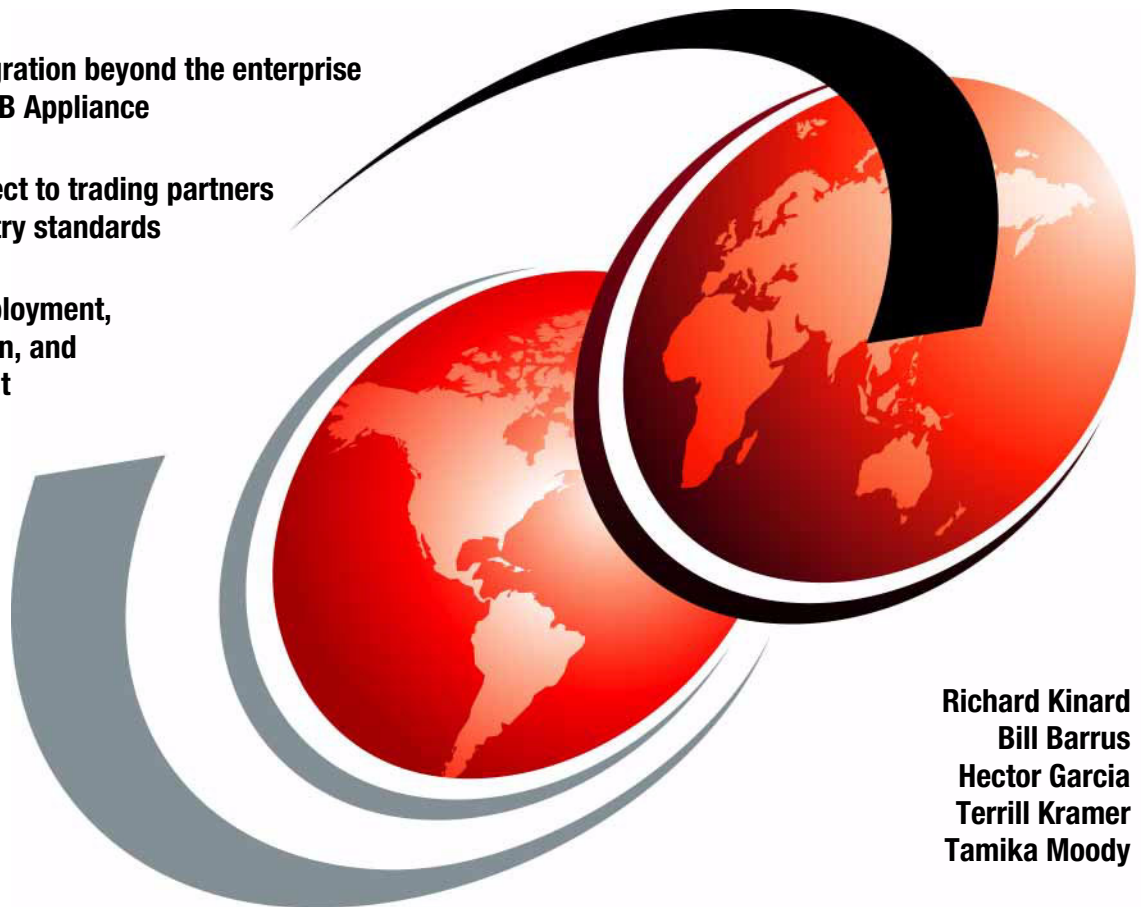


IBM WebSphere DataPower B2B Appliance XB60 Revealed

Extend integration beyond the enterprise
with IBM B2B Appliance

Easily connect to trading partners
using industry standards

Simplify deployment,
configuration, and
management



Richard Kinard
Bill Barrus
Hector Garcia
Terrill Kramer
Tamika Moody



International Technical Support Organization

**IBM WebSphere DataPower B2B Appliance XB60
Revealed**

April 2009

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

First Edition (April 2009)

This edition applies to Version 3.7.3 of IBM WebSphere DataPower B2B Appliance XB60.

© Copyright International Business Machines Corporation 2009. All rights reserved.

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

Contents

| | |
|--|-----|
| Notices | ix |
| Trademarks | x |
| Preface | xi |
| The team that wrote this book | xii |
| Become a published author | xiv |
| Comments welcome | xiv |
| Part 1. Introduction to business to business integration (B2Bi) | 1 |
| Chapter 1. Business-to-business concepts | 3 |
| 1.1 Impact of the Internet on business applications | 4 |
| 1.2 E-commerce | 4 |
| 1.2.1 Business-to-consumer | 5 |
| 1.2.2 Business-to-business | 5 |
| 1.2.3 Evolution of the B2B data structures | 7 |
| 1.2.4 Evolution of B2B data communications | 9 |
| 1.3 Enterprise application integration and B2B | 10 |
| 1.4 B2B integration | 11 |
| 1.4.1 Types of B2B integration | 13 |
| 1.4.2 Summary | 19 |
| Chapter 2. B2B technologies and standards | 21 |
| 2.1 Requirements for a B2B solution | 22 |
| 2.2 Terminology | 23 |
| 2.2.1 Messaging and queuing | 23 |
| 2.2.2 Electronic data interchange | 25 |
| 2.2.3 Transport protocols | 26 |
| 2.2.4 Security | 27 |
| 2.2.5 Extensible Markup Language | 29 |
| 2.2.6 Electronic Business using Extensible Markup Language | 29 |
| 2.2.7 Web services | 30 |
| Chapter 3. B2B deployment methodology | 31 |
| 3.1 B2B deployment planning | 32 |
| 3.2 B2B deployment methodology overview | 32 |
| 3.2.1 Knowledge transfer and training | 33 |
| 3.2.2 Discovery | 34 |
| 3.2.3 Planning | 35 |

| | | |
|-------|---|-----------|
| 3.2.4 | Installation | 38 |
| 3.2.5 | Customization | 39 |
| 3.2.6 | Testing | 41 |
| 3.2.7 | Production Deployment | 42 |
| 3.2.8 | Partner Ramping | 43 |
| 3.2.9 | Support | 44 |
| 3.3 | Time estimates | 45 |
| 3.4 | Partner Ramping Effort Estimator (hours per Partner) | 46 |
| 3.5 | Roles | 48 |
| | Chapter 4. Aspects of B2B security | 51 |
| 4.1 | Overview | 52 |
| 4.2 | Areas of B2B security | 52 |
| 4.2.1 | Deployment security | 52 |
| 4.2.2 | Connection security | 53 |
| 4.2.3 | Document security | 55 |
| 4.2.4 | Access control | 56 |
| 4.3 | Security technologies | 56 |
| 4.3.1 | Reverse proxy server | 57 |
| 4.3.2 | Firewalls | 57 |
| 4.3.3 | Network Address Translation | 58 |
| 4.3.4 | Port Address Translation | 59 |
| | Chapter 5. WebSphere DataPower B2B Appliance XB60 | 61 |
| 5.1 | Why an Appliance for B2B | 62 |
| 5.1.1 | SOA appliances simplify SOA deployment | 63 |
| 5.1.2 | Drop-in integration for heterogeneous environments | 63 |
| 5.1.3 | Innovative enablement of existing infrastructure for XML and Web services | 63 |
| 5.1.4 | Policy-driven approach to Web services management and SOA governance | 64 |
| 5.1.5 | Integration with registry and repository, security, identity, and service management software | 64 |
| 5.1.6 | Support for advanced Web services standards and interoperability | 64 |
| 5.1.7 | IBM SOA Foundation for Smart SOA deployments integration | 65 |
| 5.2 | Easily connect to trading partners using industry standards | 65 |
| 5.2.1 | IBM WebSphere DataPower B2B Appliance XB60 | 65 |
| 5.2.2 | How Data flows through the B2B Gateway Service | 67 |
| | Part 2. Getting started with the XB60 | 73 |
| | Chapter 6. Device setup and administrative tasks | 75 |
| 6.1 | Initializing the device | 76 |
| 6.2 | Defining the base configuration | 77 |

| | | |
|---|---|------------|
| 6.2.1 | Startup method | 78 |
| 6.2.2 | Manual procedure | 81 |
| 6.2.3 | Verifying the configuration | 83 |
| 6.2.4 | Checking and managing storage | 83 |
| 6.3 | Domains, groups, and users | 87 |
| 6.3.1 | Domains | 87 |
| 6.3.2 | Specifying access control | 88 |
| 6.4 | Backing up the appliance | 89 |
| Chapter 7. B2B configuration options | | 91 |
| 7.1 | XB60 B2B services | 92 |
| 7.1.1 | B2B Partner Profiles | 92 |
| 7.1.2 | B2B Gateway Service | 99 |
| 7.1.3 | B2B Transaction Viewer | 106 |
| 7.2 | Transaction Viewer examples using RBM | 111 |
| 7.2.1 | XML Management Interface | 111 |
| 7.2.2 | Command line interface (CLI) | 111 |
| 7.2.3 | WebGUI interface | 111 |
| 7.2.4 | Working with transactions in the B2B Viewer | 116 |
| 7.3 | B2B Data Persistence | 118 |
| 7.3.1 | Transaction store | 119 |
| 7.3.2 | Document storage | 120 |
| 7.3.3 | Monitoring hard drive space | 131 |
| Chapter 8. Configuration management | | 133 |
| 8.1 | Configuration management | 134 |
| 8.1.1 | File system directories and domains | 134 |
| 8.1.2 | Startup sequence for DataPower | 135 |
| 8.2 | Configuration options | 137 |
| 8.2.1 | WebGUI interface | 137 |
| 8.2.2 | Command line interface | 138 |
| 8.2.3 | XML Management Interface | 139 |
| 8.3 | Role Based Management (RBM) | 146 |
| 8.4 | Package importing and exporting | 148 |
| Chapter 9. Troubleshooting the appliance | | 151 |
| 9.1 | Overview | 152 |
| 9.2 | Troubleshooting the network setup | 152 |
| 9.2.1 | Ping and TCP Connect | 152 |
| 9.2.2 | Packet Capture | 153 |
| 9.3 | Using built-in tools to diagnose appliance problems | 154 |
| 9.3.1 | Using the B2B Transaction Viewer | 155 |
| 9.3.2 | Checking the appliance status | 155 |
| 9.3.3 | Checking the system log | 155 |

| | | |
|--|--|------------|
| 9.3.4 | Checking the audit log | 157 |
| 9.3.5 | Checking the Object Status | 158 |
| 9.3.6 | Generating an error report | 158 |
| 9.4 | XB60 firmware level 3.7.3 limitations and known problems | 161 |
| 9.5 | Common B2B XB60 configuration mistakes | 161 |
| 9.5.1 | The hard disk array is unresponsive or down | 161 |
| 9.5.2 | B2B Gateway is unresponsive (down) | 161 |
| 9.5.3 | B2B Transaction Viewer not visible to partners | 162 |
| 9.5.4 | B2B Gateway not sending MDNs as expected | 162 |
| 9.5.5 | Binary documents are not routed properly | 163 |
| 9.6 | Life cycle considerations | 166 |
| 9.7 | Getting help and technical assistance | 167 |
| Part 3. B2B patterns and service-oriented architecture (SOA) integration | | 169 |
| Chapter 10. XB60 and WTX integration for HIPAA | | 171 |
| 10.1 | Business value | 172 |
| 10.2 | Prerequisites: Technical and infrastructure | 174 |
| 10.2.1 | Software prerequisites | 174 |
| 10.2.2 | Skills prerequisites | 175 |
| 10.3 | Presenting the scenario | 175 |
| 10.3.1 | The health care claim: Inbound flow | 176 |
| 10.3.2 | The claim payment: Outbound flow | 177 |
| 10.4 | Scenario solution | 177 |
| 10.4.1 | Scenario outline | 177 |
| 10.4.2 | Scenario implementation | 179 |
| 10.5 | Testing our solution | 214 |
| 10.5.1 | Inbound flow | 214 |
| 10.5.2 | Outbound flow | 218 |
| Chapter 11. XB60 with transformation | | 225 |
| 11.1 | Business value | 226 |
| 11.2 | Prerequisites | 226 |
| 11.2.1 | Software prerequisites | 226 |
| 11.2.2 | Skills prerequisite | 227 |
| 11.3 | Presenting the scenario | 227 |
| 11.4 | Scenario solution | 228 |
| 11.4.1 | Scenario outline | 229 |
| 11.4.2 | Scenario implementation | 231 |
| 11.5 | Testing our solution | 266 |
| 11.5.1 | Test results | 269 |
| Chapter 12. Trading outbound binary documents using the B2B Gateway Service | | 273 |

| | |
|--|------------|
| 12.1 Business value | 274 |
| 12.2 Prerequisites | 274 |
| 12.2.1 Software prerequisites | 274 |
| 12.2.2 Skills prerequisites | 275 |
| 12.3 Presenting the scenario. | 275 |
| 12.4 Scenario solution. | 276 |
| 12.4.1 Scenario outline | 277 |
| 12.4.2 Scenario implementation. | 278 |
| 12.5 Testing our solution. | 318 |
| 12.5.1 Test results | 319 |
| | |
| Chapter 13. Trading binary documents using a Multi-Protocol Gateway service | 323 |
| 13.1 Business value | 324 |
| 13.2 Prerequisites | 324 |
| 13.2.1 Software prerequisites | 324 |
| 13.2.2 Skills prerequisites | 324 |
| 13.3 Presenting the Binary AS2 over HTTP multi-step use case | 325 |
| 13.4 Binary AS2 over HTTP multi-step use case solution. | 326 |
| 13.4.1 Use case outline | 327 |
| 13.4.2 Use case implementation | 328 |
| 13.5 Testing the Binary AS2 over HTTP multi-step use case | 345 |
| 13.6 Presenting the binary FTP multi-step use case. | 349 |
| 13.7 Binary FTP multi-step use case solution. | 350 |
| 13.7.1 Use case outline | 351 |
| 13.7.2 Use case implementation | 351 |
| 13.8 Testing the binary FTP multi-step use case | 358 |
| | |
| Chapter 14. Handling SOAP Messages with Attachments in a B2B environment. | 363 |
| 14.1 Business value | 364 |
| 14.2 Prerequisites | 364 |
| 14.2.1 Software prerequisites | 364 |
| 14.2.2 Skill prerequisites | 365 |
| 14.2.3 Infrastructure prerequisites | 365 |
| 14.3 Presenting the scenario. | 365 |
| 14.4 Scenario solution. | 367 |
| 14.4.1 Scenario outline | 367 |
| 14.4.2 Scenario implementation. | 369 |
| 14.5 Testing our solution. | 380 |
| 14.5.1 Test results | 383 |
| | |
| Part 4. Appendixes | 387 |

| | |
|--|-----|
| Appendix A. Additional material | 389 |
| Locating the Web material | 389 |
| Using the Web material | 389 |
| How to use the Web material | 390 |
| Abbreviations and acronyms | 391 |
| Related publications | 393 |
| IBM Redbooks publications | 393 |
| Other publications | 393 |
| Online resources | 394 |
| How to get IBM Redbooks publications | 394 |
| Help from IBM | 394 |
| Index | 395 |

Notices

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

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

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

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

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

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

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

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

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


COPYRIGHT LICENSE:

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

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

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

| | | |
|-------------------|-----------|---|
| AIX® | HACMP™ | Redbooks (logo)  ® |
| DataPower device® | IBM® | System z® |
| DataPower® | Lotus® | Tivoli® |
| DB2® | Rational® | WebSphere® |
| developerWorks® | Redbooks® | |

The following terms are trademarks of other companies:

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

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

RPM, and the Shadowman logo are trademarks or registered trademarks of Red Hat, Inc. in the U.S. and other countries.

Java, JVM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

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

Preface

This IBM Redbooks publication was developed as a guide for anyone who is interested in deploying business-to-business (B2B) integration solutions utilizing purpose-built appliances. The book is divided into three parts.

Part 1 of this book gives you a brief introduction to B2B. It contains five chapters, each covering a specific area of interest related to B2B integration:

- ▶ Chapter 1 presents an introduction to B2B concepts.
- ▶ Chapter 2 discusses the most common B2B technologies and standards.
- ▶ Chapter 3 describes IBM® B2B Deployment Methodology.
- ▶ Chapter 4 describes the various types of security to use when deploying B2B.
- ▶ Chapter 5 provides an overview of the DataPower® B2B Appliance.

Part 2 of this book gives you a good overview of how to configure specific B2B function in the XB60 and contains chapters describing configuration information, performance testing, and troubleshooting:

- ▶ Chapter 6 describes device setup and common administrative tasks.
- ▶ Chapter 7 discusses configuration management of the XB60.
- ▶ Chapter 8 demonstrates configuration options specific to the XB60.
- ▶ Chapter 9 presents common troubleshooting tips.

Part 3 of this book provides you with five common B2B scenarios and demonstrates how each scenario was completed using the XB60:

- ▶ Chapter 10 demonstrates how to use the XB60 in front of WebSphere® Transformation Extender for processing Health Insurance Portability and Accountability Act (HIPAA) transactions.
- ▶ Chapter 11 demonstrates how to use the XB60 to receive documents from a trading partner and then transform the document on the XB60.
- ▶ Chapter 12 and 13 provide examples of how you can trade binary documents (no partner information in the document) using the XB60.
- ▶ Chapter 14 demonstrates how you can use Web Services to transport B2B documents and then use protocol bridging to route the document to a WebSphere MQ queue.

The team that wrote this book

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

Richard Kinard is the Product Manager for WebSphere DataPower B2B Appliances. He is a subject matter expert in B2B technologies and has over ten years of experience in designing, developing, and implementing B2B solutions. He has worked on many initiatives with Internet standards organizations to promote B2B interoperability and was a Senior Product Manager of a very successful B2B application prior to working for IBM.

Bill Barrus is a Senior Software Engineer in IBM Software Group Business Partner Technical Strategy and Enablement organization. He began his IBM career performing mechanical design trade-off studies for the U.S. Navy F-14 avionics upgrade program, moved into mechanical computer-aided engineering software, and most recently contributed to emerging business opportunity challenges in support of CATIA Engineering Analysis, Lotus® Workplace Client Technology - Micro Edition, and IBM WebSphere Voice Server. He is currently involved in providing consultation and support to IBM partners on various IBM products, including DataPower Service-Oriented Architecture (SOA) Appliances.

Hector Garcia works as an IT Specialist in the IBM Software Services for WebSphere organization, mainly focusing on DataPower and the enterprise service bus (ESB) layer of the service-oriented architecture. He holds a Physics Bachelors degree from University of Vigo, where he graduated with honors. He was selected for IBM Spain as part of a graduate program and has been involved in several DataPower and SOA engagements since then. Originally from Galicia, he is now based in Madrid.

Terrill Kramer is a Managing Consultant in IBM Software Services for WebSphere. He has 15 years of experience in the IT industry. He holds a Bachelor of Science degree in Computer Science from Kennesaw State University. His areas of expertise include DataPower SOA Appliances and SOA applications. He has been a DataPower Consultant for the past two years.

Tamika Moody is a WebSphere Business Integration Message Broker/WebSphere DataPower Consultant and IT Specialist for IBM. She has over seven years of experience in the IT integration area. Tamika has broad experience in leading middleware engagements ranging from electronic data interchange (EDI) and B2B implementations to design, implement, and determine problems for DataPower and IBM middleware solutions.

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

Chris Rayns
International Technical Support Organization, Raleigh Center

Ronan Dalton, IBM Software Group, WPLC
IBM Ireland

Mac Devine, IBM Distinguished Engineer, Master Inventor
IBM US

Neal Alewine, STSM; SWG Voice Architect, DataPower B2B Software Architect
IBM US

Matt McLarty, Worldwide Technical Sales Manager
IBM Canada

Andre Manriquez, B2B Application Integration Specialist
IBM US

Ken McCauley, Middleware Software WebSphere Services
IBM US

Kyle G. Brown, Distinguished Engineer, SOA and Emerging Technologies
IBM US

Joy Howard, WebSphere Product Marketing Manager
IBM US

Marc-Thomas Schmidt, Distinguished Engineer; Chief Architect SOA
Connectivity
IBM US

Sudhir (Sid) Bhatia, Manager, WebSphere Connectivity Product Management
IBM US

David Maze, Senior Software Engineer, DataPower XML Technologies
IBM US

Jeremy N Shapiro, DataPower Security Software Development
IBM US

F Hackerman, Software Developer, AIM
IBM US

Scott Norris, Application Integration and Middleware Solutions Specialist
IBM US

Mario De Armas, Software Developer, DataPower Web Technologies
IBM US

Become a published author

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

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us.

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

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

ibm.com/redbooks

- ▶ Send your comments in an e-mail to:

redbooks@us.ibm.com

- ▶ Mail your comments to:

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



Part 1

Introduction to business to business integration (B2Bi)

Part 1 of this book gives you a brief introduction to B2B. It contains five chapters, each covering specific areas of interest related to B2B integration:

- ▶ Chapter 1 presents an introduction to B2B concepts.
- ▶ Chapter 2 discusses the most common B2B technologies and standards.
- ▶ Chapter 3 describes IBM B2B Deployment Methodology.
- ▶ Chapter 4 describes the various types of security to use when deploying B2B.
- ▶ Chapter 5 provides an overview of the DataPower B2B Appliance.



Business-to-business concepts

This chapter presents an overview of business-to-business (B2B), including its nature, and its evolution over time. It also discusses the concept of Enterprise Application Integration (EAI), which is commonly confused with or mistaken for B2B. It compares EAI and B2B by examining the similarities and differences that make them separate concepts and processes.

1.1 Impact of the Internet on business applications

At the beginning of the Internet era, IBM invented the term *e-business* to give a name to a new class of powerful software applications and services that, in its vision, needed to be developed in the following years. This class of applications derives its power from combining the universal access and standards of the Internet with the reliability, security, and availability of existing content, core business processes, and applications.

In simplified terms, e-business refers to the use of Internet technologies to improve and transform key enterprise processes. Most organizations understand this concept and have begun the transformation from traditional applications to their e-business counterparts. This transformation has begun to Web-enable core processes to strengthen customer service operations, streamline supply chains, and reach existing and new customers.

e-business affects virtually every industry. The pace might vary, but the impact is still being felt. Industry players need to at least consider the changes that e-business will have on their industry in general and their company in particular. Those “out in front” might face more risks, but they also harvest the rewards of creating and sustaining real competitive advantage. Those practicing a wait-and-see strategy might not get locked out of the game, but they will at best run with the pack.

Even with the fall of dot-com companies, most companies still recognize the need to at least take steps down the path to becoming an e-business company. Probably one of the best-known applications of e-business is *e-commerce*, which refers to buying and selling activities over a digital medium. However, e-business embraces e-commerce and includes intranet applications.

Note: e-business, now referred to as *On Demand Business*, is a broad concept and can affect nearly all aspects of your business. It is the overall strategy, where e-commerce is an extremely important subset of e-business.

1.2 E-commerce

The world of e-commerce is changing rapidly. Ten years ago *e-commerce* was mostly defined as participating in an *electronic data interchange* (EDI) initiative. Today, e-commerce means much more than just EDI. It means supporting interactive Web sites; it means enabling the communications with multiple exchanges; it means using XML and the Internet to conduct interactive business-to-customer (B2C) and B2B communications.

Note: E-commerce can be divided into two main subclasses: B2C and B2B.

Every business's activities, by definition, pertain to goods and services. Those activities can be divided as to whether they involve consumption, creation, transformation, or provision of goods and services, a combination of these activities, or the management of such activities. Any of these types of activities can involve external business entities (including customers or consumers). For somewhat historical reasons, special emphasis has been given to B2B activities (those that involve other businesses) and to B2C activities.

This definition does not introduce any technology. The definition of B2B and B2C is first and foremost a business issue. The B2B and B2C classifications pertain to activities by which the business interacts with external entities. Certain types of business, such as pure B2B exchanges, wholesalers, or distributorships, might primarily conduct B2B activities. These businesses derive their revenues from other businesses. All the business activities of such companies either are B2B activities or support them, which has important implications for B2B success. Although it is common to think of B2B as being implemented by public (external) business processes, virtually every private (internal) business process is an essential element to the support of B2B activities.

1.2.1 Business-to-consumer

The B2C e-commerce model is a publicly accessible Web site that offers products for sale. It is analogous to a store on the street, where anyone can walk in and make a purchase. A new, unknown customer is known as a *guest shopper*. The guest shopper has the option of making purchases, after providing general information about themselves to fulfill the transaction (name, address, credit card, and so on). Most B2C sites encourage users to register and become members. In doing so, the business can establish a relationship with the customer, provide better service, and build customer loyalty.

1.2.2 Business-to-business

The B2B e-commerce model refers to an e-commerce store specifically designed for organizations to conduct business over the Internet. B2B applications focus on using the Internet, extranet, or both to improve B2B partnerships and transform inter-organizational relationships. The two entities are known to each other, and all users are registered. Trading can be conducted directly between buyers and sellers or supported by a third-party (an intermediary) within an e-Marketplace.

There are two styles of B2B:

- ▶ Business-to-marketplace-to-business (B2M2B; e-Marketplaces)
- ▶ Business-to-business integration (B2Bi)

Figure 1-1 shows the breakdown between e-Marketplaces and B2Bi based on the number of buyers and sellers. In the case of B2Bi, there is usually a one-to-one relationship between buyers and sellers. Any other relationship, such as one-to-many, many-to-many, or many-to-one, falls into the e-Marketplace category.

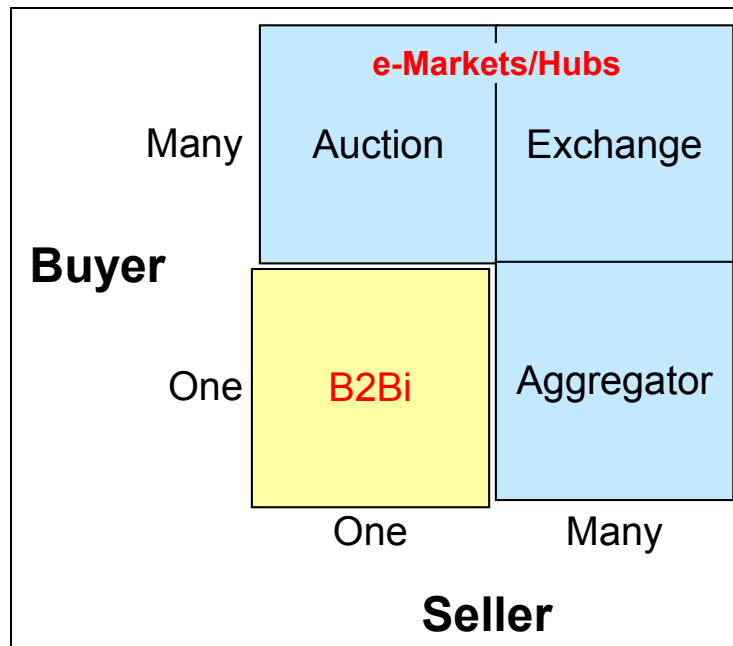


Figure 1-1 Relationship between B2Bi and e-Marketplaces

Each of the e-Marketplace categories has its own unique characteristics, which are reflected in their implementation. For example, in the case where there is one seller and many buyers, the interaction is similar to that of a traditional auction. Therefore, the digital model has to allow for multiple disclosed bids.

Business-to-business integration

B2Bi covers programmatic links between arm-length businesses, between companies where a trading partner agreement might be appropriate. A good example is supply chain applications or trading partners that engage in an exchange.

The remainder of this book focuses on B2Bi. For simplicity in this book, from here forward, we call B2Bi simply B2B.

Business-to-marketplace-to-business

The second style covers the e-Marketplace where the model supports B2M2B. The M represents the *e-Marketplace*, which supports multiple buyers and suppliers. The buying function can be performed online or programmatically.

The traditional B2B model, centered around the buyer-seller transaction paradigm, shows its limitations. It is definite in scale and displays only partial efficiency in terms of market economics. B2M2B overcomes these limitations and leverages existing B2B applications and technology. The e-Marketplace or online trading communities assist multiple buyers and suppliers to exchange information and transactions.

Trading communities are Internet-based hubs that focus on specific industry verticals or specific industry processes. They use various market making mechanisms, such as auctions, exchanges, and aggregation, to mediate any-to-any transactions among businesses. Through the trading communities, hubs, buyers, and sellers can trade electronically with established partners and at the same time gain access to new markets and new parts of the supply chain.

e-Marketplaces can be a public, interactive buying and selling community. Here all members participate in the open. Or, they can be private, invitation-only communities whose members participate in special pricing arrangements or product and service offerings. Online trading communities have the potential to create excellent and efficient markets.

1.2.3 Evolution of the B2B data structures

The structure of information exchanged electronically between businesses has evolved over time. This evolution was basically an evolution to support more open and global standards, so that any business can perform electronic document exchanges with any other business.

First era: National and industry-oriented EDI data structures

Twenty years ago, many companies along vertical industries participated in defining the first standards for exchanging information throughout a supply chain. Because this early work started along vertical industries, the standards that were created focused on industries, such as retail, transportation, automotive, and so on. Besides, this kind of work was occurring in multiple countries, resulting in standards that had a vertical industry orientation as well as geographical characteristics. The result was the overlapping of data structures across multiple

industries in different geographies with no interoperability among these standards.

Second era: International EDI data structures

The proliferation of multiple electronic data interchange (EDI) standards dramatically increased the implementation cost of EDI. Cross-industry players, such as transportation companies, found themselves having to learn and implement different EDI data structures depending on the industry that they served and the region in which they operated. Standards that became the dominant format for conducting e-commerce are ANSI X12 in North America; TRADACOMS in the United Kingdom; GENCOD in France (retail); Uniform Communication Standard for the U.S. grocery industry; and MDA SEDAS in Germany.

Gradually, the various industry groups came together to create one international data standard: Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT). Nowadays, most trade between companies in two separate countries is based on the EDIFACT standard. The migration from industry-oriented or nation-oriented standards to international standards is still happening and might take years. However, the EDI world has achieved one common data structure, which will help drive the costs of EDI much lower.

Third era: National and industry-oriented XML data structures

During the last twenty years, the majority of EDI growth supported the business focus on direct material procurement and the movement of goods. EDI data structures worked well for direct material purchases given the structured nature of the procurement process. Prices, contracts, delivery, shipping, and many other details are negotiated and determined during the procurement process.

When newer technologies arose and the Internet became the platform for e-commerce, XML appeared on the horizon. XML has offered one key advantage. That is, data streams can now be interpreted and presented to both human beings and machines. The availability of XML meant that users with a browser can receive data structures without any change to that data's construct. XML offers other benefits. Because the technology is extensible, users can add data tags regardless of whether the receiving user is prepared to handle the additional data. However, adding extra tags can cause interoperability issues.

In the late 1990s, lacking any XML data structure organizations, vertical groups again led the charge to create specific vertical-industry data structures. The first group to complete this task was RosettaNet, which defined a series of XML data structures for the high-tech supply chain. The lack of already agreed-upon XML data structures and the perceived need to create them rapidly led the vertical

groups to create, publish, and endorse their own data structures. Examples are xCBL from Commerce One and cXML from Ariba.

Early cross-industry adopters of XML data structures find themselves in a similar situation to early adopters of EDI. They must still learn and implement multiple XML structures, which now include the proprietary ones just mentioned as well as several others. There is no interoperability among the differing XML data structures, and companies need to implement these multiple XML standards to reach all of the constituents in their supply chain. Supporting multiple XML standards will drive the same interoperability issues that existed with EDI during the 1980s and will similarly lead to higher implementation costs.

Fourth era: International XML data structures

XML is now being adopted for B2B e-commerce on a national basis and the international use of XML is just a step away. In the meantime, the same issues arise that historically accompanied the lack of an accepted standard. Industry groups, such as RosettaNet, find themselves making changes to accommodate international needs. Meanwhile, vendors who have defined their own XML standards want their standards to become international.

Certain participants in the e-commerce community acknowledge the role played by e-business XML (ebXML) as one of the early pioneers of a global standard industry set of XML standards. The ebXML work is a joint effort between the United Nations body for Trade Facilitation and Electronic Business (UN/CEFACT) and the Organization for the Advancement of Structured Information Standards (OASIS). UN/CEFACT and OASIS have users defining the documents. They both have previously worked in the e-commerce standards arena, which ensures that strong data dictionaries exist, as well as processes for change control, communication, and documentation regarding the standards.

1.2.4 Evolution of B2B data communications

In addition to an evolution in the structure of exchanged documents, there has been an evolution in the communication method to exchange documents.

First era: Point-to-point direct connections

Early inter-enterprise computer-to-computer data exchange moved data via primitive computer protocols. At that time, numerous communication protocols were used. Several communication protocols are still found in vertical-industry implementations, such as the BiSync 2780/3780 no-logon protocol of the Uniform Communications Standard for the U.S. grocery industry. Other early protocols still in use include ANSI Clear and X, Y and Z Modems.

Second era: Value-added networks

To communicate across a supply chain with multiple suppliers, an early EDI user needed to manage a variety of protocols. Implementing and supporting the many and different communications protocols that were proliferating had a cost. Companies often needed to buy multiple products. These products, in turn, required their own operational assistance with scheduling transmissions, executing the programs, and setting up audit and error-handling procedures.

Soon, users migrated to value-added networks (VANs) to resolve this issue. The VAN became popular, because it was able to insulate the protocol of a given trading partner from the protocols used by all the other trading partners. The VAN offered protocol conversion and insulated one company from another company so neither company logged on to the other's system. This insulation provided security and eliminated the need to build an operational communications infrastructure capable of supporting communications sessions with multiple concurrent users. The VAN also insulated users from having to synchronize communications. Companies were free to bring down their systems and perform maintenance without coordinating their activities with multiple trading partners.

Third era: Internet

In the 1990s, the Internet became the primary focus for conducting e-commerce. The prevailing view was that the Internet will replace the VAN as a network intermediary. Leaving the VANs, early Internet adopters wanted the Internet to become the universal protocol and to offer unlimited, inexpensive reach. Unfortunately, these early adopters are facing a familiar situation. Again, they have had to learn how to deal with differing format protocols, such as File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), and Simple Mail Transfer Protocol (SMTP).

First users have also had to learn how to deal with securing the data during transport. Security concerns have led to the management of certificates for using encryption methodologies, such as Secure Sockets Layer (SSL) and Secure/Multipurpose Internet Mail Extensions (S/MIME). Given the absence of prevailing format protocols, vertical industry groups again moved in to sort out confusion by endorsing a given protocol format and encryption methodology.

1.3 Enterprise application integration and B2B

A commonly used definition for enterprise application integration (EAI) is the integration of multiple, independently developed and maintained applications that use incompatible technologies and that are deployed on a wide range of platforms. EAI capabilities for integrating existing and new applications are

fundamental for reacting to business change. At its simplest, EAI enables the transfer of information between applications. But EAI can offer so much more. It can automate the flow of data between the applications that make up the business process flow. The applications in the flow must be enabled to send, receive, and work with this data and to return appropriate results.

With this definition, EAI is transformed from the relatively simple coupling of applications to a global business process implementation and integration. Important business processes comprise many applications combined into complex business tasks and they need to be up and running 24x7.

A major characteristic of EAI is automation. The integrated applications and business processes must run completely without human intervention. If human intervention is required, a workflow manager can generate work items to allow a company's staff to participate in the business processes.

1.4 B2B integration

As described earlier, B2Bi describes e-commerce where the relationships between businesses are one-to-one (Figure 1-1 on page 6).

Nowadays, business depends more and more on strategic relationships with suppliers and partners to establish value chains that provide a competitive advantage. B2B integration is application integration extended outside a single company. It is about companies that trade with partners and suppliers over the Internet in real time. It is about using middleware technologies, such as distributed objects, remote procedure calls, message queuing, data transformation, and publish/subscribe, to connect multiple applications with the added complication of getting through firewalls. It is about using the Internet to share data across company boundaries. It is about agreeing on a data structure (standards) and exchanging data electronically using these standards.

B2Bi is becoming extremely important in many business areas, for example:

- ▶ Financial transactions, such as checking account balances, transferring payments, and obtaining credit information
- ▶ Manufacturing activities, such as supply chain planning and execution
- ▶ Retail activities, such as checking a supplier's stock, placing replenishment orders, and paying suppliers automatically
- ▶ Travel tasks, such as checking flight, car, or room availability, and making or changing reservations

An IT infrastructure for automating and coordinating B2B processes is clearly necessary for B2Bi. B2Bi improves performance by supporting key principles of business success:

- ▶ Faster time-to-market with new products and services
- ▶ Better sales process
- ▶ Better service
- ▶ Lower operational and production costs
- ▶ Lower inventory costs

Implementing B2Bi solutions that span many and different independent organizations is challenging due to the following considerations:

- ▶ Heterogeneous data or information
Various applications and users represent information in multiple ways or use various kinds of information for the same task. Bridging the associated syntactic and semantic gaps in information can require a mixture of transformation capabilities and neutral information representations.
- ▶ Heterogeneous systems
Information systems at various organizations within the enterprise are composed of various applications, which include Enterprise Resource Planning (ERP) systems, existing business applications, advanced planning, product data management, document management, and Web-based intranet applications. Organizations also use various middleware technologies, such as messaging and groupware systems, and distributed object frameworks, such as Distributed Component Object Model (DCOM) and Common Object Request Broker Architecture (CORBA). B2Bi solutions must interoperate with these systems.
- ▶ Heterogeneous business processes
Businesses do things differently. The internal processes used for handling orders or managing production and planning are often unique to the organizations that deploy them. It is always a challenge to reach agreement on processes that involve multiple organizations.
- ▶ Dynamic business and technology environment
Besides heterogeneity, B2Bi is characterized by frequent change. Business processes and the internal systems environment change often. Inter-organizational agreements are also subject to change. Coordinating these changes across organizations is a difficult task.
- ▶ Security and reliability of communications
Before two systems between two separate organizations can interact, reliable, secure communications pathways must exist. When the pathway

includes an open network, such as the Internet, security is even more important.

- ▶ Organizational autonomy

Approaches to inter-enterprise processes must respect organizational autonomy and minimize the complexity of mutual commitments among multiple organizations.

1.4.1 Types of B2B integration

The following sections distinguish the three major types of B2B integration:

- ▶ Data/applications sharing
- ▶ Document exchange
- ▶ Process integration

Data/applications sharing

The data/applications sharing type of B2Bi solution makes a set of data/applications available for direct access by outside organizations. This access is usually done through an organization's Web FTP server or messaging middleware, such as WebSphere MQ. Multiple URLs are dedicated for this purpose and let outside companies access information in the form of HTML or XML or an agreed proprietary form. Returned information can be static (come from the internal database) or invoke execution of the internal enterprise applications (refer to Figure 1-2 on page 14). This approach works well for simple interactions. The primary advantage is that it requires almost no specialized software or hardware investment from participants.

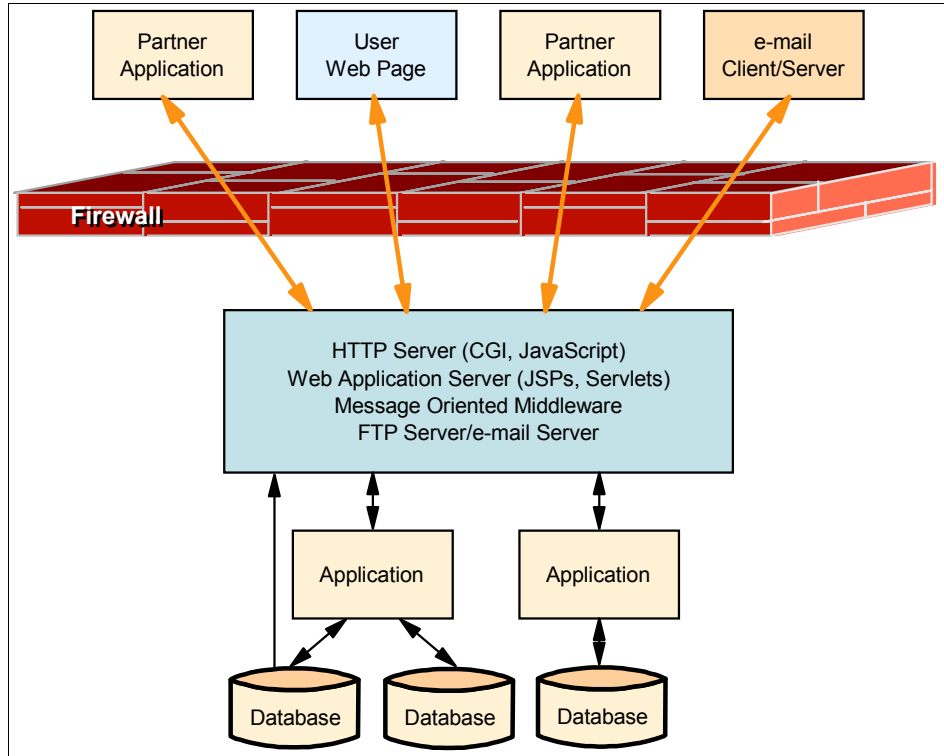


Figure 1-2 Data and information sharing B2Bi

This approach provides simplicity and speed-to-market. These advantages stem from support of well-known Web technologies to access existing enterprise databases and applications.

Data/applications sharing is about exposing data through the Web. This data is defined by analyzing the existing information within the enterprise and deciding which part is of interest to business partners, providers, or suppliers.

It is also necessary to determine from where this data originates. It can come from either existing enterprise databases or applications. The data format is another important component, which needs to be identified. It determines how information is structured, including the properties of the data elements within that structure. When the enterprise is internally integrated, EAI is usually an application that has to be accessed.

Document exchange

The document exchange approach represents the most common current practice for interactions between businesses. Each participant defines an entry point or

entry points through which various types of documents (such as EDI documents or XML documents) can be delivered. The difference between this approach and data/applications sharing is that this approach is a push as opposed to a pull communications style, providing more controlled timing for the information exchange. An enterprise with information “pushes” this information to all interested parties. There are two common implementations of this approach: EDI messaging and Web messaging.

For EDI messaging, a VAN service provider delivers messages to entry points into the enterprise and mediates interactions. Bandwidth for EDI networks is expensive, even today, which is why the creators of EDI were mainly concerned about the size of their messages. EDI messages are compressed and use codes to represent complex values. All the metadata is stripped from an EDI message, which makes it difficult to read and debug. The complexity of EDI makes EDI programmers hard to train and expensive to keep, which makes EDI applications expensive to buy and maintain.

An alternative approach to building document exchange is to use XML-based message formats. The communication occurs through the public Internet, rather than VANs, using HTTP or some proprietary messaging protocol, such as WebSphere MQ, to achieve assured message delivery. Because the public Internet is essentially free, message size is not a factor, which is why XML messages are rich in metadata, making them easy to read and debug. The simplicity of XML makes XML programmers easy to train and less expensive to keep, making XML applications less expensive to buy and maintain. Additional non-functional requirements, such as security, must also be factored into the design.

For internal support of document data collection, we can use an internal EAI (message-level) or Business Process Integration (BPI; process-level) implementation. If the integration backbone is not in place, the point-to-point integration between B2B applications and applications, which need to interact, must be in place.

Document exchange can be implemented using either a standard Web server or a specialized B2Bi server. The advantage of the first approach is cost savings. Although this approach can be a good starting point for the implementation, it is usually not a viable solution for the final system. The amount of code necessary to implement features, such as guaranteed document delivery and document data transformation, which are part of any modern B2Bi server, is too great. We recommend that you implement document exchange using specialized B2Bi servers. And if EAI is in place, you can reuse its capabilities of connecting to existing applications and transforming and routing data to a B2Bi server (Figure 1-3 on page 16).

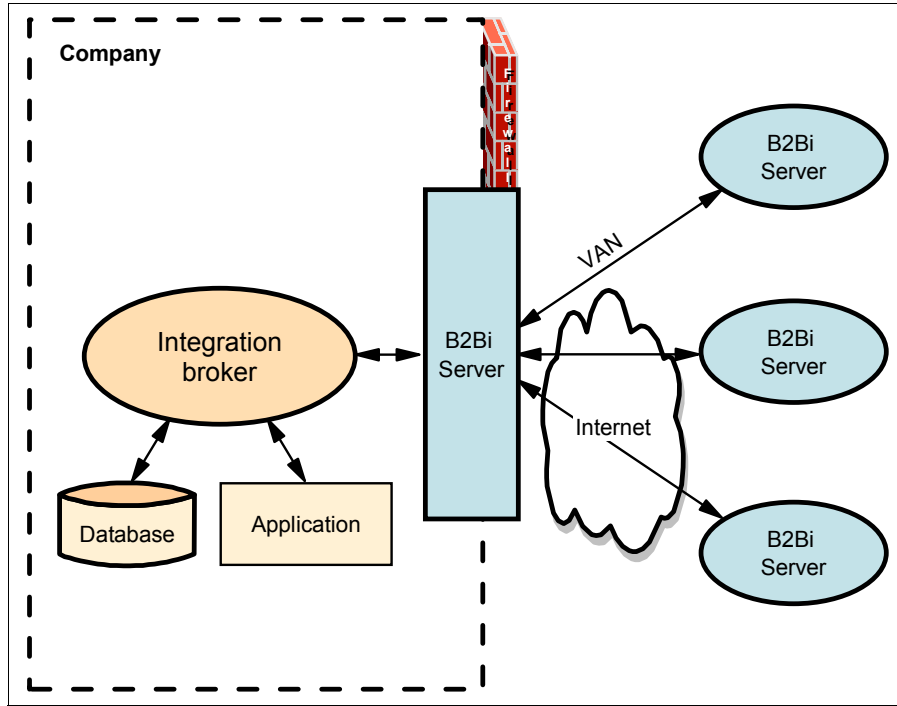


Figure 1-3 Using an internal EAI with B2Bi server

Now, think about application integration. Each time that you add a new application to the existing integration infrastructure, its complexity grows. To avoid the inter-application spaghetti phenomenon, in this case, we implement an integration broker. Each application talks with only one integration broker, and it is unaware of other applications. That is the transition from point-to-point to hub-and-spoke architecture.

We can make similar conclusions about B2Bi servers. You can solve point-to-point communications problems, where each company needs to make communication links with suppliers and partners, with a B2Bi hub (Figure 1-4).

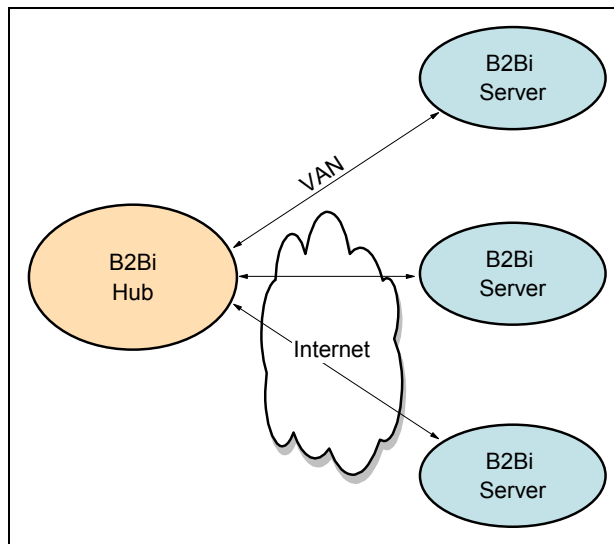


Figure 1-4 B2Bi hub

Process integration

This type of B2Bi solution deals with building inter-enterprise business processes, which incorporate existing internal enterprise processes. Process integration is an extension of the document exchange. The communications are still done through document exchange, but this exchange happens within the context of the business process. This approach is the most advanced B2Bi implementation. It transforms existing, disparate enterprises into a cohesive system of business processes, supporting all the functions required by the extended virtual enterprise.

Process-based B2Bi manages the interaction between multiple enterprises under the umbrella of integrated B2Bi and internal business workflow. Business applications or internal business processes execute major steps in the B2B workflow. On this basis, we can divide business processes into private and public.

We recommend that you use a two-level implementation of process-based B2Bi (Figure 1-5). With this approach, public processes are implemented using a B2Bi server and private processes are implemented using an internal integration broker, such as WebSphere Process Server.

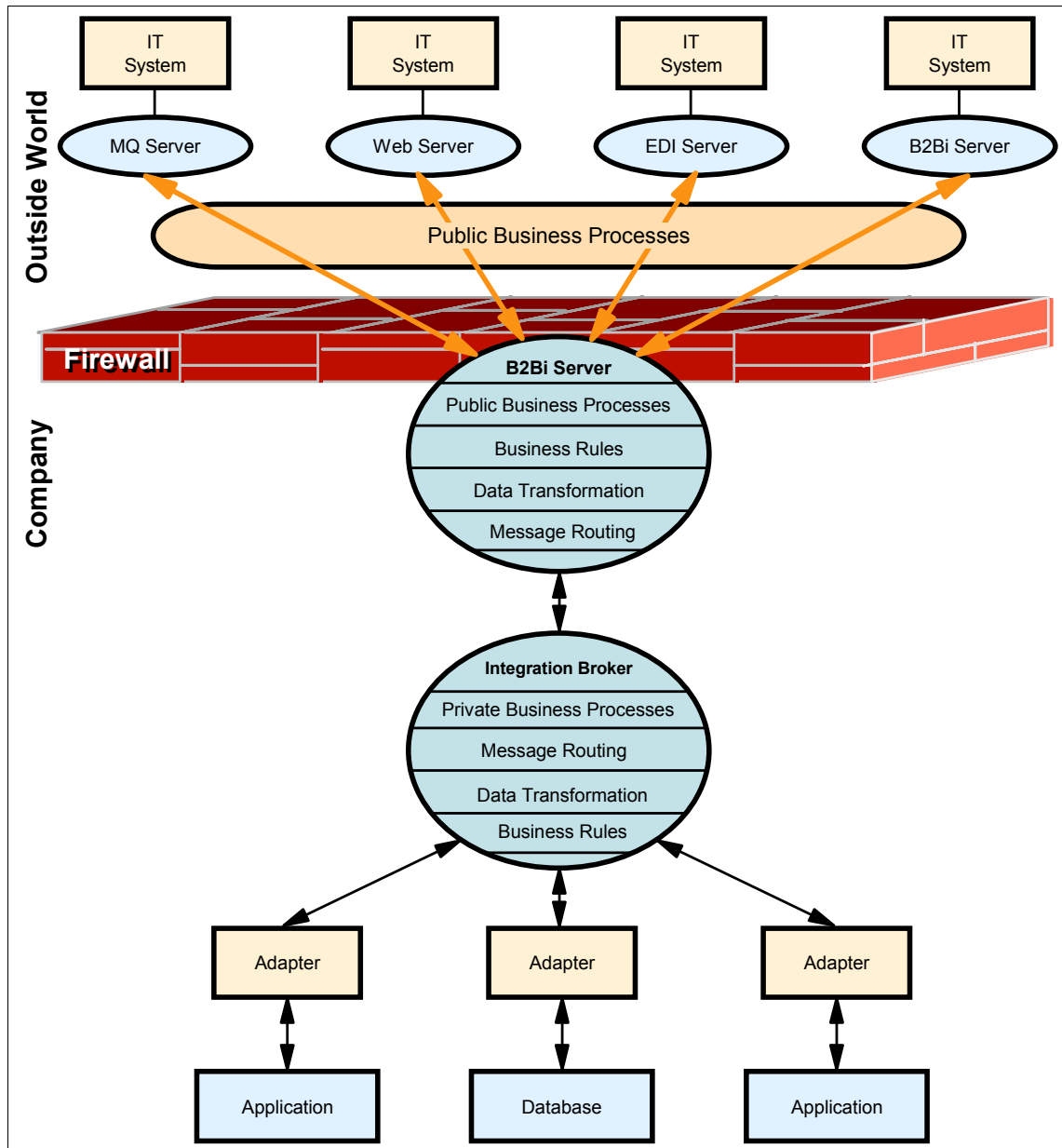


Figure 1-5 B2Bi on a business process level

The advantages of the proposed B2B integration infrastructure are:

- ▶ There is a clean separation of private and public business processes, although private processes support public ones.
- ▶ The B2Bi server is responsible for all public processes, business rules, data transformation, and routing between partners and internal systems.
- ▶ The integration broker is responsible for all private processes, business rules, data transformation, and routing, which provides a central point of control.
- ▶ Built-in GUI tools allow for process definitions, data transformation definitions, and business rules definitions.
- ▶ B2Bi servers usually have built-in security and support for assured message delivery.

The use of process-based B2Bi addresses B2B integration in virtually all areas. It includes agreed message formats, message sequencing, communications and security protocols, workflow steps, and business rules. It also reduces the application code that is required to execute business processes. In a Business Process Management solution, workflow management provides flexibility in changing the sequence of actions, while message routing and transformation provide the same kind of flexibility in changing the flow and format of communications.

1.4.2 Summary

Process-based B2Bi is the ultimate B2Bi implementation. This approach allows a virtual enterprise to formalize and automate the way that it does business. This approach is the most expensive proposition. It requires participating enterprises to be internally integrated first, but it usually provides the greatest benefits.



B2B technologies and standards

This chapter discusses business-to-business (B2B) and the technologies and standards that are typically needed to successfully send information from one business to another. It also discusses the minimum that is required by most businesses. Because each business is different and has unique requirements, you need to use a case-to-case approach when deciding which and how many components are required.

2.1 Requirements for a B2B solution

It seems that most businesses have fairly similar requirements relating to the issues of sending data either between applications or between themselves and their trading partners.

The requirements for most B2B scenarios (in all or part) are:

- ▶ Ability to send and receive data

This data can be structured or unstructured data across a variety of transport protocols, for example, Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), WebSphere MQ (WMQ), or Java Message Service (JMS).

- ▶ Definable data formats

Definable message formats allow businesses to communicate with other businesses, with data that might have to travel between different operating systems and programs written in different languages. Having a definable message format also allows for more of a plug-and-play solution so that other businesses or applications can participate in this data sharing. Most B2B applications have the business map their data into a generic data object, which allows for easier manipulation of the data and for easier mapping to other business data formats. It allows for a layer of abstraction between the data formats of an individual application. It also allows businesses to “plug in” additional business with minimal configuration.

- ▶ Security of data

Security needs to be available from the time that the data leaves the sending application until it arrives at the receiving application. This security is even more important today with many businesses using their intranet or the Internet as the travel medium.

- ▶ Availability of messaging systems

Messaging systems need to have a capability for failover or recovery and continuous operations without losing or corrupting any data.

- ▶ Monitoring and auditing capabilities

The ability to monitor the progress of data through the system is required to provide a user with the ability to see the progress of their data and an administrator with the ability to perform problem or fault investigation and resolution. Auditing capabilities are needed to determine what has been sent or received and what partner was involved.

- ▶ Transactional support

The decision to commit or back out the changes is made, in the simplest case, at the end of a transaction. However, given the distributed nature of

B2B transactions, the concept of transactional support takes on a whole new meaning and level of complexity. The coordination of local transactions and the synchronization of distributed transactions are key issues for data integrity.

► Performance

The system must have the ability to scale to handle the growing needs of the business.

Businesses choose the technology that they use based on several criteria:

- In-house skill
- Cost to retrain present employees
- Cost to adopt new technology
- Cost to integrate new technology
- Maintenance cost of new technology

For example, if the business has sufficient Java skills in-house, using a Java-based technology might not have a huge impact on the time that is required to implement a project. However, if only COBOL programming skills are available in-house, using a Java solution requires more work and adds expense and time for retraining the programmers. Does the company not use a Java solution? This answer depends on where the company is planning to go with its solution and how accepting the programmers are to this new technology. Changing from one technology to another is not difficult, but it might require learning a new programming style and using different tools to perform the programming.

Companies must choose carefully when adding new functionality and skill to a group. Never choose a technology because it is the new trend in the industry.

2.2 Terminology

Here are several examples of the types of technologies that you will encounter with a B2B solution.

2.2.1 Messaging and queuing

Message queuing has been used in data processing for many years. Without queuing, sending an electronic message over long distances requires every node on the route to be available for forwarding messages. Also, the addressees must be logged on and conscious of the fact that you are trying to send them a message. In a queuing system, messages are stored at intermediate nodes until the system is ready to forward them. At their final destination, they are stored in a queue until the consumer of the message is ready to retrieve them.

Even so, many complex business transactions are processed today without queuing. In a large network, the system can be maintaining thousands of connections in a ready-to-use state. If one part of the system suffers a problem, many parts of the system become unusable.

In message queuing, a *message* is simply a collection of data sent by one program and intended for another program.

Queuing is the mechanism by which messages are held until an application is ready to process them. Queuing allows you to:

- ▶ Communicate between programs, which might be running in different environments, without writing the communication code
- ▶ Select the order in which a program processes messages
- ▶ Balance loads on a system by arranging for more than one program to service a queue when the number of messages exceeds a threshold
- ▶ Increase the availability of your applications by arranging for an alternative system to service the queues if your primary system is unavailable

A message itself normally consists of two parts (refer to Figure 2-1):

- ▶ Control information, which contains information, such as:
 - The type of the message
 - An identifier for the message
 - The priority for delivery of the message
 - Whether a response is required
- ▶ Application data, for example:
 - Business message type, such as purchase order or shipping notice
 - Business identifiers, such as sender and receiver

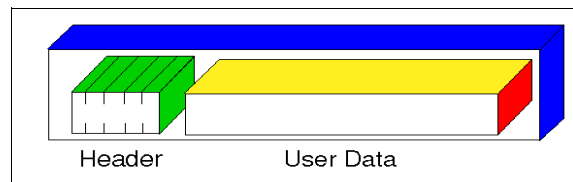


Figure 2-1 MQ message

2.2.2 Electronic data interchange

Electronic data interchange (EDI) is the direct computer-to-computer transfer of business information between applications using a standard message format.

EDI over value-added network

EDI over value-added network (EDI/VAN) is a private third-party network. It usually has built-in security features that help protect against unauthorized access to customer data. It is 99.9% available and usually has an archive capability for data copies.

It is secure and reliable, but more expensive than the Internet.

EDI over the Internet

EDI over the Internet (EDI-INT) is the transmission of EDI over the Internet. The major purpose of EDI-INT is to reduce the cost of transmission. Four key message transmission standards are used for EDI-INT:

- ▶ **AS1:** Uses Multipurpose Internet Mail Extensions (MIME) and Simple Mail Transfer Protocol (SMTP)
- ▶ **AS2:** Uses MIME and HTTP
- ▶ **AS3:** Uses MIME and FTP
- ▶ **AS4:** Uses Web Services Security and Web Services

Message format standards

The following message format standards are used:

- ▶ ANSI X12
American National Standards Institute committee X12 (ANSI X12) defines data that is separated by characters. The message is organized into documents called *Transaction Sets*. These Transactions Sets are in groups called *Functional Groups*, which are then “wrapped” in an envelope called an *Interchange*.
- ▶ EDIFACT
Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) defines which data segments are mandatory or optional, and the number and order of elements.
- ▶ Other formats include:
 - United Nations Trade Data Interchange (UNTDI) Standards
 - Organization for Data Exchange through Teletransmission in Europe (ODETTE)

- Healthcare Information Portability and Accountability Act (HIPAA)
- Health Level 7 (HL7)
- Voluntary Inter-industry Communications Standards (VICS)
- Verband Deutscher Automobilhersteller (VDA)
- Universal Multi-Octet Coded Character Set (UCS)
- Association for Cooperative Operations Research and Development (ACORD)

2.2.3 Transport protocols

Three transport protocols are used mostly when transferring documents in a B2B solution.

HTTP

HTTP is the common standard for transferring World Wide Web documents. This protocol operates over Transmission Control Protocol (TCP) connections, usually over port 80. An HTTP client sends Get, Post, or Head messages to an HTTP server, which allows the exchange of data and resources, such as a URL or file, for example:

▶ GET:

```
GET /path/to/file/index.html HTTP/1.0
```

▶ HEAD:

Similar to a GET but returns the response header only

▶ POST:

Used to send data to the server

FTP

Also called “Fetch”, FTP requires a client and a server. The client connects to the server and might have permission to do everything that can be done locally on the server, except create new files from scratch. However, FTP is mainly used for uploading and downloading large groups of files at one time.

SMTP

SMTP is the Internet standard host-to-host mail transport protocol. It is traditionally over TCP on port 25. SMTP uses a request-response protocol. Because SMTP is limited in its ability to queue messages at the receiving end, Post Office Protocol 3 (POP3) or Internet Message Access Protocol (IMAP) is used to save the message in a server mailbox.

2.2.4 Security

Security technologies are involved at several layers in a B2B solution. They simply protect access to a resource as well as make a resource unreadable for parties not involved in the interaction.

Access control list

Access control list (ACL) specifies a set of rules regarding who is allowed to access a particular resource.

Encryption

The major use of encryption is to assure the confidentiality of an exchanged document:

- ▶ Public Key Infrastructure (PKI)

With PKI, the encrypting and decrypting functions are comprised of mathematical algorithms and the keys are represented by numbers.

- ▶ Secret key cryptography

With secret key cryptography, also known as *symmetric key encryption*, one key is used to encrypt and the same key is used to decrypt (Figure 2-2).

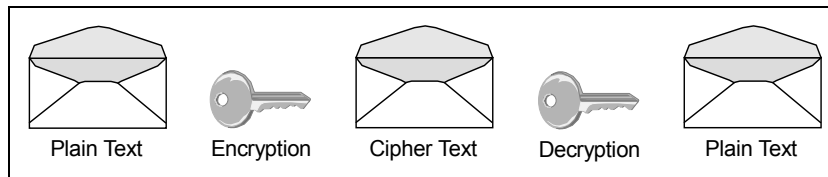


Figure 2-2 Secret key cryptography

- ▶ Public key cryptography

With public key cryptography, there are different keys for encrypting and decrypting functions. In Figure 2-3, something encrypted with key 1 can only be decrypted with key 2.

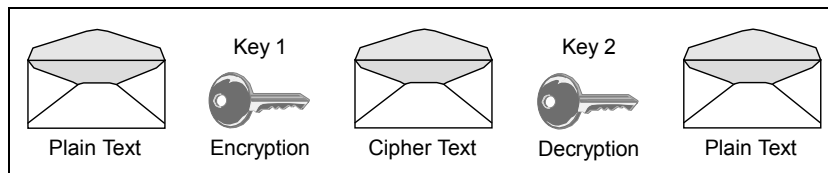


Figure 2-3 Public key cryptography

Hashing

Hashing a document is mainly used to protect a document against intended changes or tampering. Recalculating the hash from the received document and comparing it with the received hash value is a technique to discover any changes. You can choose from several algorithms to achieve hashing:

- ▶ SHA-1, 256, 384, 512

Secure hash algorithm (SHA) takes the message and pads it by adding bits to make it a certain length. It is then parsed into n Mb blocks to make sure that the message is a multiple of 512 or 1024 bits. Next, the hash value is set. This hash value is determined by the hash algorithm and by taking the first m -bits of the fractional parts of the square roots of the x through y prime numbers.

Note: The values of m , x , and y are determined by the hash algorithm.

- ▶ MD5

Message digest algorithm 5 (MD5) takes the message and then pads it by adding bits to make it 64 bits shy of being a multiple of 512 bits. Next, a 64-bit representation of the message is added so that the message is exactly a multiple of 512 bits. Next, four 32-bit values are used to compute the message digest. This message digest is used to produce the output values.

Digital signatures and certificates

This technology uses a public and a private key. Either key can be used for encrypting the data, but then only the corresponding key can be used to decrypt the data.

MIME and S/MIME

Multipurpose Internet Mail Extensions (MIME) allows messages to contain:

- ▶ Multiple objects in a single message
- ▶ Text having unlimited line length or overall length
- ▶ Character sets other than ASCII, allowing non-English language messages
- ▶ Multi-font messages
- ▶ Binary or application specific files
- ▶ Images, audio, video, and multimedia messages

Secure/MIME (S/MIME) adds:

- ▶ Message privacy
- ▶ Digital signatures
- ▶ Tamper detection
- ▶ Interoperability

- ▶ Seamless integration
- ▶ Cross-platform messaging

Secure Sockets Layer

Secure Sockets Layer (SSL) is a protocol that was designed to provide secure communications on the Internet. SSL authenticates that the server is “who” it is supposed to be. SSL creates a secure communication channel by encrypting all communication between the client and the server. SSL conducts a cryptographic word count (checksum) to ensure data integrity between the server and client. Checksum is the number of bytes in a document, and it is sent along with the encrypted document when the server receives the message.

2.2.5 Extensible Markup Language

XML has gone from the latest buzzword to an entrenched technology in record time. These days, many businesses use XML to solve business problems.

XML is an open messaging standard that provides a cross-platform portable mechanism for exchanging data. XML refers to a family of specifications based on a tagged message format for *metadata*. The tag language has been developed from older markup standards, including Generalized Markup Language (GML) and Standard Generalized Markup Language (SGML).

XML definitions for specific business objects, such as messages used by EDI or financial applications, are grouped using “schemas” or *document type definitions* (DTDs).

The XML standard is growing quickly. It is being adapted to, and supported by, an increasing number of products.

2.2.6 Electronic Business using Extensible Markup Language

ebXML is a standard sponsored by OASIS and UN/CEFACT. It is an open messaging standard that builds on top of XML, and it enables the use of electronic business information by trading partners that is interoperable and secure. This standard consists of the following five parts:

- ▶ Core Components Technical Specification
- ▶ Messaging Service Specification
- ▶ Collaborative Partner Profile Agreement
- ▶ Registry Information Model
- ▶ Registry Services Specification

The two most common parts of the ebXML standard being used by companies today are the Messaging Service Specification and the Collaborative Partner Profile Agreement.

2.2.7 Web services

Web services are self-contained, self-describing, modular applications that can be published, located, and invoked over a network. Web Services utilizes the SOAP protocol and Web Services security to protect your data. With Web services, there is a Universal Description, Discovery, and Integration (UDDI) server. On this server, Web services can be located, published, and updated. When the desired Web service is located, a Web Services Description Language (WSDL) file is associated with it and contains information about the interface, the implementation, and the service provider. With this information, a Web service can be invoked.

The use of a UDDI server is optional. Web service clients can typically retrieve the WSDL from other sources as well.



B2B deployment methodology

This chapter describes IBM business-to-business (B2B) deployment methodology; the methodology describes the typical phases of a B2B project. It is not all inclusive and all of the phases listed here do not necessarily need to be used to complete your B2B deployment. The actual phases that you use need to be determined by your requirements and the key success criteria that best meet your business needs. We list the phases of a B2B project here to help as a reference to show the typical steps in deploying B2B integration technologies.

3.1 B2B deployment planning

Many B2B deployments are unsuccessful due to the lack of proper planning prior to installing and configuring the software solution. It is true that certain B2B deployments require little up-front preparation, while others require a considerable amount of up-front design and planning.

One of the fundamental problems with many B2B projects is that, many times, a deployment is performed without consideration of the business and security requirements that are related to the B2B solution. Being successful at deploying B2B technology is much more than just deploying software and hardware. It requires a thorough understanding of:

- ▶ B2B as a technology, including the e-commerce standards that it employs
- ▶ The B2B market and the customers that drive this technology
- ▶ The value proposition of IBM B2B solutions

IBM Software Services for WebSphere created a B2B deployment methodology that is intended to guide implementors of B2B Integration technologies through a proven process that has been used and refined over many years and used for many B2B projects. We are providing the methodology in this book to give you an idea of what a typical B2B project looks like. The phases are not all inclusive, and in many cases, several of the phases will not be needed for a specific project.

3.2 B2B deployment methodology overview

The B2B deployment methodology is divided into eight distinct phases with knowledge transfer throughout the project life cycle. The phase execution flow is shown in Figure 3-1 on page 33.

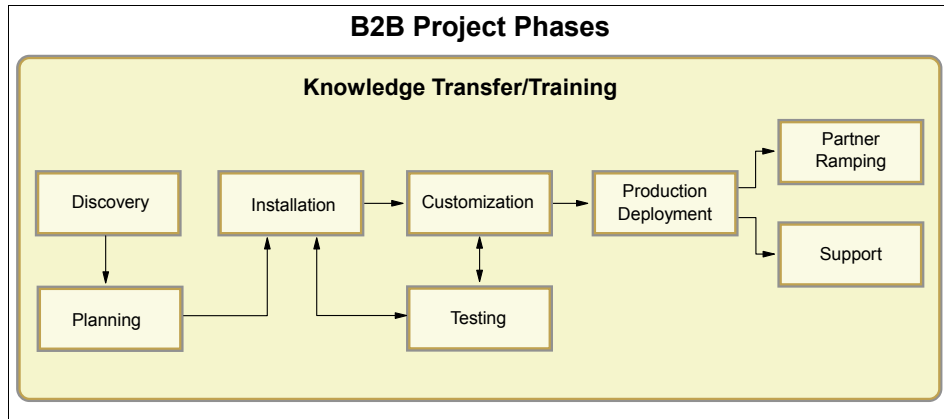


Figure 3-1 Phase execution flow

3.2.1 Knowledge transfer and training

Training and knowledge transfer need to encompass the entire project; it is important to identify the project representatives at the start of the project so that a plan can be put in place to allow ample time for training prior to the production deployment phase of the project. All of the personnel that will maintain the solution must be trained and must participate in many phases of the project to obtain as much knowledge about the specific deployment as possible.

Phase tasks

In this phase:

- ▶ Recommend training classes and provide class schedules and synopses to the project sponsor.
- ▶ Recommend appropriate IBM Redbooks publications and provide information about how to gain access to the books.
- ▶ Provide guidance to personnel during each phase of the project.

Deliverable

This phase is ongoing throughout the project, and the key deliverable associated with it is the training plan that is developed during the planning phase. It is designed to provide a roadmap to get personnel up to speed as fast as possible on the B2B technology being used.

Skill sets and resource requirements

The skill sets required for this phase are:

- ▶ IT infrastructure knowledge, including network, connection, and data security
- ▶ Expert knowledge of IBM B2B and integration solutions
- ▶ Expert knowledge of High Availability concepts
- ▶ Knowledge of the data center infrastructure, hardware, and software that are designated for the B2B project
- ▶ Knowledge of other IBM products related to B2B and application integration (You might need to bring in a specialist.)

3.2.2 Discovery

The Discovery phase is where the business and security requirements are reviewed and revised, the scope of the project and critical success factors are defined and agreed upon, the existing infrastructure and environment are reviewed, software and hardware distribution models are discussed, and roles and responsibilities are identified. This phase consists of meetings with the business owners (typically, the project sponsors) and the technical personnel who maintain the current applications and network environment. At the end of this phase, all hardware and software needed for the solution must be identified and procured. We advise that you first perform this phase prior to communicating any effort estimate. All of the discovery work must be completed prior to moving forward; this task is considered a key milestone in the project.

Phase tasks

The phase tasks include:

- ▶ Schedule a deployment assessment meeting with the business and technical project participants. This meeting can be a single meeting or can be broken up into separate meetings. This meeting is used to gather the requirements and to introduce each team member.
- ▶ Create the deployment assessment presentation and agenda.
- ▶ Create a roles and responsibilities worksheet.
- ▶ Create the requirements and scope documents.

Deliverables

After completion of this phase, you need to be ready to deliver:

- ▶ Meeting agenda presentation
- ▶ Detailed requirements document, which includes a summary of what was discovered during the meetings, including existing environment information:
 - Detailed definition of the business requirements
 - Detailed definition of the security requirements
 - Hardware and software requirements based on the distribution model and capacity estimates
- ▶ Detailed scope document
- ▶ Roles and responsibilities document

Skill sets and resource requirements

The skill sets and resources required for this phase are:

- ▶ Project planning skills and experience
- ▶ IT infrastructure knowledge, including network, connection, and data security
- ▶ Expert knowledge of IBM B2B and integration solutions
- ▶ Expert knowledge of High Availability concepts
- ▶ Knowledge of the data center infrastructure, hardware, and software that are designated for the B2B project
- ▶ Knowledge of other IBM Products related to B2B and application integration (You might need to bring in a specialist.)

3.2.3 Planning

The Planning phase takes all of the information that was learned during the Discovery phase as input into all of the plans needed to successfully deploy the B2B solution. This phase is used to build a technology deployment strategy that meets the key objectives for partner connectivity, security, integration, protocols, data validation and transformation, and so forth. The deployment strategy can include the deployment of additional hardware and software to fully meet all of the business requirements. During this phase, additional information is typically discovered that changes the project scope and schedule; it is important to implement change control to ensure that all changes are documented and agreed upon by the project sponsors.

Phase tasks

The phase tasks include:

- ▶ Analyze the requirements and scope documents to provide content for each plan.
- ▶ Schedule follow-up meetings as needed.
- ▶ Implement change control and revise scope and schedule as needed.
- ▶ Create a project plan and time line (This task is sometimes is done in the Discovery phase).
- ▶ Create physical and logical architectural diagrams.
- ▶ Create a deployment plan.
- ▶ Create a development plan and schedule if applicable.
- ▶ Create a failover plan and strategy if implementing High Availability.
- ▶ Create a disaster recover (DR) plan and strategy if implementing a DR environment.
- ▶ Create a data archive/backup plan and strategy if applicable.
- ▶ Create a testing plan.
- ▶ Create a training plan.
- ▶ Create a Partner Ramping or Partner Onboarding plan.

Deliverables

Deliverables resulting from this phase include:

- ▶ Detailed project plan defining all project tasks and time line (This task is sometimes is done in the Discovery phase)
- ▶ Detailed deployment plan, which includes:
 - Existing environment overview and diagram
 - Proposed environment overview and diagram
 - Software requirements and dependencies
 - Customization requirements
 - Requirements compared to functionality gap analysis
 - Migration consideration for existing data
 - Deployment considerations
- ▶ Detailed development plan, which includes:
 - Development requirements summary
 - Development schedule and project plan

- Development effort estimate
- Conceptual design
- Functional design
- Development test plan
- User documentation: New features and functionality, installation and configuration, and so forth
- Issues log
- ▶ Detailed testing plan, which includes:
 - System testing (installation testing and basic trading)
 - Functional testing (end-to-end document trading scenarios with integration)
 - Stress and volume testing (Verify that the proposed hardware and software are sufficient to handle current and future (two year projection) transaction volumes.). In certain cases, additional appliances, software, or hardware will need to be added to meet your throughput needs.
 - High Availability (HA) failover testing if HA is implemented
 - Disaster recovery testing if DR is implemented
 - New functionality testing (if custom code is being added)
 - Transformation testing if transformation maps are being used
 - Integration testing: Test connectivity to the back-end systems
- ▶ Detailed training plan, which includes:
 - List of personnel identified as requiring training
 - List of classes for the individuals to attend and complete electronically
 - Information about how to access relevant IBM Redbooks publications related to the hardware and software being used for this project
 - Knowledge transfer recommendations and whether to use IBM Business Partners or consultants for one-on-one training to provide your team with the maximum benefit from a subject matter expert's knowledge
- ▶ Detailed Partner Ramping plan, which includes:
 - Prioritized list of trading partners and their IDs
 - Document flows for each partner
 - Certificate information for each partner
 - Ramping schedule with effort estimates

- Integration requirements for each partner (for example, translation, validation, and so forth)

Skill sets and resource requirements

Identify the necessary skills and resources:

- ▶ Understanding of the functional and non-functional requirements related to the project
- ▶ IT infrastructure knowledge, including network, connection, and data security
- ▶ Knowledge of IBM B2B and integration solutions
- ▶ Knowledge of electronic data interchange (EDI) and mapping functions; can include knowledge of a variety of mapping tools
- ▶ Knowledge of various Enterprise Application Integration (EAI) applications for integrating to the back-end systems
- ▶ Knowledge of database administration and SQL
- ▶ Knowledge of High Availability concepts
- ▶ Project planning skills and experience
- ▶ Knowledge of other IBM products related to business integration (You might need to bring in a specialist.)

3.2.4 Installation

The Installation phase is designed to deliver a fully functional development and test environment to support the Customization, Testing, and Production Deployment phases of the project. It utilizes the information in the deployment plan from the Planning phase to determine how components are deployed and distributed in the customer's network.

Phase tasks

The tasks in this phase include:

- ▶ Review the deployment plan and revise if needed.
- ▶ Document all of the installation inputs that are needed to complete related software installations.
- ▶ Install each environment utilizing the deployment plan to guide you through the installation and configuration process. (Implementation can include the installation of many applications, such as WebSphere Application Server, WebSphere MQ, WebSphere Transformation Extender, WebSphere Message Broker, WebSphere Process Server, WebSphere Adapters, and so forth.)

- ▶ Verify component installation and network connectivity between each component and system (Verify network connections through firewalls, routers, switches, proxies, load balancers, and so forth.).

Deliverables

The deliverables resulting from this phase include:

- ▶ Revised deployment plan if needed
- ▶ Completed installation input document for each environment
- ▶ Installation and configuration of all environments

Skill sets and resource requirements

The required skills and resources from this phase include:

- ▶ IT infrastructure knowledge, including network, connection, and data security
- ▶ Experience installing and configuring B2B and application integration solutions
- ▶ Knowledge of other IBM products related to business integration (You might need to bring in a specialist.)
- ▶ Excellent troubleshooting skills

3.2.5 Customization

The Customization phase is designed to deliver functionality that the core products do not include. Customization is done by utilizing the application programming interfaces (APIs), exit frameworks, mapping tools, (Extensible Stylesheet Language Transformation (XSLT), and so forth (for example, custom messaging protocols, validation of custom document types, custom transports, custom XML, RosettaNet Partner Interface Processes (PIPs), translation maps, and so forth). It utilizes the information in the development plan from the Planning phase to determine what customizations are required to meet the customers' needs. This phase typically runs after the installation of the development environment performed in the Installation phase; however, in many cases, development is done on each developer's workstation utilizing their own local installation of the software. If this situation is the case, this phase can run in parallel to the Installation and Test phases.

In this phase, follow a reputable development management process. There are many processes available but few are geared for the relatively short development cycles that you encounter during customer-driven development. A good example of a short cycle development process is the Scrum Agile software development process.

Go to the following Web site for more information about Scrum:

<http://www.controlchaos.com/about/>

Phase tasks

The tasks include:

- ▶ Review the development plan and revise if needed.
- ▶ Utilize the requirements document to create a functional design document.
- ▶ Create custom code based on the functional design document.

Deliverables

The deliverables resulting from this phase are:

- ▶ Revised development plan and schedule
- ▶ Functional design document with estimated effort
- ▶ Issues log template
- ▶ Completed functionality that can be utilized in IBM integration products to provide a complete solution

Skill sets and resource requirements

Required skills and resources include:

- ▶ Java development experience
- ▶ Experience with EDI data sets
- ▶ Experience with document type definition (DTD) and XML Schema development
- ▶ XLST development experience
- ▶ Knowledge of Internet and B2B messaging protocols (File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), Secure Sockets Layer (SSL), AS1, AS2, AS3, RosettaNet, ebXML, and so forth)
- ▶ Knowledge of database administration and SQL
- ▶ Knowledge of WebSphere MQ
- ▶ Knowledge of other IBM products related to business integration (You might need to bring in a specialist.)

3.2.6 Testing

The Testing phase is designed to provide installation testing to verify the solution functions as required and complete system testing, which is conducted according to the customer's real-world trading scenarios. The Testing phase needs to include functional testing of any customizations that were performed in the Customization phase and integration testing to ensure that documents are flowing properly from end-to-end. Performance testing is also performed in this phase to confirm that the production hardware is capable of handling the throughput requirements. If the environment is being set up for High Availability, additional testing will be needed to ensure that failover works as expected. This phase will utilize the test plan as a guide for running the scenarios and recording the results.

Phase tasks

The tasks include:

- ▶ Review and revise the test plan if needed.
- ▶ Execute the test plan and record the results.
- ▶ Perform performance testing to assist in appropriately sizing the production environment.
- ▶ Perform failover testing when implementing a High Availability environment.

Deliverables

The deliverables that result from this phase are:

- ▶ Revised test plan with recorded test results
- ▶ Completed testing based on the scenarios in the test plan
- ▶ Updated environment sizing recommendations based on the performance testing

Skill sets and resource requirements

The required skill sets and resources are:

- ▶ Knowledge of IBM B2B and integration solutions
- ▶ Quality assurance testing experience
- ▶ Knowledge of Internet and B2B messaging protocols (HTTP, FTP, SMTP, SSL, AS1, AS2, AS3, RosettaNet, ebXML, and so forth)
- ▶ Knowledge of other IBM products related to business integration (You might need to bring in a specialist.)

3.2.7 Production Deployment

The Production Deployment phase is designed to provide the customer with an environment, which is completely configured and tested, that can be used in a production capacity. Typically, the test environment is locked and copied or moved into production. A production deployment might have partners already configured for trading prior to the move to production, and additional partners can be configured based on the Partner Ramping plan.

Phase tasks

The tasks are:

- ▶ Review and revise the production deployment section of the deployment plan if needed.
- ▶ Revise the installation inputs document for the production environment if needed.
- ▶ Create the production environment from the environment installed in the Planning phase.
- ▶ Verify component installation and network connectivity between each component (Verify network connections through firewalls, routers, switches, proxies, load balancers, and so forth.).

Deliverables

The resulting deliverables are:

- ▶ Revised deployment plan, if needed
- ▶ Revised installation inputs document, if needed
- ▶ Fully functional production environment

Skill sets and resource requirements

The required skills and resources are:

- ▶ IT infrastructure knowledge, including network, connection, and data security
- ▶ Experience installing and configuring IBM WebSphere Integration products
- ▶ Knowledge of other IBM products related to business integration (You might need to bring in a specialist.)
- ▶ Excellent troubleshooting skills

3.2.8 Partner Ramping

The Partner Ramping phase follows the Production Deployment phase and is used to connect partners based on the Partner Ramping plan, which was created in the Discovery phase. This phase is also used to build a Partner Ramping strategy to provide the customer with a guideline to follow for deploying future partners; the strategy is created if the customer needs a long-term strategy for adding partners to the hub.

Phase tasks

The tasks are:

- ▶ Schedule a trading partner review meeting with the project sponsors. This meeting is intended to allow you to verify that all trading partners are accounted for in the ramping plan and to get the each partner assigned with a priority. In most cases, you will want to force rank the partners so you can build a schedule connecting to the highest priority partners first.
- ▶ Create a trading Partner Ramping strategy or process document.
- ▶ Revise the trading Partner Ramping plan if needed.
- ▶ Connect trading partners based on the trading partner plan and strategy, which includes building integration targets and gateways as needed.

Deliverables

The resulting deliverables are:

- ▶ Detailed Partner Ramping strategy document, which includes:
 - Connection process flow:
 - Interoperability matrix of interoperable products
 - Strategy for connecting small partners to the hub.
 - Partner information form template
 - Connection to trading partners (Many customers consider this step to be part of the daily operational tasks that their own B2B staff will perform and will only require minimal support from IBM services when ramping the partners)

Skill sets and resource requirements

Required skills and resources are:

- ▶ Experience configuring IBM WebSphere Integration products
- ▶ Knowledge of Internet and B2B messaging protocols (HTTP, FTP, SMTP, SSL, AS1, AS2, AS3, RosettaNet, ebXML, and so forth)

- ▶ Knowledge of other IBM Products related to business integration (You might need to bring in a specialist.)
- ▶ Excellent troubleshooting skills

3.2.9 Support

The Support phase follows the Production Deployment phase and usually runs in parallel to the Partner Ramping phase. The Support phase is where IBM can provide post production deployment support on-site for a minimum of two weeks to ensure that the B2B personnel are equipped to handle the day-to-day support requirements of all of the deployed environments. This phase is also used to create a support process for internal customers and trading partners to report B2B-related issues. In many cases, additional time is spent providing advanced problem determination tips and tricks to your personnel.

Phase tasks

The tasks in this phase are:

- ▶ Provide ramping support to ensure efficient partner connectivity.
- ▶ Transfer knowledge to the customer's personnel about monitoring B2B transactions and the best practices for problem determination.
- ▶ Assist in defining a support process and flow (how customers get both internal and external help when problems occur, what number to call, what e-mail address to us, to what the service level agreement (SLA) is the B2B support team willing to commit, and so forth).

Deliverables

The resulting deliverables are:

- ▶ Documented support and issue resolution process
- ▶ Ongoing support transferred to the customer

Skill sets and resource requirements

The required skills and resources are:

- ▶ Experience configuring IBM WebSphere Integration products
- ▶ Excellent troubleshooting skills
- ▶ Excellent communications skills
- ▶ Knowledge of Internet and B2B messaging protocols (HTTP, FTP, SMTP, SSL, AS1, AS2, AS3, RosettaNet, ebXML, and so forth)

- Knowledge of other IBM products related to business integration (You might need to bring in a specialist.)

3.3 Time estimates

Table 3-1 is a rough estimate of the number of hours required to complete a full B2B deployment project. The table is meant to provide a general range of hours depicting how much time a typical deployment project of this size might take. A more thorough analysis for the customer's deployment will be required during the Discovery phase to provide the customer with specific numbers for the remaining phases. Tasks can be performed by a combination of customer resources and IBM resources, or IBM can be contracted to perform all of the tasks. The effort required for a complete B2B solution will depend on the number of other integration products used for the project and how much of the solution is appliance-based (appliances typically take less time to deploy).

Table 3-1 Time estimates

| Task | Low hours estimated | High hours estimated | Resources |
|------------------------|----------------------------|-----------------------------|--|
| Phase 1: Discovery | 40 | 80 | Resource: Project Lead and Solution Architect |
| Phase 2: Planning | 40 | 200 | Resource: Project Lead and Solution Architect |
| Phase 3: Installation | 16 | 120 | Resources: B2B Specialist and Project Lead |
| Phase 4: Testing | 40 | 160 | Resources: B2B Specialist, Quality Assurance (QA) Analyst, Business User, System Administrator, and Solution Architect |
| Phase 5: Customization | 40 | 1000 | Resource: Developer, Project Lead, EDI Specialist, and Solution Architect |

| Task | Low hours estimated | High hours estimated | Resources |
|--|----------------------------|-----------------------------|---|
| Phase 6: Production Deployment | 4 | 40 | Resources: B2B Specialist, Project Lead, and System Administrator |
| Phase 7: Partner Ramping per partner (see the following Partner Ramping chart) | 4 | 80 | Resources: B2B Specialist, Project Lead, System Administrator, and Solution Architect |
| Phase 8: Support (post-production) | 80 | 160 | Resources: B2B Specialist and System Administrator |
| Phase 0: Document Knowledge Transfer (5% of the total effort) | 13 | 92 | Resources: B2B Specialist and Project Lead |
| Project Management: (Throughout the project 20% of the total effort) | 53 | 386 | Project Lead |
| Total effort: | 330 | 2318 | |

3.4 Partner Ramping Effort Estimator (hours per Partner)

Table 3-2 on page 47 estimates the time Partner Ramping takes.

