</th>
<th>Requirement</th>
<th>Components</th>
</tr>
</thead>
</table>
| 19                 | Machine interface features will be as follows:  
- The functionality offered by the machine interface will be limited to placing an order and checking the status of a current or past order.  
- The following data will be delivered to the client organizations in industry standard XML format:  
  - Product information,  
  - Pricing information,  
  - Order request,  
  - Order status request and  
  - Error information for the configuration and operation of the machine interface | Product and catalog, Order management, Customer access and control |
| 20                 | Orders are paid by invoice. There will be no online payment facility | Order management |

**Back-end integration subsystem**

<table>
<thead>
<tr>
<th>Requirement number</th>
<th>Requirement</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>Pricing information will be retrieved from the pricing system within the BASE organization</td>
<td>Systems integration</td>
</tr>
<tr>
<td>22</td>
<td>Inventory information will be retrieved from the inventory system within the BASE organization</td>
<td>Systems integration</td>
</tr>
<tr>
<td>23</td>
<td>Fulfillment capability and order status information will be retrieved from the fulfillment system within the BASE organization</td>
<td>Systems integration</td>
</tr>
<tr>
<td>24</td>
<td>Detailed product information and product images will be retrieved from the product management system within the BASE organization</td>
<td>Systems integration</td>
</tr>
</tbody>
</table>

**Customer care subsystem**

<table>
<thead>
<tr>
<th>Requirement number</th>
<th>Requirement</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>Users will be able to interoperate with BASE Customer Service Representatives (CSRs) via text-based chat for support and trouble shooting</td>
<td>Customer care</td>
</tr>
<tr>
<td>26</td>
<td>CSRs will be able to view user interaction with the system to assist users</td>
<td>Customer care</td>
</tr>
<tr>
<td>27</td>
<td>Only non-anonymous users will be able to use the customer care facility</td>
<td>Customer access and control</td>
</tr>
</tbody>
</table>
5.5.8 Out of scope

To ensure there is no confusion on what is believed to be part of the proposal and what is not, this section may be provided. The following is a typical example that is relevant for this proposal.

**Client interfaces**

When we deal with online shoppers, we take no responsibility for the client environment. We assume they have an industry standard browser that will be capable of interoperating with our online store via an HTML over HTTP interface. Their browser can support SSL, redirection and other standard features used for browser and HTTP server interaction. It is irrelevant what platform or operating system they use since we do not have control over their environment. We do, however, state support for a minimum level of browser as stated in the requirements section of this redbook.

Similarly, we assume BASE’s customers interacting via the machine interface are capable of operating with an XML over HTTP interface that conforms to the cXML specification. We do not take responsibility for their technology choices.
5.5.9 Product suite

In this section of the proposal, we would detail the full product suite that would be used to implement our solution. We have not included this level of detail in our redbook. See Chapter 6, “Operational overview” on page 155 for a discussion of the products that we believe would be used in our proposed solution.

5.5.10 Proposed project phases

To ensure a solution that best fits the needs of the BASE organization well into the future, it is proposed that delivery span two phases.

It is proposed that phase one commence with a two-week detailed study of requirements to allow those to be further refined so as to ensure that both current and future requirements are well understood. Following the requirements refinement exercise and sign-off by BASE would be a five-week architectural specification exercise. This exercise is to deliver the high-level design of the solution for the online order capture business system. Phase one also delivers a detailed statement of risks, dependencies, resource requirements, the length of the project and pricing information for phase two.

Phase two would commence with BASE acceptance of the architectural specification delivered from phase one. Phase two is the detailed design, build and implement phase. The solution is constructed and integrated into the BASE operational environment.

5.6 Functional view of the architecture

Constructing a functional view of the architecture not only provides a description of the components that make up the architecture in full; it also helps to ensure the architecture is complete for the solution for which it is being designed. This is partially achieved by defining components, detailing the relationships between components along with the interactions between components. This exercise often highlights where some of the solution may not be complete.

Some of the components that make up this system were first introduced in Chapter 3, “Requirements” on page 37. The main areas of functionality, along with users of the functionality, were identified so that the Business patterns could be identified. These functional areas can be added now as components.

Constraints, such as the need to interoperate with existing systems, identify other components.
Yet other components that are more technical are added as we piece together the solution. Some have been added as we have thought about the Application patterns and the structure of areas of the solution identified when Runtime patterns for BASE’s solution were defined. Some of these components will appear as nodes in the operational model.

5.6.1 Component list

Along with a description of the component is a statement detailing whether the component is an existing component that remains unchanged, is existing and modified or has to be provided for the BASE ordering system. The following are the major components employed within the BASE order capture system.

<table>
<thead>
<tr>
<th>Component</th>
<th>Comments</th>
<th>Implementation Details</th>
<th>Unchanged Modified or New</th>
</tr>
</thead>
<tbody>
<tr>
<td>configurator</td>
<td>Allows the user to configure complex products.</td>
<td>Custom developed.</td>
<td>New</td>
</tr>
<tr>
<td>customer assistance</td>
<td>Customer assistance as the name suggests provides the functionality to allow a CSR to give assistance to customers using the system. Assistance may be via a collaborative session or a CSR performing some task on behalf of the user.</td>
<td>Provided by WebSphere Commerce Business Edition product, configuration, customization and extension.</td>
<td>New</td>
</tr>
</tbody>
</table>
### Component | Comments | Implementation Details | Unchanged Modified or New
---|---|---|---
customer registration and preferences | Customer registration and preferences is existing Self-Service functionality that allows the customer to set their own preferences such as delivery address and also provides the functionality for a user to register for access to functionality or information made available from BASE's online presence. | WebSphere Configuration and development of existing system | Modified
notification | Notification is the functionality to send a message to a user. Typically used to notify a user of an event within the order capture system. | Provided by WebSphere Commerce Business Edition product, configuration, customization and extension. | New
order | This is the order handling component of the solution | Provided by WebSphere Commerce Business Edition product, configuration, customization and extension. | New
procurement system | External procurement systems belonging to an organization that will interface to BASE's order capture system. | System able to interface via XML/HTTP interface complying to cxml standard for commerce. Configuration | Modified
<table>
<thead>
<tr>
<th>Component</th>
<th>Comments</th>
<th>Implementation Details</th>
<th>Unchanged/Modified or New</th>
</tr>
</thead>
<tbody>
<tr>
<td>product</td>
<td>Product component of the order solution. Encompasses description, price, inventory, etc.</td>
<td>Provided by WebSphere Commerce Business Edition product, configuration, customization and extension.</td>
<td>New</td>
</tr>
<tr>
<td>product catalog</td>
<td>The catalog is a collection of product, pricing and availability information. It is available to the customer for perusal.</td>
<td>Provided by WebSphere Commerce Business Edition product, configuration, customization and extension.</td>
<td>New</td>
</tr>
<tr>
<td>product search and selection</td>
<td>Allows searching and selection of product in the catalog.</td>
<td>Provided by WebSphere Commerce Business Edition product, configuration, customization and extension</td>
<td>New</td>
</tr>
<tr>
<td>user</td>
<td>The user node is most frequently a personal computing device (PC) supporting a commercial browser, for example, Netscape Navigator and Internet Explorer. The browser is expected to support SSL and some level of DHTML. The two values listed in the next column are those officially supported by BASE. Other browsers may however be used by users.</td>
<td>Microsoft Internet Explorer 4 and above, Netscape Navigator 4 and above.</td>
<td>Unchanged</td>
</tr>
</tbody>
</table>
The following components were identified by the constraints imposed by the existence of legacy systems that would continue to be used with the order capture system.

<table>
<thead>
<tr>
<th>Component</th>
<th>Comments</th>
<th>Implementation Details</th>
<th>Unchanged Modified or New</th>
</tr>
</thead>
<tbody>
<tr>
<td>fulfillment</td>
<td>Existing fulfillment system</td>
<td>IBM P Series DB2 WebSphere MQ C++ WebSphere MQ, C++ and DB2 development and configuration</td>
<td>Modified</td>
</tr>
<tr>
<td>inventory</td>
<td>Existing inventory management system</td>
<td>IBM P Series DB2 WebSphere MQ C++ WebSphere MQ, C++ and DB2 development and configuration</td>
<td>Modified</td>
</tr>
<tr>
<td>price</td>
<td>Existing Pricing system</td>
<td>IBM P Series DB2 WebSphere MQ C++ WebSphere MQ, C++ and DB2 development and configuration</td>
<td>Modified</td>
</tr>
<tr>
<td>product management system</td>
<td>Existing host system that is a definitive reference of product for the BASE organization</td>
<td>IBM P Series DB2 WebSphere MQ C++ WebSphere MQ, C++ and DB2 development and configuration</td>
<td>Modified</td>
</tr>
<tr>
<td>Component</td>
<td>Comments</td>
<td>Implementation Details</td>
<td>Unchanged Modified or New</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>technical document server</td>
<td>This component is responsible for providing online users access to technical information on the products available from BASE Pty. Ltd</td>
<td>Complete system constructed on Lotus Domino. Technical information is provided by the product management system. WebSphere MQ is used to provide the communications channel to deliver the data to the technical document server.</td>
<td>Unchanged</td>
</tr>
</tbody>
</table>

The following components were identified in moving from Business to Application and Runtime patterns.

<table>
<thead>
<tr>
<th>Component</th>
<th>Comments</th>
<th>Implementation Details</th>
<th>Unchanged Modified or New</th>
</tr>
</thead>
<tbody>
<tr>
<td>application server</td>
<td>The application server node provides the infrastructure for application logic. The application server supports order, product, product catalog and other functional components of the solution.</td>
<td>What is the series of this box???? AIX operating system WebSphere Advanced Server</td>
<td>New</td>
</tr>
<tr>
<td>database server</td>
<td>The database server node provides the infrastructure for database</td>
<td>what series is this? AIX WebSphere MQ Series Java DB2</td>
<td>New</td>
</tr>
</tbody>
</table>
### Component

<table>
<thead>
<tr>
<th>Component</th>
<th>Comments</th>
<th>Implementation Details</th>
<th>Unchanged Modified or New</th>
</tr>
</thead>
<tbody>
<tr>
<td>directory and security services</td>
<td>The directory and security services node supplies information on the location, capabilities, and attributes (including user ID/password pairs and certificates) of resources and users known to this Web application system. This node also supplies information on users of the Web systems within the BASE organization. There are two instances of this currently in place. One holds employee details, the other customer.</td>
<td>IBM SecureWay Directory Server</td>
<td>Unchanged</td>
</tr>
<tr>
<td>domain name server</td>
<td>The DNS node assists in determining the physical network address associated with the symbolic address (URL) of the requested information. The DNS is that of the Internet service provider, although DNS is also implemented on the accessed site.</td>
<td>Not Controlled by the project</td>
<td>Unchanged</td>
</tr>
<tr>
<td>Component</td>
<td>Comments</td>
<td>Implementation Details</td>
<td>Unchanged Modified or New</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>firewalls</td>
<td>Firewalls provide services that can be used to control access from a less trusted network to a more trusted network. Traditional implementations of firewall services include: Screening routers (the protocol firewall in this design) Application gateways (the domain firewall) The two firewall nodes provide increasing levels of protection at the expense of increasing computing resource requirements. The protocol firewall is typically implemented as an IP router, while the domain firewall is a dedicated server node.</td>
<td>Firewall 1 AIX CISCO Router Configuration for new traffic</td>
<td>Modified</td>
</tr>
<tr>
<td>authentication proxy</td>
<td>Existing security infrastructure that controls access to Web assets for the BASE organization</td>
<td>Tivoli WebSeal Reverse Proxy Server Configuration changes for ordering system</td>
<td>Modified</td>
</tr>
<tr>
<td>mail application</td>
<td>Functionality to receive electronic mail. Typically used to notify a user of an event within the order capture system.</td>
<td>The mail application is provided by the user.</td>
<td>Unchanged</td>
</tr>
</tbody>
</table>
Chapter 5. Proposing a solution

<table>
<thead>
<tr>
<th>Component</th>
<th>Comments</th>
<th>Implementation Details</th>
<th>Unchanged Modified or New</th>
</tr>
</thead>
<tbody>
<tr>
<td>public key Infrastructure</td>
<td>PKI is a collection of standards-based technologies and commercial services to support the secure interaction of two unrelated entities (for example, a public user and a corporation) over the Internet. In the context of the topologies defined in this redbook, PKI supports the authentication of the server to the browser client, using the SSL protocol.</td>
<td>Not controlled by the project</td>
<td>Unchanged</td>
</tr>
</tbody>
</table>
## Component Details

<table>
<thead>
<tr>
<th>Component</th>
<th>Comments</th>
<th>Implementation Details</th>
<th>Unchanged Modified or New</th>
</tr>
</thead>
</table>
| Web server / redirector | A process that supports the use and management of Web pages and other Internet-related material.  
The Web server or HTTP server started out in life as a relatively simple document server. It served documents to client applications. The client applications were and typically are browsers. Client applications can however be any application that is capable of interacting with the server using the standard communication, security and document protocols along with the interaction and behavior provided by an application extending the http server's functionality.  

Plugins are used to extend the functionality of an http server by providing a link between an application and the http server.  

With this extended functionality, an http server is often viewed as the Web gateway into an application. | IBM HTTP Server. Plugin to support WebSphere and associated configuration will be added to support the newly added application server for the order capture system | Modified |
5.6.2 Component relationships

A full proposal should show relationships using UML diagram. We do not provide this level of detail in our redbook.

5.6.3 Component interactions

A full proposal should detail typical component interactions. We do not provide this level of detail in our redbook.
Operational overview

In this chapter, we will discuss the process that leads from the Runtime patterns that have been defined in Chapter 5, “Proposing a solution” on page 71 to the actual product mapping. Taking into account the detailed requirements that were discussed in Chapter 3, “Requirements” on page 37, this chapter will now describe which product (be it hardware, software or networking components) is being used where, and why. The outcome will be the detailed runtime product mappings for the BASE solution.
6.1 Introduction

We have now arrived at the last step of the design: the physical level of the architecture. As can be seen in Figure 6-1, the conceptual and the specification level have already been dealt with in Chapter 4, “Understanding the business problem” on page 55 and Chapter 5, “Proposing a solution” on page 71.

This chapter will conclude the selection of products that has already begun in the previous chapter of this redbook. It will discuss why a specific product has been chosen in the context of the BASE Pty. Ltd example. The sizing of the necessary hardware will be discussed briefly. We will then move on to develop the physical model of the solution out of the architecture material that has been gathered in the previous chapters.

![Figure 6-1 Overview of the design process](image)

When deciding on a platform for an e-business application, several issues have to be taken into account:

- The existing environment of the customer
- The choices the customer has made in the past
- The skill that is available (both on the developer and on the customer side)
- The customer’s requirements in terms of scalability and reliability

Chapter 5, “Proposing a solution” on page 71 has provided us with an overview of the Runtime patterns which can be seen in Figure 6-2. From this starting point, we will now elaborate on the actual products that will be used in the solution.
We will start with the basic components, the network, then add the hardware and the software.

![Diagram](image)

**Figure 6-2 The Runtime pattern mapping of the projected solution**

But first, we will discuss in more detail what can be seen in Figure 6-2. We will first describe the components in general terms in 6.2, “General description of the components” on page 158, then discuss the possible products in 6.3, “Choosing the components” on page 162 and finally, list the products that have been chosen for the solution in 6.4, “Performing the product mapping” on page 164.
6.2 General description of the components

This section describes the general components of our proposed solution.

6.2.1 Outside world

The Internet and the users and services connected to it, outside of the BASE network, are called the outside world. Important parts of the outside world include:

1. User
   The user maintains a TCP/IP connection to the Internet. He or she accesses the BASE Web site with one of the industry standard Web browsers described in “System usability” on page 49.

2. Public Key Infrastructure (PKI) node
   A Public Key Infrastructure node is a collection of standards-based technologies and commercial services to support the secure interaction of two unrelated entities (for example, a public user and a corporation) over the Internet. In the context of this redbook, PKI supports the authentication of the server to the browser client, using the SSL protocol.

3. Domain Name System (DNS) node
   The DNS node assists in determining the physical network address that is associated with the symbolic address (URL) of the requested information.

6.2.2 Demilitarized Zone (DMZ)

The Internet and the Demilitarized Zone are separated by a protocol firewall. The DMZ is a network area that is situated between the untrusted outside world of the Internet and the company network. Therefore, it is considered a high-risk zone for attacks from the Internet.

Only non-confidential material should be stored on a server in a DMZ in an unprotected form. Only approved TCP/IP ports are to be opened between the Internet and the DMZ, usually only ports 80 (HTTP) and 443 (HTTPS). The components include:

1. Protocol firewall and domain firewall nodes
   Firewalls provide services that can be used to control access from a less trusted network to a more trusted network, with the following traditional implementations:
   - Screening routers (the protocol firewall in this design)
   - Application gateways (the domain firewall)
The two firewall nodes provide increasing levels of protection at the expense of increasing computing resource requirements. The protocol firewall is typically implemented as an IP router, while the domain firewall is a dedicated node.

2. Load balancer node

The load balancer is a device used to ensure availability and performance for Web applications.

It manages a pool of servers and distributes the incoming workload evenly between these servers. In addition, if one of the servers in the pool goes down (because of a failure or due to a maintenance issue), the dispatcher reroutes all traffic to the remaining servers in the pool. This is done transparently for the user, who is not able to see how many servers are actually in the server pool. To achieve the highest possible availability, load balancing has to be done separately for each of the different server pools.

Since it is in a high availability environment, the dispatcher itself is a single point of failure and should be backed up by a second “hot standby” node.

3. Authentication proxy node

The authentication proxy node usually works as a reverse proxy, that is, it passes the request from the user to the actual Web server and does the same backwards with the answer from the Web server. Before doing that, it checks whether the user with the given credentials is actually allowed to access that part of the Web site. This is done by comparing the URL the user wishes to access with a list of restricted URLs. The user's credentials and the list of URLs he or she is allowed to access is usually stored in an LDAP database.

Refer to the redbook Enterprise Security Architecture using IBM Tivoli Security Solutions, SG24-6014 for more information on load balancing and authorization using IBM Tivoli products.

4. Web server/redirector node

The Web server/redirector nodes contain a Web server component and a redirector component. The Web server component serves static HTML pages and images to the browser client, whereas the redirector component handles requests for dynamic pages that make use of JavaServer Pages (JSPs) and Java Servlets. Requests like this are forwarded to an application server in the application zone (see Section 6.2.3, “Application zone” on page 160).

A redirector is used to enhance the security and the scalability: by separating the application server from the Web server (and thereby forming a multi-tier environment), it is possible to:

a. Put the application server on a separate machine (scalability is enhanced)

b. Put the application server behind an additional firewall, the domain firewall (security is enhanced)
5. Mail gateway node
   
   This is a system that can be used by applications to send out e-mail messages to users on the Internet. It is one-way only, that is, it will not receive e-mails from the Internet.

6.2.3 Application zone

   The application zone of the BASE Pty. Ltd network is protected by an additional firewall. No direct traffic from the Internet is allowed to the application zone. The application zone is therefore considered a safe zone for applications containing confidential material.

1. Application/integration server node
   
   This node provides the infrastructure for the business logic of a solution. The responsibilities of an application server include:
   
   a. Receiving requests from the clients through the servlet redirector plugin of the Web server
   b. Selecting the appropriate business logic based on these requests and executing it
   c. If necessary, cooperating with external services (for example, an LDAP directory) to retrieve data and execute external applications
   d. Putting together the response and dispatching it back to the client

2. Commerce application server node
   
   This node provides the infrastructure for the presentation and business logic of the online buying solution. The commerce application server may be part of an application server node.

3. Collaboration server node

4. Technical documentation repository node
   
   This node provides a document management system that contains the technical documents of BASE Pty. Ltd. It is not part of the solution we are going to develop for BASE and is already in place.

5. Customer registration system node
   
   This node provides a registration facility for the customers of BASE Pty. Ltd. As it is already in place, it is not part of the solution we are going to develop for BASE, but we will make some changes to it.
6.2.4 Back-end zone

This is the secure zone where the components of BASE’s key systems are located. It contains:

1. Database server node
   
   The database server node’s general function is to provide a persistent data storage and retrieval service in support of the user-to-business transactional interaction. The data stored is relevant to the specific business interaction.

   WebSphere Commerce is implemented using a relational database. The database server runs at least the WebSphere Application Server database and the WebSphere Commerce production database. The WebSphere Commerce database contains all information that is used by the WebSphere Commerce application, including information about individual shoppers, items and prices, etc.

2. Directory node
   
   The directory and security services node supplies information on the location, capabilities and various attributes (including user ID/password pairs and certificates) of resources and users known to this Web application system. The node may supply information for various security services (authentication and authorization) and may also perform the actual security processing, for example, to verify certificates. The authentication in most current designs validates the access to the Web application server and the database server.

   WebSphere Commerce supports integration with SecureWay Directory using LDAP, a client/server protocol, for accessing a directory service. WebSphere Application Server supports JNDI access to directory services.

3. Product management system node
   
   This node can be an already existing database server containing catalog data. In some cases, catalog data is imported from a back-end system into the Commerce Suite store catalog. If the commerce database is not the master catalog database, the replication of data from the back-end system will be run on a regular basis. Data can be imported into the commerce database from the back-end system.

4. Pricing system node

5. Inventory system node

6. HR system node
6.3 Choosing the components

Many of our component choices have already been determined either by the existing systems at BASE or by the prerequisites as specified in Chapter 3, “Requirements” on page 37. For example, we have established that technologies and products such as J2EE, WebSphere Application Server, WebSphere Commerce Business Edition, WebSphere MQ are being used or will be used.

6.3.1 Network considerations

- 100 megabit Ethernet is the company’s standard to connect between the components
- Availability requirements of the Web servers are high: therefore, redundancy within the DMZ is necessary
- Redundancy is achieved by using load balancers, with hot standby

6.3.2 Hardware considerations

According to the platform standards of BASE “Platform standards” on page 50, only IBM pSeries servers with AIX 5.1 or higher are allowed in the network segments other than the intranet. This means that all the server nodes in the solution will consist of pSeries servers. Because of this, we do not need any more of the decision process on the hardware platform for the solution.

To get the best scalability, availability and security and also to reuse the existing environment, we will use a three-tier WebSphere Commerce configuration plus load balancers.

Sizing the hardware

Note: Sizing the necessary hardware for a complex system like this is not a trivial task and is outside the scope of this redbook. Some information about hardware sizing can be found in the IBM WebSphere Commerce Suite Capacity Planning Guide Version 5.1 at:


Some of this material will still apply to WebSphere Commerce V5.4.

Determining whether caching can be used

The sizing of the hardware depends heavily on the way the solution is designed. For instance, if the system makes use of advanced features of WebSphere such
as the caching of pages, this will greatly reduce the overall performance need of the system. According to the capacity planning guide mentioned previously, on an AIX system, a performance factor of about four is feasible when using the caching feature.

We therefore have to determine whether caching of catalog pages can be used in the projected solution for BASE Pty. Ltd. This depends on the level of personalization of the individual pages.

The solution will provide different prices for each client organization (called user groups in WebSphere Commerce). This means that the catalog pages will be the same for two users from the same user group, but could be different for users from another user group.

Therefore, session-independent caching cannot be used, but session-dependant caching can. Session-dependant caching will not give us as much performance boost as session independent caching will, because the caching takes place on the application server and not on the Web server, but it will nevertheless enable BASE to get more performance out of a single machine.

Please note that caching cannot be used in WebSphere Commerce when URL rewriting authentication (instead of cookie-based authentication) is used.

6.3.3 Software considerations

WebSphere Commerce Business Edition V5.4 has already been decided upon by the customer to be the platform of choice for the online buying solution. This decision also settles many other software questions, because Business Edition needs a certain environment in which to run.

Business Edition comes with a list of additional software licences, dependencies and requirements that have to be satisfied in order to use it.

If you would like to review the complete list of software that is bundled with Business Edition, please refer to the IBM WebSphere Commerce Fundamentals Version 5.4.

We will list only the products that are of interest to our customer. These are:

- Application server: WebSphere Application Server, Advanced Edition 4.02
- Database server:
  - IBM DB2 Universal Database Enterprise Edition Version 7.2
  Can be used alternatively, but is not included:
  - Oracle Database 8.1.7, Enterprise Edition or Standard Edition
Web server:
- IBM HTTP Server Version 1.3.19.1
Can be used alternatively, but is not included:
- iPlanet Web Server, Enterprise Edition 4.1.8,
- Lotus Domino Web Server 5.0.6,
- Microsoft IIS 4.0/5.0

Authorization and authentication:
- IBM SecureWay Directory Server Version 3.2.1

Discussion
- DB2 is chosen as the database server, because it is one of the standards at BASE.
- IBM HTTP Server is chosen because it is compatible with BASE’s existing environment.
- IBM Tivoli Access Manager for e-business is chosen because BASE already uses it.
- WebSphere MQ is chosen because:
  - BASE already uses it
  - WebSphere Commerce has a connector for WebSphere MQ
  - It is ideal for asynchronous connections with guaranteed delivery
- We have no requirement for online payment functionality as provided by WebSphere Payment Manager, so we do not have to cater for the additional performance and hardware that is needed for it.

6.4 Performing the product mapping

We will now draw up a possible product mapping for the projected solution for BASE Pty. Ltd. Please note that other configurations are possible and real world choices would need to be made based on the customer’s requirements and infrastructure. Our select product mappings would take work, given the assumptions we have made about the requirements and environment at BASE. Also, since our redbook concentrates on the architecture and design phases of our project, we did complete an installation of our proposed solution. However, the choices made do reflect the real experiences of our redbook team.

6.4.1 Outside world

We have no power over the outside world, and therefore, we will not perform a product mapping for it.
6.4.2 Demilitarized Zone

Table 6-1 shows the products selected for the DMZ.

Table 6-1  Product mapping for the DMZ

<table>
<thead>
<tr>
<th>Node</th>
<th>Product mapping</th>
</tr>
</thead>
</table>
| Firewall                    | IBM SecureWay Firewall 4.2 configured as a protocol firewall. The following ports need to be opened to the Internet:  
  ▶ 80 for HTTP and Sametime tunneling  
  ▶ 443 for HTTPS  
  ▶ 119 for SMTP (outgoing mail)  |
| Load balancer node          | WebSphere Edge Server 2.0 and the WebSEAL Plugin for Edge Server (see authentication proxy node) |
| Authentication proxy node   | IBM WebSeal Plugin for WebSphere Edge Server (see load balancer node)            |
| Web server/redirector node  | IBM HTTP Server Version 1.3.19.1 with the servlet redirector plugin of IBM WebSphere Application Server, Advanced Edition. There are two Web server boxes in the server pool, both connected to a shared file system which contains the HTML and the image resources of the BASE Web site and also the product images of the WebSphere Commerce catalog. |
| Mail gateway                | AIX’s sendmail used as an SMTP server                                             |

**Note:** As can be seen in Table 6-1, we decided to put the authentication proxy components on the same physical machines as the load balancers. This frees up two machines that were underutilized and can now be used for other purposes.

This is one of the differences between specification level design and physical level design that 5.4.1, “Select Application and Runtime patterns” on page 76 discusses: on the specification level, this was designed as four separate components, now these four components actually fit onto two physical boxes.

Further microdesign might result in more changes in the same manner.
### 6.4.3 Application zone

Table 6-2 shows the products selected for the application zone.

**Table 6-2  Product mapping for the application zone**

<table>
<thead>
<tr>
<th>Node</th>
<th>Product mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firewall</td>
<td>IBM SecureWay Firewall 4.2 configured as an application gateway.</td>
</tr>
<tr>
<td>Commerce application server node</td>
<td>IBM WebSphere Commerce Business Edition Version 5.4.0.2 running on the</td>
</tr>
<tr>
<td></td>
<td>application/integration server node.</td>
</tr>
<tr>
<td>technical documentation repository node</td>
<td>J2EE application running on an IBM WebSphere Application Server, Advanced Edition</td>
</tr>
<tr>
<td></td>
<td>Version 4.02 (existing system, no changes necessary).</td>
</tr>
<tr>
<td>Customer Registration System node</td>
<td>J2EE application running on an IBM WebSphere Application Server, Advanced Edition</td>
</tr>
<tr>
<td></td>
<td>Version 4.02 (existing system, changes necessary).</td>
</tr>
</tbody>
</table>

### 6.4.4 Back-end zone

Table 6-3 shows the products selected for the back-end zone.

**Table 6-3  Product mapping for the back-end zone**

<table>
<thead>
<tr>
<th>Node</th>
<th>Product mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firewall</td>
<td>IBM SecureWay Firewall 4.2 configured as an application gateway.</td>
</tr>
<tr>
<td>Database server node</td>
<td>IBM Universal Database DB2 7.2 FP 6[?].</td>
</tr>
<tr>
<td>Directory node</td>
<td>IBM SecureWay Directory (existing system)</td>
</tr>
<tr>
<td>Product management system node</td>
<td>IBM DB2, C++ application (existing system)</td>
</tr>
<tr>
<td>Pricing system node</td>
<td>J2EE application running on an IBM WebSphere Application Server, Advanced Edition</td>
</tr>
<tr>
<td></td>
<td>Version 4.02 (existing system).</td>
</tr>
<tr>
<td>Inventory system node</td>
<td>DB2 (existing system)</td>
</tr>
</tbody>
</table>
6.5 Conclusion

Considering all the implications above, we end up with a product mapping similar to that shown in Figure 6-3.

