WebSphere software

IBM

B2B Solutions using WebSphere Partner Gateway V6.0

- Implement B2B solutions using WebSphere Partner Gateway
- Explore the new and extended EDI features
- Utilize the newly imbedded mapping capabilities

Lee Gavin
Ronan Dalton
Matthew Lishok
Vamsi Krishna Namuduri
Sreelatha Sankaranarayanan

Redbooks

ibm.com/redbooks

Archived

Archived

International Technical Support Organization

B2B Solutions using WebSphere Partner Gateway V6.0

November 2005

Archived

SG24-7109-00

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

Archived

First Edition (November 2005)

This edition applies to Version 6, Release 0, Modification 0 of WebSphere Partner Gateway Enterprise (product number 5724-L69).

© Copyright International Business Machines Corporation 2005. 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 xiii
Trademarks xiv

Preface xv
The team that wrote this redbook xvi
Become a published author xvii
Comments welcome xvii

Part 1. Overview of the technology and products 1

Chapter 1. Business-to-business concepts 3
1.1 Impact of the Internet on the world of 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 12
1.4.1 Types of B2B integration 13
1.4.2 Summary 19

Chapter 2. Business-to-business technologies and standards 21
2.1 Requirements for a B2B solution 22
2.2 Some 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 26
2.2.5 Java 29
2.2.6 Extensible Markup Language 29
2.2.7 Web services 29

Chapter 3. Introduction to WebSphere Partner Gateway 31
3.1 Editions of WebSphere Partner Gateway 32
3.2 Architecture of WebSphere Partner Gateway 32
3.2.1 Runtime components 33
3.2.2 Configuration components 37
3.2.3 Profile management 39

Archived

© Copyright IBM Corp. 2005. All rights reserved.

3.3 B2B features and standards 40
3.3.1 Transport options 40
3.3.2 Messaging protocol options 41
3.3.3 Business document formats 41
3.3.4 Security options in WebSphere Partner Gateway 41
3.4 Extensible architecture 42
3.4.1 Encryption, validation and transformation 42
3.4.2 Custom transports, packages and protocols 42
3.5 Enterprise Integration 43

Part 2. Building a B2B exchange 45

Chapter 4. Implementation scenarios 47
4.1 Implementing WebSphere Partner Gateway Enterprise for Windows 48
4.2 Implementing WebSphere Partner Gateway Advanced for AIX 48
4.3 Implementing a basic exchange 49
4.4 Securing the exchange between Company E and Company A 50
4.5 Implementing WebSphere Partner Gateway Express for Windows 52
4.6 Using FTP with WebSphere Partner Gateway 53
4.7 Using FTP Scripting to enable Value Added Network (VAN) connectivity 54
4.8 Managing the B2B infrastructure 55
4.9 EDI translation 56

Chapter 5. WebSphere Partner Gateway Enterprise on Windows 59
5.1 Implementation overview 60
5.2 Verify software levels on the hub and data servers 60
5.2.1 Verifying WebSphere MQ 6.0 61
5.2.2 Verifying DB2 62
5.3 Installing the software for the data machine 63
5.3.1 Adding user IDs and a group 63
5.3.2 Configuring WebSphere MQ 6.0 65
5.3.3 Installing the database schema 79
5.3.4 Local validation 85
5.4 Install WebSphere Partner Gateway 86
5.4.1 Adding a user and group 86
5.4.2 Installing the product code 87
5.4.3 Local validation 102
5.5 Initial configuration of the WebSphere Partner Gateway server 104

Chapter 6. WebSphere Partner Gateway Advanced for AIX 109
6.1 Implementation Overview 110
6.2 Verifying software levels on the AIX machine 110
6.2.1 Verifying DB2 110
6.2.2 Verifying WebSphere MQ 111

Archived

6.3 Software Installation 111
6.3.1 Adding the user accounts 111
6.3.2 Configuring WebSphere MQ 112
6.3.3 Installing the database schema. 114
6.3.4 Installing the Product Code 122
6.3.5 Local Validation. 132
6.3.6 Starting the help system 135
6.4 Initial configuration of the WebSphere Partner Gateway server 135

Chapter 7. Creating a basic B2B exchange 137
7.1 Scenario overview. 138
7.2 Role-based configuration 138
7.2.1 Outbound flow. 140
7.2.2 Inbound Flow. 142
7.3 Configuration tasks for hubadmin of Company E 143
7.3.1 Creating targets. 144
7.3.2 Creating interactions 149
7.3.3 Creating a Community Manager 155
7.3.4 Creating a Community Participant 161
7.4 Configuration tasks for Company E administrator. 164
7.4.1 Initial logon by the Community Manager. 165
7.4.2 Creating a gateway 166
7.4.3 Providing B2B capabilities. 171
7.5 Configuration tasks for partner Company A 174
7.6 Connecting Company E to Company A. 180
7.7 Configuration tasks for the Company A hubadmin 188
7.8 Validating communication 196
7.9 Revisiting role-based configuration 202

Chapter 8. Securing the B2B exchange 207
8.1 What is needed to perform encryption an decryption 208
8.2 Enabling encryption. 211
8.2.1 Company E generates a public/private key pair 212
8.2.2 Company E uploads private key to its own server 217
8.2.3 Company E uploads public certificate to partner's server. 220
8.2.4 What happens next? 226
8.2.5 Company A generates a private/public key pair 226
8.2.6 Company A uploads a private key to its own server 231
8.2.7 Company A uploads their public key to their partner's server. 235
8.2.8 Updating the participant connections 240
8.2.9 Validating that encryption is enabled 241
8.3 What is needed to digitally sign and verify the signature? 243
8.4 Enabling digital signatures 246

Archived

8.4.1 Changes to be performed on the server of Company A 246
8.4.2 Changes to perform on the server of Company E. 253
8.4.3 Validating that digital signatures are enabled 257

Chapter 9. WebSphere Partner Gateway Express on Windows 261
9.1 Overview of the Express edition 262
9.2 Software installation and configuration 263
9.3 Initial configuration of the Express server 269

Chapter 10. Extending the B2B exchange 277
10.1 Scenario overview 278
10.2 Implementation steps 279
10.3 Configuration of Company X 280
10.3.1 Customizing My Profile 280
10.3.2 Customizing the profile of Participant Company E 281
10.4 Additional configuration of Company E 284
10.4.1 Creating a new Community Participant 284
10.4.2 Creating a new document flow definition 285
10.4.3 Creating a new XML format 290
10.4.4 Create an interaction 293
10.4.5 Updating Company X's profile on Company E's server 295
10.4.6 Updating Company E's profile on the Company E server 299
10.5 Validating Communication 302
10.5.1 Sending XML documents from Company E to Company X 302
10.5.2 Sending XML documents from Company X to Company E 303

Chapter 11. Managing the B2B exchange 307
11.1 Tools available to manage the exchange 308
11.2 System log files 309
11.3 Event Viewer 309
11.4 Document Viewer 313
11.5 AS1/AS2 Viewer 315
11.6 Gateway queue 318
11.7 Using the tools for problem determination 321
11.7.1 MDN HTTP URL not defined 321
11.7.2 Problems with encryption 323
11.7.3 Problems with digital signatures 325

Part 3. FTP support 333

Chapter 12. Integrating FTP servers 335
12.1 Overview of FTP and FTP Scripting scenarios 336
12.1.1 FTP method one: FTP servers 337
12.1.2 FTP method two: FTP Scripting 338

Archived

12.2 Configuration of Company A for FTP outbound 338
12.2.1 Outbound implementation steps 339
12.2.2 Creating the directory structure 341
12.2.3 Updating the hubadmin profile 343
12.2.4 Updating the profile for Company F 350
12.2.5 Updating the profile for Company A 354
12.2.6 Creating a participant connection 355
12.2.7 Validating outbound communication 357
12.3 Configuration of Company A for FTP Inbound 360
12.3.1 Integrating with the AIX FTP server 361
12.3.2 Updating the configuration of WebSphere Partner Gateway 366
12.3.3 Validating Inbound communication 368
12.4 Implementing FTPS 371
12.4.1 FTPS outbound 371
12.4.2 FTPS client authentication 383
12.4.3 FTPS inbound 384
12.5 Configuration of FTP Scripting outbound 384
12.5.1 Outbound FTP Scripting overview 385
12.5.2 Updating the B2B Capabilities of Company F 386
12.5.3 Creating the new interaction 387
12.5.4 Create an FTP Scripting Gateway for Company F 389
12.5.5 Update the participant connection 397
12.5.6 Validating the connection 399
12.6 Configuration of FTP Scripting inbound 403
12.6.1 FTP Scripting inbound overview 404
12.6.2 Creating the FTP Scripting Target 405
12.6.3 Updating the participant connection 410
12.6.4 Validating inbound communication with FTP Scripting target 412
12.7 FTPS for FTP Scripting Gateways and Targets 413
12.7.1 FTPS for the FTP Scripting Gateway 414
12.7.2 FTPS for the FTP Scripting Target 415
12.8 Comparing FTP and FTP Scripting 417

Chapter 13. Enabling VAN connectivity using FTP Scripting 419
13.1 Scenario overview 420
13.2 Configuration of Company E for outbound 421
13.2.1 Implementation steps 421
13.2.2 Updating the profile of hubadmin 422
13.2.3 Updating the profile of Company V 426
13.2.4 Updating the profile of Company E 431
13.2.5 Creating a participant connection 431
13.2.6 Validating communication 433
13.3 Configuration of Company E for inbound 434

Archived

13.3.1 Creating a FTP Scripting Target for Company E 434
13.3.2 Updating the WebSphere Partner Gateway configuration 438
13.3.3 Validating communication 439
13.4 Securing VAN connectivity 440
13.4.1 Uploading certificates to Company E's WebSphere Partner Gateway 440
13.4.2 Updating the FTP Scripting gateway to use secure mode 443
13.4.3 Validating outbound communication 445
13.4.4 Using secure mode for inbound communication 446
13.4.5 Validating inbound communication 447
13.5 Summary 448

Part 4. Native mapping support for non-EDI standards 449

Chapter 14. Native mapping support in WebSphere Partner Gateway . 451
14.1 Install the Data Interchange Services client 452
14.2 The system view of Data Interchange Services 456
14.3 Development environment 456
14.3.1 Configure a development environment using DB2 457
14.4 Component view of the development environment 459
14.5 Connect the Data Interchange Services client to a development database 460
14.6 Functional areas 461
14.6.1 XML functional area 461
14.6.2 EDI functional area 463
14.6.3 Record Oriented Data (ROD) functional area 465
14.6.4 Mapping functional area 467
14.7 Runtime environment 470

Chapter 15. Mapping 473
15.1 Scenario overview 474
15.2 Create a ROD document definition 475
15.2.1 Create a new ROD dictionary 476
15.2.2 Create a new Record ID 476
15.2.3 Create a new Record Oriented Document definition. 477
15.3 Import XML Schema 484
15.4 Create a data transformation map. 489
15.5 Create a connection to the runtime database 511
15.6 Configure WebSphere Partner Gateway at Company E 514
15.6.1 Verify that the map has been uploaded 514
15.6.2 Create a target for ROD document 516
15.6.3 Update the configuration for Company E 520
15.6.4 Update the configuration for Company X 523
15.6.5 Create an interaction. 528

Archived

15.6.6 Create participant connections 533
15.7 Validating communication 536

Chapter 16. Advanced mapping 541
16.1 MapSwitch() 542
16.1.1 Exporting the maps to a runtime database 553
16.2 Configure WebSphere Partner Gateway at Company E 555
16.2.1 Verify that the map has been uploaded 556
16.2.2 Update the configuration of Company E 559
16.2.3 Update the configuration for Company X 561
16.2.4 Update the configuration for Company A 563
16.2.5 Create an interaction 564
16.2.6 Create participant connections 565
16.3 Validate communication 569

Part 5. Native mapping support for EDI standards 575

Chapter 17. Introduction to EDI technology 577
17.1 EDI terms and concepts 578
17.2 Benefits of EDI 580
17.3 EDI components 581
17.3.1 Message standards 581
17.3.2 Communication 583
17.4 The evolution of EDI 585
17.4.1 Elements of an EDI solution 587

Chapter 18. Mapping for EDI transformation and validation 589
18.1 XML_EDI_ORDER 590
18.2 Mapping commands 592
18.2.1 BEG Segment 592
18.2.2 N1 Loop 596
18.2.3 PO1 Loop 597
18.3 EDI_XML_PO-ACK 601
18.4 353X410_VALIDATE 612

Chapter 19. Enveloping and de-enveloping 621
19.1 About EDI validation 622
19.1.1 Data validation 622
19.1.2 Service Segment validation 623
19.2 Scenario overview 624
19.3 Implement the enveloping scenario 625
19.4 Configure Company E for XML to EDI outbound 627
19.4.1 Enable JMS 627
19.4.2 Create a JMS Target 630

Archived

19.4.3 Transformation map 631
19.4.4 Creating a new XML format 634
19.4.5 Upload the XML Validation Map (XSD) 637
19.4.6 Add attributes to the document flow definition 640
19.4.7 Create the custom action 643
19.4.8 Create interactions 645
19.4.9 Update the Participant Profiles 648
19.4.10 Configure the enveloper 649
19.4.11 Activate the participant connections and attribute selection 657
19.5 Configure Company A to receive EDI documents 666
19.6 Sending and validating the communication 667
19.6.1 Short interval validation 668
19.6.2 Long interval validation 672
19.7 Implement the de-enveloping scenario 674
19.7.1 Configure Company A to post an EDI-X12 interchange 674
19.7.2 Uploading the EDI to XML transformation map 675
19.7.3 Upload the EDI validation map 677
19.7.4 Create the interactions for the document flow definition 680
19.7.5 Update the participant profiles 682
19.8 Receiving and validating the communication 688
19.8.1 Positive validation 688
19.8.2 Negative validation 692