Table 3-2 Partner Ramping effort estimator

| | EDI | XML | Binary | Add hours for translation config | Add hrs. for EDI map creation | Add hours for validation config | Add hours to integrate JMS | Add hours to integrate file system or POP3 | Add hours to integrate HTTP or FTP | Add Hours for proxy navigation |
|--------------------|-----|-----|--------|----------------------------------|-------------------------------|---------------------------------|----------------------------|--|------------------------------------|--------------------------------|
| AS1/SMTP | 4 | 5 | 5 | 4 | 20 | 4 | 4 | 0 | 2 | 4 |
| AS2/HTTP | 4 | 5 | 5 | 4 | 20 | 4 | 4 | 0 | 2 | 4 |
| AS2/HTTPS | 6 | 6 | 6 | 4 | 20 | 4 | 4 | 0 | 2 | 4 |
| Web Services/HTTP | 6 | 6 | 6 | N/A | 20 | N/A | 4 | 0 | 2 | 4 |
| Web Services/HTTPS | 8 | 8 | 8 | N/A | 20 | N/A | 4 | 0 | 2 | 4 |
| HTTP Post | 4 | 4 | 4 | 4 | 20 | 4 | 4 | 0 | 2 | 4 |
| HTTPS post | 6 | 6 | 6 | 4 | 20 | 4 | 4 | 0 | 2 | 4 |
| FTP | 6 | 7 | 7 | 4 | 20 | 4 | 4 | 0 | 2 | 4 |
| FTP Scripting | 10 | 12 | 12 | 4 | 20 | 4 | 4 | 0 | 2 | 4 |

Note: These hours are just rough estimates for ramping a single partner. There are many factors that can increase the effort required to successfully connect to a partner with the major issue usually revolving around interoperability and certificate management. The assumptions used when estimating these hours are:

- ▶ No interoperability issues are discovered. If issues arise because the partner is using another B2B Trading Engine or the partner has improperly configured their engine (certificates, protocol, headers, and so forth), it can take as many as 40 to 80 hours to troubleshoot the problem and ramp the partner.
- ▶ FTP, e-mail, messaging (MQ), and Web Service servers are installed and are ready to accept connections from IBM Integration applications and appliances.
- ▶ All firewalls and proxies have been configured to allow access to the systems for inbound and allow access to the Internet for outbound.
- ▶ Transformation and mapping efforts are for simple maps or XSLTs. If more complex maps are required, the effort will increase. The estimate includes creation, testing, and deployment of the map. A mapping analysis must be done for projects that have a large number of maps prior to giving estimates for the mapping effort.

3.5 Roles

Implementation of B2B integration applications and or appliances requires many skill sets. The roles described next can be assigned to a single person or multiple people depending their skill sets.

Solution Architect

The Solution Architect is responsible for working with the network and security teams to design an architecture that meets the security requirements. Key trading scenarios will be thoroughly discussed and documented and all back-end integration options will be defined. The Solution Architect will assist with areas such as firewall configuration, network configuration, and setup of day-to-day processing scripts for the administrator as needed. The Architect role is often combined with the Project Lead and B2B Specialist roles.

Project Lead/Project Manager

The Project Lead is responsible for managing the project and assisting in the development of all plans and the project schedule. The Project Lead role is often combined with the Architect and B2B Specialist role.

B2B Specialist

The B2B Specialist is responsible for the installation and configuration of the Business-to-Business Integration (B2Bi) solution, Partner Ramping, and Support. The B2B Specialist will provide knowledge transfer to personnel throughout the project and will participate at various levels in all phases of the project.

Developer

The developer is responsible for implementing and unit testing custom code (user exits), maps, and XSLT. The developer might also provide assistance during the integration phase of the project.

QA Analyst

The QA Analyst is responsible for functional and performance testing of the B2B solution to ensure it will meet the customer's business requirements. The QA Analyst role is often combined with one of the specialist roles.

System Administrator

The System Administrator is responsible for day-to-day operations of the B2B environment and will be responsible for the configuration and monitoring of the systems and software.



Aspects of B2B security

This chapter describes the various types of security to use when deploying business-to-business (B2B). In this chapter, we explain data security, connection security, deployment security, access control, and how each type of security relates to the WebSphere Business Integration Connect product. This chapter also describes common security technologies that are used to secure the B2B deployment inside the enterprise.

4.1 Overview

In many cases, when security is first discussed with the customer, they think of securing their data and possibly their connection. When it comes to deploying a B2B solution, we also need to take into consideration how to deploy the software in a manner that does not violate any existing network security policies that the customer has in place.

4.2 Areas of B2B security

The four areas of security and their definitions that apply to B2B are:

- ▶ *Deployment Security* means the placement of hardware within an existing network with access to the Internet. This type includes database servers, file shares, message queue servers, and integration servers.
- ▶ *Connection Security* means establishing a secure connection between trading participants over a Secure Socket Layer (SSL) connection.
- ▶ *Document Security* encompasses signing and encrypting the message prior to sending it to the trading partner.
- ▶ *Access Control* means providing access to data and configuration information inside the B2B application.

When organizations focus on combining these areas, security policies can be defined to establish a secure baseline so that they can trade with their partners over the Internet with greater confidence.

4.2.1 Deployment security

To protect B2B applications from unauthorized access, networking and firewall protection must be established. Firewalls work in conjunction with proxy servers, providing the ability to filter protocols, addresses, communication ports, and IP packets.

The security model that can be used is the establishment of a demilitarized zone (DMZ). The DMZ must be configured to restrict only a minimum set of communication ports for it to process requests. For a more detailed explanation of firewalls and DMZs, refer to 4.3.2, “Firewalls” on page 57.

The XB60 is a DMZ-deployable appliance and requires a minimum amount of access through the inner firewall. Any sensitive payload data persisted to the Appliance is not accessible by partners and is encrypted on the hard drive.

4.2.2 Connection security

A common method of transferring information security on the Internet is using the Secure Sockets Layer (SSL). It uses encryption that is based on the public and private key model, using authentication with basic or extended handshakes. SSL works by creating a secure connection between communicating applications over HTTP.

The SSL protocol addresses the following security issues:

- ▶ Privacy
After the symmetric key is established in the initial handshake, the messages are encrypted using this key.
- ▶ Message integrity
Messages contain a message authentication code (MAC) ensuring the message integrity.
- ▶ Authentication
During the handshake, the client authenticates the server using an asymmetric or public key.

SSL works well when securing browser-based applications, such as the administrative console in WebSphere Business Integration Connect, but it can also be useful to augment B2B document transfer.

The disadvantage of using SSL alone is that it only protects the data during the transfer process and does not continue to protect it after it has reached its destination, nor does SSL provide document integrity and nonrepudiation. These items are critical for secure electronic business transactions.

The SSL handshake

An HTTP-based SSL connection is always initiated by the client using a URL starting with `https://` instead of with `http://`. At the beginning of an SSL session, an SSL handshake is performed. This handshake produces the cryptographic parameters of the session. A simplified overview of how the SSL handshake is processed is shown in Figure 4-1 on page 54.

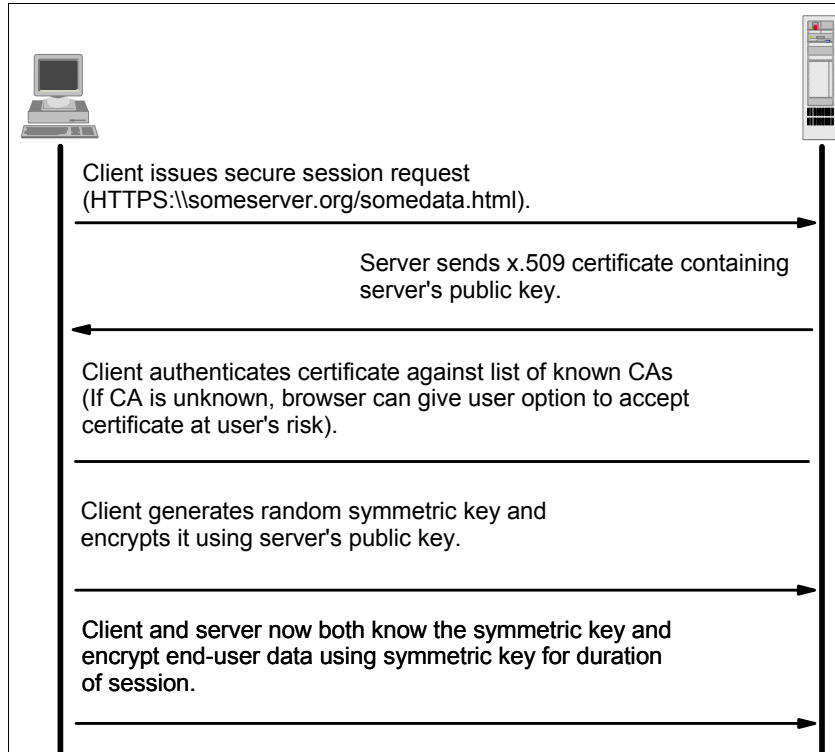


Figure 4-1 SSL handshake example

The steps are:

1. The client sends a client hello message that lists the cryptographic capabilities of the client. These capabilities are sorted in client preference order, such as the version of SSL, the cipher suites supported by the client, and the data compression methods supported by the client. The message also contains a 28-byte random number.
2. The server responds with a server hello message that contains the cryptographic method, such as cipher suite, the data compression method selected by the server, the session ID, and another random number.

Note: The client and the server must support at least one common cipher suite, or else the handshake fails. The server generally chooses the strongest common cipher suite.

3. The server sends its digital certificate. In this example, the server uses X.509 V3 digital certificates with SSL. If the server uses SSL V3, and if the server

application, for example, the Web server, requires a digital certificate for client authentication, the server sends a digital certificate request message. In the digital certificate request message, the server sends a list of the types of digital certificates supported and the distinguished names of acceptable certificate authorities.

4. The server sends a server hello done message and waits for a client response.
5. Upon receipt of the server hello done message, the client Web browser verifies the validity of the server's digital certificate and checks that the server's hello parameters are acceptable. If the server requested a client digital certificate, the client sends a digital certificate. If no suitable digital certificate is available, the client sends a no digital certificate alert. This alert is only a warning, but the server application can fail the session if client authentication is mandatory.
6. The client sends a client key exchange message. This message contains the pre-master secret, a 46-byte random number used in the generation of the symmetric encryption keys and the message authentication code (MAC) keys, encrypted with the public key of the server. If the client sent a digital certificate to the server, the client sends a digital certificate verify message signed with the client's private key. By verifying the signature of this message, the server can explicitly verify the ownership of the client digital certificate.

Note: An additional process to verify the server digital certificate is not necessary. If the server does not have the private key that belongs to the digital certificate, it cannot decrypt the pre-master secret and create the correct keys for the symmetric encryption algorithm. The handshake fails.

7. The client uses a series of cryptographic operations to convert the pre-master secret into a master secret, from which all key material required for encryption and message authentication is derived. Then, the client sends a change cipher spec message to make the server switch to the newly negotiated cipher suite. The next message sent by the client, the finished message, is the first message encrypted with this cipher method and keys.
8. The server responds with a change cipher spec and a finished message of its own.
9. The SSL handshake ends, and the encrypted application data can be sent.

4.2.3 Document security

Document security is normally accomplished through digital certificates, which provide an online identification credential for specific document exchanges, for example, AS1, AS2, AS3, RosettaNet, or custom document-level encryption

requirements. As part of document exchange, digital signatures can be calculated on the electronic document using public key cryptography. Through this process, the digital signature is tied to the document being signed, as well as to the signer, and cannot be reproduced. With the passage of the federal digital signature bill, digitally signed electronic transactions have the same legal weight as transactions signed in ink.

Document security provides the following features:

- ▶ Privacy
A document is encrypted by the recipient's public key. Only the recipient has the appropriate private key to decrypt the message.
- ▶ Authentication
The recipient can authenticate the sender of a document by verifying a digital signature.
- ▶ Integrity
A digital signature of the document provides document integrity.
- ▶ Nonrepudiation
Nonrepudiation is provided using digital signatures and encrypting the hash value with the receiver's private key then sent back to the sender, which provides a digital receipt to the sending party.

In the XB60, business documents are typically signed and encrypted before leaving the security of the sender's network. Every partner that is set up in the XB60 can, and typically will, have an X.509 certificate used to encrypt documents and validate signatures on documents received from that trading partner.

4.2.4 Access control

The XB60 uses a role-based approach for access control. Users log in by providing their partner name, user id, and password. The login determines individual access privileges. The XB60 browser interface, which is used for administering functions, operates over an SSL connection.

4.3 Security technologies

In this section, we describe several security technologies that can be used when deploying applications that need to be accessed from the Internet.

4.3.1 Reverse proxy server

A *reverse proxy*, a common form of a proxy server, is generally used to pass requests from the Internet through a firewall to isolate private networks. It is used to prevent Internet clients from having direct, unmonitored access to sensitive data residing on internal servers on an isolated network, or intranet. One advantage of using a reverse proxy is that Internet clients do not know that their requests are being sent to and handled by a reverse proxy server, which allows a reverse proxy to redirect or reject requests without making Internet clients aware of the actual content servers on a protected network. Refer to Figure 4-2.

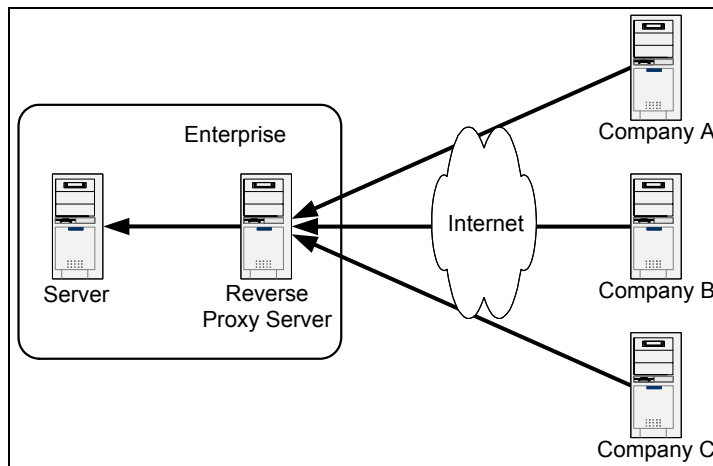


Figure 4-2 Reverse proxy server

4.3.2 Firewalls

A *firewall* is a system that enforces an access control policy between two or more networks. The actual means by which this enforcement is accomplished varies widely. In principle, the firewall can be thought of as a pair of mechanisms: one exists to block traffic, and the other exists to permit traffic. The most important thing to recognize about a firewall is that it implements an access control policy.

Firewalls section off different communication zones to the Internet. These zones are called *demilitarized zones* (DMZs). In the context of firewalls, the DMZ refers to a part of the network that is neither part of the internal network nor directly part of the Internet. Typically, this zone is the area between your Internet connection and your host server, although it can be between any two policy-enforcing components of the network. Refer to Figure 4-3 on page 58.

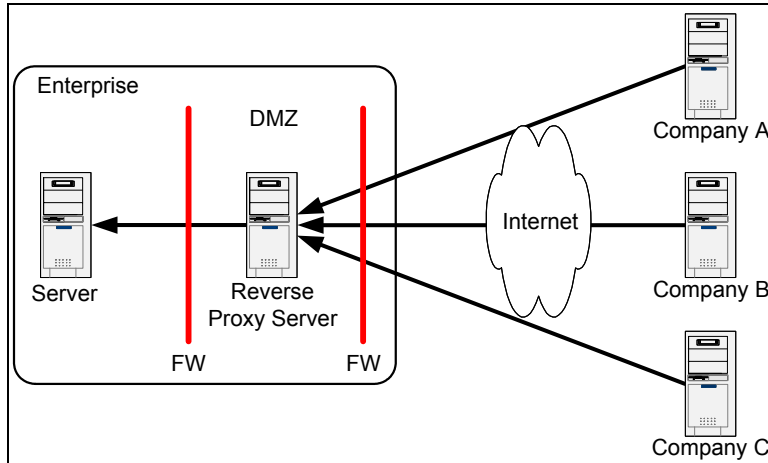


Figure 4-3 DMZ example

4.3.3 Network Address Translation

Network Address Translation (NAT) is a commonly used IP translation and mapping technology. It is a technology that allows networks to use other networks or share Internet access. Using a device or piece of software that implements NAT allows an entire network to share a single Internet connection over a single IP address. Refer to Figure 4-4. Additionally, NAT keeps the network fairly secure from hackers by hiding the private IP address.

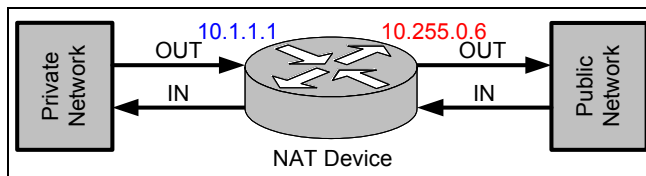


Figure 4-4 Network Address Translation

NAT acts as an interpreter between two networks. In the case of an organization, it can sit between the Internet and your internal network. The Internet is considered the public side and the internal network is considered the private side. When a computer in the private side requests data from the public side, the NAT device will open a little conduit between your computer and the destination computer. When the public computer returns results from the request, the results are passed back through the NAT device to the requesting computer.

4.3.4 Port Address Translation

Port Address Translation (PAT) provides a similar functionality to NAT, but PAT is a more specific tool. PAT forwards requests for a particular IP address and port pair to another IP address and port pair, as in Figure 4-5. This feature is commonly used with a reverse proxy scenario to hide all the content server's physical details.

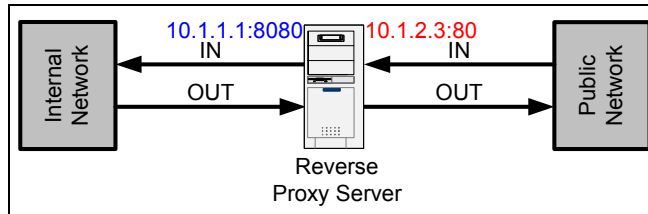


Figure 4-5 *Port Address Translation*



WebSphere DataPower B2B Appliance XB60

In this chapter, we introduce the IBM WebSphere DataPower B2B Appliance and discuss how the B2B Appliance redefines the boundaries of middleware by extending the IBM service-oriented architecture (SOA) Foundation with a specialized, consumable, dedicated SOA appliance that combines B2B standards, simplified integration, superior performance, and hardened security for SOA implementations.

5.1 Why an Appliance for B2B

As organizations move to an on demand business by implementing a service-oriented architecture (SOA), the largest barriers continue to be implementation complexity, cost-prohibitive scalability, and data security. To recognize improved time to value, companies are looking for opportunities to consolidate functions into single purpose-built solutions that provide exceptional performance, easy deployment, ease of use, and a low total cost of ownership.

Business processes extend across the supply chain and value chains on a global basis, and companies depend upon trading partners to run their businesses smoothly. Processes and rules change constantly, and you must quickly disseminate these changes to your trading partners. To keep pace, it is crucial that you strengthen your trading-partner relationships through tighter automated integration, so that you can make your trading partners an extension of your enterprise.

The global economy, large outsourcing initiatives, and the virtual supply chain have blurred the boundaries of the enterprise and the distinction between public and private processes. As a result, internal application-to-application (A2A) and external business-to-business (B2B) technologies are converging. Many enterprises are seeking a single business integration platform to meet all of their internal and external messaging needs to reduce duplication of effort and to increase the speed of “externalizing” internal processes. The appliance model enables strong B2B business value by accelerating the pace of innovative value-creating processes and strategic initiatives. You can utilize B2B services to quickly and securely connect to your external partners and integrate the partner connections to your application integration flows, all in a purpose-built hardware solution.

To take advantage of the improved business processes, flexibility, and IT efficiency that come with moving to B2B Appliances, organizations require pervasive, scalable services and controls, robust security, and high service assurances in their infrastructures. Today, enterprises often find themselves struggling to deliver these critical requirements without having to handle prohibitive costs, complexity, and hard-to-manage infrastructures. Addressing these challenges requires a pragmatic approach, one that simultaneously recognizes the evolution of standards, the value of existing infrastructure investments, your organizational challenges, and how performance can be affected across applications.

5.1.1 SOA appliances simplify SOA deployment

By integrating many core functions required for adopting B2B, SOA, or Web services into a single, purpose-built device with enterprise service bus (ESB) capability, WebSphere DataPower B2B Appliance XB60 simplifies an overall B2B/SOA infrastructure. It is designed to deploy easily into an existing environment as an inline network device. You gain business value without having to change your network or application software. As a result, proprietary schemas, coding, or application programming interfaces (APIs) are not required to install or manage the device.

5.1.2 Drop-in integration for heterogeneous environments

As a core offering in the IBM B2B and ESB product portfolio, WebSphere DataPower B2B Appliance XB60 is a purpose-built hardware B2B-enabled ESB for simplified deployment and hardened security with the ability to quickly transform data between a wide variety of formats, including XML, industry standards, and custom formats. The device provides core B2B functions, including AS2 and AS3 messaging, partner profile administration, routing of electronic data interchange (EDI), XML, and binary payloads, auto archiving and purging of B2B transactions, and B2B transaction viewing capabilities. The ESB functions include routing, bridging, transformation, and event handling. It provides a reliable, performance-oriented solution to many integration challenges. Because it is not limited to handling just XML, WebSphere DataPower B2B Appliance XB60 resonates with IT organizations that need to benefit from the connectivity of SOA deployments but must also deal with managing a combination of multiple proprietary, industry, company-specific, and existing data formats. The device is a true drop-in B2B integration point for such environments, reducing the time and cost of integrations and speeding the time to market for services.

5.1.3 Innovative enablement of existing infrastructure for XML and Web services

For accelerated, security-rich integration capabilities, WebSphere DataPower B2B Appliance XB60 provides transport mediation, routing, and transformations among binary, text, and XML message formats. Visual tools can be used to describe data formats, create mappings between different formats, and define message flows. With native connectivity to IBM DB2® and IBM System z® technology, the device offers an innovative solution for security-rich XML enablement of existing systems and mainframe connectivity.

5.1.4 Policy-driven approach to Web services management and SOA governance

By centralizing management tasks and policy enforcement for Web services and decoupling them from applications, your SOA infrastructure increases in flexibility and scalability while simultaneously offering you improved insight, visibility, and control. By moving certain functions onto WebSphere DataPower B2B Appliance XB60 (such as protocol bridging, AS2/AS3 message processing, profile management, Web services management, security processing, and policy enforcement), IT architects, operations, security personnel, and business personnel can decouple these functions from core business applications. This capability helps to simplify development, deployment, and manageability.

5.1.5 Integration with registry and repository, security, identity, and service management software

WebSphere DataPower B2B Appliance XB60 integrates with a variety of registry and repository, security, identity, and service management software. Coupled with access-control software, such as IBM Tivoli® Access Manager, the device enforces fine-grained access controls. Working with IBM Tivoli Federated Identity Manager, the device provides federated identity and policy management for Web services between organizations and enterprises. Using a registry and repository, such as IBM WebSphere Services Registry and Repository, you can discover and reuse services and configure new services for policy and security enforcement performed by WebSphere DataPower B2B Appliance XB60. The combination of these applications and the robust XB60 security features provides the comprehensive capabilities for B2B/SOA security and Web services management that enterprises increasingly require.

5.1.6 Support for advanced Web services standards and interoperability

IBM recognizes that SOA must address the need to integrate heterogeneous environments both within and outside the enterprise. The WebSphere DataPower SOA appliance portfolio has a long-standing history of support for key and advanced standards, including WS-Security, WS-Policy, WS-Reliable Messaging, SOAP, Web Services Distributed Management (WSDM), WS-I Profiles, WS-Addressing, eXtensible Access Control Markup Language (XACML), Security Assertion Markup Language (SAML), Secure Socket Layer (SSL), and proprietary Single Sign-on (SSO) tokens.

In addition, WebSphere DataPower SOA appliances support interoperability with Universal Description, Discovery, and Integration (UDDI) registries, and databases, such as Oracle® and Sybase.

5.1.7 IBM SOA Foundation for Smart SOA deployments integration

WebSphere DataPower B2B Appliance XB60 has broad and deep integration across the IBM SOA Foundation. As a result, it contributes to what IBM calls the Smart SOA approach, a set of guiding principles that benefit both business and IT, eliminating unnecessary complexity while building a strong foundation for future growth. Integration of WebSphere DataPower B2B Appliance XB60 with popular integrated development environments, such as the IBM Rational® portfolio, reduces the time that you have to spend in development and debugging. In addition to interoperability, the device also features deep integration with products, such as IBM WebSphere MQ, IBM WebSphere Enterprise Service Bus, IBM WebSphere Message Broker, and IBM DB2 to help process SOA transactions in a faster, more secure, and simplified way. Additionally, the XB60 enables you to take advantage of IBM self-management capabilities for autonomic computing, creating infrastructures that require minimal intervention, which can help lower cost of ownership and improve service availability.

5.2 Easily connect to trading partners using industry standards

You can easily connect to your trading partners using the industry standards.

5.2.1 IBM WebSphere DataPower B2B Appliance XB60

IBM WebSphere DataPower B2B Appliance XB60 simplifies, helps secure, and accelerates your B2B trading partner connectivity.

The XB60 is a purpose-built B2B Gateway for simplified deployment and hardened security. This 1U (1.75 inch thick) rack-mountable network device is powered by unique technology to help your business:

- ▶ Easily manage and connect to trading partners using industry standards
- ▶ Extend integration beyond the enterprise with a securely deployed B2B Gateway in the DMZ
- ▶ Improve the performance and scalability of B2B interfaces

- ▶ Govern B2B integration points through consolidated trading partner management

The XB60 builds on top of the DataPower Application Integration appliance by adding trading partner profile management, B2B transaction viewing capabilities, and industry standards-based B2B messaging protocols to the already robust integration capabilities of the core appliance. These three key capabilities are at the heart of the B2B Appliance. They are designed in such a way that the B2B Appliance is positioned extremely well to handle simple partner connections with data passing through directly to end applications for further processing. If more complex data flows are required, the application integration capabilities to the XB60 can be used to perform data validation, transformation, rules-based enforcement, and content-based routing.

The B2B Gateway Service is a configuration object that is responsible for processing and routing B2B data. Partner profiles are configuration objects that are capable of supporting multiple destinations; the profiles are associated with any number of B2B Gateway Services. The B2B Viewer is used to view all transactions that pass through a B2B Gateway Service.

The components that make up the B2B Gateway Object in the XB60 are depicted in Figure 5-1 on page 67.

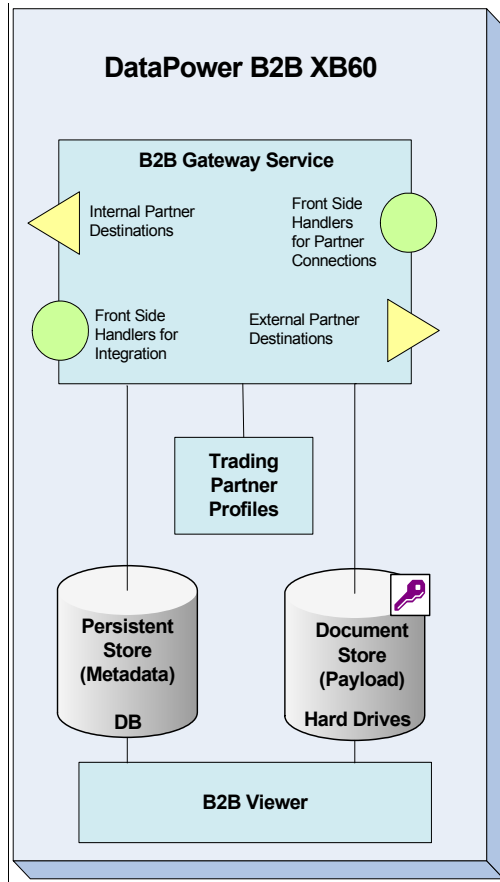


Figure 5-1 XB60 B2B components

Tip: The XB60 AS2 Trading Tutorial, which is designed for the beginner, is provided in Appendix A, “Additional material” on page 389. It will walk you through the configuration of four partner profiles and two gateways that will then be used in five testing scenarios where you get a chance to also use the B2B Transaction Viewer to see the status of your B2B transactions.

5.2.2 How Data flows through the B2B Gateway Service

The B2B Gateway Service was designed to route B2B data inbound and outbound over industry standard B2B protocols. It uses sender and receiver IDs inside the messages to route documents based on trading partner agreements. Each trading partner agreement is configured by setting attributes within a

profile. Profiles are configured and stored using the XB60 B2B Partner Profile interface. The Partner profile holds a list of IDs used by the partner, x509 Certificates/keys associated with the partner, destinations for routing data, and basic contact information.

The B2B Appliance stores metadata in an embedded database, and all documents are stored in a directory (By default, the encrypted portion of the local RAID 1 volume is used; however, an external directory can also be used.). This information is persisted to provide state management, automatic and manual resends, and the ability to view B2B transactions in the B2B Transaction Viewer.

The diagrams in this section depict how data flows through the B2B Gateway Service.

Inbound AS2 data flow with MDN processing

Incoming documents that are wrapped in an AS2 envelope flow with Message Disposition Notification (MDN) through the B2B Gateway Service as depicted in Figure 5-2.

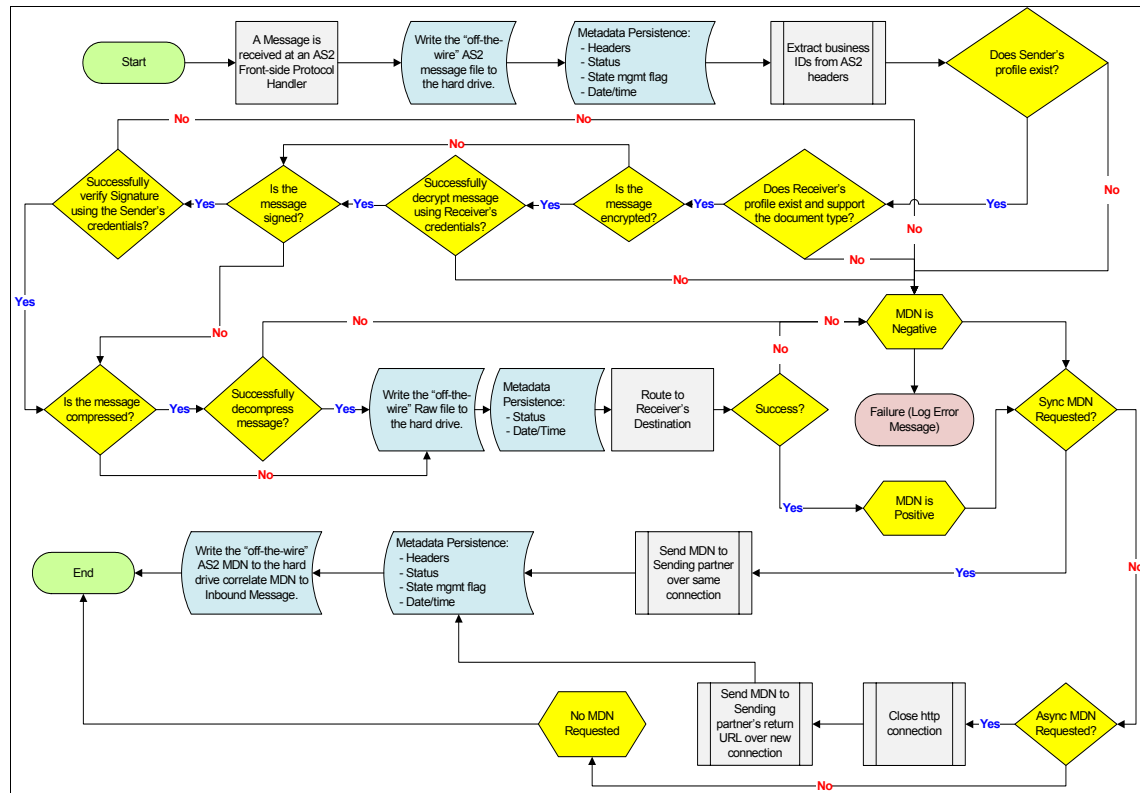


Figure 5-2 Inbound AS2 data flow

Outbound AS2 data flow with MDN processing

Outgoing documents that are wrapped in an AS2 envelope flow through the B2B Gateway Service as depicted in Figure 5-3 on page 70.

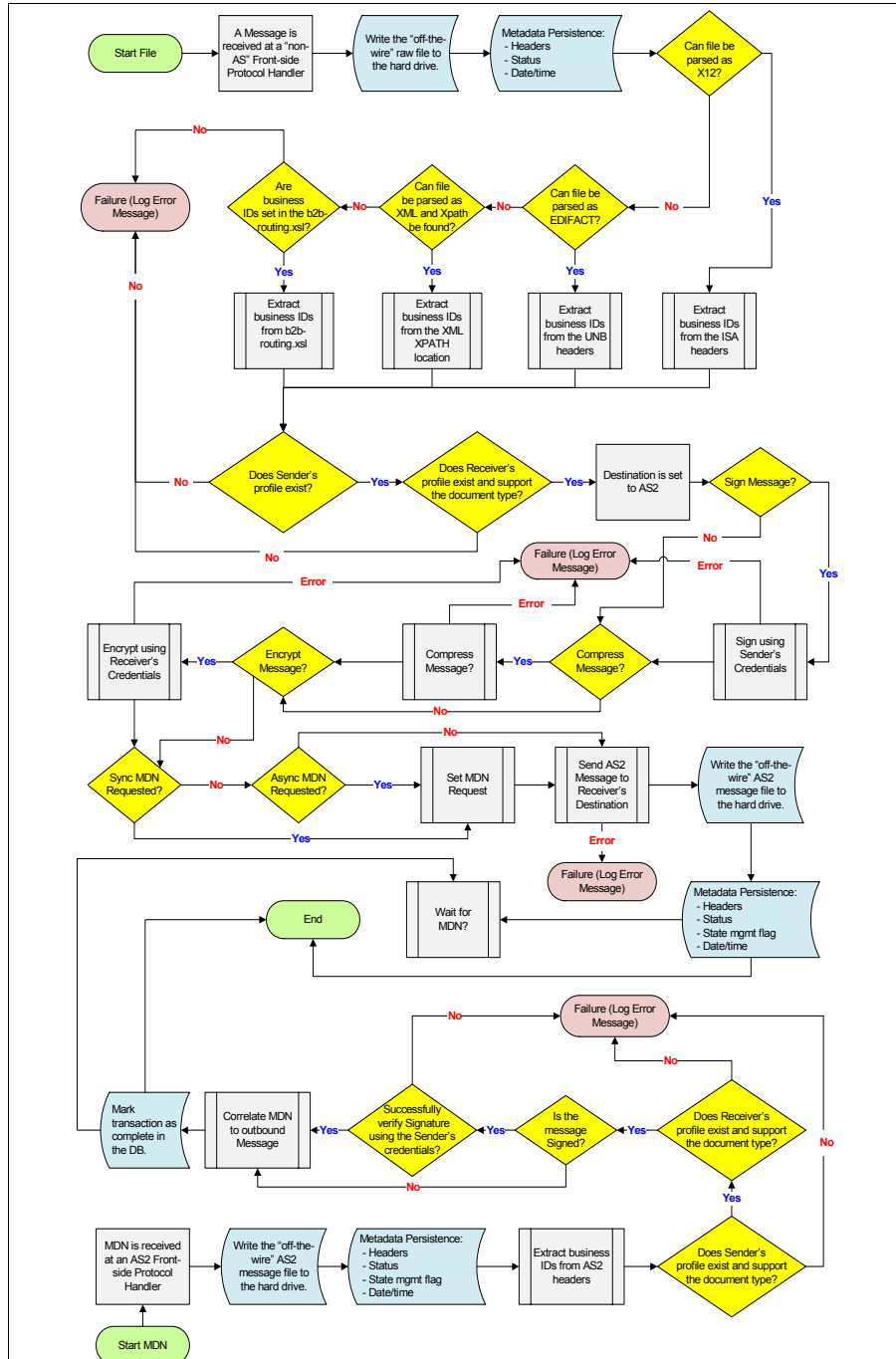


Figure 5-3 AS2 outbound data flow

Inbound AS3 data flow with MDN processing

Incoming documents that are wrapped in an AS3 envelope flow through the B2B Gateway Service as depicted in Figure 5-4.

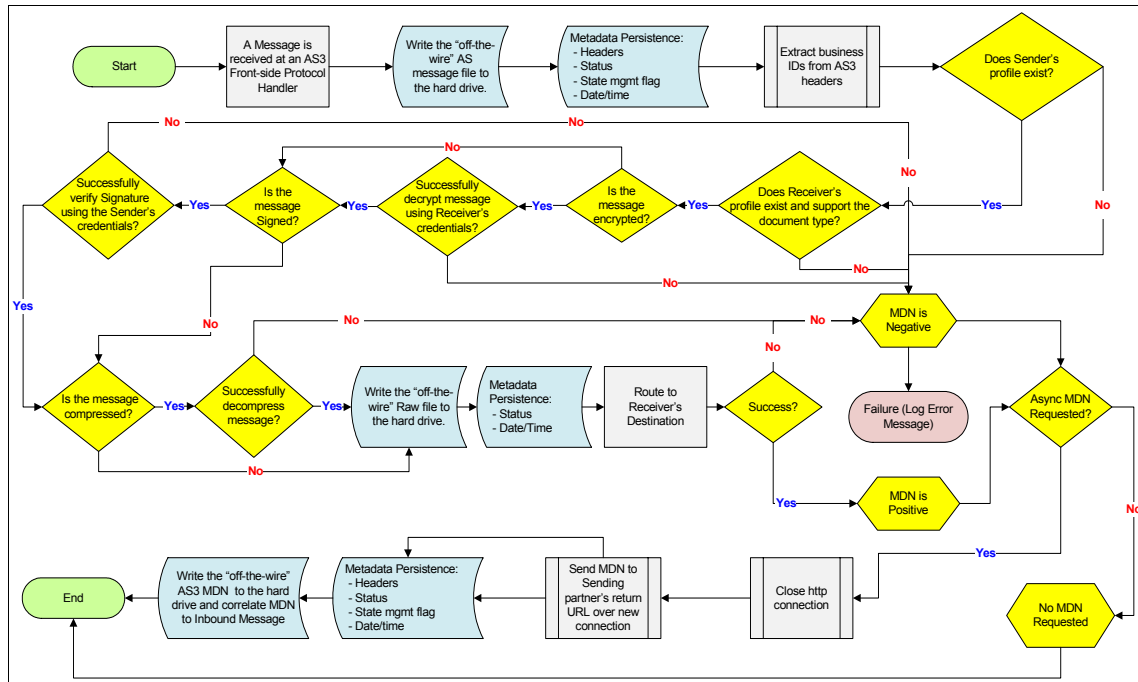


Figure 5-4 Inbound AS3 data flow

Outbound AS3 data flow with MDN processing

Outgoing documents that are wrapped in an AS3 envelope flow through the B2B Gateway Service as depicted in Figure 5-5 on page 72.

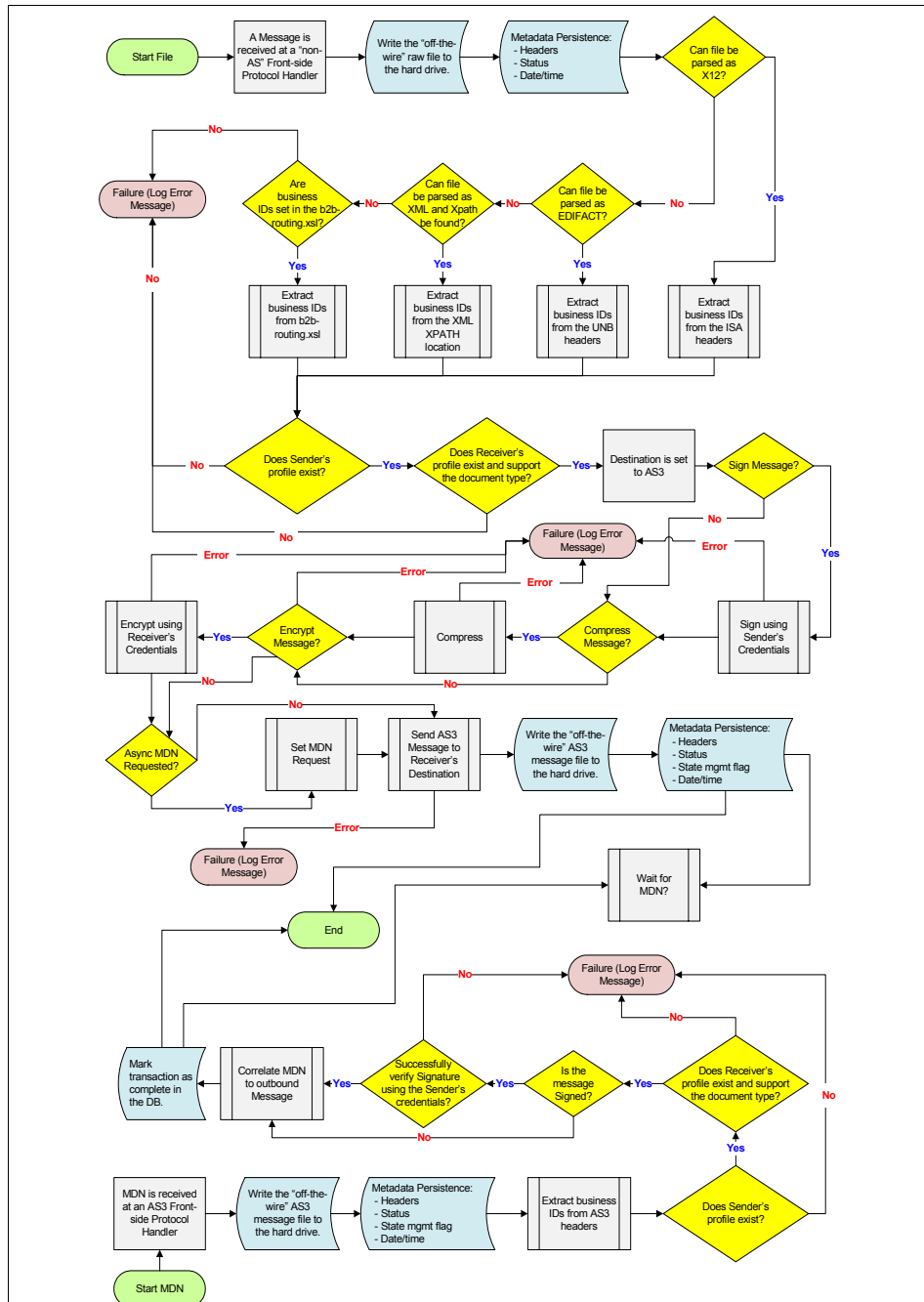


Figure 5-5 Outbound AS3 data flow



Part 2

Getting started with the XB60

Part 2 of this book gives you a good overview how to configure specific business-to-business (B2B) function in the XB60, it contains five additional chapters covering configuration information, performance testing and troubleshooting.

- ▶ Chapter 6 describes device setup and common admin tasks.
- ▶ Chapter 7 discusses configuration management of the XB60.
- ▶ Chapter 8 demonstrates configuration options specific to the XB60.
- ▶ Chapter 9 presents some common troubleshooting tips.



Device setup and administrative tasks

This chapter describes how to initialize and define the base configuration for the business-to-business (B2B) XB60 appliance. Because the device is locked down by default, the initialization of network interfaces is an essential first step. The command line interface is used to perform this step over a serial cable.

Configuration of the hard drive is covered in the startup method and the manual procedure. The subject of specifying application domains, groups, and users is introduced and followed by appliance backup instructions.

6.1 Initializing the device

By default, the device is locked down. Enablement of the device interfaces is accomplished using the serial port on the device. Connect a serial cable to the device prior to switching the power ON. Also, connect the power cords for both power supplies in order to take advantage of power supply redundancy and to avoid getting repeated error messages in the appliance log. If the terminal or personal computer is not equipped with a serial port, acquire and use a USB-to-serial cable. Ensure that the terminal or personal computer is configured for standard 9600 baud, 8N1, and no flow control. Refer to Chapter 4 of the Installation Guide for more specific details. It is a good idea to read this short chapter prior to the installation. You can obtain the 9235-Installation.pdf either in hard copy in the box when your device arrives or you can download it from the IBM WebSphere DataPower Product Documentation Portal Web page:

<http://www-01.ibm.com/support/docview.wss?rs=2362&uid=swg21377654>

To initialize the device:

1. If the device is not turned on, toggle the power switch to the ON position. The power switch is on the rear of the device. You will see the green power light-emitting diode (LED) illuminate on the front of the device and hear the fans start up. You will hear the fans change speed as the following information appears on the display: DPOS
2. Wait for a few seconds for the device to boot and the prompt `login as:` to appear.
3. Press the Enter key, because this first prompt will not accept input and always yields the message “Unauthorized access prohibited.”
4. Now enter `admin`. The `Password:` prompt appears.
5. Type the default password `admin`.
6. Read and accept the license agreement.
7. After accepting the license agreement and the proof of entitlement, the `Please enter new password:` prompt appears.
8. Type a new password. The `Please re-enter new password to confirm:` prompt appears.
9. Enter the new password again.
10. After completing these steps, the terminal will display the appliance model, version information, and serial number.

6.2 Defining the base configuration

There are two methods to define the base configuration: the manual procedure and the startup command to invoke the DataPower Installation Wizard. Both methods use the command line interface (CLI). We recommend that you use the startup procedure, because it provides the most user assistance. We describe this approach first. If necessary, the manual procedure is provided in case there is a need to make changes or if you prefer this approach.

Important: *When setting up RAID disk drive support for document and metadata storage (used in B2B Gateways), unpredictable results can occur, such as the device becoming unresponsive for up to two hours.*

To avoid unpredictable behavior and delays:

1. Make sure that the device has no other activity during this process.
2. Failing to initialize the file system on the disk, or not waiting until the initialization of the filesystem completes, before enabling the RAID volume can have “unpredictable” results. The system is attempting to mount a disk full of random data.
3. It will take up to an hour for the contents of the primary drive to be copied to the secondary drive (synchronization) after you receive the “Action Completed Successfully” message when initializing or synchronizing the RAID volume. During this time, the performance of the disk array might be slower than normal. Do not touch the device during this process. Refer to 6.2.4, “Checking and managing storage” on page 83 for directions to check the progress of synchronization.
4. *The directory name must not be the same as any domain name.* Ensure that the name chosen will never be used as a domain name.

Refer to the TechNote *Setting up the RAID support on a 9235 device with optional hard drives* for further details. You can read this TechNote at:

<http://www-01.ibm.com/software/integration/datapower/support/>

Search on 1358544 in the Search Support field.

6.2.1 Startup method

Configuring the network interface

Prior to beginning the startup procedure, obtain all the necessary network information, including static IP addresses, subnet masks, default gateway address, and Domain Name System (DNS) address.

In this example, we make the following assumptions:

- ▶ You connected your network to the MGMT Ethernet interface on the DataPower device®.
- ▶ The IP address and subnet mask for the Ethernet interface is 10.10.13.35/24.
- ▶ The IP address of the default gateway for the Ethernet interface is 10.10.13.7.
- ▶ The IP address of the DNS is 10.10.13.2.
- ▶ The specialized HTTP server to support WebGUI access listens on port 9090.
- ▶ The CLI is accessed via Secure Shell (SSH) over port 22.

While using the wizard, the default reply contained in the brackets is indicated by the Enter key. Example 6-1 shows both default responses and keyed-in responses.

Example 6-1 Startup steps 1 and 2

```
Step 1 - Do you want to configure network interfaces? [y]:y
```

To perform these tasks, you will need the following information:

- (1) The interfaces that are physically connected
- (2) The physical interface mode (Auto, Full-Duplex, and so forth)
- (3) Whether to use DHCP or a static IP address and subnet mask
- (4) The IP address of the default gateway

```
Do you have this information? [y]:y
```

```
Do you want to configure the mgt0 interface? [y]:y
```

```
Do you want to configure the eth0 interface? [y]:n
```

```
Interface configuration mode (mgt0 )
```

```
Choose one of the following interface modes:
```

- 1 - Auto
- 2 - 1000baseTx-FD
- 3 - 100baseTx-FD
- 4 - 100baseTx-HD
- 5 - 10baseT-FD
- 6 - 10baseT-HD

```
Enter the number of the interface mode [1]: 1
Interface operational parameters set (Auto)
Do you want to enable DHCP? [y]:n
Enter the IP address for the interface: 10.10.13.35
Enter the subnet mask: 255.255.255.0
Operation succeeded
Enter the IP address for the default gateway: 10.10.13.7
Operation succeeded
Do you want to configure the eth1 interface? [y]:n
Do you want to configure the eth2 interface? [y]:n

Step 2 - Do you want to configure network services? [y]:y
Do you want to configure DNS? [y]:y

This configuration requires the IP address of the DNS server.

Do you have this information? [y]:y
Enter the IP address of the DNS server: 10.10.13.2
Modify DNS Settings configuration
```

Configuring management access

It is crucial to set up access for administering and configuring the DataPower appliance after it is initialized in the network (Example 6-2). The CLI operates over SSH on port 22 by default, and the WebGUI operates over HTTPS on port 9090 by default.

Example 6-2 Startup step 3

```
Step 3 - Do you want to configure management access? [y]:y
```

These configurations require the IP address of local interface that manages the appliance.

```
Do you have this information? [y]:y
Do you want to enable SSH? [y]:y
Enter the local IP address [0 for all]: 0
Enter the port number [22]: 22
```

```
% Pending
```

```
SSH service listener enabled
Do you want to enable WebGUI access [y]:y
Enter the local IP address [0 for all]: 0
```

Enter the port number [9090]: 9090
Modify Web Management Service configuration

Configuring the hard drive

The hard disk array configuration requires patience, because it takes time to build the array and the volume and to synchronize the mirrored hard drive. The synchronization process can take about an hour. Completion of the startup wizard does not depend on the completion of the synchronization of the hard drives.

Example 6-3 shows the dialog for configuring the hard drive.

Example 6-3 Startup step 4

Step 4 - Do you want to configure the hard disk array? [y]:y

This configuration is required for a fully functional B2B appliance.

This configuration requires the name of the file system to mount. Data in this file system will be available in the local: directory.

Attention: This action destroys all data on array volume.

Do you want to continue? [y]:
Enter name for the file system [ondisk]: ondisk
Modify Hard Disk Array configuration
Array volume successfully initialized.

Reviewing and saving the base configuration

In this final step of the startup wizard, the base configuration is displayed, checked, and saved. Changes to the hard disk in Step 4 are saved immediately, and you do *not* have an option to back out the changes. If changes need to be made, there is no provision to delete an existing setting. However, there is an option to overwrite an existing setting in a subsequent pass through the startup wizard. We recommend that you make changes to the configuration after this initial pass by using the manual procedure, which we describe in 6.2.2, “Manual procedure” on page 81. Example 6-4 on page 81 provides a typical configuration of the IP addresses.

Example 6-4 Startup step 5

```
Step 5 - Do you want to review the current configuration? [y]:y

  interface          IP Address          RX (kb/pkts/errs) TX (kb/pkts/errs)
  -----          -
mgt0                10.10.13.35/24      8567/98196/0        132603/110001/0
eth0 0/0/0          0/0/0
eth1 0/0/0          0/0/0
eth2 0/0/0          0/0/0

Do you want to save the current configuration? [y]:
Overwrite previously saved configuration? [y/n]: y
Configuration saved successfully.
You have completed the Installation Wizard.
xb60(config)#
```

6.2.2 Manual procedure

Prior to beginning the manual procedure, make sure to obtain all the necessary network information, including static IP addresses, subnet masks, the default gateway address, and the DNS address.

The default reply contained in the brackets is indicated by the Enter key. In this example, we make the following assumptions:

- ▶ You connected your network to the MGMT Ethernet interface on the DataPower device.
- ▶ The IP address and subnet mask for the Ethernet interface is 10.10.13.35/24.
- ▶ The IP address of the default gateway for the Ethernet interface is 10.10.13.7.
- ▶ The IP address of the DNS is 10.10.13.2.
- ▶ The specialized HTTP server to support WebGUI access listens on port 9090.
- ▶ The CLI is accessed via SSH over port 22.

Configuring the network interface and management access

To define the base configuration, use the following procedure from a CLI session (which is run over the serial port during this phase of the device initialization):

1. Access Global configuration mode by entering the following command:
configure terminal

2. Access the configuration mode for the MGMT Ethernet interface by entering the following command:

```
int mgt0
```

3. Configure the IP address and subnet mask for the Ethernet interface by entering the following command:

```
ip address 10.10.13.35/24
```

4. Configure the IP address of the default gateway for the Ethernet interface by entering the following command:

```
ip default-gateway 10.10.13.7
```

5. Exit the configuration mode for the Ethernet interface by entering the following command:

```
exit
```

6. Define the specialized HTTP server to support WebGUI access by entering the following command:

```
web-mgmt 10.10.13.35 9090
```

7. Enable SSH by entering the following command:

```
ssh 10.10.13.35
```

8. Save the changes to the configuration by entering the following command:

```
write memory
```

9. You are prompted to confirm that you want to overwrite the existing configuration. Enter y.

Configuring the hard drive

The hard disk array configuration requires patience, because time is required to build the array and the volume and to synchronize the mirrored hard drive. The synchronization process can take about an hour. Completion of this step continues after you complete and exit the CLI, because synchronization of the hard drives can take about an hour:

```
xb60# config
```

```
xb60(config)#raid-initialize raid0
```

```
xb60(config)# raid-volume-initialize-filesystem raid0 (this step will take a few minutes)
```

```
xb60(config)# raid-volume raid0
```

```
xb60(config raid-volume raid0)# directory ondisk (The directory name appears in the file manager. This name must not be the same as any domain name.)
```

```
xb60(config raid-volume raid0)# admin-state enabled
```

```
xb60(config raid-volume raid0)# exit
```

```
xb60(config)# write mem
```

When prompted about overwrite {y/n,}, y

Exit the command line session by entering the following command: **exit**.

6.2.3 Verifying the configuration

Continuing the examples one step further, because the IP address for the Ethernet interface is 10.10.13.35 and the specialized HTTP server to support WebGUI access listens on port 9090, we access the WebGUI from any browser, using the following procedure:

1. Open the Web browser.
2. Type the following value in the Address field: `https://10.10.13.35:9090`. If the Web page displays successfully, the base firmware configuration is successful.
3. Log in to the device with the local admin account and password. If the DataPower Control Panel displays, authentication of the local admin account is successful.

You can use the WebGUI or the CLI to verify the successful configuration of persistent storage (the hard drives) and to perform additional interface, network, and service configurations.

6.2.4 Checking and managing storage

The appliance uses persistent storage to store metadata about transactions. Access to the configuration of the B2B persistence object is limited to administrators in the default domain. To properly plan for archiving and the persistence store size, you need to estimate the traffic characteristics for your appliance. Refer to the “Managing Storage” chapter of the *B2B Developers Guide*, for a detailed discussion of how to calculate capacity. The *B2BDevelopersGuide.pdf* is provided on the CD shipped with the device or it can be downloaded from the IBM WebSphere DataPower Product Documentation Portal Web page:

<http://www-01.ibm.com/support/docview.wss?rs=2362&uid=swg21377654>

By default, electronic data interchange (EDI) documents transferred through B2B Gateway objects are stored on the appliance hard disk and can eventually be transferred to a permanent archive. Each B2B Gateway object shares document store space. Additionally, metadata that is extracted from each transaction is retained in a separate B2B persistence store on the “onboard” hard disk. Refer to Figure 6-1.

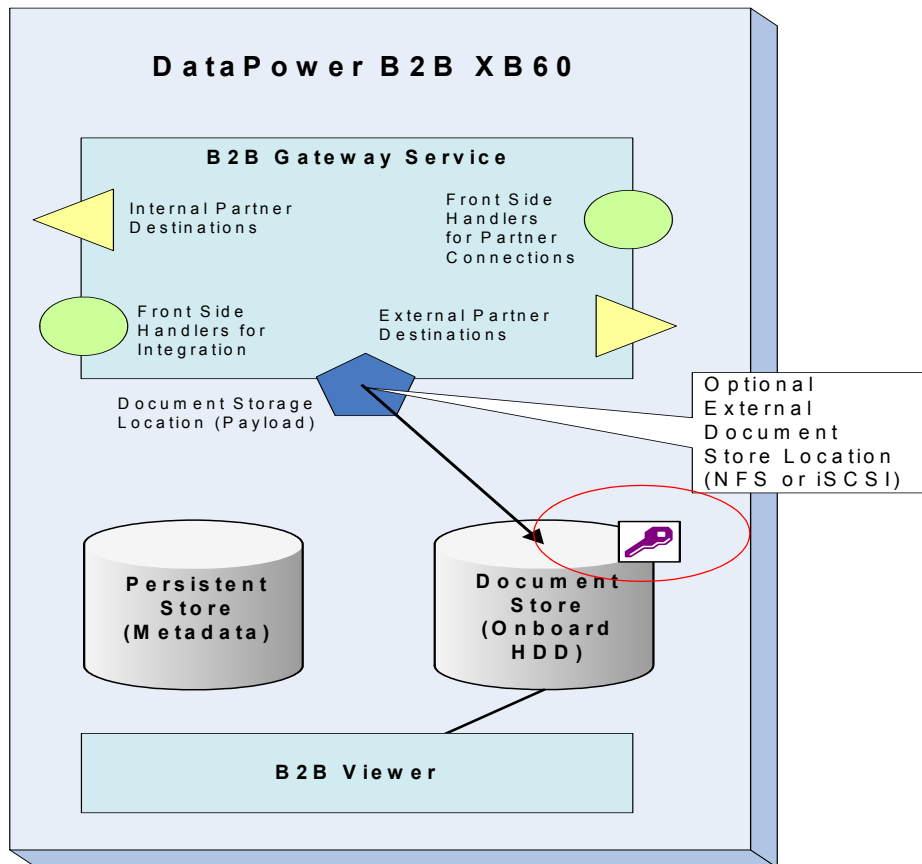


Figure 6-1 Specifying document storage

Checking synchronization progress

Time is required to synchronize the mirrored hard drives. It is important to wait on further configuration and testing until after the synchronization process completes.

Refer to Figure 6-2 on page 85 for an example of how to check the synchronization progress. From the left navigation menu, select **Status** → **RAID Volumes** to display the RAID Volumes' status.

The screenshot shows the RAID Volumes control panel. On the left is a navigation menu with sections: Status, View Logs, Main, Configuration, and System. The main area displays a table of RAID volumes. Below the table are links for 'Refresh Status' and 'Help'.

| Number | Type | Volume ID | Disks | State | Enabled | Quiesced | Resync in Progress | Resync Percentage | Inactive Status | Bad Block Table Full |
|--------|--------|-----------|-------|----------|---------|----------|--------------------|-------------------|-----------------|----------------------|
| 0 | RAID-1 | 0 | 2 | Degraded | True | False | True | 24 | Active | False |

Figure 6-2 Resync in progress

Managing store size

To check and optionally change persistent storage on the appliance, use the following procedure:

1. Select **Objects** → **System Settings** → **B2B Persistence**.
2. Place the object in an inactive administrative state by clicking **disabled**.
3. Select **raid0** from the RAID Volume drop-down list box.
4. Specify the maximum size for the store in megabytes in the Storage Size field. Use an integer in the range of 1024 up to 65536 megabytes. The default is 1024 megabytes. A good starting value in a development environment is 4096 megabytes. Figure 6-3 on page 86 depicts this configuration.

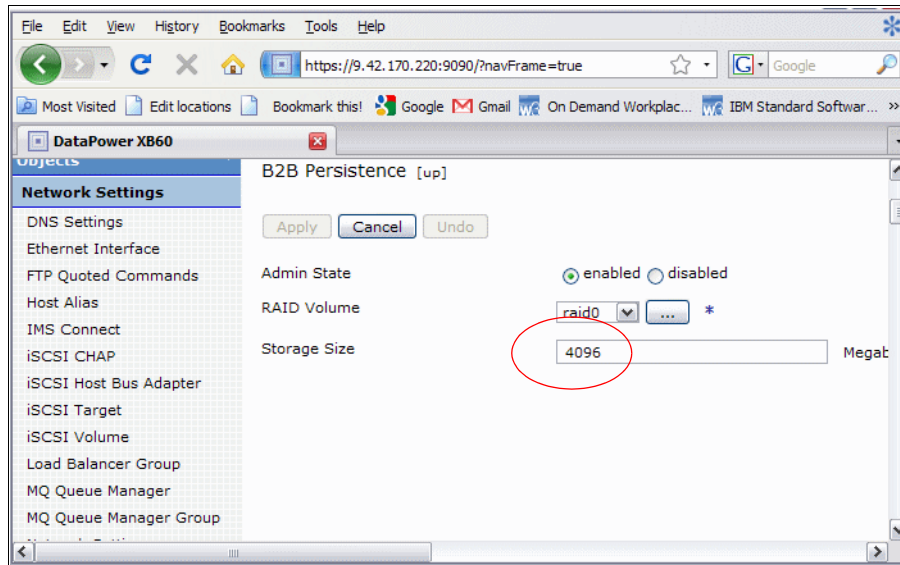


Figure 6-3 Checking B2B Persistence Storage Size

5. Click **Apply** to save the object to the running configuration.
6. Optionally, click **Save Config** to save the object to the startup configuration.

You can use the CLI to check the document store size by using the **show b2bp encrypted-space** command. You can also use the CLI to check the persistent store size by using the **service show b2bp size** command as shown in Example 6-5.

Example 6-5 The service show b2bp size command

```

xb60# service show b2bp encrypted-space
B2B encrypted space: 28037 MB free of 28166 MB total
xb60# service show b2bp size
B2B persistence storage size used: 4032 KB

```

Important: When you change the B2B Persistence Store value, you are allocating the maximum size that the embedded database can grow on the hard drive. After you change the value of the B2B Persistence Store to a higher number, it cannot be changed to a lower number.

Managing encryption settings

During the installation of a B2B XB60 appliance, the document storage partition on the hard disk drive is encrypted with the internal system key. After installation, an administrator in the default domain can change the encryption settings either to be encrypted with a passphrase or to be unencrypted. Refer to the “Managing Storage” chapter of the *B2B Developers Guide* for instructions how to change encryption settings for the document storage partition. The *B2BDevelopersGuide.pdf* is provided on the CD shipped with the device or it can be downloaded from the IBM WebSphere DataPower Product Documentation Portal Web page:

<http://www-01.ibm.com/support/docview.wss?rs=2362&uid=swg21377654>

6.3 Domains, groups, and users

Next, we explain domains, groups, and users.

6.3.1 Domains

A number of appliance-wide resources and settings can be defined only in the default domain, such as network interfaces, users, access controls, and application domains. After any user enters an application domain, either through logging in or switching domains, that user can no longer access appliance-wide resources. When viewed from the main navigation area, these resources are disabled. The default domain cannot be deleted.

Users can be assigned to specific application domains to allow for greater administrative control. Users, who are restricted to specific application domains, can perform activities in only those application domains (provided that the user has the appropriate access controls). Services defined in one application domain cannot be shared with another application domain.

Application domains can be restarted independently without affecting any other domain and without requiring a restart of the entire appliance. When a domain is restarted, the persisted configuration file for that domain is used, which might change the running configuration of the domain.

To create an application domain, use the following procedure:

1. Select **Administration** → **Configuration** → **Application Domain** to display the Application Domain catalog.
2. Click **Add** to display the Configure Application Domain window.
3. Specify the name in the Name field.

4. Click **Apply**.

Refer to the *Administrators Guide* for details about the creation and management of domains, groups, and users. The AdministratorGuide-v1.pdf is provided on the CD shipped with the device or it can be downloaded from the IBM WebSphere DataPower Product Documentation Portal Web page:

<http://www-01.ibm.com/support/docview.wss?rs=2362&uid=swg21377654>

6.3.2 Specifying access control

User accounts identify local users. Each local user account is defined by a user name and password. These credentials are used to log in to the DataPower appliance and to apply the appropriate access profile to the user account.

Creating user groups

A *user group* represents a collection of users who perform similar tasks and require the same level of access to the appliance. User groups are assigned privileges to DataPower resources. Each privilege is known individually as an access policy. A collection of policies is known as an *access profile* in DataPower terminology.

Related trading partner user accounts can be combined into one user group, which shares the same access profile. The B2B Transaction Viewer is a useful utility for checking message flow and troubleshooting when there are problems. The Transaction Viewer access can be defined at an extremely detailed level for the specific needs of external trading partners. Refer to 7.2, “Transaction Viewer examples using RBM” on page 111 for details explaining how to configure permissions to limit access to the B2B Transaction Viewer with group-defined access policies.

Creating user accounts

Only the *admin* user, when in the *default* domain, or a member of the sysadmin group with the correct access policy can manage user accounts. The best practice is to create new local users with the New User Account utility, because this utility defines a user who is a member of a group.

To create a user account, select **Administration** → **Access** → **New User Account**. The wizard prompts for the following information:

1. Restrict this user to a domain (Yes).
2. If Yes, select the domain to which to restrict this user. (If the domain does not exist, you can create it from the restrict user view.)
3. Domain Account Type. Enter the name of the group or create a new group.

4. Name of user account.
5. Summary describing the user account (optional).
6. Password and confirmed password for the user account.
7. Click **Commit**.
8. Optionally, click **Save Config** to save the object to the startup configuration.

6.4 Backing up the appliance

Now that the device is initialized and the base configuration is specified, it is a good idea to create a backup of the entire appliance. The backup and export utility copies specified configuration data from the appliance to a file in the export: directory. You can optionally download the file to your workstation.

Note: The following objects are never exported:

- ▶ User account objects
- ▶ Certificate objects
- ▶ Key objects

Certificate Objects and key objects that are DataPower-generated keys and certificates can be exported at time of creation only and then never again.

The following files are never exported:

- ▶ Log files
- ▶ Firmware files

To ensure that all other objects and files are exported, use the admin account. For any other users, only objects and files that are accessible to that user are included in the export package.

To start a backup of the entire appliance:

1. Select **Administration** → **Configuration** → **Export Configuration** to display the Initial Export Configuration window.
2. Select **Create a backup of the entire system** and click **Next** to display the File Name window:
 - a. Specify a descriptive object-specific summary in the Comment field.
 - b. The Export File Name defaults to export (.zip). If a file of this name exists in the export: directory, it is overwritten.
 - c. Click **Next**. The configuration of the entire appliance is backed up.

When the backup completes, the file is in the export: directory. You can optionally download the export file to your workstation.

Note: The Import Configuration utility requires that the export file resides on your workstation.

3. To download the export file to your workstation, click **Download**.
4. Click **Done** to close this window and return to the Control Panel.

The export file can be accessed from the export: directory. If downloaded, the export file is on your workstation.



B2B configuration options

This chapter discusses the configuration options that are specific to the business-to-business (B2B) functionality of DataPower B2B Appliance. There are many other configuration options in the XB60; however, they are core functions that are not unique to B2B and will not be discussed here.

Topics include:

- ▶ Setting the transaction store size
- ▶ Setting up the Web B2B Viewer Management Service
- ▶ Description of the B2B Gateway Service
- ▶ Description of B2B Partner Profiles
- ▶ Description of the B2B Transaction Viewer
- ▶ Description of B2B data persistence options

7.1 XB60 B2B services

Each of the DataPower services available in the XB60 device has built-in features and functionality to handle many types of transactions and protocols that can be used to route data. Because the XB60 was built on top of our core Application Integration appliance, the XB60 has many services that are application integration services in nature. In this section, we only discuss the B2B-specific services and objects. For more information about the other B2B services that are available, refer to the user documentation.

From the Control Panel under B2B, there are three shortcuts (Figure 7-1) specifically for B2B: B2B Partner Profiles, B2B Gateway Service, and B2B Transaction Viewer. This section will briefly describe each function.

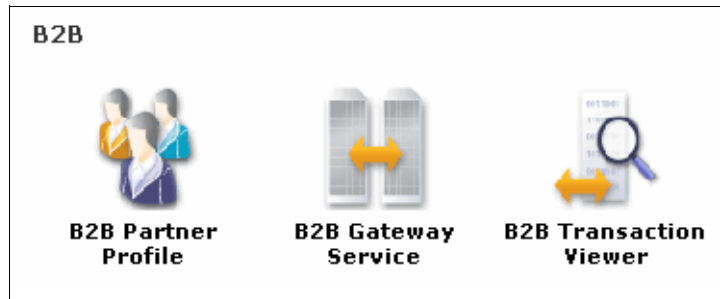


Figure 7-1 Control Panel view of B2B objects

7.1.1 B2B Partner Profiles

The B2B Partner Profile is the configuration object where the trading partner information is defined. This information includes the profile name, profile type, business IDs, AS security, destinations for document routing, and contact information.

A trading partner is either an internal or external trading partner based on the understanding that an internal trading partner exists within the corporate enterprise and an external trading partner exists outside of the enterprise. Trading partners have unique business IDs. However, if a profile is defined as internal, that trading partner might also have the same business ID defined in their external definition, because an internal trading partner and an external trading partner are different objects.

Whether a trading partner is internal or external also affects the options under the AS Security tab. Internal profiles will only use private security credentials, and external profiles will only use public security credentials. When you trade documents between two hubs, the internal profile is the private side of the profile and contains private keys, and the external profile is the public side of the profile and contains only public certificates. Figure 7-2 depicts how the hub owner's private side of the profile is stored on the owner's B2B hub and communicates with the public side of the owner's profile, which is stored on the partner's B2B hub. The partner's B2B Gateways will also work in the same manner.

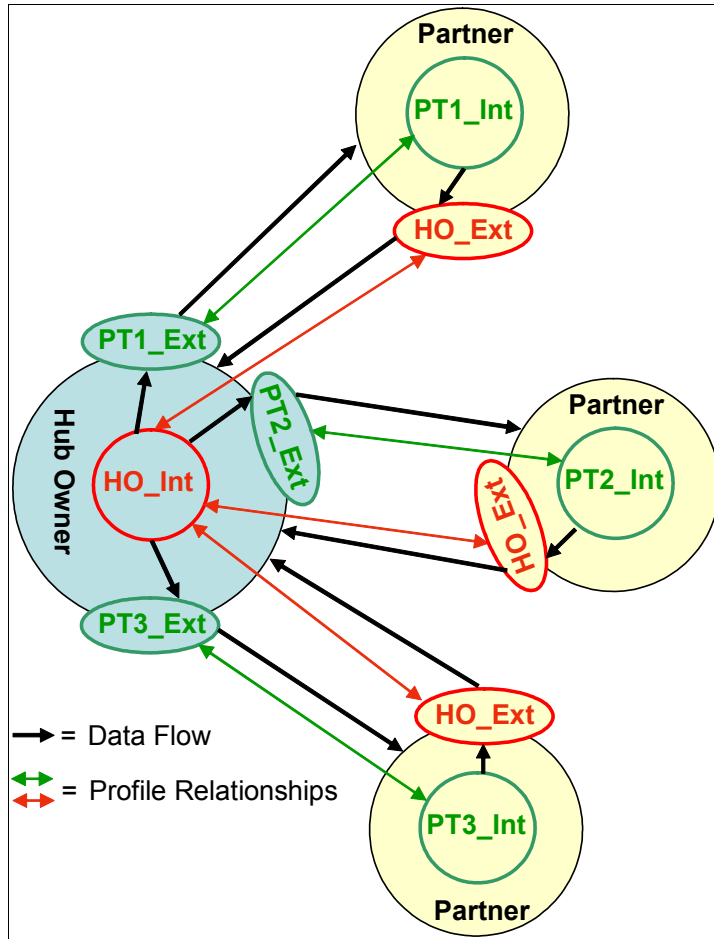


Figure 7-2 Internal and external profile linkage

The B2B Partner Profile contains four tabs that are used for service configuration: Main, AS Security, Destinations, and Contacts tabs. This section discusses the properties of each tab in detail.

The Main tab

The main tab identifies the name of the profile, the Admin State (enabled or disabled) that allows the profile to be used in a B2B Gateway object, the Profile Type (internal or external), and the Partner Business IDs associated with the profile. Refer to Figure 7-3 on page 95.

The Business IDs must be equivalent to identifiers that are expected in the transactions, such as:

- ▶ A value for an AS header: AS2-To or AS2-From for AS2 transactions and AS3-To or AS3-From for AS3 transactions
- ▶ A value that is extracted from the ISA or UNB headers of the EDIX12 or EDIFACT documents

Important: EDIX12 documents use a qualifier to identify the type of ID that is to follow in the ISA header. The XB60 will automatically concatenate the two character qualifier to the identifier, for example, *ZZpartner1*, where *ZZ* is the qualifier and *partner1* is the identifier.

- ▶ A value that is extracted from a configured XPATH entry (configured in the B2B Gateway) of an XML document

Configure B2B Partner Profile

Main AS Security Destinations Contacts

B2B Partner Profile

Apply Cancel

Name MyHub_Ext *

Admin State enabled disabled

Comments

Profile Type External Internal

Partner Business IDs

| | |
|----------------------|------------------------------------|
| myhub | ↑ ↓ ✕ |
| zzmyhub | ↑ ↓ ✕ |
| <input type="text"/> | <input type="button" value="Add"/> |

*

Figure 7-3 Main tab: Internal partner profile

The AS Security tab

Internal profiles:

- ▶ For internal profiles, AS security is optional, but if it is used, the AS Security tab contains a place for credentials to be identified for decrypting inbound AS messages. The AS Security tab also contains a place for credentials to be used when signing outbound AS messages and AS Message Disposition Notifications (MDNs). Internal profiles use private keys.
- ▶ For inbound security (refer to Figure 7-4 on page 96), optionally select **Require Signature** to indicate whether inbound messages must be signed. You can also (optionally) specify whether inbound messages must be encrypted by selecting **Require Encryption** and selecting the identification credentials necessary to decrypt the message from the **Inbound Decryption Identification Credentials** list. Credentials can be selected from the drop-down list or created by clicking + (the plus sign). Existing credentials can be modified by clicking ... (the ellipsis).

- ▶ For outbound security, outbound messages can (optionally) be signed. To sign messages, select **Sign Outbound Messages** and select the signing identification credentials to be used to sign outbound messages. Credentials can be created by clicking + (the plus sign) or existing credentials can be modified by clicking ... (the ellipsis). Also, select the hash algorithm from the **Signing Digest Algorithm** list.

Note: Although this setting controls whether to sign outbound messages, the **Send Messages Unsigned** property in the Partner Profile Destinations tab can override this setting. You override this setting if you have selected partners who do not require signatures.

Figure 7-4 Internal profile AS Security tab

External profiles

For external profiles, AS security is optional, but if used, the AS Security tab contains a place for credentials to be used to verify signatures and validate signature certificates from the partner. External profiles use public certificates:

- ▶ For inbound security (refer to Figure 7-5), optionally set the **Inbound Signature Validation Credential**. Credentials can be selected from the drop-down list box or created by clicking + (the plus sign). Existing credentials can be modified by clicking ... (the ellipsis). Additionally, if you expect to receive MDNs over the Secure Sockets Layer (SSL), you must configure the MDN SSL Proxy Profile field by selecting an existing profile from the drop-down list box or by creating a profile by clicking + (the plus sign). Existing proxy profiles can be edited by clicking ... (the ellipsis).

Figure 7-5 External profile AS Security tab

The Destinations tab

Message destinations define the routing information for the partner. The first destination is the default destination. The gateway uses the default destination when no specific destination is selected from within the B2B Gateway. If the destination protocol is AS, AS attributes can be configured to support security, MDNs, transaction time to live, and resend logic. B2B partner profiles can have multiple destinations. For those individuals familiar with DataPower terminology, it might be helpful to think of internal partner destinations as a “Backend URL.”

Tip: An internal partner profile can have multiple destinations defined, but the first entry in the list will be the default destination. To change the default destination, move the desired destination up to the first position.

When you select the Destinations tab, you will be presented with the Destinations list view as seen in Figure 7-6 on page 98. From this view, you can

create a new destination by clicking **Add** or by editing an existing destination by clicking the pencil to the right of the destination.

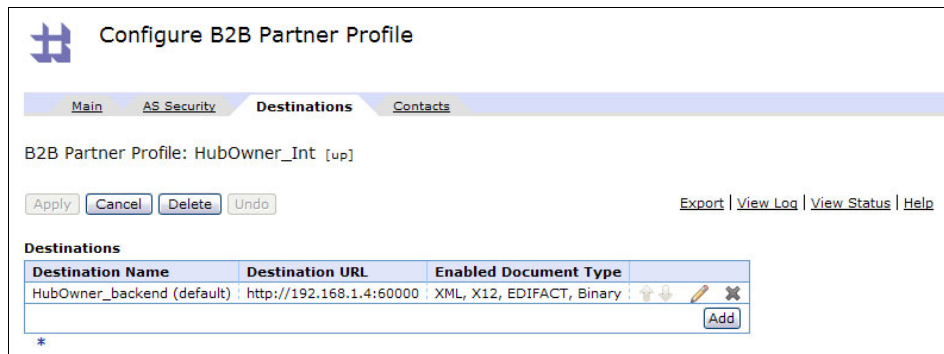


Figure 7-6 Destinations list view

When you click **Add**, you are presented with a Destinations detailed view where you can configure attributes related to your destination of choice (refer to Figure 7-7 on page 99). The XB60 provides a wide range of destinations from which to select. Supported destinations are:

- ▶ as2://
- ▶ as2s:// (as2 over SSL)
- ▶ as3://
- ▶ dpims://
- ▶ dpimsssl://
- ▶ dpmq://
- ▶ dpnfs://
- ▶ dpwasjms://
- ▶ ftp://
- ▶ http://
- ▶ https://
- ▶ mq://

In addition, you must provide:

- ▶ **Destination Name:** This field is a descriptive name given to the destination; destinations must have unique names within a profile.
- ▶ **Enabled Document Type:** In this section, you can enable the types of documents that this destination will support. The choices are XML, X12, EDIFACT, and binary. The default behavior is to enable all document types.
- ▶ **Attributes:** Each of the previous destinations has different connection and configuration attributes that can be set to determine how to route data and whether a B2B messaging envelope is applied to the file. For the purpose of

this chapter, we do not show information for specific destination options. For details about each destination type, refer to the user documentation.

Configure B2B Partner Profile

Main AS Security **Destinations** Contacts

B2B Partner Profile

Apply Cancel

Name *

| Destination Name | Destination URL | Enabled Document Type |
|------------------|-----------------|-----------------------|
| (empty) | | |

Destinations

Destination Name *

Enabled Document Type

- XML
- X12
- EDIFACT
- Binary

Figure 7-7 Destinations detailed view

The Contacts tab

The contacts section allows contact information to be entered for the profile. To create one or more contact records, enter information into the the provided fields. The Contact tab is optional; it is not meant to be treated as a contact manager but rather to provide the users of the system with the ability to store contact information about key technical people who are responsible for data that is sent or received to that profile.

7.1.2 B2B Gateway Service

A B2B Gateway Service is an object that defines the characteristics of B2B transaction processing and the association of trading partners allowed to trade data with the B2B Gateway. The B2B Gateway Service includes handling AS2 and AS3 data flows as well as the generation and consumption of the MDNs that are associated with each transaction. If you click **B2B Gateway Service** → **Add**, a new B2B Gateway template is displayed. An example of the new service is depicted in Figure 7-8 on page 100.

The screenshot displays the configuration page for a B2B Gateway service. At the top, there are four tabs: Main, Archive, XML Formats, and Advanced. The 'Main' tab is selected. Below the tabs, the title 'B2B Gateway' is shown, followed by 'Apply' and 'Cancel' buttons. The 'Name' field contains 'NewB2BGateway'. Under the 'General Configuration' section, 'Admin State' is set to 'enabled' (radio button selected), 'Comments' is empty, 'Document Storage Location' is '(default)', and 'XML Manager' is 'default'. The 'Document Routing' section includes 'Front Side Protocol Handlers' with an empty list and an 'Add' button. The 'Attach Partner Profiles' section includes 'Active Partner Profiles' with a table header: 'B2B Partner Profile', 'Profile Enabled?', and 'Profile Destination'. The table is currently empty, and there is a dropdown menu with 'HubOwner_Ext' and an 'Add' button below it.

Figure 7-8 New B2B Gateway Service

The B2B Gateway Service contains four tabs that are used for service configuration: the Main, Archive, XML Format, and Advanced tabs. In this section, we discuss the properties of each tab in detail.

Main tab

The Main tab of the service contains the general configuration parameters, such as what protocol or protocols this service will accept and what partner or partners are allowed to access this service. The mandatory fields are:

- ▶ **Name:** The name of the service object; this name is a unique name within the application domain.
- ▶ **Admin State:** Enables or disables the object. If an object is in the disabled state, it will not execute.

- ▶ **Document Storage Location:** The URL where saved copies of inbound documents, outbound documents, and intermediate documents are stored. The location can be on the local encrypted area of the hard disk, an iSCSI server, or a Network File System (NFS) mount, which stores documents off of the appliance in an unencrypted storage location. Later in this chapter, 7.3, “B2B Data Persistence” on page 118 describes this process in detail.
- ▶ **XML Manager:** This object is responsible for management of the service. The “default” XML Manager is assigned to this service by default. There is typically one manager assigned to a service. The XML Manager controls XML document caching and XML parsing and contains the User Agent, which can control how the service connects to external services. If any customizing is needed, we recommend that you create a “new” XML Manager and not to make changes to the “default” XML Manager.
- ▶ **Front Side Protocol Handlers:** The Front Side Protocol Handlers are the entry points into the service. They can be listeners that wait for transactions to be sent into the gateway or pollers that periodically look for transactions from a specific location. They contain the IP address or Host Name (recommended) and the port on which the handler is listening or from which the handler is getting data. There are many Front Side Protocol Handlers in the B2B Gateway. The AS2 and AS3 handlers expect that messages sent into them are wrapped in an AS envelope and meet all of the security requirements and attributes supported by the AS2 and AS3 specifications. All remaining handlers expect that documents received into the handlers have no special formatting requirements and are of type EDIX12, EDIFACT, XML, or binary.

The supported Front Side Protocol Handlers for the B2B Gateway Service are:

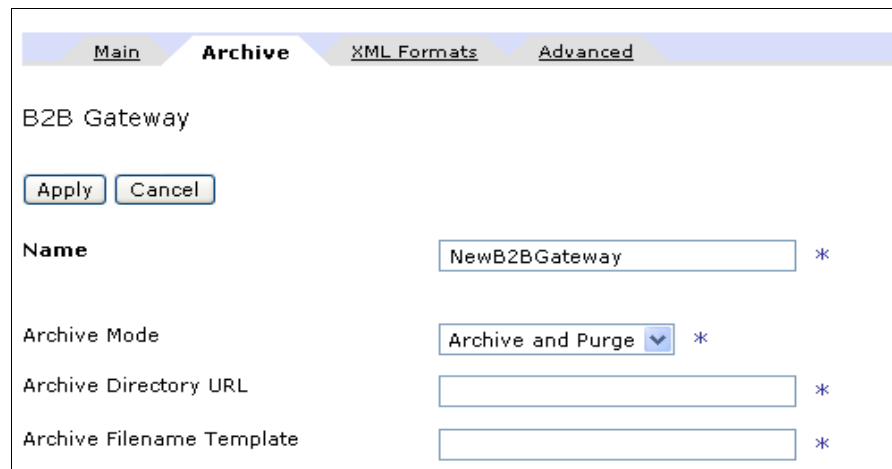
- AS2
 - AS3
 - FTP Poller
 - FTP Server
 - NFS Poller
 - HTTP
 - HTTPS
 - IMS
 - WebSphere Java Message Service (JMS)
 - MQ
 - Secure Shell File Transfer Protocol (SFTP) Server
- ▶ **Active Partner Profiles:** Partner definitions, such as AS security, business IDS, destinations, and contact information can be predefined in the Partner Profile section. When defining a B2B Gateway Service with message routing, you can simply select a preexisting internal or external trading partner relevant to the message flow or a new profile can be created within the B2B

Gateway. Refer to 7.1.1, “B2B Partner Profiles” on page 92 for a detailed description of the process of creating B2B profiles.

Archive tab

The archive mode is a required gateway configuration item. There are two modes of archiving: Archive and Purge or Purge only. As seen in Figure 7-9, the mandatory fields for the archive configuration when the archive mode is “Archive and Purge” are:

- ▶ **Archive Directory URL:** The directory to store documents. The documents can be stored locally or off the device.
- ▶ **Archive Filename Template:** This name is the unique base name of the archived file.



The screenshot shows the configuration page for a B2B Gateway. At the top, there are four tabs: "Main", "Archive" (which is selected), "XML Formats", and "Advanced". Below the tabs, the text "B2B Gateway" is displayed. There are two buttons: "Apply" and "Cancel". The configuration fields are as follows:

| | | |
|---------------------------|--|---|
| Name | <input type="text" value="NewB2BGateway"/> | * |
| Archive Mode | <input type="button" value="Archive and Purge"/> | * |
| Archive Directory URL | <input type="text"/> | * |
| Archive Filename Template | <input type="text"/> | * |

Figure 7-9 Archive and Purge mode

As seen in Figure 7-10 on page 103, there are no mandatory fields for the archive configuration when the archive mode is “Purge.” The three basic configuration parameters are:

- ▶ **Archive Document Age:** This number is the maximum number of days that the documents will be retained on the device before they are purged.
- ▶ **Disk Use Check Interval:** This number is the interval in minutes to check for documents that exceed that archive document age.
- ▶ **Maximum Disk Usage for Documents:** This number of Kilobytes is the maximum storage allocated for this service.