![Figure 6-3 The runtime product mappings for the BASE solution](image)

The Application patterns, Runtime patterns and product mappings are intended to guide the architect in defining the application requirements, key features, basic design of the solution, and network layout. The actual application development has not been addressed yet. Please see Chapter 8, “Implementation guidelines” on page 195 for some guidance on the solution development.
Chapter 7. Detailed design

In the previous chapters of our redbook, we have discussed what the Composite patterns are and how they could map to the requirements of our imaginary customer, BASE Pty. Ltd.

This chapter provides methods and guidance on how to take the high-level system overviews produced by our Patterns for e-business methodology and transform them into more detailed technical design documents. The basis for this chapter are the functional requirements discussed in Chapter 3, “Requirements” on page 37. The original requirement should always be used to get a more detailed idea what we are going to deliver to the customer. The requirements exist to ensure that there is a clear understanding of how the application works and how it should look and feel.

This chapter details the following:
- Shop navigation flow
- Use case model
- Storyboard
- Component interaction
- Screens
- Styleguide
7.1 Scenario

At this point, we already have a high-level overview of the architecture we are going to use in our imaginary customer project. We now explain how to create the detailed design using several methods. Figure 7-1 shows the context of our redbook process and also shows that we have reached the tasks displayed as unshaded on the right-hand side.

![Figure 7-1](image)

This chapter covers topics in the Develop Physical Models section of Figure 7-1. In Chapter 3, “Requirements” on page 37, the functional and the non-functional requirements of the BASE Pty. Ltd are defined. We now go a step further and break down all these requirements into concrete specification documents. These documents can be used to start the development of the shop and its pages. The methods mentioned in this chapter no longer fit into a specific part of the Patterns for e-business methodology as described in the preceding chapters.

The styleguide, for instance, is used to integrate our development with the company’s Corporate Identity (CI); this activity belongs more to the non-functional requirements than to functional requirements, and is an activity that is outside the scope of the Patterns for e-business.

In order to produce a detailed design, the Web designer also needs the styleguide so that he/she can be sure to implement the CI into screens.

This chapter gives you guidance on how the system requirements break down and produce design documents that help give the customer a good understanding of the look and feel of the new application.
We discuss several design topics in detail. Once you have some experience in using these methods, you can apply them to your projects.

**Tip:** We recommend that you get your customer’s approval for all these design documents. In projects, these approvals are crucial documents used to save effort and reduce costs. For the customer, they serve as detailed descriptions of what we are going to build, and for the developers, they are the basis of where to start. Changes in defined requirements and merging these changes back into existing documents costs a great deal of time. Once development has started, it is very costly to start over again.

### 7.1.1 Shop navigation flow

The objective of this section is to give an overview of what a navigation flow is and how it looks.

In our sample solution, the shopper has many different ways to enter the shop. He can log in with a user name and password, which is received from his buyer organization’s administrator. If the customer does not have a valid user name and password, he has to request one at his buyer organization or he can browse through the catalog as an anonymous user, but he will not have access to all the functions that are available to a registered user.

If a valid user name and password are entered, the user will be redirected to a personalized entry page which depends on the contracts the buyer organization has closed with the owner of the shop.

After entering, the user can choose different ways to browse through the shop, perhaps by browsing the catalog or searching for a product using the search form.

The user has many decisions to make on his way through the shop. If you try to put all these possibilities into words, it results in an unmanageable document with a lot of if-then conditions.

A better way to point out all of the shopping possibilities is to use a flow chart. You start at one point and draw all possible ways the user can take. This procedure has to be followed for all possibilities. In Figure 7-2, you see an example of a flow chart.

We use the following symbols in the diagram:

- A rectangle represents the Web page the user is viewing.
- Connectors are the options the user has on a Web page.
- Actions are the actions a user initiates.
- Arrows show the next Web page the user accesses after choosing an action.

![Flow chart diagram](image)

At the starting point, the user has several choices to navigate. Every decision results in another page that is loaded. On this page, the user can choose where to go next, or if it is the last page, the user can only get back to previous pages.

In Figure 7-2, the individual pages are shown as a rectangular box and the different selections the user can make as arrows which lead to another page. Much more information can be added into this diagram; for example, we could identify pages that are generally available to all users compared to pages that are only for registered users.

Later in this chapter, we show the kinds of interactions that take place when the user chooses an action.

### 7.1.2 Use case model

This section elaborates the use case model, providing an example of how use cases work and how we can take advantage of them in the context of our sample solution at BASE. We base our use case model on the use case model that is a Work Product Description (WPD) from the IBM Global Service Method (the Method). The Method provides a mechanism for practitioners to reuse already collected knowledge and assets using a consistent, integrated approach, but the same general concepts apply to other methodologies you might use to implement an e-business solution.
Use case model description
The use case model uses graphical symbols and text to describe how users in specific roles can interact with the graphical user interface (GUI) of our application. A use case describes the interaction between the user and the GUI and not how the system interacts internally with other system components.

There are many specific constructs to describe a use case. It is not important to use all the use case constructs, particularly for small projects where the user interaction is well understood or if the project is based on an existing one which will be customized incrementally. So we describe only a subset of the constructs which we regard as the most important.

- Actors (name, description)
- Use cases (number, name, description, preconditions, associations, annotations)
- Communication associations between the actor and the use case
- Relationships between use cases
- Conditions affecting termination outcomes, main flow, alternate flows, exception flows
- Input fields

Use case model purpose
The main purpose of the use case model is to get an clear understanding of what the application should look like. If all eventualities are covered in a use case, you can be sure that you have considered all details of the system to be built.

Other benefits of a use case model include the following.
- It simplifies the communication between the end user and the application developer.
- You can estimate how much effort it costs to develop the solution.
- When you have broken down all interactions between the user and the system, you get an clearer overview of all the detailed components.
- It allows simultaneous development of different use cases because the scope of each use case is clearly defined.
- Test cases can be defined on the basis of the use cases.

Note: Further information about the IBM Global Service Method can be found at:

http://method.ibm.com
Use cases describe the interaction between the user and the Graphical User Interface (GUI), so if you decide not to produce use cases, you have to make sure that the requirements of the user interface are described in other design documents that meet the same goal. Otherwise, you may get lost in the complexity of the system.

**Use case model notation**

Actors and use cases combine to execute the interaction described in the use case. This communication is described as a communication association. The direction of the arrow shows the direction in which the communication is initiated. There is a set of such communication associations for each combination of actor and use case. Figure 7-3 shows an example communication association.

![Communication-Association diagram](image)

The customer who wants to order a product in an online store first will communicate with the use case "enter the store".

The customer initiates the action which is described in the use case “enter the store”.

To summarize all the main constructs of a use case model, Figure 7-4 shows a first overview of a use case example. All elements are displayed within the use case. Some can occur more often, such as the fact that there is not just one termination outcome. One main flow exists in the majority of the cases, but there are also several more flows such as exception flows or other conditions which can change the outcome.
To provide more detail and describe all topics belonging to the use case, we recommend that you use a table such as Table 7-1 to describe the use case content. A great amount of detailed information can be stored in such a table.

**Note:** In Table 7-1 we include only the use case constructs, which are most important to our example. There are many more constructs available.

**Table 7-1  Use case table**

<table>
<thead>
<tr>
<th>Use case #1</th>
<th>Use case name. A descriptive name for the use case.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>The actor that is initiating this use case. Enter the name and description in here.</td>
</tr>
<tr>
<td>Overview</td>
<td>A brief description (one or two sentences) of what this use case is about.</td>
</tr>
<tr>
<td>Preconditions</td>
<td>These are the preconditions which have to be fulfilled before you can access this use case; for example, some other use cases need to have been executed and terminated successfully before you can go on.</td>
</tr>
</tbody>
</table>
In this section, we discuss the use of a storyboard. It is a similar concept to that of a storyboard used in the film industry. It describes specific scenes with actors and what they can do in particular situations. It shows how different elements of the solution are positioned relative to each other and how they all fit into one framework. The storyboard describes how elements move through the scene, perform specific tasks, or exist in the background.

**Storyboard purpose**

Elements are gathered in one scene. After this scene successfully ends, the next one starts. At the end, all scenes are merged into one whole film. We apply this to our solution context where the storyboard consists of several scenes. Each scene represents a Web page we are going to create, and the scene contains all

<table>
<thead>
<tr>
<th>Association</th>
<th>Any association of several use cases, for instance, a list of all the other use cases which use or extend this one.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication association between the user and the system</td>
<td>Explain how the user interacts via the graphical user interface (GUI) with the system. For instance, the user clicks a button on the Web browser.</td>
</tr>
<tr>
<td>Termination outcome #1</td>
<td>Describe the way a use case can terminate. Multiple ways are possible. Successful and unsuccessful terminations can occur.</td>
</tr>
<tr>
<td>Termination outcome #2 (if needed)</td>
<td></td>
</tr>
<tr>
<td>Condition affecting termination outcome</td>
<td>There may be some “side effects” that occur, so if the termination outcome depends on such effects, detail this here. For every termination outcome, place a condition affecting the outcome. What event leads to an alternate or exception flow?</td>
</tr>
<tr>
<td>Use case description</td>
<td>Use a brief description of how the user can act on the page to which this use case belongs. It should describe what the user does and how the applications responds. Keep it simple and avoid looping.</td>
</tr>
<tr>
<td>Input summary</td>
<td>List all inputs the user can provide. For instance, list all fields of the form if there is a form on the Web page. This data will be validated later on in a command; the field values are the parameters on which the executing command acts. List all validation policies for these fields.</td>
</tr>
<tr>
<td>Annotations</td>
<td>These are some additional notes which you want to keep in mind for later purposes.</td>
</tr>
</tbody>
</table>

7.1.3 Storyboard
elements located on the Web site. Providing the explanation of what all these elements stand for is the task of the storyboard.

The main purposes of the storyboard are as follows.

- Define the content and functions of individual screens
- Navigation description
- Specify screen documents for reuse for screen design and programming

The storyboard can only be used if some preconditions already exist:

- Navigation flow must be completed
- Use case definition must be provided

The storyboard does not provide the following.

- No layout is discussed in the storyboard. This is the task of the screen designer.
- The graphics used in a storyboard are not for reuse in the production environment.
- The positioning of the elements in relation to one another is described in the styleguide and in the screen design.

We have used the navigation flow and the use cases to explain the functions which are present on a Web page in our solution. Now we go one step further in the design process and use the storyboard to list all elements which are present as interactive components on the Web pages. All buttons, links, etc. are described in a list of all elements and these elements are assigned to the Web pages defined in our navigation flow.

This continues our step-by-step approach to design, so that as we develop more detail, we get a clearer view of the solution and are less likely forget details such as a function or link we need for running our application properly.

**Storyboard notation**

The storyboard uses text and graphical models to describe the Web site elements. The graphics used in the storyboard are not for reuse in the production environment; they only reflect what kind of element is being described.

Every element on a storyboard page has to be described by its type and function. Remember, only elements defined in the storyboard are to be developed.
7.1.4 Component interaction

In this section, we describe how the interactions between components work. We show how the business logic is implemented and how we can insert dynamic data in static Web pages.

Displaying dynamic content

WebSphere Commerce uses JavaServer Pages (JSP) technology to display the dynamic content of Web pages. The JSPs use data beans, which represent a subset of the data stored in the database. WebSphere Application Server provides the run-time support for the JSP files.

Figure 7-5 shows how the WebSphere Commerce Server uses JSPs to display technology.

Dynamic pages are typically generated at request time. The content is assembled by combining static design elements and data out of the database. Static Web pages normally are used if the content does not change frequently. An example of a static Web page may be driving directions or any further contact information such as phone numbers. A JSP page acts as a template where the page design is delivered as static code snippets and the dynamic data (such as catalog or product information) comes out of the database.
These are the steps that are performed in WebSphere Commerce when a user requests a JSP page.

1. The Web server plugin parses the HTTP request, routes the request to the WebSphere Commerce Server, and performs workload balancing functions. In this case, a WebSphere Commerce command is invoked.

2. A command has a specific relationship to view tasks used to render the output for different devices. Here we have a view task which uses a JSP for rendering HTML content.

3. A JSP file is used for the response.

4. Inside the JSP file, data beans are used to access information from the database tables. An enterprise access bean converts an Enterprise JavaBean into a Java bean for use in a JSP file.

5. Using an Enterprise JavaBean, the data in database tables can be accessed and modified using SQL statements such as select, insert, update, and so on.

6. The HTML content for the response to the browser is rendered by the JSP engine of the WebSphere Application Server and is sent back via the Web server to the browser.

The main advantage of using JSP technology is to be able to separate the business logic from the page design. The Web designer can build static HTML pages, including designing the look and feel of each page using common Web development tools such as WYSIWYG editors. The store developer using programming skills in Java, JSP, HTML and JavaScript can add the dynamic elements to the HTML files. So later in the development process, both the Web designer and the store developer can update their pages without changing the other's work.

In addition to this dynamic display, JSPs mean you can easily create multilingual pages. All language-specific text can be stored in a property file and retrieved depending on the user locale's language.

For further information about using JSP files to insert dynamic content into static files, refer to the WebSphere Commerce Version 5.4 Online Help.

**Application component design**

In this section, we give a high-level overview of the component design used in WebSphere Commerce Business Edition. We explain how the business logic is implemented and which tasks are performed on user interaction.
The main components of Business Edition include:

- **Controller command**
  
  The controller command interacts directly with the Web controller. When the command terminates successfully, it returns the name of a view task which will be executed. The Web controller provides the correct implementation class of the view command and invokes it.

- **Task command**
  
  A task command implements only specific business logic. A controller command and a set of task commands together implement the application logic.

- **View command**
  
  A command that is used to compose a view as a response to the client. In this case, a JSP file is invoked.

- **Data beans**
  
  This is a Java bean which represents data in the database so that the store developer can access attributes and methods on the data without having to know about the specifics of the database.

- **Enterprise JavaBeans**
  
  These provide the persistence layer within the WebSphere Commerce architecture. There are two types of entity beans: container managed persistence (CMP) entity beans and bean managed persistence (BMP) entity beans.

We are focused only on the high-level view of the component design, so if you need more detailed information, refer to the redbook *B2B e-commerce Using WebSphere Commerce Business Edition Patterns for e-business Series*, SG24-6181, or consult WebSphere Commerce Version 5.4 Online Help.

In Figure 7-6, you can see the relationships between commands and the command flow used in WebSphere Commerce.
We use the basic `ProductDisplay` command of the sample store to explain the command flow.

1. The user accesses a Web page via a browser. This request goes to the Web controller. For instance, if the user selects the product detail page, the following URL is called:

   http://host_name/webapp/wcs/stores/servlet/ProductDisplay?productId=123&storeId=1
2. The **ProductDisplay** command is called and the parameters are forwarded to it. Validation of the business policy takes place in the controller command. If the validation finishes successfully, a task command is invoked by the controller command, which invokes an access bean; the access bean invokes the corresponding entity bean to receive information out of the database.

The parameters needed by the command to execute successfully are shown in Figure 7-7. Not all parameters are mandatory.

![Diagram of ProductDisplay](image)

**Figure 7-7**  Parameters needed by the **ProductDisplay** command.

The parameters include:

- **http://host_name/path/**
  The fully qualified host name of your WebSphere Commerce Server and the configuration path.

- **langId**
  Sets the current language ID for the duration of the session.

- **productId**
  The reference number for the catalog entry (SKU, product, package or bundle) that is to be displayed.

- **partNumber**
  The store’s unique identifier (or SKU) for the catalog entry that is to be displayed.

- **memberId**
  The reference number of the member who owns the catalog entry. The memberId, along with the partNumber, uniquely identifies the catalog entry. If the memberId is omitted, then the owner of the current store and the partNumber are used to uniquely identify the catalog entry.
3. The Web controller returns the view task name and invokes it. The corresponding JSP page is invoked.

4. Inside the JSP, page data beans, which are extended by the access beans, are initialized.

5. An entity bean is invoked to query the database for the data needed to display all product information.

6. The final HTML file is rendered and sent as response to the user's browser.

For further information on commands and command flow, see the WebSphere Commerce Version 5.4 Online Help.

### 7.1.5 Styleguide

This section addresses the use of a styleguide in the development process for corporate applications. The number of Web pages provided by companies has increased in recent years so that it has become more and more important that companies keep all design information gathered in some central documents. All information on how a corporate Web page should look and the policy these pages should follow should be included in such documents.

The aim is to ensure that graphic artists designing Web pages for different projects can work simultaneously, but the graphical user interface (GUI) developed is the same as in every other Web project and is based on these central style documents.

**Styleguide purpose**

The main purpose of having a document like the styleguide is to assure that everyone who is involved in a Web project front end design team knows which style definitions are to be used.

The most important definitions in a styleguide include:

- **Colors**
  
  Define all the different colors that occur in the project, for example state that the background color is to be white. In HTML, this would be:

  `<bgcolor="#FFFFFF"` >

  Define text colors, and so on.

- **Screen layout**
  
  Define the different sections into which the screen can be logically grouped; for instance, screens can split into a main navigation section, a content section, and so on.
Typographical definitions

Define which type of font, font family, sizes, etc. are used in different cases, for instance in HTML:

<font face="Arial, Helvetica, sans-serif" size="13px">

**Important:** Use only Web fonts that the major browsers support instead of corporate fonts. When it is mandatory to use other fonts, you should create text images and insert the images into your Web site; this results in larger HTML files.

Element positioning

Define the spaces between the page elements using pixel notation so the Web developer can exactly format the look and feel of the pages.

Sizing

Define the minimum and maximum size a page element can have; for instance, there are many products in your catalog and all have their own image sizes. You must decide how this fits into the layout without damaging the whole appearance of a page when an image is bigger than expected.

**Tip:** We recommend the use of style sheets. All of the above styles are gathered in one file that is easy to access and edit. Once the browser has downloaded the style sheet, it holds it in the cache.

### 7.1.6 Screens

In this section, we explain how the screens can be used to show detailed front end design. This provides something like a prototype that does not have any function. Screens only look like the final page but do not have any function. This helps the customer understand the solution that will be developed. All the previous activities described in this chapter are now gathered together as inputs to help produce a screen mock-up.

**Screens purpose**

A screen is an image which is produced in an image editing program. It is designed exactly as the Web page should look when it is finished in HTML. All the elements of the design are on the screen, positioning uses real spaces and real graphics are used.
In addition, some more details that must be considered are:

- The Corporate Identity (CI) policy of existing pages must apply to the design decisions.
- All elements appearing on the page must be in proper relationship to the others.
- Details described in the styleguide must be applied.
- No functions are behind the screen. For instance, a pull-down menu cannot be opened.
- The screens file should be in a standard image format such as the Graphical Interchange Format (GIF) or the Joint Picture Expert Group (JPG/JPEG).
- The screen should fit in a browser frame so that the real proportions of the browser’s menu bar are clear.
- Screens should be created for different screen sizes such as 1024x768 or 800x600.
- Every single Web page has to be represented by one screen, even though they may differ only in slight details.

### 7.1.7 Next steps

All required information is now available to start programming.

A common first step is the creation of the HTML files according to the given specifications. These files must meet all criteria discussed in this chapter. Also consider other criteria; for instance, even in different browser types and using different browser versions, the HTML files should look the same.

A following step is to combine these static HTML files with dynamic content using the current technologies available, such as JavaServer Pages (JSP).

### 7.2 Samples

In this section, we apply our chapter’s techniques and methods to the requirements of our imaginary customer. The starting point is the sample store ToolTech. We provide examples of how to extend ToolTech with the additional features required by BASE.

#### 7.2.1 Shop navigation flow

Here, we provide the shopping flow of our sample customer BASE Pty. Ltd. As a basis, we use the shopping flow of ToolTech and modify it to suit our needs.
We describe the basic shopping flow of a registered customer placing an order. This is shown in Figure 7-8.

Figure 7-8  BASE Pty. Ltd order flow
7.2.2 Use case model

This section gives an example of how we apply the use case model approach to the customer’s need. We chose our imaginary customer, BASE Pty. Ltd, and one functional requirement to explain. The chosen requirement is customer care as described in “Customer care subsystem” on page 46. For more details on customer care in our sample, see Chapter 10, “Customer care” on page 273. We use Table 7-1 to show the details of how the customer gets in contact with a Customer Service Representative (CSR).

Table 7-2 Customer care use case

<table>
<thead>
<tr>
<th>Use case #1</th>
<th>Customer care online chat assistance.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actor</strong></td>
<td>The actor is a registered user who belongs to a buyer organization and is allowed to order goods within a specific contract closed between the buyer organization and BASE Pty. Ltd..</td>
</tr>
<tr>
<td><strong>Overview</strong></td>
<td>This use case describes the help a customer can get if he faces any problems while browsing through the shop or needs any further assistance from the help center, using an text based chat interface.</td>
</tr>
<tr>
<td><strong>Preconditions</strong></td>
<td>The user must be a registered user within a buyer organization. He also must be logged in. He must have entered the shop through the main entrance page. The user must enable JavaScript in his browser.</td>
</tr>
<tr>
<td><strong>Association</strong></td>
<td>There are no relations between other use cases.</td>
</tr>
<tr>
<td><strong>Communication association between the user and the system</strong></td>
<td>The user clicks the <strong>Customer Care</strong> link.</td>
</tr>
</tbody>
</table>
| **Termination outcome** | **Main flow**  
A new, small browser window opens for the user. This contains a chat interface. He is immediately in contact with a customer care assistant. He enters some text in the input field and the customer care assistant responds.  

**Alternate flow 1**  
Nothing happens, no window opens.  

**Alternate flow 2**  
The window opens, but the chat interface is not usable. |
### Condition affecting termination outcome

<table>
<thead>
<tr>
<th>Condition</th>
<th>General application system failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>General application system failure</td>
<td>The systems Sametime server is not reachable because of system failure. An exception flow has to be discussed.</td>
</tr>
<tr>
<td>Users system failure</td>
<td>The user’s system does not meet the requirements. No exception flow is applicable. Workaround: define a minimal system requirement section on the Web page where every user can prove that they are able to use the system properly.</td>
</tr>
<tr>
<td>Customer care assistant is busy or out of office</td>
<td>Exception flow has to be discussed.</td>
</tr>
</tbody>
</table>

| Use case description                          | The user is viewing the product detail page. There is a link where he can get assistance. The user clicks this link to get in contact with a customer care representative. The user initiates the communication. |

| Input summary                                  | No input fields are required. Only the chat window offers a text input area.                        |

| Annotations                                    | This feature enriches user experiences and customer interactions. This live help functionality allows the partner, suppliers and customers to have real time communication with BASE Pty. Ltd for decision-making and negotiating. |

#### 7.2.3 Storyboard

For a better understanding of the storyboard’s character, we give an example using a simple contact form, as shown in Figure 7-9.
Figure 7-9  Contact form storyboard sample page
7.2.4 Styleguide

This section shows a more detailed examples of the information a styleguide can include.

We chose to explain the product display page. In Figure 7-10, we split the whole browser window into several sections. Each section can be considered a module with exactly defined functionality. The following modules have been identified.

- **Header**
  The header shows the company’s logo and some graphic design layout. It is visible on every page of the shop.

- **Head navigation**
  The head navigation includes the crucial navigation points such as a personalizable welcome message, the home button to get back to the entry point of the shop, the catalog button to browse through the catalog, the account button with which the user can view and edit his personal account details, the requisition list button to get quick access to often ordered goods, the current order button, the order status button and the logoff button.

  To grant quick access to these points, the head navigation is visible on every page of the shop.

- **Side navigation**
  The side navigation includes special features such as search and quick order functionality.

- **Main content**
  The main content window is the crucial section of the page. The main content is the target where all links are loaded when you select a button such as **Catalog browse**.

Figure 7-10 shows this layout.
After designing the browser content, the architect decides how the page framework will be built. There are several possibilities available. To ensure that some modules are visible on every page, you can use framesets so that only the content that changes has to be reloaded. Otherwise, you can create one JSP file which includes all the modules as standalone files.

The single modules are then described in more detail. See Figure 7-11 on page 192 for an example. We take the product details description and define the measurements of all the elements exactly. These are given in pixel notation. The Web developer is then able to build these code snippets as a template without having production data such as real product images or descriptions.
7.2.5 Screens

As mentioned before, we use the ToolTech sample store to explain what a screen means in Web design context. For illustrative purposes, we use the product detail page as shown in Figure 7-12.
Figure 7-12 shows how the end product should look when all development work is done. The integrated company CI of ToolTech is used. This can be seen in the use of the logo, the font styles and the colors. The customer can easily see whether the project is on the right track. When we do not use screens, the customer is not able to see the page design until the HTML files or JSP files are built in a later phase of the development cycle. Much rework could result, including a review of the requirements if the customer disagrees with the provided interface. This can put the project schedule at risk.

We recommend that you reach an agreement about the screens design with your customer before you go on to the next step of development.
## 7.3 Summary

In this section, we provide a summary of our chapter as a guideline for use in design development.

The individual design steps in the correct order are:

1. Get all requirements from your customer.
2. Analyze existing systems, if they exist, and get the design guidelines, such as the CI documents.
3. Create navigation flows of the scenes.
4. Inherit the sequences of the navigation flow and define the use cases.
5. Define in the storyboard all screen elements visible to the user and include all functions of these elements.
6. Define the styleguide document.
7. Apply all this information to screen designs.
8. Create HTML files.
9. Create custom commands, EJBs, view commands and beans.
10. Begin the JSP programming.

We recommend that you get the customer’s approval after the first crucial steps, so that the project schedule remains on the right track.
Implementation guidelines

This chapter builds on Chapter 7, “Detailed design” on page 169 by providing guidelines for developing and implementing a B2B site. We describe runtime methodologies and high-level installation steps for implementing WebSphere Commerce Business Edition V5.4 using AIX. We also discuss development planning, including a gap analysis for customization using our example site; we then describe the Business Edition development tools and development environment. Finally, we discuss guidelines for testing, including test planning and setting up a test environment using Business Edition.

The guidelines and implementation information found in this chapter include:

- Runtime environment
- Application development
- Testing
8.1 Introduction

Following on from the previous chapters, we now have a product mapping and design guidelines for our example. In the context of the patterns, we are in the right hand quadrant of the Figure 8-1. Guidelines for the runtime implementation, application development and testing are therefore part of the physical level of the patterns.

Figure 8-1 Overview of chapter context within the patterns framework

8.2 WebSphere Commerce Business Edition runtime

In this section, we will describe the implementation of an AIX enterprise two-tier runtime using DB2 and WebSphere Commerce Business Edition server in our test environment. This example is based on the detailed product mapping, in Chapter 6, “Operational overview” on page 155. For detailed installation instructions, refer to IBM WebSphere Commerce for AIX Installation Guide for use with a DB2 Universal Database Version 5.4.

Note: We downloaded IBM WebSphere Commerce for AIX Installation Guide for use with a DB2 Universal Database Version 5.4 from the technical library section of the WebSphere Commerce Web site at:

http://ibm.com/software/webservers/commerce/wc_be/lit-tech-general.html

The BASE Runtime pattern and product mapping are specified in the implementation shown in Table 8-1.
Although our example solution for BASE specified a three-tier configuration, for the purposes of testing our architecture and customizations, we implemented a two-tier node. Refer to the redbook *WebSphere Commerce Suite V5.4 Handbook, Architecture and Integration Guide*, SG24-6567 for a detailed discussion of a three-tier AIX implementation. In this example, we will implement the following nodes:

- WebSphere Commerce Business Edition V5.4
- IBM DB2 7.1.0.55

**Note:** The default browser for AIX 5.1 is Netscape, but you can only access the WebSphere Commerce tools and online help using MicroSoft Internet Explorer 5.5 from a Windows operating system on the same network as your WebSphere Commerce machine.

### 8.2.1 Two-tier configuration

We installed a two-tier configuration with the database server installed on a separate machine from all other WebSphere Commerce components. We
organized the two-tier WebSphere Commerce runtime environment for AIX installation into the following phases:

1. Install the DB2 server on machine B
2. Install the DB2 client on machine A
3. Configure and verify DB2 connectivity
4. Install the WebSphere Application Server on machine A
5. Install WebSphere Commerce Business Edition on machine A
6. Complete Business Edition Instance pre configuration steps, including:
   - Running the wpostinstall.sh script
   - Starting the WebSphere Application Server
7. Create a Business Edition instance
8. Complete Business Edition instance post-configuration steps, including
   - Enabling SSL for testing
   - Compiling the JavaServer Pages files
9. Publish the sample store

**Note:** As this was our first install of WebSphere Commerce Business Edition on AIX, we allowed four days to complete the two-tier configuration.

**Port numbers used by Business Edition**

Business Edition uses a variety of ports that other applications may not use. During the installation, you need to be aware of the ports shown in Table 8-2.

<table>
<thead>
<tr>
<th>Port Number</th>
<th>Used by</th>
</tr>
</thead>
<tbody>
<tr>
<td>2222</td>
<td>The default port through which we accessed the WebSphere Administrative Console as a non-root user.</td>
</tr>
<tr>
<td>8000</td>
<td>WebSphere Commerce tools, including Store Services and the Administration Console.</td>
</tr>
<tr>
<td>9000</td>
<td>WebSphere Application Server Location server</td>
</tr>
<tr>
<td>50000</td>
<td>DB2 Database server</td>
</tr>
</tbody>
</table>
Locales used by Business Edition

Business Edition uses valid Java locales only. Although our BASE example is a multilingual site, we used English as our default language. We used the following command to determine our locale:

```bash
# echo $LANG
```

Web addresses, User IDs and passwords

The two-tier installation of Business Edition requires a variety of Web addresses and user IDs. Some of these are default and some you are either required to create or are prompted to change. We kept track of our user IDs and passwords by creating Table 8-3.