Chapter 20. Functional acknowledgements 695
20.1 Structure of an FA 696
20.1.1 Analyzing a TA1 Transaction 696
20.1.2 Analyzing a 997 Transaction 697
20.2 Implementing our FA scenario 699
20.3 Configure Company E to generate the FAs 700
20.3.1 Associating the FA map with the EDI document flow definition . . 700
20.3.2 Set the attributes to trigger the FA generation 702
20.3.3 Create interactions 707
20.3.4 Updating the participant profiles 710
20.3.5 Activate the participant connections and attribute selection 712
20.4 Configure Company A for receiving EDI 717
20.5 Sending and validating the documents 718
20.5.1 Scenario1 718
20.5.2 Scenario2 721
20.5.3 Scenario3 724

Chapter 21. Batching and EDI splitting 729
21.1 Batching and splitting basics 730
21.2 EDI Batching scenario 730

21.2.1 Batching Scenario Overview 730
21.2.2 Configuration for Company E 731
21.2.3 Configuration for Company A 754
21.2.4 Validating the flow 754
21.2.5 Using batching to alter output 756
21.2.6 Batching review 766
21.3 Batching with other transformations 766
21.4 EDI Splitter 767

Chapter 22. EDI message tracking 771
22.1 EDI message tracking overview 772
22.2 Drill up and down capability 772
22.2.1 Drill down capability 773
22.2.2 Drill up capability 780
22.3 EDI Reports 786
22.3.1 EDI FA Overdue Report 786
22.3.2 EDI Rejected Report 787

Part 6. Appendixes 789

Appendix A. Additional material 791
Locating the Web material 791
Using the Web material 791
System requirements for downloading the Web material 792
How to use the Web material 792

Related publications 793
IBM Redbooks 793
Online resources 793
How to get IBM Redbooks 793
Help from IBM 794

Index 795

Archived

Archived

Notices

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

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

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

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

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

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

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

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

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

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

© Copyright IBM Corp. 2005. All rights reserved.

Trademarks

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

AIX®	Lotus®	Trading Partner®
DB2®	MQSeries®	WebSphere®
@server®	Redbooks™	Workplace™
@server®	Redbooks (logo) ™	Workplace Collaborative Learning™
IBM®	SupportPac™	

The following terms are trademarks of other companies:

Java, Java Naming and Directory Interface, JVM, J2EE, 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, and service names may be trademarks or service marks of others.

Archived

Preface

This IBM® Redbook introduces you to business-to-business (B2B) solutions based on IBM WebSphere Partner Gateway.

In Part 1, we learn about B2B technologies and features, architecture and integration options of WebSphere Partner Gateway.

In Part 2, we describe the implementation of three editions of WebSphere Partner Gateway on Microsoft® Windows® and AIX®. Within an environment of four Trading Partners, we explore step-by-step how to implement various B2B scenarios. This part also demonstrates how to implement an AS2 exchange of electronic data interchange (EDI) documents and custom XML documents. It also shows how to secure such exchanges with digital signatures and encryption.

In Part 3, we learn how to integrate an FTP server with WebSphere Partner Gateway so that we still have the visibility and manageability of WebSphere Partner Gateway for Trading Partners who are wanting to use a more traditional FTP approach. We also explore the new FTP scripting functionality which enables easily configured and maintained interaction with VANs.

In Part 4, we discuss the newly integrated Data Interchange Services (DIS) client. We explore the mapping capabilities, from simple mapping to mapping and validation of more complex document structures such as those that might be found in a legacy system which are known as Record Oriented Data. We also discuss soem of the advanced functions provided for mapping.

In Part 5, we continue to explore the integrated mapping, but move on to EDI message standards. We learn about the use of EDI validation maps We look at the function available for mapping and validating these complex message types. We also explore the newly added functions for things such as enveloping and de-enveloping of EDI interchanges, generating EDI functional acknowledgements and EDI batching and splitting. We also see the functions that are available for the tracking of EDI messages and how we report things such as overdue or rejected functional acknowledgments, and explore the new drill-down and drill-up reporting.

© Copyright IBM Corp. 2005. All rights reserved.

The team that wrote this redbook

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

Lee Gavin is a Consulting IT Specialist at the ITSO, Raleigh Center. She writes extensively and teaches IBM classes worldwide on all areas of the WebSphere® family, WebSphere Business Integration and Business Process Management. Before joining ITSO in May 2001, Lee worked for IBM Global Services, Australia where she specialized in Middleware and Integration solutions for customers .

Ronan Dalton is a Software Engineer working on IBM Workplace™ Collaborative Learning with the Dublin Software Lab. Prior to joining Lotus®, Ronan worked as a Consultant with the IBM Software Services for WebSphere team. He has been working with both WebSphere Data Interchange and WebSphere BI Connect since their inclusion into the WebSphere Portfolio of software. Ronan co-authored the IBM Redpaper WebSphere Data Interchange Installation and Configuration, REDP-3600, the IBM Redbooks™ Implementing EDI Solutions, SG24-6906 and B2B Solutions using WebSphere BI Connect Version 4.2.2, SG24-6355. He has also published several technical papers that are available on WebSphere Developer Domain and has developed SupportPacs for both WebSphere Data Interchange and WebSphere BI Connect. Ronan holds a Post Graduate Diploma in Computing and an MSc in Computing from Griffith College Dublin.

Matthew Lishok is a Software Engineer working with IBM in Raleigh, North Carolina in the United States. He has worked at IBM for seven years, primarily in system and functional test. He has tested products in IBM's Networking Hardware and Storage Systems divisions. The last three years have been on WebSphere Business to Business software. He holds a degree in Computer Engineering from the University of Florida.

Vamsi Krishna Namuduri is a Software Engineer in IBM Software Labaratories, India. He has two years of experience in functional and Integration test. He holds a degree in Electronics & Communications Engineering from Osmania University, Hyderabad,India. He has been working in the core development team of WebSphere Partner Gateway for the past 18 months.

Sreelatha Sankaranarayanan is a Software Engineer in IBM India. She has over four years experience in Java™, J2EE™ and Internet technologies. She holds a degree in Software Systems from BITS, India.She has been wih the WebSphere group for the past two years.

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

Keith Clayton
IBM Software Group, Application and Integration Middleware Software, WebSphere Data Interchange Development

Larissa Leybovich
IBM Software Group, Application and Integration Middleware Software

Pete Murphy
IBM Software Group, EMEA Technical Sales, WebSphere B2B

Margaret Ticknor and Geert Van de Putte
International Technical Support Organization, Raleigh Center

Become a published author

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

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

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

`ibm.com/redbooks/residencies.html`

Comments welcome

Your comments are important to us!

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

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

  `ibm.com/redbooks`

- Send your comments in an email to:

  `redbook@us.ibm.com`

Archived

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. 8IB; HZ8 Building 662
  P.O. Box 12195
  Research Triangle Park, NC 27709-2195

Archived

Part 1

Overview of the technology and products

Archived

© Copyright IBM Corp. 2005. All rights reserved.

Archived

1

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. Lastly, this chapter compares EAI and B2B, by examining the similarities and differences that make them separate concepts and processes.

Archived

© Copyright IBM Corp. 2005. All rights reserved.

1.1 Impact of the Internet on the world of 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 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 can 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 can face more risks, but also harvest the rewards of creating and sustaining some real competitive advantages. Those practicing a wait-and-see strategy might not get locked out of the game, but 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. E-commerce 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 into whether they involve consumption, creation, transformation, or provision of goods and services, some combination of these, 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. This 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 some 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, would fall into the e-Marketplace category.

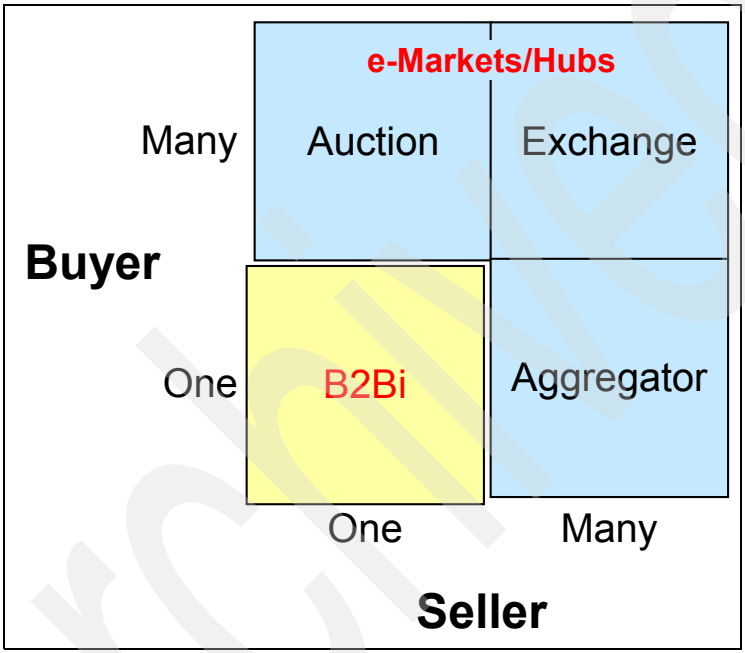


Figure 1-1 Relationship between B2Bi and e-Marketplaces

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 of this is supply chain applications or Trading Partners that engage in some exchange.

Archived

The remainder of this book focuses on B2Bi. For simplicity in this redbook, 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. Alternatively, they can be private, invitation-only communities with members who 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 could 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 as retail, transportation, automotive, and so forth. Besides, this kind of work was occurring in different countries, resulting in the standards which 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 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 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 VDA SEDAS in Germany.

Gradually, the different industry groups came together to create one international data standard: Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT). Today, most trade between companies in two different 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 humans and machines. The availability of XML meant that users with a browser could 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 mentioned above 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.

Some 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 end users, defining the documents. They both have previously worked in the e-commerce-standards arena. This ensures that strong data dictionaries exist, as do 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 interenterprise computer-to-computer data exchange moved data with primitive computer protocols. At that time, numerous different communication protocols could be used. Some 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 such 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 different 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 could insulate the protocol of a given Trading Partner from the protocols used by all the others. The VAN offered protocol conversion and insulated one company from the other so neither company logged onto 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 would replace the VAN as a network intermediary. Leaving the VANs, early Internet adopters hoped that it would become the universal protocol and would 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 FTP, HTTP, and 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 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 should 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.

We can differentiate three EAI levels. This is an important aspect of the IBM business integration strategy. The strategy positions the WebSphere Business Integration family of products in three tiers:

- The lowest level of EAI is sending information directly from one application to another, for example, using WebSphere MQ messaging. This is known as *point-to-point application integration architecture*. Logic and formatting for data exchange is totally within the applications.
- Where there are many applications involved, information needs to be intelligently routed to where it is needed. It might have to be transformed or reformatted, for example, using WebSphere Message Broker or WebSphere Process Server. The connected applications do not need to know about required data formats, but the business logic that starts the exchange is still inside the calling application. This represents the *hub-and-spoke application integration architecture*.
- At the highest level of integration, a business process is represented by applications started by specific business conditions (or business logic). Here, the process is controlled by a *workflow manager*, such as WebSphere MQ Workflow, and supports sophisticated multistep process flows. WebSphere MQ Workflow also works with applications, while allowing human intervention. There is no need for the applications to know anything about the overall business process, which is defined in a business process model. The business process, controlled and executed by a workflow manager, knows when each application has to be invoked and what data is required. The applications do not link to each other. They send results back to a workflow manager. Then the workflow manager uses the business process definition to decide what happens next.

The WebSphere Business Integration family offers the unique advantage that all three levels may be easily combined within an overall solution, based on their common foundation on WebSphere MQ messaging and XML.

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).

Today, 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 which trade with partners and suppliers over the Internet in real time. It is about using middle ware technologies, such as distributed objects, remote procedure calls, message queueing, data transformation, and publish/subscribe, to connect different 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 structures (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 suppliers 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

  Different applications and users represent information in different ways or use different 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 different organizations within the enterprise are composed of various applications. These 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

  Different 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 different 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 interenterprise processes must respect organizational autonomy and minimize the complexity of mutual commitments among different organizations.

1.4.1 Types of B2B integration

The following sections distinguish three major types of B2B integration:

- Data and applications sharing
- Document exchange
- Process integration

Data and applications sharing

The data and applications sharing type of B2Bi solution makes a set of data and applications available for direct access by outside organizations. This is usually done through an organization's Web FTP server or messaging middle ware,

such as WebSphere MQ. Multiple URLs are dedicated for this purpose and let outside companies access information in the form of HTML, XML or some agreed proprietary form. Returned information can be static (come from the internal database) or invoke execution of the internal enterprise applications (see Figure 1-2). This approach works well for simple interactions. The main advantage is that it requires almost no specialized software or hardware investment from participants.