The screenshot shows a configuration window for a B2B Gateway. At the top, there are four tabs: 'Main', 'Archive', 'XML Formats', and 'Advanced'. The 'Archive' tab is selected. Below the tabs, the text 'B2B Gateway' is displayed. There are two buttons: 'Apply' and 'Cancel'. The configuration fields are as follows:

| | | |
|----------------------------------|--|-----------|
| Name | <input type="text" value="NewB2BGateway"/> | * |
| Archive Mode | <input type="text" value="Purge Only"/> | * |
| Archive Document Age | <input type="text" value="90"/> | Days |
| Disk Use Check Interval | <input type="text" value="60"/> | Minutes |
| Maximum Disk Usage for Documents | <input type="text" value="25165824"/> | Kilobytes |

Figure 7-10 Purge Only mode

Using Archive

Each B2B Gateway object has an Archive tab and can be independently configured to archive documents and metadata before purging or to purge the documents without archiving. As a best practice, we recommend that a remote file server (transferred through HTTPS or FTP) or a mounted iSCSI or NFS be used to store archived data. The relevant settings are (and can be seen in the example in Figure 7-11 on page 104):

- ▶ **Archive Mode:** Archive and Purge.
- ▶ **Archive Directory URL:** `local:///foo`, which in this example is a mounted external file system using the iSCSI protocol. Optionally, you can set this value to a local drive location or an NFS mount point.
- ▶ **Archive Filename Template:** `hubowner`, which is the base filename for your archived files. The files will be appended with the current date time stamp (`hubowner20090319193930.gz`)
- ▶ **Archive Minimum Size:** 1024 Kilobytes. Archiving will not occur unless the amount of data stored is in excess of this number.
- ▶ **Archive Document Age:** 10 days. Documents that are older than the specified number of days will be archived.
- ▶ **Archive Minimum Documents:** 100. This value is the minimum number of documents allowed on the system before archive will be triggered for this service.

- ▶ **Disk Use Check Interval:** 60 hours. Specifies how often to check for documents to archive. This field defaults to hourly.
- ▶ **Maximum Disk Usage for Documents:** 25165824 Kilobytes. This value is the maximum cumulative size of all documents used by this gateway before the archive process is triggered. The value needs to be based on the number of gateways in use, your transaction load, and how often you want archives to happen.

B2B Gateway: HubOwner [up]

Apply Cancel Delete Undo [Export](#) | [View Log](#) | [View Status](#) | [Archive/purge transactions](#) | [Help](#)

| | |
|----------------------------------|--|
| Archive Mode | Archive and Purge ▼ * |
| <u>Archive Directory URL</u> | local:///foo * |
| Archive Filename Template | hubowner.xml * |
| Archive Minimum Size | 1024 Kilobytes |
| Archive Document Age | 1 Days |
| Archive Minimum Documents | 100 Documents |
| Disk Use Check Interval | 60 Minutes |
| Maximum Disk Usage for Documents | 25165824 Kilobytes |
| AS Documents to Back Up | <input checked="" type="checkbox"/> Inbound Message <input checked="" type="checkbox"/> Inbound MDN <input checked="" type="checkbox"/> Outbound Message <input checked="" type="checkbox"/> Outbound MDN |

Figure 7-11 B2B Archive and Purge example

XML Formats tab

The XML Formats tab is used to allow the user to configure the XPATH statements that are needed to find the Sender and Receiver information from incoming XML documents. Sender and Receiver IDs are needed to properly route XML documents through the B2B Gateway Service. The XML Formats tab has one parameter called XPath Routing Policies. The XPath Routing Policies contain the xpath statements used to extract the Sender ID and Receiver ID. The appliance has a built-in XPATH Tool that can easily build the statement used in

this policy. These xpath routing policies can be shared by other B2B Gateway Services in the application domain.

Advanced tab

The Advanced tab on the B2B Gateway contains global settings for the B2B Gateway:

- ▶ **Service Priority:** Sets the priority level for the service in this application domain. This setting will allow you to give a higher quality of service for a specific service. For instance, if you receive transactions from 200 trading partners and two of those partners are responsible for about 90% of your transactions, you can configure two B2B Gateways: one for high priority trading partners and one for normal priority trading partners. The high priority B2B Gateway will process data faster than the lower priority gateway, giving your high priority trading partners a higher quality of service than the remaining trading partners.
- ▶ **Default AS2 MDN Return Path:** Set this path to the default location where you want to receive asynchronous MDNs for AS2 transactions for the entire service. This property can be overridden by the destination for the external partners; refer to Figure 7-12 for the external partner profile.

The screenshot shows the 'Advanced AS Behavior' configuration page. It contains several settings:

- Compress Messages:
- Request MDN:
- Time to Acknowledge: 1800 Seconds
- Request Asynchronous MDN:
- AS2 MDN Redirection URL: http:// (highlighted with a red box)
- Request Signed MDN:
- Attempt Message Retransmission:
- Maximum Retransmissions: 3

Figure 7-12 External partner AS2 asynchronous MDN destination

- ▶ **Default AS3 MDN Return Path:** Set this path to the location where you want to receive asynchronous MDNs for AS3 for the entire service. This property can be overridden by the destination for the external partners:

- ▶ **Document Routing Preprocessor:** This field is used to select the stylesheet that is used to process binary transactions, transactions that are not X12, EDIFACT, or XML. The B2B Gateway Service requires Sender and Receiver IDs in order to use profile management to route B2B data. Typically, binary data is not parsable for IDs. Therefore, you need to set business IDs to route binary data through the B2B Gateway, which is what the binary routing files were designed for. If you do not need profile management for your binary data, we advise that you route binary data through a Multi-Protocol Gateway. However, if you want to use it in conjunction with Profile management, the Multi-Protocol Gateway coupled with a B2B Gateway will provide you with an extremely flexible way of routing binary data through the XB60. For more information about routing binary data directly through a B2B Gateway, log on to the WebSphere DataPower SOA Appliances Support site at:

<http://www-01.ibm.com/software/integration/datapower/support/>

Search for 1330240 and 1370503 in the Search Support field.

For examples of how to trade binary data either over a Multi-Protocol Gateway or over a B2B Gateway, refer to Chapter 12, “Trading outbound binary documents using the B2B Gateway Service” on page 273.

7.1.3 B2B Transaction Viewer

The XB60 brings a new generation of transaction viewing capabilities to the DataPower appliance concept. In the XB60, all data that flows through a B2B Gateway Service is displayed in an easy to read, at a glance viewer where users can see the status of their B2B transactions. Because we have to monitor the state in the B2B Appliance to support industry standard B2B messaging protocols, we needed the capability to easily monitor that state without having to navigate large log files. In the addition to being able to monitor B2B transactions, the B2B Transaction Viewer gives the user the ability to manually resend transactions and view “off-the-wire” files as well as viewing the decrypted payload and the MDN. Because the payloads by default are stored in the encrypted portion of the RAID volume, the only way to see them in the clear is with the appropriate permissions in the B2B Transaction Viewer or after they are archived off of the device.

The B2B Transaction Viewer can be configured to allow an external client access to only transaction (row) specific data, for example, to only transactions related to a particular partner ID. The transaction viewer can also limit the following: view access to the columnar metadata, view access to transaction message documents, and the ability to resend transactions. These options will be outlined in “Manage transaction viewing with RBM” on page 107.

Enabling transaction viewing for external partners

The Web B2B Viewer Management Service can be set up and enabled to allow external partners access to view transactions. You will need to log in as admin in the default domain to configure this service.

To configure browser access to the viewer, use the following procedure:

1. Select **Objects** → **B2B Configuration** → **Web B2B Viewer Management Service**.
2. Retain the default setting for the Admin State toggle. To place the object in an inactive administrative state, click disabled.
3. Define the connection from the browser to the appliance:
 - a. Specify the IP address to accept requests in the Local IP Address field. The default is 0.0.0.0, which indicates that the service is active on all IP addresses.
 - b. Specify the listening port in the Port Number field. The default is 9091.
4. Select the instance of the Access Control List object to associate from the Access Control List list. The default is web-b2b-viewer.
5. Specify a descriptive object-specific summary in the Comment field.
6. Specify the timeout for the connection in seconds in the Idle Timeout field. The default is 600.
7. Click **Apply** to save the object to the running configuration.
8. Optionally, click **Save Config** to save the object to the startup configuration.

Manage transaction viewing with RBM

The XB60 Transaction Viewer allows controlled access to transaction data by authorized user accounts. This access is controlled by the XB60 administrator using the standard DataPower Role Based Management (RBM) functionality. Refer to 8.3, “Role Based Management (RBM)” on page 146.

This section will outline RBM techniques for limiting authorized user account access to data through the implementation of user groups. These user groups define the rights that the user has to the DataPower B2B Viewer resource with respect to column visibility, partner visibility, and send/view actions.

User accounts

Viewing transactions starts with creating user accounts for the B2B Transaction Viewer (refer to Figure 7-13 on page 108). The user account will allow the user to log in to the B2B Transaction Viewer browser. User accounts can be assigned to a predefined group upon creation.







| Name | Status | Op-State | Logs | Access Level | User Group | Comments |
|--------------|--------|----------|---|---------------|------------------------|----------|
| hubowner | saved | up |  | group-defined | partner_group | |
| limited_view | saved | up |  | group-defined | limited_columns_group | |
| no_doc | saved | up |  | group-defined | no_retrieve_docs_group | |
| no_resend | saved | up |  | group-defined | no_resend_group | |
| partner | saved | up |  | group-defined | partner_group | |
| partner2 | saved | up |  | group-defined | partner2_group | |

Figure 7-13 User account list

Managing user group accounts

A *user group* represents a collection of users who perform similar duties and require the same level of access to the DataPower appliance. User groups are assigned rights to one or more DataPower resources. When adding these rights to the access profile of the specific user group, each right is known individually as an *access policy*. A collection of access policies is known as an *access profile*.

User group account

Related trading partner user accounts can be combined into one user group. These individual user accounts are limited to the access profile of the user group account to which they are assigned. For instance, multiple partner IDs from one company can be combined in one group account for access to related transactions.

Limiting column visibility

Filtering of column data restricts the resultant dataset to specific metadata associated with each transaction. This type of filtering uses the *b2b/column-visibility* resource of the RBM policy. When defined, the user can view transactional data for the explicitly defined columns only. The policy can contain one or many columns that can be exposed for viewing. This granularity can be further refined by combining the partner visibility access policy. Refer to “Limiting partner visibility” on page 110.

The policy string has the format that is shown in Example 7-1, where each column added to the policy string will be viewable in the Transaction Viewer.

Example 7-1 B2B Transaction Viewer policy string that defines column visibility resource

```
<ip>/<domain>/b2b/column-visibility?Access=r&Columns=<column1>+<column2>+...+<columnN>
```

Tip: To allow all columns, set the access policy to:

```
<ip>/<domain>/b2b/column-visibility?Access=r
```

Table 7-1 shows the mapping of B2B Transaction Viewer column labels to dataset fields.

Table 7-1 Mapping of labels in B2B Transaction Viewer to dataset fields

| Label | Dataset field | Comments |
|---|---|---|
| Transaction Set ID | TransactionSetID InputDoc OutputDoc ContentDoc MDNBodyDoc | If you click the identifier link for the transaction set, it lists which documents can be displayed. The list contains links for the following documents: - Input - Output - Content - MDN Body |
| Transaction ID | TransactionID | |
| Gateway Name | GatewayName | |
| Sender Name (ID)/ Receiver Name (ID) | SenderName SenderID ReceiverName ReceiverID | |
| Inbound URL/ Outbound URL | InboundURL OutboundURL | |
| Input Time/Output Time | InputTime OutputTime | |
| Result Code | ResultCode | |
| MDN Status | MDNStatus | |
| MDN Time | MDNTime | |
| MDN Received | MDNReceived | |

| Label | Dataset field | Comments |
|----------------------|---|---|
| Headers | MessageIDHeader ContentTypeHeader ASFromHeader ASToHeader DateHeader DispositionHeader DispositionOptionsHeader ContentLengthHeader ContentDispositionHeader OriginalMessageIDHeader | The cell contains the string (Show Headers). If you hover over the string, the viewer displays the appropriate headers with their defined values. |
| Document ID | DocumentID | |
| Document Type | DocumentType | |
| MDN Type | MDNType | |

Limiting partner visibility

Filtering of partner-sensitive data (rows) restricts the result dataset to specific transactions associated with previously configured user accounts. This type of filtering uses the *b2b/partner-visibility* resource of the RBM policy. When a partner-visibility policy string is defined for a user account, the user can view transactional data for the explicitly defined users only. Each policy string can contain only one user account entry. Therefore, to explicitly allow a user to view data for specific users, add a policy string for each account.

The policy string has the format that is shown in Example 7-2.

Example 7-2 B2B Transaction Viewer policy string that defines partner visibility resource

```
<ip>/<domain>/b2b/partner-visibility?Access=r&Partner=<partner-name>
```

The administrator can add one or more partner visibility access policies to a user group. The RBM B2B logic will look for these policy strings and filter the resultant data sets returned to users.

Limiting B2B access control to actions

There is a requirement to allow the B2B administrator the ability to control access to certain B2B Viewer operations. These operations provide the ability to resend (retransmit) B2B transactions and to retrieve B2B documents associated with transactions. When the resend transaction access policy is enabled in a user group, it only applies to transactions for the trading partners that are listed in the B2B Partners Visibility RBM string.

The policy string to limit the ability to resend transactions has the format that is shown in Example 7-3, where access to the function is unavailable unless explicitly listed in an access policy.

Example 7-3 B2B Transaction Viewer policy string allowing resend transaction access

*<ip>/<domain>/b2b/resend-transaction?Access=x
(affects b2b-resend-all and b2b-resend)*

7.2 Transaction Viewer examples using RBM

Here, we show several examples of the RBM policies and their effect on the B2B Transaction Viewer.

7.2.1 XML Management Interface

Currently, there is no support for XML Management Interface support for the B2B Transaction Viewer.

7.2.2 Command line interface (CLI)

Currently, there are no active CLI commands for the B2B Transaction Viewer.

7.2.3 WebGUI interface

The following examples illustrate the functionality of the RBM policies and their effect on the B2B Transaction Viewer. The user will be able to view specific data for explicitly defined columns, partners, and actions.

Column visibility examples

Example 7-4 shows the policy string format for column visibility.

Example 7-4 Column-visibility RBM access policy defines access to specific columns

**/student07/b2b/column-visibility?Access=r&Columns=TransactionSetID+TransactionID+GatewayName+SenderID+ReceiverID*


Figure 7-14 shows the access policies for a user group that defines the columns available to partner2.

enabled disabled

| | |
|---|---|
| */student07/*?Access=r | ✕ |
| */student07/b2b/column-visibility?Access=r&Columns=TransactionSetID+TransactionID+GatewayName+SenderId+ReceiverID | ✕ |
| */student07/b2b/partner-visibility?Access=r&Partner=partner2 | ✕ |

Figure 7-14 User group access policy that explicitly allows column visibility to specific column data

Columns can be explicitly set to be visible in the access policy. Columns not explicitly defined will not be visible. In Figure 7-15, the B2B Transaction Viewer displays the columns for TransactionID, TransactionSetID, GatewayName, SenderID, and ReceiverID for Partner2 as defined in Figure 7-14.

 **B2B Viewer** [Help](#)

[Modify Query](#) | [Refresh](#) |

| | Transaction Set ID | Transaction ID | Gateway Name | SenderName (ID) / Receiver (ID) | Action |
|-------------------------------------|---------------------|----------------|--------------|---------------------------------|------------------------|
| <input checked="" type="checkbox"/> | 105 | 81942 | HubOwner | partner2 | Resend |
| <input type="checkbox"/> | 104 | 171592 | Partner2 | partner2 | Resend |

Figure 7-15 B2B Viewer showing explicitly defined columns

Example 7-5 on page 112 shows the format for the B2B policy string that restricts column visibility for specific columns.

Example 7-5 B2B policy string that restricts column visibility for specific columns

```
/*/*/b2b/column-visibility?Access=NONE&Columns=TransactionID+TransactionSetID+GatewayName+InboundURL+OutboundURL+MDN
```


Figure 7-16 displays the user group with the access policy string in Example 7-5 that explicitly denies visibility to specific columns.

Note: All columns are given initial visibility as noted by the access policy, and then, the columns to be denied visibility are added as a separate partner-visibility string.

enabled disabled

Figure 7-16 User group access policy that explicitly restricts access to specific column data

Figure 7-17 shows the B2B Viewer result set that displays explicitly restricted column data for a user group.

 **B2B Viewer** [Help](#)

[Modify Query](#) | [Refresh](#) |

| | SenderName (ID) / Receiver (ID) | Input Time / Output Time | Result Code | MDN Status | MDN Time | MDN Received | Headers | Document ID | Document Type | Action |
|--------------------------|-----------------------------------|--------------------------|-------------|---------------------|----------|-----------------------|----------------|-------------|---------------|------------------------|
| <input type="checkbox"/> | Sender: zzpartner2 (zzpartner2) | 2009-03-11 15:53:49.0 | Success | Received (Positive) | | 2009-03-11 15:53:50.0 | (Show Headers) | 000000002 | 850 | Resend |
| | Receiver: zzhubowner (zzhubowner) | 2009-03-11 15:53:50.0 | | | | | | | | |

Figure 7-17 B2B Viewer showing explicitly restricted columns

Partner visibility examples

This section shows examples that use the b2b/partner-visibility access policy.

Example 7-6 shows the structure of the access policy string for the partner-visibility resource that gives access to users xxxxxx and yyyyyy.

Example 7-6 B2B policy strings that grant specific partner visibility

```

/**/b2b/partner-visibility?Access=r&Partner=xxxxxx
/**/b2b/partner-visibility?Access=r&Partner=yyyyyy
    
```

Figure 7-18 shows the access policies for the user group that define the data set specific to partner2 and zzpartner2.

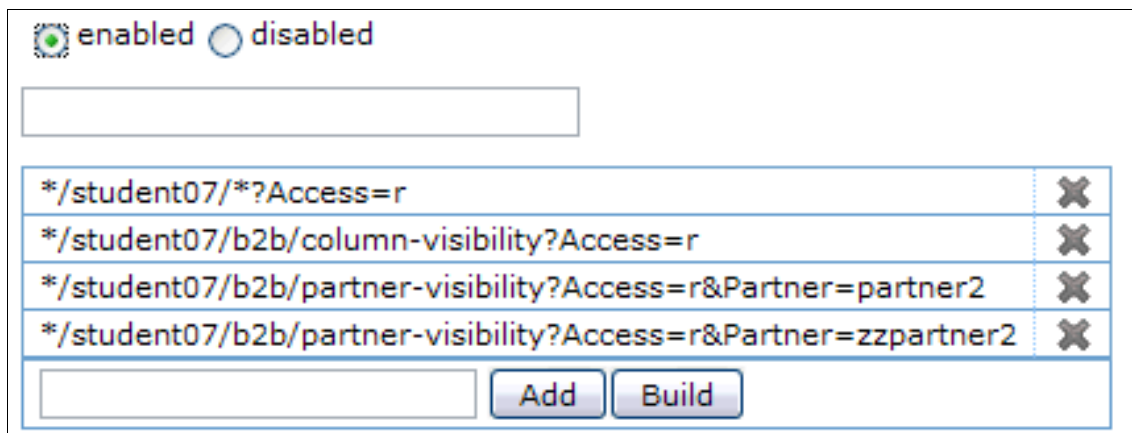


Figure 7-18 User group access policy explicitly allowing access to all columns for partner2 and zzpartner2 data

Figure 7-19 on page 114 shows the B2B Viewer result set for data specific to partner2 and zzpartner2 only.

| | Transaction Set ID | Transaction ID | Gateway Name | SenderName (ID) / Receiver (ID) | Inbound URL / Outbound URL | Input Time / Output Time | Result Code | MDN Status | MDN Time | MDN Received | Headers | Document ID | Document Type | MDN | Action |
|--------------------------|---------------------|----------------|--------------|---------------------------------|----------------------------|--------------------------|-------------|---------------------|-----------------------|-----------------------|----------------|-------------|---------------|-----|------------------------|
| <input type="checkbox"/> | 155 | 15543 | HubOwner | Sender: partner2 (partner2) | as2://9.42.170.220:5103/ | 2009-03-17 08:34:20.0 | Success | Sent (Positive) | 2009-03-17 08:34:20.0 | | (Show Headers) | | CustomXML | 0 | Resend |
| <input type="checkbox"/> | 154 | 20920 | Partner2 | Sender: partner2 (partner2) | http://9.42.170.220:5106/ | 2009-03-17 08:34:20.0 | Success | Received (Positive) | | 2009-03-17 08:34:20.0 | (Show Headers) | | CustomXML | 0 | Resend |
| <input type="checkbox"/> | 153 | 20904 | HubOwner | Sender: zzpartner2 (zzpartner2) | as2://9.42.170.220:5103/ | 2009-03-17 08:34:02.0 | Success | Sent (Positive) | 2009-03-17 08:34:02.0 | | (Show Headers) | 000000002 | 850 | 0 | Resend |
| <input type="checkbox"/> | 152 | 10246 | Partner2 | Sender: zzpartner2 (zzpartner2) | http://9.42.170.220:5106/ | 2009-03-17 08:34:02.0 | Success | Received (Positive) | | 2009-03-17 08:34:02.0 | (Show Headers) | 000000002 | 850 | 0 | Resend |

Figure 7-19 B2B Viewer showing result set for partner2 and zzpartner2

Controlling access to operations examples

This section shows examples that use the b2b/resend-transaction and b2b/get-document access policies.

Example 7-7 shows the access policy structure for defining the b2b/resend-transaction and b2b/get-document access policies.

Example 7-7 B2B policy string that controls B2B operation

```
/*/*/b2b/resend-transaction?Access=x
```

Limiting the access to various operations controls whether the user can perform specific B2B Viewer actions. Unless explicitly defined, the user cannot perform the action. Figure 7-20 on page 115 shows resend transaction access and get document access granted for the partners partner2 and zzpartner2. Figure 7-21 on page 116 shows the B2B Viewer displaying Show Document and Resend Document actions.

| | |
|---|---|
| <input checked="" type="radio"/> enabled <input type="radio"/> disabled | |
| <input type="text"/> | |
| */student07/b2b/partner-visibility?Access=r& Partner=partner2 | X |
| */student07/b2b/partner-visibility?Access=r& Partner=zzpartner2 | X |
| */student07/b2b/get-document?Access=x | X |
| */student07/b2b/resend-transaction?Access=x | X |
| <input type="text"/> | <input type="button" value="Add"/> <input type="button" value="Build"/> |

Figure 7-20 User group access policy explicitly allowing resend transactions and get document actions

| Transaction Set ID | | Transaction ID | Gateway Name | SenderName (ID) / Receiver (ID) | Inbound URL / Outbound URL | Input Time / Output Time | Result Code | MDN Status | MDN Time | MDN Received | Headers | Document ID | Document Type | MDN | Action |
|-------------------------------------|---|----------------|--------------|---------------------------------|----------------------------|--------------------------|-------------|---------------------|-----------------------|-----------------------|----------------|-------------|---------------|-----|------------------------|
| <input type="checkbox"/> | Show Document <input type="checkbox"/> Input Output Content MDN | | HubOwner | Sender: partner2 (partner2) | as2://9.42.170.220:5103/ | 2009-03-17 08:34:20.0 | Success | Sent (Positive) | 2009-03-17 08:34:20.0 | | (Show Headers) | | CustomXML | 0 | Resend |
| <input type="checkbox"/> | | | Partner2 | Sender: partner2 (partner2) | http://9.42.170.220:5106/ | 2009-03-17 08:34:20.0 | Success | Received (Positive) | | 2009-03-17 08:34:20.0 | (Show Headers) | | CustomXML | 0 | Resend |
| <input checked="" type="checkbox"/> | 153 | 20904 | HubOwner | Sender: zzpartner2 (zzpartner2) | as2://9.42.170.220:5103/ | 2009-03-17 08:34:02.0 | Success | Sent (Positive) | 2009-03-17 08:34:02.0 | | (Show Headers) | 000000002 | 850 | 0 | Resend |

Figure 7-21 B2B Viewer showing Show Document and Resend Document actions

7.2.4 Working with transactions in the B2B Viewer

After accessing the B2B Transaction Viewer, you can perform the following actions:

- ▶ Control the number of transactions to display per page.
- ▶ Define a query statement to filter which transactions to view.
- ▶ Resend transactions that did not complete.

Controlling display limits

In the viewer, use the Page Size list to control the number of transactions to display on each page. The default value is 50.

Filtering transactions

To display transactions that match a specific criteria, modify the query statement.

Modifying the query statement

To modify the query statement, use the following procedure:

1. Click **Modify Query**.
2. Use the fields to define the query statement, as needed. For information about the data, refer to “Data to modify the query statement”.
3. Click **Update Query**.

The viewer refreshes to display only the transactions that match the criteria in the query statement.

Data to modify the query statement

The control at the top of the query page filters the query results based on the processing status of the transaction. You can filter based on all transactions, successful transactions, or failed transactions. The viewer shows the processing

status for transactions in the Results Code column. Values used to modify the query statement are entered as string values (refer to Figure 7-22 on page 118):

- ▶ **Transaction ID:** The identifier for a specific transaction. Specify the gateway-generated identifier as shown in the viewer to limit transactions to the specific transaction.
- ▶ **Sender Name:** The name of a specific sender of B2B messages. The name is the name of the partner as defined in the B2B Partner Profile object. Specify the name of the partner as defined in the B2B Partner Profile object to limit transactions to the specific sender.
- ▶ **Receiver Name:** The name of a specific receiver of B2B messages. The name is the name of the partner as defined in the B2B Partner Profile object. Specify the name of the partner as defined in the B2B Partner Profile object to limit transactions to the specific receiver.
- ▶ **Gateway Name:** The name of a specific B2B Gateway object. Specify the name of the B2B Gateway object to limit transactions to the specific gateway.
- ▶ **MDN:** The status of the MDN. Valid values are 0 or 1.
- ▶ **Input Timeframe:** The time interval during which the gateway received the transaction:
 - Specify a start and end time to limit transactions to the explicit interval.
 - Specify a start time only to limit transactions from the start time onward.Specify each aspect of the time frame using the format as shown in the viewer. You cannot specify an end time only.
- ▶ **Output Timeframe:** The time interval during which the gateway sent the transaction:
 - Specify a start and end time to limit transactions to the explicit interval.
 - Specify a start time only to limit transactions from the start time onward.Specify each aspect of the time frame using the format as shown in the viewer. You cannot specify an end time only.

Modify Query

[Help](#)

Status:

Transaction ID:


Sender Name:


Receiver Name:

Gateway Name:


MDN:

Input Timeframe:
(YYYY-MM-DD HH:MM:SS)

Start: 

Stop: 

Output Timeframe:
(YYYY-MM-DD HH:MM:SS)

Start: 


Stop: 

Figure 7-22 Modify Query window

Resending transactions

To resend transactions, use the following procedure:

1. Select the transactions to resend.
2. Click **Resend Selected**.

7.3 B2B Data Persistence

We describe B2B Persistence in detail.

7.3.1 Transaction store

The XB60 requires a transaction store for persisting B2B transaction metadata and to provide state management for processing AS messages. The embedded database that is used for metadata persistence is not accessible by the users of the system outside of access through the B2B Transaction Viewer.

The major issue with persisting data to a database and hard drive is that the database or hard drive can fill up to capacity extremely quickly when supporting high transaction volumes. To prevent the database and hard drive from filling up with B2B data, you can set an Archive and Purge process for each B2B Gateway Service as described in “Archive tab” on page 102. The amount of disk drive space allowed for the local persistence store can be changed to control how much the database file will be allowed to grow; this setting pertains to all B2B data in all domains on the XB60. The B2B Persistence object is only available to the Admin user in the Default domain and can be configured by using the following procedure:

1. Log on to the Appliance as Admin into the Default domain.
2. Select **Objects** → **B2B Persistence** from the left navigation menu.
3. The Configure B2B Persistence view will be displayed as seen in Figure 7-23 on page 120.
4. You have the option of setting the Admin State and RAID Volume fields; you do not have to change the default under normal circumstances.
5. Change the Storage Size field to a higher number. Use a number ranging from 1024 to 65536 Megabytes. The Default is 1024 Megabytes.
6. Click **Apply** to save the object to the running configuration.
7. Optionally, click **Save Config** to save the object to the startup configuration.

Important: When you change the B2B Persistence Store value, you allocate the maximum size that the embedded database can grow on the hard drive. After you change the value of the B2B Persistence Store to a higher number, it cannot be changed to a lower number.

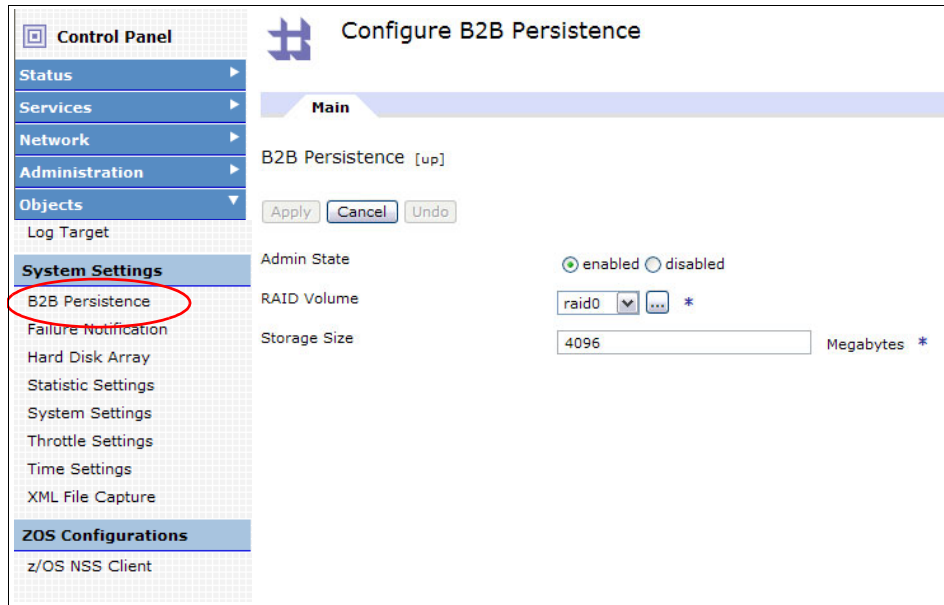


Figure 7-23 Configure B2B Persistence

7.3.2 Document storage

The DataPower XB60 device utilizes a pair of mirrored hard disk drives for metadata and document persistence. They contain an encrypted area and an area that is not encrypted. Transaction metadata extracted from each message is retained in the B2B Persistence store (database file) on the hard disk, which is the area of the hard disk that is not encrypted and which must not be used for document storage. This space can be shared with configuration and log files if the appliance is configured to use the `local:///ondisk/` directory. The default setup of the device does not use the hard disk for anything other than B2B message contents and metadata.

By default, the encrypted area of the hard disk drives is used for storing B2B payloads for non-repudiation and viewing purposes. The encrypted portion of the local drive is only 73 GB in size, so it becomes extremely important to Archive and Purge data frequently when using the default encrypted disk location. The XB60 also gives you the option of storing payloads off the device using either an NFS mount point or on an iSCSI disk subsystem.

The document storage location for messages is defined on a service-by-service basis in the Document Storage Location parameter in the B2B Gateway Main tab. This location needs to be a locally accessible file system: iSCSI, NFS, or RAID.

The following sections describe how to configure both iSCSI and NFS on the XB60.

Note: We advise that you do *not* set the document storage location to a directory on the flash drive (`local:///`). Storing B2B payload data on the flash drive can result in filling the flash drive to capacity.

iSCSI

DataPower XB60 appliance can connect to a Storage Area Network (SAN) using the iSCSI protocol starting with the 9004-based platforms. *SANs* are storage networks that focus on special network configurations that are designed for storage-specific network traffic. iSCSI is defined by Request for Comments (RFC) 3720 and stands for SCSI over Internet Protocol. Like other SAN protocols, iSCSI is a block network protocol. An initiator (client) connects to a target (server). From the initiator perspective, this connection looks like a SCSI device.

Currently, iSCSI on the XB60 appliance is implemented via a *host bus adapter* (HBA). The network ports eth1 and eth2 on the XB60 function as both Ethernet devices and as HBAs. When functioning as Ethernet devices, the physical ports are defined on eth1 and eth2 interfaces. When functioning as an HBA device, the physical ports are defined on iscsi1 and iscsi2. The iscsi1 and eth1 share the same physical Ethernet port, and the same goes for iscsi2 and eth2. From a configuration point of view, iscsi1 and eth1 are separate devices sharing the same physical port and require a different IP.

Key factors for using an external iSCSI device are:

- ▶ Data storage capacity compared to local drives.
- ▶ Flexibility and scalability when more storage space is needed.
- ▶ XB60 device failure will not cause any loss of historical payload data.

Note: Data is not encrypted on the iSCSI location. The external drive subsystem configuration in use is responsible for all data security.

iSCSI reference objects

This section outlines the objects that are needed to configure the XB60 device as an iSCSI initiator. The appliance, through the iSCSI HBA, can use the iSCSI protocol to communicate with the remote iSCSI server. The iSCSI HBA, acting as the software initiator, establishes connectivity, and when connected, an iSCSI session is started. The appliance provides two Ethernet ports that support iSCSI:

The following components need to be configured to use iSCSI for document storage:

- ▶ iSCSI host bus adapter object: The HBA establishes communications between the appliance and the remote iSCSI server.
- ▶ iSCSI target object: The target defines the connection information to the remote iSCSI server.
- ▶ iSCSI remote server: The remote server is needed to communicate with the iSCSI initiator using the iSCSI protocol. The iSCSI remote server does not need to be a real storage device. There are several free software targets that can be downloaded and easily configured.
- ▶ Initialized iSCSI volume object: Initializing the iSCSI volume allows it to be made active. The iSCSI volume must be disabled before it can be initialized.

Note: You can configure the iSCSI objects via the WebGUI, but you will need to use the CLI for setting this location on the B2B Gateway Service.

Configuring the iSCSI host bus adapter

This section describes the information that is needed to configure the iSCSI host bus adapter (HBA) on the device. The iSCSI host bus adapter (HBA) is the hardware that is responsible for the management of iSCSI communications. The iSCSI HBA initiates the iSCSI session between the DataPower appliance and the iSCSI remote target.

The HBA object can be found on the WebGUI by clicking **Objects** → **Network - Settings** → **iSCSI Host Bus Adapter** (refer to Figure 7-24).

iSCSI Host Bus Adapter: iscsi1 [up]

Apply Cancel Undo

Admin State enabled disabled

Comments

iSCSI Name

Use DHCP on off

Figure 7-24 iSCSI-enabled host bus adapter

We need the following information to enable the iSCSI HBA object:

- ▶ **iSCSI Name:** A valid iSCSI Name for this HBA instance. There is a predefined iSCSI Qualified Name (IQN), but it is not visible. If you leave the field blank, it defaults to the predefined name. It is possible to assign a iSCSI name as in Figure 7-25 on page 123. To view this value, select **Status** → **Other Network** → **iSCSI Host Bus Adapter Status**. There is currently no way to revert back to the default IQN name after the predefined iSCSI Name has been modified.

| HBA | Op-State | iSCSI Name | IP Address | Default Gateway |
|--------|----------|--|-----------------|-----------------|
| iscsi1 | up | iqn.2009-04.com.example:storage.disk2.sys1.xyz | 9.42.170.175/23 | 9.42.170.1 |
| iscsi2 | down | iqn.2000-04.com.qlogic:qle4062c.yk10ny887v9v.2 | 0.0.0.0/0 | 0.0.0.0 |

Figure 7-25 iSCSI host bus adapter status

- ▶ **DHCP (Dynamic Host Configuration Protocol):** This setting determines whether or not DHCP will be used for this interface. This setting is optional if a valid IP Address is supplied.
- ▶ **IP Address:** The IP address assigned to this interface followed by the subnet mask. The subnet mask can be in the Classless Inter-Domain Routing (CIDR) format as a suffix onto the end of the IP address or in dotted quad format. This setting is optional if DHCP is set to 0n. In our example, we use DHCP.
- ▶ **Default Gateway:** This field is optional. It is the default gateway for this interface.

Figure 7-26 shows you the default HBA objects.

Configure iSCSI Host Bus Adapter

[Refresh List](#)

| Name | Status | Op-State | Logs | Admin State | Comments |
|--------|--------|-------------|------|-------------|----------|
| iscsi1 | saved | down | | disabled | |
| iscsi2 | saved | down | | disabled | |

[Add](#)

Figure 7-26 Default HBA objects

As you can see in Figure 7-26 on page 124, there are two predefined interfaces that can be used for iSCSI: `iscsi1` and `iscsi2`. The Add button is grayed out, because you cannot add any additional interfaces for iSCSI. The parameters that we used to enable one of the iSCSI HBA interfaces are shown in Figure 7-27.

iSCSI Host Bus Adapter: `iscsi1` [up]

[Apply](#) [Cancel](#) [Undo](#)

Admin State enabled disabled

Comments

iSCSI Name

Use DHCP on off

Figure 7-27 The `iscsi1` HBA instance with a user-defined IQN name

Configuring the iSCSI target object

The iSCSI target object is depicted in Figure 7-28 on page 125 and has three required fields: the IQN name of the remote iSCSI server, the host name or the

IP address of the remote server, and the `iscsi1` HBA instance that we enabled in the previous section. The iSCSI target object waits for SCSI commands. An iSCSI target cannot initiate an iSCSI session. The iSCSI target is a connection instance of a remote iSCSI target.

The iSCSI target can be found by using the WebGUI in your application domain and clicking **Objects** → **Network - Settings** → **iSCSI Target**. Refer to Figure 7-28 on page 125.

The screenshot shows the configuration page for an iSCSI Target named 'iscsiITS0target' which is in an 'up' state. At the top, there are four buttons: 'Apply', 'Cancel', 'Delete', and 'Undo'. Below these are several configuration fields:

- Admin State:** Radio buttons for 'enabled' (selected) and 'disabled'.
- Comments:** A text input field containing 'Sample Target for ITS0'.
- Target Name:** A text input field containing 'iqn.2005-02.com.ricecake.iscsi:00' with an asterisk indicating it is required.
- Host:** A text input field containing '9.42.171.244' with an asterisk indicating it is required.
- Port:** A text input field containing '3260'.
- Host Bus Adapter:** A dropdown menu showing 'iscsi1' with a plus icon and an asterisk.
- CHAP:** A dropdown menu showing '(none)' with a plus icon.

Figure 7-28 iSCSI Target connection instance of the remote iSCSI target

Configuring the iSCSI Volume

The iSCSI Volume object is depicted in Figure 7-29 on page 126 and has three required fields: the directory for which the file system is mounted, the logical unit number (LUN) that is provided by the remote iSCSI server, and the iSCSI Target object that we defined in the previous section. The iSCSI Target object is actually the remote connection instance to the iSCSI server. After the object has been configured for the first time, you will need to issue the action “Initialize File System.” This action is visible from the WebGUI or the CLI by using the `init-fs` command. This action will partition and format the iSCSI device. After the object is in the “up” admin-state, it is mounted.

The iSCSI Volume can be found on the WebGUI in your application domain by selecting **Objects** → **Network - Settings** → **iSCSI Volume**. Refer to Figure 7-29 on page 126.

Note: You must *initialize the file system* on the Volume before enabling the object. Failure to do so will result in startup errors. The iSCSI volume must be disabled before it can be initialized.

iSCSI Volume: iscsiITSOVolume [up]

Apply Cancel Delete Undo

Admin State enabled disabled

Comments

Read-Only on off

Directory *

LUN *

iSCSI Target + ... *

Figure 7-29 iSCSI Volume

After the file system has been initialized and the volume object has been enabled, you will see the `foo` subdirectory under `local` and `logstore` (Figure 7-30). Each application domain contains these subdirectories. These subdirectories are not shared across application domains.

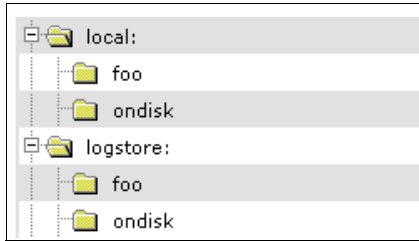


Figure 7-30 File system directory

Setting the Document Storage Location of the B2B Gateway

In order to store documents off the appliance, the Document Storage Location in the B2B Gateway can be set to store the B2B documents off-device on an iSCSI server or an NFS mount. However, the iSCSI and NFS mount locations cannot be set using WebGUI; they must be set in the CLI using the **doc-location setting** command, and they will not be visible in the WebGUI until it has been set on the B2B Gateway Service using the CLI method. Refer to Example 7-8 for using CLI to set the Document Storage Location to an iSCSI server on the B2BGateway Service called HubOwner, using the CLI. To learn more about the CLI, refer to the user documentation.

Example 7-8 Setting the Document Storage Location using the CLI

```
xb60[student05] (config b2bgw HubOwner)# show
  admin-state enabled
...
xb60[student05] (config b2bgw HubOwner)# doc-location local:///foo
xb60[student05] (config b2bgw HubOwner)# show
  admin-state enabled
  summary "Primary B2B Hub"
  priority normal
doc-location local:///foo
...
xb60[student05] (config b2bgw HubOwner)# exit
xb60[student05] (config)#
```

In Figure 7-31, you can see that now the Document Storage Location is set to the subdirectory foo. This subdirectory was created when we enabled the iSCSI Volume.

B2B Gateway: HubOwner [up]

General Configuration

Admin State enabled disabled

Comments

Document Storage Location ▼

XML Manager ▼

Figure 7-31 B2B Gateway Document Storage Location set to iSCSI server

NFS

The *Network File System* (NFS) protocol is another way of storing payload data off the appliance. NFS is a network file system protocol that allows a client application access to files over a network as though the network devices were attached to its local file system. Ports must be opened through the inner firewall to support NFS (2049 and 111 both TCP and User Datagram Protocol (UDP)).

NFS mounts can be statically or dynamically mounted. Dynamic mounts are constructed via URL in the form of: `dpnfs://hostname/path/file`, causing the directory `hostname:/path` to be automatically mounted by NFS. It remains mounted until it times out due to inactivity. Defining a static mount allows for the referencing of the NFS Static Mount object in the Document Storage Location URL and avoids the connection overhead associated with dynamic mounting. Mounted NFSs are exposed as a folder with the appliance's file systems. The following section provides configuration details about how to configure the Document Storage Location to write files to a static mount point defined on an external server.

Key factors for using a NFS mount point:

- ▶ Data storage capacity compared to local drives.
- ▶ Flexibility and scalability when more storage space is needed.
- ▶ XB60 device failure will not cause any loss of historical data.

Note: Data is not encrypted on the NFS mount point; the external drive subsystem configuration in use is responsible for all data security.

NFS reference objects

This section outlines the objects that need to be configured to use a static mount point to store copies of payload data:

- ▶ **NFS Client Settings:** This field contains the client properties for either the dynamic mount or static mount. This object must be enabled in the default domain by a device administrator and the Mount Refresh Time must be set. If authentication is required, DataPower supports NFS Version 4. This version of the protocol provides access to files on mounted file systems that use Kerberos security. To access the configuration object, click **Objects** → **Network - Settings** → **NFS Client Settings** in the left navigation menu.
- ▶ **NFS Static Mounts:** This object defines the connection information to the remote static mount.

Note: You can configure the NFS Static Mounts objects via the WebGUI, but you will need to use the CLI for enabling the B2B Gateway Service to write documents to this location.

Configuring the NFS Static Mounts

This section describes the information needed to configure the NFS Static Mounts. Defining a static mount allows you to reference the NFS Static Mount object in the document storage location URL.

The NFS Static Mounts object is depicted in Figure 7-32 on page 130 and has one required field: the Remote NFS Export. This field uses the following format *host:/path* (notice only a single slash is used), where *host* is the DNS name or IP address of the NFS server, and *path* is the path exported by the host to mount.

The NFS server must be configured to accept requests from the IP address of the DataPower XB60 device and any firewalls that are between the XB60 and the NFS must be configured to allow the connection. This example uses AUTH_SYS authentication, and the NFS server must also be configured to accept that form of authentication. Kerberos can alternately be used for authentication. It might be a better choice, because it provides data integrity and confidentiality if it is supported by the NFS server.

The iSCSi Target can be found in the WebGUI in your application domain by using the left navigation menu to select **Objects** → **Network - Settings** → **NFS Static Mounts**. Refer to Figure 7-32 on page 130.

NFS Static Mounts: nfsStaticMount [up]

Apply Cancel Delete Undo

Admin State enabled disabled

Comments

Remote NFS Export *

Local Filesystem Access on off

NFS Version

Transport Protocol ▼

Authentication Protocol ▼

Read-Only on off

Read Size bytes

Write Size bytes

Retransmission Timeout tenths of seconds

Maximum Retransmissions

Figure 7-32 Sample NFS Static Mounts object

NFS mount locations cannot be set using the WebGUI. An NFS mount location must be set in the CLI using the **doc-location setting** command and will not be visible until it has been set on the B2B Gateway Service. The URL must be in the format `nfs-[object name]`: where *object name* in our example is `nfs Static Mount`. When a path is not specified, the file will be written to wherever the static-mount points. Refer to Example 7-9 to use CLI to set the Document Storage Location on the B2B Gateway Service called HubOwner.

Example 7-9 CLI for NFS mount point

```

xb60[student05](config)# b2bgw HubOwner
xb60[student05](config b2bgw HubOwner)# doc-location nfs-b2bmount:
...
xb60[student05](config b2bgw HubOwner)# show
admin-state enabled
summary "Primary B2B Hub"
priority normal
doc-location nfs-b2bmount:
...

```

```
xb60[student05] (config b2bgw HubOwner)# exit
xb60[student05] (config)# dir
Options:
  local:
  logtemp:
  logstore:
...
nfs-b2bmount:
xb60[student05] (config)#
```

This setting will not be visible in the WebGUI. The Document Storage Location will be set to (default). The CLI is required for verification, as stated in the previous step. When the mount is created correctly, you see the “mounted directory” in our example, `nfs-b2bmount`, in the directory listing using the CLI.

7.3.3 Monitoring hard drive space

The hard drive space is shared across all B2B Gateway objects, and it is a good idea to monitor the available space. In Version 3.7.3 of the firmware, there is no WebGUI interface available for monitoring.

CLI

There are two commands that you can run in the CLI to monitor the disk space and the size of the persistence storage. Example 7-10 shows the results of the two commands. For more information about capacity planning for the XB60, log on to the WebSphere DataPower SOA Appliances Support site at:

<http://www-01.ibm.com/software/integration/datapower/support/>

Search for 1329746 in the Search Support field.

Example 7-10 Commands to monitor disk space

```
xb60# service show b2bp encrypted-space
B2B encrypted space: 28037 MB free of 28166 MB total
xb60# service show b2bp size
B2B persistence storage size used: 4032 KB
```



Configuration management

This chapter discusses several methods for managing the configuration of DataPower devices. We can leverage the way that the system works to make it self-configuring, which allows for rapid swapping of devices within a production environment, if the need arises. Topics include:

- ▶ DataPower file system directories and domains
- ▶ Devices, environments, and load balancers
- ▶ Configuration using the WebGUI, the command line interface (CLI), and the XML Management Interface
- ▶ Role Based Management (RBM)
- ▶ Package importing and exporting

8.1 Configuration management

Each DataPower device contains a configuration. The device is configured using objects that are hierarchically organized into *services*. These services expose ports for the consumption of traffic over supporting protocols, such as File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), MQ, Java Message Service (JMS), and Network File System (NFS). Services implement functionality, such as authentication and authorization of Web services, acceleration of Extensible Stylesheet Language Transformation (XSLT), and enterprise service bus (ESB) protocol mediation.

8.1.1 File system directories and domains

The DataPower file system is an encrypted RAM data source separated into several named directories. Refer to Figure 8-1 on page 135. Directories are used to manage configuration data, store XSLT stylesheets, capture logging events, manage cryptographic certificates and keys, and control other system functions. Configuration files are stored in the config directory, while custom data maintained by the user is stored in the local directory. The device stores most of its required files in the store: directory. For a complete description of the file system, refer to the *Administrator Guide* for your device. The AdministratorGuide-v1.pdf is provided on the CD shipped with the device or it can be downloaded from the IBM WebSphere DataPower Product Documentation Portal Web page:

<http://www-01.ibm.com/support/docview.wss?rs=2362&uid=swg21377654>

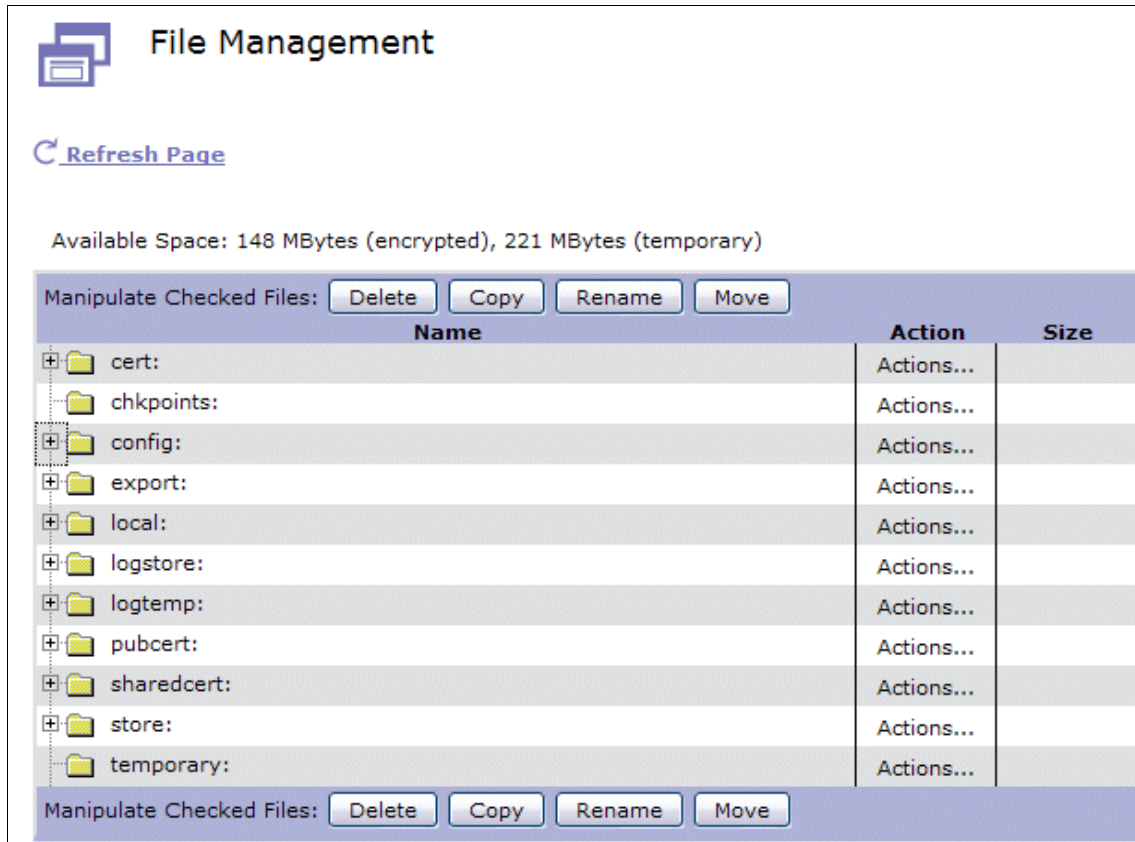


Figure 8-1 DataPower file system

8.1.2 Startup sequence for DataPower

Before discussing the methods available for configuration management, it is important to describe what happens at startup time within a DataPower device.

When a DataPower appliance first boots, it loads a startup configuration (from `config:///autoconfig.cfg`, which can be modified using the `boot config` command) in the default domain. All the information in the configuration file is in command line interface (CLI) format. It is executed *synchronously*, one line at a time, until initialization is complete, which is why all the device-level initialization precedes any other domain statements in this file.

Therefore, at the very least, the configuration file must define the IP addresses for each Ethernet port and Domain Name System (DNS) names for resolution. After that, domain definitions and other artifacts can be set. However, we

recommend that this file be kept small so that only critical values are placed within it in order to improve readability. Additional configuration files can be nested in a daisy-chain fashion as necessary to complete the configuration of the device.

Note: The lines in this file are executed in sequence, so basic properties, such as IP address, must come before anything else. Object order is significant, too. Subordinate objects, such as the match rule, must be defined prior to parental objects, such as the XML firewall. Compare this method to the SOMA/XML system where the XML files are not necessarily executed in a strict order. Using the CLI configuration files removes this dependency.

Example 8-1 shows a snippet from the configuration file of a business-to-business Gateway (B2BGW) object called HubOwner. The snippet defines an AS2 Front Side Handler (FSH), a HTTP FSH, and three partner profiles: two external profiles and one internal profile, among other objects.

Example 8-1 Example of ASCII configuration file for B2BGW called HubOwner

```
%if% available "b2bgw"

b2bgw "HubOwner"
  admin-state disabled
  summary "Primary B2B Hub"
  doc-location local:///
  as-fsph
    front-protocol HubOwner_AS2
    mdn-receiver
  exit
  as-fsph
    front-protocol HubOwner_http
    mdn-receiver
  exit
  as2-mdn-url "http://127.0.0.1:5103"
  b2b-profile
    profile HubOwner_Int
  exit
  b2b-profile
    profile Partner_Ext
  exit
  b2b-profile
    profile Partner2_Ext
  exit
```



```
arch-mode purgeonly
arch-document-age 5
arch-backup-documents ""
xpath-routing CustomXML
exit
```

8.2 Configuration options

There are several ways to configure the device. The solution requirements dictate your method to configure the device. They consist of graphical tools and automated/scripted and programmatic processes. You can use these methods in combination, and all methods result in a similar implementation on the device.

The WebGUI and the CLI through Secure Shell (SSH) are the two major ways of administering configurations. For every operation that can be done in the WebGUI, there is a corresponding operation in the CLI. Refer to the *Command Reference Guide* for your device for details. XB-1.0.0-CommandReference.pdf is provided on the CD shipped with the device or it can be downloaded from the IBM WebSphere DataPower Product Documentation Portal Web page:

<http://www-01.ibm.com/support/docview.wss?rs=2362&uid=swg21377654>

Regardless of the configuration method used, the end result of each operation is a modification to the onboard device configuration. This data is expressed as an ASCII file resident on the device's encrypted RAM file system, and this data contains a list of CLI commands. If the WebGUI or XML Management Interface was used, the execution of the interface is translated into the corresponding CLI commands. Upon the restart of the device, the designated configuration file is loaded, and services and objects are reconstructed from its content and references to external resources. This onboard configuration file can be downloaded, edited, and reloaded onto the device.

Important: The WebGUI is typically used in the development phase of a project. Configurations are usually exported or imported using the other mechanisms (CLI or XML Management Interface) as you get closer to the test phase and production staging.

8.2.1 WebGUI interface

The WebGUI interface is the simplest management interface to use. On most pages, a help link provides online help through a pop-up browser window. Most fields also provide in-line help when selected. Lastly, the two-step process for

committing configuration changes provides an opportunity to discard changes. Refer to the *Administrator Guide* for your device for a complete description. The AdministratorGuide-v1.pdf is provided on the CD shipped with the device or it can be downloaded from the IBM WebSphere DataPower Product Documentation Portal Web page:

<http://www-01.ibm.com/support/docview.wss?rs=2362&uid=swg21377654>

8.2.2 Command line interface

The command line interface (CLI) provides a simple but powerful management interface. Its syntax is familiar to terminal users on a UNIX®-like environment. Unlike the WebGUI, configuration changes are immediately committed. An **undo** command allows administrators to revert to a previous configuration. All administrators need to be familiar with basic CLI commands, because this management interface is the only interface that is available on first use to enable one of the other management interfaces by using the **configure terminal** command.

The CLI is a streamlined yet powerful system for controlling every facet of the appliance. Although it looks like a command shell, there is no compiler or interpreter to run arbitrary code. The alias function is a macro for multiple CLI commands, and the exec function executes configuration scripts that are limited to the device itself.

In the global configuration mode, the administrator can create, modify, or remove any DataPower service or interface that can be found in the WebGUI.

The CLI is not a true scripting facility, such as Perl, Tcl, or ANT. The **alias** CLI command does allow a simple form of defining a set of commands to execute. Automating the repetition of tasks in a scripting language is much easier than using a Web GUI. *Scripting* in this sense is the sequencing of commands. Example 8-2 shows a CLI shell file that imports a WS-Proxy to the Demo application domain of the device.

Example 8-2 CLI shell file

```
ssh DataPowerIP<<EOF
admin
passwOrd
config
copy -f https://HTTPServerIP/HRServiceExport.zip
temporary:///Demo/HRServiceExport.zip
passwOrd
import-package HRServiceImport
auto-execute off
```

```
destination-domain Demo
source-url temporary:HRServiceExport.zip
exit
import-execute HRServiceImport
no import-package HRServiceImport
write mem
y
exit
exit
EOF
```

Tasks executed in Web GUI require the user to remember what they performed, especially when they need to undo actions. Administrators can create an undo script to ensure that changes are undone.

Many options exist for migrating changes from development to production. For example, using the Web GUI, administrators can create the domain configuration and store it on a Web server. The domain on the appliance can reference this configuration on startup. Another option for importing shared objects into a domain is to use packages. The *package* contains the set of shared objects used by several domains.

These methods can be used to migrate changes from development to production. They might not be the preferred methods, because they cannot be audited and the consistency in the files cannot be guaranteed.

8.2.3 XML Management Interface

You can also use the XML Management Interface (MI).

XML Management Interface description

The XML Management Interface provides a structured language for sending a batch of configuration commands. This interface allows for the rapid automated configuration of new application domains or entire DataPower service-oriented architecture (SOA) Appliances. A complete IBM Redpaper publication has been written about this interface: *WebSphere DataPower SOA Appliance: The XML Management Interface*, REDP-4446.

The XML Management Interface allows appliance management using different XML-based interfaces and specifications. The SOAP Management URI (SOMA) interface enables management using SOAP over HTTPS. The SOMA interface provides advantages over other approaches. Because it is programmatic, it can be used to build a custom management application using a programming language, such as Java. Also, it executes remotely without any tools as

compared to the WebGUI, which requires a Web browser, and CLI, which requires an SSH client.

The SOAP interface extends its functionality to third-party Web service clients. It involves the programmatic modification of configuration data and can be based on external conditions and events. For example, a brokerage organization can define a service level agreement (SLA) that limits the number of trades that a client can execute in a given time frame. However, based on market conditions, the parameters of this SLA can change.

The SOAP interface demonstrates a potential use of the XML Management application programming interface (API). This facility allows for the authenticated real-time modification of configuration data via SOAP messages sent over a secured HTTPS interface. All the functions of the WebGUI and CLI are supported using the XML Management API. Each request is packaged as a SOAP message.

Formatting of request SOAP documents is derived via the Web Services Description Language (WSDL) and XML Schema Definition (XSD) documents available from the store directory of the device. It also provides configuration management tools. For information regarding request formats, refer to the WebGUI documentation for your device.

XB60-specific functionality

In addition to the standard XML Management API, there are new functions that are tailored to B2B business scenarios. Creation of trading partner profiles, creation of partner profile groups, creation of B2B Gateways, and so forth are all functions that can be addressed programmatically via the XML Management Interface.

This flexibility, ease of use and decoupling from the WebGUI interface can have advantages for administrators who handle many partner profiles, for instance, the automated creation of hundreds of partner profiles as compared to the manual entering of each profile in the WebGUI.

Creating B2B profiles

The SOAP request in Example 8-3 shows the creation of B2B partner profiles through the use of the XML Management Interface. This example creates two partner profiles called `Test_Ext` (external partner) and `Test_Int` (internal partner). These profiles are created in the domain called `student07`. Notice the `student07` attribute in the `dp:request` element.

Example 8-3 XML Management Interface request creating partner profiles

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
```

```

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:man="http://www.datapower.com/schemas/management">
<SOAP-ENV:Header />
<SOAP-ENV:Body>
  <dpmgmt:request domain="student07"
    xmlns:dpmgmt="http://www.datapower.com/schemas/management">
    <dpmgmt:set-config>
      <B2BProfile name="Test_Ext"
        xmlns:env="http://www.w3.org/2003/05/soap-envelope"
        xmlns:dp="http://www.datapower.com/schemas/management">
        <mAdminState>enabled</mAdminState>
        <ProfileType>external</ProfileType>
        <BusinessIDs>testhubbin</BusinessIDs>
        <Destinations>
          <DestName>HubBin_FTP</DestName>
          <DestURL>
            ftp://userid:pwd7@127.0.0.1:5117
          </DestURL>
          <EnabledDocType>
            <EnableXML>off</EnableXML>
            <EnableX12>off</EnableX12>
            <EnableEDIFACT>off</EnableEDIFACT>
            <EnableBinary>on</EnableBinary>
          </EnabledDocType>
          <SSLProxy />
          <OverrideTimeout>120</OverrideTimeout>
          <EnableFTPSettings>on</EnableFTPSettings>
          <UserName />
          <Password />
          <Passive>pasv-req</Passive>
          <AuthTLS>auth-off</AuthTLS>
          <UseCCC>ccc-off</UseCCC>
          <EncryptData>enc-data-off</EncryptData>
          <DataType>binary</DataType>
          <SlashSTOU>slash-stou-on</SlashSTOU>
          <QuotedCommands />
          <SizeCheck>size-check-optional</SizeCheck>
          <ASCompress>off</ASCompress>
          <ASCompressBeforeSign>off</ASCompressBeforeSign>
          <ASSendUnsigned>off</ASSendUnsigned>
          <ASEncrypt>off</ASEncrypt>
          <ASEncryptCert />
          <ASMDNRequest>off</ASMDNRequest>
          <ASMDNRequestAsync>off</ASMDNRequestAsync>
          <AS2MDNRedirectURL />

```

```

<AS3MDNRedirectURL />
<ASMDNRequestSigned>off</ASMDNRequestSigned>
<Retransmit>off</Retransmit>
<ACKTime>1800</ACKTime>
<MaxResends>3</MaxResends>
</Destinations>
<InboundRequireSigned>off</InboundRequireSigned>
<InboundRequireEncrypted>
  off
</InboundRequireEncrypted>
<OutboundSign>off</OutboundSign>
<OutboundSignDigestAlg>sha1</OutboundSignDigestAlg>
</B2BProfile>
<B2BProfile name="Test_Int"
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:dp="http://www.datapower.com/schemas/management">
  <mAdminState>enabled</mAdminState>
  <ProfileType>internal</ProfileType>
  <BusinessIDs>testpartnerbin</BusinessIDs>
  <Destinations>
    <DestName>PartnerBin_MQ</DestName>
    <DestURL>dpmq://XB60/?RequestQueue=Q12</DestURL>
    <EnabledDocType>
      <EnableXML>off</EnableXML>
      <EnableX12>off</EnableX12>
      <EnableEDIFACT>off</EnableEDIFACT>
      <EnableBinary>on</EnableBinary>
    </EnabledDocType>
    <SSLProxy />
    <OverrideTimeout>120</OverrideTimeout>
    <EnableFTPSettings>off</EnableFTPSettings>
    <UserName />
    <Password />
    <Passive>pasv-req</Passive>
    <AuthTLS>auth-off</AuthTLS>
    <UseCCC>ccc-off</UseCCC>
    <EncryptData>enc-data-off</EncryptData>
    <DataType>binary</DataType>
    <SlashSTOU>slash-stou-on</SlashSTOU>
    <QuotedCommands />
    <SizeCheck>size-check-optional</SizeCheck>
    <ASCompress>off</ASCompress>
    <ASCompressBeforeSign>off</ASCompressBeforeSign>
    <ASSendUnsigned>off</ASSendUnsigned>
    <ASEncrypt>off</ASEncrypt>
  </B2BProfile>

```

```

    <ASEncryptCert />
    <ASMDNRequest>off</ASMDNRequest>
    <ASMDNRequestAsync>off</ASMDNRequestAsync>
    <AS2MDNRedirectURL />
    <AS3MDNRedirectURL />
    <ASMDNRequestSigned>off</ASMDNRequestSigned>
    <Retransmit>off</Retransmit>
    <ACKTime>1800</ACKTime>
    <MaxResends>3</MaxResends>
  </Destinations>
  <InboundRequireSigned>off</InboundRequireSigned>
  <InboundRequireEncrypted>
    off
  </InboundRequireEncrypted>
  <OutboundSign>off</OutboundSign>
  <OutboundSignDigestAlg>sha1</OutboundSignDigestAlg>
</B2BProfile>
</dpmgmt:set-config>
</dpmgmt:request>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

To invoke the the command in Example 8-3 on page 140 through the XML Management Interface, we can issue the command using the cURL utility:

```

curl --data-binary @CreateTestProfiles.xml
https://<IP>:5550/service/mgmt/3.0 -k -u admin:<pass>

```

Add partner profiles to an existing B2B Gateway

The snippet of code in Example 8-4 shows how to add existing profiles (that were perhaps created in Example 8-3 on page 140) to an existing B2B Gateway object. The profiles Partner_Int, Partner_Ext, and Partner2_Ext will be added to the existing B2B Gateway called MyTest.

Important: Remember that invoking a script, such as the script in Example 8-4, only associates the profiles that are listed in the script to the B2B Gateway. All existing profiles are overwritten. If you want to keep existing profiles, be sure to add them to the script.

Example 8-4 Adding partners to an existing B2B Gateway

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:man="http://www.datapower.com/schemas/management">

```

```

<SOAP-ENV:Header />
<SOAP-ENV:Body>
  <dpmgmt:request domain="student07"
xmlns:dpmgmt="http://www.datapower.com/schemas/management">
    <dpmgmt:modify-config>
      <B2BGateway name="MyTest">
        <B2BProfiles>
          <PartnerProfile>Partner_Int</PartnerProfile>
          <ProfileEnabled>on</ProfileEnabled>
          <ProfileDest>HubBin_FTP</ProfileDest>
        </B2BProfiles>
        <B2BProfiles>
          <PartnerProfile>Partner_Ext</PartnerProfile>
          <ProfileEnabled>on</ProfileEnabled>
          <ProfileDest>HubBin_FTP</ProfileDest>
        </B2BProfiles>
        <B2BProfiles>
          <PartnerProfile>Partner2_Ext</PartnerProfile>
          <ProfileEnabled>on</ProfileEnabled>
          <ProfileDest>HubBin_FTP</ProfileDest>
        </B2BProfiles>
      </B2BGateway>
    </dpmgmt:modify-config>
  </dpmgmt:request>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Advanced XML management concepts

WebSphere DataPower SOA Appliances contain a powerful management framework that is accessible through XML messages that are sent over HTTPS. These messages can be chained and scripted in order to perform automated configuration management and operations on the device.

In order to perform configuration management through the XML Management Interface, the interface must first be enabled through the CLI or WebGUI. After this enablement is done, the XML commands can be sent to the specified host/port/URL. In Example 8-5 and Example 8-6 on page 145, we perform a substitution on the file to create a new domain with its own user and group defined.

Example 8-5 Initial XML file

```

<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

```



```

<env:Body>
  <dp:request xmlns:dp="http://www.datapower.com/schemas/management">
    <dp:set-config>
      <Domain name="%domainName%">
        <UserSummary>%domainDesc%</UserSummary>
        <NeighborDomain
class="Domain">%domainNeighbor%</NeighborDomain>
        </Domain>
        <UserGroup name="%groupName%">
          <UserSummary>%groupDesc%</UserSummary>
%groupPolicies%
        </UserGroup>
        <User name="%userName%">
          <Password>%userPass%</Password>
          <GroupName>%groupName%</GroupName>
          <AccessLevel>%userAccess%</AccessLevel>
          <UserSummary>%userDesc%</UserSummary>
        </User>
      </dp:set-config>
    </dp:request>
  </env:Body>
</env:Envelope>

```

Example 8-6 is a shell script that can make the substitutions.

Example 8-6 Shell script file

```

#!/bin/ksh
domainName=${1:-'testDomain'}
domainDesc=${2:-'testDomain for applications'}
domainNeighbor=${3:-'default'}

userName=${4:-'testUser'}
userPass=${5:-'passw0rd'}
userDesc=${6:-'test user for this domain'}
userAccess=${7:-'group-defined'}

groupName=${8:-'testGroup'}
groupDesc=${9:-'testGroup for application domain'}
groupPolicy01=${10:-'*/$domainNeighbour/*Access=r'}
groupPolicy02=${11:-'*/$domainName/*Access=r+w+a+d+x'}
groupPolicies=""
<AccessPolicies>$groupPolicy01</AccessPolicies>
<AccessPolicies>$groupPolicy02</AccessPolicies>
,,

```

```
sed -e "s/%domainName%/$domainName/" \  
-e "s/%domainDesc%/$domainDesc/" \  
-e "s/%domainNeighbor%/$domainNeighbor/" \  
-e "s/%userName%/$userName/" \  
-e "s/%userName%/$userName/" \  
-e "s/%userDesc%/$userDesc/" \  
-e "s/%userAccess%/$userAccess/" \  
-e "s/%groupName%/$groupName/" \  
-e "s/%groupPolicies%/$groupPolicies/" \  
-e "s/%groupDesc%/$groupDesc/" sampleTemplate.xml > domain.xml
```

This newly created file (domain.xml) can then be sent to the XML Management Interface using the cURL utility:

```
curl --data-binary @domain.xml https://<IP>:5550/service/mgmt/current  
-u admin:<pass> -k
```

In this statement, <IP> is the address of your DataPower device, and <pass> is your admin password for this device.

Note: In order to provide for a cleaner configuration that minimizes migration issues, we recommend that you use the host alias instead of the dot decimal address in services that expose external ports. Also, you need to use an environment-specific DNS when possible rather than a dot decimal address and use static hosts to handle DNS aberrations. In order to eliminate extra work, migrate only those objects that require migration.

For a more in-depth discussion of configuration for High Availability, configuration promotion, and control, refer to the following article, "Managing WebSphere DataPower SOA Appliance configurations for High Availability, consistency, and control," by John Rasmussen, 16 January 2008, developerWorks, at:

http://www.ibm.com/developerworks/websphere/library/techarticles/0801_rasmussen/0801_rasmussen.html

8.3 Role Based Management (RBM)

RBM controls the relationships between authenticated users and resources. Users are authenticated either by a remote authentication system or by the DataPower appliance. The RBM policy determines whether to allow an authenticated user to access specific resources.

When authentication uses a remote authentication system, such as a Lightweight Directory Access Protocol (LDAP) server, RBM extracts the identity of the authenticated user, maps the identity to a credential, and determines whether to authorize access to the resource based on the credential. If a problem occurs during remote authentication, RBM can use one or more locally defined fallback users.

Figure 8-2 shows the basic components of RBM and how they relate to each other.

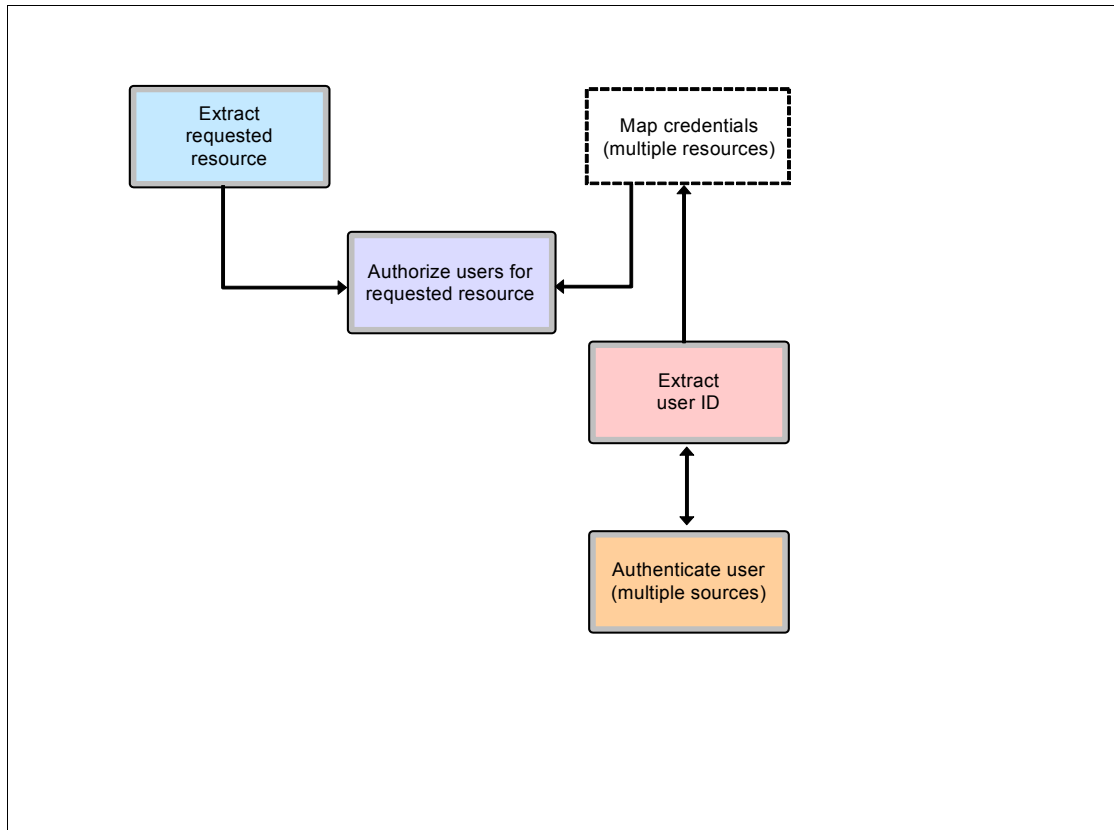


Figure 8-2 Basic components of RBM and their relationships

When authentication is local, authentication is by user name and password. The group in which the user is a member determines whether to authorize access to the resource. Users who are not members of a group are not under RBM control.

The RBM policy uses access profiles to determine authorization to resources. An *access profile* is made up of one or more access policies. Each access policy defines which privileges to provide to a single resource. An access policy can

use wildcard characters in regular expressions to define the same set of privileges to multiple resources. Because RBM distances access policies from individual users, you can modify an access profile that affects a collection of users instead of modifying each user individually. For example, you can modify the access profile in a user group to change resource authorization for all members of that group. Alternatively, you can change the access profile associated with a credential to modify all users who map to that credential.

8.4 Package importing and exporting

The backup and export utility copies specified configuration data from the appliance to a file in the `export:` directory. You can optionally download the file to your workstation.

Note: User accounts, certificates and keys, files referenced by error rules, log files, firmware files, and any object or file that is not accessible to the user running the export (because of user and group permissions) are not exported (included in the export package). The Admin account must perform the export.

The `export.zip` (default file name) file can be accessed from the `export:` directory on the flash file system. Click the File Management icon. Select the export directory. The contents of this directory contain the file that was created.

Important: Exported configuration data must not be imported to an appliance with an earlier firmware release level. Between releases, configuration data for properties can change. If you attempt to import configuration data from an appliance of a later firmware release level into an appliance of an earlier firmware release level, the operation might report success, but the configuration data might not be the same. Therefore, as a best practice, use this utility to exchange configuration data among appliances of the same release level.

Entire appliance configuration

The entire XB60 DataPower appliance configuration can be exported for archive and recovery purposes. The export function copies configuration data from the device to a development workstation.

The exported appliance configuration can be imported to the same XB60 appliance for recovery purposes or to a separate XB60 appliance of the same release level for cloning purposes in the case of a High Availability configuration.

The Export Configuration can be accessed by clicking **Export Configuration** in the Files and Administration section of the Control Panel. Figure 8-3 displays the export options offered when exporting the system information.

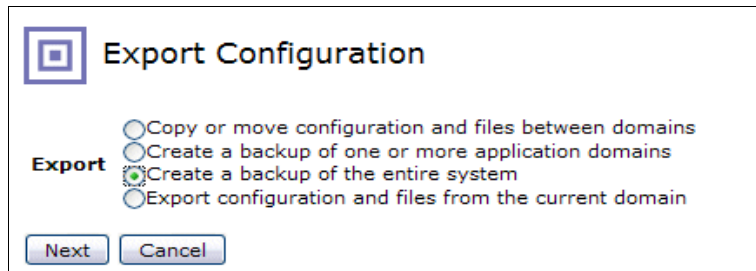


Figure 8-3 Export Configuration

Entire domain

Best practices dictate that individual domains are backed up on a regular basis. These exports can then be archived on an external Software Configuration Management (SCM) system and available for reimporting into an XB60 domain.

The export procedure can also be scripted via CLI commands to create an automated process.

Specific configuration objects

You can copy configured objects, such as B2B Gateway Services and trading partner profiles.

B2B Gateway Services

Configured objects, such as a B2B Gateway Service, can also be copied to leverage this configuration in other domains on the current XB60 or on another XB60 entirely. Export options allow the inclusion of all referenced objects to the main object or just individual objects. There is a similar option to include referenced local files that are used by an exported object.

These exported objects can be saved in XML format or, more commonly, as a compressed ZIP format.

Referenced object options are:

- ▶ Include all objects required by selected objects
- ▶ Include only the selected objects

Export file options are:

- ▶ Export all local files
- ▶ Export files that are referenced by selected objects

- ▶ Export no files

Trading partner profiles

Much like the ability to export whole B2B Gateway Services, we can, at a more granular level, export objects, such as trading partner profiles, and import these objects into another domain or to another XB60 appliance. Exporting trading partner profiles allows rapid configuration of other B2B Gateway Services that can make use of the existing partner profile information.

To export specific trading partner profiles, use the Export Configuration option as seen in Figure 8-3 on page 149 and select **Export configuration and files from the current domain**. In the Export Configuration view, you can then select the specific trading partner profiles to export as seen in Figure 8-4.

The screenshot shows the 'Export Configuration' dialog box. At the top, there are fields for 'Comment', 'Deployment Policy' (set to '(none)'), and 'Export File Name' (set to 'export'). Below these are radio buttons for 'To' (XML Config and ZIP Bundle, with ZIP Bundle selected). To the right, there are sections for 'Configuration' (Currently running configuration selected), 'Referenced Objects' (Include all objects required by selected objects selected), and 'Export Files' (Export files referenced by selected objects selected). The main area is titled 'Select configuration objects to export' and contains two list boxes: 'Objects' and 'Selected Objects'. The 'Objects' list includes 'B2B Partner Profile', 'All B2B Partner Profile Objects', 'HubOwner_Ext', 'HubOwner_Int', 'Partner2_Ext', 'Partner2_Int', 'Partner_Ext', and 'Partner_Int'. The 'Selected Objects' list includes 'B2B Partner Profile HubOwner_Ext' and 'B2B Partner Profile HubOwner_Int'. Navigation buttons (>, <, <<) are between the lists. At the bottom are 'Back', 'Show Contents', 'Next', and 'Cancel' buttons.

Figure 8-4 Exporting trading partner profiles



Troubleshooting the appliance

In this chapter, we describe the tools that are available to troubleshoot configuration errors with the DataPower XB60 appliance and examine the appliance logs to illustrate how to troubleshoot application problems. We list and discuss the tools that are available for troubleshooting message flow and services and use the business-to-business (B2B) Transaction Viewer to investigate the messages that are sent and received on B2B Gateways. We also discuss life cycle considerations.

We include a description of common configuration mistakes and suggested solutions followed by recommendations about seeking help in product documentation and IBM support.

9.1 Overview

There are many factors that can lead to DataPower XB60 errors, including network connectivity, unreachable destinations, incorrect configuration, port conflicts, and low performance. The DataPower XB60 provides several tools to aid in solving these issues. All of the following tools are accessible from the Control Panel: Ping, TCP Connect, System Logs, and B2B Transaction Viewer. We will explore ways to configure log settings for the correct amount of detail and to leverage the B2B Transaction Viewer to analyze B2B traffic flow.

Over the product life cycle, there are varying strategies to follow when troubleshooting. This chapter closes with general recommendations about what tools to use in the various life cycle phases.

9.2 Troubleshooting the network setup

There are several tools available to help you troubleshoot the network setup.

9.2.1 Ping and TCP Connect

Prior to any other debug steps, it is important to verify network connectivity. You can verify network connectivity by checking the connectivity between the internal and external destination with the Ping and TCP Connect tools. Ensure that both source and target locations are running. Then, verify that the target IP destination address is reachable from the DataPower appliance. Also, ensure that the target ports are reachable using TCP Connect.

You can test connectivity to the remote host by using these two methods, which are shown in Figure 9-1 on page 153. Enter the IP address or host name. When the appliance cannot connect to the back-end application server, use this tool. Use the TCP Connection Test. As an example, an FTP server uses port number 21, so we click **TCP Connection Test** to check if the FTP server is up and running. Note that the **Ping Remote** command is successful, although the FTP server is down. We see that the FTP server cannot be reached.

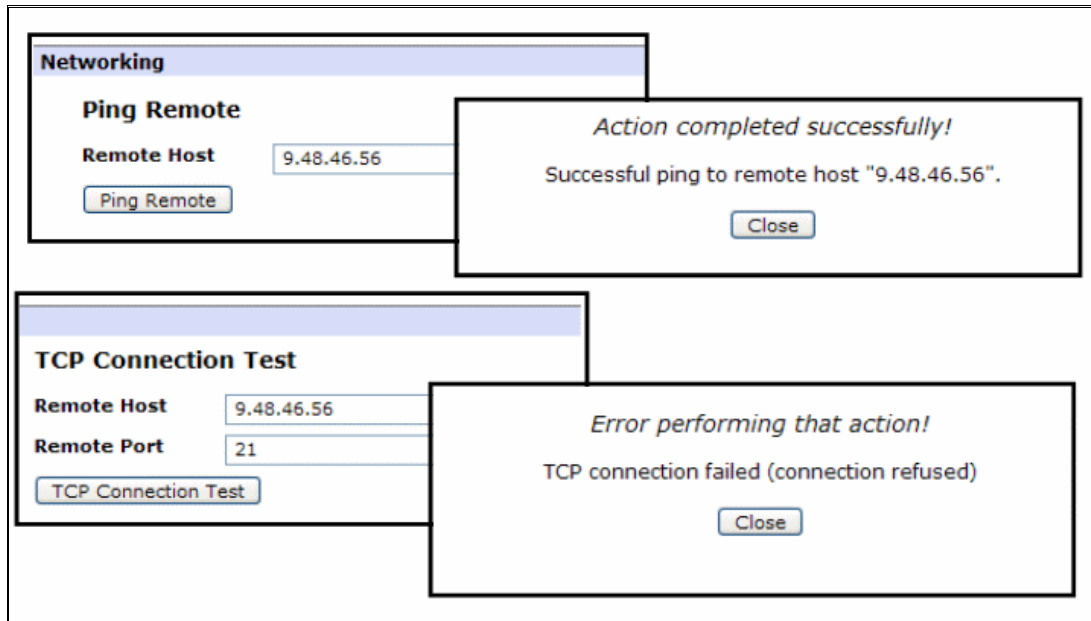


Figure 9-1 Ping Remote and TCP Connection Test

When you receive the error, start the FTP server and repeat the test. The result is then successful.

9.2.2 Packet Capture

In certain instances, it is necessary to capture the full network-level exchange between an appliance and another resource in the network in order to understand what is happening. A packet trace contains a capture of network traffic in *pcap* format and is stored in the temporary directory. The tool is useful when troubleshooting network connectivity, TCP sequencing, or other network-level problems.

To capture network packets:

1. Switch to the default domain.
2. On the Control Panel, click the Troubleshooting icon.
3. Look for the Networking section.

You can initiate a packet capture session on an Ethernet or virtual LAN (VLAN) interface. The options are set in the Packet Capture section, which is shown in Figure 9-2 on page 154. The appliance initiates a default packet capture session on the target interface. The appliance closes the session after 30 seconds or

after the capture of 10 MBs of data, whichever occurs first. Alternatively, click Stop Packet Capture to end a packet capture session. By default, captured data is stored in the capture.pcap file in the temporary: directory (temporary:///capture.pcap).

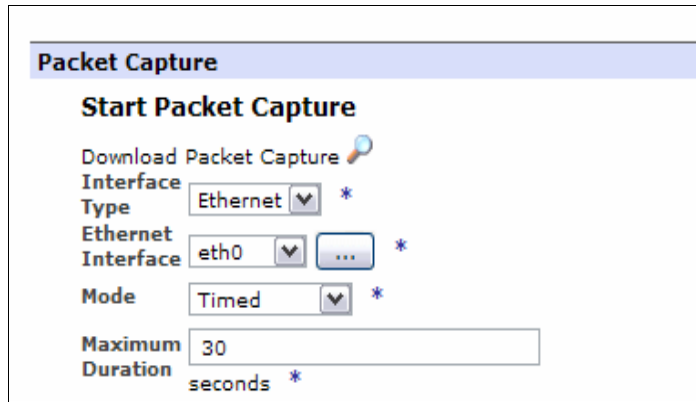


Figure 9-2 Packet Capture tool

The data in a packet capture is saved in the pcap format. Use a utility, such as tcpdump or ethereal, to interpret the file. Refer to Figure 9-3 for an example of a pcap capture that is viewed in ethereal.