### Table 8-3  Business Edition user IDs and passwords

<table>
<thead>
<tr>
<th>Component</th>
<th>User ID</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 server</td>
<td>db2inst1 (default)</td>
<td>db2inst1 (default)</td>
</tr>
<tr>
<td>DB2 client</td>
<td>db2inst1 (default)</td>
<td>db2inst1 (default)</td>
</tr>
<tr>
<td>AIX Non-Root User ID</td>
<td>wasuser (created)</td>
<td>wasuser (created)</td>
</tr>
<tr>
<td>WebSphere Commerce Configuration Manager</td>
<td>webadmin</td>
<td>wcsadm1n (changed from webibm)</td>
</tr>
<tr>
<td>IBM HTTP Server</td>
<td>db2inst1 (created)</td>
<td>db2inst1 (created)</td>
</tr>
<tr>
<td>WebSphere Commerce instance administrator</td>
<td>wcsadmin</td>
<td>wcsadm1n (changed from wcsadmin)</td>
</tr>
<tr>
<td>including Store Services</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://m10df.itso.ral.ibm.com/storeservices">http://m10df.itso.ral.ibm.com/storeservices</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8.2.2 Hardware and software prerequisites

For detailed information on hardware and software prerequisite, refer to Chapter 2, “Applying patterns” on page 9. Pre-installation instructions can be found in the IBM WebSphere Commerce for AIX Installation Guide for use with a DB2 Universal Database Version 5.4.
Hardware used in our test environment

We used the hardware shown in Table 8-4 in our test environment.

Table 8-4  Test machine specifications

<table>
<thead>
<tr>
<th></th>
<th>Commerce application server (machine A)</th>
<th>Database server (machine B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine type</td>
<td>IBM RS/6000 44P Model 170 (7044-170)</td>
<td>IBM RS/6000 44P Model 170 (7044-170)</td>
</tr>
<tr>
<td>Processor</td>
<td>1 x 450 MHZ POWER3</td>
<td>1 x 450 MHZ POWER3</td>
</tr>
<tr>
<td>System memory</td>
<td>1 GB</td>
<td>1 GB</td>
</tr>
<tr>
<td>Disk space allocation</td>
<td>/usr 4GB /tmp 300MB</td>
<td>/usr 1GB</td>
</tr>
</tbody>
</table>

Note: In addition to the above configurations, we also allocated 1 GB of paging space per CPU (refer to Appendix A, “E-commerce business models” on page 311). Refer to the AIX tasks of the IBM WebSphere Commerce for AIX Installation Guide for use with a DB2 Universal Database Version 5.4 for detailed instructions on the following tasks:

- Allocating and mounting a CD file system
- Increasing free space
- Paging space
- Finding the size of physical partitions

We recommend that you allocate adequate paging space and disk space before commencing the install.
Software used in our test environment

We used the software shown in Table 8-5 for our test environment and accepted the default installation paths:

Table 8-5  Test software specifications

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Application server (machine A)</th>
<th>Database server (machine B)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IBM HTTP Server 1.3.19.1</td>
<td>IBM DB2 Universal Database Enterprise Edition 7.1.0.60</td>
</tr>
<tr>
<td></td>
<td>/usr/HTTPServer</td>
<td>/usr/lpp/db2_07_01</td>
</tr>
<tr>
<td></td>
<td>IBM WebSphere Application Server, Advanced Edition 4.0.3 /usr/WebSphere/App Server</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IBM WebSphere Commerce Business Edition 5.4.2 /usr/WebSphere/CommerceServer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DB2 Administration Client /usr/lpp/db2_07_01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DB2 Application Development Client /usr/lpp/db2_07_01</td>
<td></td>
</tr>
</tbody>
</table>

Required filesets

In addition to the application software listed above, we used the wc54aixpreq.sh script to check that our AIX installation included the following files before commencing the install.

- x11.adt.lib
- x11.adt.motif
- x11.base.lib
- x11.base.rte
- x11.motif.lib
The pre-install requirements check script is available in the Software Patches directory on the WebSphere Commerce Disk 2 CD.

8.2.3 Installing a two-tier enterprise runtime environment

This section provides details of the installation tasks.

Install DB2 server on machine B and DB2 client on machine A

This section describes the high-level installation steps for IBM DB2 Database DB2 Universal Database Enterprise Edition 7.1.0.55 and installation of the DB2 client. We will also create a database for the WebSphere Application Server.

Pre-installation procedure
Before we installed DB2, we checked that a port was available for use by the remote database configuration on the AIX machine. DB2 uses 50000 and 50001 by default. We found the available ports for use in the /etc/services file and used port 50000. We also did the following:

1. Installed AIX 5.1 recommended maintenance package
2. Installed security fix for AIX logging vulnerability
3. Allocated extra disk space
   - /usr 4GB
   - /tmp 1 GB
   - /home 1GB
   - paging space 1GB per CPU

Installation procedure
If you use the Db2 UDB CD provided in the WebSphere Commerce product package, the CD includes and automatically install the FixPak 5 (level 7.1.0.55); refer to Chapter 3, “Requirements” on page 37. Install IBM DB2 Universal Database 7.1.0.55; look in the IBM WebSphere Commerce for AIX Installation Guide for use with a DB2 Universal Database Version 5.4 for detailed instructions.
Chapter 8. Implementation guidelines

1. Log in as user ID root and run the following command:
   
   ```
   # su - root
   # cd /usr/cdrom
   # ./db2setup
   ```

2. Highlight the following components from the DB2 Install window:
   
   - DB2 UDB Enterprise Edition
   - DB2 Application Development Client.

3. From the Create DB2 Services window, select **Create a DB2 Instance**.

4. Accept the defaults for creating a database instance and enter a password (db2inst1).

5. From the Create DB2 Services window, select **Create the Administration Server**.

6. Accept the defaults for creating a creating the Administration Server.

7. Accept the defaults for creating a Fenced User; do not set up the DB2 Warehouse Control Database, and complete the registration.

   **Note:** We used db2inst1 as our database instance owners user ID and password. We also used the same user ID and password for the administration client on the WebSphere Commerce machine. The instance IDs must be the same on both machines.

### Install Db2 V7.2 FixPak 6 (7.1.0.60)

The IBM DB2 Universal Database V7 FixPak 6 is required for Business Edition Fix Pack 5.4.0.2. Refer to the FixpakReadme.txt for details.

The high-level FixPak 6 installation steps are as follows:

1. Ensure that all DB2 processes have been stopped:
   
   ```
   # su - db2inst1
   $ db2 force applications all
   $ db2 terminate
   $ db2 stop
   $ exit
   ```
2. Download the FixPak 6 at:

3. From a command window as the root user, decompress the FixPak file to /tmp using the following command:
   ```
   # su - root
   # cd /tmp
   # uncompress -c FP6_U481406.tar.Z | tar-xvf -
   ```

4. Install the FixPak from the delta_install sub-directory using smitty as follows:
   ```
   # cd delta_install
   # smitty update_all
   ```

5. From the smitty window, select the current directory ./ as the input directory.

6. Update the DB2 instance using db2iupdt:
   ```
   # /usr/lpp/db2_07_01/instance/db2iupdt db2inst1
   ```

**Enable jdbc2**
To update the configuration for DB2 to use jdbc2, do the following.

1. Stop all DB2 services:
   ```
   # su - db2inst1
   $ db2stop
   ```

2. Use a text editor to add the following line to the .profile file located in the home directory /home/db2inst1:
   ```
   # . ./sqllib/java12/usejdbc2
   ```

**Update .profile to use extended memory**
We used extended memory for our DB2 server to improve performance. To enable extended memory, update the .profile located in the home directory /home/db2inst1 by doing the following:

1. Log in as the DB2 instance owner:
   ```
   # su - db2inst1
   ```

2. Add the following lines at the bottom of the .profile file:
   ```
   # EXTSHM=ON
   # export EXTSHM
   # db2set DB2ENVLIST=EXTSHM
   ```

3. Refresh the user environment by reloading the .profile file.
   ```
   $ . .profile
   ```

4. Verify the new settings with the following commands:
   ```
   $ echo $CLASSPATH
The CLASSPATH should include the updated path
/home/db2inst1/sqllib/java12/db2java.zip

$ echo $EXTSHM

$EXTSHM should be set to ON

$ db2set

The DB2 settings should include DB2ENVLIST=EXTSHM

---

**Note:** Although we did not enable it for this example, we recommend that you create a separate file system for DB2 if the file system is shared with other users or DB2 instances. Creating a separate file system will guarantee there will always be sufficient disk space available for database growth in a production Business Edition system. For detailed instructions on creating a separate file system for DB2, refer to Chapter 13, “AIX three-tier runtime with DB2 and IBM HTTP Server” in *WebSphere Commerce Suite V5.4 Handbook, Architecture and Integration Guide*, SG24-6567.

---

**Create the WebSphere Application Server repository database**

To create the WebSphere Application Server repository database, do the following.

1. Log in as the DB2 instance owner:
   
   # su - db2inst1

2. Create a new database called was:

   $ db2 create db was

3. Update the default application heap size for the was database from 128 to 512 pages:

   $ db2 update db cfg for was using applheapsz 512

4. Stop and restart DB2 to enable the changes.

   $ db2stop
   $ db2start

---

**Install the DB2 client on machine A**

The WebSphere Commerce machine will connect to the DB2 server via a DB2 client. A DB2 instance owner must be created on the client machine that matches the same DB2 instance owner ID and password on the server. We used db2inst1 as our instance owner ID and password to match the server. Since the client code must match the server, we installed and configured the same updates and JDBC drivers.
Installation procedure
If you use the DB2 UDB CD provided in the WebSphere Commerce product package, the CD includes and automatically install the FixPak 5 (level 7.1.0.55). Refer to Chapter 3, “Requirements” on page 37. Install IBM DB2 Universal Database 7.1.0.55 as detailed in the IBM WebSphere Commerce for AIX Installation Guide for use with a DB2 Universal Database Version 5.4 for detailed instructions.

The high-level steps to install the Db2 client on the Business Edition server are as follows.

1. Log in as user ID root and run ./db2setup from the IBM DB2 Universal Database CD.
2. Highlight and select the following components:
   - DB2 Administration Client
   - DB2 Application Development Client.
3. Create a DB2 instance for WebSphere Commerce called db2inst1, owned by db2inst1 (member of group db2adm1), with home directory /home/db2inst1.

Install Db2 v7.2 FixPak 6 (7.1.0.60)
For details, refer to “Install Db2 v7.2 FixPak 6 (7.1.0.60)” on page 206.

Enable jdbc2
For details, refer to “Enable jdbc2” on page 204.

Update .profile to use extended memory
For details, refer to “Update .profile to use extended memory” on page 204.

Step 3: Configure and verify DB2 connectivity
To verify the DB2 client/server connection, we need to catalog our server and database configuration on the client machine.

Catalog the DB2 TCPIP node
To catalog the Db2 TCPIP node, do the following:

1. Log in as the DB2 instance owner:
   
   # su - db2inst1

2. Define the remote DB2 server machine as follows:
   
   $ db2 catalog tcpip node m10df51f remote m10df51f.itso.ral.ibm.com server 50000
Catalog the WebSphere Application Server remote database
To catalog the WebSphere Application Server remote database, do the following.
1. Add an entry for the remote database to the local system database directory:
   $ db2 catalog db was at node m10df51f

Verify DB2 connectivity
To verify the connectivity between the DB2 server machine and the Business Edition Db2 client machine, do the following:
1. Connect to the was database by using the db2connect command as follows:
   db2 connect to was user db2inst1 using db2inst1
   The output will look like this:
   Database Connection Information
   Database server = DB2/6000 7.2.4
   SQL authorization ID = DB2INST1
   Local database alias = WAS
2. Disconnect from the database:
   $ db2 connect reset

The installation of the DB2 server and client is now complete.

Install WebSphere Application Server on machine A
Now that we have installed the DB2 server on machine B and the DB2 Client on machine A, we can proceed with installing WebSphere Application Server on machine A. As our test environment is a two-tier installation, we also installed the IBM HTTP Server on machine A. In our environment, we installed WebSphere Application Server V4 FixPak 3. This eliminated the need to install WebSphere Application Server V4 FixPak 2 and e-fixes.

Refer to Chapter 6, "Installing WebSphere Application Server" in the IBM WebSphere Commerce for AIX Installation Guide for use with a DB2 Universal Database Version 5.4 for detailed instructions.

Pre-installation procedure
An X-Windows terminal is required to install WebSphere Application Server. We ran the following command to add the hostname to the X-Windows access control list.

# /usr/bin/X11/xhost m10df51f.itso.ral.ibm.com
Installation procedure
To install WebSphere Application Server V4.0.3, follow these steps.

1. Log in as user ID root and run the IBM WebSphere Application Server, Advanced Edition CD by running:

   #su - root
   #cd /usr/cdrom
   # ./install.sh

   **Note:** The WebSphere Application Server installer checks for required filesets. As we are running AIX, all our filesets are at the required level or higher. You should see the following message:

   All packages appear to be proper versions.

2. Choose a custom installation and select the following packages:
   - Server
   - Admin
   - Samples
   - Application Assembly and Deployment Tools
   - IBM HTTP Server 1.3.19.1
   - Web server plugins

   **Note:** Be careful when making selections in the AIX WebSphere Application Server installer, as the default motif for unselected options is black. Ensure that you highlight and select the Custom installation option. The selected option will then appear grey.

3. Select the IBM HTTP Server plugin.

4. In the Database Options dialog box, complete the following:
   a. Select DB2 from the Database Type field pull-down menu.
   b. Select the Remote DB option.
   c. Enter was in the Database SID field.
   d. Select the full path name of /home/db2inst1 in the DB Home field by using the Browse button.
   e. Enter db2inst1 as the instance owner in the user ID and password fields.

5. Accept the default destination directory for the WebSphere Application Server: /usr/WebSphere/AppServer/bin.
6. From the Install Options Selected dialog box, click **Install** to begin the installation.

7. The Location of configuration files dialog box opens; before accepting the default install path `/usr/HTTPServer/bin` you must open another command window and run the following commands:

   ```
   # cd /usr/HTTPServer/conf
   # mv httpd.conf httpd.conf.orig
   # cp httpd.conf.sample httpd.conf
   ```

   **Note:** If you miss this step during installation, you can manually enter the `httpd` configuration changes. To do this, open the `httpd.conf.sample` file, copy the last three lines of the file and paste them into the bottom of the `httpd.conf` file.

8. When the Setup Complete dialog box opens, click **Finish** and unmount the CD.

**Install WebSphere Application Server V4 FixPak 3**

After installing WebSphere Application Server, we installed WebSphere Application Server V4 FixPak 3. For detailed instructions, refer to the FixPak Readme.txt file. The following are high-level instructions for installing FixPak.

1. Remove any libraries which have been loaded into system memory by running the following command:

   ```
   # /usr/sbin/slibclean
   ```

2. Download WebSphere Application Server V4 FixPak 3 Advanced Edition for AIX to the `/tmp` directory. You will require 75MB of free space in the `/tmp` directory.

   `http://www-1.ibm.com/support/manager.wss?rs=180&rt=0&org=SW&doc=4001292`

3. Decompress the tar file:

   ```
   # cd /tmp
   # tar -xvf was40_ae_ptf_3_AIX.tar
   ```

4. To run the FixPak installer, enter the following command:

   ```
   # cd tmp
   # ./install.sh
   ```

5. We selected the following options:

   - Accept license agreement
   - Update Application Server
   - Update JDK
   - Update IBM HTTP Server
Create a password for IBM HTTP Server administrator
As the IBM HTTP Server 1.3.19.1 requires authentication to access various configuration files, you need to create a user ID and password for the IBM HTTP Server 1.3.19.1. Perform the following steps.

1. Open a command window and run the following commands:

   # su - root
   # cd /usr/HTTPServer/bin
   # ./htpasswd -m ../conf/admin.passwd ibmadmin

2. When you are prompted for a password, enter:

   ibmadm1n

Set up the IBM HTTP Server administration server
To set up the IBM HTTP Server administration server, perform the following steps.

1. Open a command window and run the following commands:

   # su - root
   # cd /usr/HTTPServer/bin
   # ./setupadm

2. When prompted, complete the following:
   a. Enter httpadmin in the user ID and password fields.
   b. Enter httpadmin in the group field.
   c. Accept the default path /usr/HTTPServer/conf.
   d. Enter 1 to make changes.
   e. Enter 1 to update the administration server configuration file.
   f. Exit the setupadm program.

Verify the two-tier WebSphere Application Server configuration
To verify the functionality of the two-tier WebSphere Application Server runtime, do the following:

1. Log on to the WebSphere Application Server as the root user:

   # su - root

2. Start the IBM HTTP Server by running the following command:

   # cd /usr/HTTPServer/bin
   # ./apachectl start

3. Start the WebSphere administrative server by executing the startupServer script, as follows:

   # cd /usr/WebSphere/AppServer/bin
   # ./startupServer.sh &
4. Ensure the administrative server has started by checking the tracefile file located in /usr/WebSphere/AppServer/logs/. When the server has started, you should see the following message:

```
# server_adminServer open for e-business
```

**Note:** To monitor the administrative server startup, run the following command:

```
# tail -f /usr/WebSphere/AppServer/logs/tracefile
```

5. Start the WebSphere Administrative Console by running the following command:

```
cd /usr/WebSphere/AppServer/bin
./adminclient.sh &
```

6. When the console displays the message Console Ready, start the Default Server application in the WebSphere Administrative Console.

7. After the Default Server has started, test the Snoop servlet from a browser client:

```
http://m10df51f.itso.rai.ibm.com/servlet/snoop
```

**Note:** You can run the Snoop servlet from a browser on the WebSphere Application Server machine; it is not necessary to run the servlet from a Windows machine with Internet Explorer for this test.

---

**Step 4: Install WebSphere Commerce Business Edition**

After installing and configuring the IBM HTTP Server and WebSphere Application Server, as well as the DB2 Universal Database, we installed the X11.adt.motif fileset from the AIX 5.1 installation CD1.

We also ran the Installation and Configuration Checker, which can be downloaded from:

```
```

This tool ensures that we had the correct levels of software to support WebSphere Commerce. Instructions for running this tool can be found in the readme.txt file.

**Installation procedure**

1. Log in as user ID root and run the IBM WebSphere Commerce Disk 1 CD by entering:

```
# su - root
# smitty install_all
```
2. Select the following filesets:
   – Commerce Base:
     • 5.4.0.0 Commerce Base Files
     • 5.4.0.0 Commerce license
     • 5.4.0.0 Commerce properties
     • 5.4.0.0 Commerce runtime
     • 5.4.0.0 Commerce samples
     • 5.4.0.0 Commerce schema
   – Commerce.Docs
     • 5.4.0.0 Commerce en_us docs, readme, help

3. In the Detailed Output field, press the Tab key to toggle from No to Yes.

**Note:** The WebSphere Commerce installation will require 600 MB of disk space in /usr and may take up to 20 minutes to complete.

**Run the post-install script**

Run the post-install script wcpostinstall.sh to set up the Commerce environment and create a new user ID that will allow you access to WebSphere Application Server and Business Edition as a non-root user. The post install script chooses port 2222 by default as the non-privileged port number for the WebSphere Application Server bootstrap port. Ownership of the WebSphere Application Server directories will also be transferred to this new user.

To run the script complete the following:

1. Open a new command window and run the following commands:
   ```sh
   # su - root
   # export DISPLAY=m10df51f.itso.ral.ibm.com:0.0
   # cd /usr/WebSphere/AppServer/bin
   # ./adminclient.sh m10df51f.itso.ral.ibm.com
   ```

2. From the WebSphere Administrative Console, select and stop the m10df51f node. You will get a warning message, click Yes to continue.

2. Confirm that all WebSphere Application Server services have stopped by running the following command:
   ```sh
   # ps -ef | grep AppServer | grep -v grep
   ```

3. Type the following command to run the install script:
   ```sh
   # cd /usr/WebSphere/CommerceServer/bin
   # ./wcpostinstall.sh
   ```
4. Follow the on-screen prompts, selecting **Yes** when asked if you want to run as a non-root user. In our example, we used the settings shown in Table 8-6.

**Table 8-6  Business Edition settings**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language locale</td>
<td>en_US</td>
</tr>
<tr>
<td>Short hostname</td>
<td>m10df51f</td>
</tr>
<tr>
<td>Domain</td>
<td>itso.ral.ibm.com</td>
</tr>
<tr>
<td>WebSphere Application Server home</td>
<td>/usr/WebSphere/AppServer</td>
</tr>
<tr>
<td>Database</td>
<td>1 (DB2 UDB 7.1.0.55)</td>
</tr>
<tr>
<td>Database Install directory</td>
<td>/usr/lpp/db2_07_01</td>
</tr>
<tr>
<td>Database user</td>
<td>db2inst1</td>
</tr>
<tr>
<td>Primary group</td>
<td>db2iadm1</td>
</tr>
<tr>
<td>Web server</td>
<td>1 IBM HTTP Server</td>
</tr>
<tr>
<td>Non-root user</td>
<td>1 (yes)</td>
</tr>
<tr>
<td>Password</td>
<td>wasuser</td>
</tr>
<tr>
<td>User</td>
<td>wasuser</td>
</tr>
<tr>
<td>User group</td>
<td>wasgroup</td>
</tr>
<tr>
<td>Admin port</td>
<td>2222</td>
</tr>
</tbody>
</table>

5. After running the postinstall script, you must update the .profile file for the new wasuser by adding the following lines:

```
PATH=$PATH:/usr/ucb
export PATH
DISPLAY=m10df51f.itso.ral.ibm.com:0:0
export DISPLAY
```

6. Re start the WebSphere Application Server as the new non-root user:

```
# su - wasuser
# export DISPLAY=m10df51f.itso.ral.ibm.com:0.0
# cd /usr/WebSphere/AppServer/bin
# ./startupServer.sh &
```

**Step 5: Create a WebSphere Commerce instance**

This step creates an Business Edition instance, including a database, and deploys the Business Edition Enterprise JavaBeans within the WebSphere Application Server (refer to Chapter 10, “Creating or modifying and instance with
1. Ensure that the WebSphere Application Server and database server are running. Open a command window and run the following commands to start the configuration manager server:

   # su - wasuser
   $ export DISPLAY=M10DF51F.itso.ral.ibm.com:0.0
   $ cd /usr/WebSphere/CommerceServer/bin
   $ ./config_server.sh

   Wait for the following message:
   Registry created.
   CMServer bound in registry

   **Note:** Do not close the terminal window in which you entered the `config_server.sh` command or the Configuration Manager will stop. When you have completed the configuration, you must close the configuration manager server to secure your instance.

2. Open another command window, and launch the Configuration Manager:

   # su - wasuser
   $ export DISPLAY=m10df51f.itso.ral.ibm.com:0.0
   $ cd /usr/WebSphere/CommerceServer/bin
   $ ./config_client.sh

3. When prompted, enter the Configuration Manager user ID (webadmin) and password (webibm).

4. You will be prompted to change the password the first time you log in; we entered webadm1n.

5. To create the instance, do the following in the WebSphere Commerce Configuration Manager:
   a. Expand your hostname.
   b. Right-click **Instance List**.
   c. Select **Create Instance**.

6. Complete the fields in each of the following panels. We used the settings shown in Table 8-7 and accepted all the other defaults.
Table 8-7  Configuration Manager settings

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance name</td>
<td>base</td>
</tr>
<tr>
<td>Instance's root path</td>
<td>/usr/WebSphere/CommerceServer/instances/base</td>
</tr>
<tr>
<td>Merchant Key</td>
<td>0123456789abcdefgh</td>
</tr>
<tr>
<td>Database administrator name</td>
<td>db2inst1</td>
</tr>
<tr>
<td>Database administrator password</td>
<td>db2inst1</td>
</tr>
<tr>
<td>Database Administrator home directory</td>
<td>/home/db2inst1</td>
</tr>
<tr>
<td>Database name</td>
<td>MALL</td>
</tr>
<tr>
<td>Database type</td>
<td>DB2</td>
</tr>
<tr>
<td>Database user name</td>
<td>db2inst1</td>
</tr>
<tr>
<td>Database user password</td>
<td>db2inst1</td>
</tr>
<tr>
<td>Use remote database</td>
<td>enable</td>
</tr>
<tr>
<td>Database server hostname</td>
<td>m10df51f.itso.ral.ibm.com</td>
</tr>
<tr>
<td>Database server port</td>
<td>2222</td>
</tr>
<tr>
<td>Database node name</td>
<td>m10df51f</td>
</tr>
<tr>
<td>Languages</td>
<td>English, German, Spanish, Italian, French</td>
</tr>
</tbody>
</table>

Note: The Configuration Manager creates DB2 tables for the instance, loads the Commerce Server EJBs into WebSphere Application Server and attempt to start the Commerce Server. This process took twenty minutes on our server and may take up to an hour, depending on the configuration of your server.

Verify the instance creation
To verify the instance creation, check the following files in /usr/WebSphere/CommerceServer/instances/base/logs directory:

- base.xml - contains all the configuration information for our instance
- createdb.log - check to see that there are no errors in creating the database
- populatedb.log - contains information about the Commerce database population process
▶ WASConfig.log - check to see there are no errors in installing and configuring the instance
▶ sec_check.log - contains information about potential security exposures
▶ wcs.log - check to see that the server has started successfully

Alternatively, you can open the WebSphere Administrative Console and verify that the Commerce Server has started.

**Step 6: Post-configuration steps**

After creating the instance, we performed the following steps to complete our install.

**Create a security key file for testing**

A security key file is required to create a Business Edition store for testing. The creation of a security key file enables SSL on the Web server. To create a security key file, perform the following steps.

1. Open a command window and stop the IBM HTTP Server:
   ```
   # su - root
   # cd /usr/HTTPServer/bin
   # ./apachctl.sh stop
   ```

2. Export JAVA_HOME by running the following commands:
   ```
   # DISPLAY=m10df51f.itso.ral.ibm.com:0.0
   #export DISPLAY
   # JAVA_HOME=/usr/java130
   # export JAVA_HOME
   ```

3. Open the Key Management Utility by running:
   ```
   # ikeyman
   ```

4. Click the **Key Database File** menu, and select **New**.

5. Enter the file name (keyfile.kdb) and the file location (/usr/HTTPServer/ssl) in the New window.

6. Enter and confirm your IBM HTTP Server password (ibmadmin), and enable the Stash the password to a file option.

7. Click the **Create** menu and select **New Self-signed Certificate**.

8. Fill in all the fields not listed as optional.

9. Start the IBM HTTP Server:
   ```
   # ./apachctl.sh start
   ```
10. Test the key you have created by opening a browser and entering the following URL:

   # https://m10df51f.itso.ral.ibm.com

**Compile the WebSphere Commerce tools JavaServer Pages**

We recommend that you compile your WebSphere Commerce tools JavaServer Pages files before proceeding with creating a store. Compiling the JSPs will significantly reduce the amount of time to load the Store Services pages. To batch compile the tools JSPs, do the following.

Open a command window and run the following commands:

   # su - wasuser
   $ cd /usr/WebSphere/CommerceServer/bin
   $ ./WCSJspBatchCompiler.sh -enterpriseApp "WebSphere Commerce Enterprise Application - base" -webModule "WCS Tools" -nameServerHost m10df51f -nameServerport 2222

**Note:** in our example, we verified the name of our Enterprise Application, “WebSphere Commerce Enterprise Application - base”, by opening the WebSphere Commerce instance from the WebSphere Administrative Console. Compiling the pages took fifteen minutes on our machine. Several errors will be logged, but these can be ignored.

**Step 7: Publish the sample store**

We created a store for our example site using ToolTech, a B2B sample site that ships with Business Edition, for testing and developing. Refer to Chapter 12, “Creating a store from a sample store archive” in the IBM WebSphere Commerce for AIX Installation Guide for use with a DB2 Universal Database Version 5.4 for detailed instructions.

**Creating a store archive**

1. Ensure the following services are running:
   - DB2
   - IBM HTTP Server
   - WebSphere Application Server
   - WebSphere Commerce Server

2. Start Store Services on a Windows machine running MicroSoft Internet Explorer 5.5 by entering the following URL:

   https://m10df51f.itso.ral.ibm.com:8000/storeservices
3. Enter the default Instance Administrator's name (wcsadmin) and the default password (wcsadmin). When we were prompted to change the password, we entered wcsadmin.

4. We entered and selected the following information in the Create Store Archive page:
   - Store archive - base
   - Store directory - base
   - Store owner - Seller Organization
   - Sample store - ToolTech
   - Language - En_De

**Publishing a store archive from Store Services**

To complete the store creation, you must now publish the store archive from Store Services by completing the following steps:

1. From the Store Archive list in Store Services, select the checkbox beside the store you want to publish and click **Publish**. In our example, we selected base.

2. Accept the default publishing options.

3. Select the store from the Store Archive and click **Refresh**.

4. When the status reads **Publish complete**, select the store and click **Publish Summary**.

5. Click the **Launch Store** button to view or test your store.

**Note:** The store publishing task completed in five minutes on our test machine. When you launch the store, it is useful to bookmark the URL so that you can return to the store without using Store Services.

6. Set up a buyer organization, a contract and an account so that you can test the order flow. Refer to IBM WebSphere Commerce Fundamentals Version 5.4 for more information.

**Compile the WebSphere Commerce store's JavaServer Pages**

We recommend that you compile your store JavaServer Pages files before proceeding with testing the store. Compiling the JSPs will significantly reduce the amount of time needed to load the pages. To batch compile the store's JSPs, perform the following steps.
Open a command window and run the following commands:

```
# su - wasuser
$ cd /usr/WebSphere/CommerceServer/bin
$ ./WCSJspBatchCompiler.sh -enterpriseApp "WebSphere Commerce Enterprise Application - base" -webModule "WCS Stores" -nameServerHost m10df51f
-nameServerport 2222
```

**Note:** In our example, we verified the name of our Enterprise Application, “WebSphere Commerce Enterprise Application - base”, by opening the WebSphere Commerce instance from the WebSphere Administrative Console. Compiling the pages took fifteen minutes on our machine. Several errors will be logged, but these can be ignored.

## 8.2.4 Application development

In this section, we discuss guidelines for developing a B2B store using Business Edition. While WebSphere Commerce provides a comprehensive set of integrated components for developing your store, it is also necessary to integrate the site with back-end systems. This process is not trivial and the development process will not only involve configuration and customization of WebSphere Commerce, but also configuration and development of existing and new back-end systems. In some cases, the effort required to integrate with the back-end systems will be equal, if not more than the effort required to develop the WebSphere Commerce site.

The process for developing a B2B site begins with development planning. This includes a thorough gap analysis between the requirements and the functionality that is available in WebSphere Commerce. This exercise should also be repeated for the back-end integration component of the system. The gap analysis will identify development and configuration tasks as well as highlighting risks. Having identified the tasks required for developing the site, a development plan can be created and development roles identified.

The next step is to implement a development environment capable of supporting the development effort. This will include both WebSphere Commerce development and existing back-end development environments. Each build cycle will also require a managed release including deployment of the store archive and source code for a release.

In the following sections, we discuss:

- Development planning
- Development tools
- Development environment
8.2.5 Development planning

The development plan will include the following assets:

- Gap analysis between the requirements and the out of the box functionality of WebSphere Commerce and the existing infrastructure including legacy, messaging and security systems.

- Listing of the total number of JavaServer Pages files, tables, Enterprise JavaBeans, data beans, and commands that need to be developed in WebSphere Commerce. In addition, you will need to scope the effort required to integrate with legacy systems. A listing of the required graphical and media design elements for UI development will also be required. Together, this will drive the size of the project and the required resources. A staffing plan that describes the skills required of the resources is required to design and develop the infrastructure, code, and pages.

- Development schedule listing major checkpoints and deliverable. The schedule must accommodate system testing at the end.

WebSphere Commerce gap analysis

A gap analysis between exiting functionality in Business Edition and the customer requirements is a useful starting point for your development plan. Because we are building a B2B site, we have selected the ToolTech store archive file to build our BASE application on, as specified in Chapter 7, “Detailed design” on page 169. The gap analysis shown in Table 8-8 is only partially complete and would require further development.

Table 8-8   Gap analysis

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Requirement</th>
<th>Business Edition functionality using ToolTech</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalog</td>
<td>Anonymous browse</td>
<td>No - In ToolTech all users must login before accessing the catalog</td>
<td>Yes - Modify home page to allow anonymous users ability to browse the catalog by modifying the ToolTech home page which is the login page for registered users only.</td>
</tr>
<tr>
<td></td>
<td>Catalog search</td>
<td>Yes - ToolTech allows the user to search the catalog based upon a product name or SKU.</td>
<td>Yes - Update URLREG table to allow anonymous users to access the catalog commands.</td>
</tr>
<tr>
<td>Subsystem</td>
<td>Requirement</td>
<td>Business Edition functionality using ToolTech</td>
<td>Implementation</td>
</tr>
<tr>
<td>--------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Filter catalog based upon a contract</td>
<td>Yes - You can use WebSphere Commerce's catalog filtering features to restrict or limit the view of the catalog to a specific set of users. The contents of the filtered catalog is determined by entitlement business policies for contracts. Each contract can decide how to filter the catalog information for the qualifying participants of the contract.</td>
<td>Yes - Update URLREG table to allow anonymous users to access the catalog commands.</td>
</tr>
<tr>
<td></td>
<td>The provided solution will interface with a custom-built external product configurator application for the configuration of complex analytical equipment that is configured on a per use basis.</td>
<td>Yes - While WebSphere Commerce does not supply a product configurator, full support and integration is provided for an external product configurator.</td>
<td>Yes - Although several third party product configurators are available, a custom built configurator will be built by BASE developers.</td>
</tr>
<tr>
<td></td>
<td>BASE is planning to connect their catalog system to one of their suppliers in the future.</td>
<td>Yes - Catalog Manager provides the ability to aggregate information from multiple sources into a consolidated WebSphere Commerce system and to remap all that diverse data into a standard catalog- and product-definition format using XML files as the standard means of managing information.</td>
<td>No - Future requirement only</td>
</tr>
</tbody>
</table>

Chapter 8. Implementation guidelines
<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Requirement</th>
<th>Business Edition functionality using ToolTech</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Membership</td>
<td>New users will be able to self register in order to gain access to the ordering system if the organization they belong to is an existing customer with BASE Pty. Ltd.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>New registration requests will be validated by a user organization representative before individual user access is provided.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>A client organization’s buyer administrator will be validated by a BASE employee before access is provided.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>The solution will make use of the single sign-on capability provided by the existing BASE corporate solution. Customer representatives who have been supplied with user ID and password to access the technical documentation repository will be able to use these credentials to connect to the online shop system.</td>
<td>Yes - WebSphere Commerce supports single sign-on using WebSphere Application Server security</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>The system will use the user identity information from the existing corporate user identity repositories.</td>
<td>Yes - WebSphere Commerce supports user identity information from an LDAP directory</td>
<td>Yes - enable</td>
</tr>
</tbody>
</table>
Because BASE is dealing with customers all across Europe, the site will support multiple languages and multiple currencies. These attributes will be stored in the user's profile. Changes will become effective the next time the user logs on. The first release will only contain English and German language.

Yes - WebSphere Commerce includes several features that allow you to create a site that can be tailored to fit the needs of an international or culturally diverse customer base.

Yes - Create a store archive based on the English/German ToolTech sample.

The solution will provide client organizations with the ability to place an order through the Internet. It will service both
- Requests from users from client organizations wishing to place an online order via Web browser and
- Requests from compatible client organization systems that will interoperate with it to place an online order (using a machine interface).

Yes - WebSphere Commerce supports both browser access and machine user access through procurement system integration

Yes - Configure WebSphere Commerce to communicate with a procurement system

Users must authenticate in order to place an order.

Yes - ToolTech requires users to be authenticated before placing an order.

Yes - Only logged on users will have access to the shopping cart and ordering process.
<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Requirement</th>
<th><strong>Business Edition functionality using ToolTech</strong></th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Users will have the ability to place standing orders. Standing orders are orders with always the same shopping cart that will reoccur over time, following a given schedule.</td>
<td>Yes - Scheduled orders allow customers to submit an existing order as a recurring order. The customer enters shipping and payment information as well as the recurring time scheme, and the scheduler submits the order according to the times specified by the user.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Users will have the ability to create template orders. Template orders are based on previous orders that can be modified to satisfy the current need. They then can be named and saved for further usage.</td>
<td>Yes - A requisition list is a reusable list of items used for repeated order processing. Users can create requisition lists of items that they order frequently and use it to periodically re-order the items. Requisition lists should contain the required information for order processing.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Users will be able to use the bulk order form. Bulk order forms let users who already know what they need to bypass catalog browsing. This is done by a Web page with a list of entry fields where users enter several product stock keeping units (SKUs) and their quantities into at simultaneously. By submitting the bulk order form, all products are put into the user's shopping cart.</td>
<td>Yes - Business Edition supports bulk order forms.</td>
<td>Yes</td>
</tr>
<tr>
<td>Subsystem</td>
<td>Requirement</td>
<td>Business Edition functionality using ToolTech</td>
<td>Implementation</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>---------------------------------------------</td>
<td>-----------------</td>
</tr>
</tbody>
</table>
|           | Order approval functions will be as follows:  
|           | - Approval must be provided by a client organization's buyer approver for orders over a pre-defined monetary value.  
|           | - The buyer approver will have the ability to configure the trigger level for approval of orders for their company.  
|           | - The system will notify the buyer approver as well as the buyer every time an approval is necessary.  
|           | - An order has been rejected by the buyer approver | Partial - The requirement for an order approval is specified in a contract between a buying organization and a selling organization and can be modified by the buyer approver. Order approval terms specify what limits an order must be within before it requires order approval. Some terms apply to all users in an organization, and some apply to certain divisions.  
|           | | In the default order approval flow, if an order requires approval, the order system sends a notification to an approver, but not to the buyer. If an order is rejected, the order system sends a notification to the buyer, but not the buyer approver. If an order is approved, the order system will process the order and if the order status message is enabled, will send an order received message to the buyer. An order approved message is not sent to either the buyer approver or the buyer.  
|           | | Yes - Add the following notifications to the messaging systems:  
|           | | - a message notifying buyer that an order requires approval  
|           | | - a message notifying the buyer approver that they have approved an order  
|           | | - a message notifying the buyer that an order has been approved  
|           | | - a message notifying the buyer approver that they have rejected an order  
|           | | Note: These additional notification requirements may be deprecated if the default order approval flow is deemed to meet the requirements sufficiently. Or, they may not be implemented if they are low priority and scoped out of the project due to either financial, or time constraints.  
<p>| | | |
|           | | |</p>
<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Requirement</th>
<th>Business Edition functionality using ToolTech</th>
<th>Implementation</th>
</tr>
</thead>
</table>
| Order status functions will be as follows:  
- Users will be able to view the status of current and past orders. This also includes orders placed through other channels.  
- Authorization control will provide the ability for a standard buyer to see all orders placed by that user.  
- Buyer approvers will have the ability to access all orders placed by their organization, regardless of the channel that was used for placing the order. | Partial - the order subsystem allows buyers and buyer approvers to view current orders only. | Yes - the order subsystem will be extended to include all current and historical orders from the fulfillment system via a batch update using the messaging subsystem. |
| Procurement system integration features will be as follows:  
- The functionality offered by the procurement system interface will be limited to placing an order and receiving the status of an order.  
- The following message will be delivered to the procurement system in industry standard XML format:  
  - Purchase Order  
  - Order status update | Yes - WebSphere Commerce includes functionality for configuring procurement system integration using the program adapter and order status updates through the messaging system. | Yes - Configure WebSphere Commerce for procurement system integration and enable order status notification using XML. |
<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Requirement</th>
<th>Business Edition functionality using ToolTech</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Orders are paid by invoice. There will be no online payment facility.</td>
<td>Yes - A business account can record whether the invoice should be shipped by e-mail, regular mail or included with a shipment. If the business account is configured to have invoices sent by e-mail, the invoice will be sent automatically when the pick batch for the release is generated.</td>
<td>Yes - create business accounts for buyer organizations and enable payment by invoice.</td>
</tr>
</tbody>
</table>
| Legacy systems integration | Pricing information will be retrieved from the pricing system within the BASE organization.  
Product information and product images will be retrieved from the product management system within the BASE organization.  
Inventory information will be retrieved from the inventory system within the BASE organization.  
Fulfillment capability and order status information will be retrieved from the fulfillment system within the BASE organization. | Yes - The messaging subsystem has messages for populating the catalog from the product and pricing systems. It also has messages for populating inventory information and order status information from backend systems.  
The catalog loader can be used for populating product information during the initial load of catalog data. | Yes - enable the inbound messaging subsystem and the following back end integration inbound messages:  
Pricing information - inbound pricing messages  
Product information - Add a new back end interrogation XML message to update product information.  
Inventory information - Inbound inventory messages  
Fulfillment - Inbound fulfillment messages  
Order Status - Inbound order status messages |
<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Requirement</th>
<th>Business Edition functionality using ToolTech</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Care</td>
<td>Users will be able to interoperat with BASE Customer Service Representatives (CSRs) via text-based chat for support and trouble shooting.</td>
<td>Yes - WebSphere Commerce supports e-commerce collaboration functionality including customer text-based customer care support using the Lotus Sametime server and Sametime applet.</td>
<td>Yes - Enable Sametime collaboration for the WCS instance by using the WebSphere Commerce Configuration Manager and Store Services. Sametime uses normally port 1533. This port is disabled in our customers firewall. But the sametime admin console has a feature where the main ports from sametime can be tunneled via port 80. So we have just to configure the port 80 tunnelling tool of the Sametime server properly.</td>
</tr>
<tr>
<td></td>
<td>CSRs will be able to view user interaction with the system to assist users.</td>
<td>Yes - WebSphere Commerce supports CRS viewing user interaction using the Lotus Sametime server.</td>
<td>Yes - Enable Sametime collaboration for the WCS instance by using the WebSphere Commerce Configuration Manager and Store Services.</td>
</tr>
<tr>
<td></td>
<td>Only registered users will be able to use the customer care facility.</td>
<td>Partial - in ToolTech an anonymous user can access the customer care feature.</td>
<td>Yes - hide customer care link from anonymous users using JSPs.</td>
</tr>
<tr>
<td></td>
<td>CSR viewing of user interaction will only be granted on a per occurrence permission from the user.</td>
<td>Yes - WebSphere Commerce supports per occurrence permission from the user.</td>
<td>Yes - this is defined for the instance in the collaboration section of the configuration manager.</td>
</tr>
<tr>
<td></td>
<td>CSR will have the ability to perform all interactions on behalf of users.</td>
<td>Yes - This is a feature of Commerce Accelerator under the Customer Care feature.</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Chapter 8. Implementation guidelines

Having gone through the gap analysis, we can create a list of assets that we will need to modify, extend or customize. This list should include the total number of JavaServer Pages files, tables, Enterprise JavaBeans, data beans, and commands that need to be developed in WebSphere Commerce. In addition, you will need to scope the effort required to integrate with legacy systems. Together, this will drive the size of the project and the required resources.

Included in this list will be the type of resource or development role that will need to be assigned to complete the task. For example, modifying the ToolTech login and browser will require a store developer with JSP and HTML skills. This developer will not need to have extensive Java experience, unlike the developer who is assigned to extend the order history functionality who will need to work with WebSphere Commerce commands and EJBs.

At this stage, we can also make an estimate of the effort for each task and the associated risk. In this case, each development tasks includes unit testing by the developer. We calculated effort for each task using the following guidelines:

- Simple - we know how to do it and have done it before: 5 days
- Medium - we know how to do it but haven’t done it before: 10 days
- Complex -we don’t know how to do it and haven’t done it before: 20 days

An example of our estimates for BASE systems is shown in Table 8-9.

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Requirement</th>
<th>Business Edition functionality using ToolTech</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>The system will produce periodic marketing reports. These reports will provide information allowing targeted advertising based on customer usage and other customer activity.</td>
<td>Yes - WebSphere Commerce provides marketing reports functionality using WebSphere Commerce Analyzer.</td>
<td>Yes - Install and configure WebSphere Commerce Analyzer.</td>
</tr>
<tr>
<td></td>
<td>The frequency of report generation will be configurable.</td>
<td>Yes - WebSphere Commerce Analyzer reporting is fully configurable.</td>
<td>Yes - configure WebSphere Commerce Analyzer for scheduled reports.</td>
</tr>
</tbody>
</table>
Table 8-9  WebSphere Commerce asset list.

<table>
<thead>
<tr>
<th>Name</th>
<th>Task</th>
<th>Resource</th>
<th>Complexity</th>
<th>Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSP: LogonForm.jsp</td>
<td>Modify login flow and catalog browse to allow guest users to browse the catalog without logging in.</td>
<td>Store Developer</td>
<td>Low</td>
<td>5</td>
</tr>
<tr>
<td>CatalogMainDisplay.jsp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TrackOrderStatus page.jsp</td>
<td>Customize order status display to include order history from all products ordered from BASE including those created in legacy systems offline. Use the customizable fields in the ORDER table to identify offline orders.</td>
<td>Store Developer</td>
<td>Complex</td>
<td>20</td>
</tr>
<tr>
<td>OrderDetail command</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OrderAccessBean ORDERS Table</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A similar exercise is required for legacy development tasks and UI assets such as graphics and other media.

**Staffing plan**

The staffing plan is derived from the development asset list. This list may also include resources who are available and can be assigned to the role. This plan can then be used to allocate resources to the development schedule. See Table 8-10.

Table 8-10  Staffing plan

<table>
<thead>
<tr>
<th>Role</th>
<th>Skills</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architect</td>
<td>▶  WebSphere Commerce</td>
</tr>
<tr>
<td></td>
<td>▶  WebSphere MQ</td>
</tr>
<tr>
<td></td>
<td>▶  Integration</td>
</tr>
<tr>
<td></td>
<td>▶  Security</td>
</tr>
<tr>
<td></td>
<td>▶  Single sign-on</td>
</tr>
<tr>
<td>Role</td>
<td>Skills</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Web Designer</td>
<td>▶ Usability</td>
</tr>
<tr>
<td></td>
<td>▶ Graphics</td>
</tr>
<tr>
<td>Store Developer</td>
<td>▶ Java</td>
</tr>
<tr>
<td></td>
<td>▶ Knowledge of the EJB component model</td>
</tr>
<tr>
<td></td>
<td>▶ VisualAge for Java</td>
</tr>
<tr>
<td></td>
<td>▶ JavaScript</td>
</tr>
<tr>
<td></td>
<td>▶ HTML</td>
</tr>
<tr>
<td></td>
<td>▶ JSP technology</td>
</tr>
<tr>
<td></td>
<td>▶ Familiarity with the WebSphere Commerce programming model.</td>
</tr>
<tr>
<td>Database Developer</td>
<td>▶ Db2</td>
</tr>
<tr>
<td></td>
<td>▶ XML</td>
</tr>
<tr>
<td></td>
<td>▶ Familiarity with BASE Legacy systems</td>
</tr>
<tr>
<td>Site Administrator</td>
<td>▶ WebSphere Commerce Installation on AIX and Windows 2000</td>
</tr>
<tr>
<td></td>
<td>▶ WebSphere Commerce configuration including single sign-on and procurement integration</td>
</tr>
<tr>
<td></td>
<td>▶ WebSphere Commerce performance tuning</td>
</tr>
<tr>
<td></td>
<td>▶ WebSphere Commerce access and authentication</td>
</tr>
<tr>
<td></td>
<td>▶ System backups</td>
</tr>
<tr>
<td>Store Administrator</td>
<td>▶ Publish store archives</td>
</tr>
<tr>
<td></td>
<td>▶ BASE business procedures</td>
</tr>
<tr>
<td></td>
<td>▶ Web literate</td>
</tr>
<tr>
<td>C++ Programmer</td>
<td>▶ C++</td>
</tr>
<tr>
<td></td>
<td>▶ WebSphere MQ programming</td>
</tr>
<tr>
<td>Integration Infrastructure and Support specialist.</td>
<td>▶ WebSphere MQ configuration</td>
</tr>
<tr>
<td></td>
<td>▶ Single sign-on configuration</td>
</tr>
</tbody>
</table>

**Note:** The store development process will also require input from the Marketing Manager, Product Manager and Seller. Refer to the IBM *WebSphere Commerce Fundamentals Version 5.4* for more information.

**Development schedule**

The development schedule will include a list of tasks and major checkpoints and deliverables. This schedule will also include dependencies and resources assigned to each task. A project planning tool or spreadsheet is useful for
creating the schedule. The schedule must accommodate system testing at the end.

8.2.6 Development tools

To help developers customize and develop new functionality, IBM has created WebSphere Commerce Studio. Commerce Studio ships in two packages, Developer Edition and Professional Developer Edition. A developer who is creating HTML, JSPs, commands and data beans will require the Developer Edition. For those developers creating EJBs, the Professional Edition will be required, which includes VisualAge for Java, Enterprise Edition. VisualAge for Java, Enterprise Edition also includes the WebSphere Test Environment which can be used to test EJBs.

In addition to the WebSphere Commerce Studio tools, you will also require Store Services for creating a store archive based on ToolTech, and a graphics tool such as PaintShop Pro. In addition, we recommend the use of a source control tool such as Rational ClearCase for source code control. If a design tool such as Rational Rose is used for the architecture and design, this can be integrated with VisualAge for Java.

The development tools required for the BASE site included:

- WebSphere Commerce Studio Developer Edition for developing HTML, JSPs, commands and data beans
- WebSphere Commerce Studio Professional Edition for developing HTML, JSPs commands and data beans
  - VisualAge for Java, Enterprise Edition for building EJBs
- Store Services for creating store archives
- Paintshop Pro for developing graphics and images
- Rational Rose
- Rational ClearCase

Note: For a more detailed discussion on the WebSphere Commerce development tools, refer to Chapter 8, “Application Development guidelines” in the redbook e-commerce Patterns for Building B2C Web Sites Using IBM WebSphere Commerce Suite V5.1, SG24-6180.
8.2.7 Development environment

In this section, we discuss strategies for the development environment. In our case, the development environment includes all the tools discussed above and a development server, source code server and test systems.

This section is organized into the following topics:

- Team development strategies
- Development environment installation
- Deployment
- Unit testing

Team development strategies

WebSphere Commerce Studio development requires a team of developers who have different tool and source requirements. The control of source code in a mixed development environment can be addressed for source control and deployment. The ability to version a release means allows for iterative development and release to the testing team.

There are two variations of development strategies that are described in detail in Chapter 8, “Application Development guidelines” of the redbook *e-commerce Patterns for Building B2C Web Sites Using IBM WebSphere Commerce Suite V5.1*, SG24-6180.

Full WebSphere Commerce development environment

A full Business Edition development environment is one in which every developer has a standalone version of WebSphere Commerce Studio and WebSphere Commerce Studio Professional Developer. This is most suitable for developments which require extensive customization of business logic, including EJBs. The following is a list of the elements of the full Business Edition development environment:

- Local WebSphere Commerce Studio instance per development environment
- Shared (or local) WebSphere Commerce Studio Database
- Shared (or local) VisualAge for Java repository
- Shared (or local) WebSphere Studio directory
- Source Control Management tool

The advantage of this environment is that sharing repositories for code reduces the load on each machine and allows code to be versioned and managed. Developers are also able to test their code in a fully functioning environment and can reboot without affecting other developers. The drawback with this method is
the expense of outfitting every developer with WebSphere Commerce Studio. Workstations with this configuration will require at least 1GB of memory. See Figure 8-2 for an example of the development environment.

**Light Business Edition development environment**

As we have mentioned, having WebSphere Commerce Studio and WebSphere Commerce server on the same machine is very resource intensive. Also, not every developer requires all the functionality in the full development environment. The main features of this configuration are:

- Shared WebSphere Commerce server
- Shared WebSphere Commerce Studio database
- Shared VisualAge for Java repository
- Shared WebSphere Studioo directory
- Source Control Management tool

The disadvantage with this environment is that all the workstations are reliant on shared applications and repositories. In the event of a reboot, all workstations are affected. This is most suitable for small development teams with limited customization goals. See Figure 8-3 for a view of the Lightweight development environment.
For our BASE example, we would use a mixed environment with only the EJB developers having the full WebSphere Commerce Studio environment.

![Lightweight WebSphere Commerce development environment](image)

**Figure 8-3** Lightweight WebSphere Commerce development environment

### 8.3 Testing guidelines

This section will discuss the importance of testing an e-commerce application and runtime. In addition, we provide basic guidelines and tools to perform different types of tests required prior to deploying an e-business Web site.

With a B2B site, a managed testing strategy is required to ensure quality of the delivered application. This requirement is even more critical considering the legacy and security systems integration of our example. Refer to Chapter 8, “Testing Guidelines” in *WebSphere Commerce V5.4 Developer’s Handbook*, SG24-6190 for more information.

In this section, we highlight the following guidelines:

- Test phase and categories
- Test environments
- Different types of tests for e-business
8.3.1 Test phase and categories

Within a e-business development project, there are various categories of testing performed within different phases of the development lifecycle. In this section, we define the roles, input and output from the different cycles. These include the following.

Roles
The typical roles needed are as follows:

- Test architect
- Test specialist - business
- Test specialist - technical
- Tester
- Test environment specialist

Input
The input of this phase is as follows:

- Test design
- Test plans
- Test environment

Output
The output from this phase is as follows:

- List of defects
- Document design improvements for next release.

Test plan
A detailed test plan is required for each phase of the development lifecycle. Testing should run in parallel with development so that errors can be picked up early in the project. The test architect needs to work closely with the solution architect, business analysts, subject matter experts and business representatives from the beginning of the project to ensure a quality product. Each phase of the development will have its own test plan, as part of an overall test plan and strategy.
Test phases
The key test phases required for testing an e-business site are as follows:

- **Unit test**
  Unit tests are performed by developers before releasing the code for a build and functional testing. Tests are conducted on the developer machine using the WebSphere Test Environment that is part of the VisualAge for Java Enterprise Edition. The WebSphere Test Environment can be used to test data beans, Enterprise JavaBeans, Java Servlets and JavaServer Pages.

- **Build verification test**
  After unit testing developers release their source code into a source control tool, such as MicroSoft’s Visual Source Safe. The build team creates a release from the source code marked for release by the developers. A build verification test is performed to ensure that the release is viable to make it available to the test team for functional or system testing.

- **Functional verification test**
  Functional testing is performed by the test team on a dedicated environment. The test team tests individual Business Edition functions of the site on a dedicated Business Edition instance. Testing the login for registered buyers is an example of a function test.

- **System verification test**
  The system verification test is performed toward the end of the development lifecycle when end-to-end functionality has been developed. For example, this could be testing the ordering flow of the site. System testing is performed by the test team in a dedicated system testing environment configured to match the production environment.

- **Integration verification test**
  A typical B2B site must integrate with both internal and external systems such as legacy inventory data stores and external procurement systems. The test team, therefore, needs to perform integration tests with these other systems. The system test environment can be used for integration testing.

- **Stress test**
  Stress testing simulates the volume of traffic expected on the B2B site. This is often performed by a third-party application using the system testing environment. Business Edition ships with Segue Silk Preview for stress testing.

- **Customer acceptance test**
  Customer testing is usually performed on a staging server at the end of the development lifecycle. All aspects of the system are tested, including against
business and functional requirements. The customer acceptance test is usually conducted as part of the sign-off phase of the project.

8.3.2 Test environments

Each phase of the development project requires a different test environment. Some of the common environments are as follows.

► Unit test environment

Typically, this is a developer’s workstation, but on bigger projects a dedicated machine may be useful for unit testing and integration before release to the build team.

► Build test environment

The build test environment is one in which the build team verifies a release prior to making it available to the test team. This environment should resemble the production environment, but may not have exactly the same configuration.

► System test environment

The system test environment should be a replica of the production environment, including links to back-end and external systems. The system test environment may also be used for functional verification testing, system verification testing and integration verification testing.

► Staging environment

The staging environment is used for customer acceptance testing and updates to the catalog, as well as to simulate the production system.
This chapter discusses the technical approach to procurement integration using WebSphere Commerce Business Edition. Business Edition provides a program adapter that enables XML messages over HTTP to collect orders from external procurement systems. This chapter describes the Business Edition procurement integration architecture and details the program adapter for enabling XML messages over HTTP. We configure and implement the procurement integration reference application. Finally, we discuss the changes that need to be made to our sample site through a worked procurement integration example.

This chapter covers the following:

- Procurement integration architecture
- Configuring procurement integration
- Procurement integration customizations
- Procurement integration example

**Note:** Interaction between the procurement system and WebSphere Commerce Business Edition is covered by a Self-Service business pattern, as discussed in Section 4.4.1, “The Self-Service business pattern” on page 61. The procurement system end users work from a Web browser and whether they are using punchout or a local catalog to interact with Business Edition, they are doing Self-Service. See Figure 9-2 on page 247, Figure 9-3 on page 249, and Figure 9-5 on page 261; these will make it clear that the Self-Service is being used.
9.1 Introduction

Just as ecosystems are interrelated, very few e-business systems exist in isolation in today’s commercial world. It would therefore be a mistake to think that Sell-Side Hubs exist in isolation and will only be accessed by users with browsers. The primary business driver for creating a Sell-Side Hub may be the requirement to integrate with a major customer’s Buy-Side Hub procurement system.

Procurement integration provides the following benefits:

- Suppliers can maintain a single catalog within Business Edition and use the catalog to enable their Web presence and participate in the buyer's procurement network.
- Costs of order processing through Business Edition connectivity to supply change management, retail management, and order management systems are reduced.
- The B2B features of WebSphere Commerce Business Edition can be used to create and maintain buyer organizations, buyer-specific catalogs, price lists, and contract pricing.

This chapter discusses the technical approach to procurement integration using Business Edition. Business Edition provides a program adapter that enables XML messages over HTTP to collect orders from external procurement systems. This chapter describes the Business Edition procurement integration architecture and details the program adapter for enabling XML messages over HTTP. We also configure and implement the procurement integration reference application. Finally, we discuss the changes that need to be made to our sample site through a worked procurement integration example.

9.2 Procurement integration architecture

Business Edition provides the functionality to integrate with a customer's procurement system. The business driver for integration may be the request of a major buyer who owns the procurement system, or simply an attempt by the seller to extend their Web presence. Business Edition’s procurement system integration architecture is both extendable and customizable. Developers are able to add messages and extend or customize the procurement related business logic. In this section, we will describe two modes of procurement integration and communication supported by Business Edition. We will also discuss Business Edition’s program adapter support for XML over HTTP. Finally, we will describe procurement access control and sub-system extensions.
9.2.1 Procurement integration modes

Business Edition supports two modes of integration with procurement systems.

- Local catalog mode
- Punch-out catalog mode

**Local catalog mode**
In local catalog mode, the user does not have to connect or authenticate with the supplier’s Business Edition site to browse the catalog. Instead, the supplier’s catalog is replicated on the buyer’s procurement system. The user can then complete the shopping process and place an order with the supplier's Business Edition site.

**Punch-out catalog mode**
In punch-out catalog mode, a single catalog is maintained within the supplier’s Business Edition site. When a requisitioner from a procurement system wishes to place an order with the supplier system, they authenticate and order directly with the supplier’s Business Edition.

**Communication types**
Business Edition also supports two types of communication protocols used by procurement systems, for customers operating in either local catalog or punch-out catalog mode:

- One-step mode
- Two-step mode

The one-step mode is utilized by procurement systems that use browser URL requests for setup, authentication and completion of the order process. mySAP’s OCI protocol is an example of this protocol.

The two-step mode is used by the procurement system’s XML connector to support sending XML messages over HTTP. XML messages are sent to Business Edition for setup, authentication, and registration of the user with the buyer’s organization. When the message has been received by Business Edition, a browser is launched on the customer’s desktop to complete the ordering process. Commerce One’s XML Commerce Connection is an example of this protocol.
9.2.2 Business Edition procurement integration components

To understand how Business Edition integrates with procurement systems, it is useful to look at the how the procurement components fit into Business Edition's run-time architecture.

Note: For more details on the Business Edition run-time architecture and programming model, refer to the overview and programming module chapters in IBM WebSphere Commerce Programmer's Guide Version 5.4.

The major components of the Business Edition runtime architecture are:

- Servlet engine
- Protocol listeners
- Adapter manager
- Adapters
- Web controller
- Commands
- Enterprise JavaBeans
- Data beans
- Data bean manager
- Display files
- XML files

Components related to procurement system integration are illustrated in Figure 9-1. More details on each component and how it relates to procurement system integration can be found in the following sections.
Servlet engine
The Servlet engine acts as the Business Edition request dispatcher. It manages a pool of threads to handle URL requests. Each URL request from an external procurement system is assigned a separate thread.

HTTP request servlet
Business Edition commands can be invoked from various devices, including XML messages from external procurements systems over HTTP. When the request servlet is initialized, it reads the instance_name.xml to determine which adapter to pass requests to. When the HTTP request servlet receives a URL request from the servlet engine, it passes the request to the adapter manager.

Adapter manager
The adapter manager assigns the procurement system request to the program adapter so that it can process that request.

Program adapter
Business Edition adapters are device-specific components that perform processing functions before passing a request to the Web controller. The program adapter allows procurement systems to communicate with Business Edition by passing XML requests over HTTP. Each request is treated as a separate session. The program adapter verifies the user ID and password parameters from the control area of the message to determine the credentials of a request.
The program adapter performs the following actions for incoming XML requests.

1. Recognizes the XML request by verifying the following three attributes of the request:
   - Content type
   - Method
   - Character encoding

2. Extracts the input stream of the request.

3. Calls the message mapper.

4. Receives the CommandProperty object representing a Business Edition command returned by the message mapper.

5. Determines the proper device format to generate the response.

6. Executes the command.

7. Sends an XML response message created by the appropriate JSP specified by the command executed.

By default, the program adapter is not enabled when a Business Edition instance is created. The program adapter must be configured in the instance_name.xml file. Once the instance is restarted, the program adapter it resides in is enabled as long as the instance runs.

**XML message mapper**

The XML message mapper is a mechanism for mapping an XML message to a CommandProperty object. It does this by converting the XML data from inbound XML messages to CommandProperty objects. Example 9-1 shows a configuration for an XML message mapper found in the instance_name.xml configuration file:

```
Example 9-1   XML message mapper configuration
<MessageMapper
classname="com.ibm.commerce.messaging.programadapter.messagemapper.ecsax.ECSAXMessageMapper"
enable="true"
messageMapperId="-1"
name="WCS.INTEGRATION">
<configuration />
</MessageMapper>
```
**CommandProperty object**
The CommandProperty object is a representative of a controller command identified by the XML message mapper. The object contains the properties and parameters for a given command to be executed by the program adapter.

The CommandProperty data type is composed of the following three parts:
- **commandName**: string name of the command to be executed
- **requestProperties**: command properties when executing the command
- **executionProperties**: control data for executing the command

**Web controller**
The Business Edition Web controller provides session management services for procurement commands. Session persistence is established by the program adapter. Like the EJB container, the Web controller manages transactions for commands and will roll back transactions if there are any exceptions.

**Commands**
Business Edition commands are beans that contain the programming logic associated with handling particular requests. The key commands and messages associated with procurement system integration are as follows.

- **PunchOutSetup** is a controller command used when the PunchOutSetup message is received from a requisitioning user's procurement system. It authenticates the buyer organization and the user. The command registers the user as a new user in the member subsystem, if they are not already registered. In two-step mode, the PunchOutSetup command is called when mapping the incoming XML message. For one-step mode, it is called directly from the user's browser.

  The PunchOutSetup command calls the following task commands:
  - **AuthenticationHelper**
  - **DBProcurementAuthentication**
  - **RegisterRequistioner**

- **BatchOrderRequest** is a controller command used when a PurchaseOrderRequest message is received from a procurement system. The OrderResponse message is sent back to the procurement system upon execution of the task.

  The BatchOrderRequest command calls the following task commands:
  - **CheckBatchOrderRequest**
  - **AuthenticationHelper**
– RegisterRequistioner
– CreateBillingShippingAddress
– ProcessOrder

It also calls the following controller commands:

– ProcurementOrderPrepare.

▶ PrepareProcurementOrder extends the PrepareOrder task command. It is called from the BatchorderRequest command to prepare the new order that is created when processing the PurchaseOrderRequest message.

▶ The SubmitShoppingCart task command is invoked when the procurement buyer submits an order. This command changes the order status from pending to awaiting approval. Changes to the ordering flow can be made by altering this command in conjunction with the SendShoppingCart command.

▶ The SendShoppingCart task command prepares the quote to be sent back to the procurement system as a message. It includes order and item details.

**Instance_name.XML configuration file**

The instance_name.xml configuration file sets configuration information for the instance. It is read when the request servlet is initialized.

### 9.2.3 Procurement integration ordering flows

Now that we have described the two types of procurement integration modes and the procurement components, we can examine the ordering flows for both modes of integration based on the Business Edition runtime architecture.

**Local catalog mode ordering flow**

Figure 9-2 illustrates the interaction between components to enable a buyer from a procurement system to place an order with Business Edition using the local catalog mode.
The following describes the interactions shown in the above diagram:

1. The buyer browses the catalog from the procurement system and creates an order.
2. The OrderRequest message is sent by the procurement system to Business Edition. The OrderRequest message contains the buyer and supplier user IDs, order, shipping and billing information.
3. The OrderRequest message is received by the servlet engine and assigned a thread, then passes the request to the HTTP request servlet.
4. The HTTP request servlet passes the request to the adapter manager.
5. The adapter manager assigns the request to the program adapter to process the request.
6. When the program adapter receives the OrderRequest message, it performs the following actions.

Figure 9-2  Local catalog mode
a. Verifies it is an XML request as specified in the adapter configuration and extracts the input stream from the request. The adapter then calls the message mapper and passes the content of the input stream.

b. Receives the CommandProperty object representing the **BatchOrderRequest** command returned by the message mapper and determines the device format in which to generate the response.

7. The Web controller executes the **BatchOrderRequest** command using Business Edition’s command order framework and persistent objects.

8. The Web controller passes the response from the command order framework and persistent objects back to the program adapter.

9. The program adapter sends the **OrderResponse** message created by a JSP to the procurement system.

**Punch-out catalog mode ordering flow**

Figure 9-3 illustrates the interaction between components to enable a buyer from a procurement system to place an order with Business Edition using the punch-out catalog mode.
The following describes the interactions shown in the above punch-out catalog mode diagram.

1. The buyer selects the supplier on the procurement system.
2. The procurement system sends the PunchOutSetupRequest message to Business Edition.
3. The PunchOutSetupRequest message is received by the servlet engine and assigned a thread, then passes the request to the HTTP request servlet.
4. The HTTP request servlet passes the request to the adapter manager.
5. The adapter manager assigns the request to the program adapter to process the request.
6. When the program adapter receives the PunchOutOrderRequest message, it performs the following actions:
   a. Verifies it is an XML request as specified in the adapter configuration and extracts the input stream from the request. The adapter then calls the message mapper and passes the content of the input stream.
   b. Receives the CommandProperty object representing the PunchOutSetup command returned by the message mapper and determines the device format in which to generate the response.


8. The Web controller passes the response from the command order framework and persistent objects back to the program adapter.

9. The program adapter sends the PunchOutSetupResponse message to send the CatalogDisplay URL and information needed to bind the session back to the procurement system.

10. The procurement system launches a new browser window using the PunchOutCatalogDisplay command and the catalog is displayed to the buyer to begin browsing and ordering.

11. The buyer browses or searches the catalog and submits an order to the procurement system for approval.

12. The purchase order is approved and an order request message is sent to Business Edition for fulfillment.

### 9.2.4 Access control for procurement systems

In Business Edition, access control policies are used to determine what roles can have access to which parts of the system. For each buyer organization procurement system, there is a role for requisitioning users and for those with the authority to register users on the fly.

The two access control policies provided for procurement system users are described in the following sections.

#### Procurement buyer

The procurement buyer role is assigned to the requisitioning user of the procurement system. Procurement buyers are registered when a request is sent from the procurement system. Buyers use the account set up by the procurement systems organization in Business Edition. When the order is placed by the buyer using the SendShoppingCart task command, an order approval message is returned to the procurement system.
**Procurement buyer administrator**

The procurement buyer administrator access control policy is used to register procurement buyers. The role is also used for approving procurement buyers and administering sub-organizations within the buyer's organization. The `RegisterRequistioner` retrieves the requisitioning user if they are already registered, and registers them if they are not registered. The `RegisterRequistioner` command is called by the `PunchOutSetup` and `BatchOrderRequest` commands. The `RegisterRequistioner` command verifies that the user has the procurement buyer administrator role to register users. Each buyer organization procurement system registered with Business Edition must have a procurement buyer administrator.

**Procurement user authentication and registration**

Business Edition provides support for registering procurement system buyers on the fly using the following protocols:

- XML over HTTP
- HTTP over a browser

During registration, the user’s procurement system user ID is used to create a unique login ID for integration with Business Edition’s Member system.

Figure 9-4 illustrates the interaction between components to enable procurement system user authentication and registration on the fly.

---

**Figure 9-4**  Procurement system authentication and registration
The following describes the interactions shown in the above procurement system authentication and registration diagram.

1. A punch out catalog setup request is sent from the procurement system to Business Edition.
2. The program manager verifies the XML message and calls the XML message mapper.
3. The XML message mapper parses the credentials to the Business Edition logonID and logonPassword parameters and maps the message to a command object.
4. The program adapter calls the AuthenticationHelper task command.
5. The AuthenticationHelper task command authenticates the buyer and supplier credentials against the ORGCODE table.
6. The AuthenticationHelper task command authenticates the buyer and supplier relationship against the BUYSUPMAP table.
7. After the administrator has been authenticated, access control policies are used to determine that the user has the authority to register procurement buyer.
8. If successful, the procurement buyer is registered with Business Edition as a new user. Procurement buyers are identified by the ORG_ID and REQUISTIONER_ID fields in the BUSPROF table.

9.3 Procurement integration configuration

This section describes the necessary tasks to configure Business Edition to communicate with a procurement system using XML over HTTP.

9.3.1 Configuring procurement protocol and system information

To configure Business Edition for procurement integration, you must capture the procurement system’s protocol and system information. Typically, a buyer will invite the supplier to integrate with the buyer’s procurement system. Once invited, you must obtain specific protocol and system information about your buyer’s procurement system. One way of doing this is by capturing the information in a paper-based or electronic form that is then entered into the Business Edition configuration files.

Information that must be captured about the buyer organization includes:

- Buyer organization name
- Organization code (test)
Configuring procurement protocol and system information

To configure the procurement system protocol and system information, do the following.

1. Add the name of the procurement system to the PROCSYS database and the protocol name, version, communication type, and authentication type in the PROCPROCTL database table by inserting the following SQL statement. For our example, we used the Ariba cXML protocol version 1.0.

   insert into procsys values ('Ariba', null, null);
   insert into procprotcl (procprotcl_id, procsysname, protocolname, version, authtype, twostepmode, classifdomain, uomstandard) values (1, 'Ariba', 'cXML', '1.0', 1, 'Y', UNPSC, NULL);

2. Add a new message mapper for the protocol.
   a. Open the instance_name.XML file and add the following configuration entry:

      <MessageMapper
classname="com.ibm.commerce.messaging.programadapter.messagemapper.ecsax .ECSAXMessageMapper"
enable="true"
messageMapperId="-3"
name="WCBE.ARIBA">
<configuration
EcSystemTemplateFile="ariba_sys_template.xml"
EcTemplatePath="C:\WebSphere\CommerceServer\XML\ariba"
EcInboundMessageDtdFiles="cXML.dtd"
EcInboundMessageDtdPath="C:\ariba"
/>
</MessageMapper>

Note: The cXML 1.0 DTD file is available at http://www.cxml.org/files/downloads.cfm. All other versions are also available from this address.
3. Create a mapping file for the procurement protocol messages. Save the file in the directory specified by the EcTemplatePath in the configuration section of the instance_name.xml file. In our example, the EcTemplatePath is C:\WebSphere\CommerceServer\XML\ariba which is where we downloaded the 1.0 cXML DTD file.

4. Update the configuration for the XML over HTTP adapter by adding the new message mapper. In our example we added supportedMessageMappers="WCS.ARIBA" to our instance_name.xml file as follows within the HTTPAdapters section:

```xml
<HttpAdapters display="false">
  <HttpAdapter deviceFormatId="-10000"
    deviceFormatType="XmlHttp"
    deviceFormatTypeId="-10000"
    enabled="true"
    factoryClassname="com.ibm.commerce.programadapter.HttpProgramAdapterImpl"
    name="XML/HTTP">
    <SessionContext
      class="com.ibm.commerce.messaging.programadapter.security.CredentialsSpecifiedProgramAdapterSessionContextImpl">
      <SessionContextConfig />
    </SessionContext>
    <Configuration supportedCharacterEncoding="ISO8859-1, UTF-8"
      supportedContentTypes="text/xml, text/xml; charset=UTF-8, text/xml-SOAP"
      supportedMessageMappers="WCS.INTEGRATION, WCS.ARIBA"
      supportedMethods="POST, M-POST" />
  </HttpAdapter>
</HttpAdapters>
```

5. Add an entry in the DEVICEFMT table for the new message mapper by inserting the following SQL statement. The DEVICEFMT_ID is created by adding the HTTPadapter deviceFormatID and the messageMapperID. In our example, the DEVICEFMT_ID we used is -10003.

```sql
insert into devicefmt (devicetype_id, devicefmt_id, displayname) values ('ARIBAXMLHTTP', -10003, 'XML message received via Ariba');
```
Enabling the program adapter for XML requests over HTTP

By default, when the Business Edition instance is created, it is disabled. In order to support XML over HTTP, the program adapter must be enabled. To enable the program adapter, open the instance_name.xml file and set the enabled parameter to true.

```xml
<HttpAdapters display="false">
  <HttpAdapter deviceFormatId="-10000"
    deviceFormatType="XmlHttp"
    deviceFormatTypeId="-10000"
    enabled="true"
```

Configuring a buyer organization

Each requisitioning buyer using a procurement system that integrates with Business Edition must belong to a registered buyer organization. To configure a buyer organization that uses a procurement system, do the following.

1. Create a buyer organization from the Administration Console.
2. Assign the procurement buyer administrator and procurement buyer roles for the buyer organization.
3. Add the organization identity information in the ORGCODE table for both the buyer and the supplier. In our example for our Ariba procurement system, we used the variable AribaNetworkID for identification and used a D&B D-U-N-S number as our BASE supplier identification:

```sql
insert into orgcode(orgcode_id, orgentity_id, codetype, code) values (1, <BUYORG_ID>, 'AribaNetworkId', 'AribaNetworkIdValue');
insert into orgcode(orgcode_id, orgentity_id, codetype, code) values (1, <SUPORG_ID>, 'DUNS', '44211214');
```
4. Create a user from the Administration Console.
5. Assign the user to the buyer organization and select the buyer procurement buyer administration role.
6. Add the buyer and supplier organization mapping information in the BUYSUPMAP table.
7. Add the procurement protocol and buyer specific profile information in the PROCBUYPRF table.

**Note:** For details on creating a buyer organization and administration users and on selecting roles, refer to the site management chapter in *IBM WebSphere Commerce Fundamentals Version 5.4* and the online documentation. The D&B D-U-N-S number is a unique nine-digit identification sequence that identifies single business entities while linking corporate structures together. For more information on the D&B D-U-N-S number, refer to [http://www.dnb.com/duns_update](http://www.dnb.com/duns_update).
Setting up contract and pricing support for procurement systems

As one of the requirements of our BASE example, we also set up contract support for our procurement system. One of the key drivers to setting up the BASE Sell-Side Hub was the requirement to integrate with a major customer’s Buy-Side Hub. There are existing contracts between BASE and their customer that need to be captured in Business Edition. Business Edition supports the use of contracts between buyers and suppliers. Contracts can be based on the default contract or new contracts can be created. A Business Edition contract allows the buying organization to order products from BASE for a specified time at an agreed price under specific conditions.

The following describes how we set up a contract between BASE and its procurement system customer:

1. Open WebSphere Commerce Accelerator and create an account for the buyer organization with an account credit line.
2. From the Contract notebook, select Sales -> Accounts to display the Account page.
3. Select the buyer organization Account and click New Contract. Enter the following information in the Contract General Information page:
   – Contract name - Procurement System
   – Short description - Procurement System default contract
4. Click Customers from the navigation frame. Select the procurement system customer.
5. Click Product and Prices from the navigation frame. Enter the following information on the Pricing Page:
   – Apply an adjustment on the entire master catalog - 10%
6. Click Shipping from the navigation frame. Select a shipping provider from the Contract Shipping Providers page. Select a contract shipping charge type. Add a contract shipping address.
7. Click Payment in the navigation frame. Select a credit line as the payment method.
8. Click OK and Submit to complete the contract. As no contract approval was selected, the contract becomes active.

Note: For more information on contracts, refer to the managing business relationships in IBM WebSphere Commerce Fundamentals Version 5.4 and the WebSphere Commerce Version 5.4 Online Help.
9.4 Procurement integration customizations

WebSphere Commerce has been extended to provide procurement integration support in Business Edition. However, there may be specific procurement system customer requirements that necessitate further customizations. WebSphere Commerce Business Edition provides support for extending and customizing messages, schema and business logic. The most common reason for customizing procurement integration in Business Edition is to support protocols other than the default implementation of Ariba 7.1, cXML. As well as customizing the protocol support, the following three subsystems can be extended for procurement system integration:

- Order subsystem
- Member subsystem
- Catalog subsystem

Enabling procurement for other procurement systems

To enable procurement integration for different procurement systems such as Ariba, mySAP, Commerce One and others, Business Edition has the functionality to support different protocols such as cXML, OCI, and XCBL. Procurement integration provides out-of-the-box cXML support for Ariba. To support new protocols such as mySAP's OCI, you need to customize procurement integration. This customization includes the following:

- Registering the new protocol for procurement integration
- Creating the configuration file to map protocol-specific XML to procurement integration
- Customizing the procurement integration specific JavaServer Pages
- Customizing the order, member and catalog subsystems

Order subsystem customization

The order subsystem can be customized in the following ways:

- Adding extension tables or modifying the existing schema
- Adding entity beans to map to the table extensions or modifying existing beans to reflect changes in the schema
- Implementing new controller and task commands
- Extending the controller and task commands

Order schema extensions

The following tables can be customized to provide customer specific procurement integration requirements.
The ORDMEEXTN table extends the capability of the ORDERS table and contains procurement specific OrderItem information. Each row in the ORDER table represents an Order in the store. The ORDMEEXTN table includes the following procurement specific fields, as well as two comment and three customizable fields:

- **PAYLOADID**: unique identification generated by the procurement system such as Ariba, to track the order request.
- **BUYERORDERID**: the order number defined by the system of the buying organization.
- **REQUESTEDTIME**: the time the order was requested by the buying organization at the procurement system.

The ORDIMEXEXTN table extends the capability of the ORDERITEMS table and contains procurement specific OrderItem information. Each row in the ORDERITEM table represents an OrderItem in an Order. The ORDIMEXEXTN table includes the following procurement specific fields, as well as two comment and three customizable fields:

- **REQUESTEDSHIPTIME**: requested order item ship time.
- **SHIPASCOMPLETE**: whether partial shipment of the item will be accepted.

### 9.4.1 Member subsystem customizations

The member subsystem may be customized in the following ways:

- Adding extension tables to the existing member subsystem schema
- Adding entity beans to map to the table extensions or modifying existing beans to reflect changes in the schema
- Implementing new controller and task commands
- Extending the controller and task commands
- Receiving LogonRequest in HTML format

**Member schema extensions**

Buyer organization information is captured to facilitate the registration of a buyer with a supplier. The Member subsystem has been extended to capture procurement systems specific information in the PROCBUYPRF table. This table captures buyer identification and protocol information from the procurement system. The buyer identification parameters captured from the procurement system are user ID and department name. Specific customizations can be made by extending the PROCBUYPRF table.
Customizing authentication
The authentication functionality provided with procurement integration can be modified in one of two ways:

- Customizing the authentication type

  The default authentication method is to authenticate against the Business Edition database using the `ProcurementDBAuthenticationCmd`. To authenticate against an LDAP, you need to implement the `LdapAuthenticationCmd`. For third party authentication you need to implement the `ThirdPartyB2BAuthCmd`.

- Customizing the authentication level

  In procurement integration, you are provided with four possible levels of authentication. You may also specify a fifth level of authentication level. This must be specified in the database during buyer registration by extending the `ProcurementDBAuthenticationCmd`, `LdapAuthenticationCmd`, or `ThirdPartyB2BAuthCmd`.

9.4.2 Catalog subsystem customizations
The catalog subsystem can be customized in the following ways:

- Adding extension tables or modify the existing schema
- Adding entity beans to map to the table extensions or modifying existing beans to reflect changes in the schema
- Implementing new controller and task commands
- Extending the controller and task commands

9.5 Procurement integration example
This topic describes a procurement system integration scenario with Business Edition using BASE as our example supplier.

9.5.1 Requirements
The first task is to identify the procurement system integration requirements for the BASE site. The following requirements for procurement integration have been identified in Chapter 3, “Requirements” on page 37.

- Provide the ability for client organizations to place an order from compatible client systems to inter operate with the BASE system to place an online order.
- The functionality offered by the machine interface will be limited to placing an order and checking the status of a current or past order.
- Product information, pricing information, order request, order status request and error information for the configuration and operation of machine interface will be delivered to the client organizations in industry standard XML format.
- The ability should be provided for client organizations to either replicate the catalog locally or access the WebSphere Commerce catalog within the procurement systems network.
- The procurement system will support the following message types: quote, purchase order, and purchase order acknowledgement.
- The BASE system will support the procurement interface as a single end-use, that is, it is trusted that the employee has been properly authenticated.
- Integrate with a buyer's procurement system using Ariba's cXML protocol. Note that future requirements will include integrating with the OCI protocol.

9.5.2 Solution overview

As the BASE system has been developed using Business Edition and there is no requirement for replicating the catalog on the procurement system, we can identify the punch-out catalog as our mode of procurement system integration. We have also identified that the procurement system that BASE is required to integrate with uses Ariba cXML protocol. We can therefore identify the use of a two-step mode as our method of communication with the procurement system, the ability to send and receive XML messages over HTTP. We can also use Business Edition's contract and pricing support for procurement system integration. Authentication with BASE’s LDAP server will also be implemented. For updating status messages, the outbound messaging system will be enabled. Provisions will be made for integrating with mySAP’s OCI protocol. Figure 9-5 shows an overview of the procurement solution to be implemented at BASE.
Figure 9-5  BASE procurement integration solution overview.

The following describes the steps shown in the Figure 9-5 solution overview diagram.

1. The shopper chooses to order laboratory supplies and equipment from the procurement system browser interface.

2. A punch-out setup message is sent to the BASE system by the procurement system.

3. Business Edition authenticates or registers the user and sends the URL and information needed to bind the session back to the procurement system.

4. A new browser is launched on the requisitioning user’s desktop by the procurement system.

5. The user browses or searches the catalog and submits an order to the procurement system for approval.
6. The purchase order is approved and an order request message is sent to Business Edition for fulfilment.
7. Order status messages are sent to the procurement system by Business Edition using the Outbound messaging system.

9.5.3 Customizing BASE for procurement integration

For this redbook, it was not possible to develop an end-to-end procurement integration example against an actual procurement system. However, we did configure the procurement integration reference application that is available for download from the WebSphere Commerce support site. This section covers how to configure the buyer and supplier settings using the reference application. It also provides information on customizing the reference application for BASE specific customizations, including single sign-on and integration with the OCI protocols.

The high-level steps for configuring the reference application for BASE are as follows:
1. Download the reference application
2. Configure the supplier settings using PISupplierConfig.xml.
3. Configure the buyer settings using PIBuyerConfig.xml.
4. Create member groups for buyers
5. Create and configure a store
6. Enable single sign-on with an LDAP server
7. Configure the application for the OCI protocol

Refer to IBM WebSphere Commerce Business Edition Version 5.4 Integration Guide for the Ariba Buyer System for more detailed information.

Download the reference application

The reference application procurementintegration.zip file contains a SAR file for creating a store, XML files for configuring the buyer and supplier, as well as the IBM WebSphere Commerce Business Edition Version 5.4 Integration Guide for the Ariba Buyer System guide.

Pre configuration steps:
1. Download the reference application procurementintegration.zip from the following URL:
   http://www-1.ibm.com/support/manager.wss?rs=497&rt=0&org=SW&doc=4001593
2. Unzip this file to:
   \Websphere\CommerceServer\samples\pireference

3. Edit the config.bat file in any editor.

4. Change the commerce loader and commerce_xml path to the one on your
   machine. For example:
   ```
   set commerce_loader=c:\websphere\commerceserver\lib
   set commerce_xml=c:\websphere\commerceserver\xml\loader
   ```

5. Go to the line `java com.ibm.wca.MassLoader.MassLoad - dbname mall
   -dbuser db2admin -dbpwd db2admin` and change the dbname, dbuser, dbpwd to
   your Business Edition instance database name db2 userid and db2
   password. For example:
   ```
db2inst1
   ```

6. Save and close the file.

7. Copy the Config.bat file to the following directory:
   \Websphere\CommerceServer\xml\config

Configuring the supplier

The reference application includes the PISupplierConfig.xml file to configure a
new PISupplier as a member of Business Edition. You can modify the
PISupplier.xml for updating your own supplier information. The
PISupplierConfig.xml modifies the following Business Edition tables for the
reference application.

- MEMBER - updates membership information.
- ORGCODE - updates store owner ID.
- VIEWREG - updates the view registry for procurement integration messages,
  JSPs and commands.

To configure the buyer for the reference application, do the following.

1. Copy the PISupplierConfig.xml to the following directory:
   \Websphere\CommerceServer\xml\config

2. Open a command window and run the following command:
   ```
   cd \Websphere\CommerceServer\xml\config
   config -infile pisupplierconfig.xml -method sqlimport
   ```

Configuring the buyer

The reference application includes the PIBuyerConfig.xml file to configure a new
PIBuyer as a member of Business Edition. You can modify the

Chapter 9. Procurement integration  263
PIBuyerConfig.xml file for updating your own supplier information. The
PIBuyerConfig.xml file modifies the following Business Edition tables for the
reference application.

- MEMBER - updates membership information.
- ORGENTTY - specifies that the member is an organization.
- ORGENTTYTYPE - indicates that the type is an organization.
- ORGENTTYYNAME - gives the name of the organization.
- ORGCODE - represents the DUNS number or any other mutually agreed
  number.
- ADDRBOOK - adds an address book entry to the address book and tables.
- PROCSYS - registers the protocol that the buyer supports.

Before updating the buyer information, we found we needed to add the
addrbook, address and mbrattrval elements to the PIBuyerConfig.dtd file.

To add elements to the PIBuyerConfig.dtd file, do the following:
1. Edit the PIBuyerConfig.dtd file.
2. Add the following to the <!ELEMENT import> element just before the “*)>”
   
   | addrbook | address | mbrattrval |

3. Add the following lines at the end of the file:

   ```xml
   <!ELEMENT addrbook EMPTY>
   <!ATTLIST addrbook
      addrbook_id CDATA REQUIRED
      member_id CDATA REQUIRED
      type CDATA IMPLIED
      displayname CDATA REQUIRED
      description CDATA IMPLIED
   >
   <!ELEMENT address EMPTY>
   <!ATTLIST address
      address_id CDATA REQUIRED
      addresstype CDATA IMPLIED
      member_id CDATA REQUIRED
      addrbook_id CDATA REQUIRED
      orgunitname CDATA IMPLIED
      field3 CDATA IMPLIED
      billingcode CDATA IMPLIED
      billingcodetype CDATA IMPLIED
      status CDATA REQUIRED
      orgname CDATA IMPLIED
      isprimary CDATA IMPLIED
   ```
To configure the buyer for the reference application, do the following:

1. Copy the PIBuyerConfig.xml file to the following directory:
   
   c:\Websphere\CommerceServer\xml\config

2. Open a command window and run the following command:
   
   cd c:\Websphere\CommerceServer\xml\config
   config -infile pibuyerconfig.xml -method sqlimport

**Creating member groups for buyers**

Member groups in Business Edition are defined by the supplier. For example, a member group might be created for regional customers; another example would be grouping preferred buyers together. You can then offer different prices or customize the way the products are displayed to member groups. For procurement integration, the buyer belongs to one member group, the default group.

The PIBuyerConfig.xml sets up a member group for the reference application buyer called RefAppToolTech Store Users for Buyorg 5000. To complete the configuration of the procurement buyer, do the following.

1. Open the WebSphere Administrative Console and register a procurement organization.

2. Assign the procurement buyer administrator role to the organization.

3. Add a buyer with the following attributes:
   
   logonID = Ariba NetworkUserID
   password = SharedSecret

4. Assign the procurement buyer administrator role to the buyer.

5. Assign the buyer to the RefAppToolTech Store Users for Buyorg 5000 member group.

6. Restart the instance.

Refer to the WebSphere Commerce Version 5.4 Online Help for detailed instructions on registering a buyer.

**Creating and configuring the reference store**

After configuring the buyer and supplier, you can either create a new store using the reference application or modify an existing one. For our example, we chose to create a new store based on the PIReference application.

**Preconfiguration**

You must disable the cache in the WebSphere Commerce instance that is being used for the reference application. To disable the cache, do the following.
1. Start the IBM WC Configuration Manager Server service.

2. Open the Configuration Manager.

3. Select the instance and then the **CachingSubsystem**.

4. From the General page of the CachingSubsystem, deselect **Enable Cache** and click **Apply**. See Figure 9-6.

![Figure 9-6 Caching SubSystem](image)

**Figure 9-6  Caching SubSystem**

To publish the procurement enabled reference application, do the following.

1. Make sure the following services are running:
   
   – IBM HTTP
   
   – IBM WebSphere Administration Server
   
   – WebSphere Commerce Server instance

2. Copy the PIRefApp.sar to the WCBE_Install_Path\samplestores\ToolTech directory.
3. Go to the WCBE_Install_Path\cml\tools\devtools directory. Open the file SARRegistry.xml. Append the following code to the end before the last line, 
</SAR-properties>.