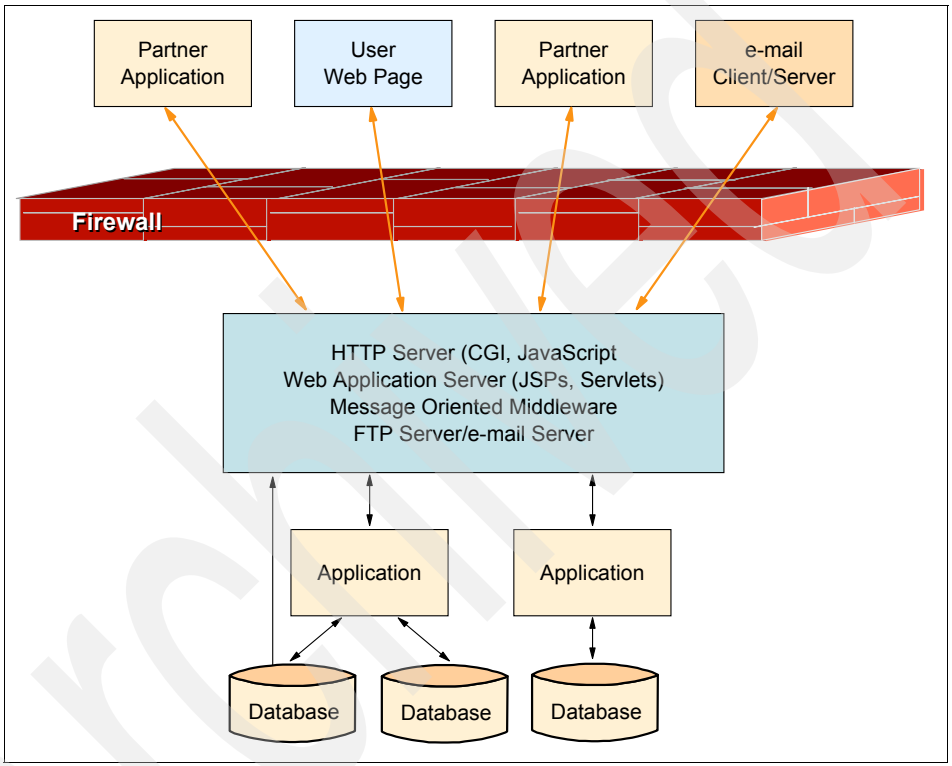


Figure 1-2 Data and information sharing B2Bi

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 and 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 where this data originates from. 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 different types of documents (such as EDI documents or XML documents) can be delivered. The difference between this approach and data and applications sharing is that this can be push or pull communications style, providing more controlled timing for the information exchange. An enterprise with information *pushes* this information to all interested parties, the interested party can choose to *pull* that data at a time that is convenient for them, for example. 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. This 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 or Java Message Service (JMS) to achieve assured message delivery. Because the public Internet is essentially free, message size is not a factor. This 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 be interacted, 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 might be a good starting point for the implementation, it is

usually not a viable solution for the final system. This is due to the amount of code necessary to implement such features as guaranteed document delivery and document data transformation, which are part of any modern B2Bi server. We recommend that you implement a 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).

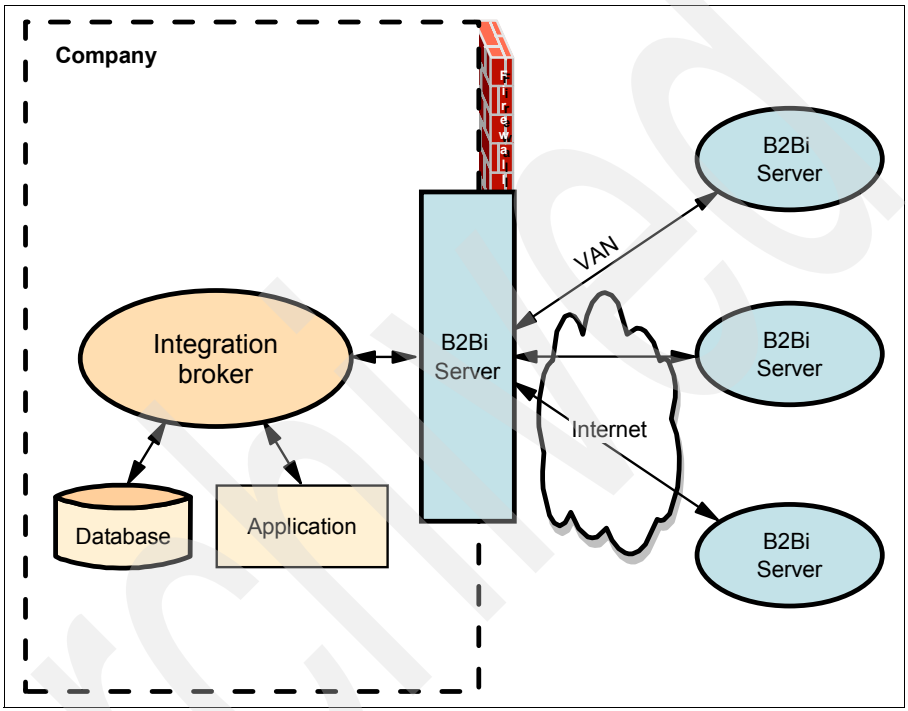


Figure 1-3 Using an internal EAI with B2Bi server

Now, think about application integration. Each time you add a new application to the existing integration infrastructure, its complexity grows. To avoid the interapplication spaghetti phenomenon, in this case, we implement a integration broker. Each application talks with only one integration broker and it is not aware of other applications. That is the transition from point-to-point to hub-and-spoke architecture.

Archived

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

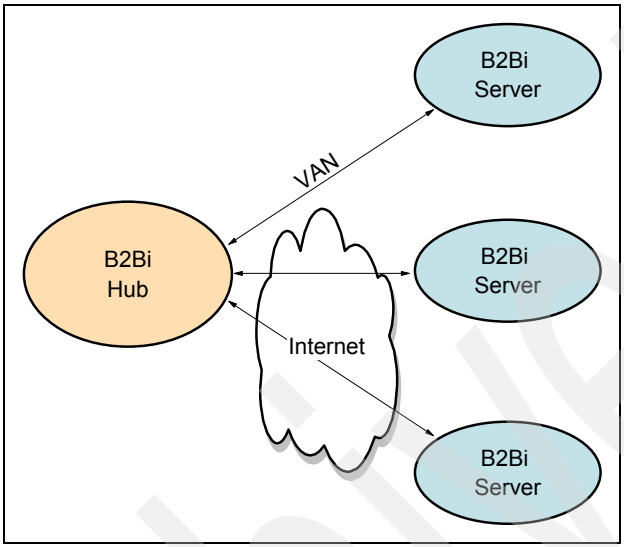


Figure 1-4 B2Bi hub

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 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 the basis of this, we can divide business processes into private and public.

Archived

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 using an internal integration broker.

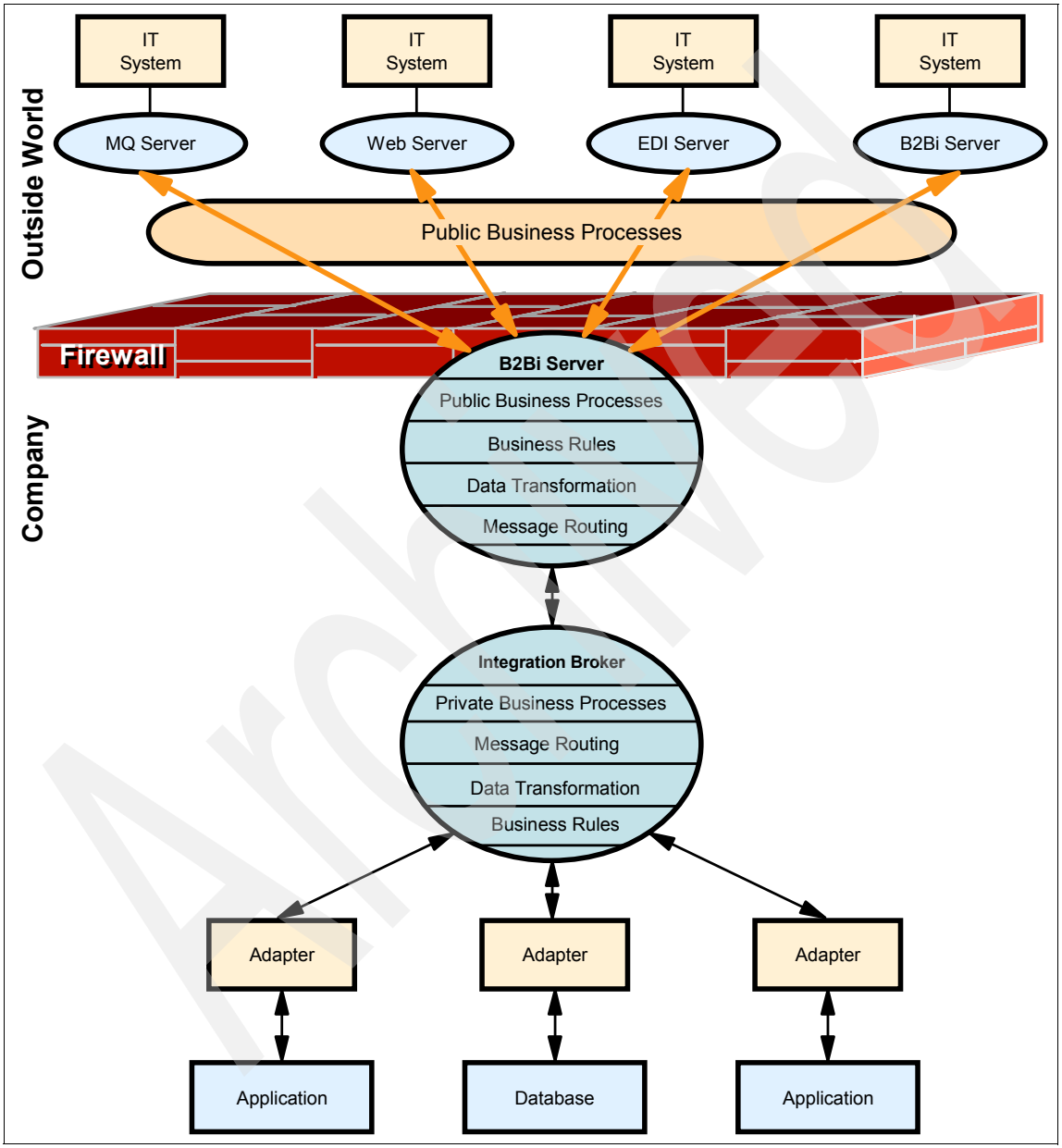


Figure 1-5 B2Bi on a business process level

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. This 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 upon message formats, message sequencing, communications and security protocols, workflow steps and business rules. It also reduces the application code required to execute business processes. In a BPI 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 it does business. This is the most expensive proposition. It requires participating enterprises to be internally integrated first, but it usually provides the greatest benefits.

Archived

Archived

2

Business-to-business technologies and standards

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

© Copyright IBM Corp. 2005. All rights reserved.

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.

These are the requirements for most B2B scenarios, in all or part:

- Ability to send and receive data

  This can be structured or unstructured data across a variety of transport protocols, for example HTTP, FTP, 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 business or applications can participate in this data-sharing. Most B2B applications have the business map their data into a generic data object. This allows for easier manipulation of the data and for easier mapping to different 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 capability with minimal configuration.

- Security of data

  Security needs to be available from the time the data leaves the sending application until it arrives at the receiving application. This 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 taken, 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. Coordination of local transactions and synchronization of distributed transactions is a key issue for data integrity.

- Performance

  The system should 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, then using a Java-based technology might not have a huge impact on the time required to implement a project. However, if only COBOL programming skills are available in-house, using a Java solution would require more work and add expense and time for retraining the programmers. Should the company not use a Java solution? This 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 require using different tools to do the programming.

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

2.2 Some terminology

Here are some examples of the different 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.

Archived

Even so, many complex business transactions are processed today without queuing. In a large network, the system can maintain 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 can run 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 (see Figure 2-1):

- Control information, which contains such information 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

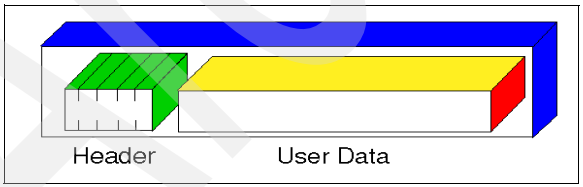


Figure 2-1 MQ message

Archived

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 main purpose of EDI-INT is to reduce the cost of transmission. Two main message transmission standards are used for EDI-INT:

- **AS1**: Uses MIME and SMTP
- **AS2**: Uses MIME and HTTP for process-to-process real time EDI

Message format standards

The following message format standards are used:

- ANSI X12

  American National Standards Institute committee X12 (ANSI X12) defines data which 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)
  - Voluntary Inter-industry Communications Standards (VICS)
  - Verband Deutscher Automobilhersteller (VDA)
  - Universal Multi-Octet Coded Character Set (UCS)

Archived

2.2.3 Transport protocols

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

HTTP

HTTP is the default 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. This 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 may 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 main 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).

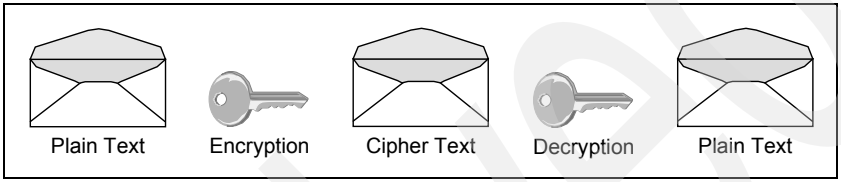


Figure 2-2 Secret key cryptography

- 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.

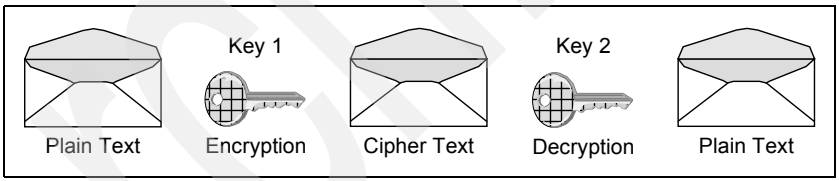


Figure 2-3 Public key cryptography

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 this.