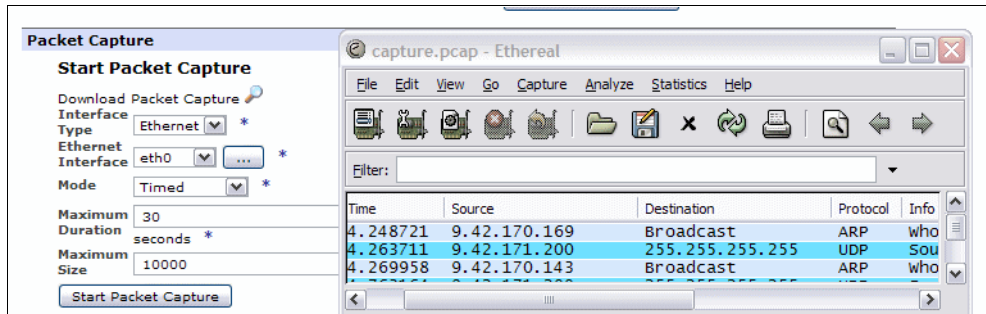


Figure 9-3 Packet Capture example

9.3 Using built-in tools to diagnose appliance problems

After network connectivity issues are ruled out, there are various tools resident on the DataPower B2B Appliance that are useful in identifying application and configuration problems.

9.3.1 Using the B2B Transaction Viewer

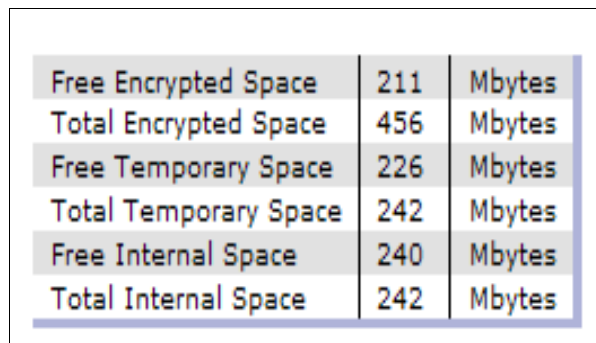
When the viewer is configured for external access, external partners can access the viewer to view transactions. Make sure that the B2B Transaction Viewer is configured correctly and refer to Chapter 7, “B2B configuration options” on page 91 for directions for assigning the correct privileges to partners.

After accessing the viewer through a browser, you can control the number of columns viewed.

9.3.2 Checking the appliance status

To get a general assessment of the appliance’s health, check the file system information to ensure that no limits are reached. The Filesystem Information object can provide information about why the appliance stopped processing messages. If the logging facility fills up the available file space, the logging facility can no longer write events. When the logging facility cannot write messages, the logging facility prevents the processing of incoming messages.

From the WebGUI, select **Status** → **System** → **Filesystem Information** to see the information displayed as in Figure 9-4.



| | | |
|-----------------------|-----|--------|
| Free Encrypted Space | 211 | Mbytes |
| Total Encrypted Space | 456 | Mbytes |
| Free Temporary Space | 226 | Mbytes |
| Total Temporary Space | 242 | Mbytes |
| Free Internal Space | 240 | Mbytes |
| Total Internal Space | 242 | Mbytes |

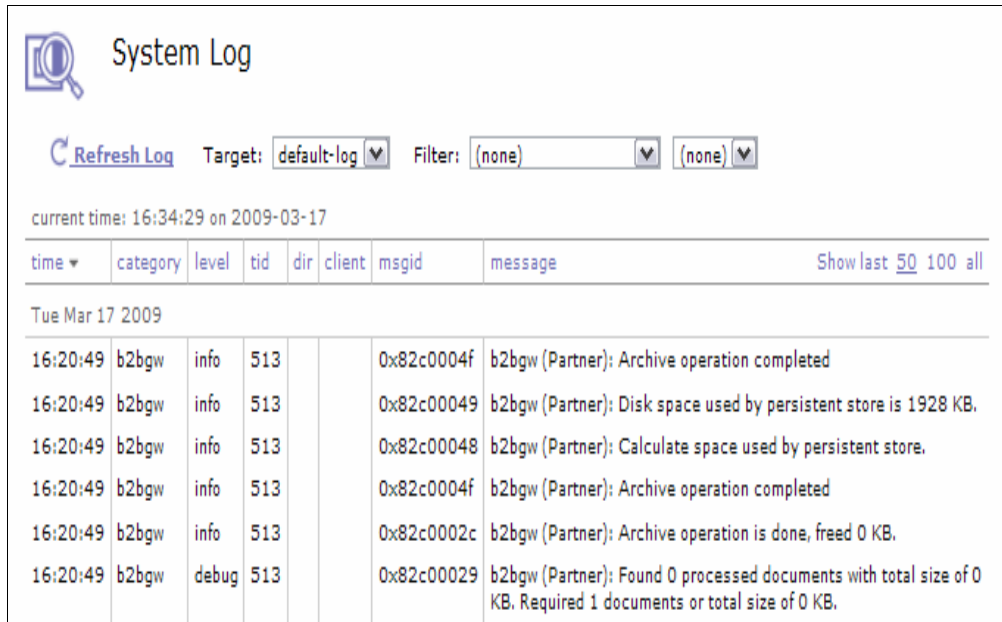
Figure 9-4 Filesystem Information

Also, check the System usage by selecting **Status** → **System** → **System Usage**.

9.3.3 Checking the system log

Logging can be viewed at the system-wide level or at the domain-specific level. Logging is the primary means to determine the cause of a problem and to verify

when a particular problem is solved. Refer to Figure 9-5 for an example of a system-wide log.



The screenshot shows the 'System Log' interface. At the top, there is a magnifying glass icon and the title 'System Log'. Below the title, there is a 'Refresh Log' button and three dropdown menus: 'Target: default-log', 'Filter: (none)', and another '(none)'. The current time is displayed as '16:34:29 on 2009-03-17'. A table with columns 'time', 'category', 'level', 'tid', 'dir', 'client', 'msgid', and 'message' is shown. The table contains six log entries for 'b2bgw (Partner)' on 'Tue Mar 17 2009'. The messages include 'Archive operation completed', 'Disk space used by persistent store is 1928 KB.', 'Calculate space used by persistent store.', 'Archive operation completed', 'Archive operation is done, freed 0 KB.', and 'Found 0 processed documents with total size of 0 KB. Required 1 documents or total size of 0 KB.' A link 'Show last 50 100 all' is visible in the top right of the table area.

| time | category | level | tid | dir | client | msgid | message |
|-----------------|----------|-------|-----|-----|--------|------------|---|
| Tue Mar 17 2009 | | | | | | | |
| 16:20:49 | b2bgw | info | 513 | | | 0x82c0004f | b2bgw (Partner): Archive operation completed |
| 16:20:49 | b2bgw | info | 513 | | | 0x82c00049 | b2bgw (Partner): Disk space used by persistent store is 1928 KB. |
| 16:20:49 | b2bgw | info | 513 | | | 0x82c00048 | b2bgw (Partner): Calculate space used by persistent store. |
| 16:20:49 | b2bgw | info | 513 | | | 0x82c0004f | b2bgw (Partner): Archive operation completed |
| 16:20:49 | b2bgw | info | 513 | | | 0x82c0002c | b2bgw (Partner): Archive operation is done, freed 0 KB. |
| 16:20:49 | b2bgw | debug | 513 | | | 0x82c00029 | b2bgw (Partner): Found 0 processed documents with total size of 0 KB. Required 1 documents or total size of 0 KB. |

Figure 9-5 System Log

By default, the appliance log level is error, which provides minimal information if and only if errors occur. In order to get more relevant troubleshooting information, change the log level to debug when the goal is to troubleshoot.

Note: We do not recommend that you set the level to information or to debug all the time, because it can impact performance. In fact, setting either level causes the following message to be displayed on all WebGUI windows:

“Debug-Level Logging is enabled, which impacts performance.”

To change the logging level for the appliance-wide log or the domain-specific log, use the following procedure:

1. From the control panel, click the Troubleshooting icon.
2. Navigate to the Logging section.
3. Select the **debug** logging level from the Log Level list.
4. Click **Set Log Level**.

Note that the log must be viewed from the bottom to the top, because the most recent chronological entries are at the top. Each new sequence has a similar time stamp. You can view the sequence of events from start to finish and note each error or warning message as another clue to the problem that you are attempting to solve.

9.3.4 Checking the audit log

The audit log in the default domain records changes depending on the configuration of the appliance. To view the audit log, select **STATUS** → **View Logs** → **System Logs**. Figure 9-6 depicts a typical audit log window.

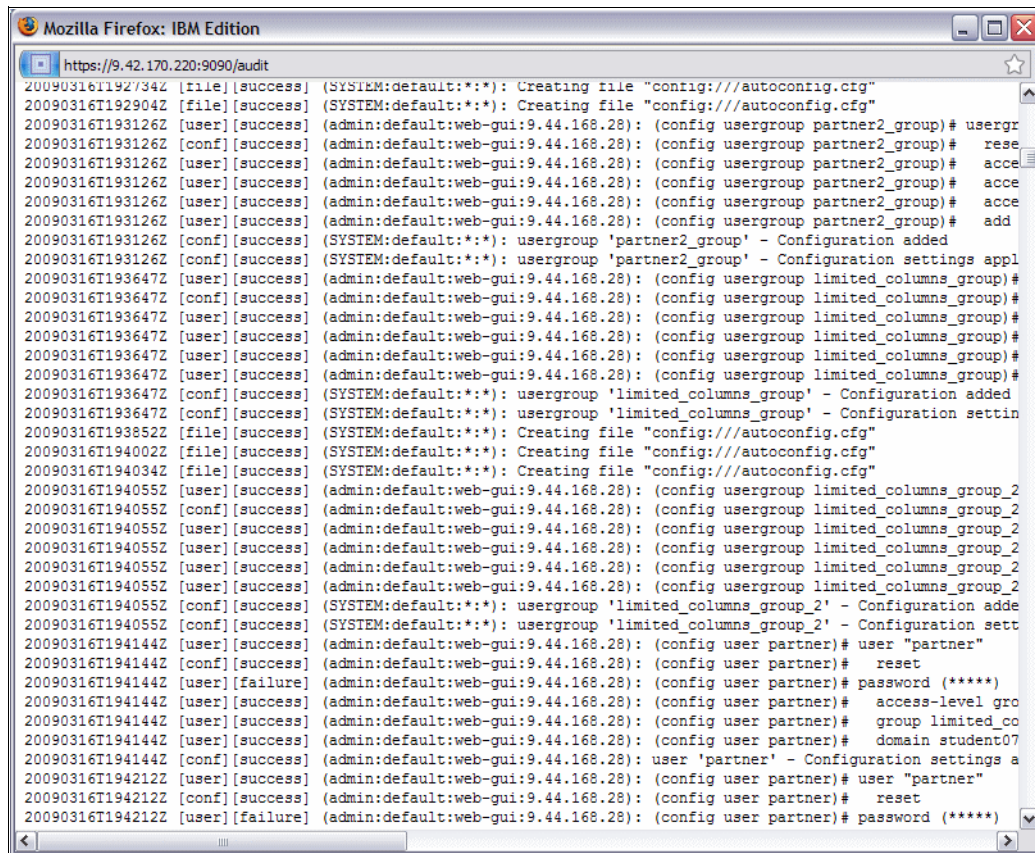


Figure 9-6 Audit log

The audit log contains entries about changes to the configuration of the appliance and files that are stored on the appliance. The length of the audit log is

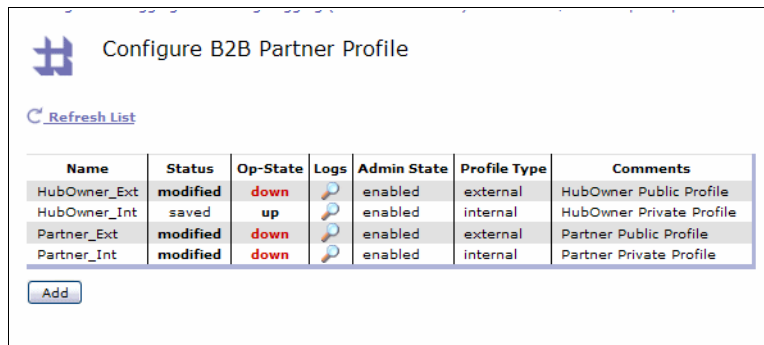
restricted to approximately 256 KB with one rotation. The audit log (audit-log) and its rotation (audit-log-1) are stored in the audit: directory.

You cannot access the audit: directory with the File Management utility. You cannot modify the entries that are written to the audit log.

9.3.5 Checking the Object Status

The Object Status indicates the operational state of objects in the appliance. Check the status of all the objects. An object in the down operational state can disable the overarching service.

In the B2B Gateway, the Partner Profile object is a good example of this overarching behavior. Select **Objects** → **B2B Configuration** → **B2B Partner Profile**. Figure 9-7 shows Partner Profile objects in the **up** and **down** states.



| Name | Status | Op-State | Logs | Admin State | Profile Type | Comments |
|--------------|----------|----------|------|-------------|--------------|--------------------------|
| HubOwner_Ext | modified | down | | enabled | external | HubOwner Public Profile |
| HubOwner_Int | saved | up | | enabled | internal | HubOwner Private Profile |
| Partner_Ext | modified | down | | enabled | external | Partner Public Profile |
| Partner_Int | modified | down | | enabled | internal | Partner Private Profile |

Figure 9-7 Partner Profile object status

9.3.6 Generating an error report

An error report file contains the current configuration, the current contents of the appliance log, and the current content of the command line interface (CLI) log. The Administrator can generate a verbose report at any time by using the Generate Error Report option. During the testing phase, generating an error report is an excellent first step for locating problems. The error report:

- ▶ Is created in the temporary directory and contains the current configuration, the current contents of the system log, and the contents of the CLI log.
- ▶ Can be sent to an e-mail address
- ▶ Is required when engaging with IBM DataPower support

To create the report, click **Generate Error Report**, as shown in Figure 9-8.

Reporting

Generate Error Report
 No Error Report Available for Viewing
Include Internal State on off

Send Error Report
Location
SMTP Server
Email Address

Figure 9-8 Generate Error Report

A dialog window asks for confirmation and indicates the location of the resulting file, as shown in Figure 9-9.

DATAPOWER

Generate error-report in temporary:///error-report.txt?

Troubleshooting Enabled (The performance of the device may be impacted!)

Action completed successfully!
 Successfully generated error-report in temporary:error-report.txt.

Figure 9-9 Error report confirmation

If an error report is available, you can view it by clicking **View Error Report**, which is shown in Figure 9-10 on page 160. The error report file opens. (You also can view the file from the File Management panel.)



Figure 9-10 View Error Report

You can send the error report to a designated e-mail account, as shown in Figure 9-11.

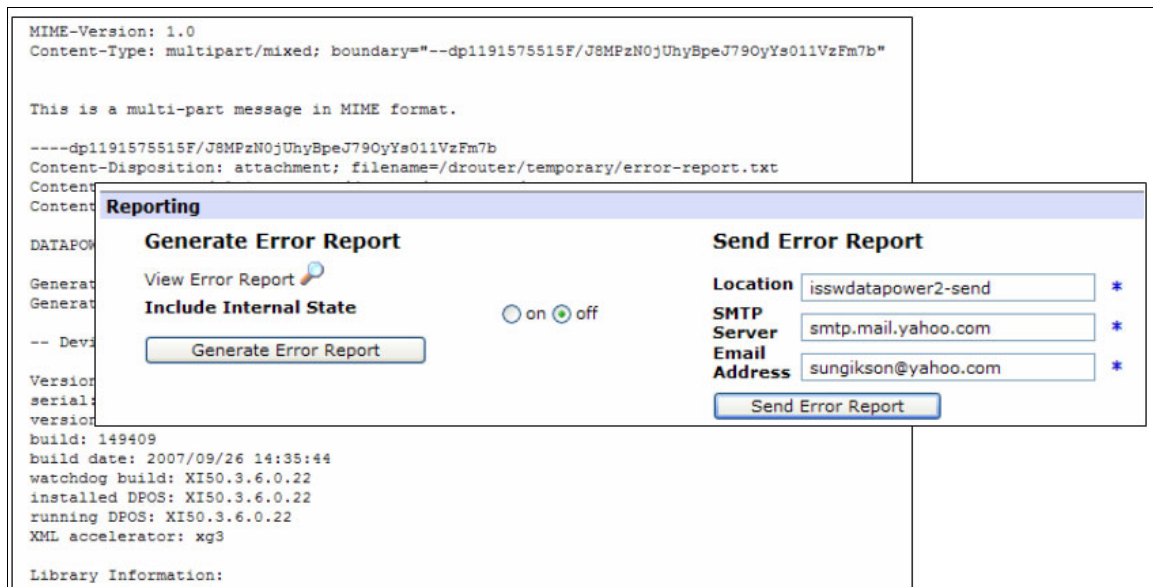


Figure 9-11 Send Error Report

9.4 XB60 firmware level 3.7.3 limitations and known problems

All of the inbound message flows from external partners into a B2B Gateway Service must be packaged in an AS2 or AS3 messaging envelope. This limitation can be overcome by using a Multi-Protocol Gateway in conjunction with a B2B Gateway.

All of the outbound message flows from internal partners into a B2B Gateway Service must not be packaged in an AS2 or AS3 messaging envelope. This limitation can be overcome by using a Multi-Protocol Gateway in conjunction with a B2B Gateway.

9.5 Common B2B XB60 configuration mistakes

Next, we describe several common configuration mistakes.

9.5.1 The hard disk array is unresponsive or down

Symptoms:

- ▶ The RAID disk drive is unresponsive for several hours.
- ▶ Error messages result when you create the B2B Gateway Service.

Solving the problem:

- ▶ Check to make sure that the hard drives are synchronized completely as directed in 6.2.4, “Checking and managing storage” on page 83.
- ▶ Make sure that the hard drive directory name is *not* the same name as any application domain name on the appliance.
- ▶ Reconfigure the hard drive storage as described in 6.2.1, “Startup method” on page 78 and verify the configuration as described in 6.2.3, “Verifying the configuration” on page 83.

9.5.2 B2B Gateway is unresponsive (down)

Symptoms:

- ▶ The B2B Gateway Service reverts to the *down* state if:
 - Any object associated with the B2B Gateway Service is in the operational state of *down*.

- Any crypto identification credentials are *down*, and the associated partner profile and the associated gateway revert to the *down* state.
- B2B persistence is not initialized properly.

To solve the problem, correct the problem in the underlying object and save the domain configuration. Restart the domain if necessary.

9.5.3 B2B Transaction Viewer not visible to partners

The symptom is the B2B Transaction Viewer is inaccessible.

To solve the problem:

- ▶ The B2B Transaction Viewer is not accessible by default.
- ▶ Check the user's group access profile and each underlying access policy.
- ▶ Refer to 7.1.3, "B2B Transaction Viewer" on page 106 for details.

9.5.4 B2B Gateway not sending MDNs as expected

The B2B Gateway is not sending Message Disposition Notifications (MDNs) as expected. Symptoms include, when requesting asynchronous MDNs, the MDN is not sent correctly, because:

- ▶ The redirection URL is not specified correctly in the B2B Gateway Service.
- ▶ The redirection URL is not specified correctly in the partner profile.

Solving the problem:

- ▶ We recommend that you always specify the redirection URL in the B2B Gateway. In the case where the URL is specified in both places, the value in the partner profile overrides the B2B Gateway.
- ▶ Make sure that the redirection URL that is specified is external to the gateway.

9.5.5 Binary documents are not routed properly

The problem is that binary documents are not routed properly.

Symptoms

When working to code the Document Routing Preprocessor stylesheet, the following error messages might appear in the appliance log (assuming that the log level is set to debug):

- ▶ “0x80000001 b2bgw: x12 sender-id, "111" does not match partner-from, "333" specified by headers.”
- ▶ “0x80000001 b2bgw: error extracting EDI partner info from message.”
- ▶ “0x03130001 b2bgw:Invalid business ID and 0x82c00001 b2bgw (): Invalid business ID.”

Solving the problem

The Document Routing Preprocessor stylesheet, which is available on the Advanced tab when configuring a B2B Gateway, can be used to set partner IDs for outgoing B2B transactions processing binary files.

The stylesheet sets partner information for the transaction by setting two DataPower service context variables as shown in Example 9-1.

Example 9-1 Stylesheet sets two DataPower service context variables

```
<dp:set-variable name="'var://service/b2b-partner-from' "  
value="'111'"/>  
<dp:set-variable name="'var://service/b2b-partner-to' "  
value="'222'"/>
```

By default, the IBM WebSphere DataPower B2B Appliance XB60 is configured to route electronic data interchange (EDI) documents between systems inside an enterprise network and a similar B2B Gateway at another company. The XB60 natively understands ANSI X12 and EDIFACT documents and has special handling for XML-formatted EDI messages. All other messages are handled as binary messages.

Possible causes

The stylesheet might not be setting the service context variables properly or the partner IDs might not be properly configured. Also, be aware that there are restrictions for inbound and outbound messages, which might cause problems.

Version 3.7.3 restriction: All of the inbound and outbound message flows through a B2B Gateway Service are assumed to be contained in B2B messaging protocols, such as AS2 or AS3 headers.

Inbound messages

The XB60 receives “inbound” messages from an external trading partner and routes them to an internal system. These messages are always received using AS2 or AS3 wire protocols. In these protocols:

- ▶ The AS2-From: or AS3-From: header identifies the trading partner that sends the message.
- ▶ The AS2-To: or AS3-To: header identifies the trading partner to receive the message.

These protocols also use the MIME Content-Type: header to identify the type of message.

Inbound binary documents are routed in the same manner as other inbound documents. The B2B Gateway finds the corresponding partner profile that contains the ID of the trading partner from the AS2-To: or AS3-To: header, finds the first destination on the partner profile that lists “binary” as an Enabled Document Type, and forwards the document to the URL for that destination.

Outbound messages

The XB60 receives “outbound” messages from an internal system and routes them to an external trading partner. The XB60 often receives these messages via a protocol, such as WebSphere MQ, that does not have the same routing headers as AS2 or AS3. For X12 and EDIFACT messages, the XB60 can parse the message contents and find the partner IDs from the ISA and UNA headers. The XPath Routing Policies tab on the B2B Gateway configuration tells the XB60 how to find the partner IDs for XML documents. For binary messages, an appliance administrator must configure an XSLT stylesheet as the Document Routing Preprocessor for the gateway to route the message properly.

The routing stylesheet is an XSLT 1.0 stylesheet with standard DataPower extensions available run over an empty tree as input. The default stylesheet is `store:///b2b-routing.xsl`. For gateway implementations, an administrator will make a copy of this stylesheet either in the appliance per-domain `local:` directory or on an external Web server, edit it to the needs of a particular gateway, and update the B2B Gateway configuration to point at the modified stylesheet. Each B2B Gateway can be configured with a different routing stylesheet.

Diagnosing the problem

In order to diagnose the problem, add the `<xsl:message>` element to your stylesheet to include debug information in the DataPower system logs. In Example 9-2, this `<xsl:message>` writes a simple message to the log indicating that the stylesheet is reaching the element that sets the service context variables.

Example 9-2 Add the `<xsl:message>` element to your stylesheet

```
<xsl:message>Setting b2b-partner-to to 222</xsl:message>
<dp:set-variable name="'var://service/b2b-partner-to'" value="'222'"/>
```

Use the Troubleshooting icon from the DataPower Control panel to set the log level to debug to see the `<xsl:message>` output written by your stylesheet.

The information that the routing stylesheet needs is available through a set of service variables via the DataPower `dp:variable()` XSLT extension. The `store:///b2b-routing.xsl` stylesheet contains several examples in comments about how to use this extension function. For a full list of service variables, refer to the *Multi-Protocol Gateway Developers Guide*, which can be downloaded from the IBM WebSphere DataPower Product Documentation Portal Web page and navigating to the XB60 documentation:

<http://www-01.ibm.com/support/docview.wss?rs=2362&uid=swg21377654>

Next, we list several common error messages that you might encounter during the development of the document routing preprocessor stylesheet with their accompanying remediation suggestions:

- ▶ “0x80000001 b2bgw: x12 sender-id, "x" does not match partner-from, "y" specified by headers.”

The gateway is treating the request associated with the transaction as an X12 EDI document. Include this line in the stylesheet to have the request processed as a binary file:

```
<dp:set-variable name="'var://service/b2b-doc-type'"
value="'binary'"/>
```

- ▶ “0x80000001 b2bgw: error extracting EDI partner info from message.”

The stylesheet did not properly set the value of the `var://service/b2b-partner-from` variables. Make sure that the literal values for the service variables are properly enclosed in quotes:

```
<dp:set-variable name="'var://service/b2b-partner-from'"
value="'1010101010'"/>
```

Use the previous `<xsl:message>` technique to display the value that you are setting in the log.

- ▶ “0x03130001 b2bgw: Invalid business ID and 0x82c00001 b2bgw (): Invalid business ID.”

The stylesheet might set partner information for outgoing transactions, so the following must be true:

- The value of `var://service/b2b-partner-from` must be a partner ID for an Internal partner profile.
- The value of `var://service/b2b-partner-to` must be a partner ID for an External partner profile.

Choose the B2B Partner Profile icon from the DataPower Control Panel and verify that your partner configurations are configured for an outgoing transaction.

9.6 Life cycle considerations

As a general rule, the best troubleshooting tools to use first when a problem occurs often depend on how the appliance is being used at the time.

During the development phase:

1. Check the appliance log for problems. If checking the appliance log does not locate the problem:
 - a. Set the appliance log to the debug level.
 - b. Run the problematic scenario again.
2. Check the appliance log for problems. If checking the appliance log does not locate the problem:
 - a. Check the B2B Transaction Viewer.
 - b. Run the problematic scenario again.
 - c. Check the B2B Transaction Viewer for problems.

During the test phase:

1. Check the appliance log for problems. If checking the appliance log does not locate the problem:
 - a. Generate an error report. If generating an error report does not locate the problem:
 - i. Enable the multistep probe.
 - ii. Run the problematic scenario again.
 - iii. Check the multistep probe for problems.

During the production phase:

1. Check the appliance Usage object for load and work list.
2. Check the Object Status object for objects that have changed to the *down* operational state.
3. Check the appliance log.

9.7 Getting help and technical assistance

If you encounter a problem, you want to resolve it quickly. You can search the available knowledge bases to determine whether your problem has already been encountered and if a resolution is already documented.

Product Documentation

The IBM WebSphere DataPower documentation library provides extensive documentation in Portable Document Format (PDF):

<http://www.ibm.com/support/docview.wss?rs=2362&uid=swg24021688>

You can use the search function of Adobe® Acrobat® to query information. If you download and store the documents in a single location, you can use the search facility to find all references to a subject across the documentation set.

IBM Support

If you cannot find an answer in the documentation, use **Search Support** (*this product*) at:

<http://www.ibm.com/software/integration/datapower/support/>

From the Search Support (*this product*) area, you can search the following IBM resources:

- ▶ IBM TechNote database
- ▶ IBM downloads
- ▶ IBM Redbooks publications
- ▶ IBM developerWorks®

If you cannot resolve the problem or answer the question from these methods, contact IBM Support.



Part 3

B2B patterns and service-oriented architecture (SOA) integration

Part 3 of this book provides you with five common business-to-business (B2B) scenarios and demonstrates how each scenario was completed using the XB60:

- ▶ Chapter 10 demonstrates how to use the XB60 in front of WebSphere Transformation Extender for processing Health Insurance Portability and Accountability Act (HIPAA) transactions.
- ▶ Chapter 11 demonstrates how to use the XB60 to receive documents from a trading partner and then transforming the document on the XB60.
- ▶ Chapter 12 and 13 provide examples of how you can trade binary documents (no partner information in the document) using the XB60.

- ▶ Chapter 14 demonstrates how you can use Web Services to transport B2B documents and then use protocol bridging to route the document to a WebSphere MQ queue.



XB60 and WTX integration for HIPAA

This chapter shows an example scenario using the WebSphere DataPower B2B Appliance XB60 in conjunction with WebSphere Transformation Extender (WTX) to provide a B2B solution in the health care industry.

10.1 Business value

A certain exchange of information occurs during each communication between a patient and a health care provider. In addition to the basics of patient demographics, symptoms, diagnoses, and treatments, a common Health Insurance Portability and Accountability Act (HIPAA) trading scenario requires the exchange of claims and payment data as well as associated payer, subscriber, eligibility, and authorization information. A single encounter can involve the transmission of large volumes of information among several participants.

The exchange of health care information can generally be viewed as a transaction between the sender and receiver participants.

Health care transactions include (but are not limited to):

- ▶ Health care claim or encounter
- ▶ Claim payment and remittance advice
- ▶ Health care claim status
- ▶ Coordination of benefits
- ▶ Eligibility for a health plan
- ▶ Referral certification and authorization
- ▶ Enrollment and unenrollment in a health plan
- ▶ Premium payments

Figure 10-1 on page 173 shows an example of a transaction that includes a health care claim and the claim payment between two partners.

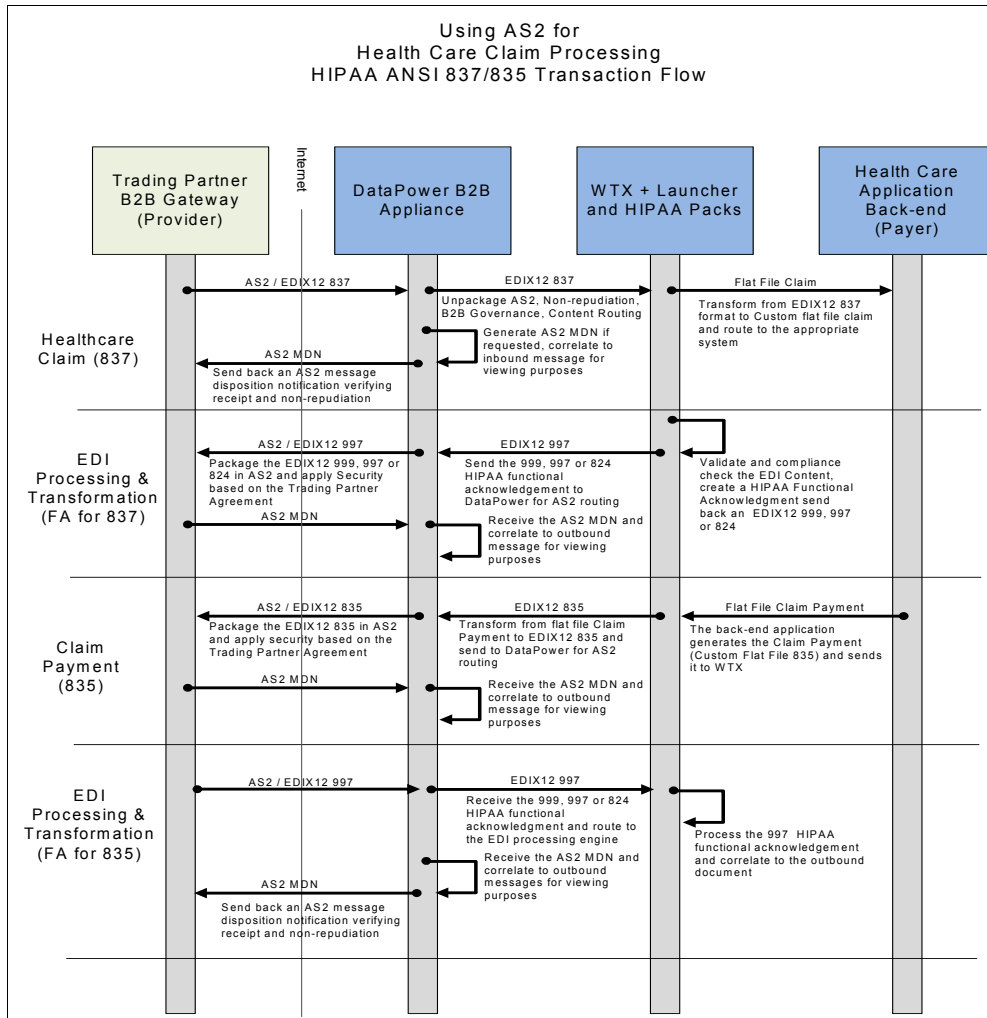


Figure 10-1 Health care claim processing

These transactions can be transmitted electronically in compliance with health care transaction standards. Health care data exchange standards allow the accurate and timely exchange of information between health care organizations. For example, a simple benefits inquiry can take 20 minutes on the phone. Using Electronic Data Interchange (EDI), this type of request can be processed almost immediately, without the need for a call to the insurer's customer service center.

WebSphere Transformation Extender and its HIPAA EDI Pack provide predefined functionality that supports tactical HIPAA compliance, reduces the risks associated with complex compliance, provides standard-checking, cuts

programming time and cost, and minimizes the need for back-end system reengineering.

WebSphere DataPower B2B Appliance XB60 provides, at the network edge, B2B governance and high performance for full end-to-end processing with validation and functional acknowledgments, together with B2B Governance by enforcing Trading Partner Agreement policy and AS2/AS3 Standards compliance.

WTX with Launcher and WebSphere DataPower B2B Appliance XB60 provide a full end-to-end solution, covering all the requirements for a robust, scalable, and maintainable B2B infrastructure for partner trading. These requirements include:

- ▶ The B2B document format must be HIPAA EDI X12 Version 5010.
- ▶ All X12 data must be validated by the payer, and a functional acknowledgement must be returned with the validation status.
- ▶ Data sent over the Internet must be signed and encrypted.
- ▶ Data sent over the Internet must have a mechanism of verifying delivery of the message before closing the connection (sync acknowledgement).
- ▶ The ability to view the status of transactions in the gateway must exist.
- ▶ The ability to resend failed transactions from the gateway must exist.

Secure Sockets Layer (SSL) is not included in this scenario in the interest of brevity. Refer to the binary scenario in Chapter 14, “Handling SOAP Messages with Attachments in a B2B environment” on page 363 for an example with SSL.

10.2 Prerequisites: Technical and infrastructure

There are prerequisites both in order to fully understand the scenario and to be able to set it up successfully in your own infrastructure.

10.2.1 Software prerequisites

In order to be able to run this scenario, you must have installed the following components:

- ▶ WebSphere DataPower B2B Appliances XB60
- ▶ WebSphere Transformation Extender V8.2 with Launcher
- ▶ WebSphere Transformation Extender Pack for HIPAA EDI
- ▶ WebSphere MQ

10.2.2 Skills prerequisites

In order to be able to fully implement and understand this scenario, you must be familiar with:

- ▶ WebSphere DataPower B2B Appliance XB60 main concepts (having completed at least the XB60 AS2 Trading Tutorial that is provided as additional material to this book in Appendix A, “Additional material” on page 389.)
- ▶ WebSphere Transformation Extender basic mapping techniques and creating a simple system using Integration Flow designer

10.3 Presenting the scenario

In this section, we provide an overview of our health care scenario. For this particular scenario, we implement a two-way flow that corresponds with an incoming health care claim (EDI 837 format) to the health provider system and its corresponding claim payment (EDI 835 format).

Note: It is important to note that the Functional Acknowledgement (EDIX12 824, 997, or 999) is not used in this example scenario in order to keep the example flow simple and easy to follow.

As you can see in Figure 10-2 on page 176, the scenario can be divided into two logical subsets:

- ▶ The inbound scenario of receiving the Health Care Claim by the trading partner.
- ▶ The outbound flow to send the corresponding claim payment back to the trading partner. All the messages between the partner and the provider are exchanged, encrypted, and signed as a consequence of the assumed trading manager agreement.

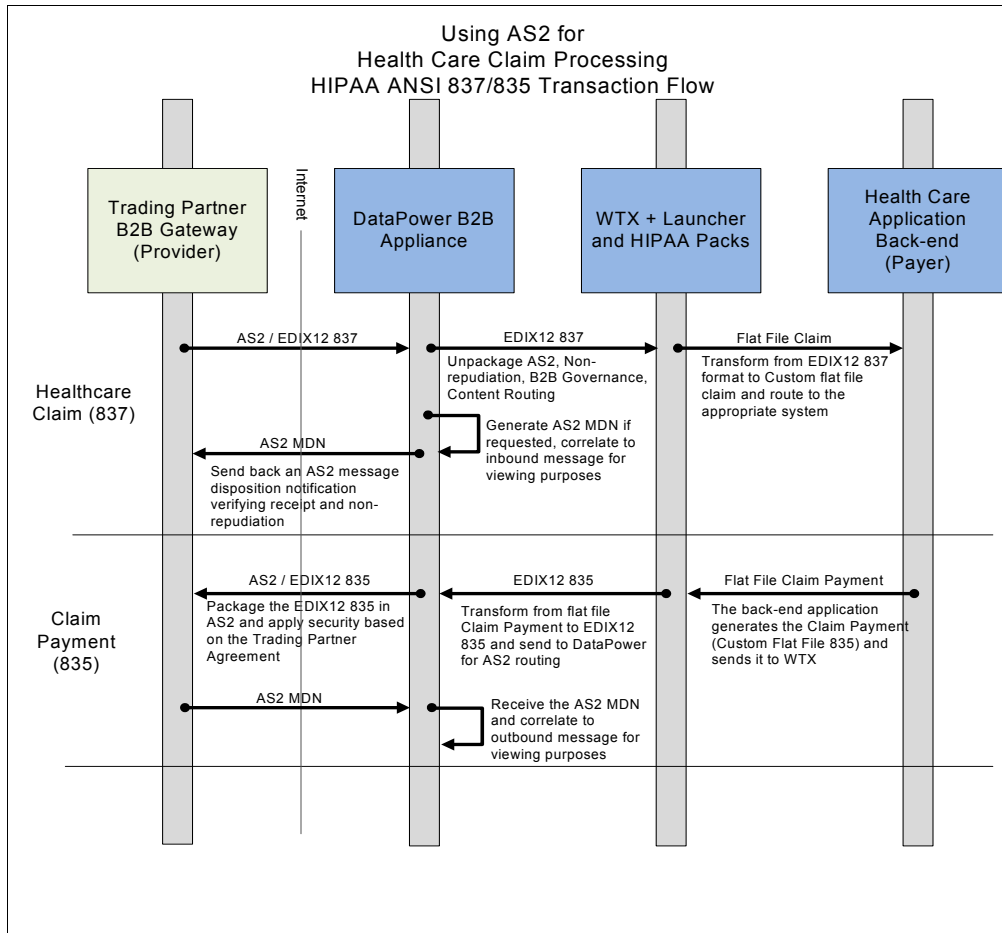


Figure 10-2 HIPAA Health Care Claim Processing Transaction Flow

From a naming perspective, we refer to our trading hub (reflected in Figure 10-2 as the Payer) as the HIPAA Provider and to our partner as the HIPAA Partner (reflected in Figure 10-2 as Provider).

10.3.1 The health care claim: Inbound flow

In the case of an incoming health care claim, the HIPAA Provider system accepts a valid EDI X12 837 message (wrapped, signed, and encrypted in an AS2 message) that comes from the Internet to the demilitarized zone (DMZ), where the DataPower device is located. The device is responsible for decrypting the message and validating its signature, identifying the type of message that arrives (EDI, X12, XML, and so forth) and, based on the partner's information, deciding

whether the partner is allowed to trade with that kind of document. If that is the case, the device routes the document to whichever destination is set for this specific partner (in our case, it is the WTX Launcher, which performs the format transformation). DataPower also manages sending a signed and encrypted Message Disposition Notification (MDN), which is positive or negative depending on the results of the actions just explained.

After the document is delivered to the WTX Launcher, a transformation map is triggered. This transformation map is responsible for transforming the incoming X12 837 format into a custom flat file claim that can be consumed by a potential customer's back end when the map is finished.

10.3.2 The claim payment: Outbound flow

For the claim payment, the WTX Launcher system is triggered by a new flat file message arriving to the system, which is transformed into a compliant X12 835 message. This message is sent to the appliance, and the appliance packages the message in a standard signed and encrypted AS2 message to be routed to the partner system, based on the partner information that is configured in the device.

After the message is sent to the trading partner, the device waits for the corresponding MDN, which is decrypted and signature-validated before considering that transaction complete.

10.4 Scenario solution

After describing at a high level the key aspects of the scenario that we implement, we explain in technical detail all of the steps necessary in order to fully implement the solution. One of the key aspects of the scenario implementation is that we simulate the trading partner within the DataPower device, which means that we create a specific B2B Gateway Service for the trading partner. We present a complete picture of all the services and objects to implement in the scenario outline, where we also present a summary of what is created.

10.4.1 Scenario outline

Figure 10-3 on page 178 shows the architecture of the scenario.

This architecture demonstrates that the DataPower device can provide the gateway services for both the partner and the provider. The left side of

Figure 10-3 corresponds to the simulated trading partner, which is referred to as the HIPAA Partner, and the right side corresponds to the hub system, which is referred to as the HIPAA Provider.

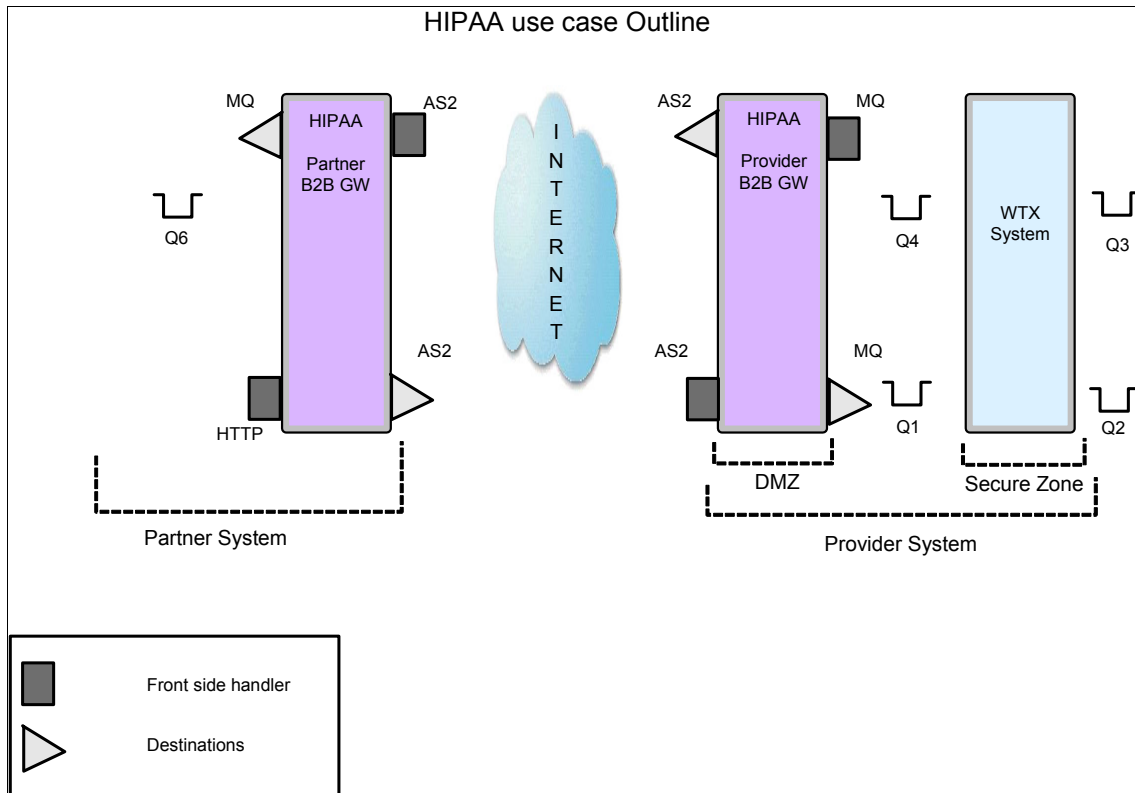


Figure 10-3 HIPAA scenario outline

As you can see in Figure 10-3, we use WebSphere MQ (WMQ) queues to trigger WTX Launcher to transform incoming messages from both sides (Partner and back end) as well as a potential partner back end. The rest of the interactions are HTTP-based, and more specifically, AS2 over HTTP when going to the Internet.

Here is a summary of the steps that were used to implement the scenario:

- ▶ Step 1: Creating all the necessary crypto objects
- ▶ Step 2: Creating the HIPAA Partner partner profiles
- ▶ Step 3: Creating the HIPAA Provider partner profiles
- ▶ Step 4: Creating HIPAA Partner B2B Gateway Service
- ▶ Step 5: Creating HIPAA Provider B2B Gateway Service
- ▶ Step 6: Creating the inbound mapping system
- ▶ Step 7: Creating the outbound mapping system

10.4.2 Scenario implementation

Now that we have the details of everything that needs to be configured, it is time to take a deeper look into each of the steps.

Step 1: Creating all the necessary crypto objects

This scenario is intended for people who are experienced with the DataPower device, so detail about how to create the crypto objects is not included. However, it is important to point out that several objects are required for signing and validating signatures and encrypting and decrypting payloads.

We created the following objects specifically for this scenario:

- ▶ Public/Private keys:
 - Health_partner keys pair
 - Hipaa_provider keys pair
- ▶ Validation credentials:
 - HIPAAPartner_valcred: Used to validate signatures coming from the HIPAA partner
 - HIPAAProvider_valcred: Used to validate signatures coming from the HIPAA provider
- ▶ Identification credentials:
 - HIPAAPartner_IDcred: Used to sign messages from the HIPAA partner and to decrypt payloads coming from the HIPAA provider.
 - HIPAAProvider_IDcred: Used to sign messages from the HIPAA Provider and to decrypt payloads coming from the HIPAA provider.

Both ID Credentials and validation credential objects are configured with the corresponding key pair.

Step 2: Creating the HIPAA Partner partner profiles

A *partner profile* is an object that defines the routing for messages by defining the destinations to it, as well as establishing the AS Security rules when interchanging information with that specific partner.

Because we simulate our trading within the DataPower device, we need two HIPAA Partner profiles: an *internal profile* for the B2B Gateway Service that will represent the partner back end and an *external profile* to be included in the HIPAA Provider B2B Gateway Service and where we set the destination pointing to the other B2B Gateway Service.

HIPAA Partner internal profile

The HIPAA Partner internal profile contains the details that help manage the information about what kind of AS Security the partner expects, including how to sign the outgoing messages, decipher the incoming messages (AS Security tab), and where to route the messages (Destinations tab).

Here are the aspects of the configuration that you need to consider in order to successfully configure this profile, which is named it HIPAA_Partner_int. We follow this methodology for the next objects as well.

First of all, we need to add Business IDs, so that the B2B Gateway (B2BGW) Service can associate the incoming business ID in the AS2 headers to this partner profile and route the message to its destination.

For this particular case, we have two Business IDs: zzhipaapartner comes in the X12 sent from our back end and 01hipaapartner comes from the partner to our system.

Figure 10-4 shows the configuration.

The screenshot shows the 'Configure B2B Partner Profile' web interface. The title bar includes a logo and the text 'Configure B2B Partner Profile'. Below the title bar is a navigation menu with tabs: 'Main' (selected), 'AS Security', 'Destinations', and 'Contacts'. The main content area displays 'B2B Partner Profile: HIPAA_Partner_int [up]'. There are four buttons: 'Apply', 'Cancel', 'Delete', and 'Undo'. On the right side, there are links for 'Export' and 'View Log'. The configuration fields are as follows:

- Admin State:** Radio buttons for 'enabled' (selected) and 'disabled'.
- Comments:** An empty text input field.
- Profile Type:** Radio buttons for 'External' and 'Internal' (selected).
- Partner Business IDs:** A table with two rows: '01hipaapartner' and 'zzhipaapartner'. Each row has up/down arrows and a delete icon. Below the table is an empty text input field and an 'Add' button.

A small asterisk (*) is located below the 'Add' button.

Figure 10-4 HIPAA Partner internal profile Main configuration tab

The next step is to properly configure AS Security. We use the crypto objects that we have created in “Step 1: Creating all the necessary crypto objects” on page 179. Refer to Figure 10-5.

The screenshot displays the 'Configure B2B Partner Profile' window with the 'AS Security' tab selected. The profile name is 'HIPAA_Partner_int'. The 'Inbound Security' section has 'Require Signature' and 'Require Encryption' checked, and 'Inbound Decryption Identification Credentials' set to 'HIPAA_Partner_IDcred'. The 'Outbound Security' section has 'Sign Outbound Messages' checked, 'Signing Identification Credentials' set to 'HIPAA_Partner_IDcred', and 'Signing Digest Algorithm' set to 'sha1'. Action buttons 'Apply', 'Cancel', 'Delete', and 'Undo' are at the top left, and 'Export' and 'View Log' are at the top right.

Figure 10-5 HIPAA Partner internal profile AS Security configuration tab

As you can see in Figure 10-5, we require Signature and Encryption on the Inbound Security, which means that we only accept signed and encrypted messages. In order to decrypt those messages, we need our HIPAA Partner ID Credentials, which contain the private key.

From an Outbound Security perspective, we sign the AS2 messages, so we need to check that option and use the HIPAA _Partner ID credentials to sign messages with the HIPAA Partner private key.

Important: Notice that we use the same ID Credential objects, because in order to simplify the scenario, we assume that the HIPAA Partner uses the same certificate and key pair for all the crypto tasks.

The next step is to configure destinations. As we can see in Figure 10-6, the HIPAA Internal Partner sends its messages via WebSphere MQ (WMQ) to the MQ Queue that simulates a back end in the Partner system.

This MQ destination, which is named MQ_2_backend, will only accept X12 messages to be routed to the back end.

Figure 10-6 shows the configuration. Do not forget to click **Apply** to save your destination.

The screenshot shows the 'Configure B2B Partner Profile' interface with the 'Destinations' tab selected. The profile is 'HIPAA_Partner_int [up]'. A table lists the destination 'MQ_2_backend (default)' with URL 'dpmq://QM1/?ReplyQueue=Q6' and 'Enabled Document Type' 'X12'. Below the table, a form allows editing the destination name to 'MQ_2_backend' and selecting 'X12' as the enabled document type. The connection URL is 'dpmq://QM1/?ReplyQueue=Q6' and the timeout is '300' seconds. The 'Apply' button is circled in red.

| Destination Name | Destination URL | Enabled Document Type |
|------------------------|---------------------------|-----------------------|
| MQ_2_backend (default) | dpmq://QM1/?ReplyQueue=Q6 | X12 |

Destinations

Destination Name: *

Enabled Document Type:

- XML
- X12
- EDIFACT
- Binary

Connection

Destination URL: *

Connection Timeout: Seconds

Figure 10-6 HIPAA Partner internal profile Destinations tab details

HIPAA Partner external profile

The HIPAA Partner external profile contains the details that manage the information that is needed by the provider in order to successfully trade with the HIPAA Partner (AS Security tab) and the Partner destination point (Destinations tab).

Here, we discuss the aspects of the configuration in order to successfully configure this profile. We have named the profile HIPAA_Partner. In our naming convention, external partners do not have any special indication of that on the name, but they can choose their own naming convention.

From a Business ID perspective, we have the same configuration that we had on the internal profile. Figure 10-7 shows the configuration.

Configure B2B Partner Profile

Main AS Security Destinations Contacts

B2B Partner Profile: HIPAA_Partner [up]

Apply Cancel Delete Undo

Admin State enabled disabled

Comments

Profile Type External Internal

| Partner Business IDs | | | |
|----------------------|---|---|-----|
| 01hipaapartner | ↑ | ↓ | ✕ |
| zzhipaapartner | ↑ | ↓ | ✕ |
| <input type="text"/> | | | Add |

*

Figure 10-7 HIPAA Provider external profile Main configuration tab

From an AS Security standpoint, we need to enter all the information to validate incoming signatures from the HIPAA Partner, because this partner profile will attach to the Provider B2B Gateway Service. Validation Credentials need to be configured to verify that the signature coming from the HIPAA Partner has been created with the HIPAA partner private key. Refer to Figure 10-8 on page 184.



Figure 10-8 HIPAA Partner internal profile AS Security configuration tab

In the Destinations tab, we now set up the URL where our HIPAA partner waits for HIPAA Provider AS2 messages. Therefore, we configure an AS2 destination. Figure 10-9 on page 185 shows the upper part of the configuration page, where we set up the destination URL and the document types that are enabled (we will only use AS2).

Configure B2B Partner Profile

Main AS_Security Destinations Contacts

B2B Partner Profile: HIPAA_Partner [up]

Apply Cancel Delete Undo Export View Log View

| Destination Name | Destination URL | Enabled Document Type |
|------------------------------|-----------------------|-----------------------|
| AS2_2_HIPAAPartner (default) | as2://127.0.0.1:10071 | X12 |

Destinations

Destination Name: AS2_2_HIPAAPartner *

Enabled Document Type:

- XML
- X12
- EDIFACT
- Binary

Connection

Destination URL: as2:// 127.0.0.1:10071 *

Connection Timeout: 300 Seconds

User name: []

Password: []

Figure 10-9 HIPAA Partner external profile Destination details

If we scroll down, we see the rest of the configuration options as depicted in Figure 10-10 on page 186. AS outbound security deals with how we want to send our AS2 messages. In our HIPAA Partner internal profile, we stated that it was a requirement to sign and encrypt all the incoming messages. We discuss signing in the HIPAA Provider internal profile (refer to step 3), because we use its private key. However from an encryption standpoint, we must configure the HIPAA Partner encryption certificate in Figure 10-10 on page 186. *Notice that this entry is only a certificate, not a validation credential object.*

In the Advanced AS Behavior section section of Figure 10-10 on page 186, we request an MDN when we send our messages to the HIPAA Partner, and we request that message retransmission is attempted up to three times, in case a problem occurs.

The screenshot shows a configuration window with two main sections:

- AS Outbound Security:**
 - Send Messages Unsigned:
 - Encrypt Messages:
 - Encryption Certificate: health_partner (dropdown menu) with '+' and '...' buttons and an asterisk.
- Advanced AS Behavior:**
 - Compress Messages:
 - Request MDN:
 - Time to Acknowledge: 1800 (input field) Seconds
 - Request Asynchronous MDN:
 - Request Signed MDN:
 - Attempt Message Retransmission:
 - Maximum Retransmissions: 3 (input field)

At the bottom, there are 'Apply' and 'Cancel' buttons, and a small asterisk icon below the window border.

Figure 10-10 HIPAA Partner external profile Destination details (continuation)

We have finished configuring all the required profiles for the HIPAA Partner.

Step 3: Creating the HIPAA Provider partner profiles

Similar to Step 2, we also need two HIPAA Provider profiles: an internal profile to associate with the HIPAA Provider B2B Gateway Service (refer to Step 6 for further details) and an external profile to associate with the HIPAA Partner B2B Gateway Service that will enable our partner to successfully send all the messages to the HIPAA Provider.

HIPAA Provider internal profile

The HIPAA Provider internal profile contains the details that manage the information about the type of AS Security that the provider expects (AS Security tab), including signing the outgoing messages, deciphering the incoming messages, and where the messages are routed (Destinations tab).

Here are the aspects of the configuration to consider in order to successfully configure this profile. We have named it HIPAA_Provider_int.

Business IDs are the first required aspect, so the B2BGW service can associate the incoming business ID in the AS2 headers to this partner profile and route a message to its destination.

We have two Business IDs: zzhpaaprovider comes in on the X12 sent from the back end and 01hipaaprovider comes from the partner to our system.

Figure 10-11 shows the configuration.

The screenshot shows the 'Configure B2B Partner Profile' web interface. The title bar includes a logo and the text 'Configure B2B Partner Profile'. Below the title bar is a navigation menu with tabs for 'Main', 'AS Security', 'Destinations', and 'Contacts'. The 'Main' tab is selected. The main content area displays 'B2B Partner Profile: HIPAA_provider_int [up]'. Below this are buttons for 'Apply', 'Cancel', 'Delete', and 'Undo', and an 'Export' link. The 'Admin State' section has radio buttons for 'enabled' (selected) and 'disabled'. The 'Comments' section has a text input field. The 'Profile Type' section has radio buttons for 'External' and 'Internal' (selected). The 'Partner Business IDs' section contains a table with two rows: '01hipaaprovider' and 'zzhipaaprovider'. Each row has up/down arrows and a delete icon. Below the table is an 'Add' button and an asterisk.

| Partner Business ID | Up Arrow | Down Arrow | Delete Icon |
|---------------------|----------|------------|-------------|
| 01hipaaprovider | ↑ | ↓ | ✕ |
| zzhipaaprovider | ↑ | ↓ | ✕ |

Figure 10-11 HIPAA Provider internal profile Main configuration tab

Figure 10-12 on page 188 shows all of the AS Security aspects. As in our Partner case, we also require Signature and Encryption on the Inbound Security. We will only accept signed and encrypted messages. In order to decrypt those messages, we need our HIPAA Partner ID credentials that contain the private key.

From an Outbound Security perspective, we sign the AS2 messages, so we need to check both options and use HIPAA _Provider ID credentials to sign messages with the HIPAA Provider private key.

Configure B2B Partner Profile

Main **AS Security** Destinations Contacts

B2B Partner Profile: HIPAA_provider_int [up]

Apply Cancel Delete Undo Export

Inbound Security

Require Signature

Require Encryption

Inbound Decryption Identification Credentials HIPAA_Provider_IDcred + ... *

Outbound Security

Sign Outbound Messages

Signing Identification Credentials HIPAA_Provider_IDcred + ... *

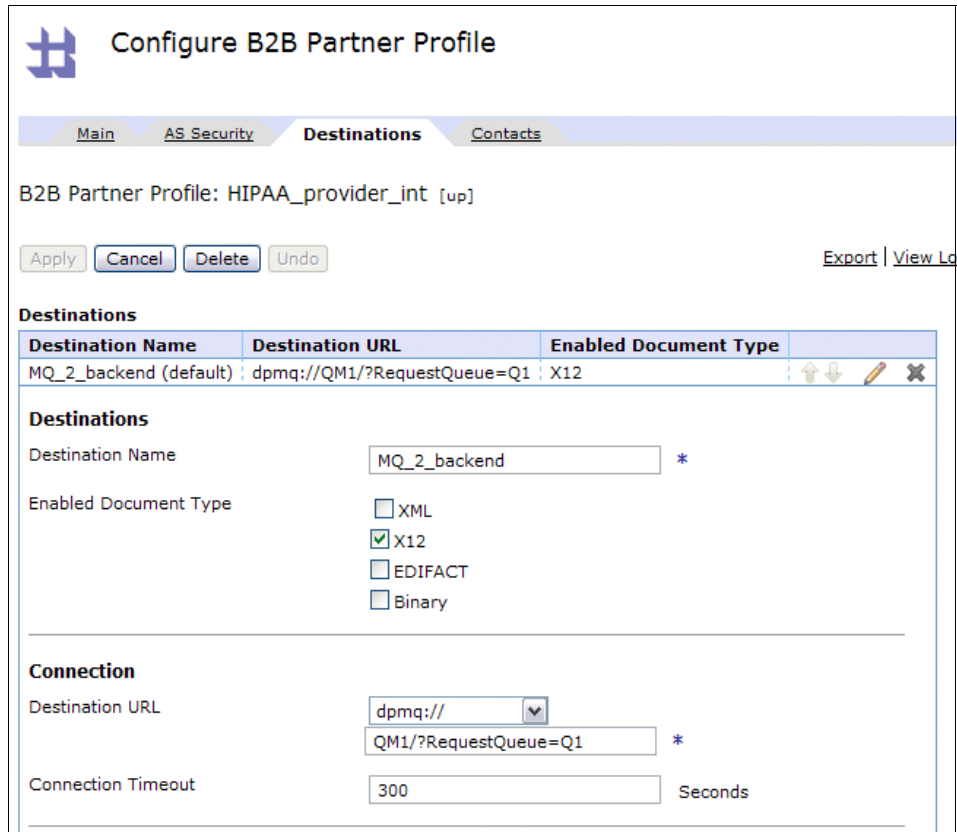
Signing Digest Algorithm sha1

Figure 10-12 HIPAA Provider internal profile AS Security configuration tab

The next step is to configure Destinations. Figure 10-13 on page 189 shows that the HIPAA Provider Internal Profile will send its messages via WebSphere MQ (WMQ) to the WTX system to trigger the message transformation.

Therefore, you must set an MQ destination. This MQ destination, which is named MQ_2_backend, only accepts X12 messages to route to the back end.

Figure 10-13 on page 189 shows the configuration. Do not forget to click **Apply** on the window to save your destination.



Configure B2B Partner Profile

Main AS Security **Destinations** Contacts

B2B Partner Profile: HIPAA_provider_int [up]

Apply Cancel Delete Undo Export | View Log

| Destination Name | Destination URL | Enabled Document Type |
|------------------------|-----------------------------|-----------------------|
| MQ_2_backend (default) | dpmq://QM1/?RequestQueue=Q1 | X12 |

Destinations

Destination Name: *

Enabled Document Type:

- XML
- X12
- EDIFACT
- Binary

Connection

Destination URL:

*

Connection Timeout: Seconds

Figure 10-13 HIPAA Provider internal profile Destinations tab details

Tip: If you have problems configuring the MQ URL, you can use the Multi-Protocol Gateway MQ URL Builder to create it for you, and then, you can copy and paste the result into the MQ URL field.

HIPAA Provider external profile

The HIPAA Provider external profile contains the details that manage the information needed by the partner to successfully trade with the HIPAA Provider (AS Security tab) and the Provider destination point (Destinations tab).

We have named the HIPAA Provider external profile HIPAA_Provider. In our naming convention, external partners are not indicated by their name, but you can choose a meaningful convention.

From a Business ID perspective, we have the same configuration that we had on the internal profile. Figure 10-14 shows the configuration.

The screenshot shows the 'Configure B2B Partner Profile' interface. At the top, there is a logo and the title 'Configure B2B Partner Profile'. Below the title is a navigation bar with tabs: 'Main' (selected), 'AS Security', 'Destinations', and 'Contacts'. The main content area displays the following information:

- B2B Partner Profile:** HIPAA_Provider [up]
- Buttons:** Apply, Cancel, Delete, Undo
- Admin State:** enabled disabled
- Comments:** [Empty text box]
- Profile Type:** External Internal
- Partner Business IDs:**
 - 01hipaaprovider [up/down/delete icons]
 - zzhipaaprovider [up/down/delete icons]
 - [Empty text box] [Add button]

A small asterisk (*) is located below the Partner Business IDs section.

Figure 10-14 HIPAA Provider external profile Main configuration tab

From an AS Security standpoint, we need to enter all the information to validate incoming signatures from the HIPAA Provider, because this partner profile is attached to the Partner B2B Gateway Service.

Only validation credentials need to be configured at this point. They will be used to verify that the signature coming from the HIPAA provider was created with the HIPAA provider private key. Refer to Figure 10-15 on page 191.



Figure 10-15 HIPAA Provider external profile AS Security configuration tab

In the Destinations tab, we now set up the URL where our HIPAA partner is waiting for HIPAA Partner AS2 messages. Therefore, we configure an AS2 destination. Figure 10-16 on page 192 shows the upper part of the configuration page, where we set up the destination URL and the document types that are enabled (we only use AS2).

Configure B2B Partner Profile

Main AS Security **Destinations** Contacts

B2B Partner Profile: HIPAA_Provider [up]

Apply Cancel Delete Undo Export | View

| Destination Name | Destination URL | Enabled Document Type |
|--------------------------------|-----------------------|-----------------------|
| AS2_2_HIPAA_provider (default) | as2://127.0.0.1:10070 | X12 |

Destinations

Destination Name: AS2_2_HIPAA_provider *

Enabled Document Type:

- XML
- X12
- EDIFACT
- Binary

Connection

Destination URL: as2:// 127.0.0.1:10070 *

Connection Timeout: 1800 Seconds

User name:

Password:

Figure 10-16 HIPAA Provider external profile Destinations tab details

If we scroll down (refer to Figure 10-17 on page 193), we find the rest of the configuration options. AS outbound security deals with how we send AS2 messages. In our HIPAA Provider internal profile, we stated that it was a requirement to sign and encrypt all the incoming messages. We discuss the signing information in “HIPAA Partner internal profile” on page 180.

In the Advanced AS Behavior section of Figure 10-17 on page 193, we request an MDN when we send our messages to the HIPAA Provider, and we request that message retransmission is attempted up to three times, in case a problem occurs.

| AS Outbound Security | |
|--|-------------------------------------|
| Send Messages Unsigned | <input type="checkbox"/> |
| Encrypt Messages | <input checked="" type="checkbox"/> |
| Encryption Certificate | hipaa_provider ▼ + ... * |
| Advanced AS Behavior | |
| Compress Messages | <input type="checkbox"/> |
| Request MDN | <input checked="" type="checkbox"/> |
| Time to Acknowledge | 1800 Seconds |
| Request Asynchronous MDN | <input type="checkbox"/> |
| Request Signed MDN | <input checked="" type="checkbox"/> |
| Attempt Message Retransmission | <input checked="" type="checkbox"/> |
| Maximum Retransmissions | 3 |
| <input type="button" value="Apply"/> <input type="button" value="Cancel"/> | |
| * | |

Figure 10-17 HIPAA Provider external profile Destinations tab details (continuation)

Step 4: Creating the HIPAA Partner B2B Gateway Service

After creating all the profiles for the scenario, it is time to create the B2B Gateway Services. For the HIPAA Partner, this service acts as the entry point to their system, identifying the incoming messages and mapping the incoming messages to the specific profiles that need to handle them. As we can see in Figure 10-18 on page 194, the HIPAA Partner B2B Gateway Service has two partners attached: HIPAA Partner internal profile and HIPAA Provider external profile. You can add as many partner profiles as your business needs require.

The HIPAA Partner B2B Gateway Service has two Front Side Handlers (FSHs), which are used to receive incoming traffic. For this particular case, an AS2 FSH handler is needed to receive messages coming from the HIPAA Provider, and an HTTP FSH is needed to receive messages from the HIPAA Partner back end.

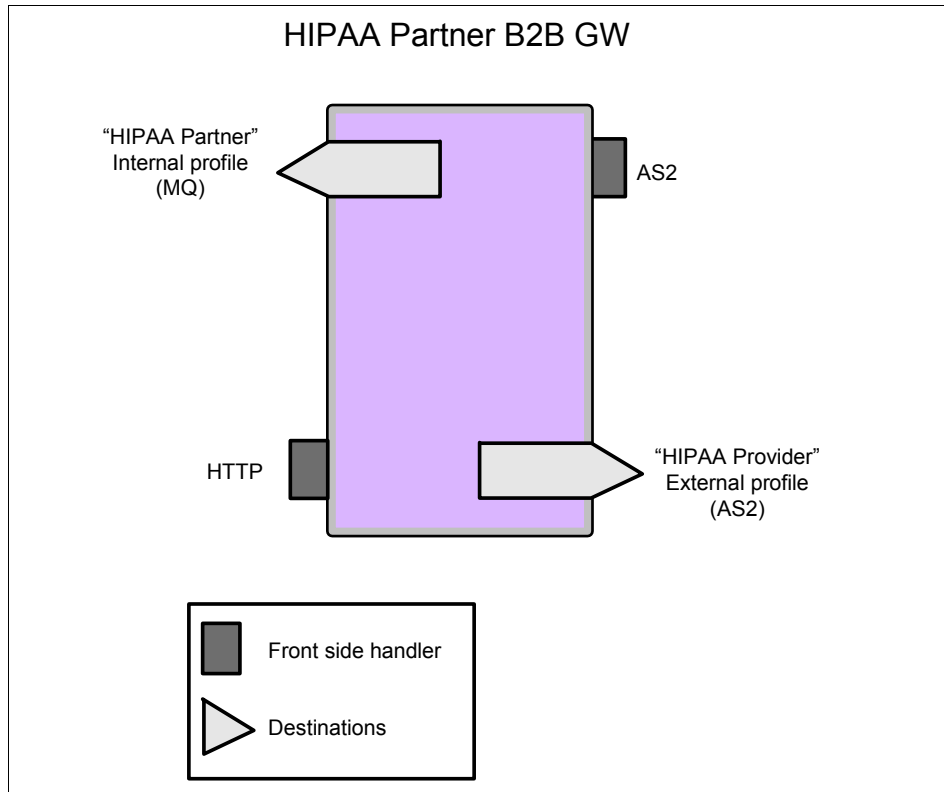


Figure 10-18 HIPAA Partner B2B Gateway architecture

Note: For details about how to configure Front Side Handlers, refer to the XB60 AS2 Trading Tutorial (which is found in the Additional Materials folder. You can access the Additional Materials folder by using the instructions in Appendix A, "Additional material" on page 389).

Figure 10-19 on page 195 shows the main window for the HIPAA Partner B2B Gateway Service.

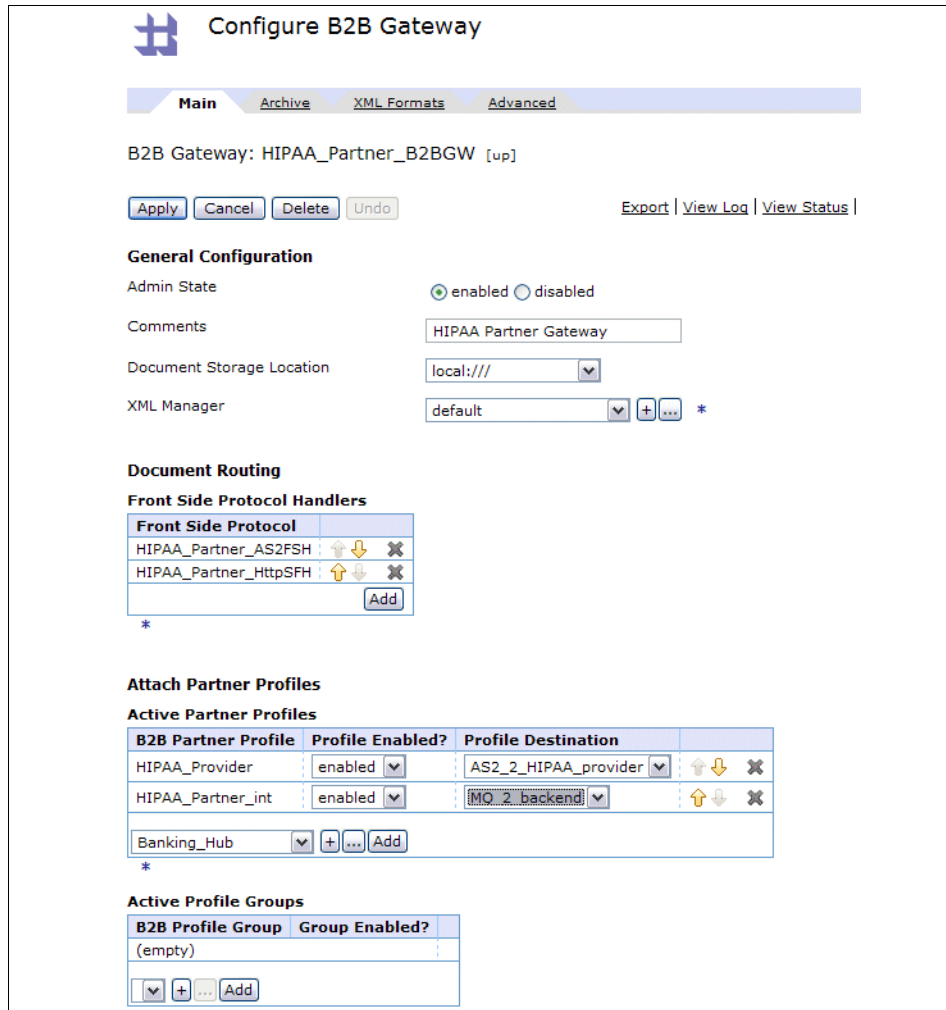


Figure 10-19 HIPAA Partner B2B Gateway Main configuration tab

Figure 10-19 shows both Front Side Handlers are attached (in the Front Side Protocol Handlers section) and the two Partner Profiles are attached (in the Attach Partner Profiles section). In order to add a new partner profile, select the profile that you want from the drop-down list and click **Add**. Then, select the Profile Destination if the profile has more than one destination, or leave it as default if the profile only has one destination.

Figure 10-20 on page 196 shows the detailed configuration on the AS2 FSH.

Main

AS2 Front Side Handler: HIPAA_Partner_AS2FSH [up]

Apply Cancel Undo

Admin State enabled disabled

Comments

Local IP Address tradingpartner *

Port Number 10071 *

HTTP Version to Client HTTP 1.1 ▾

Allowed Methods and Versions

- HTTP 1.0
- HTTP 1.1
- POST method
- GET method
- PUT method
- HEAD method
- OPTIONS
- TRACE method
- DELETE method
- URL with Query Strings
- URL with Fragment Identifiers
- URL with ..
- URL with cmd.exe

Persistent Connections on off

Figure 10-20 HIPAA Partner AS2 FSH

Important: Notice that we have created a host alias called tradingpartner in order not to write directly to 127.0.0.1, so that we do not couple IPs with services. For this particular scenario, we use this host alias on both B2B Gateway Services.

On the Archive configuration tab (Figure 10-21 on page 197), we have selected **Purge only** and entered 3 Days for the Archive Document age, even though you might need to resize this property in a production environment.

Configure B2B Gateway

Main Archive XML Formats Advanced

B2B Gateway: HIPAA_Partner_B2BGW [up]

Apply Cancel Delete Undo Export View Log View St

Archive Mode Purge Only *

Archive Document Age 3 Days

Disk Use Check Interval 60 Minutes

Maximum Disk Usage for Documents 25165824 Kilobytes

Figure 10-21 HIPAA Partner B2B Gateway Archive configuration tab

There are no XML Formats to consider in this use case, because we are not trading with XML messages. Do do not add anything on this XML Formats tab (shown in Figure 10-22).

Configure B2B Gateway

Main Archive XML Formats Advanced

B2B Gateway: HIPAA_Partner_B2BGW [up]

Apply Cancel Delete Undo Export View Log

XPath Routing Policies

(empty)

▼ Add + ...

Figure 10-22 HIPAA Partner B2B Gateway XML Formats tab

Our AS2 MDNs are set up as synchronous, so you do not need to set a Default AS2 MDN Return path. Do not make any configuration changes in the Advanced tab. Make sure to click **Apply** to save the configuration changes.

The screenshot shows the 'Configure B2B Gateway' interface with the 'Advanced' tab selected. The gateway is identified as 'HIPAA_Partner_B2BGW'. The configuration fields are as follows:

- Service Priority: Normal
- Default AS2 MDN Return Path: http://
- Default AS3 MDN Return Path: ftp://
- Document Routing Preprocessor: store:/// b2b-routing.xsl

Buttons include Apply, Cancel, Delete, Undo, Export, View Log, View Status, and Archive/purge transactions.

Figure 10-23 HIPAA Partner B2B Gateway Advanced tab

Step 5: Creating the HIPAA Provider B2B Gateway Service

For the HIPAA Provider case, our B2B Gateway Service acts as the entry point to their system, identifying the incoming messages and mapping the incoming messages with the specific profiles that need to handle them. As we can see in Figure 10-24 on page 199, the HIPAA Provider B2B Gateway Service has two partners attached: HIPAA Provider internal profile and HIPAA Partner external profile.

The HIPAA Provider B2B Gateway Service has two Front Side Handlers to receive incoming traffic. For this particular case, an AS2 FSH is needed to receive messages coming from the HIPAA Partner, and an MQ FSH is needed to receive messages from the WTX system back end.

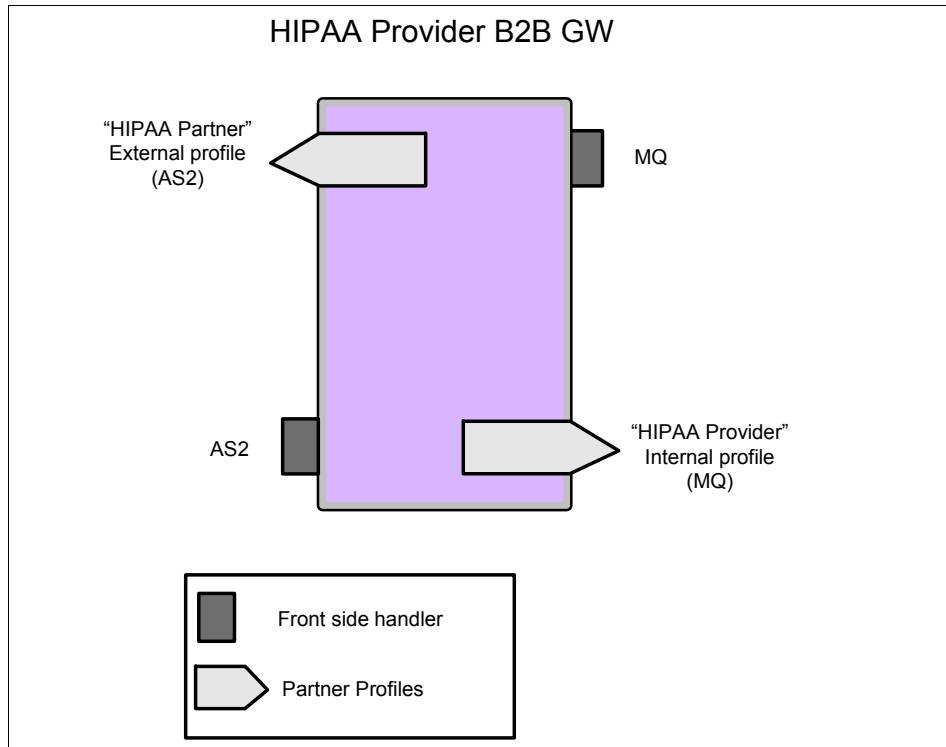


Figure 10-24 HIPAA Provider B2B Gateway architecture

Figure 10-25 on page 200 shows a view of the B2B Gateway Service main window with both Front Side Handlers and the two Partner Profiles.

Configure B2B Gateway

Main
Archive
XML Formats
Advanced

B2B Gateway: HIPAA_Provider_B2BGW [up]

[Export](#) | [View Log](#) | [View Status](#) | [Archive/p](#)

General Configuration

Admin State enabled disabled

Comments

Document Storage Location

XML Manager *

Document Routing

Front Side Protocol Handlers

| Front Side Protocol | ↑ | ↓ | ✕ |
|------------------------------------|---|---|---|
| MQ_FSH | ↑ | ↓ | ✕ |
| HIPAA_Provider_AS2FSH | ↑ | ↓ | ✕ |
| <input type="button" value="Add"/> | | | |

*

Attach Partner Profiles

Active Partner Profiles

| B2B Partner Profile | Profile Enabled? | Profile Destination | ↑ | ↓ | ✕ |
|---|------------------|---------------------|---|---|---|
| HIPAA_provider_int | enabled | MQ_2_backend | ↑ | ↓ | ✕ |
| HIPAA_Partner | enabled | AS2_2_HIPAAPartner | ↑ | ↓ | ✕ |
| <input style="width: 100px;" type="text" value="Banking_Hub"/> <input type="button" value="+"/> <input type="button" value="..."/> <input type="button" value="Add"/> | | | | | |

*

Active Profile Groups

| B2B Profile Group | Group Enabled? |
|---|----------------|
| (empty) | |
| <input type="button" value="↓"/> <input type="button" value="+"/> <input type="button" value="..."/> <input type="button" value="Add"/> | |

Figure 10-25 HIPAA Provider B2B Gateway Main configuration tab

Figure 10-26 on page 201 shows the detailed configuration on the AS2 FSH named HIPAA_Provider_AS2SFSH.

Main

AS2 Front Side Handler: HIPAA_Provider_AS2FSH [up]

[Export](#)

Admin State enabled disabled

Comments

Local IP Address *

Port Number *

HTTP Version to Client ▼

Allowed Methods and Versions

- HTTP 1.0
- HTTP 1.1
- POST method
- GET method
- PUT method
- HEAD method
- OPTIONS
- TRACE method
- DELETE method
- URL with Query Strings
- URL with Fragment Identifiers
- URL with ..
- URL with cmd.exe

Persistent Connections on off

Compression on off

Maximum Allowed URL Length

Maximum Allowed Total Header Length

Maximum Number of HTTP Request Headers Allowed

Figure 10-26 HIPAA Provider HTTP FSH

Figure 10-27 on page 202 shows the Archive tab view. Make sure to click **Apply** to save the changes.

Configure B2B Gateway

Main Archive XML Formats Advanced

B2B Gateway: HIPAA_Provider_B2BGW [up]

Apply Cancel Delete Undo Export View Log View Status

Archive Mode: Purge Only *

Archive Document Age: 3 Days

Disk Use Check Interval: 60 Minutes

Maximum Disk Usage for Documents: 25165824 Kilobytes

Figure 10-27 HIPAA Provider B2B Gateway Archive tab in the configuration

Step 6: Creating the inbound mapping system

Now, it is time to build the transformation infrastructure to map the incoming Health care payment in X12 837 to the flat file format required by the back end systems.

We have already stated that these transformations are triggered by a message arriving into a specific MQ Queue; therefore, we are dealing with an *event-based* transformation that requires a Launcher system to handle it.

Important: Although we chose Launcher in this scenario as the event-driven engine, you can use other methods, such as WebSphere Message Broker or WebSphere Enterprise Service Bus (WSB), which also are tightly integrated with WebSphere Transformation Extender.

A portion of the incoming data that will be passed from the DataPower XB60 device to the MQ Queue is shown in Example 10-1 on page 203.

Example 10-1 Input data coming from X12 837 format

```
ISA*00*          *00*          *01*hipaapartner  *01*hipaaprovider
*010408*1058*U*00401*000000203*0*T*:
GS*HC*PROFSERV*DEVELOPMENT*20010101*120000*1*X*004010X098
ST*837*0001
BHT*0019*00*0123*19981015*1023*RP
REF*87*004010X098
NM1*41*2*PREMIER BILLING SERVICE*****46*TGJ23
PER*IC*JERRY*TE*3055552222*EX*231
NM1*40*2*REPRICER XYZ*****46*66783JJT
HL*1**20*1
NM1*85*2*PREMIER BILLING SERVICE*****24*587654321
N3*234 SEAWAY ST
N4*MIAMI*FL*33111
NM1*87*2*KILDARE ASSOC*****24*581234567
N3*2345 OCEAN BLVD
N4*MIAMI*FL*33111
HL*2*1*22*0
```

Notice that 01hipaapartner and 01hipaaprovider are the business IDs that we configured earlier for the Partner profile objects.

Example 10-2 shows a part of the expected output that the system will produce after the transformation is complete.

Example 10-2 Output coming from the mapping inbound 837

```
NUM_CLAIMS=7
HDR,587654321,PREMIER BILLING SERVICE,ALLIANCE HEALTH AND LIFE
INSURANCE, TEDSMITH,,SELF,26462967,100
LINE,1,19981003,HC,99213,,UN,1,40
LINE,2,19981003,HC,99214,,UN,1,15
LINE,3,19981003,HC,87072,,UN,1,35
LINE,4,19981010,HC,86663,,UN,1,10
HDR,587654321,PREMIER BILLING SERVICE,KEY INSURANCE
COMPANY,JANESMITH, TEDSMITH,DEPENDENT,26463774,100
LINE,1,19981003,HC,99213,,UN,1,40
LINE,2,19981003,HC,99214,,UN,1,15
LINE,3,19981003,HC,87072,,UN,1,35
LINE,4,19981010,HC,86663,,UN,1,10
....
```

Inbound map

Now, we look at the map structure.

Important: We do not discuss field-level detail for the mapping specification. We describe the major guidelines of the map structure so that you understand what type of transformation this map performs. For more information about how to work with transformations, refer to *IBM WebSphere Transformation Extender 8.2*, SG24-7693-00.

The transformation map Inbound837 uses the Partner X12 Inbound Interchange EDI component of the HIPAA_X12_835_837.mtt, a type tree that is included in the WebSphere Transformation Extender pack for HIPAA. We will not have to create a type tree representation for that format. For the output type tree component, it uses Claim_FlatFile.mtt, which contains a representation of the flat file format that will be consumed by the back end.

From a mapping perspective, the key element is that at the Claims level, we will need to split the mapping between Subscriber claims and Patient claims. We will create two functional maps for each case and will use the EITHER function to choose between these functional maps as shown in Figure 10-28.

```
=Either( F_Subscriber_Claim(Claim CLM:Subscriber:Provider HL
Loop::TransmissionIn, Subscriber:Provider HL Loop::TransmissionIn,
Provider HL Loop::TransmissionIn),
        F_Patient_Claim(Claim CLM:Patient:Subscriber:Provider HL
Loop::TransmissionIn, Patient:Subscriber:Provider HL Loop::TransmissionIn,
Subscriber:Provider HL Loop::TransmissionIn,
Provider HL Loop::TransmissionIn))
```

Figure 10-28 Functional map call to choose between subscriber and patient claim

Figure 10-29 on page 205 and Figure 10-30 on page 206 show the logic of both functional maps, including input and output cards.