```xml
<SampleSAR
fileName="PIRefApp.sar"
relativePath="ToolTech">
<html locale="en_US"
featureFile="ToolTech/Feature_refapp_en_US.html"
sampleSite="ToolTech/preview/en_US/index.html"/>
</SampleSAR>
```

4. Copy the Feature_refapp_en_US.html file to the WCBE_Install_Path\samplestores\ToolTech directory.

5. Edit the file store-all.dtd in WCBE_Install_Path\xml\sar directory.

6. Add the following to the <store-asset> element just before the 
</store-asset>).

```xml
| clasifcode | clsfcodeds | catclsfcod |
```

7. Add the following at the end of the file:

```xml
<!ELEMENT clasifcode EMPTY>
<!ATTLIST clasifcode
clasifcode_id CDATA REQUIRED
domain CDATA REQUIRED
code CDATA REQUIRED
parentcode CDATA IMPLIED
>
<!ELEMENT clsfcodeds EMPTY>
<!ATTLIST clsfcodeds
language_id CDATA REQUIRED
clasifcode_id CDATA REQUIRED
description CDATA REQUIRED
>
<!ELEMENT catclsfcod EMPTY>
<!ATTLIST catclsfcod
catentry_id CDATA REQUIRED
domain CDATA REQUIRED
code CDATA REQUIRED
>
```


10. Click the **New** button on the right-hand pane.

11. We entered and selected the following information in the Create Store Archive page:

- Store archive - **base**
- Store directory - **base**
Chapter 9. Procurement integration

- Store owner - Seller Organization
- Sample store - PIRefApp

12. From the Store Archive list in Store Services, select the checkbox beside the store (PiBase) you want to publish and click Publish.

13. Accept the default publishing options.

14. Select the store from the Store Archive and click Refresh.

15. When the status reads Publish complete, select the store and click Publish Summary.

16. Click the Launch Store button to view or test your store.

**Note:** When buyers access your catalog through punch-out catalog mode, they see a slightly different navigation sequence than when viewing the store directly through Business Edition. The procurement integration messages replace certain standard WebSphere Commerce order flow operations. For example, the checkout process is replaced with message extensions to incorporate a procurement order processing flow.

**Business account for the supplier and buyer**

To complete the configuration of the procurement integration reference application, you must do the following.

1. Create a business account using WebSphere Commerce Accelerator.
   
   When creating an account, take into account the following:
   - Do not select the **Allow customers to purchase under the terms and conditions of store’s default contract** checkbox in the Customer page.
   - Do not select the **Purchase order number may be specified at the time of the order** checkbox in the Purchase Order Page.
   - Do specify the credit line account number in the Credit Line Page.

2. Create a contract for the business account created previously.
   
   When creating a contract, do the following:
   - Select the **Allow the payment using the account’s credit line** checkbox.
   - Do not specify any contract shipping address.

3. Modify the payment method information for the store using the following SQL:

   ```sql
   update cmdreg set
   classname='com.ibm.commerce.payment.commands.DoPaymentMPFCmdImpl' where storeent_id='0' and
   interfacename='com.ibm.commerce.payment.DoPaymentCmd'
   ```

   Your store is now enabled with procurement integration.
Customizing the application for single sign-on
In our example, there is a requirement for single sign-on using an LDAP directory. The default authentication mechanism provided by Business Edition is authentication against credential information stored in the instance database using the DBAuthenticationCmd. To customize authentication to use an LDAP directory, we need to implement the LdapAuthenticationCmd. Refer to the WebSphere Commerce Version 5.4 Online Help for more information on implementing the LdapAuthenticationCmd.

Enabling the application for the OCI protocol
The OCI protocol supports a logon request to the supplier’s site in HTML format. OCI is used by the mySAP e-procurement system and the Commerce One BuySite e-procurement system.

The following are the high-level steps needed to enable the reference application for the OCI protocol:
1. Register the OCI protocol with procurement integration
2. Create configuration files for OCI
3. Customize JavaServer Pages
4. Customize the order, member and catalog subsystems

In addition, modifications need to be made to the PunchOutSetupCmd and ClData Class to parse and return the HTML parameters as represented in OCI.
9.6 More information

For more information on procurement system integration, see the following references:

  

- *IBM WebSphere Commerce Version 5.4 Business Edition Integration Guide for the Ariba Buyer System*, IBM
  
  http://www-1.ibm.com/support/manager.wss?rs=497&rt=0&org=SW&doc=4001593

  
  http://www.cxml.org

- *Open Catalog Interface (OCI)*, Release 2.0B, SAP AG,
  
Customer care

This chapter describes how the sample store ToolTech is extended with additional benefits to the user. The requirements of our customer demand an integration with a collaboration feature. The WebSphere Commerce Business Edition provide this new feature in two ways: first, there are collaboration workspaces and the customer care feature, and secondly, there are approval cycles which are performed within the system. We chose to discuss the customer care feature in our store.

We discuss a high-level view of the Collaboration patterns described in 4.4.2, “The Collaboration business pattern” on page 61 and examine what WebSphere Commerce Business Edition provides to implement the requirements stated in Chapter 3, “Requirements” on page 37. As a detailed example, we describe the steps necessary to implement the customer care feature in the sample store of WebSphere Commerce Business Edition and consider whether there is more customized code needed to meet the requirements.

The following topics are covered:

► Positioning statement
► Implementation
10.1 Positioning statement

In this chapter, we discuss the link between the Collaboration patterns discussed in 4.4.2, “The Collaboration business pattern” on page 61 and the collaboration features included in WebSphere Commerce Business Edition. In the collaboration requirements identified, there are two major areas to consider:

1. In the first place there is the approval cycle. When a customer places an order, it may have to be approved by the buyer’s organization representative. This occurs when the order has exceeded a predetermined monetary value, after which the order has to be approved.

2. Secondly, collaboration is required to support an online customer who is facing difficulties in placing an order correctly. This user can request online help from a CSR via a synchronous chat interface. Also, buyer and seller representatives can use collaboration to negotiate terms and conditions of contracts.

In this section, we link our selected e-Marketplace Sell-Side Hub composite pattern with our customer requirements and describe how we can use WebSphere Commerce Business Edition to fulfill their needs. In 4.4.2, “The Collaboration business pattern” on page 61 you can see the Collaboration pattern as an optional artifact of the Sell-Side Hub composite pattern. The two major topics within collaboration are directed collaboration and managed collaboration.

Directed collaboration is described as a synchronous interaction between several clients using one central server. The users communicate directly among themselves. On the central server, the user data is stored in a directory. Every client has a shared application and a local address book.

Managed collaboration is described as clients using a central server to interact. This interaction can be synchronous or asynchronous. There is no direct interaction between the users themselves. There is one central directory and shared application repository on the server. Additionally, there is a workflow component behind the interaction which acts on underlying workflow rules.

We can split our business drivers identified in Chapter 3, “Requirements” on page 37 into these collaboration groups as follows:

1. Directed collaboration
   Maintain customer satisfaction by providing online assistance to users when required.
2. Managed collaboration
   – Reduce costs by reducing the staff workload associated with servicing client orders.
   – Decrease end-to-end response order time by providing a Self-Service capability.
   – Delegation support will be provided to customers for some areas of functionality, allowing users to perform some functionalities, but not others, without approval.

**WebSphere Commerce Business Edition and collaboration**
WebSphere Commerce Business Edition offers several functions to implement the collaboration in an online shop. We can group them in the same collaboration topics.

1. Directed collaboration
   We can implement directed collaboration using the customer care and collaborative workspaces features of Business Edition.

2. Managed collaboration
   This is used in the approval workflows by the organizations participating in the Sell-Side Hub.

Whether a buyer or seller is discussing details of a purchasing contract or a Customer Service Representative is assisting a customer in placing an order, e-commerce collaboration is necessary to maintain the communication flow between parties.

Traditionally, these communications have taken place by phone or fax, but now WebSphere Commerce Business Edition has two Web-based collaboration features available for its e-commerce solutions. These features are accessed through the WebSphere Commerce Accelerator.

**Approval cycles**
The approval features are quite complex in WebSphere Commerce Business Edition so we explain their functionality with an example.

The seller organization offers products or services through the online shop. Registered customers can access the shop and browse through the catalog to place an order. There are contracts between the seller and the buyer in which the terms and conditions are fixed for specific trading activities. When order approvals are enabled, if the customer exceeds a pre-determined order limit which is defined in the contract, the order is placed, but enters the approval status. The buyer organization has an approver who logs into the organization...
administration console and finds the order he must approve or reject. After the approval has been given, the order status changes in the buyer's account. The next step will now be procurement approval. Any procurement approval steps depend on the company's policy, but after these steps are followed, the order is fulfilled and the order status is back-ordered. Approval communications are in asynchronous mode. The internal workflow behind an approvals cycle automatically informs the appropriate people of actions they need to take.

**Collaborative workspaces**
Collaborative workspaces allow the different parties (a buyer and a seller) to negotiate contracts, including terms and conditions, using the Lotus QuickPlace interface. Business users within the seller's organization can also communicate using this interface. Collaborative workspaces support asynchronous communication. Collaborative workspaces can be created independently of the buyer's role in the organization, but only an account representative or an account manager with QuickPlace manager access can add buyers to the collaboration workspace. Member data is stored in an LDAP server with WebSphere Commerce, not in a relational database, so QuickPlace must be installed on a separate machine from the one used by your WebSphere Commerce and LDAP installation. For detailed information about the installation, refer to the *IBM WebSphere Commerce Professional and Business Edition for Windows NT and Windows 2000 Additional Software Guide Version 5.4*. The WebSphere Commerce system manages all details about the collaborative workspaces and member information. In contrast, the discussion threads, postings or file attachments are handled by the QuickPlace server.

**Customer care**
The customer care feature enables the customer to get in contact with a CSR via the Internet. It provides real-time customer service support using a text-based chat interface. The customer simply clicks a link on the Web page to get in contact with a CSR. On the support site, the CSR accesses his support interface through the WebSphere Commerce Accelerator. The CSR chats with the customer but can also see which page the customer is looking at or how many products he has in his shopping cart. In the same way, the CSR can use his chat window to communicate with other CSRs in the seller's organization. To use this feature, Sametime has to be installed and configured with WebSphere Commerce. For installation instructions, refer to the *IBM WebSphere Commerce Professional and Business Edition for Windows NT and Windows 2000 Additional Software Guide Version 5.4*. 
10.1.1 BASE requirements for collaboration

In this section, we explain the collaboration requirements of BASE as described in Chapter 3, “Requirements” on page 37. Our customer requests the following functionality.

- Users will be able to interoperate with BASE Customer Service Representatives (CSRs) via text-based chat for support and troubleshooting.
- CSRs will be able to view user interaction with the system to assist users.
- Only non-anonymous users will be able to use the customer care facility.
- CSR viewing of user interaction will only be granted on a per occurrence permission from the user.
- The CSR will have the ability to perform all interactions on behalf of users.

Proposed solution

We will provide, on every page of the online shop, a link to get in contact with a CSR. When the user clicks this link, an applet pops up and welcomes the user. As soon as the CSR accesses this request, they can chat synchronously with the customer via the Internet. In WebSphere Commerce Business Edition, the system administrator can decide whether the CSR can only get in contact with the customer if the customer initiates the help requests. In addition, the administrator can set the following options.

1. Monitor type:
   - Monitor waiting queue
   - Monitor all customers in shop
   - Monitor waiting queue and all customers in shop
     These settings change the view of the CSR applet and control which customers the CSR can serve.

2. Initiation type:
   - The user can initiate a session
   - The CSR can initiate a session
     These settings define whether only the customer can initiate a help session or if both the customer and the CSR can initiate help sessions.

3. Number of sessions:
   This defines how many help sessions every CSR can communicate with in parallel.
The actions the CSR can execute within an online help session include:

- Viewing all or just waiting customers.
- Viewing profile and shopping cart information of customers.
- Communicating with customers.
- Sending a Web page to the customer’s browser.
- Viewing the page the customer is looking at.
- Switching between different LiveHelp sessions.
- Communicating with co-workers.

More customizations can be performed by the administrator in cooperation with the store developer, for instance, specific pages can be marked as private, so that the CSR cannot access them if the customer is browsing them. For more detailed information on which customizations are possible, refer to the IBM WebSphere Commerce Store Developer’s Guide Version 5.4.

10.1.2 Example workflows using customer care

As we have seen, there are several ways to understand the term collaboration. We selected one example to consider the details. This is the customer care feature. We set up the ToolTech sample store and applied the necessary customer care features. Some key flows in the sample store which we expect to be used the most within the live help context are described in this section.

The help request cycle

In this section, we describe how the help requests flow through the WebSphere Commerce system. There are several tasks involved in using the live help feature. These include the following.

**Note:** There must be at least one CSR registered by the seller organization.

1. The CSR logs in to his help desk.
2. The customer is logged in and requests help by clicking the help link.
3. A CSR sees the queue of waiting customers and serves the next one.

Registration process for a CSR

Figure 10-1 shows the registration process used by a CSR to interact with the help request system.
Figure 10-1  Registration of a CSR

The steps in the registration process are:

1. The site administrator logs in to the WebSphere Administrative Console.
2. Using the access management tools, a new user is added to the local repository. This user belongs to the seller’s organization.
3. The role of CSR is assigned to the new user and the user is registered for customer care. In order to apply roles to users, these roles must have been enabled first. This is usually done when the seller’s organization is set up.
4. The CSR is now added to the Domino directory. For registration in the Sametime server, a user name is provided and the password is generated.
5. The site administrator receives a message that confirms the successful addition of the user to the Domino directory.
**Customer logon process**

Figure 10-2 shows the customer logon process.

![Diagram of customer logon process](image)

Figure 10-2  Customer logon process

The steps in the logon process are:

1. The customer enters the shop using a Web browser.
2. After logon, the start page is downloaded from WebSphere Commerce.
3. The Java applet is downloaded from the Sametime server in a hidden frame, so that the user does not notice the applet until the help link is clicked.
4. The Java applet logs on to the Sametime server using the customer's user ID.
5. Information about the shopper is sent by the applet to the customer care workspace. The information sent includes the customer's ID, the current page being viewed, and the time since the customer logged in.
CSR logon

Figure 10-3 shows the CSR logon process.

The steps in the CSR logon process are:

1. The CSR logs on to the WebSphere Commerce Accelerator.
2. The CSR is authenticated by the system.
3. The role of the CSR is retrieved from the database.
   Depending on his role, the CSR sees only a subset of possible functions in WebSphere Commerce Accelerator.
4. The CSR logs on to customer care.
5. The customer care Java applet is downloaded from the Sametime server.
6. The applet logs on to the Sametime server using the CSR’s user ID.
7. The applet retrieves informations about the list of customers and their status.
Shopper initiates live help

Figure 10-4 shows an overview of the process followed when a shopper requests live help.

The steps in the live help process are:
1. The CSR is logged on to the customer care applet.
2. The customer is logged on at the storefront.
3. The customer initiates live help by clicking the help link.
4. The Java applet pops up and shows a message to the customer.
5. The CSR gets information about the customers in the shop.
6. The CSR serves the next customer in the waiting queue. The message in the customer applet changes to a welcome message and the input field becomes active.
7. The CSR’s applet changes to a chat window.
8. The chat begins.

10.1.3 Communication process

Figure 10-5 shows a view of the Java applet that pops up after the customer has clicked the online help link. The customer is now connected through the applet to a Sametime server. Major information relevant to the customer is pushed to the applet of the CSR. The CSR and the customer can now start chatting. The customer has a limited number of choices to act upon using this applet. The
customer can send messages, or end the session by clicking the Close button or by closing the applet window.

Figure 10-5  The customer’s view

Figure 10-6 shows the CSR's applet. The CSR has entered a session with the customer who initiated the help request. In this view, the CSR has multiple types of information he can receive from the customer and can take a number of actions, including:

- Sending the customer’s browser to a new URL.
- Looking at the customer’s profile.
- Opening the customer’s shopping cart.
- Opening the same page the customer is currently viewing.
- Displaying the number of items in the customer’s shopping cart.
- Ending the session
Additionally, the CSRs can chat among themselves using the Coworker tab in their customer care applet. Using this tab, registered and logged on CSRs can participate in the conference. By default, the coworker functionality is not public to all the business members inside the seller’s organization, but co-worker chat is available since the CSR can choose to open another chat window which is public to all other business members in their company.

Using the monitoring tab, the CSR can see, if this is enabled, how many customers are already in the store. The CSR can choose a customer and look at their shopping cart and profile even without an initiated help session. The CSR can then decide which of the customers he prefers to serve first.

The customer’s monitoring information is structured as follows.

- Customer
  Displays the customers user ID if they signed in, otherwise displays as a guest shopper with an assigned user ID.

- InSite
  This is the time the customer has already spent online at the shop. This time is reset after the CSR logs into his workspace.

- Current page
  Displays the title of the Web page the customer is currently viewing.
In Page

This is the time the customer has spent already on the current Web page. The timer resets when the CSR logs on to his workspace.

Cart

This number reflects the number of items in the customer’s shopping cart.

**Restriction:** Because we decided in our requirements that only registered customers can use the live help feature, you must first register a user to retrieve valid information about them. In the ToolTech sample stor, a self register form is available that is not part of the BASE Pty. Ltd requirements.

**CSR access to the customer care workspace**

To get access to the customer care workspace, the CSR uses the WebSphere Commerce Accelerator as follows.

1. Select Programs -> IBM WebSphere Commerce -> WebSphere Commerce Accelerator.
2. Enter your CSR user ID and password.
3. Select the desired store and desired language. This means the language in which the WebSphere Commerce Accelerator is displayed, not the store language. Click OK.
4. The WebSphere Commerce Accelerator interface depends on the role of the user. For a CSR, there are three main navigation points: Store, Sales and Help.
5. Select Sales -> Customer Care.
6. The customer care applet pops up. Click Yes on the security warnings window.
7. The CSR is now registered in his role as Customer Service Representative and already logged on to the Sametime server. He is ready to serve the customers.

The customer care workspace is divided into three tabs.

- The monitoring tab
- The session tab
- The coworker tab

Each of the tabs provides access to different collaboration features.

The monitoring tab shows the active shoppers in the shop with all their available information, as shown in Figure 10-7.
The session tab shows the active sessions of the CSR. The status of these sessions is visible on the Session buttons which change color depending on the customer's status. See Figure 10-8.
The coworkers tab provides access to a conference where all logged-on CSRs of the same business organization can communicate with each other. See Figure 10-9.

![CSR applet: coworkers tab](image)

**Figure 10-9  CSR applet: coworkers tab**

The CSR can also communicate with other colleagues in their business organization who are not in the role of a Customer Service Representative. To do this, select View -> Buddy List. Since the CSR is already logged on, he can add some more buddies. See Figure 10-10.
In addition to the features already mentioned, the system administrator can allow the CSRs to do more than just react to the requests of customers. The administrator can allow the CSRs to see all customers who are currently browsing the store, not just the ones requesting help. The CSR can see how long a customer has been browsing through the shop and then initiate a help session. In this case, the applet pops up on the customer's view asking whether or not he/she needs some help.

**Language support**

The customer care workspace as well as the customer's applet support several languages. The CSR's workspace applet language depends on the languages provided when publishing the store, but the customer applet's language depends on the language of the customer's computer.

### 10.1.4 Customer care customization

All the functionality requested by our BASE customer comes out of the box with WebSphere Commerce Business Edition, except the requirement that only non-anonymous users may get online help from a CSR.

However, if we base our solution on the ToolTech sample store to implement the customer care functionality, we meet this requirement, because in ToolTech only
registered users will be able to request an online chat with a CSR, since users must first log in to access the shop.

The requirement for BASE Pty. Ltd is access to a limited catalog view for anonymous users. To provide access to help sessions only for registered users, we can place a simple if condition in the JSP files for our store. The condition tests whether the user is a registered shopper. If not, the online link is not displayed on the Web page. Also, the commands used to display the catalog pages must be enabled for use by unregistered users. This is done by updating the AUTHENTICATED column in the URLREG table of the WebSphere Commerce database. The following commands need to be updated:

- StoreCatalogDisplay
- TopCategoryDisplay
- CategoryDisplay
- ProductDisplay
- CatalogSearchResultView
- AdvancedSearchView

These commands are by default enabled for anonymous users in ToolTech.

10.2 Implementation

In this section, we provide guidance on what has to be done to use the Sametime customer care feature with ToolTech. We use a frameset approach to integrate the Java applet to the shop pages. The customer care feature is already integrated in the ToolTech store. Nevertheless, we describe the differences from a standard installation of a WebSphere Commerce shop without customer care. For detailed information about the frameset approach and the implementation of customer care into your own shop, refer to IBM WebSphere Commerce Store Developer’s Guide Version 5.4.

View Java applet

To have the customer care Java applet accessible on all Web pages inside the store, one approach is to use a frameset where the applet resides on a hidden frame in the store pages. In this case, the applet loads immediately after the customer enters the shop. The applet pops up in a new browser window only when the customer clicks the Live Help link.

To access a store main page in WebSphere Commerce, you often use a URL such as:

```
https://<host_name>/webapp/wcs/stores/servlet/Logoff?storeId=10001&langId=-1&catalogId=10001
```
The frameset approach assumes that the customer can only access the shop through a single entry point. This must be the frameset page, otherwise the subframes are not loaded and the applet is not accessible. The new URL to access the shop is now:

https://<host_name>/webapp/wcs/stores/servlet/<store_name>/index.jsp

where <store_name> is the name of your store archive.

For example:

https://ka0kltb.itso.rai.ibm.com/webapp/wcs/stores/servlet/mystore/index.jsp

This means that the user cannot use common browser features such as accessing an inner shop page through a bookmark or using the reload button of the browser, because it loads the frameset again, as well as the default subframes. This can be fixed by coding the index.jsp to only use the frameset if customer care is enabled. If customer care is enabled, the URL to call the shop switches automatically to the frameset when you launch the shop using Store Services. No frameset is used if customer care is not enabled. Example 10-1 shows a code snippet of the index.jsp of ToolTech.

Example 10-1  Code snippet of the frameset JSP index.jsp

```html
<flow:ifEnabled feature="customerCare">
  <script src="<%="Sametime.js"%>"></script>
  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "DTD/xhtml1-transitional.dtd">
  <html>
  <head>
  <title></title>
  </head>
  <FRAMESET border=0 frameBorder=0 ROWS="100%,1,1,1,1,1">
  <FRAME NAME="main"
  SRC="/webapp/wcs/stores/servlet/Logoff?storeId=<%=storeId%>&catalogId=<%=catalogId%>&langId=<%=langId%>" MARGINWIDTH=0 SCROLLING="Auto" FRAMEBORDER="no" noresize>
  <FRAME NAME="sametime" SRC="<%="StFrame.jsp"%>" MARGINWIDTH=0 SCROLLING="no" FRAMEBORDER="no" noresize>
  <FRAME NAME="JSFrame" SRC="<%="StBlank.jsp"%>" MARGINWIDTH=0 SCROLLING="no" FRAMEBORDER="no" noresize>
  </FRAMESET>
  </html>
</flow:ifEnabled>

<flow:ifDisabled feature="customerCare">
</flow:ifDisabled>
```

290  B2B e-commerce With WebSphere Commerce Business Edition
If the customer care feature is enabled, the code within the tags:

```html
<flow:ifEnabled feature="customerCare">
</flow:ifEnabled>
```

is executed. This builds a frameset that consists of three frames:

- **main**
  
  The source of the main content page:

  `/webapp/wcs/stores/servlet/Logoff?storeId=<%=storeId%>&catalogId=<%=catalogId%>&langId=<%=langId%>`

- **sametime**
  
  This frame contains the Java applet:

  StFrame.jsp

- **JSFrame**
  
  This is a blank page:

  StBlank.jsp

To make the frameset dynamic, the source for the main frame can be delivered as a parameter and this parameter can be read out of the URL.

If the customer care feature is disabled, the code within the tags:

```html
<flow:ifDisabled feature="customerCare">
</flow:ifDisabled>
```

is executed and instead of using the frameset, the browser is forwarded to the default URL for the store:

`/webapp/wcs/stores/servlet/Logoff?storeId=<%=storeId%>&langId=<%=langId%>&URL=LogonForm?storeId=<%=storeId%>&catalogId=<%=catalogId%>`

### Retrieving the customer's name or ID

As soon as a customer accesses the shop, he is monitored by the customer care system. The customer care applet of the CSR uses this information to identify the customer who is requesting help through the chat interface. The customer is identified by name or simply by a generated number, depending on his
registration status. If he is an unregistered guest shopper without any items in
the shopping cart, the CSR is only able to see an generated ID for the customer
with a shopper ID set to -1002.

If the customer is a guest shopper with some items in his cart, his proper shopper
ID is displayed. His first name and last name displays if he has already logged in.

This is done by calling a JavaScript function in the NavHeader.jsp. In this case,
the user is already logged on. See Example 10-2.

Example 10-2 Retrieving the customer name or ID in the JSP file

```javascript
<script language="javascript">
  if (typeof parent.setCustomerName == 'function')
    parent.setCustomerName(<%=uid%>, '<%=customerName%>');
</script>
```

These functions are located in the JavaScript file Sametime.js. They send
information to the CSR’s applet as we can see in Example 10-3.

Example 10-3 Send customer ID or name to customer care applet

```javascript
function setCustomerName (newCustId, newCustName) {
  var needToSend = false;
  var sendName = "";
  var newId;

  if (newCustName=="") newCustName=CustomerName;

  if (newCustId != WcsCustomerId)
  {
    WcsCustomerId = newCustId;
    needToSend = true;
  }