- 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

Archived

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, 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 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

With this technology, there is 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

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
- Multifont 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 the identity of the server. 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. This 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 Java

Java is a programming language based on the concept that software, once written, should run on different kinds of computers, without being rewritten or recompiled. The Java programs are run on Java virtual machines (JVM™). Therefore, a Java program written and tested on one platform runs on another platform's JVM.

2.2.6 Extensible Markup Language

XML has gone from the latest buzzword to an entrenched technology in record time. Today, many businesses are using 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 *metadat*a. The tag language has been developed from older markup standards including GML and 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 fast-growing. It is being adapted to, and supported by, an increasing number of products.

2.2.7 Web services

Web services are self-contained, self-describing, modular applications that can be published, located, and invoked over a network. 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.

Archived

Archived

3

Introduction to WebSphere Partner Gateway

The previous two chapters cover a more general discussion about B2B technologies, B2B models, and standards. This chapter considers the B2B product in the IBM WebSphere Business Integration family of products, WebSphere Partner Gateway. It describes its main features and architecture.

© Copyright IBM Corp. 2005. All rights reserved.

3.1 Editions of WebSphere Partner Gateway

WebSphere Partner Gateway enables B2B process integration and data sharing between partners of all types and sizes. It is implemented on top of a Java 2 Platform, Enterprise Edition (J2EE), and designed for multitier and single-server implementations.

WebSphere Partner Gateway is provided in three editions:

- WebSphere Partner Gateway Express

  This edition is specifically designed for the small-and-medium business (SMB) market. It has a small footprint and is easy to use, but it has limited features and deployment options.

- WebSphere Partner Gateway Advanced

  This edition has all the features of WebSphere Partner Gateway. It has a rich set of features and can handle any complexity in your B2B environment. Many more protocols and standards are supported for both inbound and outbound communication. Also, WebSphere Partner Gateway Advanced has a document engine that can handle the transformation and validation of documents. It also checks for duplicate documents.

- WebSphere Partner Gateway Enterprise

  This edition has the same features of WebSphere Partner Gateway Advanced. Its main differentiator is that a user of WebSphere Partner Gateway Enterprise is licensed to implement more connections than WebSphere Partner Gateway Advanced.

This redbook discusses the use of version 6.0 of WebSphere Partner Gateway Express on Windows, version 6.0 of WebSphere Partner Gateway Advanced on AIX, and version 6.0 of WebSphere Partner Gateway Enterprise on Windows.

3.2 Architecture of WebSphere Partner Gateway

This section introduces the main runtime components of WebSphere Partner Gateway. It discusses the configuration process and profile management of WebSphere Partner Gateway.

Archived

3.2.1 Runtime components

The runtime environment of WebSphere Partner Gateway consists of the following components:

- Receiver, Document Manager, Community Console

  These are application servers built on top of imbedded WebSphere Application Server Version 6.0

- Queue manager and publish/subscribe broker

  These are used to pass around messages and events within the system components and to the outside world.

- Database

  This is used to store profile information and state information about ongoing processes in WebSphere Partner Gateway.

- Common storage

  This is usually a large stand-alone storage device where documents are stored during the processing in WebSphere Partner Gateway and after the processing finishes.

Figure 3-1 on page 34 shows the high-level view of these components. What is not mentioned in this list or in the figure is the graphical tool called Data Interchange Services (DIS). There is more on this later in the chapter and also in Chapter 14, “Native mapping support in WebSphere Partner Gateway” on page 451.

Archived

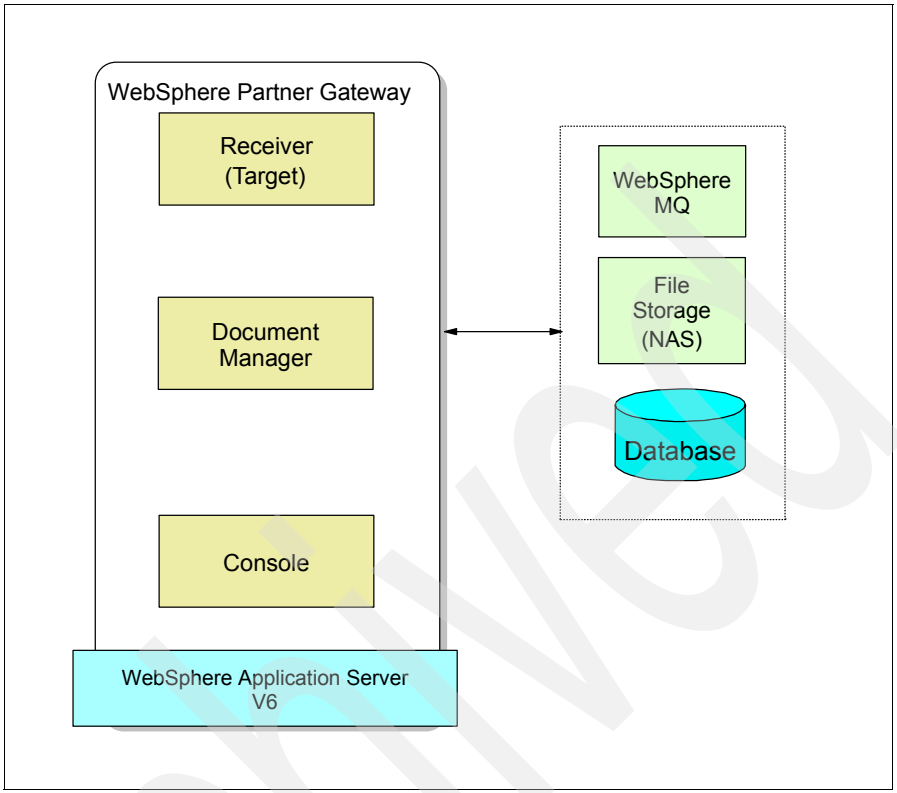


Figure 3-1 Component high-level view

Receiver

The Receiver component is responsible for receiving documents into the WebSphere Partner Gateway server. This consists of receiving documents from internal applications and from external trading partners. Within the Receiver component, several listeners run that poll targets for documents that need to be processed. Such a target can be a file system directory, a JMS queue, or an HTTP URL, for example.

When a document is received by the receiver, it is copied into the common storage. From here, the Document Manager component picks it up and performs its part of the work. The Document Manager is informed about any new documents by polling the router_in or sync_in directories. Except for retrieving configuration information, the Receiver does not interact with the database.

Document Manager

The Document Manager is the real workhorse of the system. When it receives a notification from the Receiver about a new file, Document Manager performs several functions that can be grouped into the following categories:

- Document handling or parsing and validation

  This step consists of determining the type of document, validating the syntax, retrieving business identifiers if applicable, and possibly invoking activities such as transformation maps.

- Document encryption and signing, if requested
- Packaging

  This involves wrapping the document in a header, trailer, or both. Example packages include AS2 and RosettaNet Implementation Framework (RNIF).

- Transmission to trading partners

  After retrieving the participant connection that matches with document type and business identifiers, the Document Manager sends the document to the trading partner. It handles retries and time-outs as well.

- State management

  Every change of state is recorded in the database by the Document Manager, so that it can recover in case of failures.

Community Console

The Community Console component is a Web application that allows you to perform configuration work, such as profile management. It also enables you to to manage the working system. You can use it to help determine which transactions are in progress and which transactions have failed. The Community Console retrieves most of this information from the database. When using a document viewer in the Console, the Console application retrieves the actual document from the common storage area.

Queue manager and publish/subscribe broker

A queue manager has to be created to support communication between the three main runtime components of the product. It is used to pass information about received documents and information about system events for external event and system management.

The publish/subscribe broker is used to publish changes to the configuration to the running systems. For example, if a profile is changed, WebSphere Partner Gateway publishes a reload configuration message to the subscribing runtime components, such as the Receiver and the Document Manager.

Archived

Database

The database is used to store data that can be classified in two broad categories: profile information and state management information. The *profile information* is largely a read-only category after the system is configured. Changes occur only when new profiles are added or when an existing profile is updated, for example replacing an expired certificate. *State management information* is a highly active category of information. A single document exchange results in many logging events that contain the state transitions of a document.

The actual document, which can be very large, is not stored in the database. It is always kept in the common storage.

Common storage

The common storage component can be a specific directory for a single server installation or a network file storage (NFS) unit that is attached to all servers that run WebSphere Partner Gateway components. Documents move in this common storage when their state changes. Also, each document that has ever been processed by WebSphere Partner Gateway is stored in this common storage area. This allows you to analyze document volumes and retrieve historical information for auditing purposes. Archiving mechanisms exist so that you can offload documents to long-term storage units that are not necessarily connected to the WebSphere Partner Gateway server that is running.

Figure 3-2 on page 37 shows the high-level flow through the WebSphere Partner Gateway runtime components.

Archived

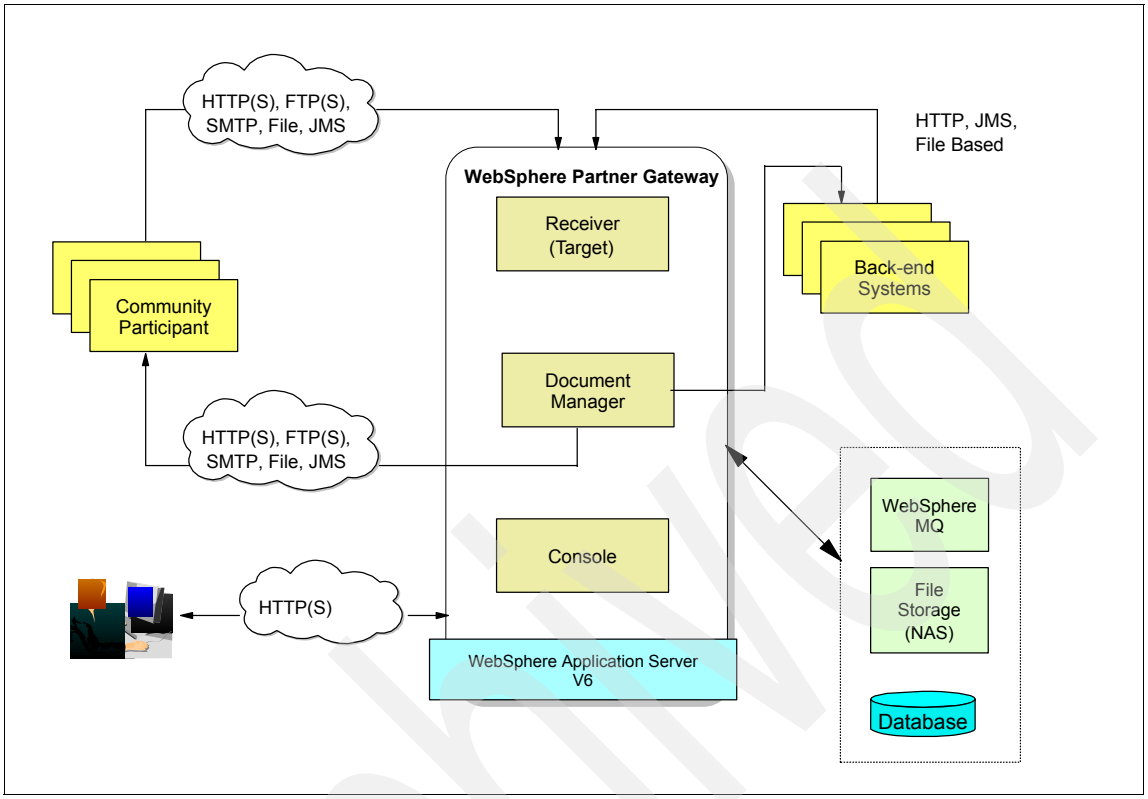


Figure 3-2 High-level document flow through the components

3.2.2 Configuration components

The configuration components can be divided into three categories:

- Targets: The entry-points into WebSphere Partner Gateway
- Gateways: The exit-points out of WebSphere Partner Gateway
- Interactions: The processing between target and gateway

Targets

A *target* is a listening or polling component. For example, you can poll a folder on the file system, a JMS queue, or an HTTP URL for documents coming from business partners or from internal systems. A target is used for both inbound and outbound flows. A target always refers to something that WebSphere Partner Gateway manages and that other resources have to use. A file system target points to a folder where, for example, WebSphere Process Server or WebSphere Message Broker has to deliver an outbound document. An HTTP target points to a URL to which the trading partner has to post their documents.

Gateways

A *gateway* is a resource that is used by WebSphere Partner Gateway to deliver documents to an existing resource. For example, when sending a document to your partner, your own WebSphere Partner Gateway server needs a gateway that points to the URL that is configured on your partner's server. Thus, a gateway can point to a remote target. That is a target on the WebSphere Partner Gateway server of your partner. A gateway can also point to a resource in the internal network, such as a JMS queue that acts as the input queue of WebSphere Data Interchange. Again, such a JMS queue is a resource that exists already, independent of the WebSphere Partner Gateway server.

Interactions

An *interaction* is anything that happens between the target and the gateway. At the target, WebSphere Partner Gateway might have received a document using a certain transport, using a certain protocol and a certain structure. However, the gateway might need a different transport and protocol. An interaction performs the required steps to remove transport headers, unpackage the document, repackage the document, and add different protocol headers.

To illustrate this process, see Figure 3-3 on page 39. Starting from the right, a document might have been received on a file system target. Because it is retrieved from the file system, it is typically not packaged. The document payload is electronic data interchange (EDI), which is validated by WebSphere Partner Gateway. The outbound gateway is an HTTP gateway along which we use the protocol EDIINT AS1/AS2. Thus, WebSphere Partner Gateway needs to package the EDI document in AS headers. And because the gateway is an HTTP gateway, HTTP headers need to be added as well. This sequence of actions is called an *interaction*.