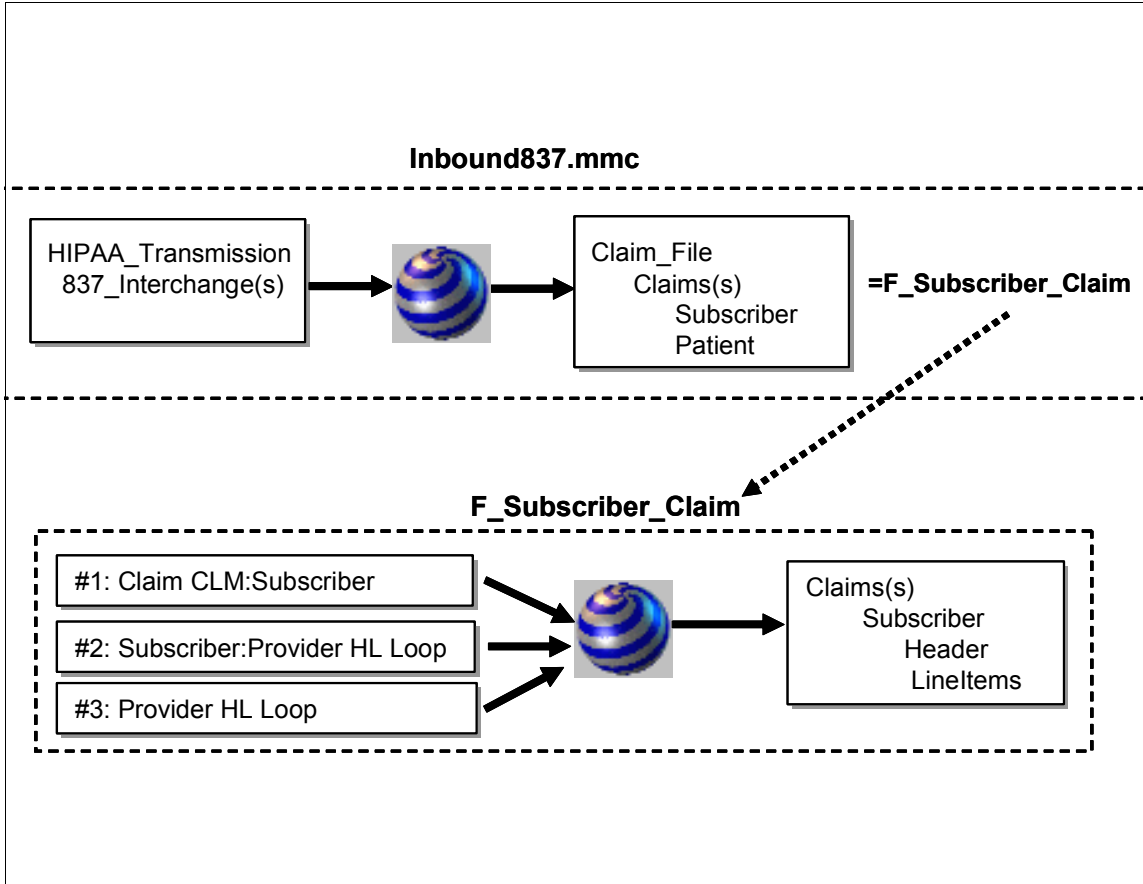


Figure 10-29 Logical structure of the Inbound837.mmc map when we work with a Subscriber claim

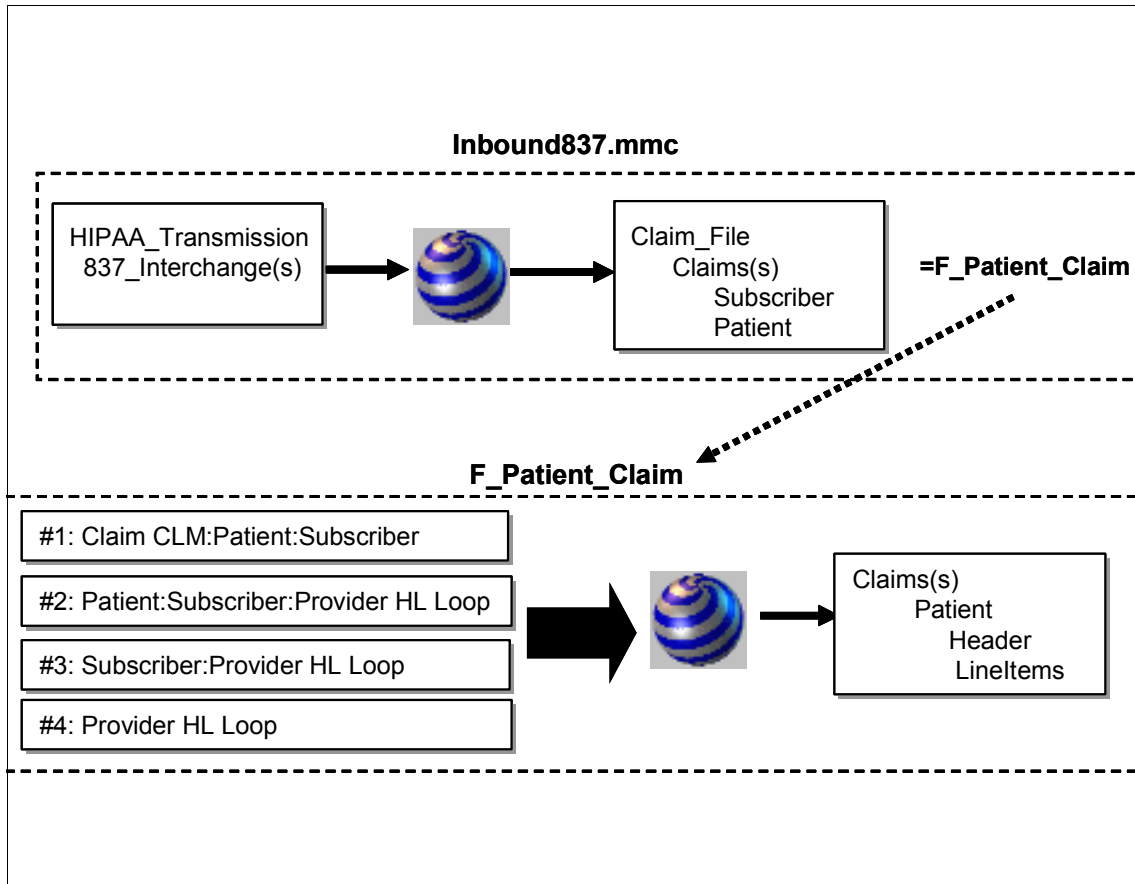


Figure 10-30 Logical structure of the `Inbound837.mmc` map when we work with a Patient claim

Input system

After creating the map, it is time to create the Launcher system by using the Integration Flow Designer tool, which comes with WebSphere Transformation Extender V8.2. Create an event-based system that listens on Q1 from the Queue Manager QM, and put the output transformed file in Q2 from the same Queue Manager.

These are the settings that you use for the Input Card and Output Card inside Map Settings.

Notice that the two most important aspects are the MQ syntax and that the Source Event parameter is set to 0N, which will cause the system to be triggered every time a message arrives on Q1.

Figure 10-31 shows a widely used property within Transaction where you can specify what to do with the message in case of success (the On Success property) and in the case of failure (the On Failure property). This configuration is commonly used. If the message is transformed successfully, it will be deleted from the input queue. The transaction will be rolled back if the transformation fails, so that no message is lost.

| | |
|----------------------|--------------------------|
| Input(s) | |
| #1 TransmissionIn | |
| FetchAs | Integral |
| GET | |
| Source | IBM WebSphereMQ (server) |
| Command | -QMN QM1 -QN Q1 |
| Transaction | |
| OnSuccess | Delete |
| OnFailure | Rollback |
| Scope | Map |
| Warnings | Ignore |
| Retry | |
| DocumentVerification | Never |
| Backup | |
| SourceEvent | ON |
| Metadata (XML) | |

Figure 10-31 Input Card map settings for Inbound 837 system

Figure 10-32 shows the Output Card with Q2.

| | |
|----------------------|--------------------------|
| Output(s) | |
| #1 Flat_File_Out | |
| PUT | |
| Target | IBM WebSphereMQ (server) |
| Command | -QMN QM1 -QN Q2 |
| Transaction | |
| Retry | |
| DocumentVerification | Never |
| Backup | |
| Metadata (XML) | |

Figure 10-32 Output Card map settings for Inbound 837 system

Figure 10-33 on page 208 shows how the final system appears. The small pair of glasses in the center of the figure indicates that it is an event-based system.

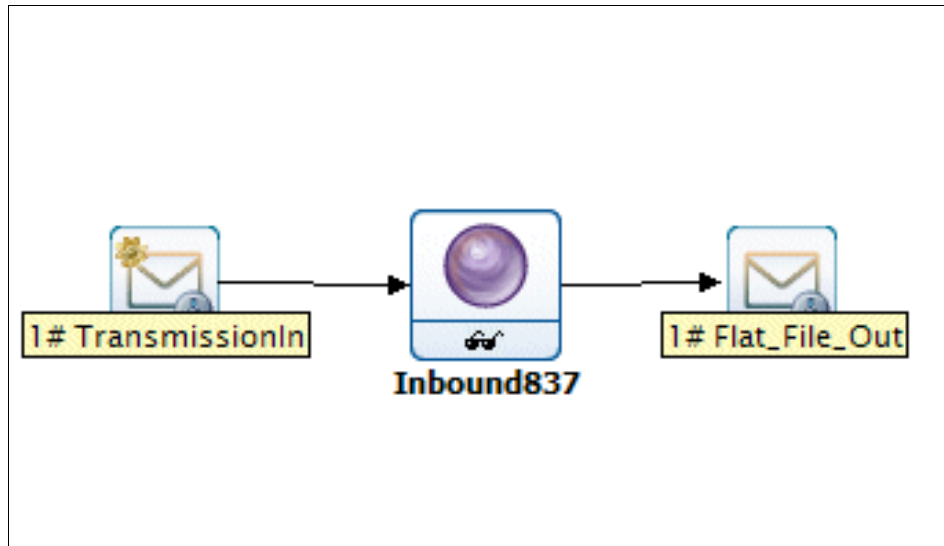


Figure 10-33 Launcher system created to handle inbound mapping

After completing those steps, deploy the system in your system and start the Windows® Launcher service. Your system will be ready to transform.

Step 7: Creating the outbound mapping system

Now we must build the transformation infrastructure that will map the outgoing flat file coming from our back-end system to a well-formed Health Claim payment in X12 835 to be sent to our trading partner.

Similarly to the previous step, our transformations will be triggered by a message arriving into a specific MQ Queue, and another Launcher system will be used.

Important: Although we chose Launcher in this scenario for the event-driven engine, you can use other methods, such as WebSphere Message Broker or WebSphere ESB, which both are tightly integrated with WebSphere Transformation Extender.

A portion of the incoming data that will be passed from the back-end system device to the WTX system is shown in Example 10-3.

Example 10-3 Sample custom flat file format coming from the HIPAA Provider system

```

HDR:019382 07-Jul-2004ASCENTIAL SOFTWARE 2424 FEDERAL HWY BOCA RATON
FL06897012-34-5678
PYE:EMMINENT PROVIDER 999-26-1039
  
```


| | | | | | |
|-------------------------|--------|---------|-----------|---|-----------|
| CLM:01900324KIRSHENBAUM | | DAVID | | L | |
| INS:KIRSHENBAUM | ROBERT | | H123 MAIN | | IOWA CITY |
| IA52240999-64-3019 | | | | | |
| LIN:07/28/2001 A0431 | 2300 | | ONME | | |
| LIN:07/28/2001 A0436 | 341.28 | | ONME | | |
| LIN:07/28/2001 A0426 | 18.34 | 14.67 | COP | | |
| LIN:07/28/2001 C1039 | 240 | 192 | COP | | |
| CLM:01900325MALVINE | | BLANCHE | | P | |

Example 10-4 shows part of the expected output that the system will produce after the transformation is completed.

Example 10-4 Output X12 835 after the outbound 835 mapping

```

ISA*00*          *00*          *zz*hipaprovider *zz*hipapartner
*090311*1628*U*00401*000000452*0*T*<
GS*HP*hipaprovider*524652145*20090311*16280409*452*X*004010X091
ST*835*1
BPR*H*294.62*C*NON*****20090311
TRN*1*20040707019382*999 26 103
DTM*405*20040707
N1*PR*ASCENTIAL SOFTWARE
N3*2424 FEDERAL HWY
N4*BOCA RATON*FL*06897
N1*PE*EMMINENT PROVIDER*FI*999-26-1039
LX*1
CLP*01900324*1*2899.62*206.67*2692.95*13*999-64-3019
NM1*QC*1*KIRSHENBAUM*DAVID*L
NM1*IL*1*KIRSHENBAUM*ROBERT*H***34*999-64-3019
DTM*233*20010728
SVC*HC<A0431*2300*0
CAS*PR*40*2300
SVC*HC<A0436*341.28*0
CAS*PR*40*341.28

```

Note that zzhpaapartner and zzhpaaprovider are the business IDs that we configured before as the Partner profile objects. These business IDs need to be set at the mapping level, at InterchangeSenderID and Interchange Rcv'rID element, as part of the ISAPartner Info group.

Outbound map

Important: We do not discuss field-level detail for the mapping specification. We describe the major guidelines of the map structure so that you understand what type of transformation this map performs. For more information about how to work with transformations, refer to *IBM WebSphere Transformation Extender 8.2*, SG24-7693.

The transformation map Outbound835 uses the Partner X12 Inbound Interchange EDI component of the HIPAA_X12_835_837.mtt, a type tree that is included in the WebSphere Transformation Extender pack for HIPAA, which means that we will not have to create a type tree representation for that format. As the output type tree component, the transformation map uses Payment Advice.mtt, which contains a representation of the flat file format sent by the back end.

From a mapping perspective, the most remarkable aspects are that several functional maps need to be created in order to manage the data. Figure 10-34 shows the functional map needed to map each transaction separately.

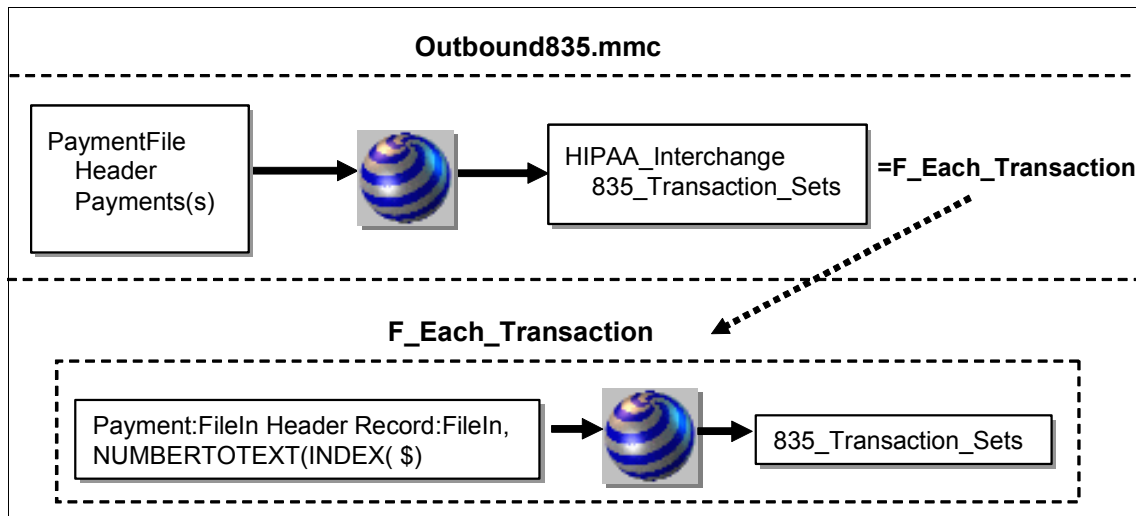


Figure 10-34 Functional map for mapping Each Transaction

After we have each transaction mapped separately, each claim needs to be treated individually in a new Functional map call as depicted in Figure 10-35 on page 211.

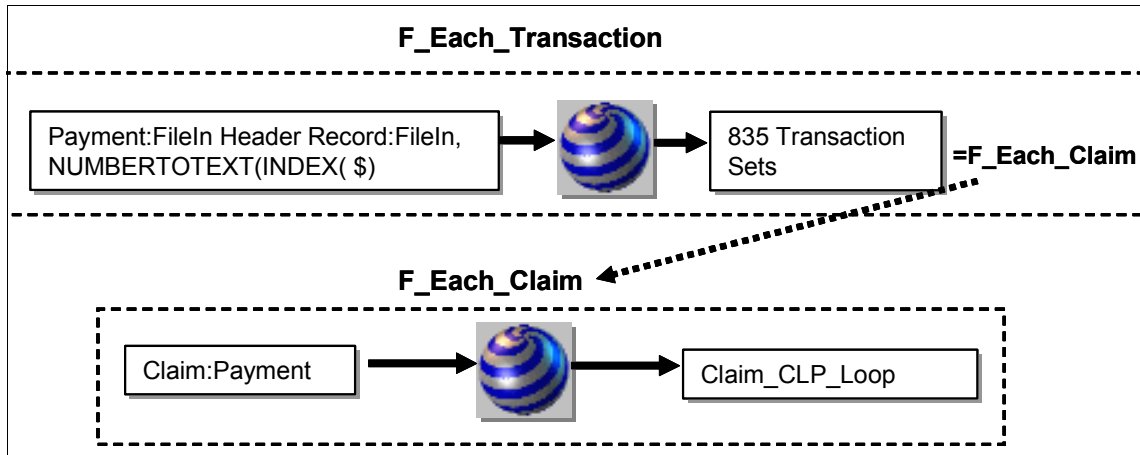


Figure 10-35 Functional map for mapping Each Claim

Inside every claim, we need to process each single service line separately, so a new functional map call needs to be done at this level (Figure 10-36).

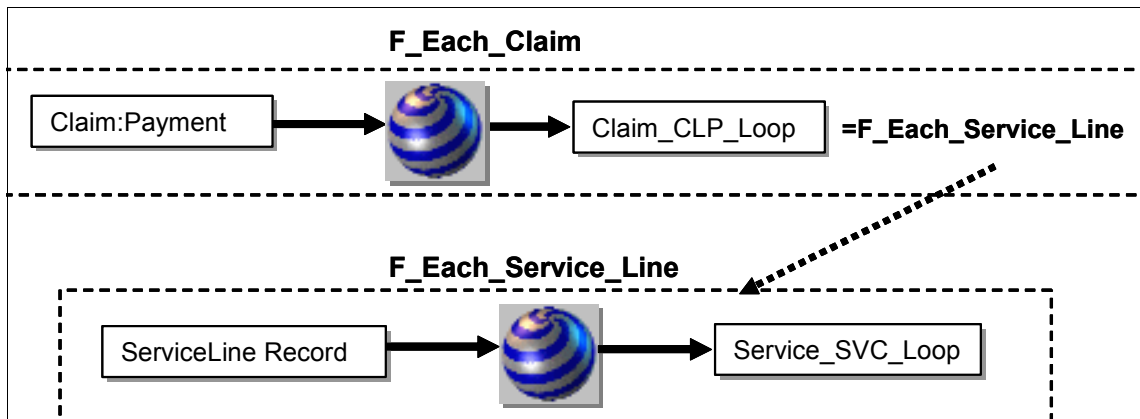


Figure 10-36 Functional map for mapping Each Service Line

Finally, each Service Line requires a specific Claim adjustment to be mapped to the CAS segment from the X12 format. Notice that this call will only be made if the first index of M'amt (Monetary amount field) in the SVC segment is not equal to the SVC segment monetary amount field (M'amt). You can see this call in Figure 10-37 on page 212.

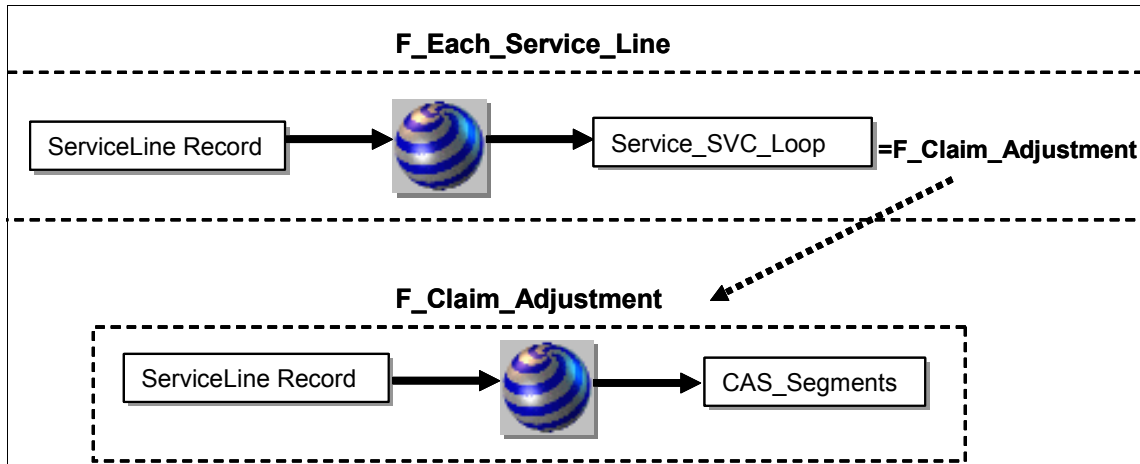


Figure 10-37 Functional map for mapping Each Claim Adjustment

Outbound system

After creating the map, create the Launcher system using the Integration Flow Designer tool, which comes with WebSphere Transformation Extender V8.2, and create an event-based system that will listen on Q3 from our Queue Manager QM, and put the output transformed file in Q4 from the same Queue Manager.

These are the major settings that you use for the Input Card and Output Card inside Map Settings.

From an Input Card perspective (Figure 10-38 on page 213), the two most important aspects are the MQ syntax and that Source Event parameter is set to 0N. These settings cause the system to be triggered every time that a message arrives on Q3.

Figure 10-38 on page 213 shows a widely used property within Transaction where you can specify what to do with the message in case of success (On Success property) and in the case of failure (On Failure property). This configuration is commonly used. If the message is transformed successfully, it will be deleted from the input queue. The transaction will be rolled back if the transformation fails so that no message is lost.

| | |
|----------------------|--------------------------|
| Input(s) | |
| #1 Flat_File_In | |
| FetchAs | Integral |
| GET | |
| Source | IBM WebSphereMQ (server) |
| Command | -QMN QM1 -QN Q3 |
| Transaction | |
| OnSuccess | Delete |
| OnFailure | Rollback |
| Scope | Map |
| Warnings | Ignore |
| Retry | |
| DocumentVerification | Never |
| Backup | |
| SourceEvent | ON |
| Metadata (XML) | |

Figure 10-38 Input Card map settings for outbound 835 system

Figure 10-39 shows the Output Card where the MQ Syntax is similar except that the queue name is Q4.

| | |
|----------------------|--------------------------|
| Output(s) | |
| #1 InterchangeOut | |
| PUT | |
| Target | IBM WebSphereMQ (server) |
| Command | -QMN QM1 -QN Q4 |
| Transaction | |
| Retry | |
| DocumentVerification | Never |
| Backup | |
| Metadata (XML) | |

Figure 10-39 Output Card map settings for outbound 835 system

Figure 10-40 on page 214 depicts the outbound system.

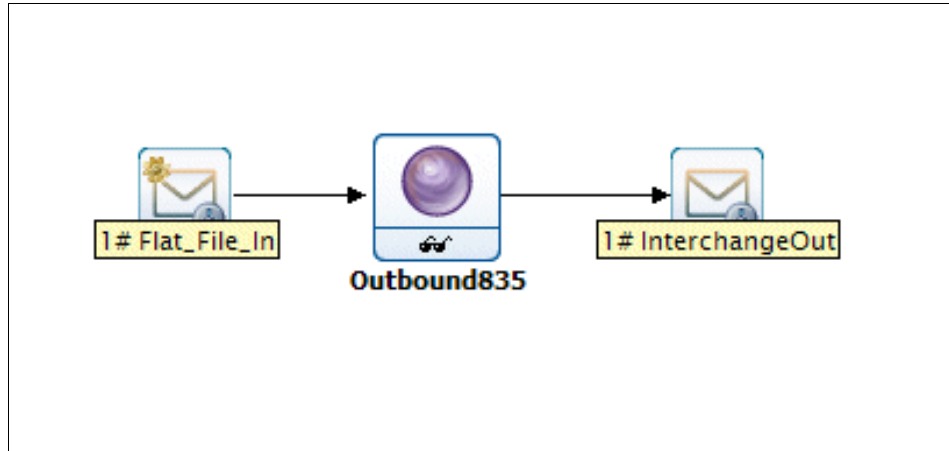


Figure 10-40 Launcher system created to handle outbound mapping

The small pair of glasses indicates that it is an event-based system.

After completing those steps, deploy the system in your system and start the Windows Launcher service. Your system will be ready to transform.

The scenario is ready to be tested.

10.5 Testing our solution

Everything has been successfully configured and the infrastructure is “up and running,” so now it is time to test our scenario and actually see the transaction results in our Transaction Viewer and the system logs. In the interest of simplicity, testing is divided into inbound flow and outbound flow.

10.5.1 Inbound flow

Before we actually start with the testing, we review the overview of the inbound flow and the actual steps that will be performed. Refer to Figure 10-41 on page 215.

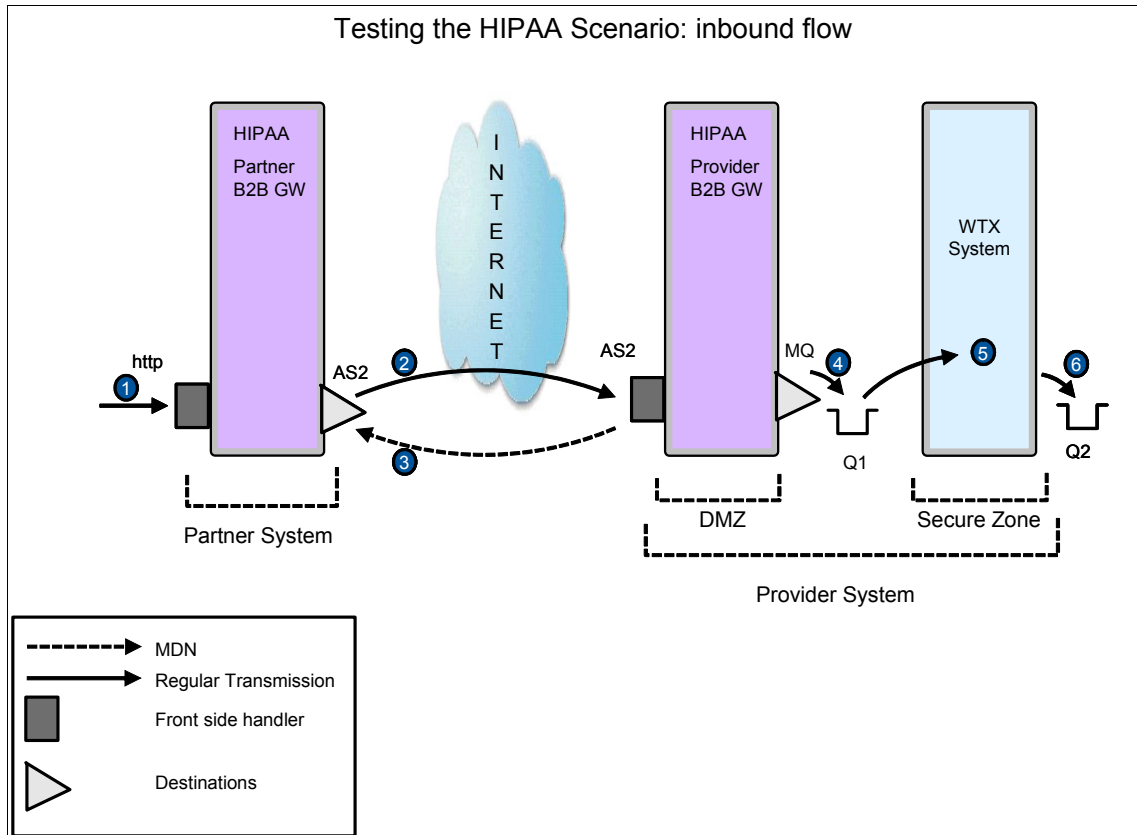


Figure 10-41 Inbound flow steps for HIPAA scenario

Here are the steps and their explanations. Messages can be sent using an HTTP utility. In our case, we use NetTool (Figure 10-42 on page 216):

1. The message (HIPAAdata.txt) is sent to the http HIPAA_Partner_HttpSFH (10073), which is in HIPAA_Partner_B2BGW service.

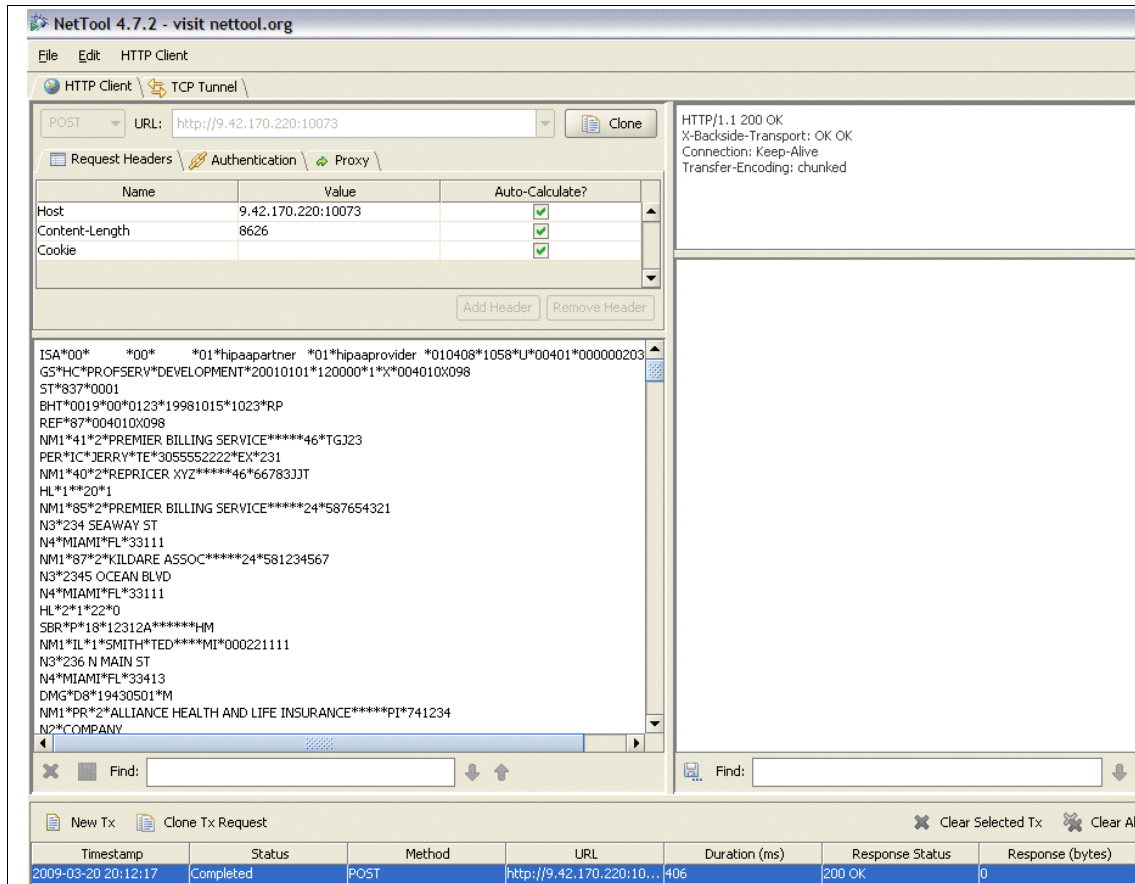


Figure 10-42 Sample of how to perform an HTTP message using NetTool

2. The message is wrapped in AS2 and sent using HIPAA_Provider profile AS2 Destination to port 10070.
3. The AS2 Message is received by HIPAA_Provider_B2BGW with HIPAA_Provider_AS2FSH (10070) and sends MDN back to port 10071.
HIPAA_Partner_B2BGW receives MDN within the same connection.
4. Message is unwrapped and sent to MQ using HIPAA_provider_int profile with MQ_2_Backend destination (Q1).
5. WTX is listening on Q1 and transforms the message using Inbound_system.ms1.
6. The flat file is put on Q2.

Testing results

Figure 10-43 shows how our inbound flow looks in the B2B Transaction Viewer. Note the Gateway Name column and the inbound successful results.

| Transaction Set ID | Transaction ID | Gateway Name | SenderName (ID) / Receiver (ID) | Inbound URL / Outbound URL | Input Time / Output Time | Result Code | MDN Status | MDN Time | MDN Received |
|--|----------------|----------------------|--|---|--|-------------|------------------------|--------------------------|--------------------------|
| <input type="checkbox"/> 229 | 1956 | HIPAA_Provider_B2BGW | Sender: 01hipaapartner (01hipaapartner) Receiver: 01hipaprovider (01hipaprovider) | as2://127.0.0.1:10070/ dpmq://QM1/? RequestQueue=Q1 | 2009-03-20 15:12:56.0 2009-03-20 15:12:56.0 | Success | Sent (Positive) | 2009-03-20 15:12:56.0 | |
| <input type="checkbox"/> 228 | 2115 | HIPAA_Partner_B2BGW | Sender: 01hipaapartner (01hipaapartner) Receiver: 01hipaprovider (01hipaprovider) | http://9.42.170.220:10073/ as2://127.0.0.1:10070 | 2009-03-20 15:12:56.0 2009-03-20 15:12:56.0 | Success | Received (Positive) | | 2009-03-20 15:12:56.0 |

Figure 10-43 Transaction Viewer of the inbound flow

In Figure 10-44, you can see in the Document Type column that this message is an X12 837 message.

| Headers | Document ID | Document Type | MDN |
|----------------|-------------|---------------|-----|
| (Show Headers) | 000000203 | 837 | 0 |
| (Show Headers) | 000000203 | 837 | 0 |

Figure 10-44 Detail on Transaction Viewer of Document ID and Type

If we click any of the Transaction Set ID hyperlinks, we can see various parts of the document as depicted in Figure 10-45.

| Modify Query Refresh | | | | | | |
|--|---------------------|---|----------------------|---|---------------------------------|--------------------------|
| | Transaction Set ID | Transaction ID | Gateway Name | SenderName (ID) / Receiver (ID) | Inbound URL / Outbound URL | Input Time / Output Time |
| <input type="checkbox"/> | | | HIPAA_Provider_B2BGW | Sender: 01hipaapartner (01hipaapartner) | as2://127.0.0.1:10070/ | 2009-08-20 15:15:15 |
| | | <div style="border: 1px solid black; padding: 5px;"> Show Document X Input Output Content MDN </div> | | Receiver: 01hipaprovider (01hipaprovider) | dpmq://QM1/? RequestQueue=Q1 | 2009-08-20 15:15:15 |
| <input type="checkbox"/> | 228 | 2115 | HIPAA_Partner_B2BGW | Sender: 01hipaapartner (01hipaapartner) | http://9.42.170.220:10073/ | 2009-08-20 15:15:15 |
| | | | | Receiver: 01hipaprovider (01hipaprovider) | as2://127.0.0.1:10070 | 2009-08-20 15:15:15 |
| | | | | Sender: | | 2009-08-20 15:15:15 |

Figure 10-45 Detail of how to download and show various content

10.5.2 Outbound flow

Now, we review an overview of the outbound flow and the steps that are performed. To get started, refer to Figure 10-46 on page 219.

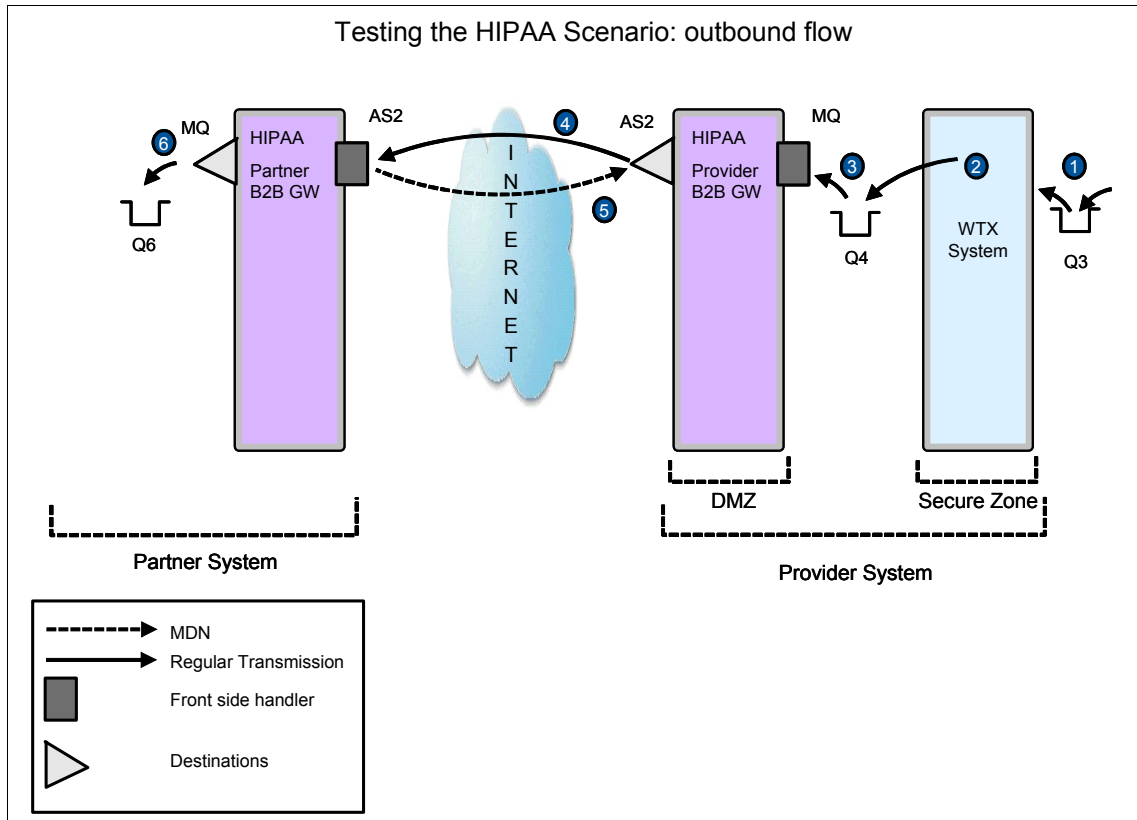


Figure 10-46 Outbound flow steps for HIPAA scenario

Next, we describe each step. An MQ utility is used to put messages on queues. In our case, we used RfhUtil:

1. The message (Flat_Payment_Advice.txt) is put on Q3 as shown in Figure 10-47 on page 220.

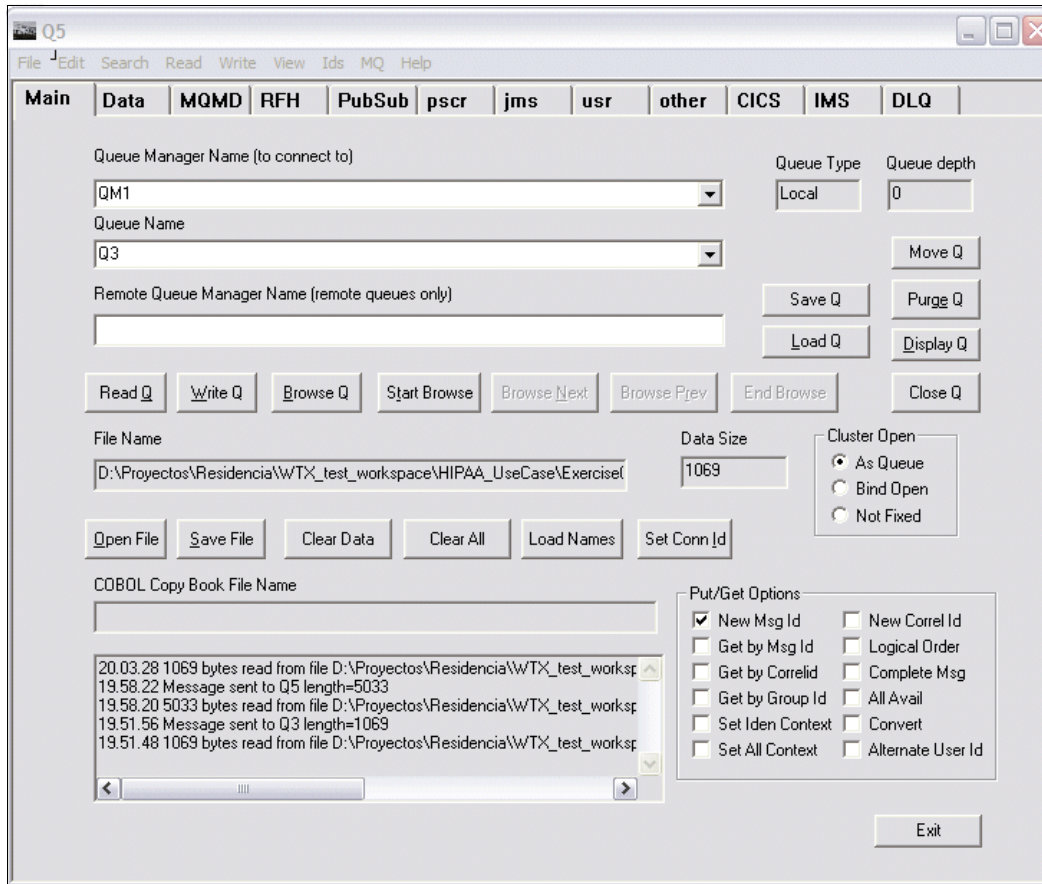


Figure 10-47 Putting the test message into Q3 using RfhUtil

2. WTX transforms the message into an 837 HIPAA format and puts its result in Q4.
3. The message is received in HIPAA_Provider_B2BGW with MQ_FSH.
4. The message is wrapped in AS2 and sent to HIPAA_Partner_B2BGW using the HealthPartner profile (AS2_2_Healthpartner destination, on port 10071) which receives the message.
5. An MDN is sent through 10071 to HIPAA_Provider_B2BGW.
HIPAA_Partner_B2BGW receives the AS2 message using HIPAA_Partner_AS2FSH, which is included in it.
6. HIPAA_Partner_B2BGW unwraps the message and sends it back through the back end, putting it on Q6.

Testing results

Figure 10-48 shows the outbound flow in the Transaction Viewer. This view is a comprehensive view of the message flow.

| Transaction Set ID | Transaction ID | Gateway Name | SenderName (ID) / Receiver (ID) | Inbound URL / Outbound URL | Input Time / Output Time | Result Code | MDN Status | MDN Time | MDN Received | Headers | Document ID | Document Type | MDN |
|--|----------------|----------------------|--|--|--|-------------|---------------------|-----------------------|-----------------------|----------------|-------------|---------------|-----|
| <input type="checkbox"/> 227 | 4566 | HIPAA_Partner_B2BGW | Sender: zzhpaaprovider (zzhpaaprovider) Receiver: zzhpaapartner (zzhpaapartner) | as2://127.0.0.1:10071/ dpmq://QM1/?RequestQueue=Q6 | 2009-03-20 15:04:16.0 2009-03-20 15:04:16.0 | Success | Sent (Positive) | 2009-03-20 15:04:17.0 | | (Show Headers) | 000000452 | 835 | 0 |
| <input type="checkbox"/> 226 | 13105 | HIPAA_Provider_B2BGW | Sender: zzhpaaprovider (zzhpaaprovider) Receiver: zzhpaapartner (zzhpaapartner) | dpmq://QM1/MQ_FSH?RequestQueue=Q4 as2://127.0.0.1:10071 | 2009-03-20 15:04:16.0 2009-03-20 15:04:16.0 | Success | Received (Positive) | | 2009-03-20 15:04:17.0 | (Show Headers) | 000000452 | 835 | 0 |

Figure 10-48 Transaction Viewer of the outbound flow

In Figure 10-49 on page 222, you can see in the Document Type column that this message is an X12 835 message.

| MDN Time | MDN Received | Headers | Document ID | Document Type | MDN | Action |
|--------------------------|--------------------------|----------------|-------------|---------------|-----|------------------------|
| 2009-03-20 15:04:17.0 | | (Show Headers) | 000000452 | 835 | 0 | Resend |
| | 2009-03-20 15:04:17.0 | (Show Headers) | 000000452 | 835 | 0 | Resend |

Figure 10-49 Transaction Viewer detail of the Document ID and Type

Figure 10-50 on page 223 depicts the message content at Q6, which arrived successfully as expected.

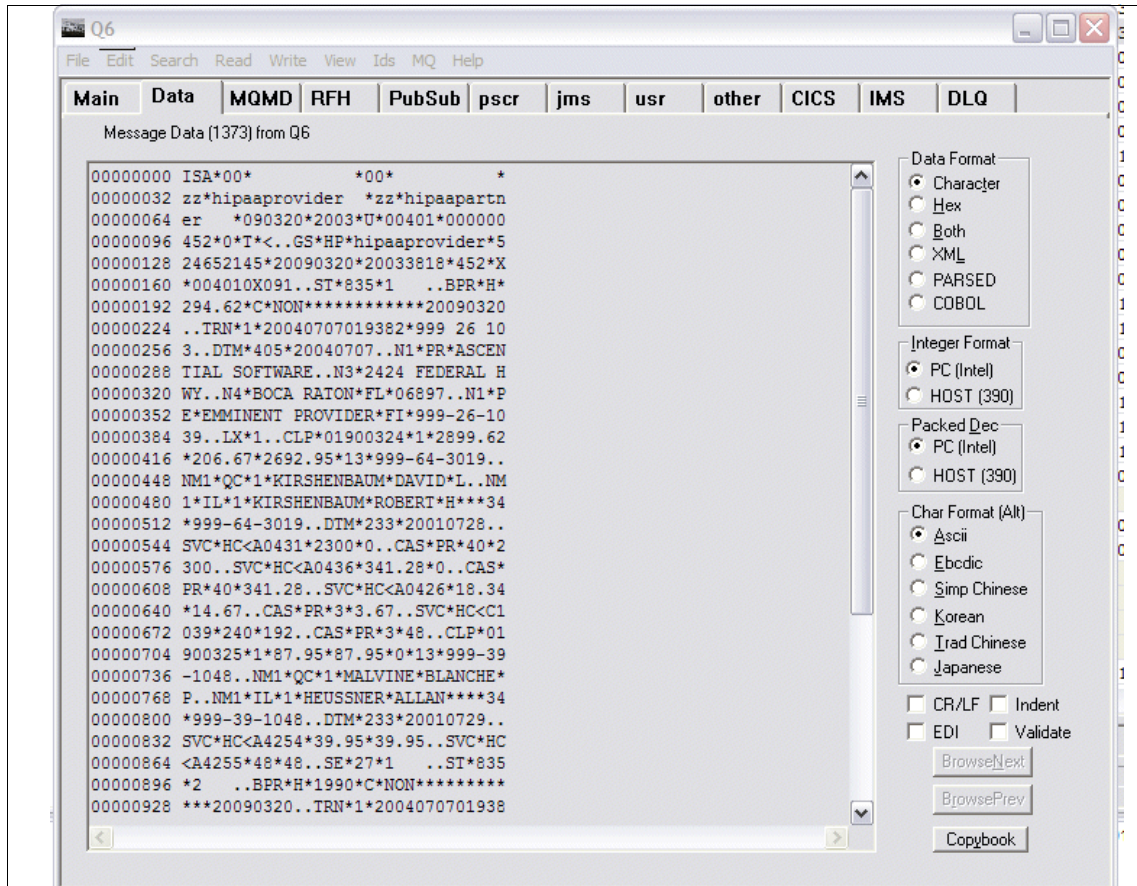


Figure 10-50 Output content coming from Q6 browsed with RfhUtil



XB60 with transformation

In this chapter, we show you how to use WebSphere DataPower B2B Appliances XB60 for trading scenarios where data transformation is performed within the device in run time.

11.1 Business value

There are many B2B scenarios where electronic data interchange (EDI) is not the standard that is used, and other data formats, such as flat file, XML, iDocs, and so forth, might be used. In these scenarios, the DataPower XB60 can play a key role by being positioned inside the demilitarized zone (DMZ), performing the tasks of both transformation and partner management. This way, transformation needs can be covered, and back-end systems do not have to worry about data integration with our partners, because it will be handled by the device.

In order to present an example of the XB60 performing data transformation, we introduce another business scenario where a banking hub needs to trade payments with their trading partners in a secure fashion. The banking hub receives data from its back-end Enterprise Resource Planning (ERP) system that needs to be sent to its partners.

This customer needs the following core requirements and business value:

- ▶ The customer must have the ability to verify partner information in the DMZ.
- ▶ The customer requires support for receiving AS2 B2B messages.
- ▶ All AS3 data must be signed and encrypted.
- ▶ AS3 Message Disposition Notifications (MDNs) must be signed.
- ▶ The customer must be able to transform XML documents to and from flat file format.
- ▶ The customer requires the ability to validate XML documents against their supported schema.
- ▶ Any payload data stored on the appliance must be encrypted.

Next, we look at how we can address all of these requirements.

11.2 Prerequisites

This scenario requires the following skills and software.

11.2.1 Software prerequisites

In order to be able to run this scenario, you need to install the following components:

- ▶ WebSphere DataPower B2B Appliances XB60

- ▶ WebSphere Transformation Extender V8.2 Design Studio
- ▶ WebSphere MQ V6

11.2.2 Skills prerequisite

This scenario is intended for an Intermediate user, which means that in order to be able to fully implement and understand this scenario, you must be familiar with:

- ▶ WebSphere DataPower B2B Appliances XB60 major concepts
- ▶ WebSphere Transformation Extender basic mapping techniques

11.3 Presenting the scenario

In this section, we provide an overview of a banking integration scenario. In this case, we implement a one-way flow, which corresponds with an incoming payment from the hub Enterprise Resource Planning (ERP) system, that needs to be sent to a specific banking partner. A message will come through the DataPower device via WebSphere MQ (WMQ) queues and perform a message transformation to XML. This XML will then be wrapped in AS3 and then sent to the specific partner's back end. The DataPower device will wait for an asynchronous AS3 Message Disposition Notification (MDN) before the transaction is treated as complete.

All the messages between the partner and the provider will be exchanged in encrypted format and signed as a consequence of the assumed trading manager agreement.

Figure 11-1 on page 228 shows the flow from a high-level perspective.

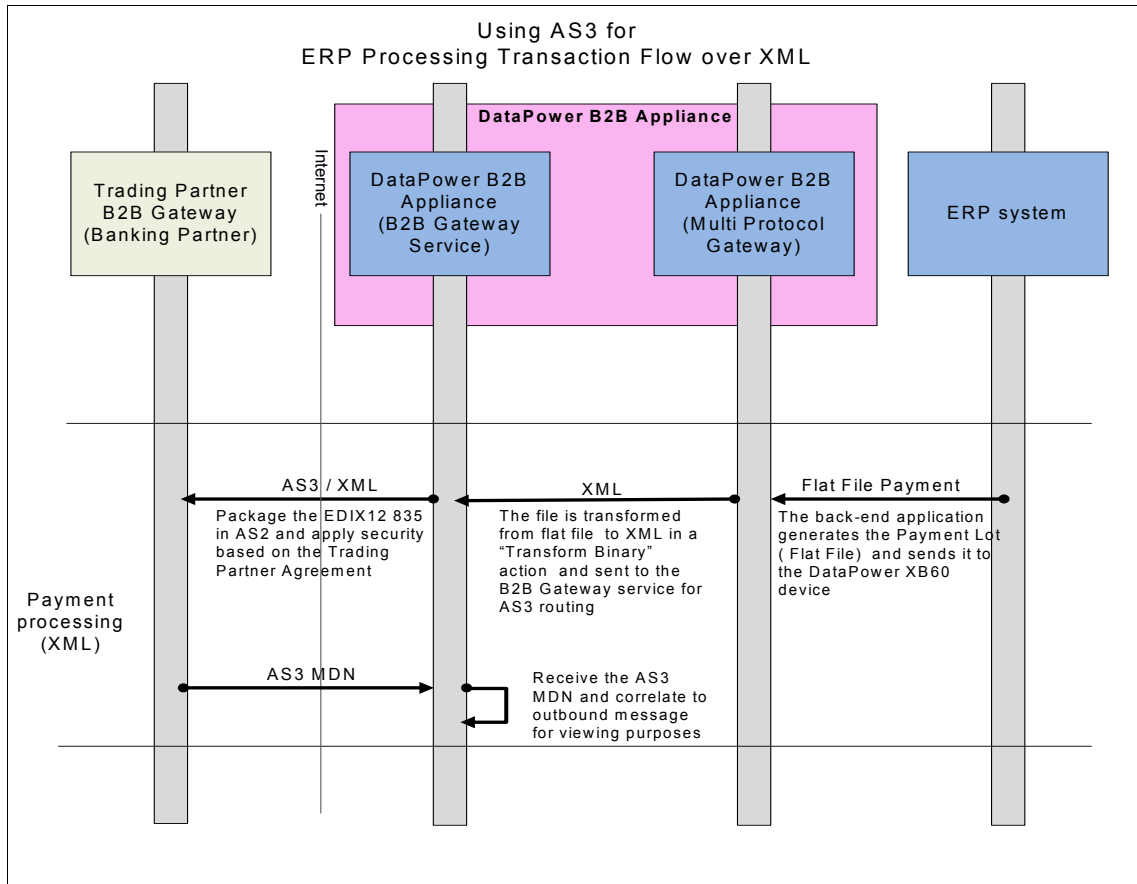


Figure 11-1 Scenario overview

Important: We merely use an ERP system as the hub back end for an example. You can use this same scenario with many back ends.

11.4 Scenario solution

After we describe the major aspects of the scenario from a high-level perspective, we will provide a technical explanation of all the steps to fully implement the solution.

One of the key aspects of the scenario implementation is that we simulate the trading partner within the DataPower device, which means that you need to create a specific B2B Gateway (B2BGW) Service for the trading partner.

The Multi-Protocol Gateway (MPG) service has a key role in this scenario, because it holds a policy rule for the transformation map. The synergic interaction between the B2B Gateway Service and the MPG is what makes this solution an extremely powerful use case for many trading scenarios.

We present a full picture of all the services and objects to implement in the scenario outline, where we also summarize what we will create.

11.4.1 Scenario outline

The scenario's architecture is presented in Figure 11-2 on page 230.

The left side of Figure 11-2 on page 230 corresponds to the simulated trading partner, which is referred to as the Banking partner, and the right side corresponds to the hub system, which is referred to as the Banking Hub. This architecture demonstrates that the DataPower device can fully accomplish both the partner and provider perspective.

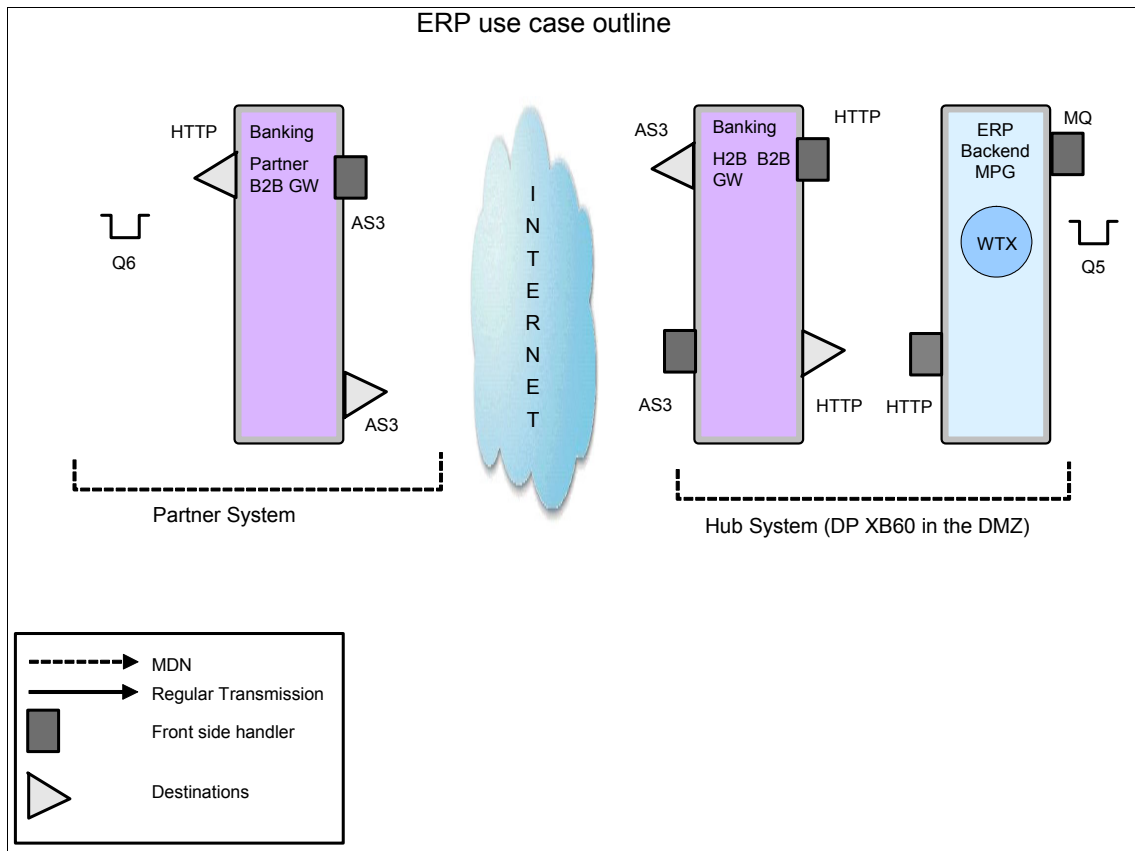


Figure 11-2 Using AS3 and transformation outline

As you can see in Figure 11-2, we use WebSphere MQ (WMQ) queues to put the messages in the Multi-Protocol Gateway service that will be responsible for running the WebSphere Transformation Extender (WTX) map, inside a processing rule that has a “binary transform” action in it.

The communication between both B2B Gateway Services will be made using AS3. In this case, we have only considered a one-way flow to keep the scenario as simple as possible.

These steps summarize what we need to do in order to implement the scenario:

- ▶ Step 1: Creating all the necessary crypto objects
- ▶ Step 2: Creating the Banking Partner profiles
- ▶ Step 3: Creating the Banking Hub profiles
- ▶ Step 4: Creating the Banking Partner B2B Gateway Service
- ▶ Step 5: Creating the Banking Hub B2B Gateway Service

- ▶ Step 6: Creating the Bank2Xml WTX map
- ▶ Step 7: Creating the ERP back-end system

11.4.2 Scenario implementation

Follow these steps to implement this scenario.

Step 1: Creating all the necessary crypto objects

This scenario is intended for people who are experienced with the DataPower device, so we do not explain in detail how to create the crypto objects (we assume that you have this knowledge), but it is important to point out that several objects are required for signing and validating signatures and encrypting and decrypting payloads.

The following objects have been created specifically for this scenario:

- ▶ Public/Private keys:
 - Bankingpartner key pairs
 - Bankinghub key pairs
- ▶ Validation credentials:
 - Banking_Partner_valcred: Used to validate signatures coming from the Banking Partner
 - Banking_Hub_valcred: Used to validate signatures coming from the Banking Hub
- ▶ Identification credentials:
 - Banking_Partner_IDcred: Used to sign messages from the Banking Partner system and to decrypt payloads coming from the Banking Hub
 - Banking_Hub_IDcred: Used to sign messages from the Banking Hub system and to decrypt payloads coming from the Banking Partner

Both ID Credentials and validation credential objects are configured with the corresponding key pair.

Step 2: Creating the Banking Partner profiles

A *Partner profile* is an object that defines the routing for messages by defining *Destinations*. It also establishes the AS Security rules when interchanging information with that specific partner.

Because we are simulating our trading within the DataPower device, we need two Banking Partner profiles: an internal profile for the B2B Gateway Service that will represent the partner back end and an external profile to be included in the

Banking Hub B2B Gateway Service where we set the destination pointing to the other B2B Gateway Service.

Banking Partner internal profile

The Banking Partner internal profile contains information, such as the type of AS Security that the partner expects, including how to sign the outgoing messages, decipher the incoming messages (AS Security tab), and where the messages are going to be routed (Destinations tab).

Here are the aspects of the configuration that you need to take into account in order to successfully configure this profile. We have named this object `Banking_Partner_int`, and we will use this methodology for the next objects as well.

First of all, we need to add Business IDs so that the B2BGW service can associate the incoming business ID in the AS3 headers to this partner profile and route it to its destination.

For this particular case, we have one Business ID, `bankpartner`, which is used in the XML that sent from our back end, and it is also used to enable us to route it to the partner destination.

Configure B2B Partner Profile

Main AS Security Destinations Contacts

B2B Partner Profile: Banking_Partner_int [up]

Apply Cancel Delete Undo

Admin State enabled disabled

Comments

Profile Type External Internal

Partner Business IDs

bankpartner Add

*

Figure 11-3 Banking Partner internal profile Main configuration tab

On the next tab (refer to Figure 11-4 on page 234), we configure AS Security aspects. Because this object is an internal profile, all the Identification Credentials are configured here. It is a requirement in this scenario to exchange AS3 messages signed and encrypted, so we check all the boxes and include the Banking Partner ID Credentials that we created in Step 1.

Note: Notice that ID Credentials contain the private key of the Banking Partner, which in this case is used for both signing and decrypting.

Configure B2B Partner Profile

Main AS Security Destinations Contacts

B2B Partner Profile: Banking_Partner_int [up]

Apply Cancel Delete Undo

Inbound Security

Require Signature

Require Encryption

Inbound Decryption Identification Credentials Banking_Partner_IDcred + ... *

Outbound Security

Sign Outbound Messages

Signing Identification Credentials Banking_Partner_IDcred + ... *

Signing Digest Algorithm sha1

Figure 11-4 Banking Partner internal profile AS Security configuration tab

The next step is to configure the Destination to where the documents from the Partner must be routed (refer to Figure 11-5 on page 235). As we can see in Figure 11-1 on page 228, the banking partner back end is simulated with WebSphere MQ. Therefore, an MQ destination is needed.

It is also important to notice that we only enable XML as the document type, because XML is the only type we will handle.

Configure B2B Partner Profile

Main AS Security **Destinations** Contacts

B2B Partner Profile: Banking_Partner_int [up]

Apply Cancel Delete Undo Export View L

Destinations

| Destination Name | Destination URL | Enabled Document Type |
|-------------------------|---------------------------|-----------------------|
| ERP_2_backend (default) | dpmq://QM1/?ReplyQueue=Q6 | XML |

Destinations

Destination Name: MQ_2_backend *

Enabled Document Type: XML, X12, EDIFACT, Binary

Connection

Destination URL: dpmq:// QM1/?ReplyQueue=Q6 *

Connection Timeout: 1800 Seconds

Apply Cancel

Figure 11-5 Banking Partner internal profile Destinations tab details

The Contact tab is an optional tab, which we do not use in this scenario. You can use it to enter the partner information if required.

Banking Partner external profile

The Banking Partner external profile (refer to Figure 11-6 on page 236) will handle the information that is needed by the provider in order to successfully trade with the Banking Partner (AS Security tab) along with the Partner destination point (Destinations tab).

Here are the aspects of configuration that you need to take into account in order to successfully configure this profile. We have named it Banking_Partner. In our naming convention, external partners do not have any indicator on the name.

From a Business ID perspective, we use the same configuration that we had on the internal profile. Figure 11-6 shows the configuration.

Configure B2B Partner Profile

Main AS Security Destinations Contacts

B2B Partner Profile: Banking_Partner [up]

Apply Cancel Delete Undo

Admin State enabled disabled

Comments

Profile Type External Internal

Partner Business IDs

bankpartner Add

*

Figure 11-6 Banking Partner external profile Main configuration tab

Because this profile is an external profile, from an AS3 Security perspective, validation credentials are required in order to verify the signed payload.

Again, we use the object `Banking_partner_valcred` that was created in Step 1. We do not add anything to the MDN Secure Sockets Layer (SSL) profile, because we do not use SSL for our transport layer.

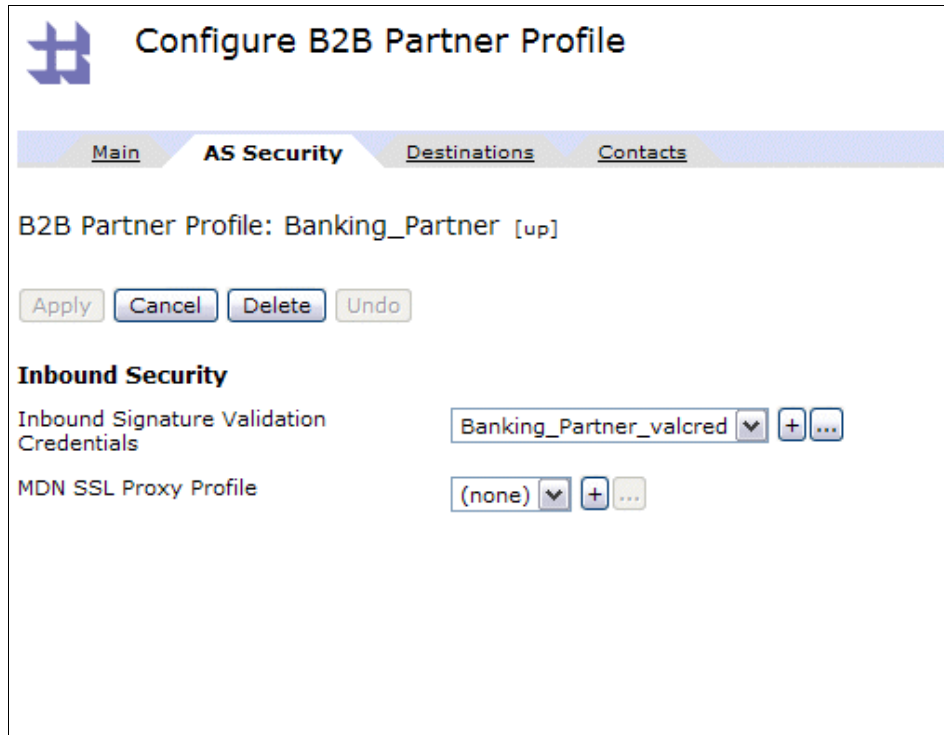


Figure 11-7 Banking Partner external profile AS Security configuration tab

In the Destinations tab, we add an AS3 destination, which is the information that describes how to route messages to the Partner system from the Hub system. We will communicate by using AS3.

Again, only XML is enabled as a valid document type.

Next, we add the URL of the Banking Partner's AS3 front side handler. Refer to Figure 11-8 on page 238.

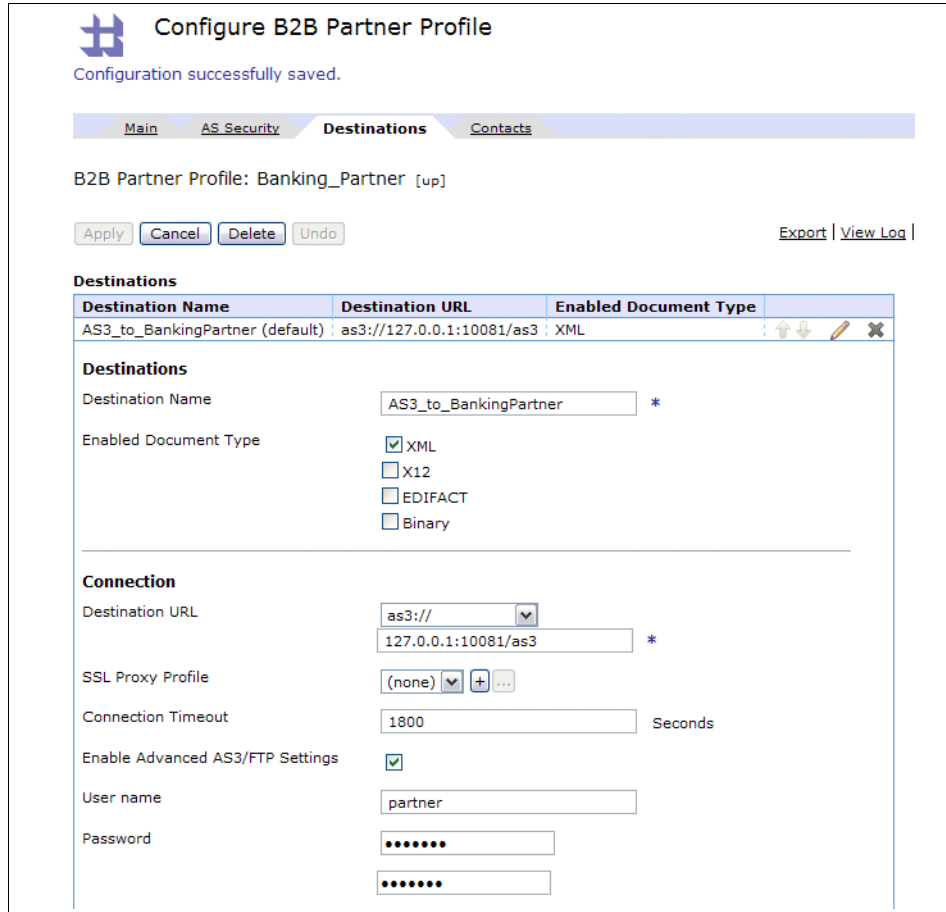


Figure 11-8 Banking Partner external profile Destination details

If we scroll down the window, we see AS Outbound security and Advanced AS Behavior details. Refer to Figure 11-9 on page 239.

In the Outbound security details, make sure that you select **Encrypt messages** and select the certificate that you generated in Step 1. Do not select “Send messages unsigned,” because we will sign our messages.

In the advanced AS3/FTP settings, leave everything as it is. We do not require any authentication in this case, because the Partner’s Front Side Handler will be configured without security.

However, it is worthwhile showing that more security can be added if it is a requirement.

| | |
|--|--|
| AS Outbound Security | |
| Send Messages Unsigned | <input type="checkbox"/> |
| Encrypt Messages | <input checked="" type="checkbox"/> |
| Encryption Certificate | banking_partner ▼ + ... * |
| Advanced AS Behavior | |
| Compress Messages | <input type="checkbox"/> |
| Request MDN | <input checked="" type="checkbox"/> |
| Time to Acknowledge | 1800 Seconds |
| AS3 MDN Redirection URL | ftp:// 127.0.0.1:10082/as3 |
| Request Signed MDN | <input checked="" type="checkbox"/> |
| Attempt Message Retransmission | <input checked="" type="checkbox"/> |
| Maximum Retransmissions | 3 |
| Advanced AS3/FTP Settings | |
| Passive Mode | Require Passive Mode ▼ |
| Encrypt Command Connection | No Authentication Requested ▼ |
| Data Type | Image (Binary) Data ▼ |
| Write Unique Filename if Trailing Slash | Request Unique File Name When Trailing Slash ▼ |
| Quoted Commands | (none) ▼ + ... |
| Size Check | Optional ▼ |
| <input type="button" value="Apply"/> <input type="button" value="Cancel"/> | |

*

Figure 11-9 Banking Partner external profile Destination details (continuation)

The Contact tab is an optional tab, and we do not use it in this case. However, you can use it to enter partner information if needed.

Step 3: Creating the Banking Hub profiles

As we explained in Step 2, we also need two Banking Hub profiles: an internal profile to associate with the Banking Hub B2B Gateway Service (refer to Step 5 for further details) and an external profile to associate with the Banking Partner B2B Gateway Service, which will enable our partner to be able to successfully send all the messages to the Banking hub, as well as validating the incoming signature, among other tasks.

Banking Hub internal profile

The Banking Hub internal profile manages the detail with regard to the type of AS Security that the provider expects (the AS Security tab), which includes how to sign the outgoing messages, how to decipher the incoming messages, and where the messages are routed (the Destinations tab). In this case, the messages are routed to the partner system.

Here are the aspects of the configuration that you need to take into account in order to successfully configure this profile. We have named it Banking_Hub_int to keep with our previous naming convention.

First, you configure the Business IDs, so that the B2BGW service can associate the incoming business ID in the AS3 headers to this partner profile and route it to its destination.

We enter our one Business ID, which is erpcustomer, which will come in the XML that is sent from our back end.

Figure 11-10 shows this configuration.

The screenshot shows the 'Configure B2B Partner Profile' web interface. At the top, there is a logo and the title 'Configure B2B Partner Profile'. Below the title is a navigation bar with four tabs: 'Main', 'AS Security', 'Destinations', and 'Contacts'. The 'Main' tab is currently selected. The main content area displays the profile name 'B2B Partner Profile: Banking_Hub_int [up]'. Below the profile name are four buttons: 'Apply', 'Cancel', 'Delete', and 'Undo'. The 'Admin State' is set to 'enabled' with a radio button. The 'Comments' field is empty. The 'Profile Type' is set to 'Internal' with a radio button. The 'Partner Business IDs' field contains the text 'erpcustomer' and has an 'Add' button next to it. A small asterisk is visible below the 'Partner Business IDs' field.

Figure 11-10 Banking Hub internal profile Main configuration tab

Next, we complete the AS Security tab information (Figure 11-11).

Because this profile is an internal profile, all the identification credentials are configured here. It is a requirement in this scenario to exchange signed and encrypted AS3 messages, so we must select **Require Signature** and **Require Encryption**. Include the Banking Partner ID Credentials that we created in Step 1. Notice that the ID Credentials contain the private key of the Banking Partner that, in this case, is used for both signing and decrypting.

The screenshot shows a configuration window titled "Configure B2B Partner Profile" with a navigation bar containing "Main", "AS Security", "Destinations", and "Contacts". The "AS Security" tab is active. Below the navigation bar, the profile name is "Banking_Hub_int" with an up arrow icon. There are four buttons: "Apply", "Cancel", "Delete", and "Undo". A red exclamation mark is visible on the right side. The "Inbound Security" section includes three items: "Require Signature" (checked), "Require Encryption" (checked), and "Inbound Decryption Identification Credentials" (set to "Banking_Hub_IDcred" with a dropdown arrow, a plus sign, an ellipsis, and an asterisk). The "Outbound Security" section includes three items: "Sign Outbound Messages" (checked), "Signing Identification Credentials" (set to "Banking_Hub_IDcred" with a dropdown arrow, a plus sign, an ellipsis, and an asterisk), and "Signing Digest Algorithm" (set to "sha1" with a dropdown arrow).

Figure 11-11 Banking hub internal profile AS Security configuration tab

Next, we configure the Destination from where the documents come. As we can see in Figure 11-2 on page 230, which shows the main architecture, the Banking Hub system has an Multi-Protocol Gateway (MPG) service, which transforms the data. The connection is done with HTTP. Therefore, an HTTP destination is needed.

Note that we have only enabled XML as the document type, which is the only type we will handle.

Important: Notice that, when testing this scenario, we do not use this destination, because we only have one flow. However, it is worth showing in case you want to expand the scenario configuration.

Configure B2B Partner Profile
Configuration successfully saved.

Main AS Security **Destinations** Contacts

B2B Partner Profile: Banking_Hub_int [up]

Apply Cancel Delete Undo Export View Log View

Destinations

| Destination Name | Destination URL | Enabled Document Type | |
|------------------------------|------------------------|-----------------------|---------|
| http_2_ERP_backend (default) | http://127.0.0.1:10084 | XML | ↑ ↓ ✎ ✕ |

Destinations

Destination Name *

Enabled Document Type

- XML
- X12
- EDIFACT
- Binary

Connection

Destination URL *

Connection Timeout Seconds

User name

Password

Apply Cancel

Figure 11-12 Banking hub internal profile Destinations tab details

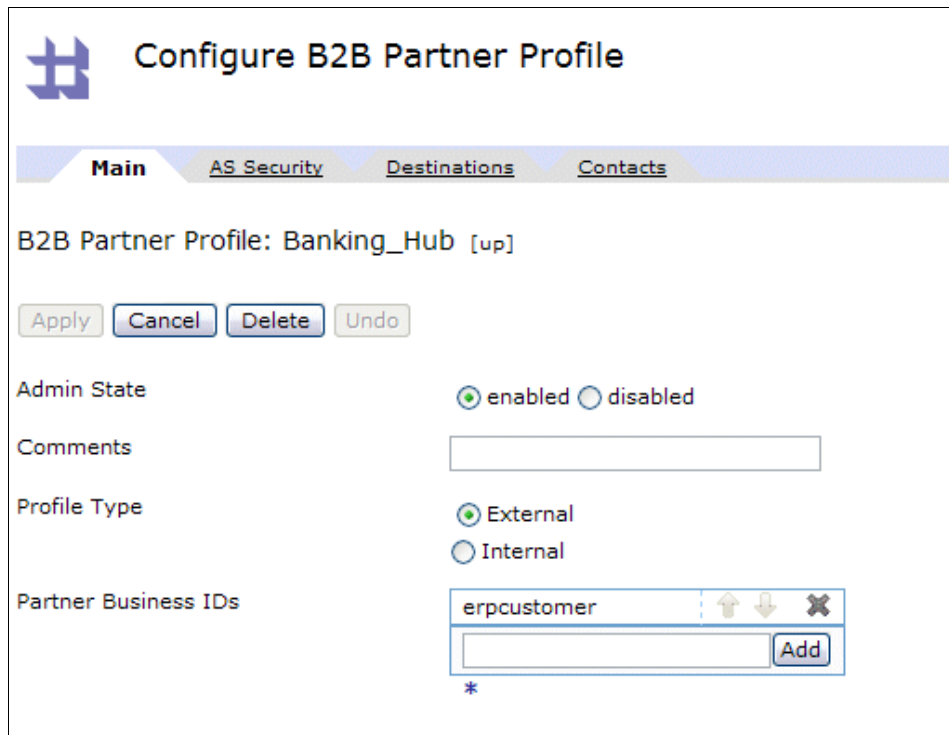
The Contact tab is an optional tab that we do not use in this case; however, enter the partner information if needed.

Banking Hub external profile

The Banking Hub external profile manages the information details, such as what is needed by the partner in order to successfully trade with the Banking Hub (AS Security tab) and where the Provider destination point is located (Destinations tab).

Here are the parts of the configuration that you need to consider in order to successfully configure this profile. We have named it `Banking_Hub`. In our naming convention, external partners do not have any indicator on the name, and everyone chooses their own naming convention.

From a Business ID perspective, we create the same configuration that we had on the internal profile. Figure 11-13 shows the configuration.



The screenshot shows the 'Configure B2B Partner Profile' interface with the 'Main' tab selected. The profile name is 'Banking_Hub'. Below the name are buttons for 'Apply', 'Cancel', 'Delete', and 'Undo'. The 'Admin State' is set to 'enabled'. The 'Comments' field is empty. The 'Profile Type' is set to 'External'. The 'Partner Business IDs' list contains 'erpcustomer' and an 'Add' button. A small asterisk is visible below the list.

Figure 11-13 Banking Hub external profile Main configuration tab

Because this profile is an external profile, from an AS3 Security perspective, only good validation credentials are required in order to verify the signed payload. Again, we use the object `Banking_hub_valcred` that we created in Step 1. We do not add anything to the MDN SSL profile, because we do not use SSL for our transport layer.

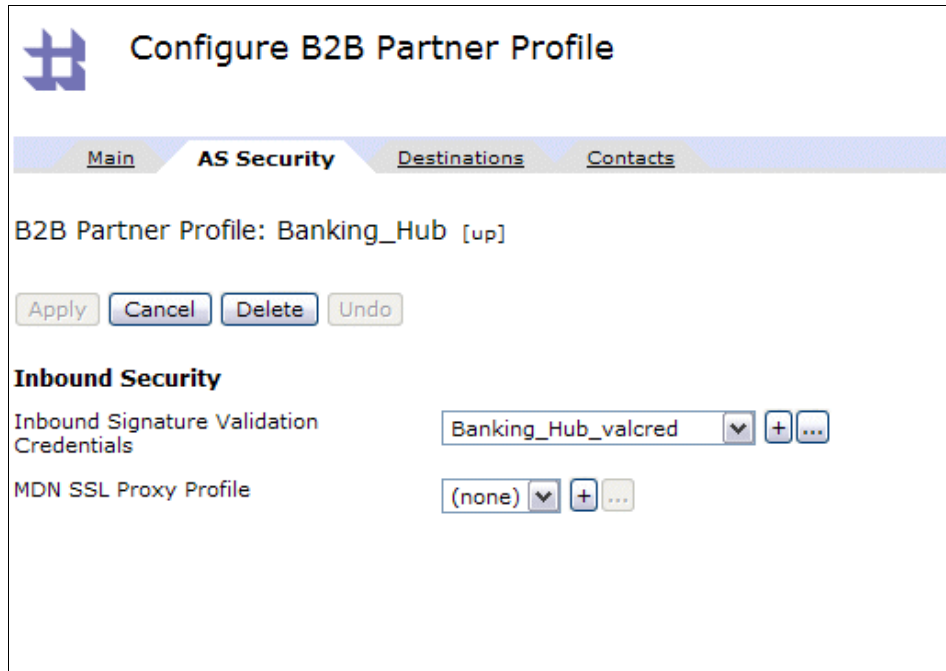


Figure 11-14 Banking Hub external profile AS Security configuration tab

In the Destinations tab, we add an AS3 destination that is the information that describes how to route messages to the Partner system from the Hub system, which we communicate using AS3.

Only XML is enabled as a valid document type. Figure 11-15 on page 245 shows how to add the URL of the Banking partner's AS3 Front side handler.

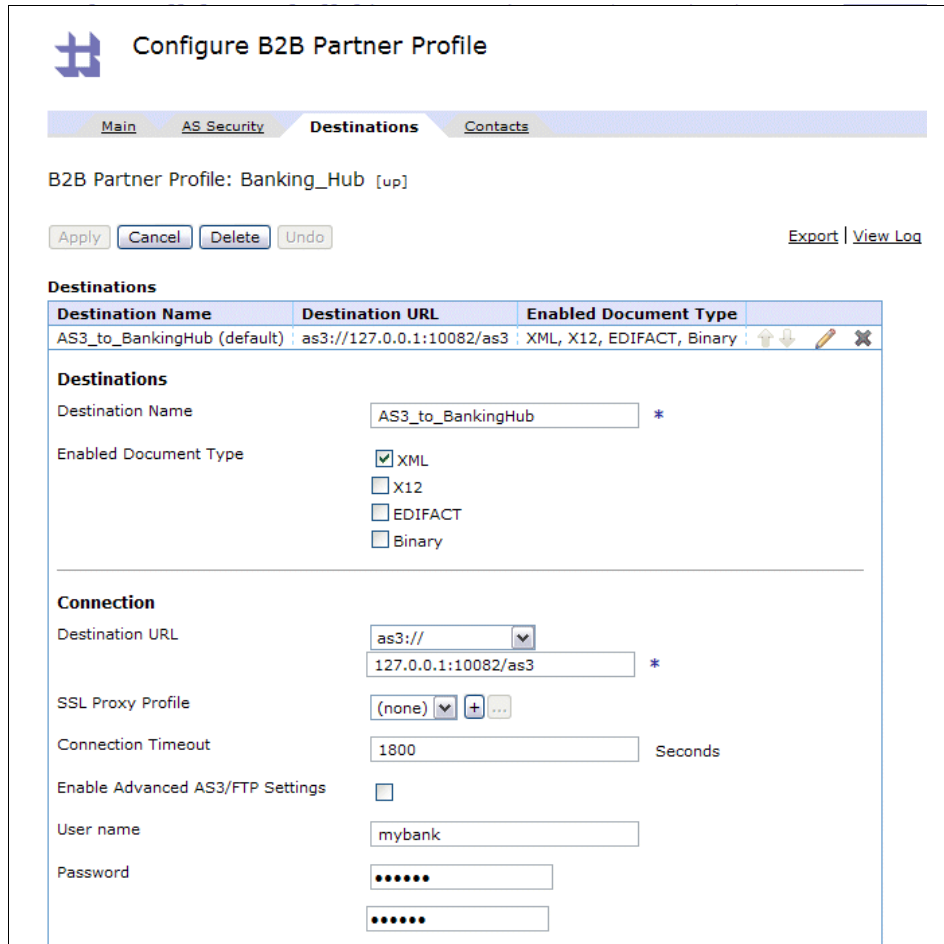


Figure 11-15 Banking Hub external profile Destinations tab details

If we scroll down the window (Figure 11-16 on page 246), we see AS Outbound security and Advanced AS Behavior sections.

In the Outbound security section, make sure that you select **Encrypt messages** and select the certificate that you generated in Step 1. Do not select “Send messages unsigned,” because our messages are signed.

In the advanced AS3/FTP settings, leave everything as it is. We do not require any authentication in this case, because the Banking Hub’s Front Side Handler will be configured without security. However, it is worth showing that more security can be added if more security is a requirement.

| AS Outbound Security | |
|--|-------------------------------------|
| Send Messages Unsigned | <input type="checkbox"/> |
| Encrypt Messages | <input checked="" type="checkbox"/> |
| Encryption Certificate | banking_hub ▼ + ... * |
| Advanced AS Behavior | |
| Compress Messages | <input type="checkbox"/> |
| Request MDN | <input checked="" type="checkbox"/> |
| Time to Acknowledge | 1800 Seconds |
| AS3 MDN Redirection URL | ftp:// 127.0.0.1:10082/as3 |
| Request Signed MDN | <input checked="" type="checkbox"/> |
| Attempt Message Retransmission | <input checked="" type="checkbox"/> |
| Maximum Retransmissions | 3 |
| <input type="button" value="Apply"/> <input type="button" value="Cancel"/> | |

Figure 11-16 Banking Hub external profile Destinations tab details (continuation)

The Contact tab is an optional tab, which we do not configure in this case; however, feel free to enter the partner information if needed.

Step 4: Creating the Banking Partner B2B Gateway Service

Having created all the profiles for the scenario, it is time to create the B2B Gateway Services that will handle all the traffic and attach the profiles, depending on the case. For the Banking Partner, this service acts as the entry point to their system, identifying the incoming messages and mapping the incoming messages with the specific profiles that need to handle them. In Figure 11-17 on page 247, we see the Banking Partner B2BGW service has two partners attached, the Banking Partner internal profile and the Banking Hub external profile, but you can add as many profiles as your business needs require.

We have two Front Side Handler objects, which are used in all the DataPower services, to receive incoming traffic. For this particular case, an AS3 FSH is needed to receive messages coming from the Banking Hub and an HTTP FSH is needed to receive messages from the Partner back end.