  //Handle the Wsc CustomerName

  if (newCustName != CustomerName || newCustName == null || newCustName == "")
  {
    needToSend = true;
    if (newCustName == null || newCustName == "")
    {
      newId = Math.round(Math.random() * 1000000);
      CustomerName = "" + newId;
      if (newCustId == WCSGUESTID)
        WcsCustomerId = WCSGUESTID + "" + newId
      sendName = CustomerName;
    }
```
else
{
    CustomerName = newCustName;
    sendName = newCustName;
}
} 
else 
{
    sendName = newCustName;
} 
if (needToSend)
{
    if (CustomerAppletIsUp)
    {
        sametime.document.applets["InteractivePresenceApplet"].SetWcsCustomerId(WcsCustomerId+"@"+sendName);
        sametime.document.applets["InteractivePresenceApplet"].SetCustomerName(sendName);
    }
}

This function serves the Java applet with the user ID which can be generated by this function or can be the real name of the customer. Example 10-3 includes a code snippet which shows how the script sends information to the CSR's Java applet.

Determining what page the customer is currently browsing

The JavaScript function also passes other information to the customer care applet. The page that the customer is currently browsing is captured by code in the NavHeader.jsp, as shown in Example 10-4.

Example 10-4   Determining the current page

String headerType = (String) request.getAttribute("liveHelpPageType");
if (headerType==null)
    headerType = "";

<script language="javascript">

String pname = request.getRequestURI();
int indpn = pname.lastIndexOf('/');

indpn = pname.lastIndexOf('/', indpn-1);
if(indpn != -1)
    pname = pname.substring(indpn+1);

    //Determine if this is a personal page or not
    if (headerType.equals("personal")) {
        if (typeof parent.setPageParams == 'function')
            parent.setPageParams('PERSONAL_URL', '<%=pname%>');
        } else {
            if (typeof parent.setPageParams == 'function')
                parent.setPageParams(location.href, '<%=pname%>');
        }
    }
</script>

Before the URL is sent, the function checks whether the page is marked as personal. To mark a page as personal, the JSP generating the page uses code such as that in Address.jsp, as shown in Example 10-5.

Example 10-5   Marking a page as personal

<% request.setAttribute("liveHelpPageType", "personal"); %>
<% String incfile;
    incfile = includeDir + "NavHeader.jsp";
%>
<jsp:include page="<%=incfile%>" flush="true"/>

If the page is marked as personal, the setPageParams() function sends the page URL and a default message to the applet instead of sending the page content. See the code of Sametime.js shown in Example 10-6.

Example 10-6   Sending the URL and page name to the customer care applet

function setPageParams(pCurrPageURL, pCurrPageDscr)
{
    CurrPageURL = pCurrPageURL;
    CurrPageDscr = pCurrPageDscr;

    if (CustomerAppletIsUp)
    {
        sametime.document.applets["InteractivePresenceApplet"].SetPageParameters(CurrPageURL, CurrPageDscr);
    }
}
Retrieving the number of items in the shopping cart
The customer care feature enables the CSR to track the number of items in the customer’s shopping cart even if the shopping cart page is marked as personal. The CSR cannot see the content of the cart, but is able to use the Shopping cart button in the customer care applet.

There are several tasks to complete to view the number of items in the cart.

1. First, a variable has to be defined in the head section of the Shoppingcart.jsp.  
   ```java
   int liveHelpShoppingCartItems = 0;
   ```

2. Next, for every item in the shopping cart, this function is called:
   ```java
   liveHelpShoppingCartItems++ = orderItem.getQuantityInEJBType().intValue();
   ```

3. Then the following code at the end of the Shoppingcart.jsp submits the number of items.
   ```javascript
   if (typeof parent.setCustomerName == 'function')
   parent.setCustomerName(<%=cmdcontext.getUserId()%>, parent.CustomerName);
   if (typeof parent.setShoppingCartItems == 'function')
   parent.setShoppingCartItems(<%=liveHelpShoppingCartItems%>);
   ```

4. To ensure that the number is reset if the customer completes his order, the following code must be added to the Confirmation.jsp.
   ```javascript
   if (typeof parent.setShoppingCartItems == 'function')
   parent.setShoppingCartItems(0);
   ```

This resets the number of items for display in the CSR’s applet.

10.2.1 Installation steps
Lotus Sametime enables customer care collaboration. It provides customer care real-time support through synchronous chat interface, also known as Instant Messaging (IM). The Customer Service Representative can also communicate with other CSRs in their company. Other registered business users inside the company are also able to communicate via Sametime using an additional chat window supplied by Sametime.
To use the customer care feature, follow these steps:

1. Verify the system where you are going to install this feature.
3. Install the WebSphere Commerce Sametime integration packages. For information on how to apply this, see the IBM WebSphere Commerce Professional and Business Edition for Windows NT and Windows 2000 Additional Software Guide Version 5.4.
4. Enable Sametime in the configuration manager.
5. Create CSRs and shoppers and register CSRs for customer care using the Administration Console.

### 10.2.2 Prerequisites

Lotus Sametime cannot be installed on the same system where your Commerce Server or your LDAP server is installed.

First, you have to download the latest fixpack for WebSphere Commerce Business Edition. Go to the Web site:


At the time we wrote this redbook, we downloaded the WebSphere Commerce V5.4.0.2 FixPak. Read the installation instructions and apply the fixpack. This is necessary because, in the basic version of TechTech which shipped with WebSphere Commerce Business Edition, not all the customer care functions worked properly. In particular, we had problems with the functions used to retrieve the user information. The chat functionality worked but other features such as the display of the user information (name or shopping cart items) did not work until we applied the WebSphere Commerce fixpack.

We also applied the Sametime public Fixpack#1. We obtained this from the URL:


Follow the installation instructions in the readme.txt file.

### 10.2.3 Enabling customer care

You must now enable the customer care feature in the WebSphere Commerce Configuration Manager. On your WebSphere Commerce machine, do the following:
1. In the Services window, stop the IBM WS AdminServer 4.0 service.

2. Open the WebSphere Commerce Configuration Manager by doing the following:
   a. In the Services window, start the IBM WC Configuration Manager Server.
   b. Select **Programs -> IBM WebSphere Commerce -> Configuration**.
   c. Enter your Configuration Manager User ID and password.
   d. Expand the tree `host_name -> Instance List -> instance_name -> Instance Properties -> Collaboration -> Sametime`, and do the following:
      i. Select the **Enable** checkbox.
      ii. Enter your fully qualified hostname into the host name field.
      iii. Enter the host name of your Sametime server in the registration URL field.
         **Note**: Only change the host name. The field should look like this:
         
         `http://<host_name>/streg.nsf/557a6148a8f846d38525610e0000ca95?CreateDocument`

   iv. Enter the host name of your Sametime server in the applet codeBase URL.
      
      Only change the `host_name`. For example:
      
      `http://host_name/wc`
      
      See Figure 10-11 for details.
v. Click **Apply**. A message appears stating that Sametime has successfully configured with WebSphere Commerce. Click **OK**.

vi. Close the WebSphere Commerce Configuration Manager. A message appears stating that, for security reasons, the IBM WebSphere Commerce Configuration Manager Server has been stopped. Click **OK** to continue.

e. In the Services window, start the IBM WebSphere AdminServer 4.0.

After enabling customer care in the Configuration Manager, we must enable our shop to use the customer care feature. To do this, follow these steps.

1. Open Store Services:
   a. Select **Programs** - > **IBM WebSphere Commerce** - > **Store Services**.
   b. Log on using your Store Services User ID (wcsadmin) and password.

2. Select **View** - > **Stores**.

3. Select the checkbox of your store and click **Configure** in the right menu.

4. On the Customer Care tab, select **Enable Customer Care**.
5. Click **Apply** or **Apply Permanently**. Selecting **Apply** allows you to enable or disable the feature in future. If you choose **Apply Permanently**, you will not be able to reverse your choice. For more information, refer to the WebSphere Commerce Version 5.4 Online Help.

### 10.2.4 Registering users

To use the customer care feature you first have to register at least two users:

- A Customer Service Representative (CSR) of the seller organization
- A shopper of a buyer organization

To register, do the following:

1. Select **Programs -> IBM WebSphere Commerce -> Administration Console**.
2. Log on using your Administration Console User ID (wcsadmin) and password.
3. Select **Site** and click **OK**.
4. Select **Access Management -> Users**.
5. First, we register a CSR. Click **New** in the right-hand menu.
   a. On the Details tab, fill out all required fields and in the Account Policy, select **Administrator**. Click **Next**.
   b. On the Business Profile tab, fill out the required fields. At the Parent Organization section, expand the Root Organization Tree and click **Seller Organization**. This is a default organization which is pre-configured by ToolTech. Click **Next**.
   c. On the Address tab, fill out all required fields. Click **Next**.
   d. On the Contact tab, you can fill in some contact information, but this is not mandatory. Click **Finish**.

   On the user's view, you will see your recently created user.

6. To register a shopper, click **New** in the right-hand menu.
   a. On the Details tab, fill out all required fields and in the Account Policy, select **Shopper**. Click **Next**.
   b. On the Business Profile tab, fill out the required fields. At the Parent Organization section, expand the Root Organization Tree and click **Buyer Organization A**. In Figure 10-12, you see the highlighted Parent Organization. This is a default organization which is pre-configured by ToolTech. Click **Next**.
   c. On the Address tab, fill out all required fields. Click **Next**.
d. On the Contact tab, you can fill in some contact information, but this is not mandatory. Click **Finish**.

On the user's view, you will see your recently created users.

![Registering CSR view with focus on the Parent Organization](image)

**Figure 10-12  Registering CSR view with focus on the Parent Organization**

7. The next step is to apply some user roles to the users. To do this, you must first apply the role for our CSR.
   a. In the user's overview, select the checkbox of your designated CSR and click **Roles**.
   b. Select **Seller Organization**.
   c. In the Role select box, select **Customer Service Representative** and click **Add**.
      
      The user roles appearing here depend on the roles you have chosen when creating an organization.
   d. The user role appears in Selected roles. Click **OK**.
   e. In the user's overview, you can see this user and the corresponding applied role. Select the checkbox of the CSR user and click **Register Customer Care**.
      
      **Note:** This link is clickable if the user has an appropriate user role.
   f. If any Security pop-up windows appear, click **OK**.
g. Your CSR is now a registered Sametime user. In your browser window, you see the Sametime User Registration confirmation for your CSR. Click OK.

8. Apply the role for our shopper.
   a. In the user's overview, select the checkbox of your designated shopper and click Roles.
   b. Select Buyer Organization.
   c. In the Role select box, select Buyer (buy-side) and click Add.
      The user role appearing here depends on the roles you have chosen when creating an organization.
   d. The applied user role appears in the Selected roles. Click OK.
   e. In the user's overview, you can see this user and the corresponding applied role.

Privacy consideration
When the customer enters the shop and authenticates himself through the login page, he does not know what information the CSR can get about his session. This depends on the privacy policy of the store. The store developer can add code to JSPs to mark them as personal pages so the CSR is unable to retrieve information from these pages.

The system administrator can also determine what the CSRs can view from the customer's profile or shopping cart as described in “Proposed solution” on page 277.

10.2.5 TCP/IP ports used by Lotus Sametime

Lotus Sametime uses many different ports to communicate over the network between its components. The main component we use in our customer care example is the chat interface. After the Java applet is downloaded and initialized, it connects to the Sametime server using port 1533. For security reasons, this port is disabled in BASE's firewall. Moreover, the customer may be behind his corporate firewall and unable to communicate over the Internet using port 1533.

Table 10-1 shows the ports used by Lotus Sametime when using customer care:
Table 10-1  Ports used by Sametime.

<table>
<thead>
<tr>
<th>Port</th>
<th>Used by customer care in BASE?</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>Yes, the Sametime HTTP server listens for incoming HTTP requests from browsers.</td>
</tr>
<tr>
<td>389</td>
<td>No, used for LDAP communication.</td>
</tr>
<tr>
<td>443</td>
<td>No, the Sametime HTTP server listens to the SSL port, but it is not usually not used, unless you configure Sametime to use explicit SSL connections.</td>
</tr>
<tr>
<td>1352</td>
<td>No, the Sametime Application Service listens on this port for Lotus Notes clients and Domino servers. It is used only if you installed Sametime in an existing Domino environment.</td>
</tr>
<tr>
<td>1516</td>
<td>No, is listening on this port for other Sametime servers, if you have installed more than one.</td>
</tr>
<tr>
<td>1533</td>
<td>Yes, this is the main port. The community Service listens to direct HTTP and TCP connections.</td>
</tr>
<tr>
<td>8082</td>
<td>No, it is used for backward compatibility with previous Sametime releases.</td>
</tr>
</tbody>
</table>

Further ports can be used for features such as Meeting Services, Broadcast Services and Audio/Video Services.

Lotus Sametime supports tunneling on port 80 and for all ports except those used by Audio/Video Services. We did not use the tunneling feature for our redbook. If you need more information about this topic, refer to the Lotus Sametime online help database sthelpad.nsf in your sametime/data/doc directory.
User identity integration

In this chapter, we discuss the technical detail of changes to our sample application that would be necessary in order to satisfy our business requirements categorized under the member classification, in particular, those requirements that require our solution to integrate with an existing user identity infrastructure.

The relevant requirements are as follows.

- New users will be able to self-register in order to gain access to the ordering system if they are existing customers
- Registration requests will be validated by user organization representatives before individual user access is provided
- The primary user representative will be validated by a BASE employee before access is provided
- Single sign-on must be provided by the existing corporate solution
- User identity information must be used in existing corporate user identity repositories
11.1 Business drivers

This section of the user identity and integration chapter provides an introduction to the business drivers for customer identity infrastructure within an organization.

11.1.1 Common customer view

As with businesses offering products and services through traditional access channels, businesses that deal with customers through online channels strive to know their customer and provide him/her with a pleasurable experience that will entice the customer to come back and do business again.

To get to know their customers, many organizations construct a corporate repository that is designed as a definitive reference of customer-related information.

This definitive reference provides a common view of the customer rather than the disjointed, channel or application-specific view that is all too common. As an example, if a contact number in the repository changes as a result of a telephone conversation between a customer and a Customer Service Representative, an online application also using the corporate repository will reflect the newly corrected contact number through its online interface to the customer.

Quite simply, from the customer’s perspective, the experience of dealing with the business offering the product or service is compromised if the same information has to be provided multiple times because the business does not have a common view of that customer.

From the perspective of the organization that is offering the service or product to customers, customer-related activities are simplified if the information is held in one central location. This enables a single set of management processes and supporting tools to be constructed to support staff dealing with customer information.

In our redbook, the repository enabling a common view of the customer is known as the customer identity repository.

11.1.2 Common employee view

As they need to have a common view of customers as described in 11.1.1, “Common customer view”, so organizations also require a common view of their employees.

This, again, requires a definitive reference of information to be established that is used to retain all employee-specific information.
Although it may be possible to store both employee and customer information in one repository, organizations often opt for a separate repository for security considerations.

In our redbook, the repository enabling a common view of the employee is known as the employee identity repository.

### 11.1.3 Common customer registration process

Another consideration in the common view of customer arena is registration, that is, the process that populates the customer identity repository with customer-related information.

It simply does not make sense for each application operating with the customer identity repository to replicate existing functionality to capture customer information.

One component of the customer identity solution often employed to solve this problem is the provision of a common registration process to feed details into the customer identity repository.

In practice, the process is typically supported by an IT application constructed per access channel. With advanced technologies such as International Business Machines Corporation's WebSphere Everyplace product, it is possible to construct a single implementation to serve multiple access channels.

### 11.1.4 Common employee registration process

As there is a need for a common process for customers, as detailed in the previous section, there is also a need for a standardized approach to employee registration.

In most organizations, employee details are already captured in a human resource (HR) system. The HR system typically either doubles as the employee identity repository or triggers the creation of an entry in the employee identity repository when a new employee starts with the organization.

### 11.1.5 Step-up registration

After setting the scene with the common approach to customer and employee registration, we need to remember that there will always be an application, domain or perhaps geographic component to the registration process.

This involves capturing additional information beyond our base registration information that either increases our trust in the customer, allowing us to offer
additional services to him/her, or provides for a need to gather additional information not relevant for common use.

In our redbook, this additional component of the registration process is termed step-up registration because we step up or increase our knowledge of the customer.

11.1.6 Single sign-on

Much of the online process dealing with business customers involves the collection of customer-provided information that may be of a sensitive nature and may provide a competitive advantage to the customer’s competitors if made available to them.

The seller’s organization is likely to have other sensitive information retained in IT systems that would offer a competitive advantage to the seller’s competitors if made available inappropriately.

Even without providing a competitive edge to competitors, it would probably be damaging to an organization that deals through an online channel if a person with malicious intent were to have the ability to inappropriately modify the content or configuration of information or functionality made available via the online channel.

The business solution to this problem is the provision of a standard process and implementation solution securing IT solutions offering information and functionality to users via an online channel.

One of the common services offered by the online security infrastructure is an authentication service, that is, functionality that provides a user with a mechanism for proving he or she is the user they are claiming to be.

When looking at this from the user’s perspective, it is desirable that the user authenticate once per interactive session with the online organization. Quite simply, why should a user need to prove who they are if they wish to use an additional application during an interactive online session when they have already authenticated to use a previous application?

The solution is single sign-on. As the name suggests, if the user has signed on or authenticated to use one application, then they do not need to sign on to other applications until their session has expired.

Session expiry is normally controlled by duration or, in the case of an Internet channel access, closing the browser.
11.2 Existing environment

The integration requirements for this discussion are based on the fact that our imaginary customer, BASE Pty. Ltd., already has an existing infrastructure used for single sign-on and user registration for access to online systems.

11.2.1 Existing technologies

The existing environment is comprised of two separate LDAP compliant directory-based user stores, a registration system for customers and a second registration system for employees.

Employee registration

The employee registration system is populated and kept current by interfaces from the organization's HR system.

Customer registration

The customer registration system is a custom application that populates the customer user store with information when a user registers for access to online applications. The registration system is quite sophisticated. It is capable of sending notification to an organization's nominated representative when another user from the company registers for access to an application. It offers an approval/management function that allows that nominated representative to approve access for that other user and establish entitlements for that user. This controls what applications and functionality a user can make use of during an online session with BASE Pty. Ltd. If the registering user is a new customer, the approval notification is sent to a CSR.

Single sign-on

There are shared Web servers that provide access to the applications online users access in the BASE environment. An IBM Tivoli Access Manager for e-business, the WebSeal plugin is installed to each of the Web servers.

When a user attempts to access a resource served by the Web server, the WebSeal plugin intercepts the call in order to check the protection offered to the resource. If the resource is protected and authentication is required, the WebSeal plugin is responsible for authenticating the user.
11.3 High-level technical solution

Figure 11-1 shows a high-level view of the technical solution that would meet the needs of our customer. Due to time constraints when writing our redbook, we did not produce a more detailed specification for this solution. At the high level, we believe our approach would fit the requirements of our user and also conform to the guidelines from Patterns for e-business and our selected Sell-Side Hub composite pattern. However, this solution is technically quite complex and we recommend that you allocate sufficient resources to this area if you should encounter real world requirements at this level.

There is also a need for a Self-Service pattern to satisfy user registration for access to the ordering system. There are two existing corporate repositories within BASE that have the role of user stores and are populated as a result of user registration. One provides a common approach for customers registering for access to BASE systems and the other provides a common approach to employees registering for access to employee systems.
Once users register for access to the online order application, the information needs to be propagated from either the customer or employee user store to the WebSphere Commerce Business Edition environment.

Although it is possible for WebSphere Commerce Business Edition to operate with an external user store implemented with a compatible LDAP compliant directory, it is not possible for WebSphere Commerce Business Edition to operate with two external user stores. As such, the appropriate user information will be propagated to the WebSphere Commerce Business Edition member subsystem if a record for a WebSphere Commerce Business Edition user is created, modified or deleted from either the customer or employee user store.

The existing registration application also satisfies the need to have approval granted by a nominated customer representative before access is provided.

As the existing registration application was constructed with the possibility of needing to offer registration capability to other channels in the future, it has also been constructed with a distinct separation of responsibilities within the application. In this application, the functionality has been constructed within EJ Bs that can be seen as separate applications capable of offering their services to multiple front-end applications.
E-commerce business models

The e-commerce business models have been identified from a combination of existing business models, the evolving e-commerce industry and information obtained from IBM engagements with customers. The models are another mechanism for the classification of specific e-commerce business types.

There are currently fourteen models. The rest of this document section provides an explanation of the models. This material has been taken directly from E-commerce Business Models, January 15, 2002, Version 1.0, authored by IMAD ALBAZZ, IBM Software Group.
Model 1 - Business-to-consumer

![Diagram showing buyers, sellers, and suppliers in a business-to-consumer e-commerce model.](image)

**Model characteristics**

- Single store brand
- Owns catalog, inventory, relationship with buyers, and order fulfillment
- Brick-and-mortar extension channel or a Net Gen.
- Special implementations of this model include Government-to-Consumer/Citizen (G2C) and Business-to-Employee (B2E)
Business responsibilities

- Attracting shoppers, increasing shopper visits and purchases
- Creating and maintaining presentable product data
- Offering consistent customer service and shopping experience across online and offline touch-points

Key requirements

- Online channel consolidation with back-end business systems and processes
- Advanced capabilities for shoppers attraction and retention
- Shopper Self-Service support
- ROI measurement and profitability analysis
Model 2 - Business-to-consumer service provider

Figure 11-3  Business-to-consumer service provider e-commerce business model

Model characteristics

- Multiple stores running in a hosted environment
- Stores are totally independent
- Service provider (xSP) charges sellers a regular fee and/or surcharge based on catalog size or transaction volume
Business responsibilities

- Responding to seller's unique requirements by providing sufficient self-provisioning capabilities
- Attracting more stores
- Reducing site running cost and maintaining a profitable business

Key requirements

- Easy to use store creation and maintenance tools
- Catalog creation and management tools for the sellers
- Business analysis tools for sellers and service provider
- Service charge billing and invoicing for service provider
Model 3 - Business-to-Business (B2B) / Sell-side

![Diagram of B2B e-commerce business model]

Figure 11-4  Business-to-Business (B2B)/ Sell-side e-commerce business model

Model characteristics

- Single vendor site selling one or multiple brands
- Owns catalog, inventory, relationship with buying firms, and order fulfillment
- Spot buyers and long term partnership with buying firms
- Product selling price is determined by multiple factors like quantity and delivery location or date. It is often uniquely offered for each buying firm.
- Special implementations of this model include Business-to-Government (B2G) and Business-to-Employee (B2E)
Business responsibilities

- Managing product data from suppliers
- Managing orders and inventory
- Integration with different buyer systems and with seller's back-end applications like ERP, SCM, CRM
- Measuring business effectiveness and responding to market dynamics

Key requirements

- Dynamic pricing and contract terms customized for each buyer
- Contract management
- Account management
- Collaboration with buyers on logistics and demand forecast
- ROI measurement, profitability analysis, and contract effectiveness
Model 4 - B2B Sell-side private trading exchange/direct selling

Model characteristics

- Site owned/run by a brand seller or manufacturer
- Direct selling to business buyers
- Long term partnership with key buying firms
- Site not accessible to non-members
Business responsibilities

- Creating, managing and sharing relevant product data
- Integrating systems and business process with those of buyers
- Attracting buyer firms to join the exchange
- Business effectiveness measurement

Key requirements

- Business relationship management with buyer
- Contract management
- Account management
- Collaboration with buyers on logistics, fulfillment, and demand forecast
- Role-based access control
- ROI measurement, profitability analysis, and contract effectiveness
Model 5 - Sell-side Private Trading Exchange/Indirect selling

There are six variations or sub-models for indirect selling. Enterprises could be using a blend of these variations to work with different sizes of partners and to achieve their business objectives.

Sub-model 5A - Shared commerce

![Diagram of Shared commerce model]

**Figure 11-6  B2B Sell-side private trading - shared commerce e-commerce business model**

**Model characteristics**

- Multiple stores running in a shared environment controlled by the vendor and shared with other partners in a franchise-like model
- Each partner controls his/her own site (Private Label Site)
- Partners manage their own catalog, buyers, orders, and site
- More applicable to single-vendor partners or vendor subsidiaries
Orders, payments, etc. are segregated by partner

Some key processes, like promotions and loyalty programs, could be managed centrally by the vendor

**Business responsibilities**

- Creating, maintaining and managing customizable product data for each partner
- Integrating systems and business processes with those from partners
- Business effectiveness measurement for each partner
- Getting a good view of end-user needs, satisfaction with vendor product or partner service, and market demand forecast
- Enabling partners to add and manage their value-add products and services
- Attracting and supporting more partners
- Delivering a better partner support quality at a lower cost

**Key requirements**

- Bilateral trading relationship management
- Distributed site creation and administration for self-enabled partners
- Account management
- Sales management
- Online partner support (education, technical, etc.)
- Collaboration with partners
- Catalog creation and content management tools for partners and vendor
- ROI measurement and profitability analysis for partners and vendor
Sub-model 5B - Commerce syndication

Model characteristics

- Partner works in an “off-line” mode with irregular connection to vendor (loose integration)
- Predefined messages and formats for exchanging catalog updates, orders, etc.
- Partners manage their catalog and business offline from vendor
- Updates/refreshes on regular basis
- Same model works for mobile (offline) sales force

Business responsibilities

- Creating, managing and sharing customizable product data for each partner
- Sharing real-time inventory and order status with partners
- Business effectiveness measurement for each partner
- Getting a good view of end-user needs, satisfaction with vendor product or partner service, and market demand forecast
Enabling partners to further customize product data and add their value-add products and services

Delivering a better partner support quality at a lower cost

Key requirements

- Bilateral trading relationship management
- Account management
- Sales management
- Online partner support (education, technical, etc.)
- Controlled process to create, “syndicate”, and update product information
- ROI measurement and profitability analysis

Sub-model 5C - Commerce collaboration

Figure 11-8  B2B Sell-side private trading - Commerce collaboration e-commerce business model
Model characteristics