From left to right, we have a similar flow. A document is received on the inbound target, which is HTTP. Thus, it has an HTTP header that needs to be removed. The payload of the HTTP stream has an RNIF structure within which we have a RosettaNet Service Content (RNSC) document. The inbound gateway is a JMS gateway. That means that JMS headers are required, and possibly back-end integration packaging, so that the internal application has sufficient metadata to process the RSNC document.

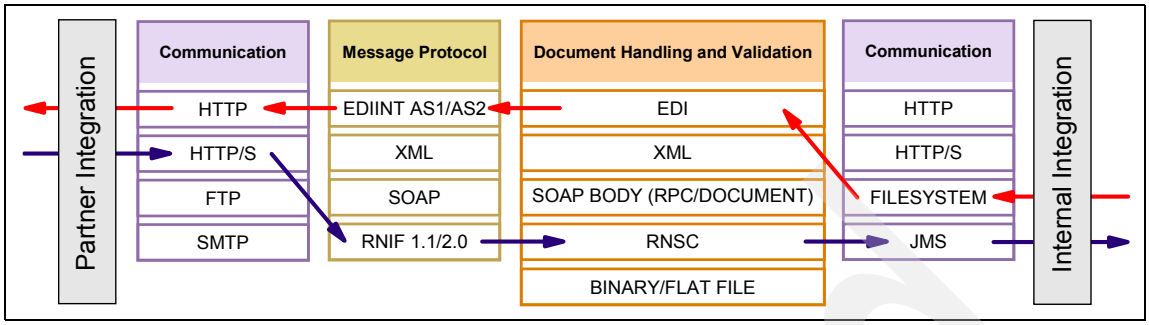


Figure 3-3 Going from target to gateway

3.2.3 Profile management

Three different types of profiles exist within WebSphere Partner Gateway. The first profile is a built-in profile. It is called *Operator*, but you can change this name, if you want. Members of the profile (or company) Operator have the widest authorities in the system. Usually, the members or users belonging to the company Operator are the system administrators who are responsible for a smooth operation of the server.

A second profile is the *community manager*. This profile is usually reserved for the company that owns the server. Members or users belonging to this profile have a wide range of authorities but are usually not authorized to make system-wide changes. There can only be one community manager for a given WebSphere Partner Gateway server. However, in a network of interconnected WebSphere Partner Gateway servers, you have more than one community manager. Usually, every WebSphere Partner Gateway has exactly one community manager.

The trading partners of the community manager are defined to the WebSphere Partner Gateway server as *community participants*. As a community participant, you have limited options on the server. For example, you can only use or view objects and documents that relate to you as a community participant.

In a network of interconnected WebSphere Partner Gateway servers, you may be the community manager of your own server and a community Participant on the server of your trading partners. If implemented, you can log on as a community Participant to the server of your trading partner. This allows for some level of self-definition and self-management within your trading partner community.

3.3 B2B features and standards

As seen in the previous chapters, many standards are playing a role in a B2B solution. These standards and technologies operate at three different levels:

- Transport: How the payload goes from the sender to the receiver
- Message protocol: The language of the payload
- Business protocol: The meaning of the terms used in the message

Figure 3-4 shows this stack of technologies, above which we list the features of WebSphere Partner Gateway.

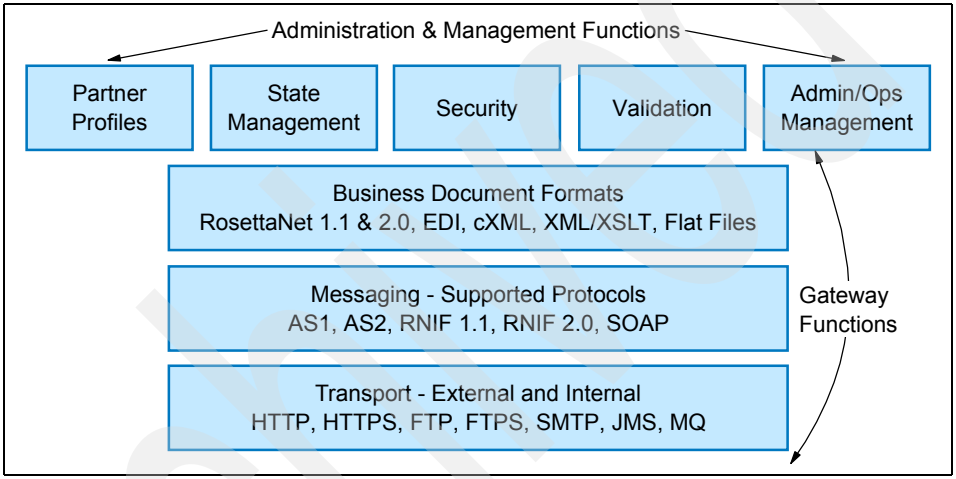


Figure 3-4 Stack of B2B formats and standards

3.3.1 Transport options

WebSphere Partner Gateway supports several transport options. Messages can be sent to your partners or received from your partners with:

- HTTP and HTTPS
- FTP(S) and FTP scripting
- SMTP(for AS1)
- JMS and MQ

Direct access to the file system is possible as well for internal transport. It is possible to implement custom transport options.

3.3.2 Messaging protocol options

Three styles of messaging are supported by WebSphere Partner Gateway. You can use:

- AS1 and AS2
- RNIF 1.1 and RNIF 2.0
- SOAP, to support Web services

It is possible to implement custom messaging protocol options.

3.3.3 Business document formats

At the business level, WebSphere Partner Gateway can handle and validate documents that adhere to the following standards:

- RosettaNet 1.1 and RosettaNet 2.0
- EDI X12 and EDIFACT
- cXML
- Custom XML
- Flat files with or without any formalized structure.

Note: For more information and examples of RosettaNet processing, see:

B2B Solutions using WebSphere BI Connect Version 4.2.2, SG24-6355-01

3.3.4 Security options in WebSphere Partner Gateway

B2B transactions over the Internet imply that strong security is required. WebSphere Partner Gateway has security features at three different levels:

1. Transport level

   WebSphere Partner Gateway supports HTTP/S and FTP over SSL, so that the exchange of documents occurs in a secure way with authorized partners.

2. Document level

   Besides using a secure transport, you can also protect the actual document by encrypting and signing it. WebSphere Partner Gateway supports using several encryption and signing algorithms. It also has an interface to manage certificates and private keys.

3. Interface level

   Using the console is reserved for authenticated users. You need to log on to WebSphere Partner Gateway with a browser. The browser interface is supported over HTTP and HTTP/S, providing a secure interface to WebSphere Partner Gateway. As an authenticated user, you belong to a

Archived

certain group or profile which implies certain authorities. The hub administrator can assign or deny certain authorities so that a user can only perform authorized tasks.

3.4 Extensible architecture

WebSphere Partner Gateway provides a mechanism for users to add custom functions, called exits or handlers, at different stages of the document processing. WebSphere Partner Gateway uses this support to expand its own capabilities, like adding of the new EDI functionality.

3.4.1 Encryption, validation and transformation

As WebSphere Partner Gateway receives and processes a document, users have the ability to call-out to a program easily as part of processing. These User Exits or handlers can be used (for document encryption and decryption for a protocol that does not have it, for example), validation (think plug-in EDIFACT for SNIP level validation with HIPAA), or transformation.

3.4.2 Custom transports, packages and protocols

As mentioned earlier, WebSphere Partner Gateway can be extended to enable custom transports, packages and protocols to be added within the framework of the WebSphere Partner Gateway runtime and Community Console based administration. There are a number of exit points that enable adding custom listeners for additional transports, parsers for additional packages and protocols.

In this way' the rest of the infrastructure of WebSphere Partner Gateway can be used in conjunction with these new capabilities.

For example, a new transport could deliver a document, such as an EDI X12 that would be reprocessed in the normal way through the rest of the system. Also, AS2 might be used to deliver a new protocol as an AS2 binary document. So the combination of standard and custom components can be used to extend and reuse all of the capabilities of WebSphere Partner Gateway in a seamless way.

Archived

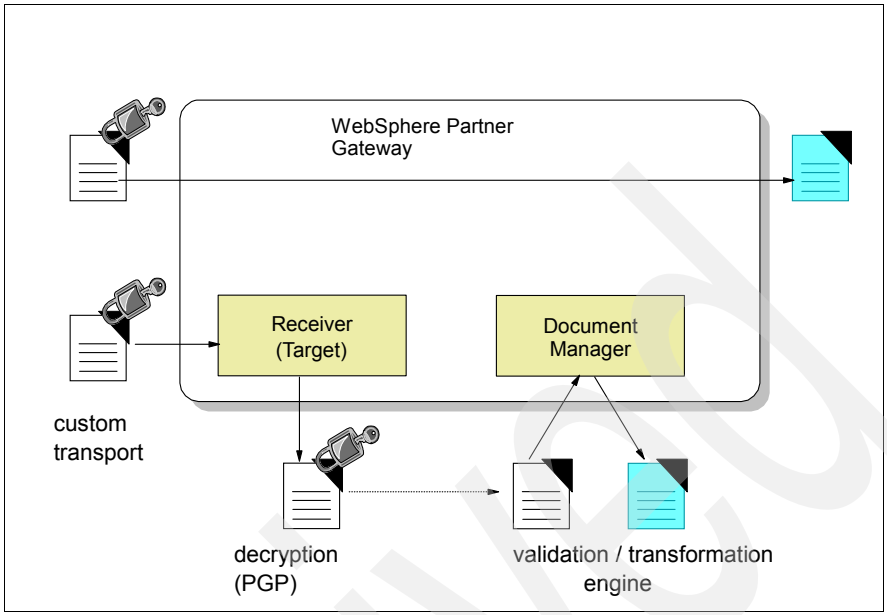


Figure 3-5 Exit example

Note: The exits functionality is carried forward from WebSphere Business Integration Connect V4.2. For more information and samples see:

B2B Solutions using WebSphere BI Connect Version 4.2.2, SG24-6355-01

3.5 Enterprise Integration

WebSphere Partner Gateway does not operate on an island within an organization. Documents that are received by WebSphere Partner Gateway usually need to be processed by one or more internal applications. WebSphere Partner Gateway can interact directly with those internal applications using JMS, HTTP, or the file system (refer to Figure 3-2 on page 37).

For more complex scenarios, it is common for WebSphere Partner Gateway to interact with an EAI layer, for example, with an integration broker such as WebSphere Message Broker. The integration broker can then handle the distribution of incoming documents to one or more internal applications (possibly using WebSphere Integration Adapters). If an incoming document needs to be delivered to only one application, WebSphere Partner Gateway could interact with that application directly.

However, if a document needs to be delivered to more than one internal application in a dynamic way (based on the content or type of the document for example), the use of an integration broker will greatly simplify the overall infrastructure. Integration between WebSphere Partner Gateway and the integration broker can be performed with several communication options. This allows you to exchange detailed metadata between WebSphere Partner Gateway and an integration broker so that you can build flexible solutions.

Archived

Part 2

Building a B2B exchange

Archived

© Copyright IBM Corp. 2005. All rights reserved.

Archived

4

Implementation scenarios

This chapter describes the different steps to implement a B2B solution between four business partners. By adding features and functions in a gradual way, it becomes easier to understand why configuration tasks are done. It also offers the one who implements it, a check-point to perform some validation before moving to the next step.

This chapter presents a general discussion about the implementation process, and refers you to subsequent chapters for details.

© Copyright IBM Corp. 2005. All rights reserved.

4.1 Implementing WebSphere Partner Gateway Enterprise for Windows

The first implementation phase is setting up WebSphere Partner Gateway Enterprise for Windows in a dual server environment. The three runtime components of WebSphere Partner Gateway are running on a single machine that has the host name wpgv6hub. The DB2® server and the MQ queue manager are running on a separate machine, with the host name wpgv6db.

Chapter 5, “WebSphere Partner Gateway Enterprise on Windows” on page 59, discusses details about the planning and decisions that you need to make to build such an environment successfully.

Figure 4-1 shows an overview of this setup. At the end of this phase we have a working WebSphere Partner Gateway server, but with no connected partners and no connections with internal systems.

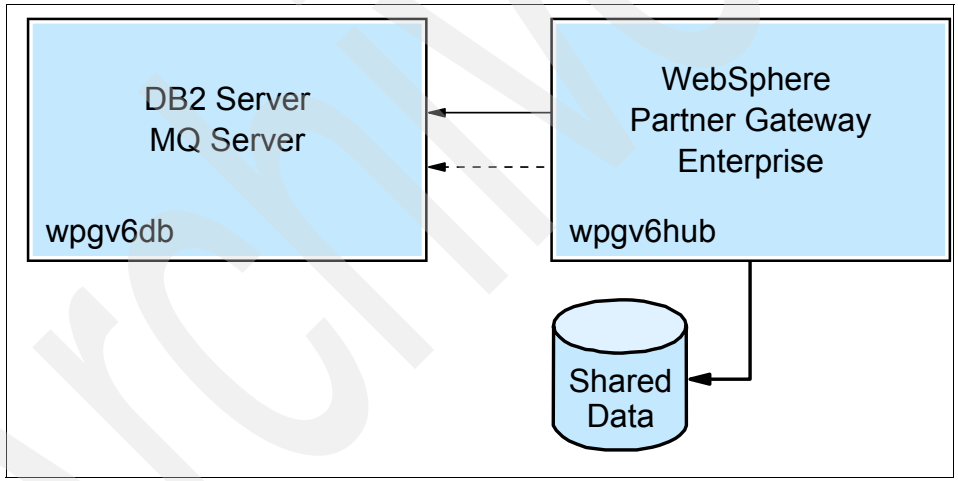


Figure 4-1 WebSphere Partner Gateway Enterprise in a two server environment

4.2 Implementing WebSphere Partner Gateway Advanced for AIX

The next phase is the implementation of WebSphere Partner Gateway Advanced on a single AIX machine. It involves using a local DB2 server and a local MQ queue manager.