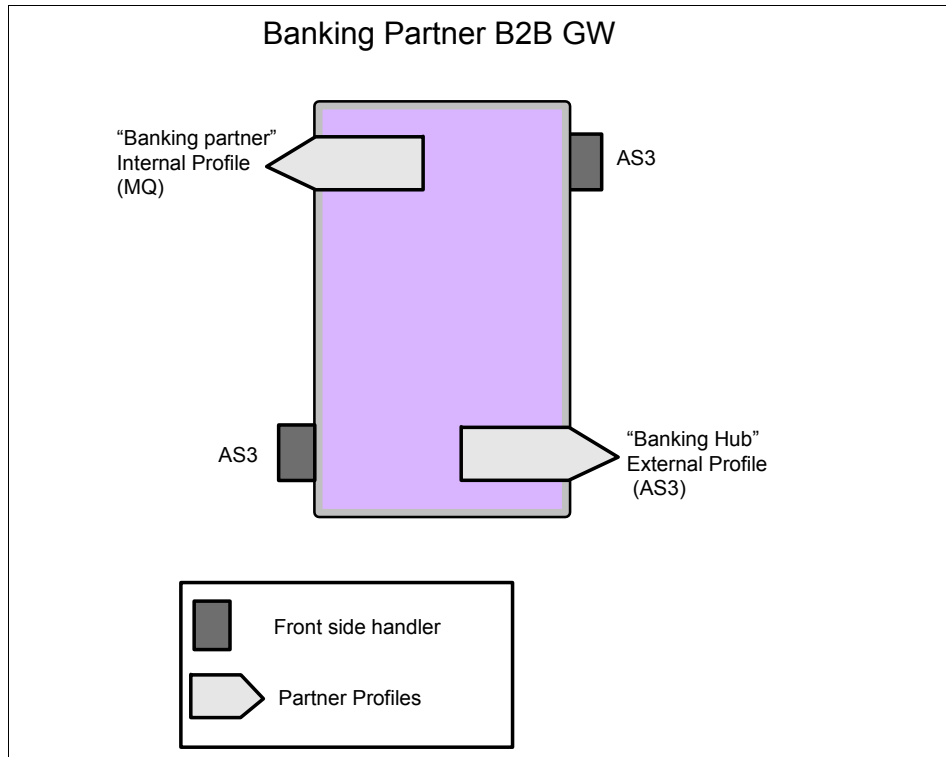


Figure 11-17 Banking Partner B2B Gateway architecture

Figure 11-18 on page 248 shows the Main tab window for the Banking Partner B2B Gateway service.



Configure B2B Gateway

Main

Archive

XML Formats

Advanced

B2B Gateway: Banking_Partner_B2BGW [up]

Apply Cancel Delete Undo

Export | View Log | View Status | Archive/purge

General Configuration

Admin State enabled disabled

Comments

Document Storage Location ▼

XML Manager ▼ + ... *

Document Routing

Front Side Protocol Handlers

| Front Side Protocol | |
|------------------------|-------|
| AS3_BankingPartner_FSH | ↑ ↓ ✕ |
| <input type="text"/> | + |

*

Attach Partner Profiles

Active Partner Profiles

| B2B Partner Profile | Profile Enabled? | Profile Destination | |
|--|------------------|---------------------|-------|
| Banking_Partner_int | enabled ▼ | http_2_backend ▼ | ↑ ↓ ✕ |
| Banking_Hub | enabled ▼ | AS3_to_BankingHub ▼ | ↑ ↓ ✕ |
| <input type="text" value="Banking_Hub"/> | ▼ | + ... | Add |

*

Active Profile Groups

| B2B Profile Group | Group Enabled? |
|----------------------|----------------|
| (empty) | ▼ |
| <input type="text"/> | + ... |

Figure 11-18 Banking Partner B2BGW Main configuration tab

As you can see, we have only one AS3 Front Side Handler attached in the Front Side Handlers tab. There will be no communication from the partner back end to the Banking Partner B2BGW, because this scenario is a one-way flow.

We do however have two active Partner Profiles (Banking_Partner_int and Banking_hub) attached in the Attach Partner Profiles section. If you want to add a new Partner profile, select the partner profile that you want from the drop-down list and click **Add**; then, select the Profile Destination.

Figure 11-19 shows the detailed configuration of the AS3 FSH.

The screenshot displays the configuration interface for the AS3 Front Side Handler (FSH) named 'AS3_BankingPartner_FSH'. The interface includes a navigation bar with 'Main' and 'Virtual Directories' tabs. Below the title, there are control buttons for 'Apply', 'Cancel', and 'Undo', along with utility links for 'Export', 'View Log', 'View Status', and 'Help'. The configuration is organized into several sections:

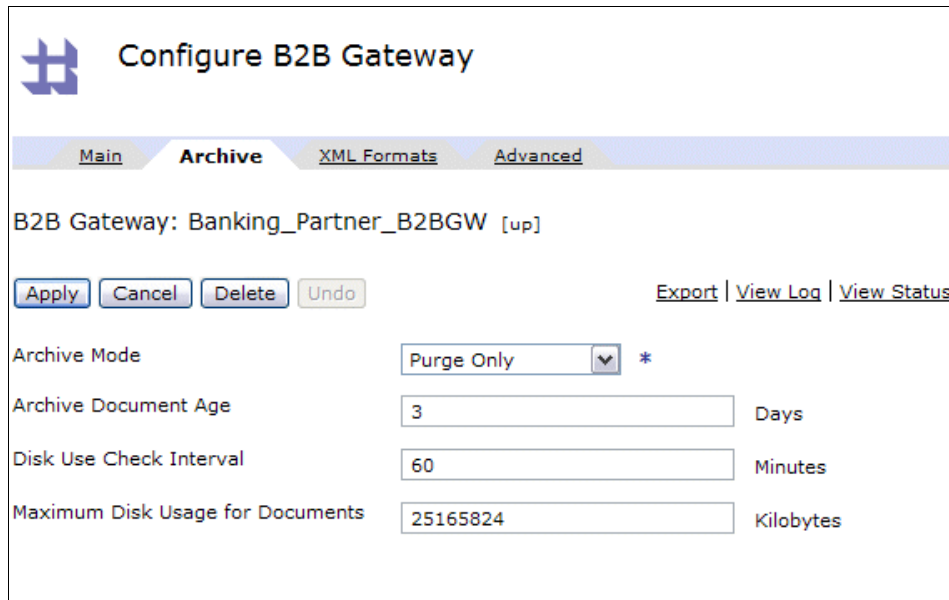
- Admin State:** Radio buttons for 'enabled' (selected) and 'disabled'.
- Comments:** An empty text input field.
- Local IP Address:** A text input field containing 'localhost' and a 'Select Alias' button.
- Port Number:** A text input field containing '10081'.
- Filesystem Type:** A dropdown menu set to 'Virtual Ephemeral'.
- Default Directory:** A text input field containing '/'.
- Maximum Filename Length:** A text input field containing '256'.
- Access Control List:** A dropdown menu set to '(none)' with '+' and '...' buttons.
- Require TLS:** Radio buttons for 'on' and 'off' (selected).
- SSL Proxy:** A dropdown menu set to '(none)' with '+' and '...' buttons.
- Password AAA Policy:** A dropdown menu set to '(none)' with '+' and '...' buttons.
- Certificate AAA Policy:** A dropdown menu set to '(none)' with '+' and '...' buttons.
- Allow CCC Command:** Radio buttons for 'on' (selected) and 'off'.
- Passive (PASV) Command:** A dropdown menu set to 'Allow Passive Mode'.
- Limit Port Range for Passive Connections:** Radio buttons for 'on' and 'off' (selected).
- Passive Data Connection Idle Timeout:** A text input field containing '60' followed by the unit 'seconds'.
- File Transfer Data Encryption:** A dropdown menu set to 'Allow Data Encryption'.
- Allow Compression:** Radio buttons for 'on' (selected) and 'off'.
- Allow Unique File Name (STOU):** Radio buttons for 'on' and 'off' (selected).

Figure 11-19 Banking Partner AS3 FSH

Note that we have left everything as default, because our scenario has no special requirements for security.

We use localhost as a Host Alias for 127.0.0.1 in order to not tie our Front Side Handler with any specific IP address and to limit the incoming traffic only to the IP address coming from the device internally. In a real scenario, the host alias is an external IP address.

On the Archive configuration tab (Figure 11-20), we select **Purge Only** and enter 3 days for the Archive Document Age, even though, depending on the use case, this property will need to be resized later.



The screenshot displays the 'Configure B2B Gateway' interface. At the top, there is a navigation bar with tabs for 'Main', 'Archive', 'XML Formats', and 'Advanced'. The 'Archive' tab is selected. Below the navigation bar, the gateway name 'Banking_Partner_B2BGW [up]' is shown. A row of buttons includes 'Apply', 'Cancel', 'Delete', and 'Undo'. To the right of these buttons are links for 'Export', 'View Log', and 'View Status'. The main configuration area contains four rows of settings:

| | | |
|----------------------------------|------------|-----------|
| Archive Mode | Purge Only | * |
| Archive Document Age | 3 | Days |
| Disk Use Check Interval | 60 | Minutes |
| Maximum Disk Usage for Documents | 25165824 | Kilobytes |

Figure 11-20 Banking Partner B2BGW Archive configuration tab

We must now configure the XML format (Figure 11-21 on page 251), which allows us to identify all the core routing information from inside the XML message, which includes Sender, Receiver, Document ID, and Timestamp. Click + (the plus sign) to create a new routing policy. We have already created a routing policy called Banking_XMLFormat.

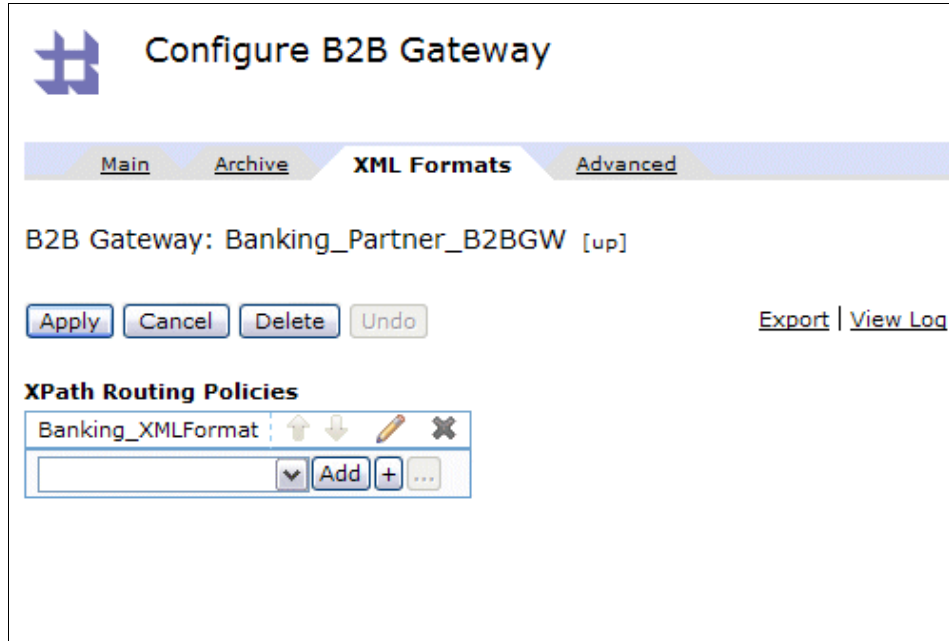


Figure 11-21 Banking Partner B2BGW XML Formats tab

With the help of the XPath tool (refer to Scenario 1 in Appendix A, “Additional material” on page 389 for more details about how to use it), we upload our sample incoming XML and identify the required fields explained earlier.

For our specific case, these fields are located in tags whose names are Sender_ID, Receiver_ID, FileReference, and Date.

Example 11-1 Header detail of the incoming XML data

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tns:PaymentsList xmlns:tns="http://www.example.org/My_Payment_Format">
<tns:Sender_ID>erpcustomer</tns:Sender_ID>
<tns:Receiver_ID>bankpartner</tns:Receiver_ID>
<tns:FileReference>1138</tns:FileReference>
<tns:Account_Debit_Number>456158353710</tns:Account_Debit_Number>
<tns:Payment_Lot>N1015</tns:Payment_Lot>
<tns:Date>2007-10-26</tns:Date>
```

After uploading the XML file and selecting each field in the XPath tool, you will have a result similar to Figure 11-22 on page 252.

Configure B2B XPath Routing Policy

Main

B2B XPath Routing Policy: Banking_XMLFormat [up]

Apply Cancel Undo

Admin State enabled disabled

Sender XPath XPath Tool *

Receiver XPath XPath Tool *

Document ID XPath XPath Tool

Transaction Timestamp XPath XPath Tool

Figure 11-22 Banking Partner B2BGW XML Formats configuration details

In the Advanced configuration tab (Figure 11-23 on page 253), we need to set the default AS3 MDN Return Path. This information will be in the AS3 headers when the partner sends any AS3 messages to the Hub, so that the Hub knows where, by default, to send the As3 MDNs back. For this reason, we indicate the port number, which matches the Partner's AS3 Front Side Handler.

Configure B2B Gateway

Main | Archive | XML Formats | **Advanced**

B2B Gateway: Banking_Partner_B2BGW [up]

Apply | Cancel | Delete | Undo | Export | View Log | View Status | Ar

Service Priority: Normal

Default AS2 MDN Return Path: http://

Default AS3 MDN Return Path: ftp:// localhost:10081/as3

Document Routing Preprocessor: store:/// b2b-routing.xsl | Upload... | Fetch...

Figure 11-23 Banking Partner B2BGW Advanced tab

We have now finished configuring the Banking Partner B2B Gateway.

Step 5: Creating the Banking Hub B2B Gateway Service

For the Banking Hub case, our B2B Gateway Service acts as the entry point to the hub system, identifying the incoming messages and mapping them with the specific profiles that need to handle them. As we can see in Figure 11-24 on page 254, the Banking Hub B2BGW service has two partners attached: the Banking Hub internal profile and the Banking Partner external profile, but you can add as many profiles as you need.

We have two Front Side Handlers, which are used in all DataPower services, to receive incoming traffic.

In contrast with the previous Partner B2B Gateway, in this particular case, we still need two FSHs: an AS3 FSH is needed for receiving MDNs (and only MDNs because we only have a one-way flow) coming from the Banking Partner and an HTTP to receive messages from the Multi-Protocol Gateway ERP_backend_MPG.

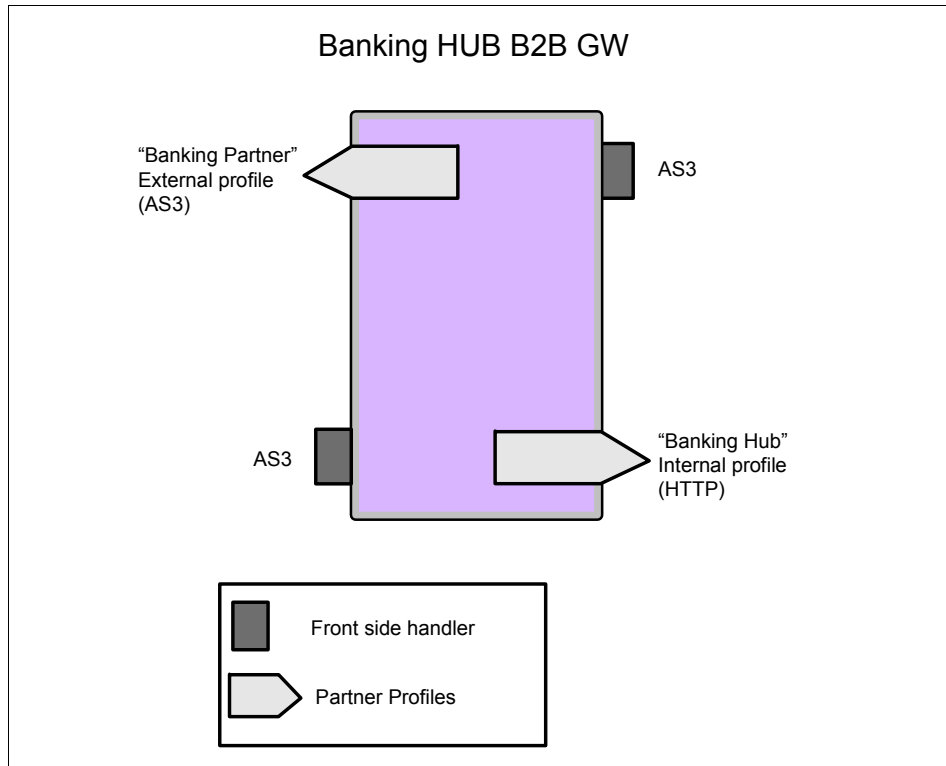


Figure 11-24 Banking Hub B2B Gateway architecture

The B2B Gateway Service Main configuration tab window is shown in Figure 11-25 on page 255.

Configure B2B Gateway

Main Archive XML Formats Advanced

B2B Gateway: Banking_Hub_B2BGW [up]

Apply Cancel Delete Undo Export View Log View Status Archive/purge transacti

General Configuration

Admin State enabled disabled

Comments

Document Storage Location local:///

XML Manager default + ... *

Document Routing

Front Side Protocol Handlers

| Front Side Protocol | | | |
|----------------------|---|---|---|
| Banking_Hub_httpFSH | ↑ | ↓ | ✕ |
| AS3_BankingHub_FSH | ↑ | ↓ | ✕ |
| <input type="text"/> | | | |

Add *

Attach Partner Profiles

Active Partner Profiles

| B2B Partner Profile | Profile Enabled? | Profile Destination | |
|----------------------|------------------|-----------------------|-------|
| Banking_Hub_int | enabled | (default) | ↑ ↓ ✕ |
| Banking_Partner | enabled | AS3_to_BankingPartner | ↑ ↓ ✕ |
| <input type="text"/> | | | |

Banking_Hub + ... Add *

Active Profile Groups

| B2B Profile Group | Group Enabled? |
|----------------------|----------------|
| (empty) | |
| <input type="text"/> | |

+ ... Add

Figure 11-25 Banking Hub B2BGW Main configuration tab

We have two active partner profiles (Banking_Partner and Banking_Hub_int) attached in the Attach Partner Profiles section. If you want to add a new partner profile, select a partner from the drop-down list and click **Add**; then, select the Profile Destination.

Figure 11-26 on page 256 shows the detailed configuration of the AS3 FSH.

Configure AS3 Front Side Handler

Main Virtual Directories

AS3 Front Side Handler: AS3_BankingHub_FSH [up]

Apply Cancel Undo [Export](#)

Admin State enabled disabled

Comments

Local IP Address [Select Alias](#) *

Port Number *

Filesystem Type ▼

Default Directory

Maximum Filename Length

Access Control List ▼ [+](#) [...](#)

Require TLS on off

SSL Proxy ▼ [+](#) [...](#)

Password AAA Policy ▼ [+](#) [...](#)

Certificate AAA Policy ▼ [+](#) [...](#)

Allow CCC Command on off

Passive (PASV) Command ▼

Limit Port Range for Passive Connections on off

Passive Data Connection Idle Timeout seconds

Figure 11-26 Banking Hub AS3 FSH

Note that we have left everything as default, because there are no special configurations needed for security for our use case.

It is important to mention that we are using localhost as a Host Alias for 127.0.0.1 in order to not tie our Front Side Handler with any specific IP address and to limit the incoming traffic only to the IP address coming from the device internally. In a real scenario, the host alias is an external IP address.

The Archive configuration tab (Figure 11-27) is set up as **Purge only** and we entered 3 days of Archive Document Age. Depending on the use case, you will need to resize this property later.

The screenshot shows the 'Configure B2B Gateway' window with the 'Archive' tab selected. The gateway name is 'Banking_Hub_B2BGW'. The configuration fields are as follows:

| Property | Value | Unit |
|----------------------------------|------------|-----------|
| Archive Mode | Purge Only | * |
| Archive Document Age | 3 | Days |
| Disk Use Check Interval | 60 | Minutes |
| Maximum Disk Usage for Documents | 25165824 | Kilobytes |

Figure 11-27 Banking Hub B2BGW Archive configuration tab

Now, it is time to configure the XML format (Figure 11-28 on page 258) that allows us to identify where we perform all the core routing. These fields include Sender, Receiver, Document ID, and Timestamp. We have the same type of messages in both gateways, so we can use the same XML format that one we created before in Banking Partner B2B Gateway. Choose that XML format in the drop-down list and click **Add**.

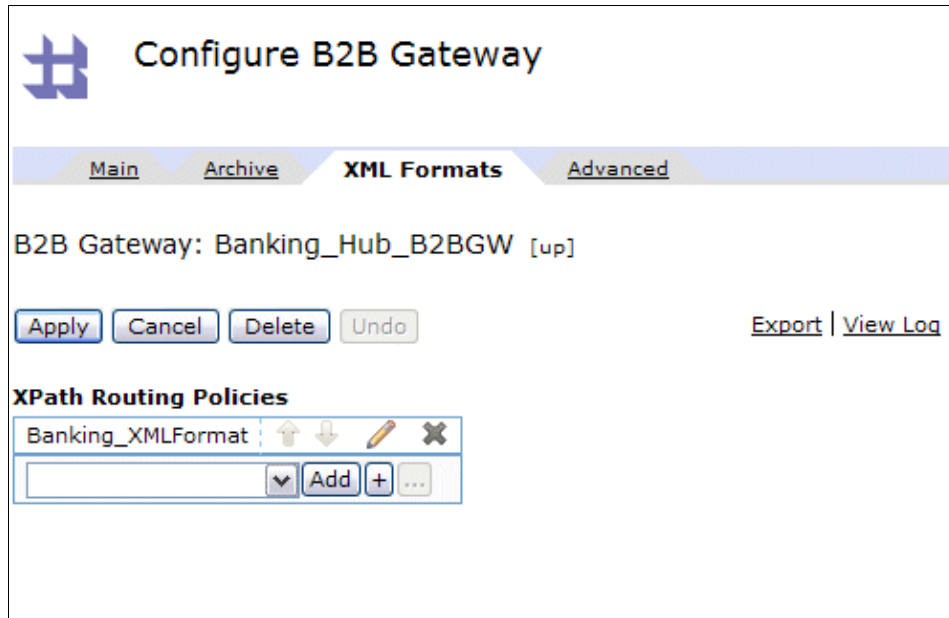


Figure 11-28 Banking Partner B2BGW XML Formats configuration tab

In the Advanced configuration tab (Figure 11-29 on page 259), we need to set the default AS3 MDN Return Path. This information is used for the AS3 return MDN path in the outbound message headers when a return path value is not set in the Partner's AS3 destination.

Configure B2B Gateway

Main Archive XML Formats **Advanced**

B2B Gateway: Banking_Hub_B2BGW [up]

Apply Cancel Delete Undo Export View Log View Status

Service Priority Normal

Default AS2 MDN Return Path http://

Default AS3 MDN Return Path ftp:// localhost:10081

Document Routing Preprocessor store:/// b2b-routing.xsl Upload...

Figure 11-29 Banking Hub B2BGW Advanced configuration tab

We have successfully created the Banking Hub B2B Gateway.

Step 6: Creating the Bank2Xml WTX map

Important: We do not describe field-level detail for the mapping specification, because this detail can represent a whole scenario by itself, but we describe the major guidelines on the map structure for you to have an idea of what kind of transformation this map performs. For more information about how to use WebSphere Transformation Extender, refer to *IBM WebSphere Transformation Extender* at:

<http://www.redbooks.ibm.com/abstracts/SG247693.html?Open>

The transformation map ERP2XML uses one input type tree that represents the Flat File coming from the ERP system. We built this input type tree specifically for this scenario. From a structure standpoint, it has one Header record (initiated by 01) that contains all the information to describe the whole payment lot and then a set of separate Detail records, each of which matches a single payment.

Example 11-2 on page 260 shows a sample of the Input file.

Example 11-2 Input flat file coming from the back end

```
01;1138;26.10.07;X1015;899999068-1;DKLBD;456158353710;01;00020;000000000022463959.73
02;06701;1040098071;01;SOME GREAT COMPANY;8400510218;06;CL 114 No 9-01
OF.706;PARLADOIRO
CALVELLE;CO;1-6181645;fa@fa.com;000000000000261271.04;EUR;300;26.10.08;26.10.08;;;200
0046888;;067011456;;;SOME BANK;;GZVIGO
02;02100;36195698;01;troita de pe;8002069302;06;KR 7 No.71-52 TO B OF.1403;OLEIROS A
CORUNHA;1-3170946;troita@depe.com.gz;00000000000034861.00;EUR;300;27.10.07;27.10.07;
;;2000046899;;021000089;;IBM BANK;;USNEW YORK
```

The output type tree is an XML Schema that the Banking Partner has provided so that we can transform our data into the XML that the Banking Partner expects. It is structured in a root tag called `PaymentFile` that contains a set of tags that contain the main information about the document (sender, receiver, and so forth) and then several `Payment file` tags, each of which contains the payment specification for each of the payments coming in the input file.

Example 11-3 shows a sample of the Output file.

Example 11-3 Output XML that is sent to the trading partner

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tns:PaymentsList
  xmlns:tns="http://www.example.org/My_Payment_Format">
  <tns:Sender_ID>erpcustomer</tns:Sender_ID>
  <tns:Receiver_ID>bankpartner</tns:Receiver_ID>
  <tns:FileReference>1138</tns:FileReference>

  <tns:Account_Debit_Number>456158353710</tns:Account_Debit_Number>
  <tns:Payment_Lot>X1015</tns:Payment_Lot>
  <tns>Date>2007-10-26</tns>Date>
  <tns:SinglePayment>
    <tns:PaymentDate>2008-10-26</tns:PaymentDate>
    <tns:BeneficiaryAccountNumber>
      1040098071
    </tns:BeneficiaryAccountNumber>

    <tns:BeneficiarySWIFT_Code>067011456</tns:BeneficiarySWIFT_Code>
    <tns:AccountType>1</tns:AccountType>
```

```

<tns:Currency>EUR</tns:Currency>
<tns:Amount>261271</tns:Amount>
<tns:CustomerData>
  <tns:CustomerIdentifier>8400510218</tns:CustomerIdentifier>
  <tns:Address>CL 114 No 9-01 OF.706</tns:Address>
  <tns:City>PARLADOIRO CALVELLE</tns:City>
  <tns:Country>GZ</tns:Country>
  <tns:Email>fa@fa.com</tns:Email>
  <tns:Phone>1-6181645</tns:Phone>
</tns:CustomerData>
</tns:SinglePayment>

```

The key concept, from a mapping perspective, is that all the information that comes in the Detail records (initiator 02) must be mapped into Single Payment tags.

Figure 11-30 shows the logic of both Functional Maps, including the Input and Output cards.

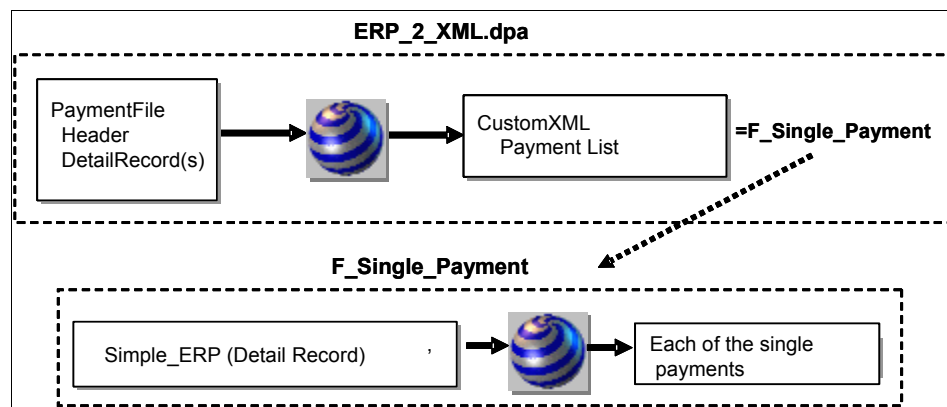


Figure 11-30 Functional map for mapping each Single payment

When you build your map in WebSphere Transformation Extender Design Studio, make sure that you use WebSphere DataPower as the map Runtime mode. You can configure this in Map Settings.

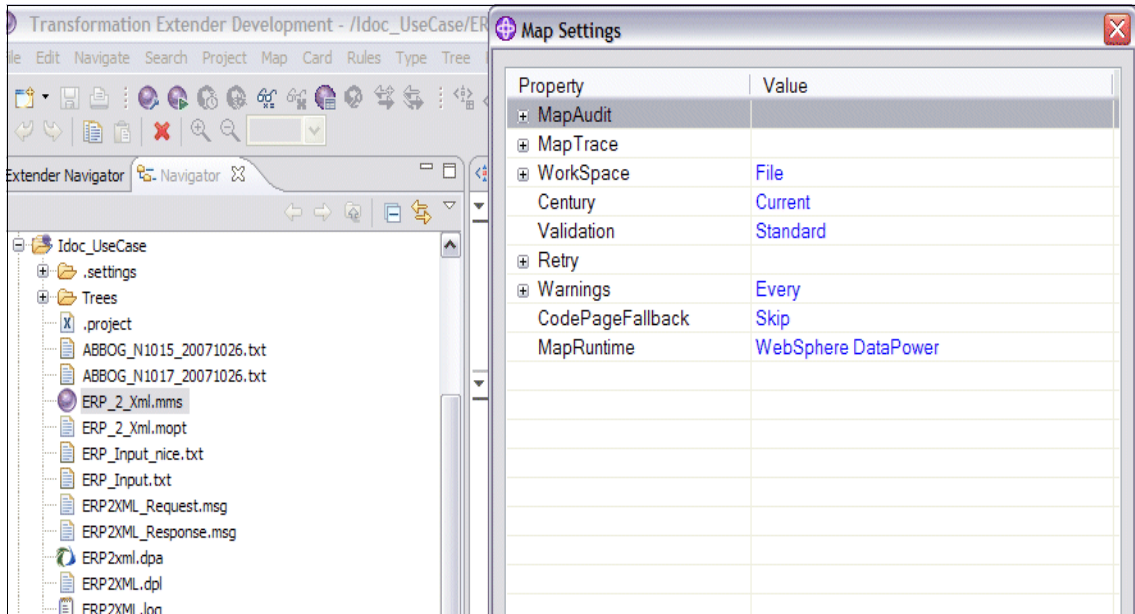


Figure 11-31 Map Settings configuration

After building the map, you have a .dpa extension on your workspace, which is the compiled map that must be uploaded into the DataPower device for the transformation action.

Make sure that you also test the map locally, using **Run locally**, so that you are sure that everything works as you expect. At this point, we are ready to create the MPG that will host the transformation.

Step 7: Creating the ERP back-end system

In order to configure the MPG (Figure 11-32 on page 263), we will need an MQ FSH, which is called ERP_MQ_FSH that will listen on Q5 from our queue manager QM, and a back-end URI that will be the Banking hub B2BGW (port 10080).

The screenshot displays the 'Configure Multi-Protocol Gateway' interface. At the top, there is a navigation bar with tabs: 'General' (selected), 'Advanced', 'Stylesheet Params', 'Headers', 'Monitors', 'WS-Addressing', and 'WS-ReliableMessaging'. Below the navigation bar are buttons for 'Apply', 'Cancel', and 'Delete', and a set of utility links: 'Export', 'View Log', 'View Status', 'Show Probe', 'Validate Conformance', and 'Help'. The main content area shows the 'General Configuration' for a gateway named 'ERP_Backend_MPG'. The 'Multi-Protocol Gateway Name' field contains 'ERP_Backend_MPG'. The 'Summary' field contains 'MPG that transforms using WTX n'. The 'Type' section has two radio buttons: 'dynamic-backend' (unselected) and 'static-backend' (selected). The 'XML Manager' dropdown is set to 'default'. The 'Multi-Protocol Gateway Policy' dropdown is set to 'ERP_Transformation_Policy'. The 'URL Rewrite Policy' dropdown is set to '(none)'. Below this, there are two sections: 'Back side settings' and 'Front side settings'. The 'Back side settings' section includes a 'Backend URL' field with 'http://localhost:10080' and four buttons: 'MQHelper', 'TibcoEMSHelper', 'WebSphereJMSHelper', and 'IMSConnectHelper'. The 'Front side settings' section includes a 'Front Side Protocol' dropdown set to 'ERP_MQ_FSH (MQ Front Side Handler)' and an 'Add' button.

Figure 11-32 ERP_Backend General configuration tab

In Figure 11-33 on page 264, make sure that you select **Non-XML** as the Request and Response type, because we will send non-XML payload to it.

SSL Client Crypto Profile
 (none) [v] [+] [...]

Response Type
 Non-XML
 Pass-Thru
 SOAP
 XML

Request Type
 Non-XML
 Pass-Thru
 SOAP
 XML

Back attachment processing format
 Dynamic
 MIME
 DIME
 Detect

Front attachment processing format
 Dynamic
 MIME
 DIME
 Detect

Back Side Timeout
 *

Front Side Timeout
 *

Stream Output to Back
 Buffer Messages
 Stream Messages

Stream Output to Front
 Buffer Messages
 Stream Messages

HTTP Version to Server
 HTTP 1.0
 HTTP 1.1

Propagate URI
 on off

Compression
 on off

Figure 11-33 ERP_Backend General configuration tab (continuation)

If we examine the policy that we have created in Figure 11-34 on page 265 (click + to create a new policy in DataPower), we can see that it only has a simple **Match all**, based on the incoming URL, and a transform binary action.

When you configure this action, drag and drop the **Transform** action, and then select **Use XSLT specified in this action on a non-xml Message**, and all of the WTX map options appear (Figure 11-35 on page 266).

Important: Our ERP_Transformation Policy has only one request rule, because we will only send messages, and no response is expected. If you were to test this scenario with data going both ways, make sure that you add whichever response rule suits you best and add a Reply Q to the MQ FSH.

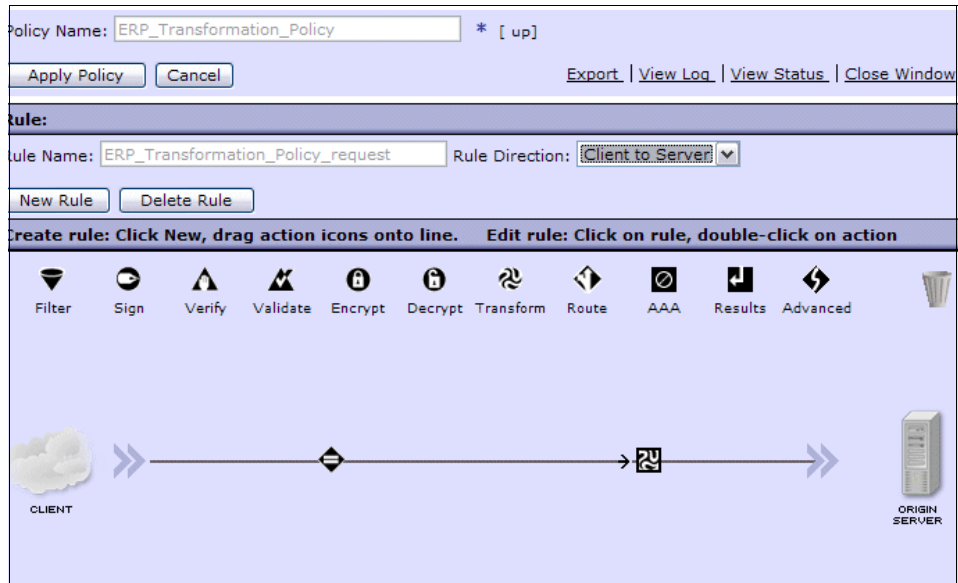


Figure 11-34 ERP_Transformation_Policy overview

In Figure 11-35 on page 266, we can see all the options for adding the transformation map. Make sure that you have uploaded your map previously or just use **Upload** at this point.

The screenshot shows the configuration for the 'Transform Binary' action. The 'Input' field is set to 'INPUT'. Under 'Options', the 'Use Document Processing Instructions' section has three radio buttons: 'Use XSLT specified in this action on a non-XML message' (selected), 'Use XSLT specified in this action', and 'Use XSLT specified in XML document processing instructions, if available'. The 'Processing Control File' is set to 'local:///'. The 'WTX Map file' is set to 'local:///'. The 'WTX Map Mode' is set to 'DPA'. The 'URL Rewrite Policy' is set to '(none)'. The 'Asynchronous' option is set to 'off'. The 'Output' field is set to 'OUTPUT'.

Figure 11-35 Transform Binary action configuration details

Click **Done** and apply your policy and then your MPG settings. We are now ready to test our scenario.

11.5 Testing our solution

Everything has been successfully configured, and the infrastructure is “up and running,” so now it is time to test our scenario and see transaction results in our Transaction Viewer and system logs.

Figure 11-36 on page 267 shows the steps that will occur when we trigger the scenario putting the ERP message in Q5.

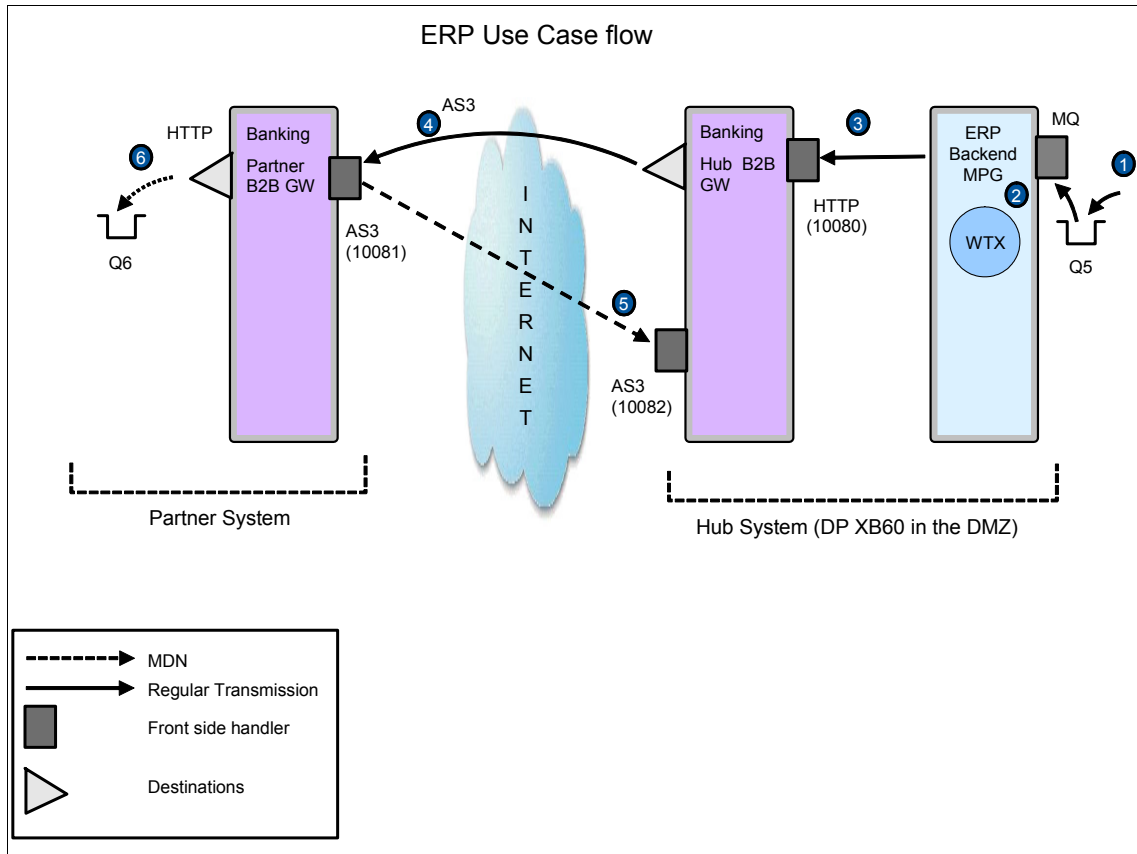


Figure 11-36 Inbound flow steps for the scenario

Here is an explanation of the steps that are shown in Figure 11-36. The numbers correspond to Figure 11-36:

1. A message (flat file) is put on Q5 with RfhUtil, where the ERP_MQ Front Side Handler is listening (Figure 11-37 on page 268).

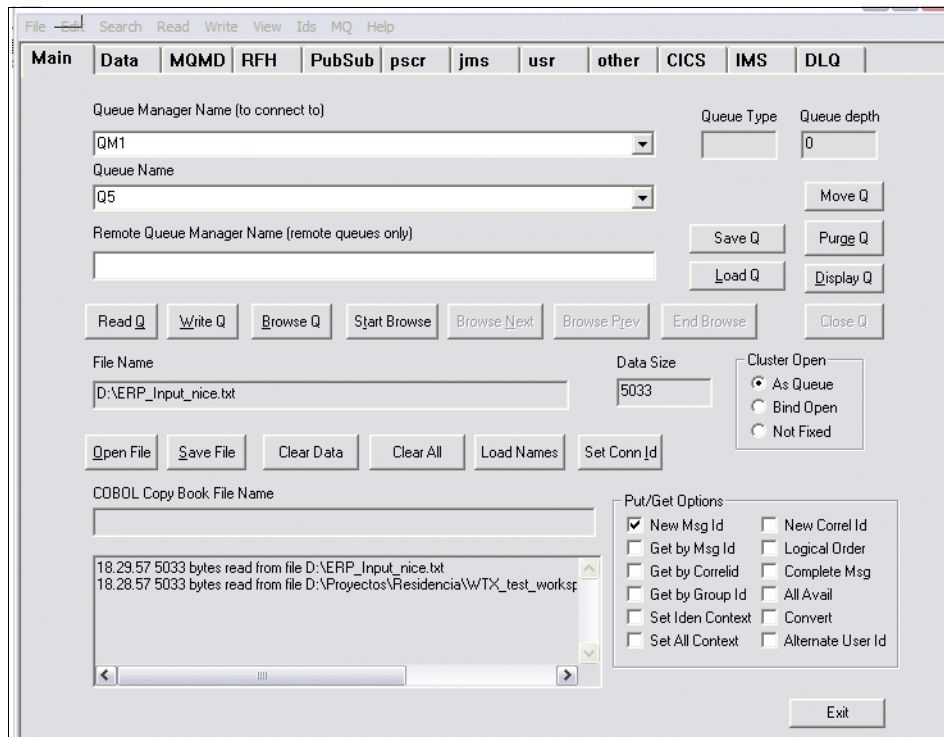


Figure 11-37 Writing a message in Q5 using RfhUtil

2. The message is grabbed by ERP_MQ FSH and the message enters in the MPG Policy ERP_Transformation_Policy where the transformation occurs that converts the Input Content to XML.
3. After the message is transformed, it is sent via HTTP to the Banking Hub B2B Gateway Service, which is listening with its HTTP FSH.
4. The message is wrapped in AS2 (including signing and encryption) and sent to Banking_Partner_B2BGW using the BankingPartner profile (AS3_2_bankingpartner destination on port 10081).
Banking_Partner_B2BGW receives the AS3 message using Banking_Partner_AS3FSH , which is included in it.
5. An MDN is sent through 10082 to Banking_Hub_B2BGW.
6. Banking_Partner_B2BGW verifies the message signature, deciphers it, and unwraps the message. Then, it is sent back through MQ to Q6.

11.5.1 Test results

After the completion of step 1, the rest of the steps are triggered within the device, and the processing finishes with a message in Q6. Let us examine the queue using the RfhUtil utility (Figure 11-38).

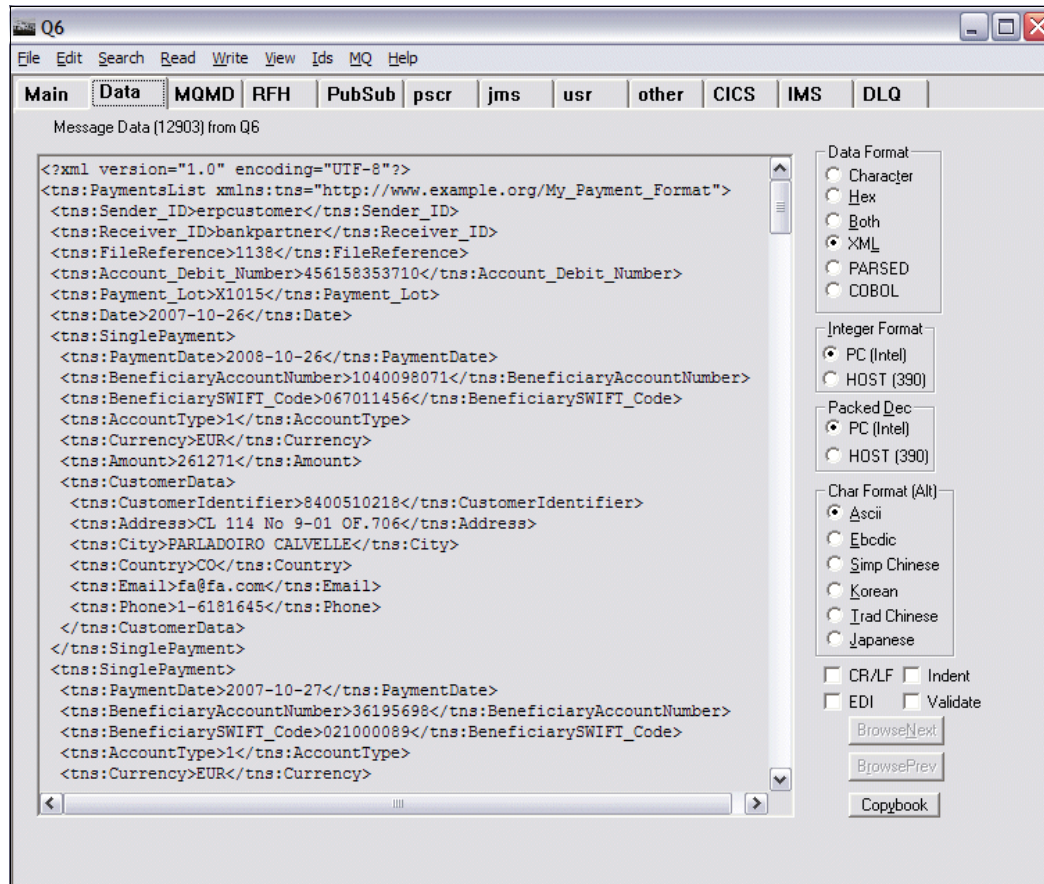


Figure 11-38 Browsing output messages in Q6 using RfhUtil

And, let us see our transaction in the Transaction Viewer (Figure 11-39 on page 270).

| | | | | | | | | | |
|--------------------------|---------------------|------|-----------------------|---|-----------------------------------|------------------------------|---------|------------------------|------------------------------|
| | | | | erpcustomer | | | | | |
| <input type="checkbox"/> | 224 | 7591 | Banking_Partner_B2BGW | Sender: erpcustomer (erpcustomer) | as3://127.0.0.1:10081/% 2F/as3 | 2009-03- 20 14:59:01.0 | Success | Sent (Positive) | 2009-03- 20 14:59:01.0 |
| | | | | Receiver: bankpartner (bankpartner) | dpmq://QM1/? RequestQueue=Q6 | 2009-03- 20 14:59:01.0 | | | |
| <input type="checkbox"/> | 223 | 7575 | Banking_Hub_B2BGW | Sender: erpcustomer (erpcustomer) | http://127.0.0.1:10080/ | 2009-03- 20 14:59:01.0 | Success | Received (Positive) | 2009-03- 20 14:59:01.0 |
| | | | | Receiver: bankpartner (bankpartner) | as3://127.0.0.1:10081/as3 | 2009-03- 20 14:59:01.0 | | | |
| | | | | Sender: | | | | | |

Figure 11-39 Transaction Viewer showing transactions

If we examine the content in detail, you can see that Document type and Document IDs have been selected using the XML Format that we configured, and they are displayed in the Transaction Viewer (Figure 11-40).

| | | | | | |
|------------------------------|----------------|------|--------------|---|------------------------|
| | | | | | |
| | (Show Headers) | 1138 | PaymentsList | 0 | Resend |
| 2009-03- 20 14:59:01.0 | (Show Headers) | 1138 | PaymentsList | 0 | Resend |
| | | | | | |

Figure 11-40 Document type (Payment list) and Document ID (1138) details

If we want to examine the Header data, we can by clicking **Show Headers** (Figure 11-41 on page 271).

| | | | | | | | | | |
|------|--|---|-----------------------------------|------------------------------|---------|----------|--|------------|-------------------|
| 7217 | Banking_Hub_B2BGW | Sender: bankpartner | as3://127.0.0.1:10082/% 2F/as3 | 2009-03- 20 14:34:41.0 | Success | Positive | | | (Show Headers) |
| 2198 | <div style="border: 1px solid black; padding: 5px;"> <p>Header Data</p> <p>Message ID Header 2bbe0e25-759a-460b-8694-90f3c20f4c13@127.0.0.1</p> <p>Content Type Header multipart/report; report-type=disposition-notification; boundary=0e6c5f16-40d6-40b0-a602-5f9ff0bba4ff</p> <p>AS From Header bankpartner</p> <p>AS To Header erpcustomer</p> <p>Date Header Fri, 20 Mar 2009 14:34:41 GMT</p> <p>Disposition Header</p> <p>Disposition Options Header</p> <p>Content Length Header -1</p> <p>Content Disposition Header</p> <p>Original Message ID Header 6a3add0e-3b1f-4ef8-aaa4-41b3a1eb4970@127.0.0.1</p> </div> | | | | | | | | |
| 1429 | | Receiver: bankpartner (bankpartner) | as3://127.0.0.1:10081/as3 | 2009-03- 20 14:34:41.0 | | | | 14:34:41.0 | |
| 645 | Banking_Hub_B2BGW | Sender: bankpartner | as3://127.0.0.1:10082/% 2F/as3 | 2009-03- 20 14:29:13.0 | Success | Positive | | | (Show Headers) |

Figure 11-41 Showing AS3 Headers

if we want to see any of the incoming or outgoing content, we click the Transaction Set ID that corresponds to our transaction, and a Document pop-up menu appears where you can select **Content** (Figure 11-42 on page 272).

The screenshot shows the B2B Viewer application interface. On the left, there is a table with columns for Transaction Set ID and Transaction ID. Transaction 211 is selected, and a 'Show Document' dialog box is open over it, displaying XML content. The XML content is as follows:

```

<?xml version="1.0" encoding="UTF-8" ?>
- <tns:PaymentsList xmlns:tns="http://www.example.org/My_Payment_Format">
  <tns:Sender_ID>erpcustomer</tns:Sender_ID>
  <tns:Receiver_ID>bankpartner</tns:Receiver_ID>
  <tns:FileReference>1138</tns:FileReference>
  <tns:Account_Debit_Number>456158353710</tns:Account_Debit_Number>
  <tns:Payment_Lot>X1015</tns:Payment_Lot>
  <tns>Date>2007-10-26</tns>Date>
- <tns:SinglePayment>
  <tns:PaymentDate>2008-10-26</tns:PaymentDate>
  <tns:BeneficiaryAccountNumber>1040098071</tns:BeneficiaryAccountNumber>
  <tns:BeneficiarySWIFT_Code>067011456</tns:BeneficiarySWIFT_Code>
  <tns:AccountType>1</tns:AccountType>
  <tns:Currency>EUR</tns:Currency>
  <tns:Amount>261271</tns:Amount>
- <tns:CustomerData>
  <tns:CustomerIdentifier>8400510218</tns:CustomerIdentifier>
  <tns:Address>CL 114 No 9-01 OF.706</tns:Address>
  <tns:City>PARLADOIRO CALVELLE</tns:City>
  <tns:Country>CO</tns:Country>

```

At the bottom of the application, a status bar shows details for the selected transaction: Banking_Hub_B2BGW, bankpartner, 2F/as3, 14:29:13.0, Success, Positive.

Figure 11-42 Showing content of the trading document using Transaction Viewer



Trading outbound binary documents using the B2B Gateway Service

This chapter shows how to configure WebSphere DataPower B2B Appliances XB60 to trade with partners using binary payloads coming from a back end to the trading partner.

12.1 Business value

Trading binary content between partners is an integral use case in the business-to-business (B2B) world and has many real-world applications, for instance, exchanging medical image data between a health company and an insurance company or media companies exchanging video data.

To display an example of trading binary content, we present a business scenario where Company A (Hub) needs to exchange images in a secure jpg format with a number of trading partners. Part of the core requirements and perceived business value that the customer is looking for includes:

- ▶ The ability to verify partner information in the demilitarized zone (DMZ)
- ▶ Support for receiving AS2 B2B messages
- ▶ The ability to sign and encrypt all AS2 data
- ▶ The capability to sign the AS2 Message Disposition Notifications (MDNs)
- ▶ The ability to transport data (including MDNs) in a secure fashion via Secure Sockets Layer (SSL)
- ▶ Encryption of all payload data stored on the appliance

12.2 Prerequisites

This scenario requires the following software and skills.

12.2.1 Software prerequisites

In order to be able to run this scenario, you must have installed the following components:

- ▶ WebSphere DataPower B2B Appliances XB60
- ▶ WebSphere MQ V6
- ▶ RfhUtil V6 utility

12.2.2 Skills prerequisites

This scenario is intended for the intermediate user, meaning that in order to be able to fully implement and understand this scenario, you need to be familiar with:

- ▶ WebSphere DataPower B2B Appliances XB60 major concepts (you need to have completed having scenario 1)
- ▶ Basic Extensible Stylesheet Language Transformation (XSLT) techniques

12.3 Presenting the scenario

For this use case, we implement a one-way flow that corresponds to an incoming binary file (we have chosen a jpg file). This message will be sent from a back-end application and routed to a specific partner trading in binary documents.

To achieve this transaction flow, the message will be sent to DataPower via MQ queues. From there, it will be sent to the trading partner wrapped in AS2 over HTTP/SSL. The Hub will wait for an asynchronous MDN response sent as AS2 over SSL to mark the transaction complete.

All the messages exchanged between the hub company and trading partner will be encrypted and signed as per the trading agreement between the partners.

Figure 14-1 shows the flow from a high-level perspective.

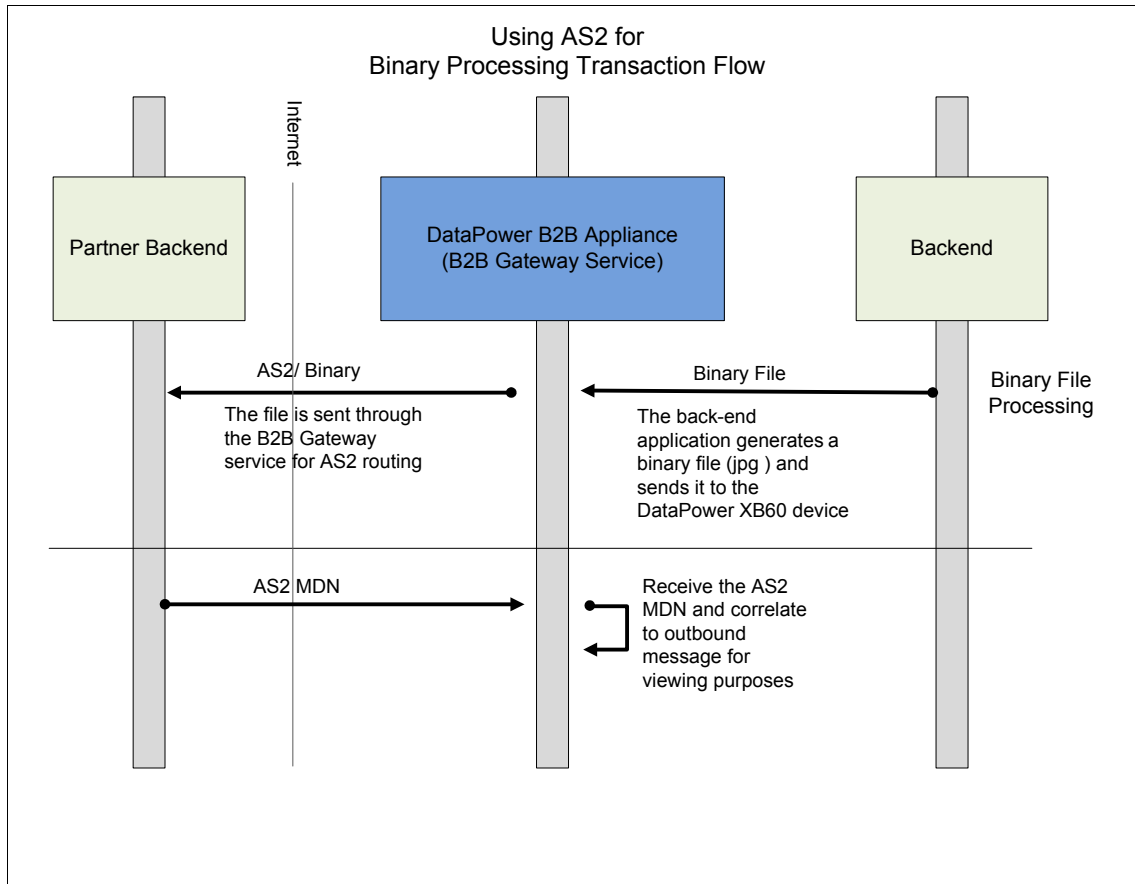


Figure 12-1 Scenario overview

12.4 Scenario solution

We describe the implementation of the scenario from a high-level perspective, and then, we explain the technical steps necessary to fully implement the solution.

The key aspect of the scenario implementation is that we are dealing with binary payloads. The XB60 often receives these messages via a protocol, such as WebSphere MQ (as our case), that does not have the same routing headers as AS2 or AS3, where specific information about the sender and the receiver is set. In a regular X12 scenario (same as with EDIFACT) messages, the XB60 can parse the message contents and find the partner IDs from the ISA and UNA

headers. In XML, the XPath Routing Policies tab on the B2B Gateway configuration tells the XB60 how to retrieve the partner IDs from the payload.

Important: In the case of binary messages, an appliance administrator must configure an XSLT stylesheet as the Document Routing Preprocessor for the gateway to route the message properly.

Another important point of this scenario is the fact that all the communications will be over SSL (including MDNs), which makes communication with our trading partners more secure at the transport layer.

A full picture of all the services and objects to implement is presented in the scenario outline, where we also present a summary of what we will create.

12.4.1 Scenario outline

Figure 12-2 on page 278 outlines the message flow for the binary with AS2 payload scenario. The left side of the Figure 12-2 on page 278 corresponds to the simulated trading partner, which is referred as the binary partner, and the right side corresponds to the hub system, which is referred as the Binary Hub. This architecture demonstrates that the DataPower device is fully implemented from both a partner and provider perspective.

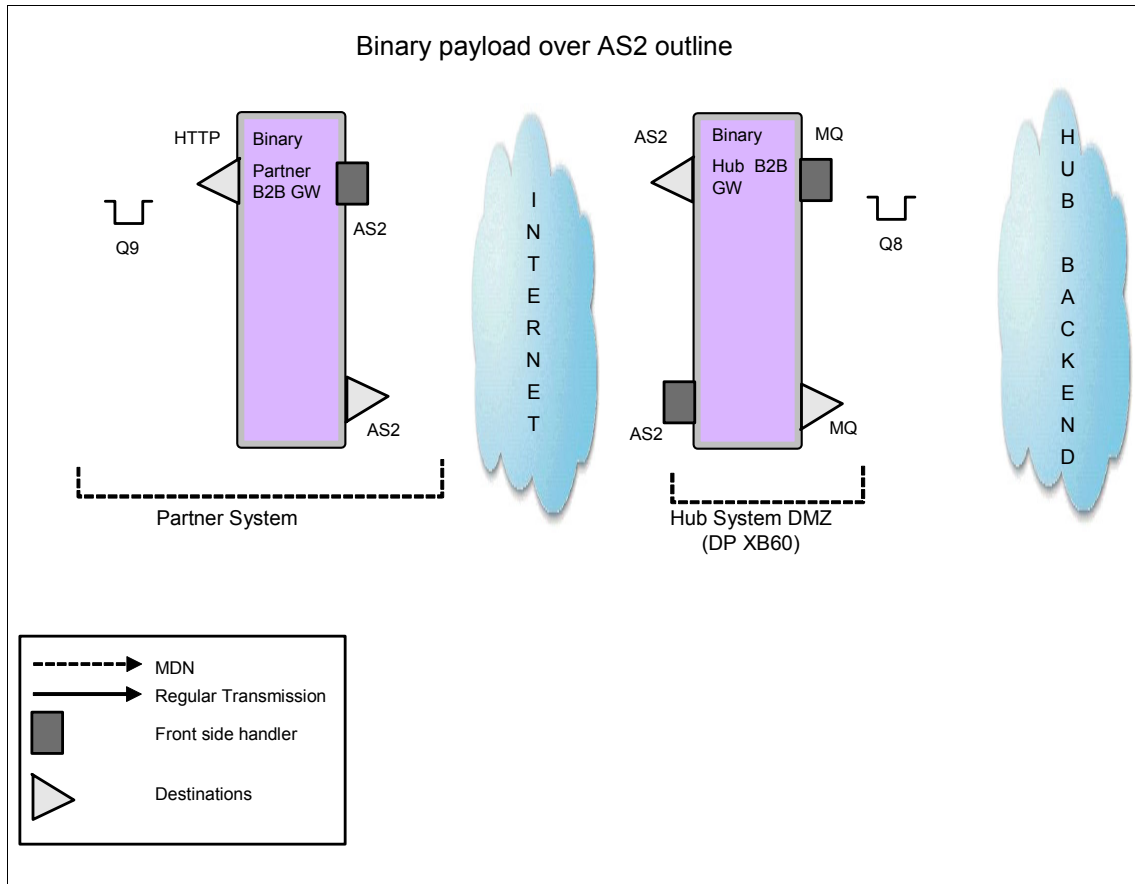


Figure 12-2 Binary files trading outline

Here is a summary of the steps needed to implement the scenario:

- ▶ Step 1: Creating all the necessary crypto objects
- ▶ Step 2: Creating the Binary Partner profiles
- ▶ Step 3: Creating the Binary Hub profiles
- ▶ Step 4: Creating the Binary Partner B2B Gateway Service
- ▶ Step 5: Creating the Binary Hub B2B Gateway Service

12.4.2 Scenario implementation

Next, we explain the steps to implement our scenario.

Step 1: Creating all the necessary crypto objects

This scenario is intended for those individuals who have experience with the DataPower device, so we do not go into detail about how to create the crypto objects, but it is important to point out that several objects are required for signing and validating signatures, as well as encrypting and decrypting payloads.

The following objects have been created specifically for this scenario:

- ▶ Public/Private keys:
 - Bankingpartner key pairs
 - Bankinghub key pairs
- ▶ Validation credentials:
 - Banking_Partner_valcred: Used to validate signatures coming from the Banking Partner
 - Banking_Hub_valcred: Used to validate signatures coming from the Banking Hub
- ▶ Identification credentials:
 - Banking_Partner_IDcred: Used to sign messages from the Banking Partner system and to decrypt payloads coming from the Banking Hub
 - Banking_Hub_IDcred: Used to sign messages from the Banking Hub system and to decrypt payloads coming from the Banking Partner

Both ID credentials and validation credential objects are configured with the corresponding key pair.

Important: We will explain SSL-specific objects, as SSL Proxy Profile and Crypto Profile, in an upcoming section.

Step 2: Creating the Binary Partner profiles

As explained in the previous chapters, a *Partner profile* is an object that defines routing destinations for messages, as well as establishes the AS Security rules, when interchanging information with that specific partner.

Because we simulate trading between partners within the single DataPower device, we need two separate binary partner profiles to be included in the B2B Gateway: one internal profile for the B2B Gateway Service that represents the partner back end and one external profile whose destination will be pointing to the other B2B Gateway Service.

Binary partner internal profile

The binary partner internal profile contains information for the type of AS Security that is defined for this partner, including signing outgoing messages and decrypting incoming messages. This information is located in the AS Security tab. The destination definition for this partner profile is listed in the Destinations tab.

The Main tab of the Binary_Partner_int internal partner profile (Figure 12-3) allows you to add a Business ID, which allows the the B2B Gateway (B2BGW) service to identify which profiles must be handling the request. In this case, we have one Business ID called `binarypartner`. This name is the identifier of the business partner that is waiting for the binary file.

The screenshot shows a web-based configuration interface titled "Configure B2B Partner Profile". It has four tabs: "Main", "AS Security", "Destinations", and "Contacts". The "Main" tab is selected. Below the tabs, the profile name is "B2B Partner Profile: Binary_Partner_int [up]". There are four buttons: "Apply", "Cancel", "Delete", and "Undo". The "Admin State" is set to "enabled" (radio button selected). The "Comments" field is empty. The "Profile Type" is set to "Internal" (radio button selected). The "Partner Business IDs" field contains "binarypartner" and has an "Add" button. A small asterisk is visible below the "Partner Business IDs" field.

Figure 12-3 Binary partner internal profile Main configuration tab

Figure 12-4 on page 281 shows the configuration of the AS Security parameters. Because this profile is an internal profile, all the Identification credentials must be configured in here. A requirement of this scenario is to sign and encrypt AS2 messages that we exchange, so we check all those boxes, and we include the Binary Partner ID Credentials that we created in “Step 1: Creating all the necessary crypto objects” on page 279.

Note: The ID Credentials contain the private key of the Banking Partner that is used both for signing and decrypting in this case.

Configure B2B Partner Profile

Main **AS Security** Destinations Contacts

B2B Partner Profile: Binary_Partner_int [up]

Apply Cancel Delete Undo

Inbound Security

Require Signature

Require Encryption

Inbound Decryption Identification Credentials Binary_Partner_IDcred + ... *

Outbound Security

Sign Outbound Messages

Signing Identification Credentials Binary_Partner_IDcred + ... *

Signing Digest Algorithm sha1

Figure 12-4 Binary partner internal profile AS Security configuration tab

In Figure 12-5 on page 282, we configure the Destinations tab, where documents from the Partner must be routed. Figure 12-2 on page 278 shows that the binary partner back end is simulated by MQ queue, therefore, we define an MQ destination. It is also important to notice that we have only enabled **Binary** as the document type, because this type is the only type that we will handle. The Contacts tab is an optional tab and we will not populate this tab.

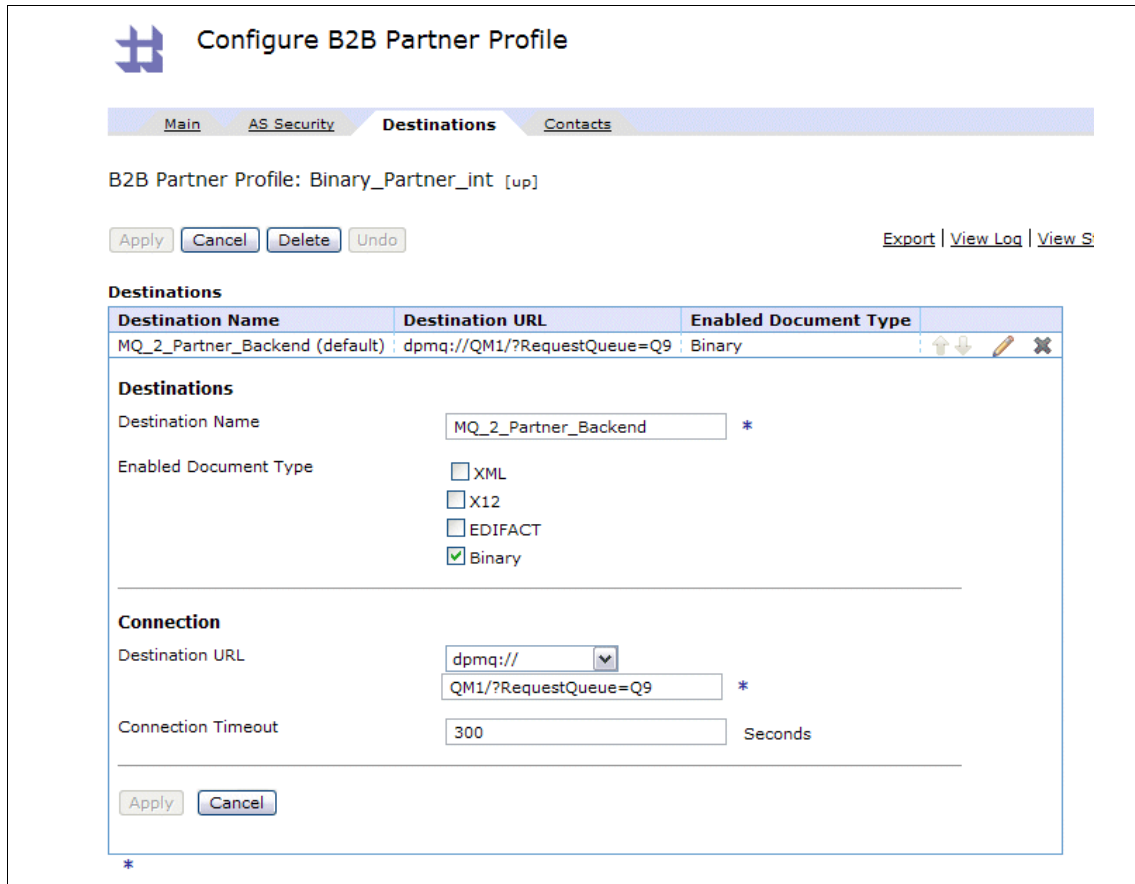


Figure 12-5 Binary Partner internal profile Destinations configuration tab details

Binary partner external profile

The binary partner external profile (refer to Figure 12-6 on page 283) manages the information needed by the hub in order to successfully trade with the Binary Partner. We will define parameters in the AS Security tab and Destinations tab.

The Main tab of the Binary_Partner external partner profile (Figure 12-6 on page 283) allows you to add a Business ID, which allows the B2BGW service to identify which profiles need to be handling the request. In this case, we have one Business ID called binarypartner.

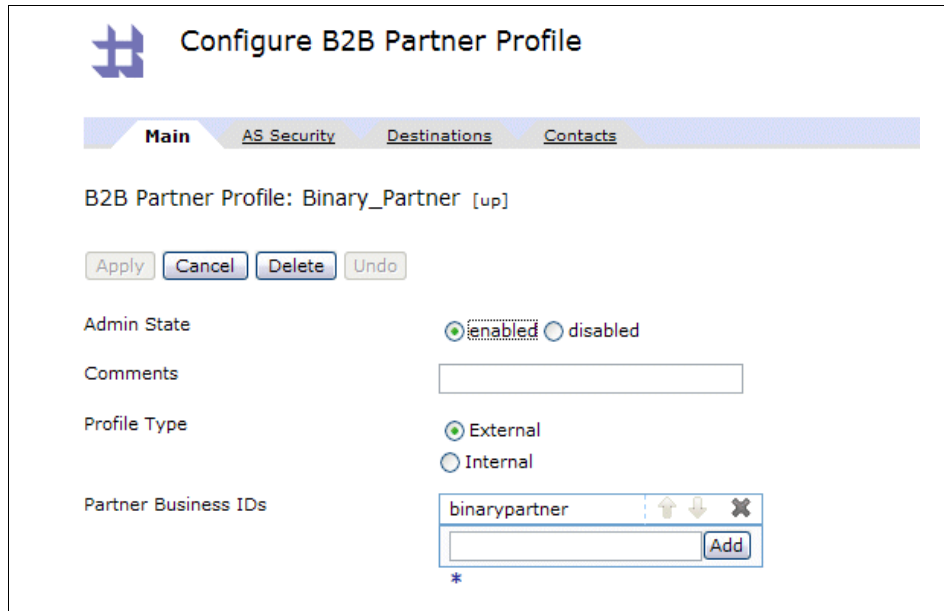


Figure 12-6 Binary Partner external profile Main configuration tab

Figure 12-7 on page 284 shows the AS2 Security tab, and we focus on two aspects:

- ▶ Inbound Signature Validation Credentials are required in order to verify the signed payload. For this value, we use the object **Binary_partner_valcred**.
- ▶ In addition, an MDN SSL Proxy Profile is needed to establish the SSL connection when sending the MDN to the partner. We have named it **AS2_2_BinaryPartner_PP**.

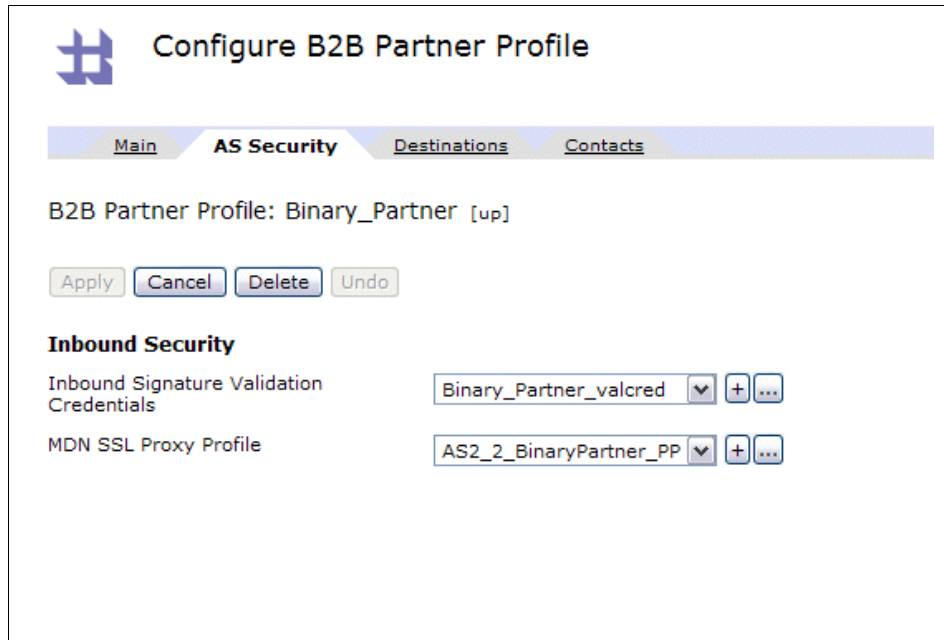


Figure 12-7 Binary Partner internal profile AS Security configuration tab

Inspecting the MDN SSL Proxy Profile's configuration (refer to Figure 12-8 on page 285), we see that it has been defined as **Forward**, meaning that when we establish the SSL handshake, we act as clients of the Partner, because it is the Hub that connects to the Partner system to make the SSL connection for the MDN.

To finish this configuration, we need to provide an SSL Forward Crypto Profile, which we name AS2_2_BinaryPartner_CP.

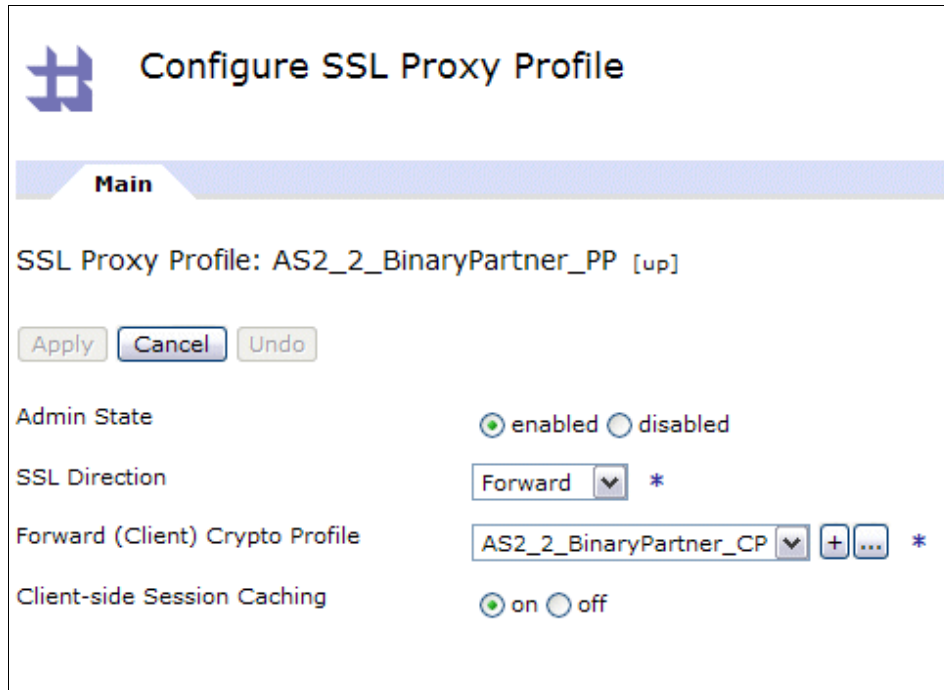


Figure 12-8 Binary Partner internal profile SSL Proxy Profile configuration

Inspecting the Crypto Profile's configuration (refer to Figure 12-9 on page 286), we see that the object Binary Hub validation credentials have been added to it. This way, when the Hub tries to establish the SSL connection to the Partner, and the Partner presents its certificate to it, the connection will be made only if it presents a trusted certificate, which is stored in the Binary_Partner_valcred object.

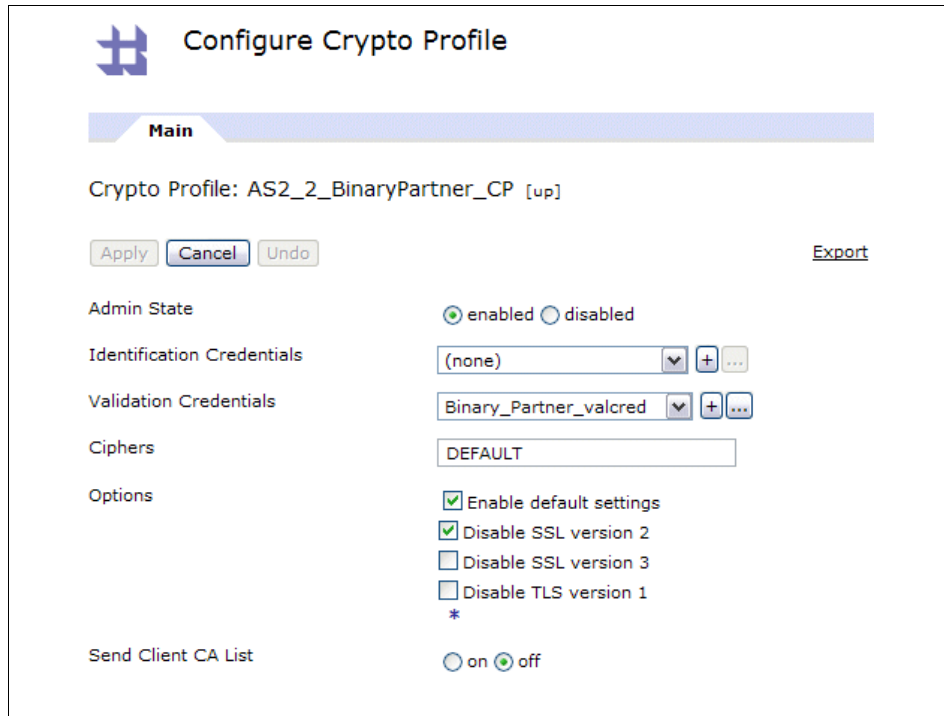


Figure 12-9 Binary Partner internal profile Crypto Profile configuration

In the Destinations tab (refer to Figure 12-10 on page 287), we must add an AS2 destination: this is the information that describes how to route messages to the Partner system from the Hub system, and **AS2** is the selected protocol.

Only **Binary** is enabled as a valid document type, because for this specific partner, we will only handle this kind of documents.

Next, we add the URL of the Binary Partner's AS2 Front Side Handler of the Binary_Partner B2B Gateway Service, which is where it listens for incoming AS2 messages.

Moreover, we include the AS2_2_BinaryPartner SSL proxy profile that was used in the MDN SSL Proxy profile. In this case, we are acting as clients to our trading partner.

Configure B2B Partner Profile

Main AS Security Destinations Contacts

B2B Partner Profile: Binary_Partner [up]

Apply Cancel Delete Undo Export View Log View Status

Destinations

| Destination Name | Destination URL | Enabled Document Type |
|-------------------------------|------------------------|-----------------------|
| AS2_2_BinaryPartner (default) | as2s://127.0.0.1:20060 | Binary |

Destinations

Destination Name: AS2_2_BinaryPartner *

Enabled Document Type:

- XML
- X12
- EDIFACT
- Binary

Connection

Destination URL: as2s:// 127.0.0.1:20060 *

SSL Proxy Profile: AS2_2_BinaryPartner_PP + ... *

Connection Timeout: 300 Seconds

User name: []

Password: []

Figure 12-10 Binary Partner external profile Destinations details

If we scroll down on the window (refer to Figure 12-11 on page 288), we see the AS Outbound security and the Advanced AS Behavior details. This security is message-oriented (the way the AS2 messages are signed and decrypted), not at the transport layer as in SSL.

| AS Outbound Security | |
|--|---|
| Send Messages Unsigned | <input type="checkbox"/> |
| Encrypt Messages | <input checked="" type="checkbox"/> |
| Encryption Certificate | binary_partner <input type="button" value="+"/> <input type="button" value="..."/> * |
| <hr/> | |
| Advanced AS Behavior | |
| Compress Messages | <input type="checkbox"/> |
| Request MDN | <input checked="" type="checkbox"/> |
| Time to Acknowledge | <input type="text" value="300"/> Seconds |
| Request Asynchronous MDN | <input checked="" type="checkbox"/> |
| AS2 MDN Redirection URL | https:// <input type="button" value="v"/> <input type="text" value="127.0.0.1:20061"/> |
| Request Signed MDN | <input checked="" type="checkbox"/> |
| Attempt Message Retransmission | <input checked="" type="checkbox"/> |
| Maximum Retransmissions | <input type="text" value="3"/> |
| <hr/> | |
| <input type="button" value="Apply"/> <input type="button" value="Cancel"/> | |

Figure 12-11 Binary Partner external profile Destinations details (continuation)

In the Outbound security details section (refer to Figure 12-11), make sure that you select **Encrypt messages** and select the **binary_partner** certificate that was previously generated. Leave “Send Messages Unsigned” unchecked, because we want to sign our messages.

In the advanced AS behavior section, select **Request MDN** and **Request Asynchronous MDN** and make sure that the AS2 MDN Redirection URL is selected using **https** protocol.

The As2 MDN Redirection URL is the profile that we use to send the AS headers to the trading partner, so it knows where to route the MDNs, which is why we set up the partner’s own Front Side Handler (FSH) port.

Important: Remember that the scenario that we implement is a one-way flow from the Hub to the Partner, so in this case, there will be no MDN coming from the Hub.

Step 3: Creating the Binary Hub profiles

Similar to “Step 2: Creating the Binary Partner profiles” on page 279, we also need two separate Binary Hub profiles. The internal profile is associated with the Binary Hub B2B Gateway service (refer to Step 4 for further details), and an external profile will be associated with the Binary Partner B2B Gateway service. This profile will allow our partner to successfully send all the messages to the Binary Hub, as well as, among other things, validating the incoming signature.

Binary Hub internal profile

The Binary Hub internal profile (refer to Figure 12-12 on page 290) contains information that will manage the kind of AS Security that the provider expects. This security includes how outbound messages will be signed, how the incoming messages will be decrypted, and where the messages will be routed to the partner system.

The Main tab of the Binary_Hub_int internal partner profile (Figure 12-12 on page 290) allows us to add a Business ID so that the B2BGW service can identify which profiles need to handle the request. Remember that for binary use cases that *do not include AS2 or AS3 wrapping*, we will use an XSLT that will help us set up the sender and receiver information, so the message processing can be associated with the right profiles that come in the XML sent from our back end.

Configure B2B Partner Profile

Main AS Security Destinations Contacts

B2B Partner Profile: Binary_Hub_int [up]

Apply Cancel Delete Undo

Admin State enabled disabled

Comments

Profile Type External Internal

Partner Business IDs

*

Figure 12-12 Binary Hub internal profile Main configuration tab

Figure 12-13 on page 291 defines the AS Security tab parameters. Because this profile is an internal profile, all the identification credentials must be configured here. It is a requirement in this scenario to exchange signed and encrypted AS2 messages, so we must check all those boxes, and include the Binary Partner ID Credentials that we created in “Step 1: Creating all the necessary crypto objects” on page 279. Notice that the ID Credentials contain the private key of the Banking Partner that, in this case, is used both for signing and decrypting.

Configure B2B Partner Profile

Main **AS Security** Destinations Contacts

B2B Partner Profile: Binary_Hub_int [up]

Apply Cancel Delete Undo

Inbound Security

Require Signature

Require Encryption

Inbound Decryption Identification Credentials Binary_Hub_IDcred [v] + ... *

Outbound Security

Sign Outbound Messages

Signing Identification Credentials Binary_Hub_IDcred [v] + ... *

Signing Digest Algorithm sha1 [v]

Figure 12-13 Binary Partner internal profile AS Security configuration tab

Figure 12-14 on page 292 shows the definitions of the Destinations tab, where documents from the Partner need to be routed. Figure 12-2 on page 278 shows that in the main architecture, the binary Hub back end is simulated by an MQ queue so an MQ destination is needed.

It is also important to notice that we have only enabled **Binary** as the document type, because this type is the only type that will be handled.

Important: When testing this scenario, we do not use this destination, because we only have one flow.

Configure B2B Partner Profile

Main AS Security **Destinations** Contacts

B2B Partner Profile: Binary_Hub_int [up]

Apply Cancel Delete Undo Export View Log V

Destinations

| Destination Name | Destination URL | Enabled Document Type |
|------------------------|-------------------------|-----------------------|
| MQ_2_Backend (default) | dpmq://RequestQueue?Q10 | Binary |

Destinations

Destination Name: MQ_2_Backend *

Enabled Document Type:

- XML
- X12
- EDIFACT
- Binary

Connection

Destination URL: dpmq:// RequestQueue?Q10 *

Connection Timeout: 300 Seconds

Apply Cancel

Figure 12-14 Binary Hub internal profile Destination configuration tab details

Binary Hub external profile

The Binary Hub external profile (Figure 12-15 on page 293) manages the information needed by the Partner in order to successfully trade with Binary Hub and where the messages will be routed to the partner system.

The Main tab of the Binary_Hub (Figure 12-15 on page 293) external partner profile, from a Business ID perspective, is the same configuration that we had on the internal profile.

Configure B2B Partner Profile

Main AS Security Destinations Contacts

B2B Partner Profile: Binary_Hub [up]

Apply Cancel Delete Undo

Admin State enabled disabled

Comments

Profile Type External Internal

Partner Business IDs

*

Figure 12-15 Binary Hub external profile Main configuration tab

From an AS2 Security perspective, we focus on two aspects:

- ▶ Inbound Signature Validation Credentials will be required in order to verify the signed payload. For this value, we will use the object **Binary_Hub_valcred**.
- ▶ In addition, an MDN SSL Proxy Profile is needed to establish the SSL connection when sending the MDN to the partner. We have named it **AS2_2_BinaryHub_PP**.