- Partner’s applications and/or business processes are tightly integrated with those from the vendor.
- Partner’s inventory levels visible to the vendor, allowing inventory reallocation, just-in-time inventory, and inventory-based pricing
- Partner’s orders and back orders visible to the vendor

Business responsibilities

- Creating, maintaining and managing product data which is customizable for each partner
- Streamlining internal business processes
- Integrating systems and business processes with those from partners
- Business effectiveness measurement for each partner
- Getting a good view of end-user needs, satisfaction with vendor product or partner service, and market demand forecast
- Enabling partners to add and manage their value-add products and services
- Delivering a better partner support quality at a lower cost

Key requirements

- Bilateral trading relationship management
- Real-time information to/from partners
- Role-based access control
- Account management
- Sales management
- Online partner support (education, technical, etc.)
- Collaboration with partners by sharing visibility of forecast, logistics, and inventory
- ROI measurement and profitability analysis
Sub-model 5D - Commerce brokerage

Model characteristics

- Partner's applications and business processes integrated with those from the vendor
- Partner handles final order fulfillment, billing, payment collection and any value-add service
- Orders and inventory allocation are directly controlled by the vendor

Business responsibilities

- Creating, maintaining and managing customizable product data for each partner
- Streamlining internal business processes, exposing some processes to partners
- Integrating systems and business processes with those from partners
- Business effectiveness measurement for each partner
- Getting a good view of end-user needs, satisfaction with vendor product or partner service, and market demand forecast
- Enabling partners to add and manage their value-add products and services
- Delivering a better partner support quality at a lower cost

**Key requirements**

- Bilateral trading relationship management
- Sharing buyer profiles from partners
- Role-based access control
- Account management
- Sales management
- Online partner support (education, technical, etc.)
- Collaboration with partners on logistics, planning, and forecast
- ROI measurement and profitability analysis

**Sub-model 5E - Commerce delegation**

*Figure 11-10  B2B Sell-side private trading - commerce delegation e-commerce business model*
Model characteristics

- Buyers search, configure and select products on vendor’s site.
- The lead is passed over to a partner based on product, service, geography, deal size, etc.
- Partner handles fulfillment, billing, payment collection and after-sale services.
- In some cases, the order is handled by the vendor who shares the profit with a dealer. In return, the dealer owns the after-sale support.
- Buyer profiles and vendor-controlled promotions need to be synchronized.

Business responsibilities

- Creating, maintaining and managing customizable product data for each partner
- Streamlining internal business processes
- Integrating systems and business processes with those from partners
- Business effectiveness measurement for each partner
- Getting a good view of end-user needs, satisfaction with vendor product or partner service, and market demand forecast
- Offering end users an intuitive way to define, configure, or search for required products
- Enabling partners to add and manage their value-add products and services
- Delivering a better partner support quality at a lower cost

Key requirements

- Bilateral trading relationship management
- Sharing buyer profiles and order lists to partners
- Role-based access control
- Account management
- Sales management
- Online partner support (education, technical, etc.)
- Collaboration with partners on logistics, planning, and forecast
- ROI measurement and profitability analysis
Sub-model 5F - Partners cooperation

Model characteristics

- Partner/vendor relationship is similar to that in previous models. The unique characteristic is the direct partner applications integration with other peers.
- Partners have visibility on each other’s inventory of vendor products.
- Transferred inventory is either owned by the vendor, who covers the shipping cost, or sold from partner to partner at special internal price. In some cases, the selling partner is paid a percent commission by the sourcing partner. In all cases, the selling partner owns the business relationship with the buyer.

Business responsibilities

- Creating, maintaining and managing customizable product data for each partner
- Streamlining internal business processes at the vendor and each partner, exposing necessary functions to other parties
- Integrating applications and business processes with those from partners
Business effectiveness measurement for each partner
Getting a good view of end-user needs, satisfaction with vendor product or partner service, and market demand forecast
Enabling partners to add and manage their value-add products and services and to cooperate on inventory availability
Delivering a better partner support quality at a lower cost

Key requirements
- Bilateral trading relationship management
- Role-based access control
- Account management
- Sales management
- Online partner support (education, technical, etc.)
- Collaboration with partners by sharing visibility of forecast, logistics, and inventory
- ROI measurement and profitability analysis
Business Model 6 - Business-to-Business (B2B)/Buy-side

Model characteristics

- Central procurement process for direct or indirect supplies
- Choice to buy from multiple suppliers
- Model is extended for large enterprises by adding e-sourcing capabilities for automatic supplier selection
Business responsibilities

- Reducing e-procurement and e-sourcing cost and streamlining the direct/indirect material purchasing and strategic sourcing processes
- Negotiating best prices, terms and specifications
- Ensuring uninterrupted supply of strategic direct material
- Enterprise demand aggregation

Key requirements

- Flexible and easy to use user interface
- Seamless connectivity with multiple suppliers with no supplier “switching cost”
- Catalog aggregation
- Online negotiation
- Contract management
- Demand aggregation
- Extensible catalog, often industry specific
- Approval workflow
- Spending analysis

Model characteristics

- Central procurement process for direct/indirect supplies
- Choice to buy from multiple suppliers
- Choice of buying model (spot, contract, RFQ, reverse auction)
Business responsibilities

- Reducing procurement cost
- Negotiating best prices, terms and specifications
- Ensuring supplies of strategic direct material

Key requirements

- Flexible and easy to use user interface
- Seamless connectivity with multiple suppliers for receiving product information and sending orders
- Extensible catalog that meets buyer industry requirements.
- Collaborative sourcing
Business Model 8 - B2B Buy-side Service Provider

Model characteristics

- A procurement application running in a hosted environment
- Buyers are totally independent
- Service provider (xSP) charges buyers a regular fee and/or surcharge based on catalog size or transaction volume

Business responsibilities

- Responding to buyer's unique procurement system and administration requirements.
- Attracting more buyers and suppliers
- Reducing site running cost and maintaining a profitable business
Key requirements

- Foolproof procurement application setup and administration
- Catalog creation, maintenance, and management tools for the buyers
- Business analysis tools for buyers and service provider
- Service charge billing and invoicing for service provider.
Business Model 9 - B2B Market Place

Model characteristics

- Site owned/run by a mediator (MarketMaker); participation is open for any qualified party (public)
- Multiple buyers and multiple suppliers
- Spot-buying, strategic buying and exchanges
- Standard taxonomy catalog

Business responsibilities

- Creating, managing and sharing product content
- Integrating systems and business process with those from buyers and suppliers
- Attracting buyers and suppliers
Key requirements

- Dynamic pricing (contracts, RFQs, auctions)
- Buyers and suppliers management
- Advanced order management capabilities
Business Model 10 - B2B Exchange

Model characteristics

- A network of businesses interacting directly in any-to-any model
- Each business could act as a buyer or as a supplier for the same product
Business responsibilities

- Defining and enforcing standards for data exchange
- Managing large volume transactions, complex roles and accesses control over a public network

Key requirements

- Self-enabled sellers site creation and management
- Catalog creation and content management tools for the vendor and/or sellers
- Business relationship management with sellers/partners
## Abbreviations and acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B2B</td>
<td>Business-to-Business</td>
</tr>
<tr>
<td>AAT</td>
<td>Application assembly tool</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming interface</td>
</tr>
<tr>
<td>B2C</td>
<td>Business-to-Consumer</td>
</tr>
<tr>
<td>BMP</td>
<td>bean-managed-persistence</td>
</tr>
<tr>
<td>CI</td>
<td>Corporate Identity</td>
</tr>
<tr>
<td>CICS</td>
<td>Customer Information Control System</td>
</tr>
<tr>
<td>CMP</td>
<td>container-managed-persistence</td>
</tr>
<tr>
<td>CRM</td>
<td>Customer relationship management</td>
</tr>
<tr>
<td>CSR</td>
<td>Customer Service Representative</td>
</tr>
<tr>
<td>CTS</td>
<td>Compatibility Test Suites</td>
</tr>
<tr>
<td>DMZ</td>
<td>De-militarized zone</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
</tr>
<tr>
<td>DoS</td>
<td>Denial of Service</td>
</tr>
<tr>
<td>EAI</td>
<td>Enterprise Application Integration</td>
</tr>
<tr>
<td>EAR</td>
<td>Enterprise Application Archive</td>
</tr>
<tr>
<td>ECI</td>
<td>External call interface</td>
</tr>
<tr>
<td>EDI</td>
<td>Electronic Data Interchange</td>
</tr>
<tr>
<td>EIS</td>
<td>Enterprise Information Systems</td>
</tr>
<tr>
<td>EJB</td>
<td>Enterprise JavaBean</td>
</tr>
<tr>
<td>EJB-QL</td>
<td>EJB Query Language</td>
</tr>
<tr>
<td>ENC</td>
<td>Environment naming context</td>
</tr>
<tr>
<td>ERP</td>
<td>Enterprise resource planning</td>
</tr>
<tr>
<td>ERP</td>
<td>Enterprise Resource Planning</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical user interface</td>
</tr>
<tr>
<td>HTML</td>
<td>HyperText Markup Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transfer Protocol</td>
</tr>
<tr>
<td>IBM</td>
<td>International Business Machines Corporation</td>
</tr>
<tr>
<td>IIOP</td>
<td>Internet Inter-ORB Protocol</td>
</tr>
<tr>
<td>ITSO</td>
<td>International Technical Support Organization</td>
</tr>
<tr>
<td>J2EE</td>
<td>Java 2 Platform, Enterprise Edition</td>
</tr>
<tr>
<td>J2SE</td>
<td>Java 2 Platform, Standard Edition</td>
</tr>
<tr>
<td>JAAS</td>
<td>Java Authentication and Authorization Service</td>
</tr>
<tr>
<td>JAF</td>
<td>Java Activation Framework</td>
</tr>
<tr>
<td>JAXM</td>
<td>Java API for XML Messaging</td>
</tr>
<tr>
<td>JAXP</td>
<td>Java API for XML Processing</td>
</tr>
<tr>
<td>JCA</td>
<td>J2EE Connector architecture</td>
</tr>
<tr>
<td>JDBC</td>
<td>Java database connectivity</td>
</tr>
<tr>
<td>JDK</td>
<td>Java Development Kit</td>
</tr>
<tr>
<td>JMS</td>
<td>Java Message Service</td>
</tr>
<tr>
<td>JNDI</td>
<td>Java Naming and Directory Interface</td>
</tr>
<tr>
<td>JNLP</td>
<td>Java Network Launching Protocol</td>
</tr>
<tr>
<td>JSP</td>
<td>JavaServer Pages</td>
</tr>
<tr>
<td>JTA</td>
<td>Java Transaction API</td>
</tr>
<tr>
<td>JTS</td>
<td>Java Transaction Service</td>
</tr>
<tr>
<td>LDAP</td>
<td>Lightweight Directory Access Protocol</td>
</tr>
<tr>
<td>MTBF</td>
<td>Mean Time Between Failure</td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 2003. All rights reserved.
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTTR</td>
<td>Mean Time To Recovery</td>
</tr>
<tr>
<td>NVP</td>
<td>Name value pair</td>
</tr>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure</td>
</tr>
<tr>
<td>PMS</td>
<td>product management system</td>
</tr>
<tr>
<td>RAM</td>
<td>Random access memory</td>
</tr>
<tr>
<td>RFQ</td>
<td>Request for quote</td>
</tr>
<tr>
<td>RMI</td>
<td>Remote method invocation</td>
</tr>
<tr>
<td>RMIC</td>
<td>RMI Compiler</td>
</tr>
<tr>
<td>SAR</td>
<td>Store archive file</td>
</tr>
<tr>
<td>SCM</td>
<td>Source configuration management</td>
</tr>
<tr>
<td>SKU</td>
<td>Stock keeping unit</td>
</tr>
<tr>
<td>SKU</td>
<td>stock keeping unit</td>
</tr>
<tr>
<td>WLQL</td>
<td>WebLogic Query Language</td>
</tr>
<tr>
<td>WPD</td>
<td>Work Product Description</td>
</tr>
<tr>
<td>XMI</td>
<td>XML Metadata Interchange</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Markup Language</td>
</tr>
<tr>
<td>XPath</td>
<td>XML Path Language</td>
</tr>
</tbody>
</table>
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 347.

- e-commerce Patterns for Building B2C Web Sites Using IBM WebSphere Commerce Suite V5.1, SG24-6180
- WebSphere V3.5 Handbook, SG24-6161
- WebSphere Version 4 Application Development Handbook, SG24-6134
- IBM WebSphere V4.0 Advanced Edition Handbook, SG24-6176
- WebSphere Commerce Suite V5.1 Customization and Transition Guide, SG24-6174
- WebSphere Commerce V5.4 Developer's Handbook, SG24-6190
- WebSphere Commerce Suite V5.1 Handbook, SG24-6167
- WebSphere Commerce Suite V5.4 Handbook, Architecture and Integration Guide, SG24-6567

Other resources

These publications are also relevant as further information sources:

- Grady Booch, Object-Oriented Analysis and Design with Applications, Addison Wesley, 1994, ISBN 0805353402


**Referenced Web sites**

These Web sites are also relevant as further information sources:

- Patterns for e-business

- WebSphere Commerce Suite Capacity Planning Guide

- IBM Global Service Method

- WebSphere Commerce Business Edition library

- WebSphere Application Server Version 4 FixPak3

- WebSphere Commerce Support Tools

- cXML resources - downloads

- D&B D-U-N-S Number

- WebSphere Commerce procurement integration

- E-commerce interoperability with IBM’s WebSphere Commerce products
How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.
# Index

## A
- access control 
  - procurement 250
- Access Integration patterns 5, 20, 33, 66, 74
- Account access composite pattern 24
- actors 14–15, 55, 58, 73, 173
- adapter manager 243
- Administration Console 255
- administrator 171
- Agent application pattern 27, 82
- AIX 51, 162–163, 195, 197–198
- analysis 72
- application development 219
- Application Integration patterns 5, 20–21, 28, 33, 67, 74
  - selecting 111
- application integration points 111
- Application patterns 2, 11, 25, 75
  - combined view 125
  - selection 14
- application server 4, 27, 82, 124, 159
- application standards 51
- application zone 160
- approval 45, 77, 274–275
- architecture 4, 7, 32, 170, 197
  - functional view 143
  - logical view 75–76
  - operational 76, 129
  - physical 156
  - procurement integration 240
  - reference 10
- Ariba 257, 260
- asset list 229
- assets 10
- associations 173
- auctions 5
- authentication 42, 51, 163, 251
- authentication proxy 159
- AuthenticationHelper task command 245
- authorization 42, 51
- availability 48, 162

## B
- B2B xiii, 3, 9, 18, 30, 131, 195, 220, 237, 240
- B2B Buy-side 330
- B2B Buy-side Private Trading Exchange 332
- B2B Buy-side Service Provider 334
- B2B Exchange 338
- B2B Market Place 336
- B2B patterns 63
- B2B Sell-side 316
- B2B Sell-side private trading exchange/direct selling 318
- B2C 3, 312
- Back-end integration subsystem 46
- back-end zone 161
- backup 48
- BASE 32, 71, 118, 169
  - business goals 41
  - business profile 41
  - collaboration 277
  - development estimates 229
  - existing environment 41
  - IT perspective 42
  - navigation flow 185
  - non-functional requirements 47
  - procurement 259
  - requirements 40, 75
  - screens 192
  - storyboard 188
  - styleguide 190
- BASE Pty. Ltd See BASE
- BatchOrderRequest controller command 245
- BMP 180
- build verification test 237
- bulk orders 45
- business account 269
- business description 15, 57
- business drivers 13, 25, 76, 274
  - application integration 112
  - catalog population 97
  - customer assistance 88
  - notification 93
  - orders 77
  - registration 84

© Copyright IBM Corp. 2003. All rights reserved.
technical document repository population 104
user identity 304
Business Edition
locales 199
business functions 23, 58
business goals 13, 32, 41
business logic 160
Business patterns 2, 5, 10, 13, 72, 74
review 13
selecting 16, 56, 60, 72
business policy 182
business problem xiii, 9–10, 55, 71
business processes 18
business to consumer service provider 314
Business-to-Business
See B2B
Business-to-Consumer
See B2C 3
buyer 171, 245, 255, 263
administrator 251
procurement 250
Buy-Side Hub 4, 33, 240, 256

C

caching 163
  session-dependant 163
  session-independent 163
capacity 47
catalog 5, 14, 22, 35, 67, 102, 161, 163, 171, 239–240
  aggregation 117
data 52
  population 96, 133
catalog subsystem 43, 259
CheckBatchOrderRequest task command 245
Cl 170, 185, 193
CMP 180
collaboration 35, 60, 92, 273
  directed 274
  managed 274–275
  requirements 277
  solution 277
Collaboration business pattern 16, 33, 61, 75, 87
Collaboration patterns 5
collaboration server 160
collaborative workspaces 276
colors 183
CommandProperty object 245
commands 179, 220, 232, 289
  AuthenticationHelper 245
  BatchOrderRequest 245
  CheckBatchOrderRequest 245
  controller 180, 245
  CreateBillingShippingAddress 246
  DBProcurementAuthentication 245
  flow 181
  PrepareOrder 246
  ProcessOrder 246
  procurement 245
  ProcurementOrderPrepare 246
  ProductDisplay 182
  PunchOutSetup 245
  RegisterRequisitioner 245
  SendShoppingCart 246
  SubmitShoppingCart 246
  task 180, 245
  view 180
common customer view 304
common employee view 304
component
  interactions 7
  components 15, 158, 162
    interaction 178
    procurement 242
Composite patterns 2, 4, 10, 13, 23, 69, 73
configuration 219
connectors 15
contracts 171, 240
  procurement 256
controller command 180, 245
cookies 49
Corporate Identity See CI
costs 77, 131
CreateBillingShippingAddress task command 246
credit 34
CSR 35, 46, 59, 77, 90, 187, 274, 276–277
Custom designs 2, 11
customer acceptance test 237
customer assistance 87
customer care 187, 273, 276
  customization 288
  implementation 289
  installation 295
  user registration 299
  workflows 278
Customer care subsystem 46
Customer interface  5
customer registration  160
customer satisfaction  77
Customer Service Representative See CSR
customization  7, 195, 219
customer care  288
procurement  257
cXML  46, 142, 257, 260
directed collaboration  274
Directed Collaboration application pattern  90
Directed Collaboration runtime pattern  92
directory  161
dispatcher  159
DMZ  51, 158, 162
product mapping  165
DNS  158
e-commerce  xiii, 35
e-commerce business models  311
  B2B Buy-side  330
  B2B Buy-side Private Trading Exchange  332
  B2B Buy-side Service Provider  334
  B2B Exchange  338
  B2B Market Place  336
  B2B Sell-side  316
  B2B Sell-side private trading exchange/direct selling  318
  business to consumer  312
  business to consumer service provider  314
  Commerce brokerage  325
  Commerce collaboration  323
  Commerce delegation  326
  Commerce syndication  322
  Partners cooperation  328
  Sell-side Private Trading Exchange/Indirect selling  320
  Shared commerce  320
EDI  52
  EDI client  205
  EDI container  245
Electronic commerce composite pattern  24
e-Marketplace  xiii, 4, 6, 24, 33, 69, 134, 274
e-Marketplaces  23
ERP  4, 51
Ethernet  162
Extended Enterprise  60
Extended Enterprise business pattern  5, 18, 34–35, 63, 75
failure recovery  49
features and benefits  132
filesets  201
firewalls  150
domain  158
<table>
<thead>
<tr>
<th>Term</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>protocol</td>
<td>158</td>
</tr>
<tr>
<td>flow chart</td>
<td>171</td>
</tr>
<tr>
<td>font</td>
<td>184</td>
</tr>
<tr>
<td>fulfillment</td>
<td>14, 22, 46, 48, 67, 77</td>
</tr>
<tr>
<td>functional components</td>
<td>55</td>
</tr>
<tr>
<td>functional nodes</td>
<td>13</td>
</tr>
<tr>
<td>functional requirements</td>
<td>6, 37–38, 43, 170</td>
</tr>
<tr>
<td>functional verification test</td>
<td>237</td>
</tr>
<tr>
<td>gap analysis</td>
<td>7, 195, 220</td>
</tr>
<tr>
<td>gateways</td>
<td>158</td>
</tr>
<tr>
<td>GIF</td>
<td>185</td>
</tr>
<tr>
<td>Global Service Method</td>
<td>172</td>
</tr>
<tr>
<td>goals</td>
<td>25</td>
</tr>
<tr>
<td>graphical user interface See GUI</td>
<td></td>
</tr>
<tr>
<td>GUI</td>
<td>173</td>
</tr>
<tr>
<td>guidelines</td>
<td>25, 32, 37–38, 75</td>
</tr>
<tr>
<td>hardware</td>
<td>162</td>
</tr>
<tr>
<td>prerequisites</td>
<td>199</td>
</tr>
<tr>
<td>sizing</td>
<td>156, 162</td>
</tr>
<tr>
<td>high-level summary</td>
<td>131</td>
</tr>
<tr>
<td>hosting</td>
<td>53</td>
</tr>
<tr>
<td>HR</td>
<td>161</td>
</tr>
<tr>
<td>HTML</td>
<td>52, 159, 179, 193, 229, 232</td>
</tr>
<tr>
<td>HTTP</td>
<td>142, 158, 179, 239, 243</td>
</tr>
<tr>
<td>HTTPS</td>
<td>158</td>
</tr>
<tr>
<td>IBM HTTP Server</td>
<td>152, 164, 208, 210, 216</td>
</tr>
<tr>
<td>IIS</td>
<td>164</td>
</tr>
<tr>
<td>implementation</td>
<td>xiii, 7, 9, 32</td>
</tr>
<tr>
<td>customer care</td>
<td>289</td>
</tr>
<tr>
<td>Information Aggregation</td>
<td>60</td>
</tr>
<tr>
<td>Information Aggregation business pattern</td>
<td>5, 17, 33, 62, 74–75</td>
</tr>
<tr>
<td>installation</td>
<td></td>
</tr>
<tr>
<td>customer care</td>
<td>295</td>
</tr>
<tr>
<td>DB2</td>
<td>202</td>
</tr>
<tr>
<td>DB2 client</td>
<td>205</td>
</tr>
<tr>
<td>development environment</td>
<td>233</td>
</tr>
<tr>
<td>Lotus Sametime</td>
<td>296</td>
</tr>
<tr>
<td>two-tier configuration</td>
<td>197</td>
</tr>
<tr>
<td>WebSphere Application Server</td>
<td>207</td>
</tr>
<tr>
<td>WebSphere Commerce Business Edition</td>
<td>195, 211</td>
</tr>
<tr>
<td>instance</td>
<td></td>
</tr>
<tr>
<td>WebSphere Commerce</td>
<td>213</td>
</tr>
<tr>
<td>integration</td>
<td></td>
</tr>
<tr>
<td>procurement</td>
<td>240</td>
</tr>
<tr>
<td>Integration patterns</td>
<td>2, 10, 20, 66</td>
</tr>
<tr>
<td>integration points</td>
<td>121</td>
</tr>
<tr>
<td>integration server</td>
<td>27, 160</td>
</tr>
<tr>
<td>integration verification test</td>
<td>237</td>
</tr>
<tr>
<td>interface design</td>
<td>7</td>
</tr>
<tr>
<td>Internet</td>
<td>158</td>
</tr>
<tr>
<td>Internet Explorer</td>
<td>197</td>
</tr>
<tr>
<td>interviews</td>
<td>39</td>
</tr>
<tr>
<td>intranet</td>
<td>51, 162</td>
</tr>
<tr>
<td>inventory</td>
<td>46, 48, 51–52, 67, 100, 161</td>
</tr>
<tr>
<td>invoice</td>
<td>46</td>
</tr>
<tr>
<td>IP router</td>
<td>159</td>
</tr>
<tr>
<td>iPlanet Web Server</td>
<td>164</td>
</tr>
<tr>
<td>IT drivers</td>
<td>25, 76</td>
</tr>
<tr>
<td>catalog population</td>
<td>99</td>
</tr>
<tr>
<td>customer assistance</td>
<td>89</td>
</tr>
<tr>
<td>notification</td>
<td>95</td>
</tr>
<tr>
<td>orders</td>
<td>77</td>
</tr>
<tr>
<td>registration</td>
<td>85</td>
</tr>
<tr>
<td>technical document repository population</td>
<td>106</td>
</tr>
<tr>
<td>IT environment</td>
<td>37, 75</td>
</tr>
<tr>
<td>IT goals</td>
<td>13</td>
</tr>
<tr>
<td>IT infrastructure</td>
<td>53</td>
</tr>
<tr>
<td>IT perspective</td>
<td>42</td>
</tr>
<tr>
<td>J2EE</td>
<td>40, 51, 130, 162</td>
</tr>
<tr>
<td>Java</td>
<td>179</td>
</tr>
<tr>
<td>Java bean</td>
<td>179–180</td>
</tr>
<tr>
<td>Java Servlets</td>
<td>159</td>
</tr>
<tr>
<td>JavaScript</td>
<td>179</td>
</tr>
<tr>
<td>JNDI</td>
<td>161</td>
</tr>
<tr>
<td>JPEG</td>
<td>185</td>
</tr>
<tr>
<td>compile</td>
<td>218</td>
</tr>
<tr>
<td>key file</td>
<td>216</td>
</tr>
<tr>
<td>LDAP</td>
<td>159–161, 270, 276, 307</td>
</tr>
</tbody>
</table>
load balancer 159, 162
local catalog mode 241, 246
locales 199
logging 49
loose coupling 133
Lotus Domino Web Server 164
Lotus QuickPlace 276
Lotus Sametime
  installation 296
  ports used 301

M
  mail gateway 160
  maintainability 49
  maintenance 53, 78
  manageability 49
  managed collaboration 274–275
  Managed Collaboration application pattern 95
  Managed Collaboration runtime pattern 96
  marketing reports 47
  marketing subsystem 47
  Mean Time Between Failure See MTBF
  Mean Time To Recovery See MTTR
  member groups 266
  member subsystem 44, 258
  message mapper 244
MTBF 49
MTTR 49

N
  navigation 177
  navigation flow 171, 177
    sample 185
Netscape 197
network 162
node 27, 82
  application server 160
  authentication proxy 159
  collaboration server 160
  commerce server 160
  customer registration 160
  database server 161
directory 161
DNS 158
domain firewall 158
HR 161
integration server 160
inventory 161
load balancer 159
mail gateway 160
PKI 158
PMS 161
pricing 161
product management system 161
protocol firewall 158
redirector 159
security services 161
technical documentation repository 160
Web server 159

non-functional requirements 6, 14, 37–38, 47, 75, 170
notification 92, 123

O
  OCI 46, 241, 257, 260, 270
  operational architecture 129
  operational overview xiii, 6
  operational preparedness 52
Oracle 163
order subsystem 45
orders 5, 14, 22, 67, 77–78, 82, 120, 240
  approval 45, 274
  bulk 45
  business drivers 77
  IT drivers 77
  procurement integration 246
  purchase 260
  standing 45
  status 46
  subsystem 257
template 45	onganization 171, 245, 255
outside world 158

P
  patterns 10
    access integration 5, 20, 74
    application 2, 11, 14, 25, 75
    application integration 5, 20–21, 74, 111
    applying 6, 9
    business 2, 10, 13, 16, 72, 74
    collaboration 5, 16, 75
    composite 2, 4, 10, 13, 23, 69
    extended enterprise 5, 18, 75

Index 353
information aggregation 5, 17, 74–75
integration 2, 10, 20, 66
runtime 2, 6, 10–11, 27, 76
self service 5
Self-Service 16, 74
Patterns for e-business xiii, 2, 9–10, 23, 32, 55, 60, 71, 75
applying 6
methodology 170
performance 47
persistence 180, 245
personalization 163
PKI 52, 151, 158
planning
development 220
staff 220
testing 236
plugin 160, 179
PMS 42, 46, 51–52, 67, 100, 161
policies 25, 32, 37–38, 50, 75
Population-Multi Step application pattern 101, 107
Population-Multi Step runtime pattern 101
portals 24
Portals composite pattern 24
ports 198, 301
preferences 124
PrepareOrder task command 246
prerequisites
hardware 199
software 199
presentation 160
pricing 46, 48, 51–52, 67, 77, 100, 161
contracts 240
procurement 256
principles 38, 75
privacy 301
process 9–10, 55
processes 15
ProcessOrder task command 246
procurement 64, 239
access control 250
adapter manager 243
authentication and registration 251
buyer 250
buyer organization 255
commands 245
components 242
configuring WebSphere Commerce Business
Edition 252
contracts 256
customization 257
integration 240
local catalog mode 241
message mapper 244
modes 241
one-step mode 241
order flows 246
pricing 256
program adapter 244
protocols 252
punch-out catalog mode 241
reference application 262
request servlet 243
requirements 259
samples 259
servlet engine 243
solution overview 260
two-step mode 241
ProcurementOrderPrepare controller command 246
product configuration 82
product configurator 51
product management system See
PMS
product mapping 6, 11, 29, 155, 164, 196
application zone 166
back-end zone 166
ProductDisplay command 182
products 14, 46, 77, 120
search 82, 171
program adapter 239, 243–244
project phases 143
proposal 9, 129
component interactions 153
component list 144
component relationships 153
dependencies 137
enabling technologies 133
features and benefits 132
high level summary 131
key aspects 132
overview 132
product suite 143
project phases 143
requirements 137
risks 136
scope 142
solution 134
pSeries 162
Public Key Infrastructure See PKI
punchout 239
punch-out catalog mode 241, 248
PunchOutSetup command 245
purchase order 260
PurchaseOrderRequest message 246

Q
questionnaires 39
quote 246, 260

R
Rational ClearCase 51, 232
Redbooks Web site 347
  Contact us xvi
redirector 159
redundancy 162
reference application 262
reference architectures 10
RegisterRequistioner task command 245
registration 83, 124, 251
  common process 305
reports 47
repository database 205
request for proposal 32, 77
request servlet 243
requirements xiii, 2, 6, 9, 13, 25, 32, 35, 37, 55, 72, 75–76, 137, 169
  Back-end integration subsystem 46
  catalog subsystem 43
  collaboration 277
  customer care 187
  Customer care subsystem 46
  functional 6, 37–38, 43, 170
  gathering 39
  marketing subsystem 47
  member subsystem 44
  non-functional 6, 14, 37–38, 47, 75, 170
  order subsystem 45
  procurement 259
  user identity 303
risks 138
roles
  testing 236
Router application pattern 27, 82
Router runtime pattern 82
routers 158
runtime
  WebSphere Commerce Business Edition 196
runtime architecture 71
Runtime patterns 2, 6, 10–11, 27, 76
  combined view 128
samples xiii, 185
  applying patterns 12
  customer care 278
  navigation flow 185
  procurement 259
  publish sample store 217
  screens 192
  storyboard 188
  styleguide 190
  ToolTech 192, 218
  use cases model 187
scalability 49, 159, 162
schedule 220, 231
scope 142
screens 170, 177, 184
  layout 183
  purpose 184
  samples 192
search 78, 82, 171
SecureWay Directory 161
SecureWay Directory Server 164
security 48, 159, 161–162
Self-Service 16, 60, 64, 77, 120, 131
Self-Service business pattern 5, 16, 33, 61, 74, 83, 239
Self-Service business patterns 78
Sell-Side Hub 4–5, 33, 74–75, 240, 256
Sell-Side Hub composite pattern xiii, 1, 3–5, 9, 33, 69, 72–73, 127, 274
  applying 6
Sell-side Private Trading Exchange/Indirect selling 320
SendShoppingCart task command 246
servlet engine 243
shipping 34
shopping cart 14
single sign-on 77, 108, 270, 306
skills 220, 229
SKU 52
software 163
solution 11–12, 55
solution design 4
solution overview 14, 22
Business patterns 20
connectors 15
diagram 15
procurement 260
solution overview diagram 55, 58, 64, 68
source code version control system 51
source control 232
specifications 7, 38
SQL 102, 179
SSL 151, 158, 216
staffing 220
staffing plan 230
staging 53
Stand-Alone Single Channel application pattern 87
standards 25, 32, 37–38, 50, 75
application 51
business structure 50
platform 50
technical 51
standing orders 45
starting point 32
store archive 217
store developer 180
Store Services 232
stored procedures 102
storyboard 7, 176, 188
notation 177
purpose 176
stress test 237
styleguide 170, 183, 190
colors 183
element positioning 184
purpose 183
screen layout 183
sizing 184
typographical definitions 184
SubmitShoppingCart task command 246
subsystems
member 258–259
orders 257
supplier 263
support 53
system overview xiii
system proposal 6
system verification test 237
systems management 48

T
task command 180
tasks
command 180
development 229
view 180
TCP/IP 158
technical documentation repository 42, 48, 51, 103, 160
integration 119
population 103
template orders 45
termination outcome 176
test cases 173
testing 7, 195, 235
build verification 237
categories 236
customer acceptance 237
design 236
environment 53
environments 238
functional 237
integration 237
phases 236
planning 236
roles 236
stress 237
system 237
test cases 173
unit 237
Tivoli 49, 51
Tivoli Access Manager 164
tools
development 195, 232
WebSphere Commerce Studio 232
ToolTech 185, 192, 218, 220, 229
Trading Exchange 33
trading exchange 4
transactions 16
Two-tier configuration 197

U
UDB 163
UML 153
unit tests 237
URL 123, 158
URL rewriting 163
URLREG 289
usability 49
use cases 7, 14, 172–173
associations 176
customer care 187
model 172–173, 187
overview 175
preconditions 175
termination outcome 176
user 158, 171, 245
user groups 163
user identity 303
users 13, 77, 299
registration 299
user-to-business pattern 61
User-to-Data 17
user-to-user pattern 61

V
validation 182
view command 180
view task 180
VisualAge for Java, Enterprise Edition 232

W
Web controller 243, 245
Web server 42, 48, 152, 159, 162, 164, 179
WebSphere 51
WebSphere Application Server 40, 130, 161–162, 178
database 205
installation 207
WebSphere Application Server, Advanced Edition 133, 163
WebSphere Commerce 53, 161, 163–164, 178–179
create instance 213
JSP 218
WebSphere Commerce Accelerator 276
WebSphere Commerce Business Edition xiii, 1, 100, 121, 127, 133, 163, 195, 197
installation 195, 211
runtime 196
WebSphere Commerce Studio 232
WebSphere MQ 42, 103, 118, 121, 126, 133, 162,
164
WebSphere MQ Integrator 118, 126
WebSphere Payment Manager 164, 197
WebSphere Test Environment 232, 237
Work Product Description See WPD
workshop 13, 39
WPD 172

X
XCBL 257
XML 52, 100, 142, 239–240, 243
XML message mapper 244
XML messages over HTTP 240
B2B e-commerce with WebSphere Commerce Business Edition V5.4
Patterns for e-business Series

This IBM Redbook is part of the Patterns for e-business series and examines how Composite patterns can be used when developing e-commerce Web sites using WebSphere Commerce Business Edition. In particular, it describes the Sell-Side Hub composite pattern and how it can assist architects and designers who have been asked to develop an e-Marketplace using WebSphere Commerce Business Edition.

This redbook is intended for architects, designers and developers. It develops a technical architecture based on requirements common to e-Marketplaces that are Sell-Side Hubs. In developing this architecture, the redbook envisages an imaginary customer with realistic requirements, then follows a representative project development life cycle for this customer. The steps documented in this redbook include the following:

- Requirements gathering
- Understanding the business problem
- System overview
- Operational overview
- Detailed design
- Implementation
- Implementation samples.

For more information:
ibm.com/redbooks