Chapter 6, “WebSphere Partner Gateway Advanced for AIX” on page 109, describes the setup of all prerequisite components. It also provides tips about how to validate the setup of DB2 or how to verify that prerequisite components are configured correctly. Then, at the end of that chapter we have a working WebSphere Partner Gateway server that is not connected to any other partner or system, as shown in Figure 4-2.

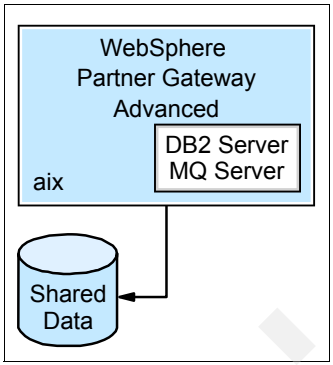


Figure 4-2 WebSphere Partner Gateway Advanced on a single AIX machine

4.3 Implementing a basic exchange

Given the two WebSphere Partner Gateway implementations on AIX and Windows, we learn how to set up a basic exchange between those two WebSphere Partner Gateway servers. Chapter 7, “Creating a basic B2B exchange” on page 137, explains how to configure WebSphere Partner Gateway so that electronic data interchange (EDI) documents can be exchanged in both the directions over HTTP in an AS2 package.

Figure 4-3 on page 50 illustrates the configuration steps to build the environment. A natural approach is to split this into a discussion about the outbound flow and a discussion about the inbound flow and possibly a validation for the outbound flow before proceeding to build the inbound flow. However, this book describes the setup by focusing more on the role-based approach.

In Chapter 7, “Creating a basic B2B exchange” on page 137, we limit the communication to one type of document and to one partner. We also limit ourselves to file-based input and output. That is, we configure WebSphere Partner Gateway to retrieve EDI documents from a directory and send it to the trading partner that is encoded in the EDI document. The target WebSphere Partner Gateway server drops the EDI document into a directory as well.

Archived

Thus, the MQ infrastructure at either side is not used much at this stage, except for communication between WebSphere Partner Gateway components. At the end of this chapter, we will see how to exchange documents in both directions and receive message disposition notifications (MDNs) from the target server.

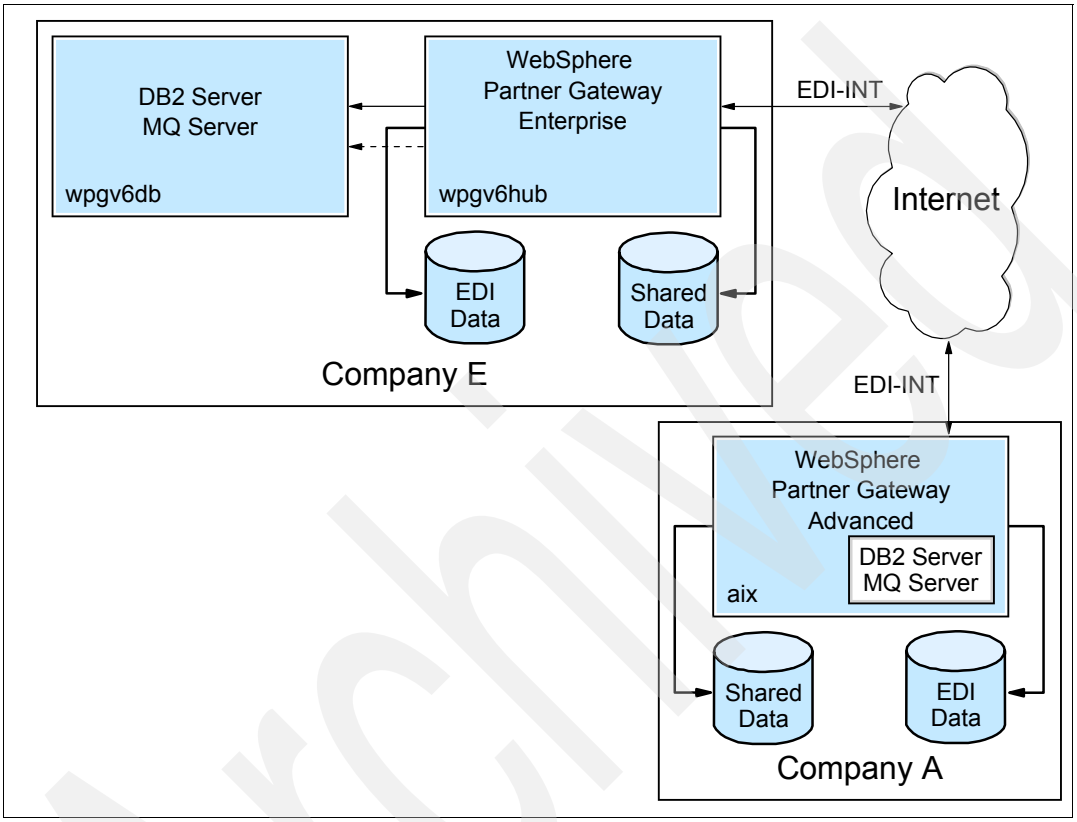


Figure 4-3 Connecting both servers with AS2/HTTP

4.4 Securing the exchange between Company E and Company A

In Chapter 7, “Creating a basic B2B exchange” on page 137, we assume default values for AS2 connection. This means that no encryption or digital signatures are used. The EDI document is sent in clear text. This is not good enough for a real- world implementation.

Chapter 8, “Securing the B2B exchange” on page 207, explains how to secure the exchange of the EDI documents. It demonstrates how to obtain keys and certificates and how to upload them to WebSphere Partner Gateway. This is done to switch to an AS2 connection where sensitive business data is encrypted and where EDI documents are signed, so that the receiver can validate the origin.

Chapter 8, “Securing the B2B exchange” on page 207, describes the process again from a role-based perspective, so that it becomes clear which tasks can be performed by either the hub administrator, the WebSphere Partner Gateway server, or by the community Participant (Figure 4-4).

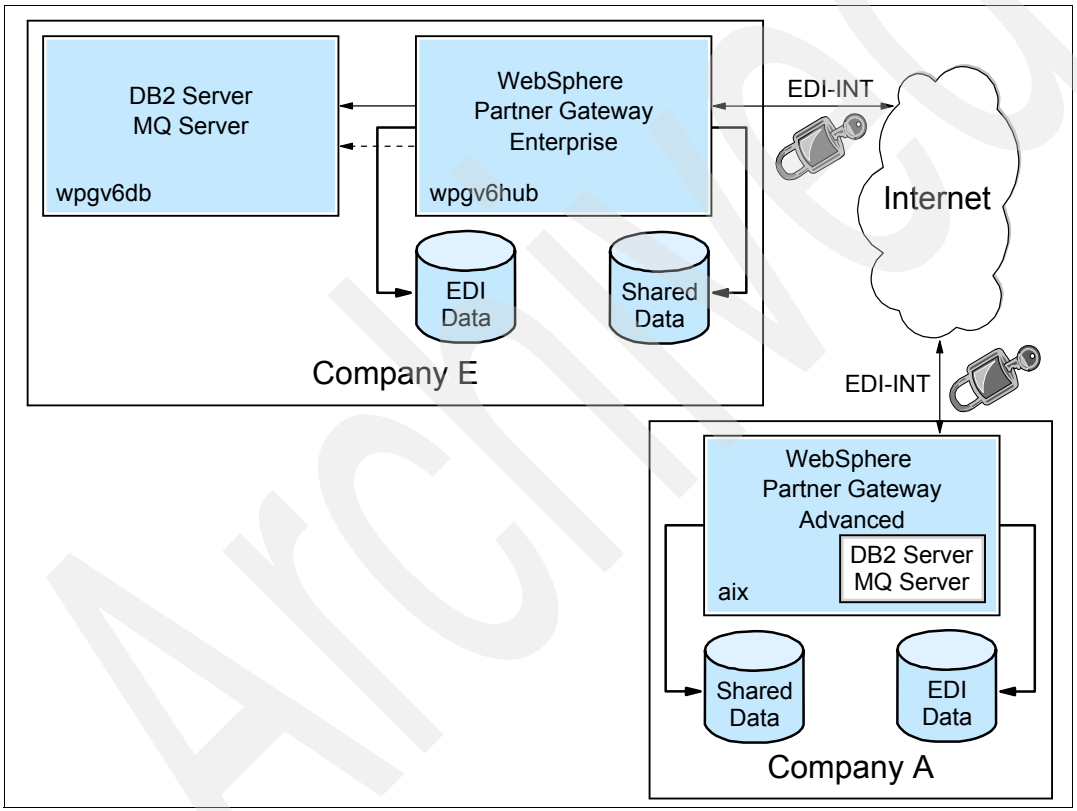


Figure 4-4 Adding encryption and digital signature to the exchange

Archived

4.5 Implementing WebSphere Partner Gateway Express for Windows

After completing the security setup, we extend the exchange with an additional partner (Figure 4-5).This is again performed in two steps. First, we implement WebSphere Partner Gateway Express for windows. Given that WebSphere Partner Gateway Express has fewer requirements for other software products, such as DB2 and WebSphere MQ, this is a straightforward process, as discussed in Chapter 9, “WebSphere Partner Gateway Express on Windows” on page 261.

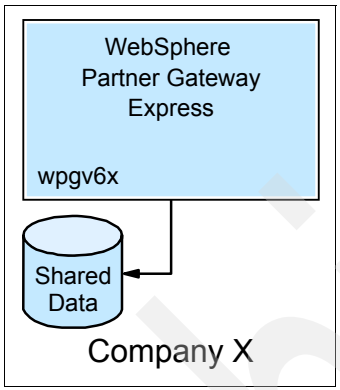


Figure 4-5 Implementing WebSphere Partner Gateway Express

Chapter 10, “Extending the B2B exchange” on page 277, explains how to add another partner to an existing WebSphere Partner Gateway server. What changes need to be performed to get another partner online? We see how to add a third community Participant and another document type (Figure 4-6 on page 53).

The WebSphere Partner Gateway Express partner is less sophisticated than the WebSphere Partner Gateway Advanced partner and wants to exchange simple custom XML documents. Thus, we introduce this new custom XML format to the WebSphere Partner Gateway Enterprise server of Company E so that it can perform the required routing to Company X, who owns the WebSphere Partner Gateway Express server. We also contrast the routing logic for WebSphere Partner Gateway Enterprise (and Advanced) versus the routing logic for WebSphere Partner Gateway Express.

Archived

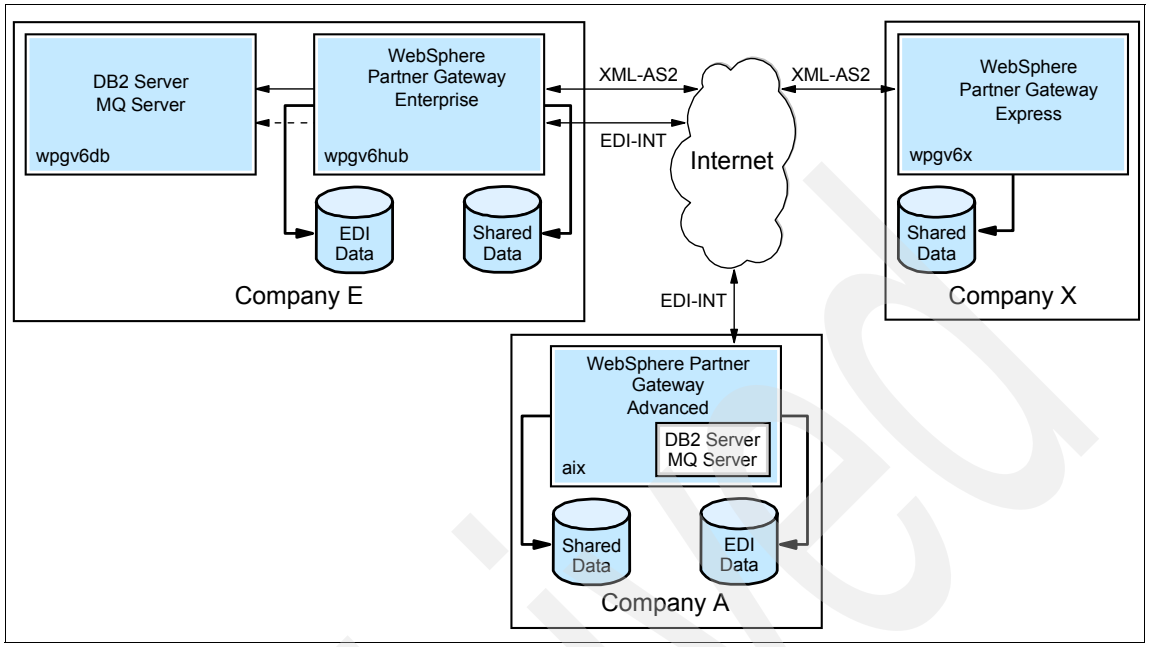


Figure 4-6 Adding a new partner and a new document type

4.6 Using FTP with WebSphere Partner Gateway

The B2B exchange is extended even further, so that it can also support FTP. Company A wants to extend its B2B infrastructure to include a document exchange with Company F. However, Company F is reluctant to jump quickly on a technology that is new to the IT department of Company F. At the same time, Company A wants to build a solution sooner rather than later. A compromise is found between Company A and Company F by implementing an FTP-based solution. Company F has extensive experience with FTP and thus they can implement a document exchange with Company A rather quickly.