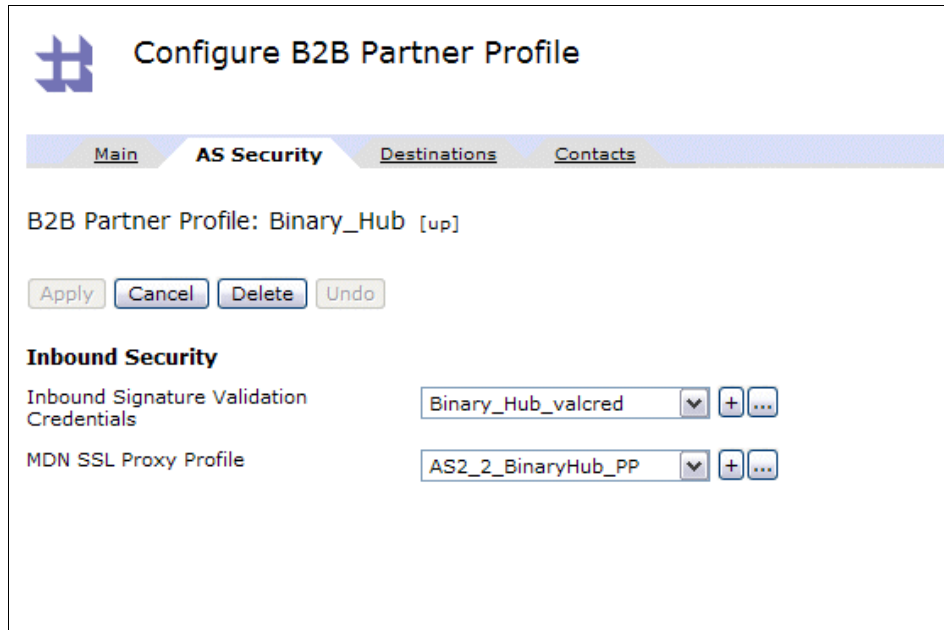


Figure 12-16 Binary Hub external profile AS Security configuration tab

If we explore MDN SSL Proxy Profile's configuration (Figure 12-17 on page 295), we can see that it has been defined as **Forward**, meaning that when we establish the SSL handshake, we act as clients of the Hub, because the Partner is the one that connects to the Hub system to make the SSL connection for the MDN.

To finish this configuration, we need to provide an SSL Forward Crypto Profile, that in this case we have named AS2_2_BinaryHub_CP.

Configure SSL Proxy Profile

Main

SSL Proxy Profile: AS2_2_BinaryHub_PP [up]

Apply Cancel Undo Export

Admin State enabled disabled

SSL Direction Forward *

Forward (Client) Crypto Profile AS2_2_BinaryHub_CP + ... *

Client-side Session Caching on off

Figure 12-17 Binary Hub external profile SSL Proxy Profile configuration

The Crypto Profile's configuration contains the Binary Hub validation credentials (Figure 12-18 on page 296). When the Partner tries to establish the SSL connection to the Hub, and the Hub presents its certificate, the connection will be made only if it presents a trusted certificate.

Main

Crypto Profile: AS2_2_BinaryHub_CP [up]

Apply Cancel Undo Export

Admin State enabled disabled

Identification Credentials (none) + ...

Validation Credentials Binary_Hub_valcred + ...

Ciphers DEFAULT

Options

- Enable default settings
- Disable SSL version 2
- Disable SSL version 3
- Disable TLS version 1
- *

Send Client CA List on off

Figure 12-18 Binary Hub external profile Crypto Profile configuration

In the Destinations tab (Figure 12-19 on page 297), we add an AS2 destination containing the information that describes how to route messages to the Hub system from the Partner system via AS2.

Again, only **Binary** is enabled as a valid document type. Next, we add the URL of Banking Hub's AS2 Front Side Handler of the Binary_Hub B2B Gateway service, which is where it will listen for incoming AS2 messages.

Moreover, it will be necessary to include the AS2_2_BinaryHub SSL proxy profile that we have already used in the MDN SSL Proxy profile, because, in this case, we also act as clients to our trading partner, to whom we will be serving the SSL connection.

Configure B2B Partner Profile

This configuration has been modified, but not yet saved.

Main AS Security **Destinations** Contacts

B2B Partner Profile: Binary_Hub [up]

Apply Cancel Undo Export View

Destinations

| Destination Name | Destination URL | Enabled Document Type |
|---------------------------|------------------------|-----------------------|
| AS2_2_BinaryHub (default) | as2s://127.0.0.1:20061 | Binary |

Destinations

Destination Name: AS2_2_BinaryHub *

Enabled Document Type:

- XML
- X12
- EDIFACT
- Binary

Connection

Destination URL: as2s:// 127.0.0.1:20061 *

SSL Proxy Profile: AS2_2_BinaryHub_PP + ... *

Connection Timeout: 300 Seconds

User name: []

Password: []

Figure 12-19 Binary Hub external profile Destinations details

Figure 12-20 on page 298 defines the AS Outbound security and Advanced AS Behavior details. Remember that this security is related to how AS2 messages are signed and decrypted.

In the Advanced AS Behavior section, select **Request MDN** and **Request Asynchronous MDN** and make sure the AS2 MDN Redirector URL is selected using **Https** protocol. As you might have seen in previous chapters, the As2 MDN Redirection URL is the profile that we use to send in the AS headers to the

hub, so it knows where to route the MDNs. That is the reason why we set up the partner's own FSH port.

The screenshot shows two configuration panels. The first panel, titled "AS Outbound Security", includes the following settings: "Send Messages Unsigned" (unchecked), "Encrypt Messages" (checked), and "Encryption Certificate" (set to "binary_hub" with a dropdown arrow, a plus sign, and an ellipsis icon). The second panel, titled "Advanced AS Behavior", includes: "Compress Messages" (unchecked), "Request MDN" (checked), "Time to Acknowledge" (300 Seconds), "Request Asynchronous MDN" (checked), "AS2 MDN Redirection URL" (https:// 127.0.0.1:20060), "Request Signed MDN" (checked), "Attempt Message Retransmission" (checked), and "Maximum Retransmissions" (3). At the bottom of the panel are "Apply" and "Cancel" buttons. A small asterisk is located at the bottom left of the panel's border.

Figure 12-20 Binary Hub external profile destination details (continuation)

Step 4: Creating the Binary Partner B2B Gateway Service

After having created all the profiles for the scenario, it is time to create the B2B Gateway Services that will handle the message traffic and attach the profiles, depending on the case.

For the Binary Partner, this service acts as the entry point to that system, identifying the incoming messages and mapping these messages to the specific profiles that need them. Figure 12-21 on page 299 shows the Binary Partner B2BGW service has two partners attached: the Binary Partner internal profile and the Binary Hub external profile, but you can add as many profiles as your business needs require.

Important: Notice that we only have one FSH, because we will not be handling the flow where messages come from the Partner's back end.

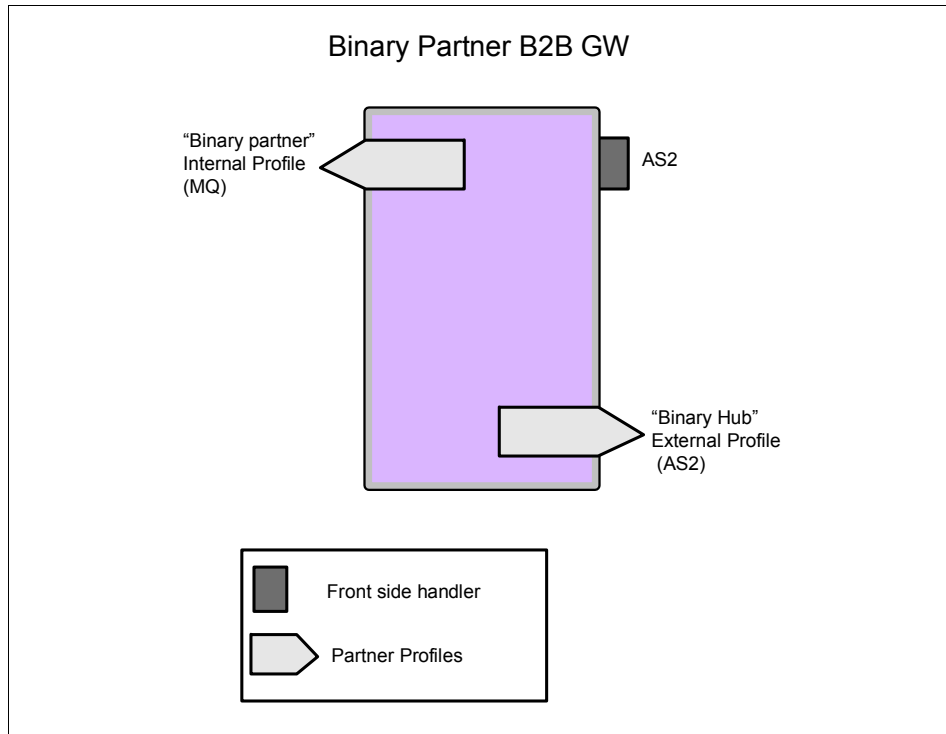


Figure 12-21 Binary Partner B2B Gateway architecture

Figure 12-22 on page 300 shows the Main tab for the Binary Partner B2B Gateway Service.

Configure B2B Gateway

Main
Archive
XML Formats
Advanced

B2B Gateway: Binary_Partner_B2BGW [up]

[Export](#) | [View Log](#) | [View Status](#) | [Archive/p...](#)

General Configuration

Admin State: enabled disabled

Comments:

Document Storage Location: ▼

XML Manager: ▼ *

Document Routing

Front Side Protocol Handlers

| Front Side Protocol | ↑ | ↓ | ✕ |
|------------------------------------|---|---|---|
| Binary_Partner_AS2FSH | ↑ | ↓ | ✕ |
| <input type="button" value="Add"/> | | | |

*

Attach Partner Profiles

Active Partner Profiles

| B2B Partner Profile | Profile Enabled? | Profile Destination | ↑ | ↓ | ✕ |
|---|------------------|------------------------|---|---|---|
| Binary_Hub | enabled ▼ | AS2_2_BinaryHub ▼ | ↑ | ↓ | ✕ |
| Binary_Partner_int | enabled ▼ | MQ_2_Partner_Backend ▼ | ↑ | ↓ | ✕ |
| <input type="text" value="Banking_Hub"/> ▼ <input type="button" value="+"/> <input type="button" value="..."/> <input type="button" value="Add"/> | | | | | |

*

Active Profile Groups

| B2B Profile Group | Group Enabled? |
|---|----------------|
| (empty) | |
| <input type="button" value="▼"/> <input type="button" value="+"/> <input type="button" value="..."/> <input type="button" value="Add"/> | |

Figure 12-22 Binary Partner B2BGW Main configuration tab

Figure 12-23 on page 302 shows that we have only one AS2 Front Side Handler attached in the Front Side Handlers section. There will be no communication from the partner back end to the Banking Partner B2BGW, because this scenario is a one-way flow.

On the other side, we have the two Partner Profiles (Binary_Partner_int and Binary_hub) attached in the Attach Partner Profiles section in Figure 12-22 on page 300.

If you want to add a Partner profile, select the profile that you want from the drop-down list and click **Add**. You can add as many profiles as necessary. The topmost profile will be the default profile.

In Figure 12-23 on page 302, it is important to notice that we are using localhost as a Host Alias for 127.0.0.1 in order not to tightly couple our Front Side Handler with any specific IP address.

Configure AS2 Front Side Handler

Main

AS2 Front Side Handler: Binary_Partner_AS2FSH [up]

Apply Cancel Undo Export | View Log

Admin State enabled disabled

Comments

Local IP Address *

Port Number *

HTTP Version to Client ▼

Allowed Methods and Versions

- HTTP 1.0
- HTTP 1.1
- POST method
- GET method
- PUT method
- HEAD method
- OPTIONS
- TRACE method
- DELETE method
- URL with Query Strings
- URL with Fragment Identifiers
- URL with ..
- URL with cmd.exe

Persistent Connections on off

Compression on off

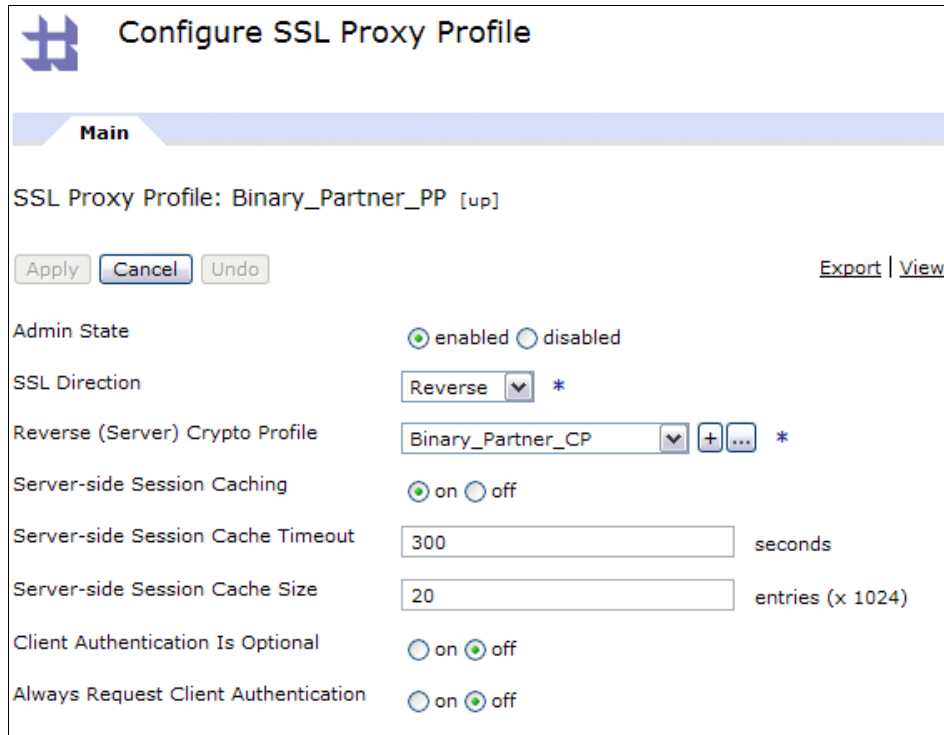
Figure 12-23 Binary Partner AS2 FSH

In Figure 12-24 on page 303, if we scroll further down, we can see that there is an SSL Proxy profile associated to Binary Partner AS2 FSH, because we will be using SSL on our communications.

| | |
|--|---|
| Allowed Methods and Versions | <input checked="" type="checkbox"/> HTTP 1.0 <input checked="" type="checkbox"/> HTTP 1.1 <input checked="" type="checkbox"/> POST method <input type="checkbox"/> GET method <input checked="" type="checkbox"/> PUT method <input type="checkbox"/> HEAD method <input type="checkbox"/> OPTIONS <input type="checkbox"/> TRACE method <input type="checkbox"/> DELETE method <input checked="" type="checkbox"/> URL with Query Strings <input checked="" type="checkbox"/> URL with Fragment Identifiers <input type="checkbox"/> URL with .. <input type="checkbox"/> URL with cmd.exe |
| Persistent Connections | <input checked="" type="radio"/> on <input type="radio"/> off |
| Compression | <input type="radio"/> on <input checked="" type="radio"/> off |
| Maximum Allowed URL Length | <input type="text" value="16384"/> |
| Maximum Allowed Total Header Length | <input type="text" value="128000"/> |
| Maximum Number of HTTP Request Headers Allowed | <input type="text" value="0"/> |
| Maximum Allowed Length of HTTP Header Name | <input type="text" value="0"/> |
| Maximum Allowed Length of HTTP Header Value | <input type="text" value="0"/> |
| Maximum Allowed Length of HTTP Query String | <input type="text" value="0"/> |
| SSL Proxy | <input type="text" value="Binary_Partner_PP"/> ▼ + ... |
| Access Control List | <input type="text" value="(none)"/> ▼ + ... |
| AAA Policy | <input type="text" value="(none)"/> ▼ + ... |

Figure 12-24 Binary Partner AS2 FSH (continuation)

Figure 12-25 on page 304 defines the Binary_Partner_PP, which is configured as a **Reverse** Proxy Profile, because the Front Side Handler will be the actual SSL server on the connections that will be received. Therefore, you need to configure a Reverse Crypto profile, **Binary_Partner_CP**.



Configure SSL Proxy Profile

Main

SSL Proxy Profile: Binary_Partner_PP [up]

Apply Cancel Undo [Export](#) | [View](#)

Admin State enabled disabled

SSL Direction Reverse ▼ *

Reverse (Server) Crypto Profile Binary_Partner_CP ▼ + ... *

Server-side Session Caching on off

Server-side Session Cache Timeout 300 seconds

Server-side Session Cache Size 20 entries (x 1024)

Client Authentication Is Optional on off

Always Request Client Authentication on off

Figure 12-25 Binary Partner AS2 FSH SSL Proxy Profile

On the Binary_Partner_CP window (Figure 12-26 on page 305), we need to include the ID credentials that this B2B Gateway Service presents to the incoming clients so that the SSL connection can be established. We will use the **Binary_Partner_IDCred** object that we created in “Step 1: Creating all the necessary crypto objects” on page 279.

Figure 12-26 Binary Partner AS2 FSH Crypto Profile

On the Archive configuration tab (Figure 12-27 on page 306), we have set the archiving as **Purge only** and entered 3 days of Archive Document Age, even though, depending on the use case, this property might need to be resized.



Figure 12-27 Binary Partner B2BGW Archive configuration tab

There are no XML Formats (Figure 12-28), because we are trading with binary files.



Figure 12-28 Binary Partner B2BGW XML Formats tab

In the Advanced configuration tab (Figure 12-29), we need to set the Default AS2 MDN Return Path: this information will come in the AS2 headers when the partner sends any AS2 messages to the Hub, so that the Hub knows where, by default, it must send the AS2 MDNs back. That it is why we are indicating the port number that matches Partner's AS3 Front Side Handler.

Configure B2B Gateway

Main Archive XML Formats **Advanced**

B2B Gateway: Binary_Partner_B2BGW [up]

Apply Cancel Delete Undo Export View Log View Status Archive/purge transactions

Service Priority Normal

Default AS2 MDN Return Path https://
127.0.0.1:20060

Default AS3 MDN Return Path ftp://

Document Routing Preprocessor store://
b2b-routing.xsl Upload... Fetch... Edit... View... *

Figure 12-29 Binary Partner B2BGW Advanced tab

We have finished configuring the Banking Partner B2B Gateway Service.

Step 5: Creating the Binary Hub B2B Gateway Service

For the Binary Hub case, our B2B Gateway Service will act as the entry point to the hub system, identifying the incoming messages and mapping them with the specific profiles that need to handle them.

In this B2B Gateway configuration, we will handle plain binary files coming from our back end. In this case, we do not have access to Sender and Receiver information, so we will have to use an XSLT stylesheet to run against our transaction. This stylesheet will examine information from transport headers and other non-content sources to select relevant trading partners.

The default sample on the XB60 is in `store:///b2b-routing.xml`, so we make our own copy to `local` and edit it to our needs.

As we can see in Figure 12-30, the Banking Hub B2BGW service has two partners attached: the Banking Hub internal profile and the Banking Partner external profile, but you can add as many profiles as your business needs require.

It also has two Front side handlers, which are used in all other DataPower services, to receive incoming traffic.

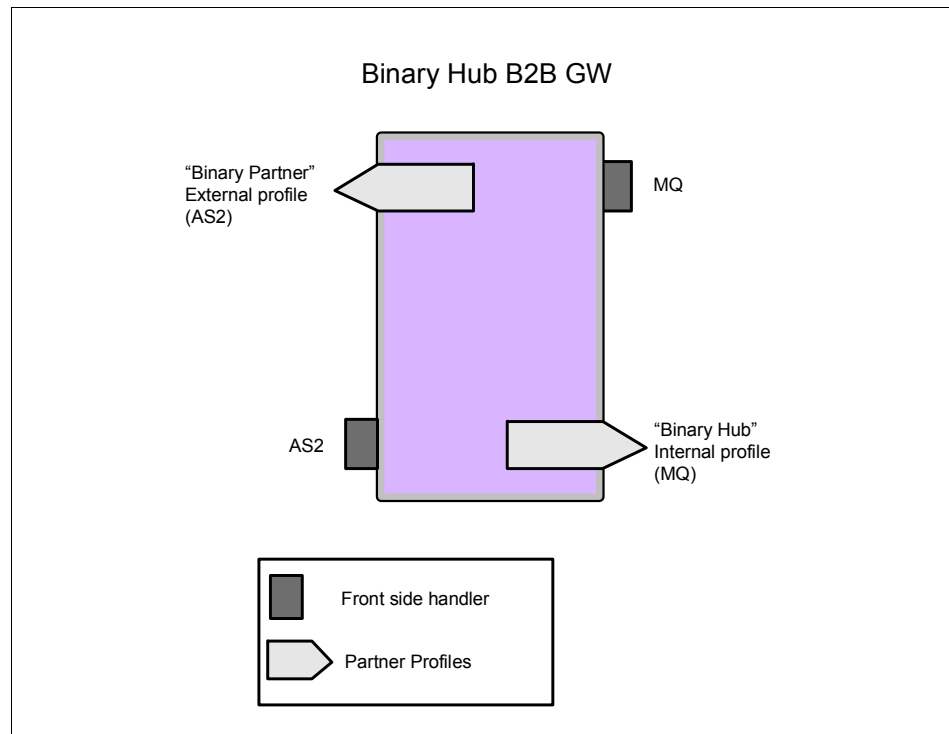


Figure 12-30 Binary Hub B2B Gateway architecture

Contrasting with the previous B2B Gateway, in this particular case, we still need two FSHs: an AS2 FSH is needed for receiving MDNs (and only MDNs, because we only have a one-way flow) coming from the Binary Partner, and also an HTTP to receive messages from our MQ back end.

Figure 12-31 on page 309 shows the Main tab of the B2B Gateway Service.

There are two Partner Profiles (`Binary_Partner` and `Binary_Hub_int`) attached that show in the Attach Partner Profiles section. If you want to add a new Partner

profile, all you have to do is select the profile that you want from the drop-down list and click **Add**; then, select the Profile Destination, if the profile has more than one destination, or leave it as the default if it only has one.

Configure B2B Gateway

Main | Archive | XML Formats | Advanced

B2B Gateway: Binary_Hub_B2BGW [up]

Apply | Cancel | Delete | Undo | Export | View Log | View Status | Arcl

General Configuration

Admin State: enabled disabled

Comments:

Document Storage Location: local:///

XML Manager: default + ... *

Document Routing

Front Side Protocol Handlers

| Front Side Protocol | | |
|---------------------|-------|--|
| Binary_Hub_AS2_FSH | ↑ ↓ ✕ | |
| Binary_Hub_MQFSH | ↑ ↓ ✕ | |
| Add | | |

*

Attach Partner Profiles

Active Partner Profiles

| B2B Partner Profile | Profile Enabled? | Profile Destination | |
|-----------------------|------------------|---------------------|-------|
| Binary_Hub_int | enabled | MQ_2_Backend | ↑ ↓ ✕ |
| Binary_Partner | enabled | AS2_2_BinaryPartner | ↑ ↓ ✕ |
| Banking_Hub + ... Add | | | |

*

Active Profile Groups

| B2B Profile Group | Group Enabled? |
|-------------------|----------------|
| (empty) | |
| + ... Add | |

Figure 12-31 Binary Hub B2BGW Main configuration tab

Figure 12-32 shows the detailed configuration on the AS2 FSH. It is important to notice that we are using localhost as a Host Alias for 127.0.0.1 in order to not tightly couple our Front Side Handler with any specific IP address.

Configure AS2 Front Side Handler

Main

AS2 Front Side Handler: Binary_Hub_AS2_FSH [up]

[Apply](#) [Cancel](#) [Undo](#) [Export](#) | [View Log](#) | [View S](#)

Admin State enabled disabled

Comments

Local IP Address [Select Alias](#) *

Port Number *

HTTP Version to Client

Allowed Methods and Versions

- HTTP 1.0
- HTTP 1.1
- POST method
- GET method
- PUT method
- HEAD method
- OPTIONS
- TRACE method
- DELETE method
- URL with Query Strings
- URL with Fragment Identifiers
- URL with ..
- URL with cmd.exe

Persistent Connections on off

Compression on off

Maximum Allowed URL Length

Maximum Allowed Total Header Length

Maximum Number of HTTP Request Headers Allowed

Figure 12-32 Binary Hub AS2 FSH

Figure 12-33 on page 311 shows that there is an SSL Proxy profile associated to it, because we will use SSL on our communications.

| | |
|--|---|
| | <input checked="" type="checkbox"/> URL with Query Strings |
| | <input checked="" type="checkbox"/> URL with Fragment Identifiers |
| | <input type="checkbox"/> URL with .. |
| | <input type="checkbox"/> URL with cmd.exe |
| Persistent Connections | <input checked="" type="radio"/> on <input type="radio"/> off |
| Compression | <input type="radio"/> on <input checked="" type="radio"/> off |
| Maximum Allowed URL Length | <input type="text" value="16384"/> |
| Maximum Allowed Total Header Length | <input type="text" value="128000"/> |
| Maximum Number of HTTP Request Headers Allowed | <input type="text" value="0"/> |
| Maximum Allowed Length of HTTP Header Name | <input type="text" value="0"/> |
| Maximum Allowed Length of HTTP Header Value | <input type="text" value="0"/> |
| Maximum Allowed Length of HTTP Query String | <input type="text" value="0"/> |
| SSL Proxy | <input type="text" value="Binary_Hub_PP"/> ▼ + ... |
| Access Control List | <input type="text" value="(none)"/> ▼ + ... |
| AAA Policy | <input type="text" value="(none)"/> ▼ + ... |

Figure 12-33 Binary Hub AS2 FSH (continuation)

In Figure 12-34 on page 312, if we explore Binary_Hub_PP, you can see that it is configured as a **Reverse** Proxy Profile, because the front side handling will be the actual SSL server on the connections that will be received. Therefore, you need to configure a Reverse Crypto profile, Binary_Hub_CP.

Figure 12-34 Binary Hub AS2 FSH SSL Proxy Profile configuration

Figure 12-35 on page 313 shows the Binary_Partner_CP. We need to include the ID credentials that this B2B Gateway Service will present to the incoming clients, so that the SSL connection can be established. We use the Binary_Partner_ID Cred object that we created in “Step 1: Creating all the necessary crypto objects” on page 279.

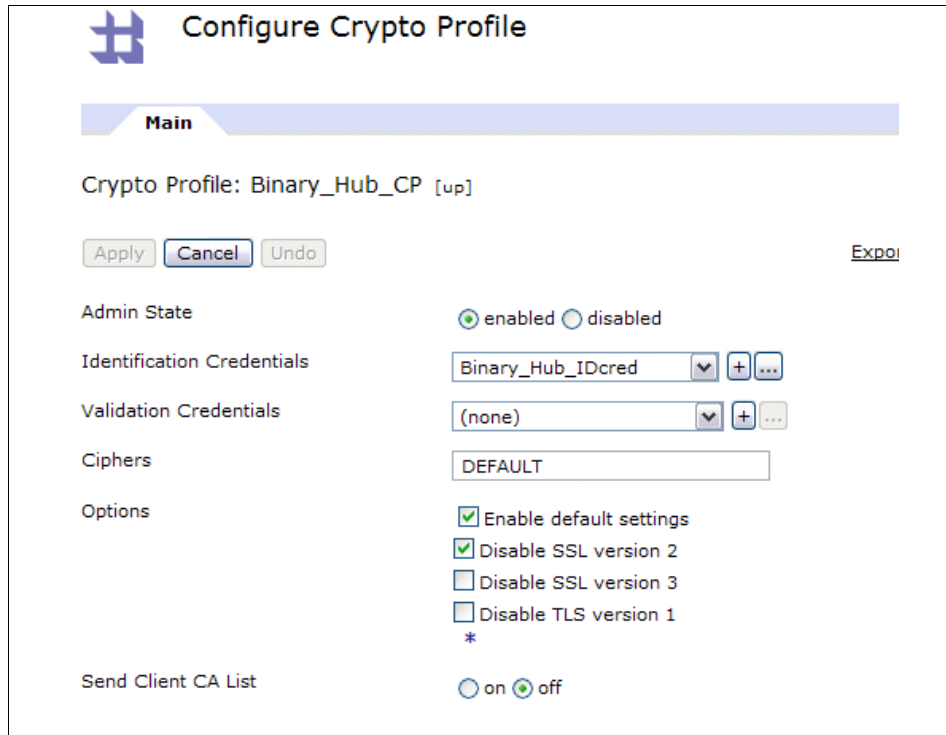


Figure 12-35 Binary Hub AS2 FSH Crypto Profile configuration

On the Archive configuration tab (Figure 12-36 on page 314), we have set the archiving up as **Purge only** and entered 3 days of Archive Document Age, even though, depending on the use case, this property might need to be resized.

Configure B2B Gateway

Main **Archive** XML Formats Advanced

B2B Gateway: Binary_Hub_B2BGW [up]

Apply Cancel Delete Undo [Export](#) | [View Log](#) | [View St](#)

Archive Mode *

Archive Document Age Days

Disk Use Check Interval Minutes

Maximum Disk Usage for Documents Kilobytes

Figure 12-36 Binary Hub B2BGW Archive configuration tab

There are no XML Formats (Figure 12-37 on page 315), because we are trading with binary files.

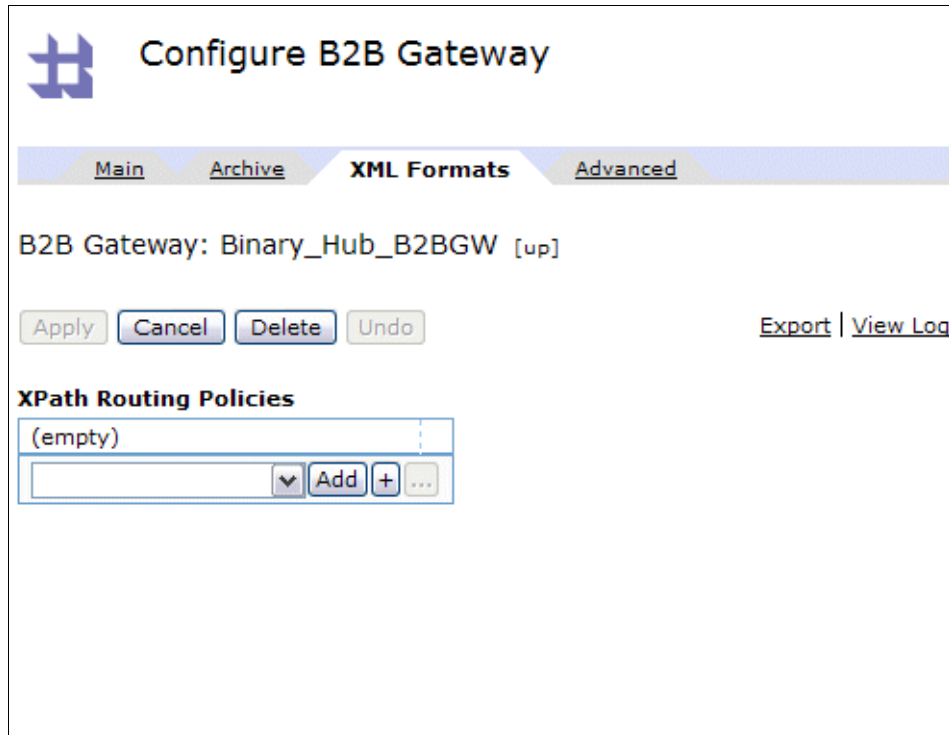


Figure 12-37 Binary Hub B2BGW XML Formats tab

Figure 12-38 on page 316 shows the Advanced configuration tab. We need to set the Default AS2 MDN Return Path. This information comes in the AS2 headers when the partner sends any AS2 messages to the Hub, so the Hub knows where, by default, it sends the AS2 MDNs back, which is why we are indicating the port number that matches the Partner's AS2 Front Side Handler.

In addition, we have also copied the `Binary_routing.xsl` to our local directory and renamed it as `Binary_hub_B2B-routing.xsl`. That way, we do not modify the original XSL file that is provided in case, in the future, we want to start from the beginning for another use case.

Configure B2B Gateway

Main Archive XML Formats **Advanced**

B2B Gateway: Binary_Hub_B2BGW [up]

Apply Cancel Delete Undo Export View Log View Status Archive/pt

Service Priority Normal

Default AS2 MDN Return Path https://
127.0.0.1:20061

Default AS3 MDN Return Path ftp://

Document Routing Preprocessor local://
Binary_hub_b2b-routing_.xsl Upload... Fetch... Edit...

Figure 12-38 Binary Hub B2BGW Advanced tab

Example 12-1 shows how we configured that XSLT. For our specific case, we have coded it so that whenever a message comes via MQ, then we set up the business ID's in the stylesheet. Therefore, if we had non-binary partners coming from HTTP, they are not affected by this rule. Again, this case is only an example, and a more convenient filter can be coded, depending on your particular needs.

Example 12-1 Binary Hub B2B Routing xslt used to assign Partner IDs on binary payloads

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
```

DataPower XB60 B2B Document Routing Preprocessor Stylesheet

Licensed Materials - Property of IBM
 IBM WebSphere DataPower Appliances
 Copyright IBM Corporation 2008. All Rights Reserved.
 US Government Users Restricted Rights - Use, duplication or disclosure
 restricted by GSA ADP Schedule Contract with IBM Corp.

-->

```

<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:dp="http://www.datapower.com/extensions"
    extension-element-prefixes="dp">
  <xsl:template match="/">

    <xsl:variable name="protocol"
      select="dp:variable('var://service/protocol')"/>
    <xsl:message dp:type="xslt" dp:priority="info">
      ****Value of protocol <xsl:value-of select="$protocol" />****
    </xsl:message>

    <xsl:choose>
      <!-- If the message comes from MQ, then we will set the the values of the
Partners in order to route the message-->

      <xsl:when test="$protocol='dpmq'">
        <dp:set-variable name="'var://service/b2b-doc-type'"
          value="'binary'"/>

        <dp:set-variable name="'var://service/b2b-partner-from'"
          value="'binaryhub'"/>

        <dp:set-variable name="'var://service/b2b-partner-to'"
          value="'binarypartner'"/>
        <xsl:message dp:type="xslt" dp:priority="error">
          ****PARTNER INFORMATION****
          TO: <xsl:value-of select="dp:variable('var://service/b2b-partner-to')"/>
          FROM:<xsl:value-of select="dp:variable('var://service/b2b-partneR-from')"/>
          ****
        </xsl:message>
      </xsl:when>
      <xsl:otherwise>
        <!-- By default, do nothing. This will autodetect the
message type, but binary messages will fail for lack of
partner IDs. -->
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>

```

After uploading that XSLT (that you can modify with any XML editor available in the market), we have successfully created the Banking Hub B2B Gateway Service.

12.5 Testing our solution

Everything has been successfully configured and the infrastructure is “up and running,” so now is the time to test our scenario and actually see the transaction results in our Transaction Viewer and the system logs.

In Figure 12-39, you can see the steps that will occur when we trigger the scenario putting the ERP message in Q8.

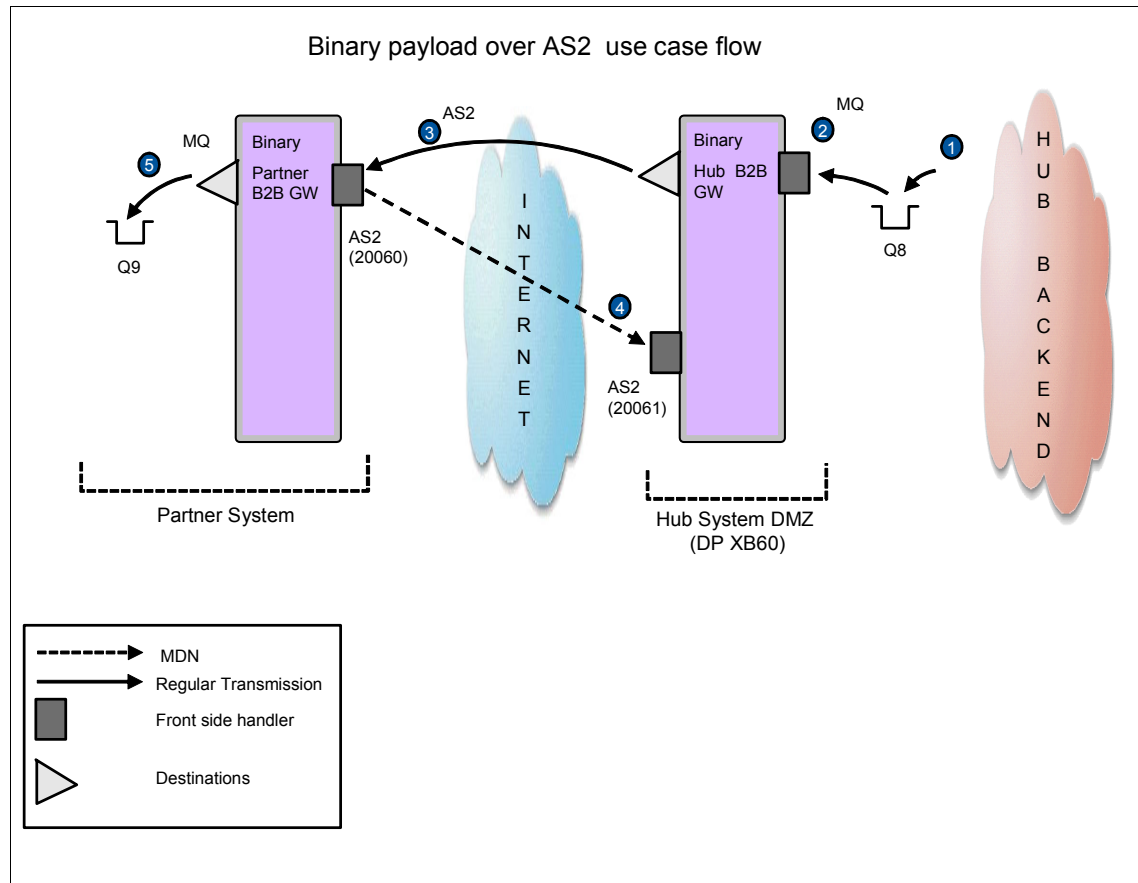


Figure 12-39 Using AS2 and binary payload flow

Here is an explanation of the numbered steps in Figure 12-39:

1. A message (jpg) is put on Q8 with RfhUtil, where MQ Front Side Handler is listening (Figure 12-40 on page 319).

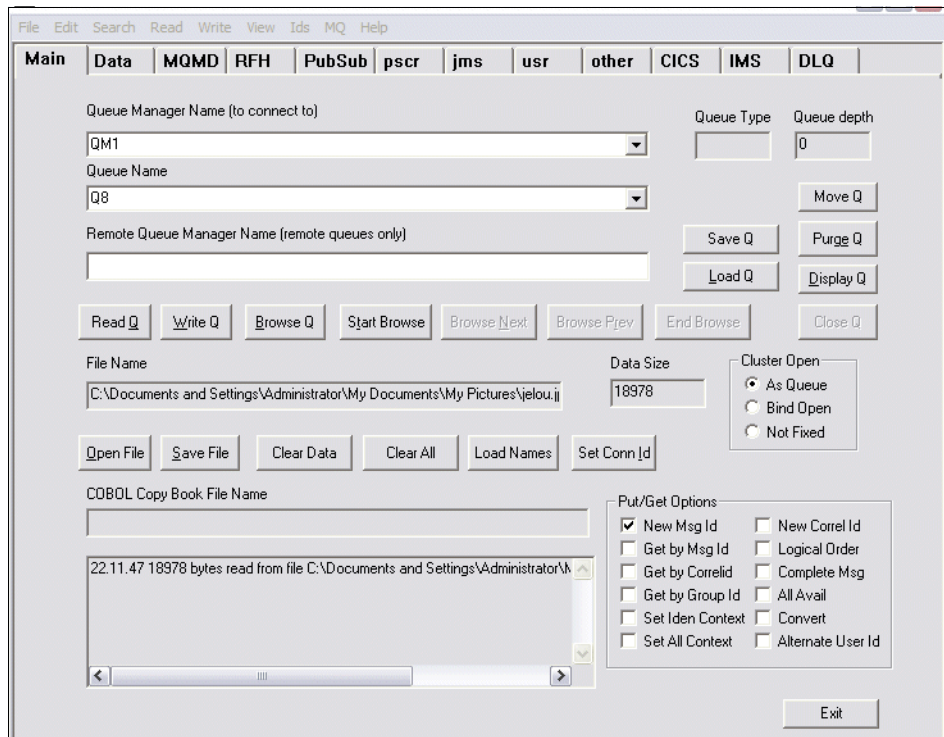


Figure 12-40 Putting jpg file in Q8 using RfhUtil

2. The message is received by the MQ FSH, and the stylesheet B2B_routing.xsl is triggered, because the incoming protocol is MQ, it sets both Sender and Receiver IDs, so B2B Gateway can point to the right profiles.
3. The message is wrapped in AS2, signed, encrypted, and sent over SSL to Binary_Partner_B2BGW using the BinaryPartner profile (AS2_2_bankingpartner destination, on port 20060).
4. Binary_Partner_B2BGW receives the AS2 message using Binary_Partner_AS2FSH. It decrypts it and verifies the signature. If everything is fine, it sends an asynchronous MDN through 20061 to Binary_Hub_B2BGW.
5. The message is processed and then sent back to its MQ Destination, on Q9.

12.5.1 Test results

After completing step 1, the rest of the steps are triggered and the processing finishes with a message in Q9, so let us look at the Queue using RfhUtil utility (Figure 12-40).

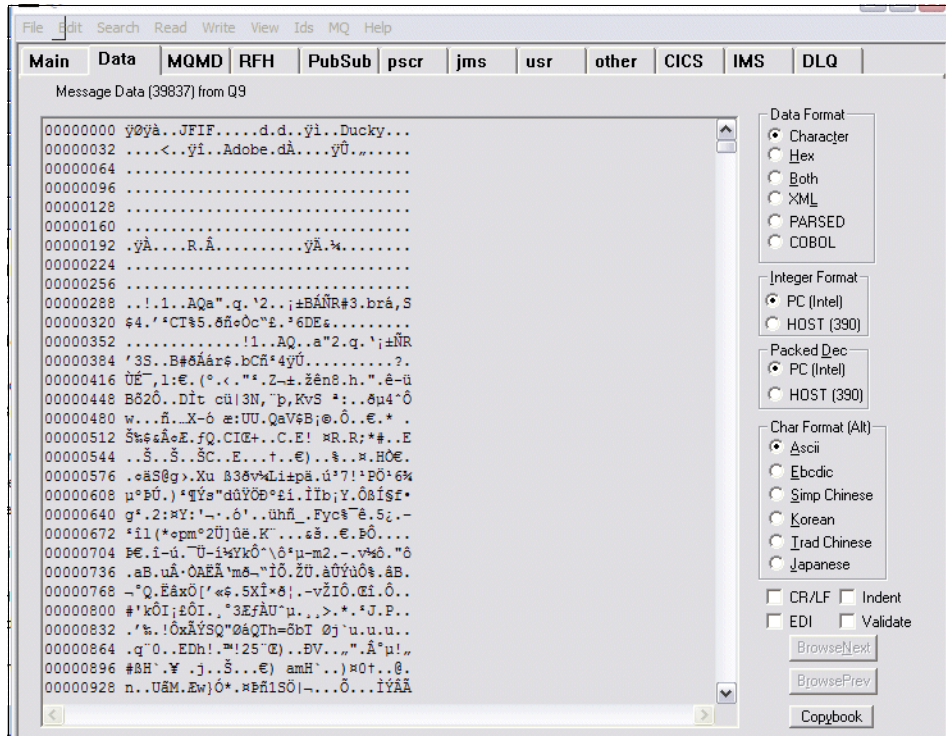


Figure 12-41 Jpg file in the outbound Q9 after the processing

In the Transaction Viewer (Figure 12-42 on page 321), we can see three transaction IDs: 444 corresponds to the BinaryHub gateway processing, 445 corresponds to the Binary Partner receiving the message, and 446 corresponds to the asynchronous MDN sent by the Binary Partner.

| Modify Query Refresh | | | | | | | | | | | | | | | | |
|--|----------------|----------------------|--|---|--|-------------|---------------------|-----------------------|-----------------------|----------------|-------------|---------------|-----|------------------------|--|--|
| Transaction Set ID | Transaction ID | Gateway Name | SenderName (ID) / Receiver (ID) | Inbound URL / Outbound URL | Input Time / Output Time | Result Code | MDI Status | MDI Time | MDI Received | Headers | Document ID | Document Type | MDI | Action | | |
| <input type="checkbox"/> 446 | 405415 | Binary_Hub_B2BGW | Sender: binarypartner Receiver: binaryhub | as2s://127.0.0.1:20061/ | 2009-03-25 09:59:45.0 | Success | Positive | | | (Show Headers) | | | 1 | Resend | | |
| <input type="checkbox"/> 445 | 109443 | Binary_Partner_B2BGW | Sender: binaryhub (binaryhub) Receiver: binarypartner (binarypartner) | as2s://127.0.0.1:20060/ dpmq://QM1?RequestQueue=Q9 | 2009-03-25 09:59:45.0 2009-03-25 09:59:45.0 | Success | Sent (Positive) | 2009-03-25 09:59:45.0 | | (Show Headers) | | | 0 | Resend | | |
| <input type="checkbox"/> 444 | 172482 | Binary_Hub_B2BGW | Sender: binaryhub (binaryhub) Receiver: binarypartner (binarypartner) | dpmq://QM1/Binary_Hub_MQFSH?RequestQueue=Q8 as2s://127.0.0.1:20060 | 2009-03-25 09:59:45.0 2009-03-25 09:59:45.0 | Success | Received (Positive) | | 2009-03-25 09:59:45.0 | (Show Headers) | | | 0 | Resend | | |

Figure 12-42 Transaction Viewer showing the trading flow

If we look at the Headers (Figure 12-43), we can see all the important information gathers there, such as the Content Type and the AS From Header and AS To Header AS2 headers.

| Header Data | |
|----------------------------|---|
| Message ID Header | 7674437c-6611-40e3-a8d2-a8637d912eb4@127.0.0.1 |
| Content Type Header | application/octet-stream |
| AS From Header | binaryhub |
| AS To Header | binarypartner |
| Date Header | Wed, 25 Mar 2009 09:59:45 GMT |
| Disposition Header | https://127.0.0.1:20061 |
| Disposition Options Header | signed-receipt-protocol=optional, pkcs7-signature; signed-receipt-micalg=optional, sha1,md5 |
| Content Length Header | 42310 |
| Content Disposition Header | |
| Original Message ID Header | |

Figure 12-43 AS2 Headers of the binary message sent



Trading binary documents using a Multi-Protocol Gateway service

This chapter describes trading binary documents between internal and external trading partners. The business-to-business (B2B) gateway natively understands ANSI X12 and EDIFACT documents and has special handling for XML-formatted electronic data interchange (EDI) messages. The B2B Gateway (B2BGW) handles all other messages as binary messages.

13.1 Business value

Trading binary content between B2B partners is an important part of conducting business in the B2B world. Increasingly, partners demand content that is not standard EDI, for instance, for medical images or video content in the entertainment industry.

Additionally, the XB60 B2BGW often receives these binary messages via a protocol, such as WebSphere MQ that does not have the same routing headers as AS2 or AS3. For inbound binary messages (messages sent to a B2BGW object) of this format, the DataPower Multi-Protocol Gateway (MPGW) object can aid in the transformation and routing of these types of messages.

Requirements include:

- ▶ Ability to determine the routing destination to a B2B Gateway
- ▶ Ability to transform metadata to a known header format: AS2 or AS3
- ▶ Ability to dynamically route a message through the B2BGW via XSLT
- ▶ Ability to send and receive messages over HTTP or FTP

13.2 Prerequisites

To implement this scenario, you must have the following software and skills.

13.2.1 Software prerequisites

In order to be able to run this scenario, you must have installed the following components:

- ▶ WebSphere DataPower B2B Appliances XB60
- ▶ WebSphere MQ V6

13.2.2 Skills prerequisites

This scenario is intended for the intermediate user, meaning that in order to be able to fully implement and understand this scenario you must be familiar with:

- ▶ WebSphere DataPower B2B Appliances XB60 main concepts
- ▶ Basic Extensible Stylesheet Language Transformation (XSLT) techniques

13.3 Presenting the Binary AS2 over HTTP multi-step use case

This section outlines a specific use case for sending binary messages that are not wrapped by the AS2 or AS3 protocol. The B2B Gateway often receives these messages via a protocol, such as WebSphere MQ, HTTP, or FTP, that do not have the same routing headers as AS2 or AS3.

These inbound messages must (currently) be processed using a multi-step process. This multi-step process allows messages to be transformed that contain customized header information in order to route these messages properly to the B2BGW. The Multi-Protocol Gateway object allows us to provide this processing step, sort of a preprocessor step.

This use case consists of two Trading Partner systems, Partner Binary System and Hub Binary System. The Hub Binary System consists of a hub application, which runs on a server in the hub owner's enterprise. The Hub Binary Application is associated with the Hub Binary MPGW in the Hub Binary enterprise. The Partner Binary System consists of a partner application, which runs on a server in the partner's enterprise. The partner application is associated with the Partner B2B Gateway (B2BGW) in the Partner Binary enterprise.

Figure 13-1 on page 326 shows HTTP To AS2 binary processing transaction flow.

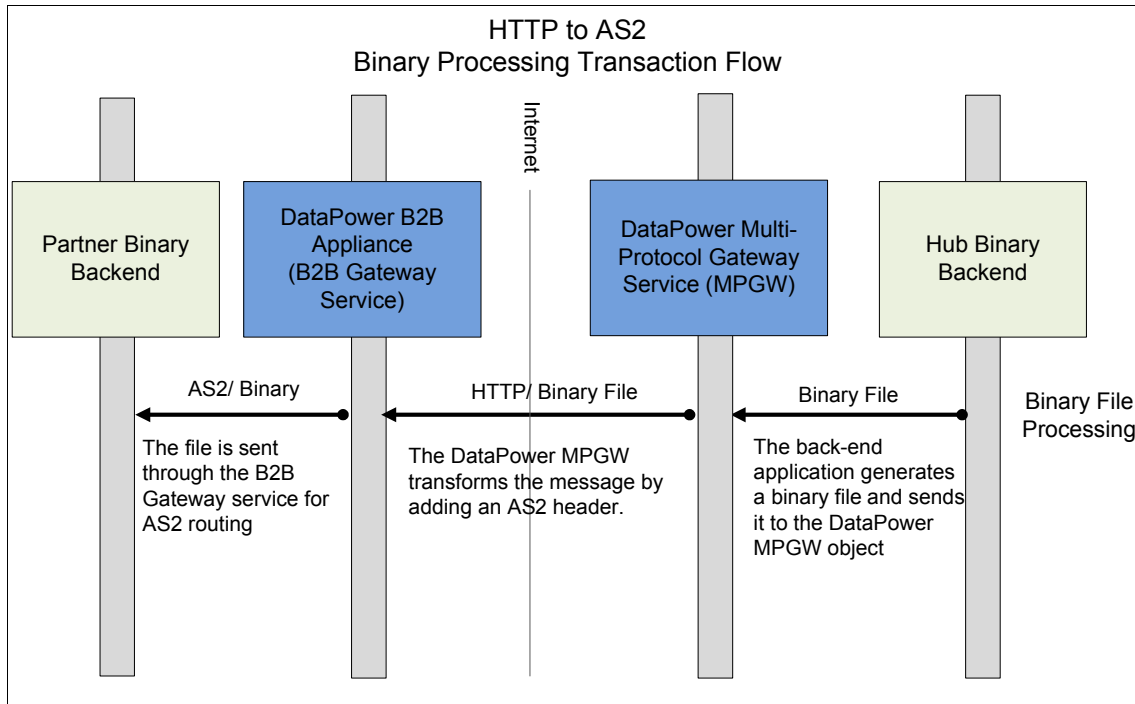


Figure 13-1 HTTP To AS2 binary processing transaction flow

13.4 Binary AS2 over HTTP multi-step use case solution

Figure 13-2 on page 327 outlines the Binary over HTTP to AS2 architecture.

These steps outline the processing flow:

1. The binary message is sent by HubBinary back-end application to the MQ queue.
2. The binary message is fetched from the queue and transformed by a MPGWS object to include AS2 headers and transmitted over HTTP to a B2BGW (refer to Figure 13-1).
3. The AS2 wrapped binary message is sent to PartnerBinary B2B Gateway over HTTP.
4. The message is received by B2BGW AS2 Front Side Handler, and the AS2 headers are processed to obtain routing information.
5. The Partner profile is read and the appropriate destination is selected based on the message type.

- The binary message is sent to the PartnerBinary back-end application via placement on MQ queue.

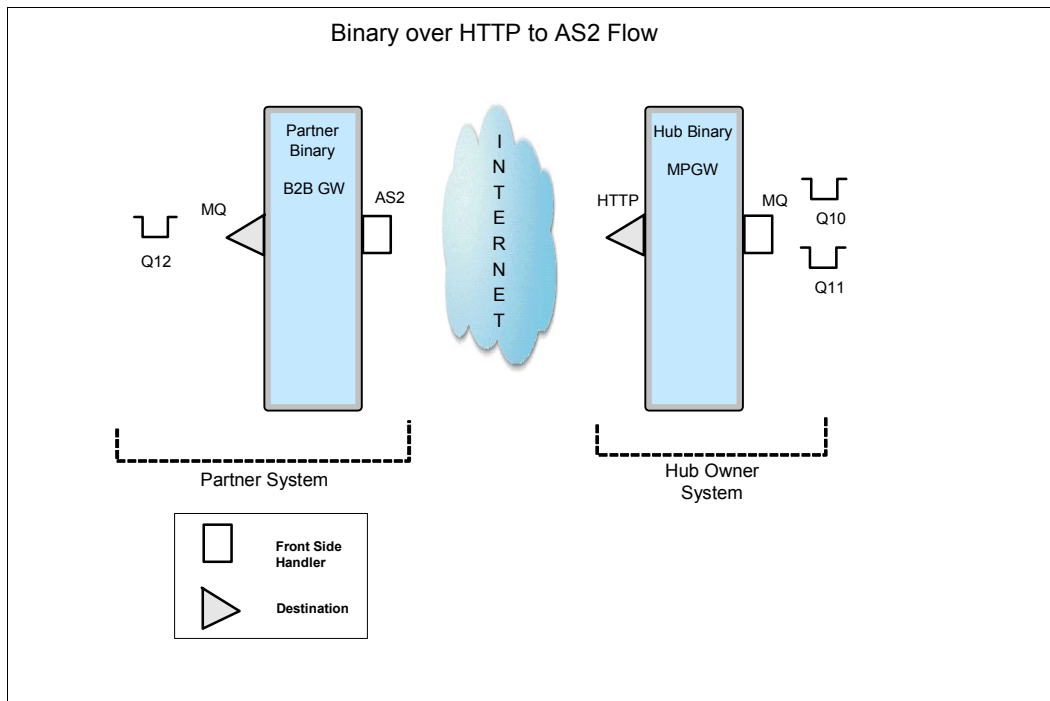


Figure 13-2 Binary over HTTP to AS2 architecture

13.4.1 Use case outline

Here is a summary of the steps to configure the appropriate processing objects to implement the scenario:

- ▶ Step 1: Creating the Partner Binary partner profile
- ▶ Step 2: Creating the Partner Binary B2B Gateway
- ▶ Step 3: Creating the Hub Binary MPGW
- ▶ Step 4: Creating the Multi-step processing policy

13.4.2 Use case implementation

We describe each step in detail.

Step 1: Creating the Partner Binary partner profile

In this section, we define the partner profiles that are necessary to allow the information interchange to occur with a specific partner. We define two separate partner profiles to represent the trading entities: an internal profile that represents the PartnerBinary back end and an external profile that represents the HubBinary back end. These profiles will ultimately be contained in the PartnerBinB2BGW object that will point to the HubBinary MPGW service.

PB_int Internal profile

The `PB_int` internal partner profile defines the object that will manage the messages sent to the PartnerBinary B2BGW. The internal profile defines, among other things, the business IDs associated with the profile and the destination route for the binary documents that are received by the B2BGW.

On the Main tab (refer to Figure 13-3 on page 329), the following items need to be defined:

- ▶ Object name
- ▶ Profile Type
- ▶ Partner Business IDs

Configure B2B Partner Profile

Main AS Security Destinations Contacts

B2B Partner Profile: PB_Int [up]

Apply Cancel Undo

Admin State enabled disabled

Comments

Profile Type External Internal

Partner Business IDs

partnerbin

*

Figure 13-3 PB_int partner profile Main configuration tab

We do not use any AS Security in this scenario, so we do use that configuration tab.

On the Destinations tab, we define the destination that is available for a binary message that is sent to the PB_int profile. Figure 13-4 on page 330 shows the configuration of the PartnerBin_MQ destination for the PB_int internal partner profile. This destination object sends binary documents to the Q12 queue associated with the XB60 queue manager object.

Only binary documents will be routed to this queue. Because this destination is the only destination that is defined for binary documents, all binary documents will route to this queue.

B2B Partner Profile: PB_Int [up]

Apply Cancel Undo Export

Destinations

| Destination Name | Destination URL | Enabled Document Type |
|-------------------------|-------------------------------|-----------------------|
| PartnerBin_MQ (default) | dpmq://XB60/?RequestQueue=Q12 | Binary |

Destinations

Destination Name *

Enabled Document Type

- XML
- X12
- EDIFACT
- Binary

Connection

Destination URL *

Connection Timeout Seconds

Apply Cancel

Figure 13-4 PB_int internal profile Destinations configuration tab

The Contacts tab is optional so we do not make any configuration changes using that tab for this scenario.

HB_Ext external profile

The HB_Ext external partner profile defines the object that will manage the messages sent to the HubBinary MPG. The external profile defines, among other things, the business IDs associated with the profile and the destination route for the binary documents that are received by the MPG.

On the Main tab (refer to Figure 13-5 on page 331), the following items need to be defined:

- ▶ Object name
- ▶ Profile Type
- ▶ Partner Business IDs

The screenshot shows a web-based configuration interface titled "Configure B2B Partner Profile". At the top, there are four tabs: "Main", "AS Security", "Destinations", and "Contacts". The "Main" tab is currently selected. Below the tabs, the profile name is "B2B Partner Profile: HB_Ext [up]". There are three buttons: "Apply", "Cancel", and "Undo". The "Admin State" is set to "enabled" (radio button selected). The "Comments" field is empty. The "Profile Type" is set to "External" (radio button selected). The "Partner Business IDs" field contains the text "hubbin" and has an "Add" button next to it. A small asterisk "*" is visible below the "Partner Business IDs" field.

Figure 13-5 HB_Ext partner profile

We do not use any AS Security in this scenario, so we do not use that tab.

On the Destinations tab (refer to Figure 13-6 on page 333), we define the destination that is available for a binary message that is sent to the HB_Ext profile. This tab shows the configuration of the HubBin_FTP destination for the HB_Ext external partner profile. This destination object sends binary documents to the FTP Server Front Side Handler listening on port 5117 on the HubBin MPGW.

Important: The Destination object defined in an External Partner Profile is responsible for routing messages from the current B2B Gateway to the destination outlined in the Destinations tab.

The Destination outlined here will not be used for the Binary HTTP multi-step use case; however, it will be used in the FTP multi-step use case in 13.6, “Presenting the binary FTP multi-step use case” on page 349.

Only binary documents will be routed to the HubBin MPGWS over FTP, because this destination is the only destination defined for binary documents.

Main AS Security **Destinations** Contacts

B2B Partner Profile: HB_Ext [up]

Apply Cancel Undo [Export](#) | [View Log](#) |

Destinations

| Destination Name | Destination URL | Enabled Document Type | |
|----------------------|----------------------------------|-----------------------|---------|
| HubBin_FTP (default) | ftp://userid:pwd7@127.0.0.1:5117 | Binary | ↑ ↓ ✎ ✕ |

Destinations

Destination Name *

Enabled Document Type

- XML
- X12
- EDIFACT
- Binary

Connection

Destination URL *

SSL Proxy Profile + ...

Connection Timeout Seconds

Enable Advanced AS3/FTP Settings

User name

Password

Figure 13-6 HB_Ext external profile Destinations configuration tab

Figure 13-7 on page 334 shows extra fine-tuning parameters available for AS3/FTP destinations. We have defined our connection to use Passive Mode exclusively (**Require Passive Mode**). We have set the Data Type as **Image (Binary) Data**.

Advanced AS3/FTP Settings

Passive Mode

Encrypt Command Connection

Data Type

Write Unique Filename if Trailing Slash

Quoted Commands

Size Check

Figure 13-7 HB_Ext external profile Advanced AS3/FTP Settings

Step 2: Creating the Partner Binary B2B Gateway

The B2B Gateway object is responsible for defining the characteristics of the transaction and the trading partners in the transaction. Characteristics, such as Front Side Handler objects, partner profiles, and transaction archive parameters, will be covered in this section.

Figure 13-8 on page 335 details the Main tab for the PartnerBinaryB2BGW object. Here, we associate partner profiles PB_Int (internal profile) and HB_Ext (external profile) to the B2BGW object.

B2B Gateway: PartnerBinaryB2BGW [up]

Apply Cancel Delete Undo Export | View

General Configuration

Admin State enabled disabled

Comments

Document Storage Location ▼

XML Manager ▼ *

Document Routing

Front Side Protocol Handlers

| Front Side Protocol | |
|------------------------------------|-------|
| PartnerBin_as2_fsh | ↑ ↓ ✕ |
| PartnerBinary_mq | ↑ ↓ ✕ |
| <input type="button" value="Add"/> | |

*

Attach Partner Profiles

Active Partner Profiles

| B2B Partner Profile | Profile Enabled? | Profile Destination | |
|--|------------------|---------------------|-------|
| PB_Int | enabled ▼ | PartnerBin_MQ ▼ | ↑ ↓ ✕ |
| HB_Ext | enabled ▼ | HubBin_FTP ▼ | ↑ ↓ ✕ |
| <input type="text" value="CompanyA_Ext"/> ▼ <input type="button" value="+"/> <input type="button" value="..."/> <input type="button" value="Add"/> | | | |

*

Figure 13-8 Configuration of PartnerBinaryB2BGW gateway object

Archive settings define how long transaction documents will be held on the XB60 hard drive and the action that will taken on those messages for archive purposes. For this scenario, we set the transactions to be purged after 5 days (refer to Figure 13-9 on page 336).

Configure B2B Gateway

Your changes were applied successfully

Main **Archive** XML Formats Advanced

B2B Gateway: PartnerBinaryB2BGW [up]

Apply Cancel Delete Undo [Export](#) | [View Log](#)

Archive Mode *

Archive Document Age Days

Disk Use Check Interval Minutes

Maximum Disk Usage for Documents Kilobytes

Figure 13-9 Configure the B2BGW Archive tab

Creating PartnerBin_as2_fsh AS2 Front Side Handler

The Front Side Handler objects are responsible for listening on a particular port and retrieving messages in the defined protocol. The AS2 Front Side Handler object is called PartnerBin_as2_fsh (refer to Figure 13-10 on page 337).

This AS2 Front Side Handler represents the listener object that receives the binary messages that are sent from the HubBinary MPGW over HTTP.

Configure AS2 Front Side Handler

Main

AS2 Front Side Handler: PartnerBin_as2_fsh [up]

Apply Cancel Undo Export

Admin State enabled disabled

Comments

Local IP Address Select Alias

Port Number *

HTTP Version to Client ▼

Allowed Methods and Versions

- HTTP 1.0
- HTTP 1.1
- POST method
- GET method

Figure 13-10 Configuring the AS2 Front Side Handler object

Step 3: Creating the HubBinary Multi-Protocol Gateway

Figure 13-11 on page 338 displays the configuration window for the HubBinary MPGW object. The following objects will need to be configured: Front Side Protocol, Processing Policy, and Type. The configuration of these objects will be shown in subsequent sections.

The Front Side Handler (FSH) object is responsible for receiving the binary messages sent from the back end via MQ and passes the message to the Processing Policy. The Processing Policy receives the message and applies, in a step-wise fashion, the configured actions in the selected processing rule. The Processing Rule routes the message to the HTTP URL destination.

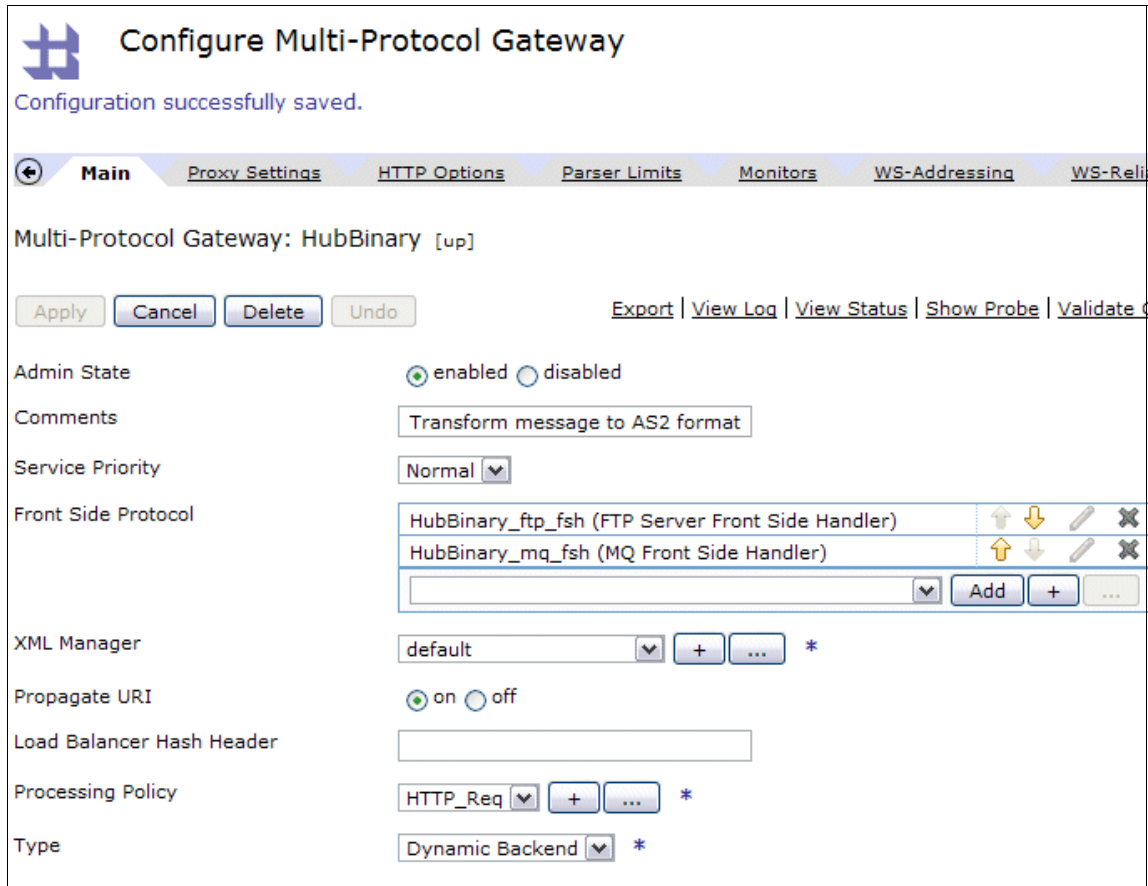


Figure 13-11 HubBinary Multi-Protocol Gateway Main tab

The Proxy Settings tab in Figure 13-12 on page 339 displays extra settings provided by the HubBinary MPG. The Client Traffic Type is set to **Non-XML** to allow binary messages, and the Server Traffic Type is set to **Passthru**.

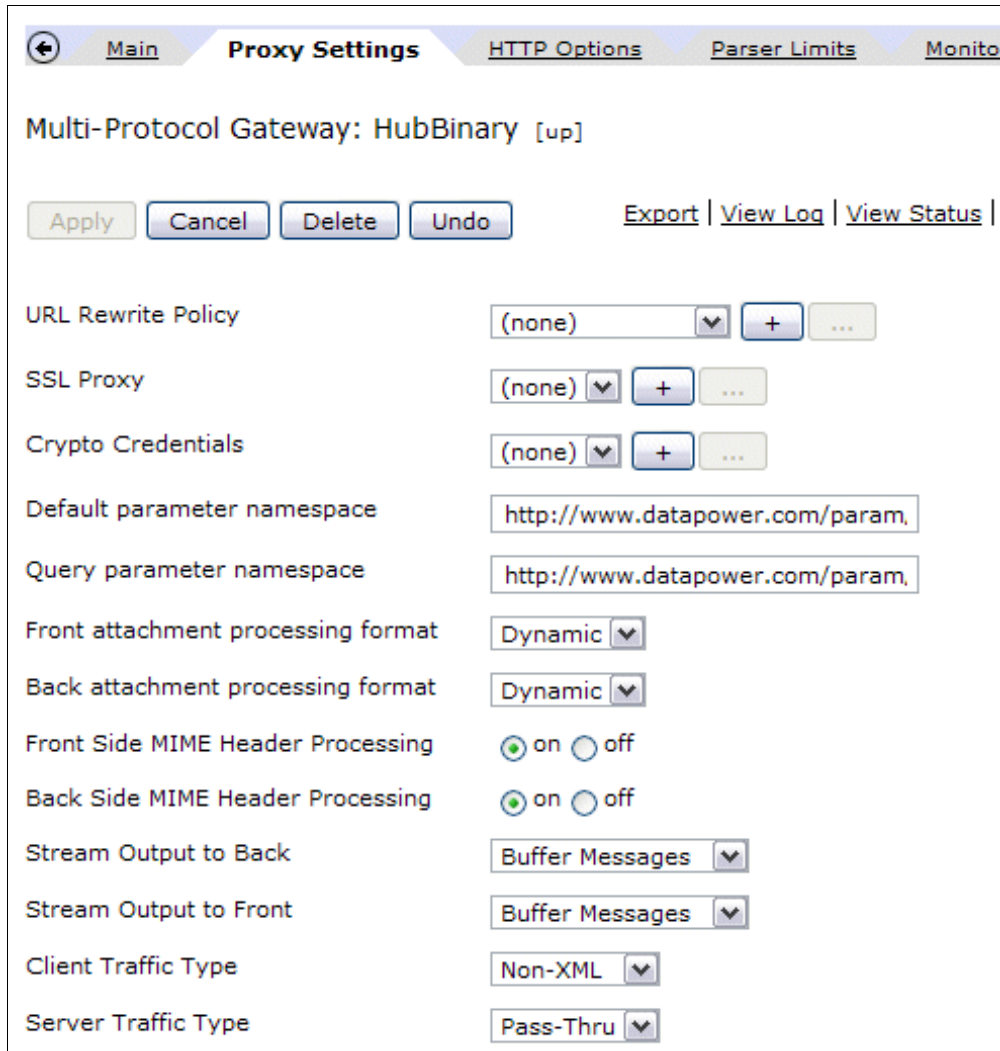


Figure 13-12 HubBinary MPGW Proxy Settings tab

Defining HubBinary MQ Front Side Handler

Figure 13-13 on page 340 shows the configuration of the MQ Front Side Handler object that is responsible for receiving the incoming message from the Hub Binary application sent over MQ. The message received from the HubBinary_mq_fsh will be passed to the multi-step processing rule for transformation and routing.

Figure 13-13 Hub Binary MQ Front Side Handler

Step 4: Creating the inbound AS2 Header values

In order to send an inbound binary message to the B2BGW over HTTP, we need to wrap the message in an AS2 Header. We will take advantage of the multi-step processing capability in the MPGW to create the AS2 header values.

Injecting AS2 header values

The HTTP Header Injection tab (refer to Figure 13-14 on page 341) of the MPGW service allows us to create header values dynamically for each binary message transaction. We will use these values to define several of the required AS2 header values, specifically, the Host and AS2-From values. The Host value identifies the current server sending the message, and the AS2-From value identifies the Business ID associated with the Sender.

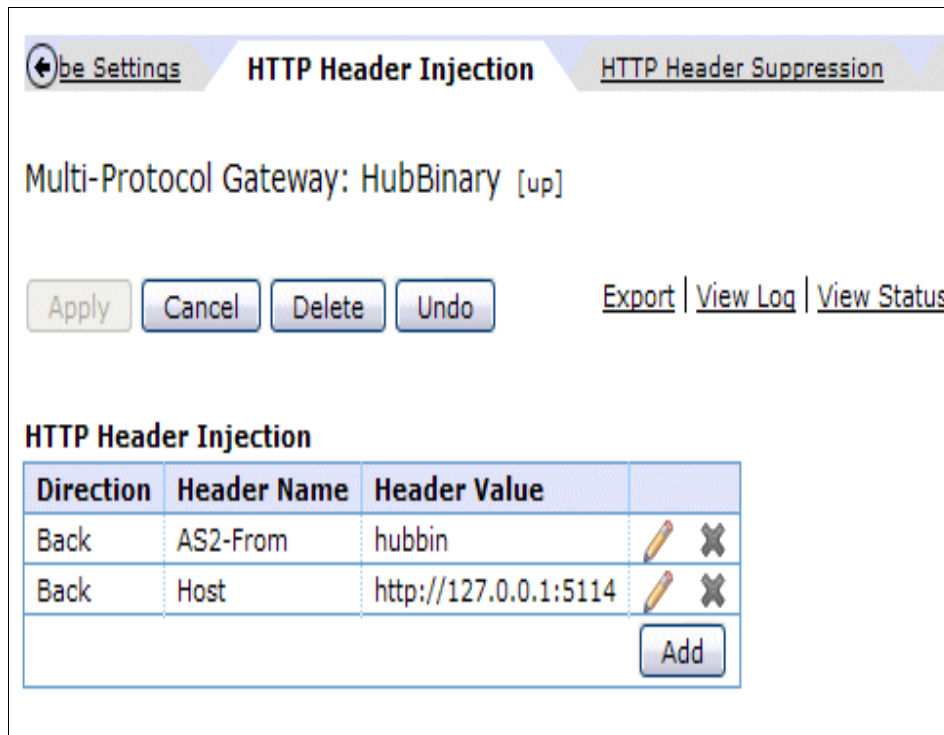


Figure 13-14 HTTP Header Injection values for the AS2 Header

MPGW HTTP to AS2 Processing Policy

The MPGW service allows us to define a multi-step processing policy that is made up of one or more processing rules. Each rule is made up of one or more processing actions. It is these actions that we will leverage to create the AS2 headers that are needed by the B2BGW Service.

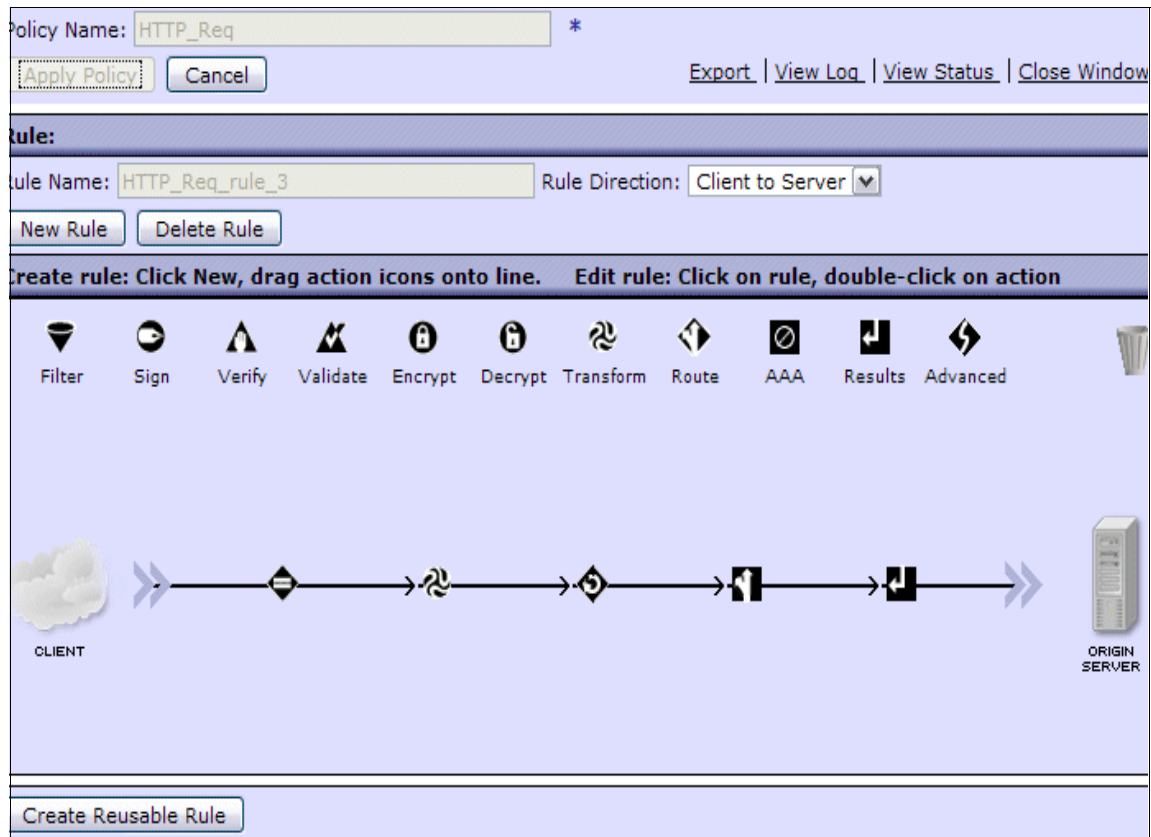


Figure 13-15 Hub Binary processing policy

Transform action

The `b2b-test-routing-mq.xml` defines the message route as well as defining several of the necessary AS2 header values. The code sets:

- ▶ 'AS2-To' value
- ▶ 'var://service/routing-url' - the route to the B2BGW
- ▶ 'Message-ID'

The code snippet in Example 13-1 checks the protocol of the message received and retrieves the partner URL.

Example 13-1 Protocol checking and partner URL retrieval code snippet

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
```

Licensed Materials - Property of IBM
 IBM WebSphere DataPower Appliances
 Copyright IBM Corporation 2008. All Rights Reserved.
 US Government Users Restricted Rights - Use, duplication or disclosure
 restricted by GSA ADP Schedule Contract with IBM Corp.

```
-->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  xmlns:regex="http://exslt.org/regular-expressions"
  extension-element-prefixes="dp">
  <xsl:template match="/">
    <!-- Do queue-based routing if MQ, default routing otherwise. -->
    <xsl:variable name="protocol" select="dp:variable('var://service/protocol')"/>
    <xsl:choose>
      <xsl:when test="$protocol='dpmq'">
        <!-- We are MQ. Fetch the routing table: -->
        <xsl:variable name="routing"
select="document('local:///b2b-routing-mq-xb60-test.xml')"/>
        <!-- Pull the request queue out of the inbound URL: -->
        <xsl:variable name="url" select="dp:variable('var://service/URL-in')"/>
        <xsl:variable name="queue" select="substring-after($url,'RequestQueue=')"/>
        <!-- Find the corresponding route in the routing table. -->
        <xsl:variable name="route" select="$routing/routing/partner[@queue=$queue]"/>
        <!-- If we found a route, great; if not, then fall back to default behavior.
-->
        <xsl:choose>
          <xsl:when test="$route">
            <dp:set-request-header name="AS2-To" value="$route/@partner"/>
            <dp:set-variable name="'var://service/routing-url'"
value="$route/@path"/>
            <xsl:variable name="uuid" select="dp:generate-uuid()"/>
            <dp:set-request-header name="Message-ID" value="$uuid"/>
          </xsl:when>
          <xsl:otherwise>
            <xsl:message dp:priority="error">
              <xsl:text>Could not find route for Queue </xsl:text>
              <xsl:value-of select="$queue"/>
            </xsl:message>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:when>
      <xsl:otherwise>
        <!-- By default, do nothing. This will autodetect the
```

```

        message type, but binary messages will fail for lack of
        partner IDs. -->
    </xsl:otherwise>
</xsl:choose>
</xsl:template>
</xsl:stylesheet>

```

Defining the Header Rewrite Action

The Header Rewrite action allows us to use the URL Rewrite Policy (refer to Figure 13-16) to configure the content-type header value. This value will be used by the B2BGW to determine the type of the incoming message.

Configure URL Rewrite Policy

Main | **URL Rewrite Rule**

URL Rewrite Policy: SetContentType [up]

Apply | Cancel | Undo | [Export](#) | [View Log](#) | [View Status](#)

URL Rewrite Rule

| URL Rewrite Type | Match Expression | Input Replace Expression | Stylesheet Replace Expression | Input URL Unescape | Stylesheet URL Unescape | Header Name |
|------------------|------------------|--------------------------|-------------------------------|--------------------|-------------------------|-------------|
| content-type | .* | application/octet-stream | | off | on | none |

Figure 13-16 Configuring the content-type in the URL Rewrite Policy

Route action

The Route action uses the value of the `var://service/routing-url`, which was defined in the Transform action, to send the message to its destination in the B2BGW (refer to Figure 13-17 on page 345).

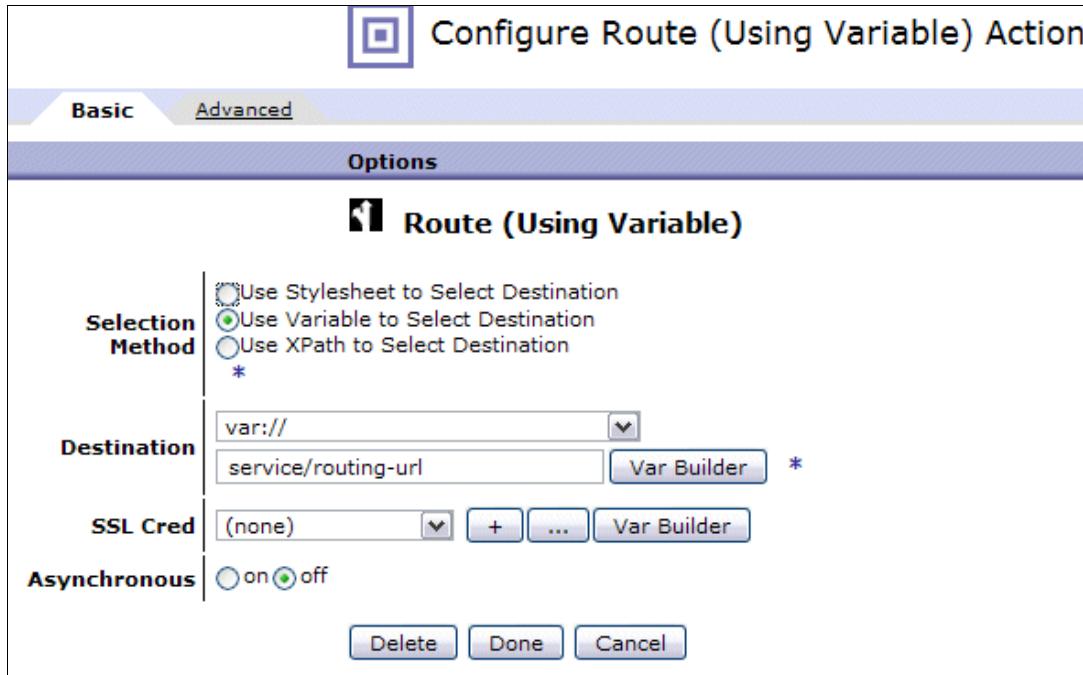


Figure 13-17 Setting destination with Route action

13.5 Testing the Binary AS2 over HTTP multi-step use case

We will perform a typical end-to-end test scenario, outlining the message flow from the Hub back-end application to the Partner back-end application (Figure 13-18 on page 346).

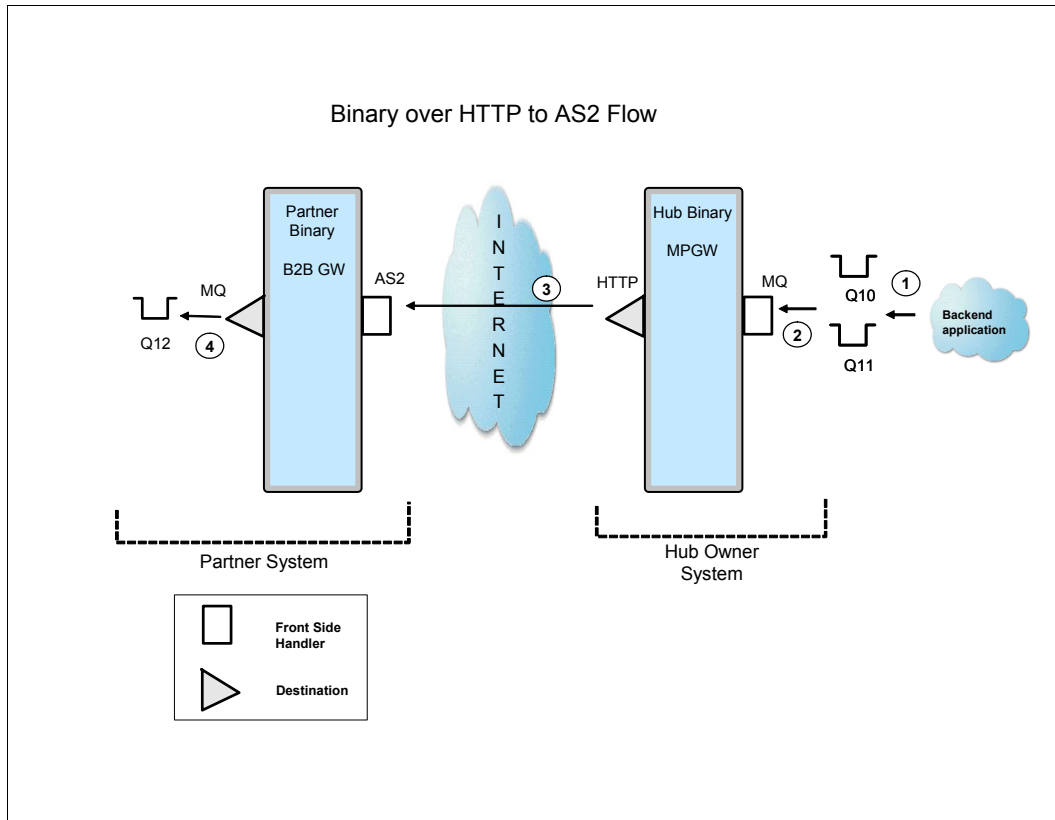


Figure 13-18 AS2 over HTTP Binary message flow

The following numbered steps refer to the numbers in Figure 13-18:

1. Binary messages can be placed on multiple queues that mimic multiple back-end applications sending binary messages. The message is placed on a queue using the WebSphere MQ (WMQ) utility, RfhUtil (refer to Figure 13-19 on page 347).

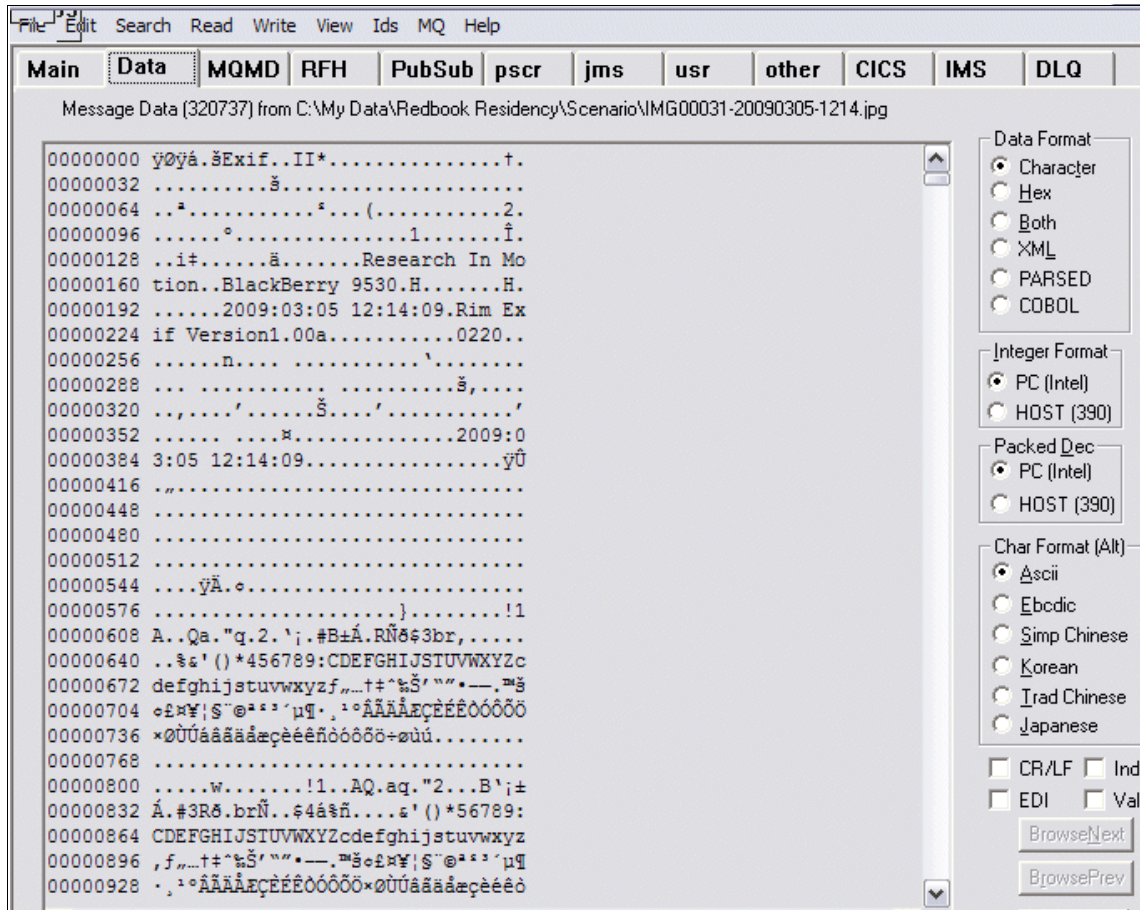



Figure 13-19 RfhUtil binary message

- Messages are received by the MQ FSH from, potentially, multiple queues. The HubBinary Processing Policy is invoked, and the message information is processed by the processing rule. The Transform action of the HTTP_Req_Rule_3 invokes the b2b-test-routing-mq.xsl stylesheet. This stylesheet looks up the destination based on the queue from which the message was received. The message is also wrapped in an AS2 header and routed over HTTP to the proper B2BGW (refer to Figure 13-20 on page 348).

| DATAPOWER | | Refresh | Flush | Disable Probe | Export Capture | View Log | Send Message |
|-----------|--------|---------|---|---------------|-----------------------------------|----------|--------------|
| view | trans# | type | inbound-url | | outbound-url | | |
| | 188533 | request | ftp://127.0.0.1:5117/%2F/00280001 | | dpmq://XB60 /?RequestQueue=Q13 | | |
| | 493473 | request | dpmq://XB60 /HubBinary_mq_fsh?RequestQueue=Q11 | | http://127.0.0.1:5114/ | | |

Figure 13-20 HubBinary Debug Probe showing routed AS2 over HTTP message

3. The message is sent over HTTP to the PartnerBinaryB2BGW. The message is received by the HubBinary_ftp_fsh FSH object on the HubBinary MPGW.
4. The message is sent to the destination that is defined in the PB_Int partner profile (refer to Figure 13-21 on page 349).

 **B2B Viewer** [Help](#)

[Modify Query](#) | [Refresh](#) |

| | Transaction Set ID | Transaction ID | Gateway Name | SenderName (ID) / Receiver (ID) | Inbound URL / Outbound URL |
|--------------------------|---------------------|----------------|--------------------|--|--|
| <input type="checkbox"/> | 464 | 315942 | PartnerBinaryB2BGW | Sender: hubbin (hubbin) Receiver: partnerbin (partnerbin) | as2://127.0.0.1:5114/ dpmq://XB60/?RequestQueue=Q12 |
| <input type="checkbox"/> | 463 | 236082 | PartnerBinaryB2BGW | Sender: partnerbin (partnerbin) Receiver: hubbin (hubbin) | dpmq://XB60 /PartnerBinary_mq?RequestQueue=Q14 ftp://userid:pwd7@127.0.0.1:5117 |

Figure 13-21 AS2 over HTTP binary message flow

13.6 Presenting the binary FTP multi-step use case

This section outlines a specific use case for sending binary messages over FTP to a Multi-Protocol Gateway service for processing (refer to Figure 13-22 on page 350). This use case reuses much of the previous use case's processing objects. The new objects that we defined are:

- ▶ FTP processing rules
- ▶ FTP Server Front Side Handlers
- ▶ B2BGW Document Routing Preprocessor

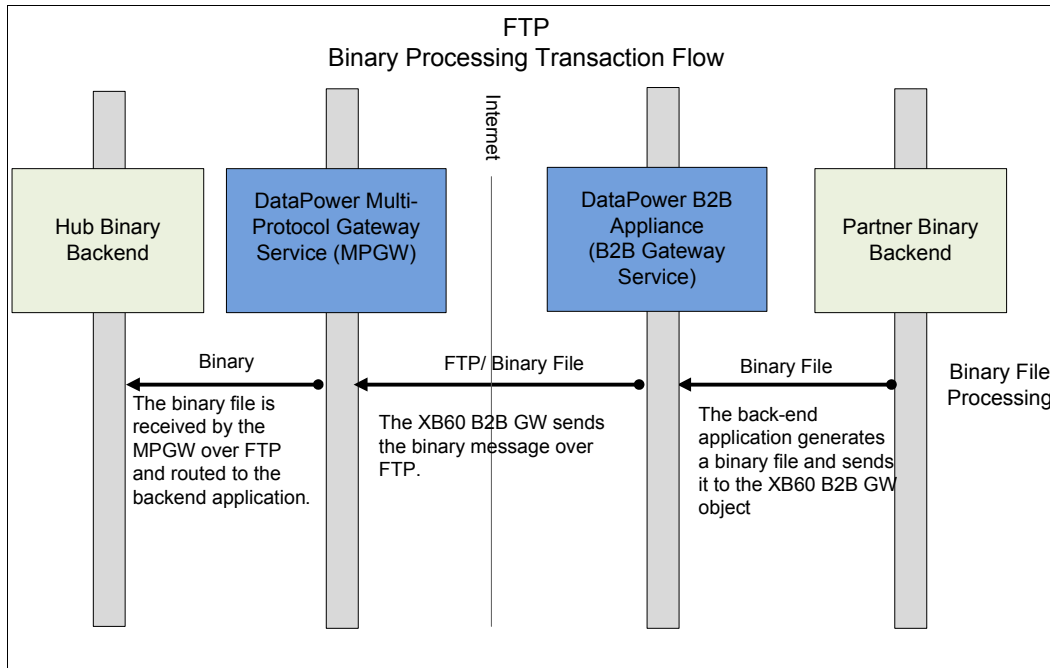


Figure 13-22 FTP To AS2 binary processing transaction flow

13.7 Binary FTP multi-step use case solution

The following steps outline the processing flow (refer to Figure 13-23 on page 351):

1. The binary message is sent by the PartnerBinary back-end application to the MQ queue.
2. The binary message is fetched from the queue by PartnerBinaryB2BGW by MQ FSH.
3. The B2BGW Document Routing Preprocessor parses the document to select the appropriate trading partners for the message.
4. After the appropriate external trading partner is determined, the destination is read from the partner's external profile.
5. The binary message is sent via FTP to the HubBinary MPGWS trading partner service.
6. The message is received by HubBinary FTP Server Front Side Handler and processed by the appropriate multi-step processing rule.

7. The routing information is obtained by the processing rule, and the message is sent to the HubBinary back-end application via MQ queue.

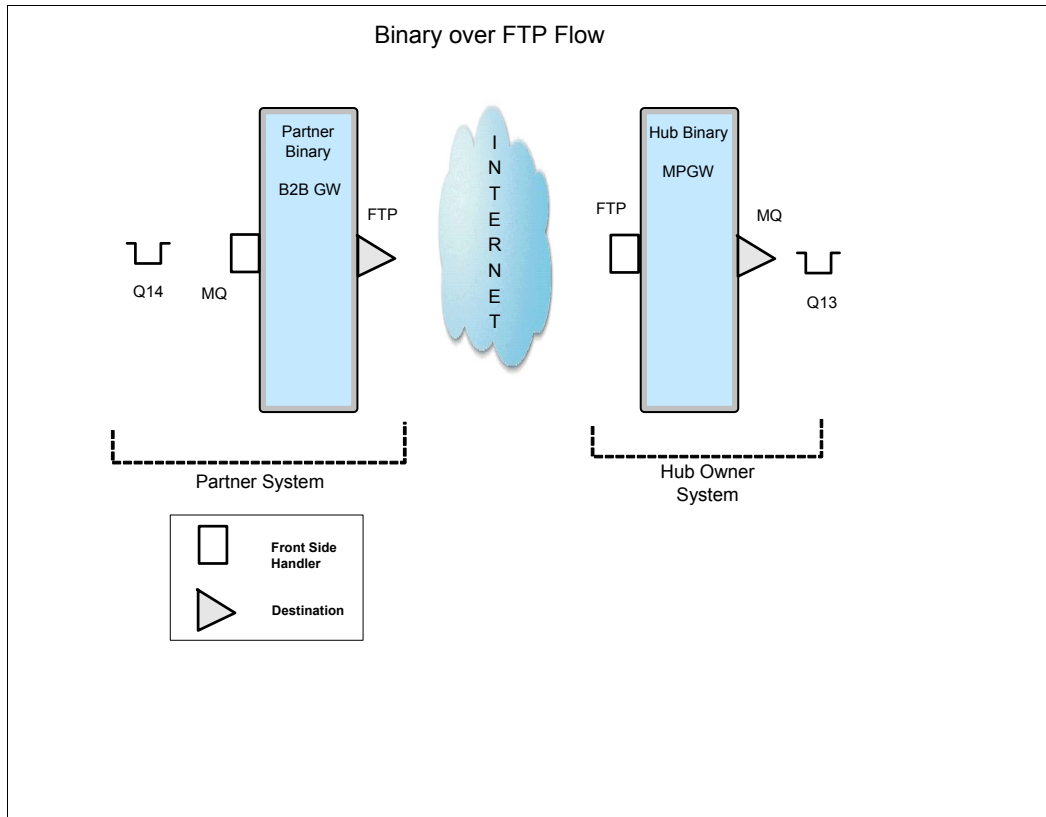


Figure 13-23 Binary over HTTP to AS2 architecture

13.7.1 Use case outline

Here is a summary of the steps to configure the appropriate processing objects to implement the scenario:

- ▶ Step 1: Modifying the existing B2B Gateway
- ▶ Step 2: Modifying the existing Multi-Protocol Gateway

13.7.2 Use case implementation

We will reuse the objects that were defined in the 13.4, “Binary AS2 over HTTP multi-step use case solution” on page 326. In this section, we will outline the additional objects that are necessary to process our FTP messages.

Step 1: Modifying the existing B2B Gateway

First, we must modify the existing B2B Gateway.

Creating PartnerBinary_mq MQ Front Side Handler

This MQ Front Side Handler object listens for a message that come from the MQ queue called Q14 (Figure 13-24). This queue represents the partner application that originates the binary messages that route through the PartnerBinB2BGW object via FTP to the HubBin_MPGW service.

Configure MQ Front Side Handler

Main

MQ Front Side Handler: PartnerBinary_mq [up]

Apply Cancel Undo [Export](#)

Admin State enabled disabled

Comments

Queue Manager + ... *

Get Queue *

Put Queue

CCSI

Get Message Options

Figure 13-24 Configuring the MQ Front Side Handler

Setting the Advanced tab Document Routing Preprocessor

The Advanced tab (refer to Figure 13-25 on page 353) outlines the message preprocessing functions for routing the binary message to the HubBinary partner over FTP (Figure 13-22 on page 350) in the Document Routing Preprocessor field.



Figure 13-25 PartnerBinaryB2BGW object Advanced tab

The Document Routing Preprocessor uses the `b2b-partner-routing.xsl` stylesheet to determine the routing information for the message. This stylesheet determines the “To” partner to route the message based on the MQ queue from which the message was received.

The code snippet in Example 13-2 determines if the message that we are receiving is from MQ. If so, we continue; if not, there is an error. It then parses the current URL string to determine the queue from which the message was sent. For our example, we use queue Q14.

We then set the doc-type (binary), the To partner (hubbin), and the From partner (partnerbin), which allows the proper external partner profile to be called based on the To field. Recall that our only external profile, `HB_Ext`, references “hubbin” as the Business ID.

Example 13-2 The `b2b-partner-routing.xsl` stylesheet

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
```

```
DataPower XB60 B2B Document Routing Preprocessor Stylesheet
```

```
Licensed Materials - Property of IBM
IBM WebSphere DataPower Appliances
Copyright IBM Corporation 2008. All Rights Reserved.
<?xml version="1.0" encoding="UTF-8"?>
```

<!--

DataPower XB60 B2B Document Routing Preprocessor Stylesheet

Licensed Materials - Property of IBM

IBM WebSphere DataPower Appliances

Copyright IBM Corporation 2008. All Rights Reserved.

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

-->

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  xmlns:regex="http://exslt.org/regular-expressions"
  extension-element-prefixes="dp">
  <xsl:template match="/">
    <!-- Do queue-based routing if MQ, default routing otherwise. -->
    <xsl:variable name="protocol"
      select="dp:variable('var://service/protocol')" />
    <xsl:choose>
      <xsl:when test="$protocol='dpmq'">
        <!-- Pull the request queue out of the inbound URL: -->
        <xsl:variable name="url"
          select="dp:variable('var://service/URL-in')" />
        <xsl:variable name="queue"
          select="substring-after($url,'RequestQueue=')" />
        <xsl:choose>
          <xsl:when test="$queue='Q14'">
            <dp:set-variable
              name="'var://service/b2b-doc-type'" value="'binary'" />
            <dp:set-variable
              name="'var://service/b2b-partner-from'" value="'partnerbin'" />
            <dp:set-variable
              name="'var://service/b2b-partner-to'" value="'hubbin'" />
          </xsl:when>
          <xsl:otherwise>
            <xsl:message dp:priority="error">
              <xsl:text>
                Could not find route for Queue
              </xsl:text>
              <xsl:value-of select="$queue" />
            </xsl:message>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:choose>
    </xsl:template>
  </xsl:stylesheet>
```

```
</xsl:when>
</xsl:choose>
</xsl:template>
</xsl:stylesheet>
```

Step 2: Modifying the existing Multi-Protocol Gateway

Next, we modify the existing Multi-Protocol Gateway.

Creating HubBinary FTP Server Front Side Handler

The configuration that is shown in Figure 13-26 shows a Port Number of 5117. This port number is the port on which the FTP Server FSH is defined to listen for FTP messages.

Configure FTP Server Front Side Handler

Main Virtual Directories

FTP Server Front Side Handler: HubBinary_ftp_fsh [up]

Apply Cancel Undo Export | V

Admin State enabled disabled

Comments

Local IP Address Select Alias *

Port Number *

Filesystem Type ▼

Default Directory

Maximum Filename Length

Access Control List + ...

Require TLS on off

SSL Proxy + ...

Figure 13-26 FTP Server Front Side Handler configuration (page one)

The configuration in Figure 13-27 is mostly default values with the exception of the “Allow Unique File Name (STOU)” parameter. This parameter is set to **on** to allow the FTP server to generate a unique file name for each transferred file.

| | | | |
|--|-------------------------------------|--------------------------------------|---------|
| Password AAA Policy | (none) ▼ | + | ... |
| Certificate AAA Policy | (none) ▼ | + | ... |
| Allow CCC Command | <input checked="" type="radio"/> on | <input type="radio"/> off | |
| Passive (PASV) Command | Allow Passive Mode | ▼ | |
| Limit Port Range for Passive Connections | <input type="radio"/> on | <input checked="" type="radio"/> off | |
| Passive Data Connection Idle Timeout | 60 | | seconds |
| File Transfer Data Encryption | Allow Data Encryption | ▼ | |
| Allow Compression | <input checked="" type="radio"/> on | <input type="radio"/> off | |
| Allow Unique File Name (STOU) | <input checked="" type="radio"/> on | <input type="radio"/> off | |
| Unique File Name Prefix | | | |
| Idle Timeout | 0 | | seconds |
| Response Type | No Response | ▼ | |
| Temporary Storage Size | 32 | | |

Figure 13-27 FTP Server Front Side Handler configuration (page two)

MPGW FTP processing policy

Next, we define the MPGW FTP processing policy (refer to Figure 13-28 on page 357).

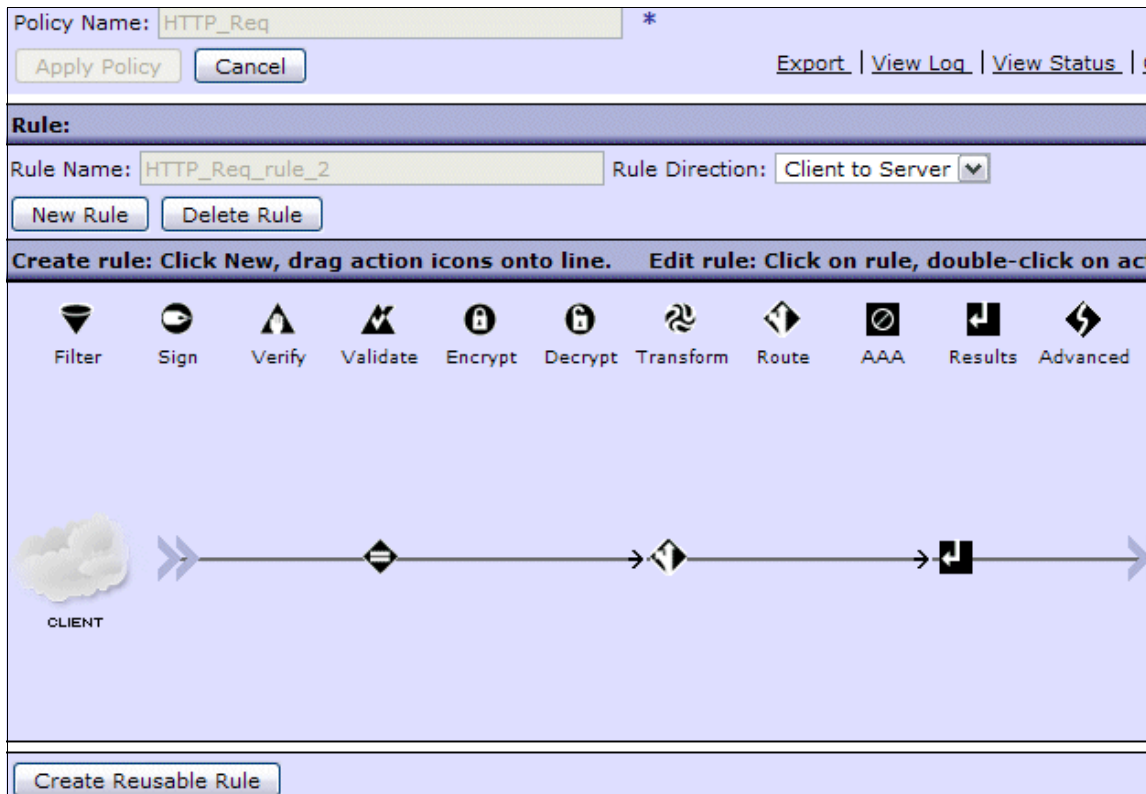


Figure 13-28 Defining the new MPGW FTP processing policy

HubBinary MPGW processing policy FTP routing code snippet

This code snippet (Example 13-3) hard codes a value for the var://service/routing-url to a predefined MQ queue.

Example 13-3 The b2b-test-routing-hub.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
```

DataPower XB60 B2B Document Routing Preprocessor Stylesheet

Licensed Materials - Property of IBM

IBM WebSphere DataPower Appliances

Copyright IBM Corporation 2008. All Rights Reserved.

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

```
-->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  xmlns:regex="http://exslt.org/regular-expressions"
  extension-element-prefixes="dp">
  <xsl:template match="/">
    <dp:set-variable name="'var://service/routing-url'"
value="'dpmq://XB60/?RequestQueue=Q13'" />
  </xsl:template>
</xsl:stylesheet>
```

13.8 Testing the binary FTP multi-step use case

Now, we perform a typical end-to-end test scenario outlining the message flow from the Partner back-end application to the Hub back-end application. Refer to Figure 13-29 on page 359.

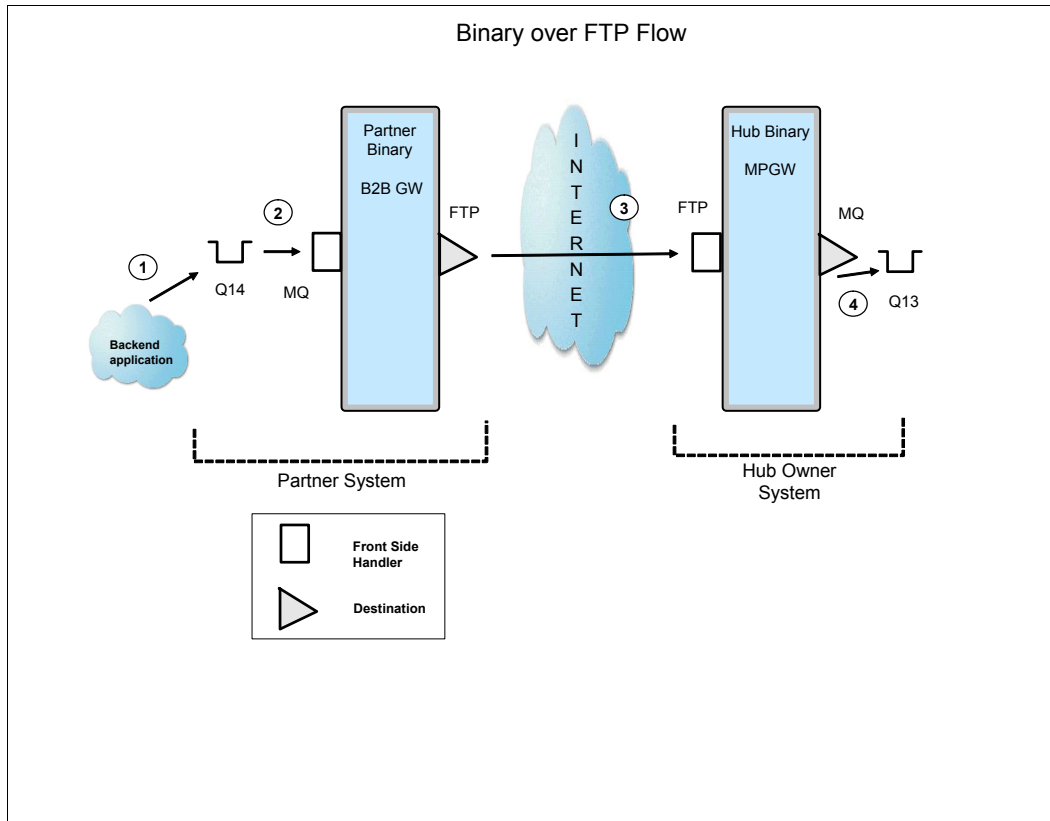


Figure 13-29 FTP and Binary message flow

The numbered steps correspond to the numbers in Figure 13-29:

1. The binary message is placed on queue Q14 using the WMQ utility, RfhUtil (refer to Figure 13-30 on page 360).

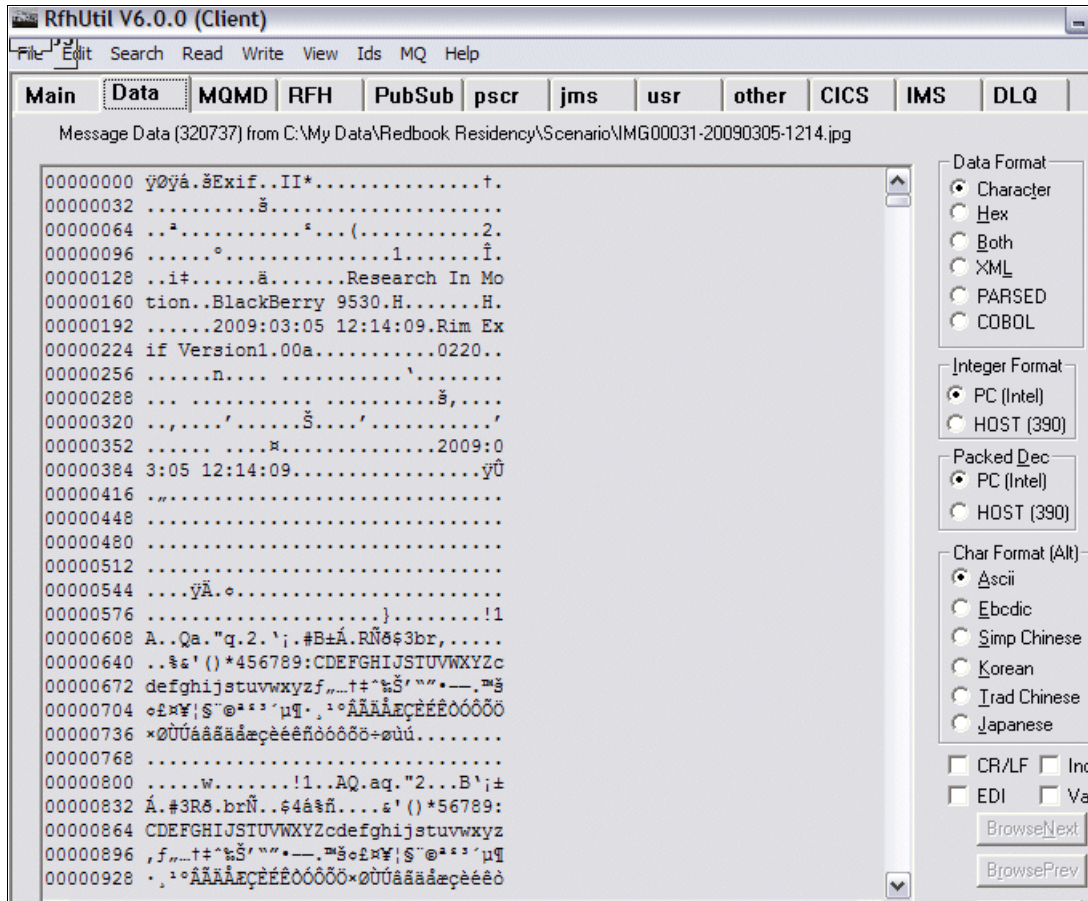



Figure 13-30 RfhUtil binary message

2. The message is received by the MQ FSH. The stylesheet b2b-partner-routing.xsl is invoked and the Sender, Receiver, and document type variables are set for routing to the HubBinary MPGW service.
3. The message is sent over FTP to the HubBinary MPGW using the destination HubBin_FTP (defined in the HB_Ext partner profile). The message is received by the HubBinary_ftp_fsh FSH object on the HubBinary MPGW (refer to Figure 13-31 on page 361).

 **B2B Viewer** [Help](#)

[Modify Query](#) | [Refresh](#)

| | Transaction Set ID | Transaction ID | Gateway Name | SenderName (ID) / Receiver (ID) | Inbound URL / Outbound URL |
|--------------------------|---------------------|----------------|--------------------|--|--|
| <input type="checkbox"/> | 464 | 315942 | PartnerBinaryB2BGW | Sender: hubbin (hubbin) Receiver: partnerbin (partnerbin) | as2://127.0.0.1:5114/ dpmq://XB60/?RequestQueue=Q12 |
| <input type="checkbox"/> | 463 | 236082 | PartnerBinaryB2BGW | Sender: partnerbin (partnerbin) Receiver: hubbin (hubbin) | dpmq://XB60 /PartnerBinary_mq?RequestQueue=Q14 ftp://userid:pwd7@127.0.0.1:5117 |

Figure 13-31 B2B Transaction Viewer showing message transactions from MQ to FTP

- The message is routed from the HubBinary MPGW to the proper MQ queue, in this case, Q13. Refer to Figure 13-32.



| view | trans# | type | inbound-url | outbound-url |
|---|--------|---------|--|--------------------------------|
|  | 188533 | request | ftp://127.0.0.1:5117/%2F00280001 | dpmq://XB60 /?RequestQueue=Q13 |
|  | 493473 | request | dpmq://XB60 /HubBinary_mq_fsh?RequestQueue=Q11 | http://127.0.0.1:5114/ |

Figure 13-32 HubBinary Debug Probe showing routed FTP message



Handling SOAP Messages with Attachments in a B2B environment

In this scenario, we will show you how to handle SOAP Messages with Attachments to trade in a business-to-business (B2B) fashion inside the WebSphere DataPower B2B Appliances XB60.

14.1 Business value

There is a strong trend for companies to integrate existing systems to implement IT support for business processes that cover the entire business cycle. Interactions in the B2B world are typically presented using a variety of schemes that can be extremely rigid point-to-point architectures, as in the case of electronic data interchange (EDI).

Many companies have already made some of their IT systems available to all of their divisions and departments, or even their customers or partners on the Web. However, techniques for collaboration vary from one case to another and are thus proprietary solutions; systems often collaborate without any visionary architecture.

Thus, there is an increasing demand for technologies that support the connecting or sharing of resources and trading data in an extremely flexible and standardized manner.

Furthermore, there is a need to further structure large applications into building blocks in order to use well-defined components within separate business processes. A shift toward a service-oriented approach not only standardizes interaction, but it also allows for more flexibility in the process.

Furthermore, in the B2B world, a common pattern for this kind of business need is to use SOAP Messages with Attachments to exchange the trading information using a Web service client, developed with the Web Services Description Language (WSDL) that we provide to our trading partner, so that the trading partner knows how to handle their requests to us.

In this scenario, we present one possible solution to provide this new facade into the WebSphere DataPower B2B Appliance XB60.

14.2 Prerequisites

You need the following software and skills to implement this scenario.

14.2.1 Software prerequisites

In order to run this scenario, you must have the following components installed and configured:

- ▶ WebSphere DataPower B2B Appliances XB60
- ▶ WebSphere Transformation Extender V8.2 Design Studio

- ▶ WebSphere MQ V6
- ▶ cURL or any other HTTP utilities that are available on the market

14.2.2 Skill prerequisites

This scenario is intended for the intermediate user, who must be familiar with the with following concepts:

- ▶ WebSphere DataPower B2B Appliance XB60 main concepts (you must have completed, at least, the self-trade case in Appendix A, “Additional material” on page 389)
- ▶ Basic Extensible Stylesheet Language Transformation (XSLT) techniques
- ▶ Basic knowledge of SOAP with Attachment specifications

14.2.3 Infrastructure prerequisites

In order to successfully test this scenario, the Partner Binary B2B Gateway Service configuration, refer to Chapter 13, “Trading binary documents using a Multi-Protocol Gateway service” on page 323.

14.3 Presenting the scenario

in this scenario, we implement a one-way flow that receives incoming SOAP Messages with Attachments (SwA) requests from a partner; these requests are consumed by the Hub B2B Gateway Service that is also configured in this book.

To achieve this scenario, the partner sends the SOAP Messages with Attachments message over HTTP to our B2B Gateway Service that is configured on the XB60. The gateway service is configured to fetch the attachment from the payload. The attachment is wrapped in AS2 and routed to the appropriate B2B Gateway Service that will handle the trading processing.

There will be no Message Disposition Notifications (MDNs) in this process, because the partner interaction is done using HTTP. Although in our Hub internally, we use AS2.

Figure 14-1 shows the flow from a high-level perspective.

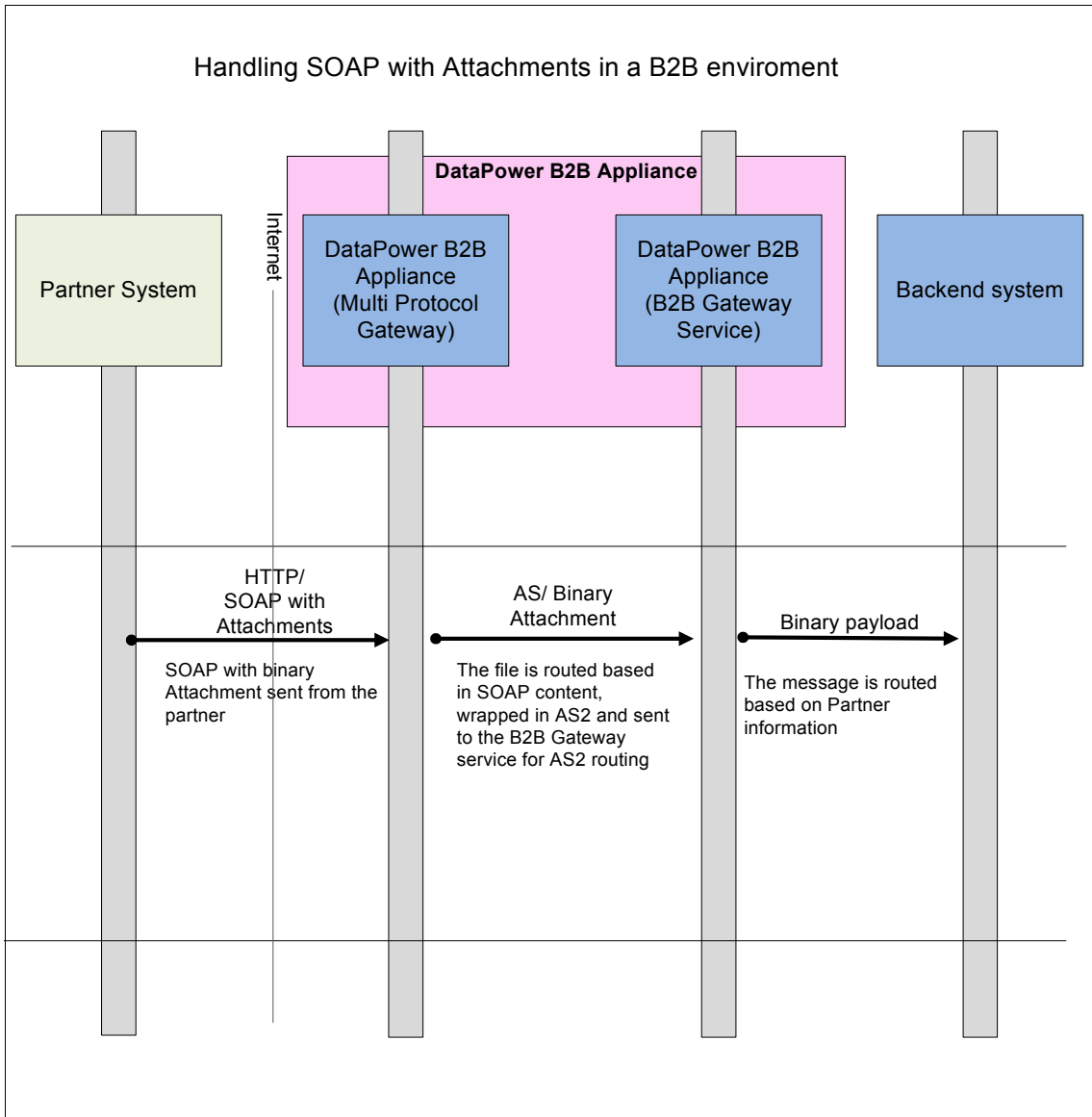


Figure 14-1 Handling SOAP Messages with Attachments in a B2B environment diagram

14.4 Scenario solution

The key aspect of this scenario implementation is configuring a service to handle SOAP Messages with Attachments.

The XB60 will receive these messages using SOAP standard over a protocol, such as HTTP, from a trading partner (as in our case).

SOAP over HTTP (or Java Message Service (JMS)) does not have the same routing headers as AS2 or AS3, where specific information about the sender and the receiver is set. Those headers will need to be set in order for our B2B Gateway Service to consider them a valid trading scenario.

Information about how to route messages to the specific B2B Gateway is required. By using the B2BGW service, we take advantage of all the profiling and viewing features that the XB60 provides.

Therefore, there is a set of information that we need to retrieve from several sources:

- ▶ The sender ID, which is retrieved from an incoming request, which, in our example, is the SOAP payload
- ▶ Routing information that is stored on the device in an XML properties file

After the AS2 headers have been added, we extract the binary attachment and route it to the appropriate B2B Gateway Service that we use to trade with that specific partner. We will configure a processing policy on the Multi-Protocol Gateway Service to perform these tasks.

Important: Notice that this scenario can also be implemented with a WS-Proxy service instead of a Multi-Protocol Gateway. The WS-Proxy service is used most commonly in use cases where a Web Services Description Language (WSDL) is involved. To keep the scenario simple, we decided not to use a WSDL in this example.

14.4.1 Scenario outline

Figure 14-2 on page 368 shows the architecture of the scenario.

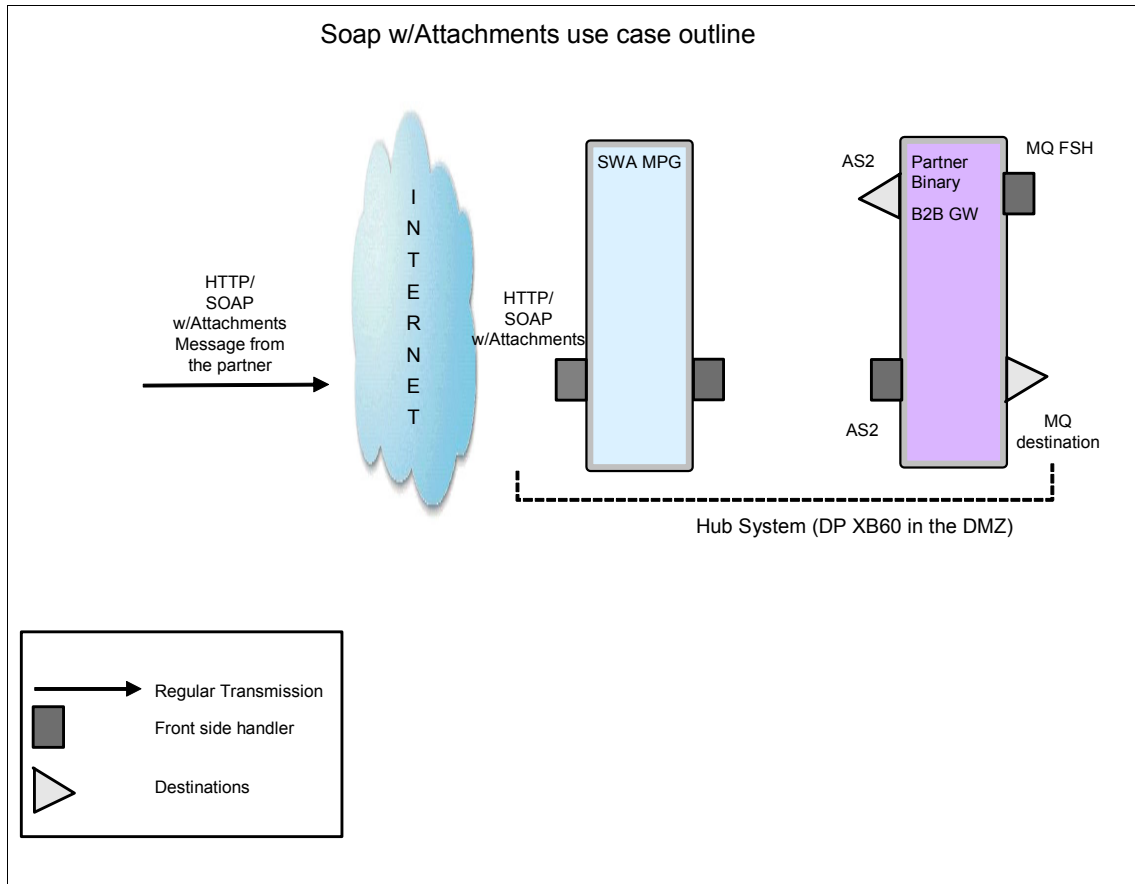


Figure 14-2 SOAP Messages with Attachments use case outline

The box in light blue represents the Multi-Protocol Gateway (SWA MPG); this service is the front end for the incoming SOAP Messages with Attachments requests. After wrapping the attachment in AS2 and setting the appropriate routing information, the message will be sent to the appropriate Binary Partner B2BGW service (purple box), where it is treated as a “regular AS2” transaction. It is important to notice that in this particular integration situation, MDNs will not be needed, because the original request from the partner was not generated with AS2, but with SOAP.

In order to implement this scenario, we perform these tasks:

- ▶ Step 1: Creating the SWA_MPG Multi-Protocol Gateway
- ▶ Step 2: Creating the SWA_Policy processing policy

14.4.2 Scenario implementation

Next, we describe the implementation.

Step 1: Creating the SWA_MPG Multi-Protocol Gateway

The MPGW service will be responsible for receiving the SOAP Messages with Attachments payload, extracting the binary attachment, and sending it to the B2B Gateway Service wrapped in AS2.

The AS2 headers will be created based on the information coming in the SOAP payload (not in the attachment), and on certain internal data, as our “host name” or our own partner ID.

Figure 14-3 on page 370 is the General configuration tab window of the B2BGW Service: we have selected **dynamic backend** for the back-end type, **SWA_http_FSH** as the Front Side Protocol handler, and **SWA_Policy** as the Multi-Protocol Gateway processing policy (which will be explained in Step 2).

□ **Configure Multi-Protocol Gateway**

← **General** Advanced Stylesheet Params Headers Monitors WS-Addressing WS-ReliableMessaging →

Apply Cancel Delete

[Export](#) | [View Log](#) | [View Status](#) | [Show Probe](#) | [Validate Conformance](#) | [Help](#)

Multi-Protocol Gateway status: [up]

General Configuration

| | |
|---|--|
| <p>Multi-Protocol Gateway Name <input type="text" value="SWA_MPG"/> *</p> <p>Summary <input type="text"/></p> <p>Type</p> <p><input checked="" type="radio"/> dynamic-backend</p> <p><input type="radio"/> static-backend</p> <p>*</p> | <p>XML Manager <input type="text" value="default"/> ▼ + ... *</p> <p>Multi-Protocol Gateway Policy <input type="text" value="SWA_Policy"/> ▼ + ... *</p> <p>URL Rewrite Policy <input type="text" value="(none)"/> ▼ + ...</p> |
|---|--|

| | | | | | | | | | |
|---|--|---|---|---|---|---|---|---|---|
| <p>Back side settings</p> <p>With a dynamic proxy back end type, the back end server address and port are determined by a stylesheet in a policy action.</p> | <p>Front side settings</p> <p>Front Side Protocol</p> <table style="border: 1px solid black; width: 100%;"> <tr> <td style="border: 1px solid black; padding: 2px;">SWA_http_FSH (HTTP Front Side Handler)</td> <td style="border: 1px solid black; padding: 2px;">↑</td> <td style="border: 1px solid black; padding: 2px;">↓</td> <td style="border: 1px solid black; padding: 2px;">✕</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;"> <input type="text" value="SWA_http_FSH (HTTP Front Side Handler)"/> ▼ </td> <td style="border: 1px solid black; padding: 2px;">Add</td> <td style="border: 1px solid black; padding: 2px;">+</td> <td style="border: 1px solid black; padding: 2px;">...</td> </tr> </table> <p>*</p> | SWA_http_FSH (HTTP Front Side Handler) | ↑ | ↓ | ✕ | <input type="text" value="SWA_http_FSH (HTTP Front Side Handler)"/> ▼ | Add | + | ... |
| SWA_http_FSH (HTTP Front Side Handler) | ↑ | ↓ | ✕ | | | | | | |
| <input type="text" value="SWA_http_FSH (HTTP Front Side Handler)"/> ▼ | Add | + | ... | | | | | | |

Figure 14-3 SWA_MPG General configuration tab details

Figure 14-4 on page 371 shows the HTTP Front Side Handler configuration.

Configure HTTP Front Side Handler

Main

HTTP Front Side Handler: SWA_http_FSH [up]

Apply Cancel Undo Export View Log

Admin State enabled disabled

Comments

Local IP Address *

Port Number *

HTTP Version to Client

Allowed Methods and Versions

- HTTP 1.0
- HTTP 1.1
- POST method
- GET method
- PUT method
- HEAD method
- OPTIONS
- TRACE method
- DELETE method
- URL with Query Strings
- URL with Fragment Identifiers
- URL with ..
- URL with cmd.exe

Figure 14-4 SWA_http_FSH configuration details

Notice that we have set the Response type as **Pass Thru**; there will be no response from the B2B Gateway Service (Figure 14-5 on page 372).

User Agent settings

| Match | Property |
|---|---|
| Note: To edit the User Agent, please access via the XML Manager above. | |
| SSL Client Crypto Profile (none) ▼ + ... | |
| Response Type | Request Type |
| <input type="radio"/> Non-XML | <input type="radio"/> Non-XML |
| <input checked="" type="radio"/> Pass-Thru | <input type="radio"/> Pass-Thru |
| <input type="radio"/> SOAP | <input checked="" type="radio"/> SOAP |
| <input type="radio"/> XML | <input type="radio"/> XML |
| Back attachment processing format | Front attachment processing format |
| <input checked="" type="radio"/> Dynamic | <input checked="" type="radio"/> Dynamic |
| <input type="radio"/> MIME | <input type="radio"/> MIME |
| <input type="radio"/> DIME | <input type="radio"/> DIME |
| <input type="radio"/> Detect | <input type="radio"/> Detect |
| Back Side Timeout <input type="text" value="120"/> * | Front Side Timeout <input type="text" value="120"/> * |
| Stream Output to Back | Stream Output to Front |
| <input checked="" type="radio"/> Buffer Messages | <input checked="" type="radio"/> Buffer Messages |
| <input type="radio"/> Stream Messages | <input type="radio"/> Stream Messages |
| HTTP Version to Server | |
| <input type="radio"/> HTTP 1.0 | |
| <input checked="" type="radio"/> HTTP 1.1 | |
| Propagate URI | |
| <input checked="" type="radio"/> on <input type="radio"/> off | |
| Compression | |
| <input type="radio"/> on <input checked="" type="radio"/> off | |

Figure 14-5 SWA MPG General configuration tab details (continuation)

Also, it is important to configure the MPGW to *allow* the attachments coming from the request; therefore, access the Multi-Protocol Gateway service through the Objects tab, and go to the Proxy Settings tab (Figure 14-6 on page 373). On that page, select **Attach** for the Request Attachment Processing Mode and configure the other fields as in Figure 14-6 on page 373.

Configure Multi-Protocol Gateway

Main Proxy Settings HTTP Options Parser Limits Monitors

Multi-Protocol Gateway: SWA_MPG [up]

Apply Cancel Delete Undo Export | View Log | View Status |

URL Rewrite Policy (none) + ...

SSL Proxy (none) + ...

Crypto Credentials (none) + ...

Default parameter namespace http://www.datapower.com/parar

Query parameter namespace http://www.datapower.com/parar

Request attachment processing mode Allow

Front attachment processing format Dynamic

Back attachment processing format Dynamic

Front Side MIME Header Processing on off

Back Side MIME Header Processing on off

Stream Output to Back Buffer Messages

Stream Output to Front Buffer Messages

Client Traffic Type SOAP

Server Traffic Type Pass-Thru

SOAP Schema URL store:///schemas/soap-envelope.

Figure 14-6 SWA_MPG Proxy Settings configuration tab details

Step 2: Creating the SWA_Policy processing policy

The processing policy included in the SWA MPG will handle the actions to fetch the attachment and route it with the appropriate AS2 headers to the back end.

Before configuring the policy, it is important to examine our sample SOAP with Attachment payload in Example 14-1 on page 374.

Important: For more information about SOAP Messages with Attachments, visit:

<http://www.w3.org/TR/SOAP-attachments>

In Example 14-1, it is important to notice the various Multipurpose Internet Mail Extensions (MIME) headers that are defined in the MIME standard as Content-type, Content-Transfer-Encoding, and Content-ID.

Specifically, the Content-ID MIME header will be used to fetch the attachment (identified as “bin”) in the Fetch action.

Example 14-1 Incoming SOAP with Attachment sample

```
--c79ab6a8-59ce-4b21-abab-2dcdd51e46d9 (MIME Boundary)
Content-Type: text/xml; charset=utf-8
Content-Transfer-Encoding: 8bit
Content-ID: <main>

<?xml version="1.0" encoding="UTF-8"?> (SOAP Payload)
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:dp="http://www.datapower.com/schemas/management"
xmlns:echo="http://example.com/echo/">
<soapenv:Body>
<echo:echo>
<senderID>binarypartner</senderID>
</echo:echo>
</soapenv:Body>
</soapenv:Envelope>
--c79ab6a8-59ce-4b21-abab-2dcdd51e46d9 (MIME Boundary)
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <bin>

¶ _ _ _ ?_A*æ Z|øP_ (Binary attachment)
--c79ab6a8-59ce-4b21-abab-2dcdd51e46d9-- (MIME Boundary)
```

Figure 14-7 on page 375 shows the processing policy. We only configure a request Rule Direction (select **Client to Server**), and no response is expected from the B2B Gateway Service (remember that this scenario is a one-way flow).

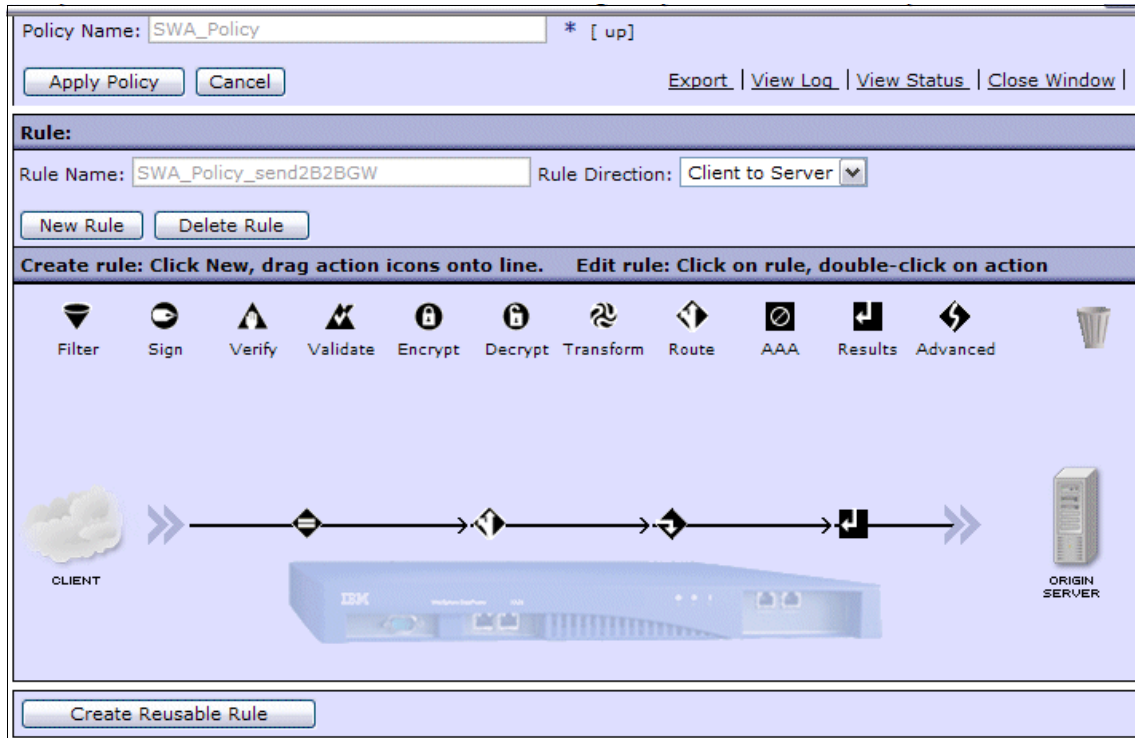


Figure 14-7 SWA_Policy processing policy configuration details

In the next steps, we describe the various processing actions.

Match action

We do not go into any detail about this action. It is a matching rule that handles all the incoming URLs with the “.” regular expression.

Route action

The Route action (Figure 14-8 on page 376) will call a stylesheet (**mpg-test-routing.xsl**). This stylesheet will route the attachment to the B2B Gateway Service and add the mandatory AS2 headers.

□ **Configure Route (Using Stylesheet or XPath Expression) Action**

Basic
Advanced

Input

Input | * *

Options

Route (Using Stylesheet or XPath Expression)

Selection Method

- Use Stylesheet to Select Destination
- Use Variable to Select Destination
- Use XPath to Select Destination

*

Processing Control File | *

Asynchronous | on off

Output

Output | * *

Figure 14-8 Route action configuration details

Example 14-2 on page 377 is the stylesheet that we used in the Route action.


```

<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:dp="http://www.datapower.com/extensions"
xmlns:regex="http://exslt.org/regular-expressions" extension-element-prefixes=" dp">
  <xsl:template match="/">
    <!--Fetch the routing table: -->
    <xsl:variable name="routing" select="document('local://b2b-
outing.xml')"/>
    <!--We extract the Sender ID coming from the SOAP message-->
    <xsl:variable name="partner" select="//*[local-name()='senderID']"/>
    <xsl:choose>
      <xsl:when test="$routing/routing/partner[@name=$partner]">
        <xsl:message dp:priority=" debug">
          <xsl:text>We have a matching partner!</xsl:text>
        </xsl:message>
        <dp:set-request-header name="AS2-To" value="binaryhub"/>
        <dp:set-request-header name="AS2-From" value="$partner"/>
        <dp:set-request-header name="Host" value="trading host"/>
        <xsl:variable name="uuid" select="dp:generate-uuid()"/>
        <dp:set-request-header name="Message-ID" value="Suuid"/>
        <dp:set-variable name="var://service/routing-url"
value="$routing/routing/partner[@name=$partner]/@path"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:message dp:priority=" error">
          <xsl:text>Could not find route for incoming Partner! </xsl:text>
          <xsl:value-of select="$partner" />
        </xsl:message>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>

```

A local XML properties file (Example 14-3 `b2b-routing.xml`) is used to retrieve the routing information for our B2B Gateway Service.

The sender ID in the incoming SOAP message is retrieved by XPath and then compared with the sender IDs stored in the XML properties file (`b2b-routing.xml`). If it finds a match, it retrieves the routing path and sets all the AS2 headers: AS2 From, AS2 to, message ID (which is internally generated by DataPower), and Host.

With this information, the AS2 FSH from any B2B Gateway Service can successfully accept our binary payload.

Here is the sample of the XML file `b2b-routing.xml` we use to route based on the incoming sender ID. More partners can be added by updating this file.

Example 14-3 The `b2b-routing.xml` that is used to provide the routing information

```
<?xml version="1.0" encoding="UTF-8" ?>
<routing>
  <partner name="binarypartner" path="http://b2bgw_IP:b2bgw_Port"/>
  <partner name="otherpartner" path="http://othergateway:someport"/>
</routing>
```

Fetch action

The Fetch action is configured to extract the attachment coming in the SOAP Messages with Attachments payload. DataPower uses the MIME Content-ID header, which appears before the attachment. In Example 14-4, this Content-ID bin is highlighted.

Example 14-4 SOAP with Attachment Content-ID used to Fetch Attachment in bold

```
(SwA omitted)
--c79ab6a8-59ce-4b21-abab-2dcdd51e46d9
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <bin>
```

```
¶ _ __?_A*æ Z|øP_
--c79ab6a8-59ce-4b21-abab-2dcdd51e46d9--
```

In the Configure Fetch Action window (Figure 14-9), set the Output context to **attachment**. This context will only contain the stripped attachment without the SOAP payload.


The screenshot shows the 'Configure Fetch Action' dialog box. It has two tabs: 'Basic' and 'Advanced'. The 'Options' section is expanded, showing a 'Fetch' icon. Below it, the 'Source' field is set to 'cid:' with a dropdown arrow, followed by a text box containing 'bin' and a 'Var Builder *' button. The 'Asynchronous' section has radio buttons for 'on' and 'off', with 'off' selected. Under the 'Output' section, the 'Output' field is set to 'attachment' with a dropdown arrow. At the bottom are 'Delete', 'Done', and 'Cancel' buttons.

Figure 14-9 Fetch action configuration details

Important: Notice that `bin` in the Content-ID is only a value that we have used here as an example, but any value can be configured as long as it matches the value coming in the Multipurpose Internet Mail Extensions (MIME) Content-ID header.

Results action

In the Configure Results Action window (Figure 14-10 on page 380), the Input context is the **attachment** context that we generated in the Fetch action.

 **Configure Results Action**

Basic
Advanced

Input

Input | | ▼ *

Options

Results

Destination | ▼ | ▼ | | | |

Asynchronous | on | off

Number of Retries |

Retry Interval | | msec

Output

Output | | ▼

| |

Figure 14-10 Results action configuration details

14.5 Testing our solution

For our testing, we will use cURL as the power on system test (POST) utility (Example 14-5 on page 381), but you can use any POST tool that is available on the Internet.

HTTP Headers need to be included in the POST Request. You can add them to the cURL request with the **-H** parameter followed by the headers separated by the (;) semicolon.

For our case, we need Content-type, Type, and Boundary http headers.

Important: The boundary header content must be the same content from the MIMEBoundary that was shown in Example 14-1 on page 374.

Example 14-5 cURL script that we use to test the scenario with important headers (in Bold) to be set

```
curl -H "Content-Type: Multipart/Related;type=\"text/xml\";  
boundary=\"c79ab6a8-59ce-4b21-abab-2dcdd51e46d9\" --data-binary @swa.soap  
http://DatapowerIP:SWA_FSH_port/
```

Important: When testing this scenario, make sure that there are spaces between each part of the message. A <CRLF> is required for each of the specific parts.

In Figure 14-11 on page 382, you can see the steps that will occur when we trigger the scenario by posting a message with cURL to the SWA MPGWS.

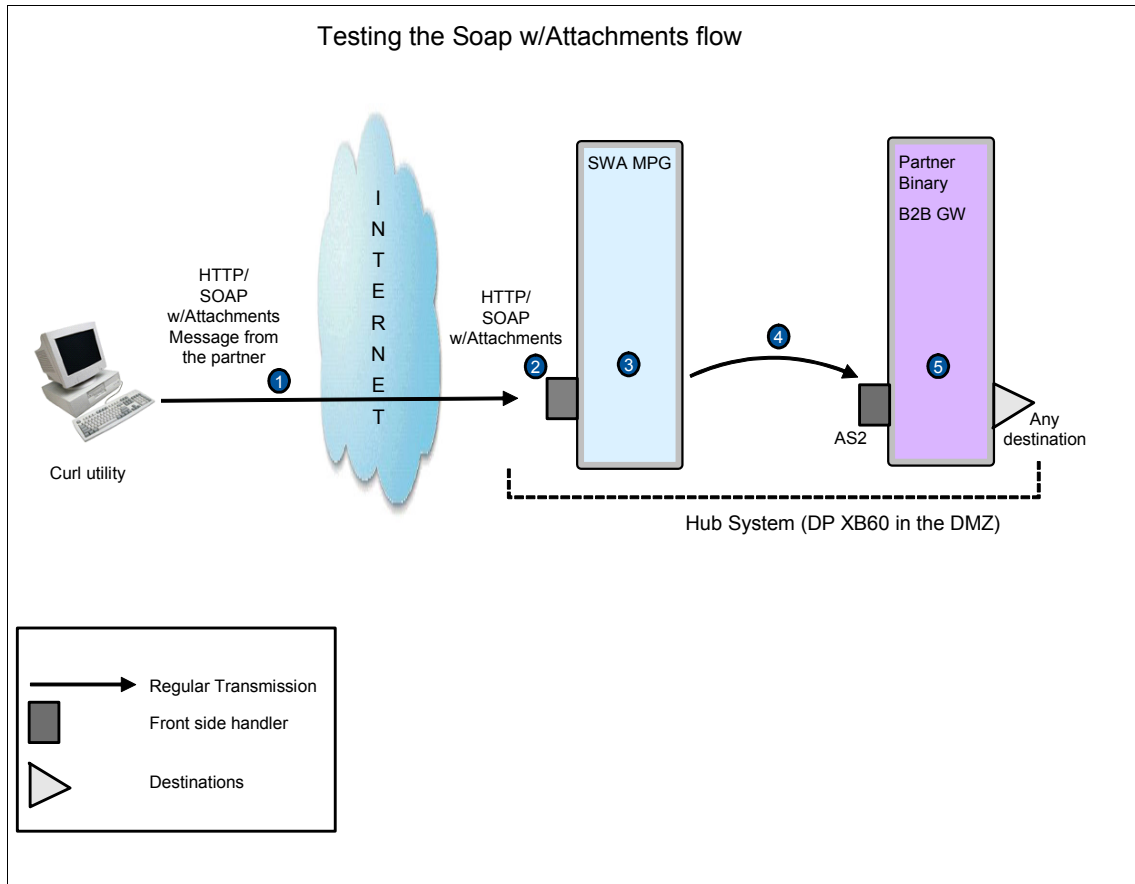


Figure 14-11 SOAP Messages with Attachments testing flow

Here is an explanation of the high-level steps in Figure 14-11. The numbers in Figure 14-11 correspond to these numbered steps:

1. The SOAP Messages with Attachments message is posted to the DataPower device using cURL, where SWA_http_FSH Front Side Handler is listening on port *NN*.
2. The SOAP Messages with Attachments Message is received by the http FSH, the SOAP content is verified as valid SOAP, and the message comes into the policy SWA_Policy.
3. Inside the SWA_Policy, the attachment is stripped and then wrapped with AS2 headers using the information coming in the SOAP payload. And then, the message is routed to the appropriate back end, using the mpg-routing-test.xml inside the Route action and the b2b-routing.xml file.

4. The new payload is sent to Partner_Binary B2BGW. This service receives the AS2 message using PartnerBin_as2_fsh front side handler.
5. The payload is processed by the B2B Gateway Service and routed to the back-end server. If you are using the Partner Binary B2B Gateway Service, its destination is an MQ queue.

14.5.1 Test results

To see the test results, you can look at the SWA_MPG probe (Figure 14-12) to check that all the headers have been generated successfully and that the attachment was stripped correctly.

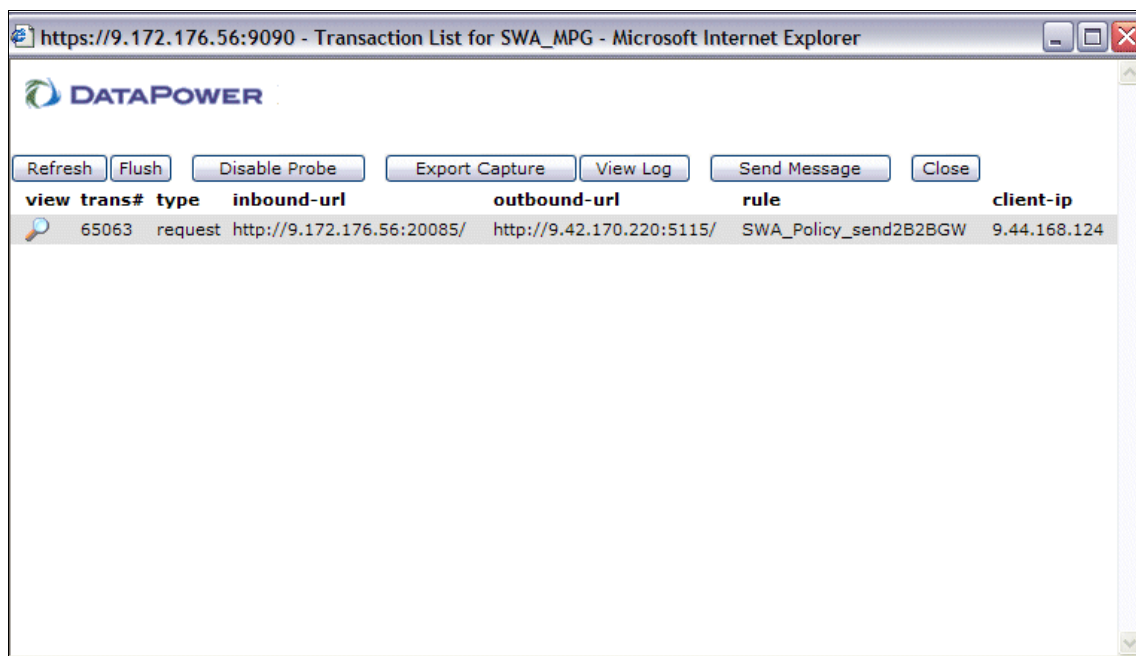


Figure 14-12 Probe of our test results

Figure 14-13 on page 384 shows the Input content that represents what comes into the device.

Previous Next

Input Context 'INPUT' of Step 1

[] - [] - [] - [] - [] - []

Step 1: Route (Using Stylesheet or XPath Expression) Action: Input=INPUT, Transform=local:///mpg-test-routing.xsl, Output=NULL, NamedInOutLocationType=default, Transactional=off, SOAPValidation=body, SQLSourceType=static, Asynchronous=off, ResultsMode=first-available, RetryCount=0, RetryInterval=1000, MultipleOutputs=off, IteratorType=XPATH, Timeout=0

Content Headers Attachments Local Variables Context Variables Global Variables Service

Content of context 'INPUT':

```

<soapenv:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:echo="http://example.com/echo/"
>
  <soapenv:Body>
    <echo:echo>
      <senderID>swapartner</senderID>
    </echo:echo>
  </soapenv:Body>
</soapenv:Envelope>

```

Show unformatted Send as message

Figure 14-13 Input context coming from our test results

Figure 14-14 on page 385 shows the headers that we attached during the process in the Output content view of the Probe, which represent what leaves the device after all the processing.

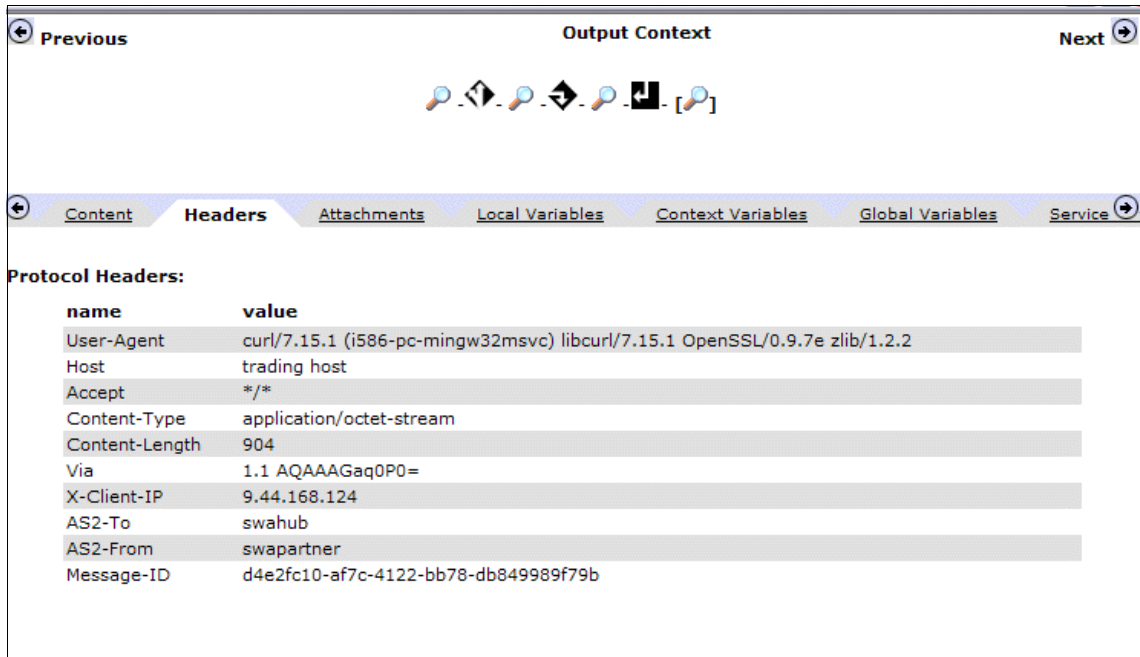


Figure 14-14 Output context coming from our test results

Part 4



Appendixes



A

Additional material

This book refers to additional material that can be downloaded from the Internet as described.

Locating the Web material

The Web material associated with this book is available in softcopy on the Internet from the IBM Redbooks publications Web server. Point your Web browser at:

<ftp://www.redbooks.ibm.com/redbooks/SG247745>

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with this IBM Redbooks publication form number, SG247745.

Using the Web material

The XB60B2B.zip file contains the sample EDIX12 and XML files that are used to test the five scenarios that are documented in the XB60 AS2 Trading Tutorial.

The XB60_AS2_Trading_Tutorial_v1.pdf document is a step-by-step tutorial that instructs you about how to trade AS2 messages between two B2B Gateways. Its intent is to give you basic experience using the B2B functions of the B2B appliance.

How to use the Web material

Create a subdirectory (folder) on your workstation and unzip the contents of the Web material zip file into this folder.

Abbreviations and acronyms

| | | | |
|----------------|--|--------------|--|
| ACL | access control list | HTTPS | Hypertext Transfer Protocol Secure |
| ANSI | American National Standards Institute | IBM | International Business Machines Corporation |
| API | application programming interface | IP | Internet Protocol |
| AS1 | Applicability Statement 1 | ITSO | International Technical Support Organization |
| AS2 | Applicability Statement 2 | JMS | Java Message Service |
| AS3 | Applicability Statement 3 | JNDI | Java Naming and Directory Service |
| BPM | business process management | LDAP | Lightweight Directory Access Protocol |
| CA | Certificate Authority | MDN | Message Disposition Notification |
| CPU | Central Processing Unit | MIME | Multi-purpose Internet Mail Extensions |
| DER | Distinguished Encoding Rules | OASIS | Organization for the Advancement of Structured Information Standards |
| DES | Data Encryption Standard | ODBC | Open Database Connectivity |
| DNS | Domain Name Server | PKI | Private Key Infrastructure |
| DTD | document type definition | RFH | Rules and Formats Header |
| EAI | Enterprise Application Integration | SOAP | Simple Object Access Protocol |
| EDI | electronic document interchange | SCSI | Small Computer System Interface |
| EDIFACT | Electronic Data Interchange For Administration, Commerce & Transport | SHA | Secure Hash Algorithm |
| EDI-INT | Electronic Data Interchange-Internet Integration | SMTP | Simple Mail Transfer Protocol |
| ERP | enterprise resource planning | SSL | Secure Socket Layer |
| FTP | File Transfer Protocol | TCP | Transmission Control Protocol |
| FSH | Front Side Handler | TPA | Trading Partner Agreement |
| GUI | graphical user interface | UDB | Universal Database |
| HIPAA | Healthcare Information Portability and Accountability Act | UDDI | Universal Descriptions, Discovery and Integration |
| HTML | Hypertext Mark-up Language | | |
| HTTP | Hypertext Transfer Protocol | | |

| | |
|------------------|--|
| UN/CEFACT | United Nations Center for Trade Facilitation and Electronic Business |
| UNTDI | United Nations guidelines on Trade Data Interchange |
| URI | Universal Resource |
| URL | Universal Resource Locator |
| VAN | value-added network |
| WMQ | WebSphere MQ |
| WMB | WebSphere Message Broker |
| WPS | WebSphere Process Server |
| WSDL | Web Services Description Language |
| WTX | WebSphere Transformation Extender |
| XML | Extensible Markup Language |
| XSLT | Extensible Stylesheet Language Transformation |

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks publications

For information about ordering these publications, see “How to get IBM Redbooks publications” on page 394. Note that several of the documents referenced here might be available in softcopy only:

- ▶ *B2B Solutions using WebSphere Partner Gateway V6.0*, SG24-7109
- ▶ *DataPower Architectural Design Patterns: Integrating and Securing Services Across Domains*, SG24-7620
- ▶ *Secure Production Deployment of B2B Solutions using WebSphere Business Integration Connect*, SG24-6457
- ▶ *B2B Appliances: Creating Customer Value Through Exceptional B2B Messaging Performance and Security*, REDP-4524
- ▶ *WebSphere DataPower SOA Appliance: The XML Management Interface*, REDP-4446
- ▶ “Setting up the RAID support on a 9235 device with optional hard drives.” You can read this TechNote at:
<http://www-01.ibm.com/software/integration/datapower/support/>
Search on 1358544 in the Search Support field.
- ▶ *IBM WebSphere Transformation Extender 8.2*, SG24-7693

Other publications

These publications are also relevant as further information sources:

- ▶ B2B Gateway Developers Guide

- ▶ “Managing WebSphere DataPower SOA Appliance configurations for high availability, consistency, and control,” by John Rasmussen, 16 January 2008, developerWorks, at:

http://www.ibm.com/developerworks/websphere/library/techarticles/0801_rasmussen/0801_rasmussen.html

Online resources

These Web sites are also relevant as further information sources:

- ▶ IBM WebSphere DataPower documentation library

<http://www.ibm.com/support/docview.wss?rs=2362&uid=swg24021688>

- ▶ IBM Support

<http://www.ibm.com/software/integration/datapower/support/>

How to get IBM Redbooks publications

You can search for, view, or download IBM Redbooks publications, Redpapers, Technotes, draft publications, and Additional materials, as well as order hardcopy IBM Redbooks publications, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

- access control 52, 56
 - firewall 57
- access control list
 - See ACL
- ACL
 - what is 27, 391
- ANSI 9
- ANSI X12 323
 - what is 25
- appliance management 139
- AS MDNs 95
- AS messages 95
- AS Security 95
- AS2 136
 - B2B messages 274
 - Header values 340
 - messages 181
 - over http 178
 - transactions 94
- AS2 Data Flow 69
- AS3
 - transactions 94
- AS3 data flows 99
- ASCII 28
- Audit Log 157
- authentication 53, 56
- automated configuration 139

B

- B2B
 - Data Persistence 91
 - Gateway 143
 - Gateway Service 91
 - Gateway Services 149
 - hub 93
 - Objects 92
 - Partner Profiles 91, 140
 - Transaction Viewer 91, 109
 - Viewer Management Service 91
- B2B Data Persistence
 - Document Storage 120
 - Transaction Store 119

- B2B Gateway 84
- B2B Governance 174
- B2B security 52
- B2C (business-to-consumer) 5
- B2M2B (business-to-marketplace-to-business) 7
- Basic XSLT transformation 324
- Binary 101
- boot sequence 135
- business-to-business (B2B) 5
 - requirements 22
- business-to-business integration (B2Bi) 6
- business-to-consumer (B2C) 5
- business-to-marketplace-to-business (B2M2B) 7

C

- certificate 28
- command-line interface (CLI) 135, 138
- configuration 133
 - automated 139
 - options 137
- connection security 52–53
- cXML 9

D

- data structures for EDI 7
- DataPower Multi-Protocol Gateway (MPGW) 324
- DataPower SOA Appliances 139
- de-militarized zone 52, 57
- deployment security 52
- DHCP (Dynamic Host Configuration Protocol) 123
- digital signature 28
- disadvantage of SSL 53
- DMZ 52, 57, 176
- DNS address 81
- document integrity 56
- document security 52, 55
- document type definition 29
- DTD
 - See document type definition

E

- e-business 4

- ebXML 9
- e-commerce 4
- EDI 4
 - and e-commerce 4
 - AS1 25
 - AS2 25
 - cXML 9
 - document exchange 14
 - ebXML 9
 - EDIFACT 8
 - first era 7
 - fourth era 9
 - message format standards 25
 - messaging 15
 - network bandwidths 15
 - over the Internet 25
 - over value-added network (EDI/VAN) 25
 - third era 8
 - UN/CEFACT 9
 - VAN 10
 - what is 25
 - xCBL 9
 - XML 8
 - XML based message formats 15
- EDI and Mapping 38
- EDI data structures 8
- EDI documents 84
- EDI over the Internet
 - See EDI-INT
- EDI over value-added network
 - See EDI/VAN
- EDI/VAN (EDI over value-added network) 25
- EDIFACT 8
 - what is 25
- EDIFACT documents 323
- EDI-INT
 - AS1 25
 - AS2 25
 - what is 25
- EDIX12 documents 94
- Electronic Business using Extensible Markup Language 29
- electronic data interchange
 - See EDI
- Electronic Data Interchange (EDI) 173
- e-Marketplace 7
- encryption 27
 - Public Key Infrastructure 27
 - secret key cryptography 27

- Ethernet interface 81
- Ethernet port 135
- eXtensible Access Control Markup Language (XACML) 64

F

- firewall 57
 - demilitarized zone 57
- Firewalls 57
- Front Side Handler (FSH) 337
- Front Side Protocol Handlers 101
- FTP 26
 - what is 26
- FTP Poller 101
- FTP Server 101
- FTP Server Front Side Handlers 349

H

- hard drive 80
- hashing 28
 - message digest algorithm (MD5) 28
 - secure hash algorithm (SHA) 28
- HB_Ext External Profile 330
- Health Care Claim Processing 173
- Healthcare Information Portability and Accountability Act
 - See HIPAA
- hiding the private IP address 58
- HIPAA 26
 - compliance 173
 - EDI 174
 - EDI Pack 173
 - EDI X12 Version 5010 174
 - Partner 179
 - Partner internal profile 180
 - Provider 176
 - trading 172
- Host Alias 146
- HTTP 26, 81, 134
 - FSH 136
 - what is 26
- HTTP FSH 193
- HTTP/SSL 275
- HTTPS
 - interface 140

I

- industry-oriented XML data structures 8
- integrity 56
- international XML data structures 9
- IP translation and mapping 58
- iSCSI 103, 121
 - Reference Objects 121
- iSCSi
 - Host Bus Adapter 122

J

- JMS 134

M

- management interface 138
- MD5 28
- MD5 (message digest algorithm) 28
- message 24
- message digest algorithm (MD5) 28
- message format standards 25
- message integrity 53
- message queuing (MQ) 23
- messaging and queuing 23
- metadata 29
- MIME 28
- MIME and FTP 25
- MIME and HTTP 25
- MIME and SMTP 25
- MIME Content-ID header 379
- MQ 134
- MQ (message queuing) 23

N

- NAT 58
- network access control 57
- Network Address Translation 58
- NFS 134
 - Client Settings 129
 - Export 129
 - Reference Objects 129
 - Static Mounts 129
- NFS (Network File System) 128
- NFS file 103
- NFS Poller 101
- nonrepudiation 56

O

- OASIS 9
- ODETTE 25
- On Demand Business 4
- Organization for Data Exchange through Teletransmission in Europe
 - See ODETTE
- Organization for the Advancement of Structured Information Standards
 - See OASIS

P

- Partner B2B Gateway Service 190
- PAT 59
- PKI (Public Key Infrastructure) 27
- PKI, 27
- Port Address Translation 59
- privacy 53, 56
- proxy server, 57
- public key cryptography 27
- Public Key Infrastructure (PKI) 27

Q

- queue 26
- queuing 24

R

- RAID Volume 84
- RBM
 - functionality 107
 - policy 146
- RBM policy 108, 110
- RBM techniques 107
- Redbooks Web site 394
 - Contact us xiv
- requirements
 - for B2B 22
- reverse proxy 57
- Role Based Management (RBM) 133
- RosettaNet 8–9

S

- S/MIME 28
- schema 29
- secret key cryptography 27
- Secure hash algorithm (SHA) 28
- secure hash algorithm (SHA) 28

- Secure Sockets Layer 29, 53
- Secure Sockets Layer (SSL) 29
- security 27
 - ACL 27
 - certificates 28
 - digital signatures 28
 - encryption 27
 - hashing 28
- Security Assertion Markup Language (SAML) 64
- Service Level Agreement 140
- SHA (secure hash algorithm) 28
- share Internet access 58
- Single Sign-on (SSO) 64
- SMTP 26
 - IMAP 26
 - POP3 26
 - what is 26
- SOAP
 - Management URI (SOMA) 139
 - message 140
- SOAP over HTTP (or JMS) 367
- SOAP over HTTPS 139
- SOAP with Attachments 367
- Software Configuration Management (SCM) 149
- SOMA interface 139
- SQL 38
- SSH 81
- SSH client. 140
- SSL 53, 174
- SSL (Secure Sockets Layer) 29
- SSL connection 53
- SSL handshake 53–54
- SSL protocol 53
- SSL session 53
- symmetric key encryption 27
- System Log 155
- System Logs 152

T

- TCP Connect 152
- Transaction Viewing with RBM 107
- transport protocol 26
 - FTP 26
 - HTTP 26
 - SMTP 26
- Troubleshooting the Appliance 151
 - Audit Log 157
 - Diagnose Appliance Problems 154

- Packet Capture 153
- System Log 155
- Transaction Viewer 155

U

- UCS 26
- UDDI (Universal Description, Discovery, and Integration) 30
- UDDI server 30
- UN/CEFACT 9
- United Nations Trade Data Interchange (UNTDI) 25
- United Nations Trade Data Interchange Standards
 - See UNTDI
- Universal Description, Discovery, and Integration (UDDI) 30
- Universal Multi-Octet Coded Character Set
 - See UCS
- UNTDI (United Nations Trade Data Interchange) 25

V

- value-added network
 - See VAN
- VAN
 - and the Internet 10
 - evolution 10
 - service provider 15
- VDA 26
- VICS (Voluntary Inter-industry Communications Standards) 26
- Voluntary Inter-industry Communications Standards (VICS) 26

W

- Web services 30
 - UDDI 30
 - WSDL 30
- Web Services Description Language (WSDL) 30
- Web Services Distributed Management (WSDM) 64
- WebGUI 83
- WebGUI interface 111, 137
- WebSphere Business Integration Connect 53
- WebSphere DataPower B2B Appliances XB60 174
- WebSphere MQ 174
- WebSphere Transformation Extender 175
- WS-Addressing 64
- WSDL 30

WSDL (Web Services Description Language) 30
WS-I Profiles 64
WS-Policy 64
WS-ReliableMessaging 64
WS-Security 64
WTX Launcher 177

X

X.509 54
XB60
 domain 149
 Specific functionality 140
XB60 administrator 107
XB60 B2B Services 92
XB60 browser interface 56
XB60 DataPower 148
xCBL 9
XML
 commands 144
 cross-industry standards 9
 cXML 9
 document definition 29
 ebXML 9
 for e-commerce 9
 Format 100
 growth of 8
 Management API 140
 Management Concepts 144
 Manager 101
 OASIS 9
 RosettaNet 8
 tag languages 29
 UN/CEFACT 9
 what is 29
 xCBL 9
XML Management Interface 137
XML Management interface 139
XML-formatted EDI 323
XPath 94
XPath Routing Policies 104
XPath statements 104
XPath Tool 104
XSD 140
XSLT 39, 134, 324
 stylesheets 134



IBM WebSphere DataPower B2B Appliance XB60 Revealed

(0.5" spine)
0.475" <-> 0.875"
250 <-> 459 pages



IBM WebSphere DataPower B2B Appliance XB60 Revealed



**Extend integration
beyond the
enterprise with IBM
B2B Appliance**

**Easily connect to
trading partners
using industry
standards**

**Simplify deployment,
configuration, and
management**

The IBM WebSphere DataPower B2B Appliance XB60 is a purpose-built, easy-to-use appliance that incorporates business-to-business (B2B) and integration functions into a consolidated B2B-i solution, providing a high-performance and security-enhanced B2B solution for trading partner connections.

Part 1 of this IBM Redbooks publication provides a brief introduction to B2B and how the technology has evolved over the years along with a description of the most common B2B technologies and an overview of the IBM Software Services for WebSphere B2B Deployment methodology.

Part 2 of this book provides step-by-step information related to the installation and configuration of the appliance and troubleshooting tips to use to resolve common configuration issues.

Part 3 of this book provides common B2B scenarios that demonstrate how to integrate with other IBM products and how to trade EDI, XML, and binary data utilizing AS2 and AS3 B2B messaging protocols.

This book was developed as a guide for anyone who is interested in deploying B2B integration solutions utilizing purpose-built appliances.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:
ibm.com/redbooks**