Figure 4-7 on page 54 shows a schematic overview of the B2B exchange. It shows the existing AS2-based exchanges of Company E with Company A (for EDI) and Company E with Company X (for XML). Within this infrastructure, we add an exchange of Company A with Company F over FTP, as explained in Chapter 12, “Integrating FTP servers” on page 335.

This version of WebSphere Partner Gateway also has a solution to offer for companies such as Company F and other companies like Company V, introduced in section 4.7, “Using FTP Scripting to enable Value Added Network (VAN) connectivity” on page 54, which is the FTP Scripting based solution.

The advantage of the FTP Scripting based solution is that now for Company A to receive files from Company F, it need not have an FTP server at its end. With the FTP Scripting solution, Company A can directly pull files from his account, on the Company F's FTP server.

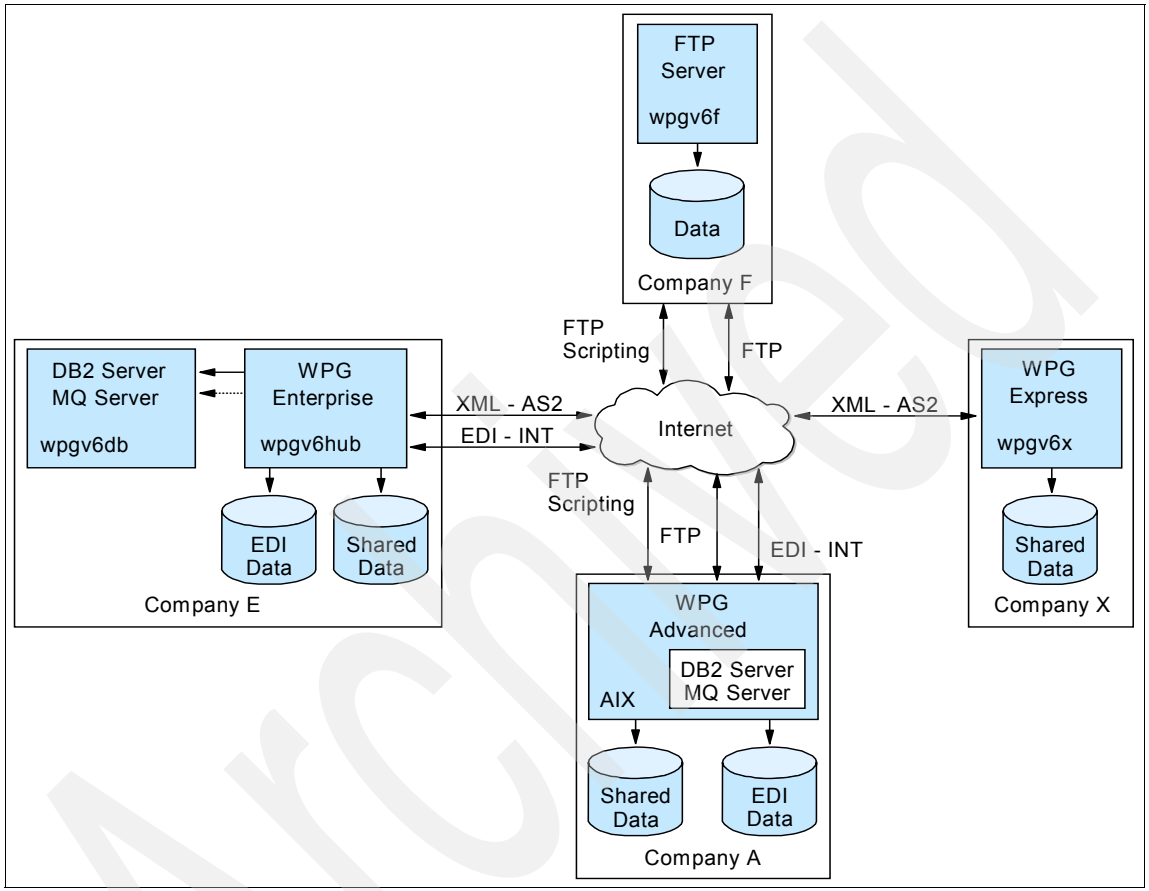


Figure 4-7 Integrating an FTP server with WebSphere Partner Gateway

4.7 Using FTP Scripting to enable Value Added Network (VAN) connectivity

The B2B exchange is further extended to enable connectivity to a VAN. Company E wants to extend its B2B infrastructure to include a document exchange with Company V. Company V, like Company F, is reluctant to jump

quickly to a new technology that is new to the IT department of Company V. Company E and Company V work out a FTP Scripting based solution.

Figure 4-8 shows a schematic overview of the complete B2B scenario. We extend this infrastructure to add an exchange with Company V as explained in Chapter 13, “Enabling VAN connectivity using FTP Scripting” on page 419.

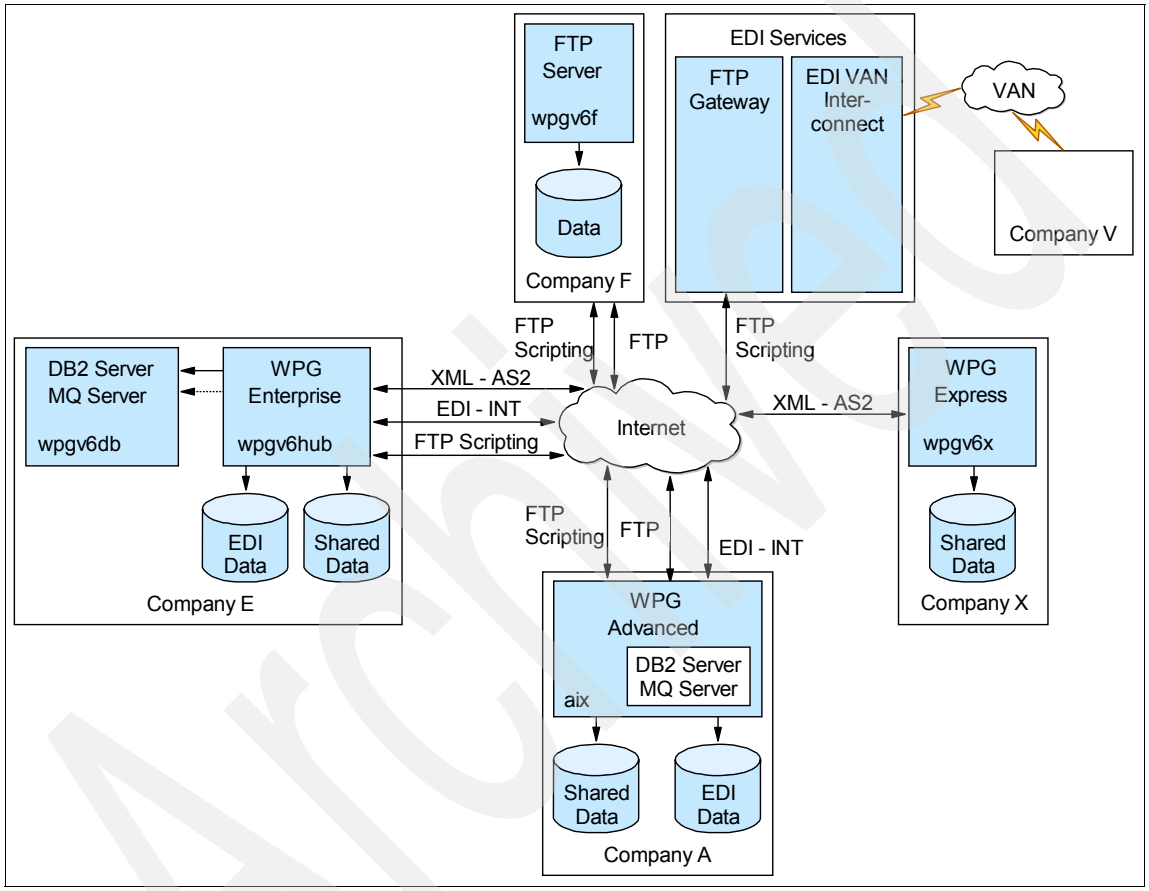


Figure 4-8 Enabling connectivity with VAN

4.8 Managing the B2B infrastructure

Because the infrastructure is now a bit more complex, we can take a closer look at the manageability of the infrastructure. As usual, the change from 1 to 2 introduces another level of complexity, and when this level is understood, moving from 2 to *n* becomes easy.

Each player in the system has an important role. The hub administrator needs to determine if the system as a whole performs well, for example, by ensuring that there are no bottlenecks or errors. Several tools and options are available to the hub administrator.

The community manager wants to look at the WebSphere Partner Gateway system from a business perspective. This person asks, “What documents have I received? From whom? What is the status of each document? Did I receive an MDN for that transaction?” For all of these questions, the WebSphere Partner Gateway server contains tools and viewers so that the community manager has a good view on their business.

In addition to management options for the hub administrator and the community managers, WebSphere Partner Gateway also has tools and viewers available for the community participant. Thus, if the community Participant receives no MDN or receives an error report, they can log on to the WebSphere Partner Gateway of his partner and try to determine why their transaction resulted in an error.

Chapter 11, “Managing the B2B exchange” on page 307, explains all of this in greater detail.

4.9 EDI translation

WebSphere Partner Gateway now has the built-in ability to support electronic data interchange (EDI) translation, which was facilitated in the earlier version of the product by having to integrate with the WebSphere Data Interchange product.

In addition WebSphere Partner Gateway offers abilities to, perform EDI packaging, transformation, validation, enveloping, de-enveloping, batching and splitting.

WebSphere Partner Gateway also ships with the Data Interchange Services (DIS) Client which is used to create data transformation maps, where in the source and target data types can be either an EDI, XML or a Record Oriented Data (ROD). The DIS client is discussed in detail in Chapter 14, “Native mapping support in WebSphere Partner Gateway” on page 451.

Chapter 18, “Mapping for EDI transformation and validation” on page 589 gives a detailed understanding on how to create the data transformation maps and the validation maps.

EDI Processing capabilities would include enveloping and de-enveloping which are discussed in Chapter 19, “Enveloping and de-enveloping” on page 621. WebSphere Partner Gateway also supports EDI acknowledgments like TA1 or

other functional acknowledgments. This is discussed in Chapter 20, “Functional acknowledgements” on page 695.

Batching and splitting, as explained in Chapter 21, “Batching and EDI splitting” on page 729, is the ability of WebSphere Partner Gateway to support large documents with multiple EDI, ROD or XML messages.

Figure 4-9 encapsulates the EDI support offered by WebSphere Partner Gateway.

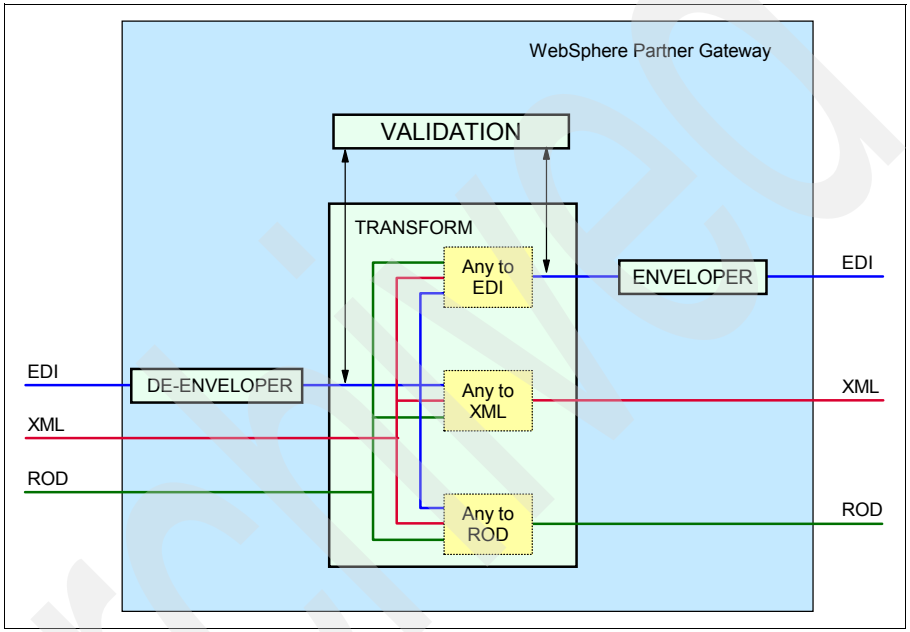


Figure 4-9 EDI support in WebSphere Partner Gateway

Archived

5

WebSphere Partner Gateway Enterprise on Windows

This chapter explains how to implement WebSphere Partner Gateway in a Windows environment. If you are mostly interested in an AIX implementation, go to Chapter 6, “WebSphere Partner Gateway Advanced for AIX” on page 109, which contains similar information as in this chapter, but that applies to the AIX environment.

© Copyright IBM Corp. 2005. All rights reserved.

5.1 Implementation overview

The three server components of WebSphere Partner Gateway will be installed on the same Windows Server (with the hostname WPGV6HUB).

The prerequisite supporting software packages for WebSphere Partner Gateway, DB2 and WebSphere MQ are installed on a separate dedicated server (with the hostname WPGV6DB).

In this WebSphere Partner Gateway installation, we use an existing installation of WebSphere Application Server V6, rather than the embedded WebSphere Application Server which ships with the product.

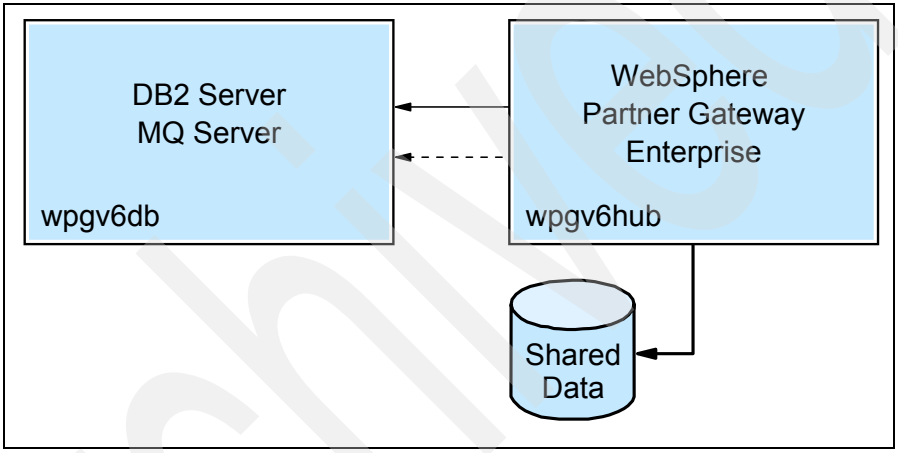


Figure 5-1 System architecture for Company E

5.2 Verify software levels on the hub and data servers

As for any implementation of a software product, you need to understand the prerequisites, both software and hardware, and you need to know how to verify that prerequisites are met.

WebSphere Partner Gateway relies on the services and features of two other products:

- A database manager, such as DB2, which is used in this environment
- WebSphere MQ

The following instructions help you to validate that these products are installed at the correct level and that required features are available.

5.2.1 Verifying WebSphere MQ 6.0

To verify the version of WebSphere MQ, you can run the command **dspmqver**.

> **Note:** In WebSphere MQ V6, this replaces the old mqver command as can be seen below.

Example 5-1 shows our installed version as Version 6.0.

Example 5-1 Output from dspmqver command

```
C:\>mqver
mqver has been replaced by the dspmqver command.
See the WebSphere MQ System Administration documentation for details.

C:\>dspmqver
Name:        WebSphere MQ
Version:     6.0.0.0
CMVC level:  p000-L050519
BuildType:   IKAP - (Production)
```

WebSphere Partner Gateway also uses Java Message Service (JMS). With the latest version of WebSphere MQ installation of the Java support is a part of the default installation. We verify whether JMS support is installed, following these steps:

1. Start the registry editor. Select **Start** → **Run**.
2. Type:

   regedt32
3. Expand the registry. Select **HKEY_Local_Machine** → **Software** → **IBM** → **MQSeries®** → **CurrentVersion** → **Components**.

Archived

4. In the right pane (see Figure 5-2), locate the value JavaMsg.

Name	Type	Data
(Default)	REG_SZ	(value not set)
Development Toolkit	REG_SZ	Installed
Explorer	REG_SZ	Installed
FTA_Client	REG_SZ	Installed
FTA_Server	REG_SZ	Installed
JavaMsg	REG_SZ	Installed
Local Clients\Windows NT Client	REG_SZ	Installed
Server	REG_SZ	Installed
XA_Client	REG_SZ	Installed
XA_Java	REG_SZ	Installed

Figure 5-2 Verifying the installed components of WebSphere MQ

5.2.2 Verifying DB2

WebSphere Partner Gateway requires version 8.2 of DB2. To verify which versions of DB2 and FixPack are installed, open a db2 command window and run the program **db2level**. The output of this program tells you which version of DB2 is installed and which FixPack level is installed.

Example 5-2 shows the output of the **db2level** program in our environment. It shows that we use DB2 8.2(DB28.1 with fixpack8).

Example 5-2 Output of db2level command

```
C:\>db2level
DB21085I  Instance "DB2" uses "32" bits and DB2 code release "SQL08021" with
level identifier "03020106".
Informational tokens are "DB2 v8.1.8.806", "special_13153", "WR21348_13153",
and FixPak "8".
```

Important: In earlier releases of DB2, when being used for WebSphere Business Integration Connect, it was necessary to ensure that a C/C++ compiler was available to compile the many stored procedures that are built during installation. The current release of DB2 that is used has an embedded compiler, so this is not necessary.

5.3 Installing the software for the data machine

This section explains how to configure the already installed WebSphere MQ 6.0, and how to install the database required for WebSphere Partner Gateway. This includes:

- Adding users and groups
- Configuring WebSphere MQ
- Installing the database loader and creating the database

At the end of this section, we perform some simple validation to make sure that the database is created and loaded correctly before installing the WebSphere Partner Gateway on the hub machine.

5.3.1 Adding user IDs and a group

WebSphere Partner Gateway uses four different user IDs and one group to manage security roles. The names of these user IDs and groups you are free to choose. However, we recommend that you use default names. On the database server, only three users are needed. The user bcguser is needed on the WebSphere Partner Gateway server itself.

> **Note:** The server where the user IDs need to be defined, depends on the setup of the Windows security domain and the type of database authentication that is implemented. In our environment, we work in a Windows workgroup with locally defined user IDs and not in a Windows domain where user IDs are defined centrally. Also, the database authentication is server-side.

To define the user IDs, follow these steps:

1. Right-click **My Computer** and select **Manage**.
2. Expand the tree structure in the left pane and select **System Tools** → **Local Users and Groups** → **Users**.
3. Right-click the folder **Users** and select **New User**.

Archived

4. In the New User window (see Figure 5-3), complete these items:
   a. Name the user `bcgcon`.
   b. Provide a password.
   c. Deselect the **User must change password at next logon** option.
   d. Select the **Password never expires** option.
   e. Click **Create**.
   f. Repeat the same steps to add users `bcgdoc` and `bcgrecv`.

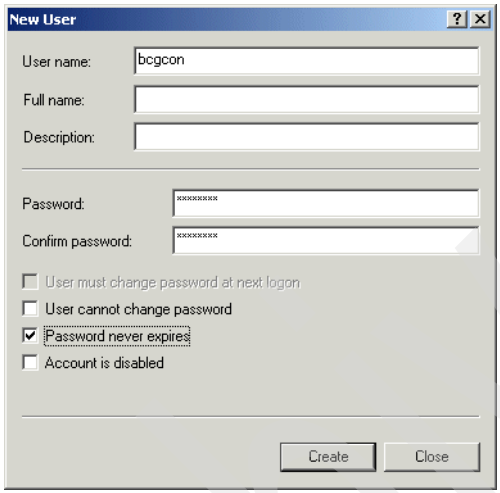


Figure 5-3 Adding a new user

5. Expand the tree structure in the left pane and select **System Tools** → **Local Users and Groups** → **Groups**.
6. Right-click the folder **Groups** and select **New Group**.

Archived

7. In the New Group window (see Figure 5-4), complete these tasks:
   a. Name the group `bcggroup`.
   b. Click **Add** to add members to this group.
   c. In the Members box, you see the list of defined users on this computer. Select the following users: **bcgcon**, **bcgdoc**, and **bcgrecv**.
   d. Click **Create**.
   e. Click **Close** to return to Computer Management window.

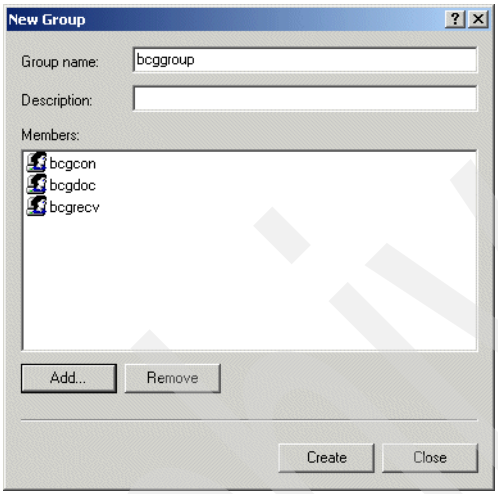


Figure 5-4 Creating a new group and adding member

5.3.2 Configuring WebSphere MQ 6.0

This section describes how to create and configure the WebSphere MQ 6.0 resources for use by WebSphere Partner Gateway.

Start by creating a new queue manager:

1. Start the MQ Explorer. Select **Start** → **Programs** → **IBM WebSphere MQ** → **WebSphere MQ Explorer**.
2. Expand the tree in the left pane and right-click the folders **Queue Managers.**
3. Select **New** → **Queue Manager**.
4. Enter a name for the queue manager. We chose `partner_e.bcg.queue.manager`.
5. Enter a dead letter queue. We chose the `SYSTEM.DEAD.LETTER.QUEUE`.
6. Click **Next**.

Archived

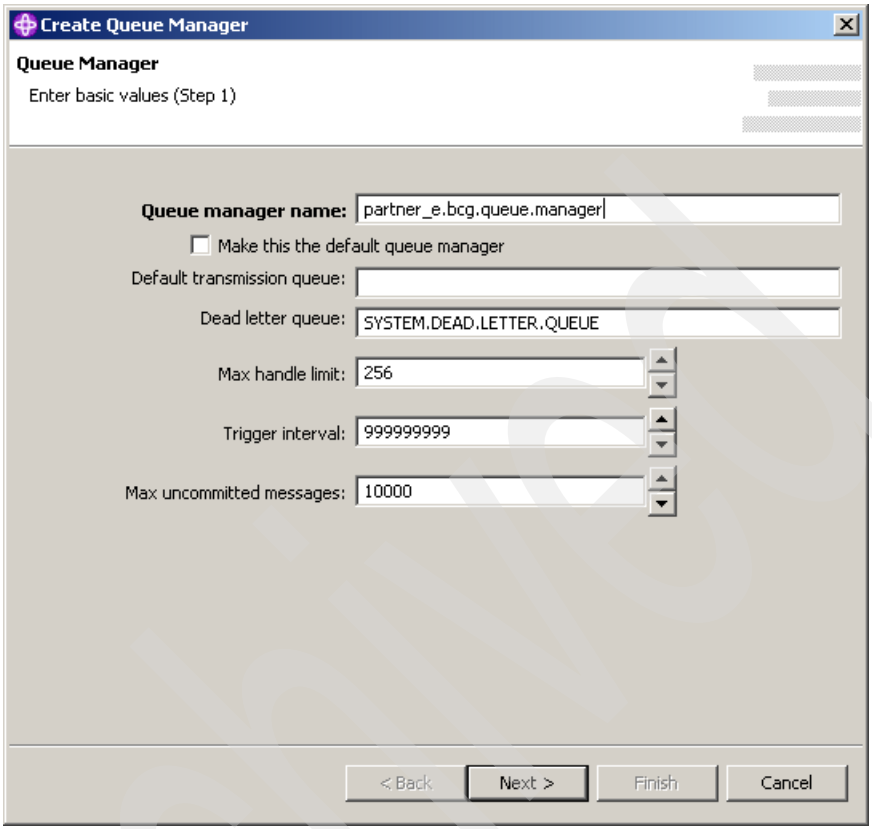


Figure 5-5 Create Queue Manager: Step 1

7. We now need to customize some of the log settings. Two types of logs exist, circular and linear. A *circular log* provides a transactional log, but is easier to manage. A *linear log* is required to make and restore backups within WebSphere MQ and to recover from media failures, assuming that the messages and log are not stored on the same media.

   Circular logging is the default, which is fine for our scenario.

8. Change the number of primary log files to **62**, leave the secondary log files as 2. The total of primary and secondary may not exceed 64.

9. Change the log file size to be 2048 pages of 4KB as shown in Figure 5-6 on page 67.

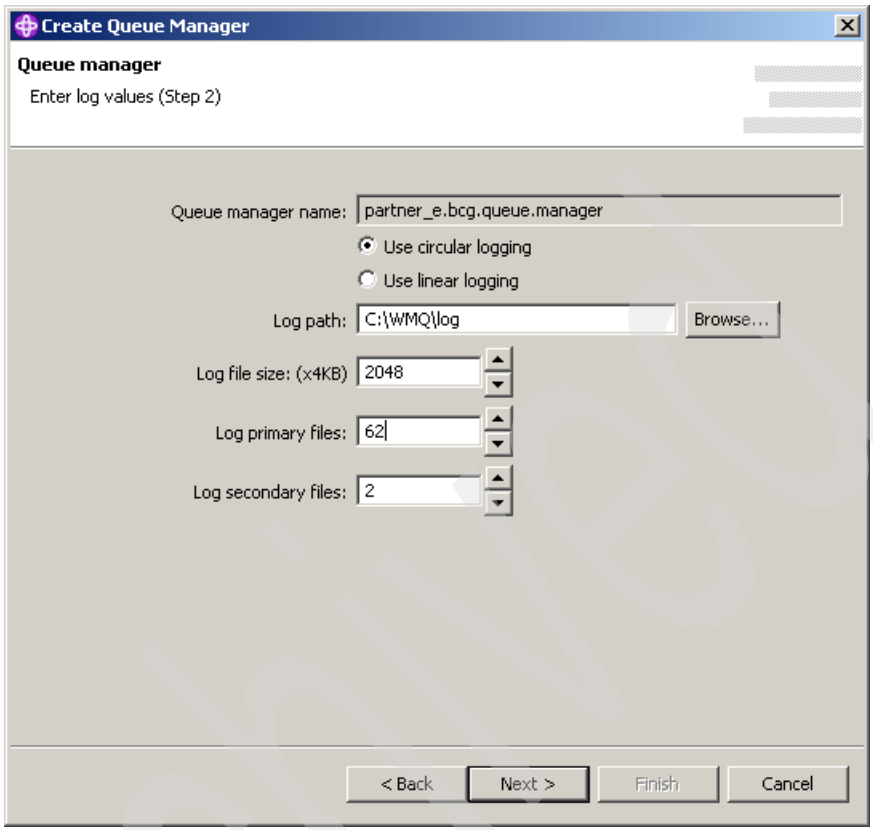


Figure 5-6 Create Queue Manager: Step 2

10. Click **Next.**

11. Select **Auto start queue manager**. This ensures that the queue manager starts when the WebSphere MQ service is started. See Figure 5-7 on page 68.

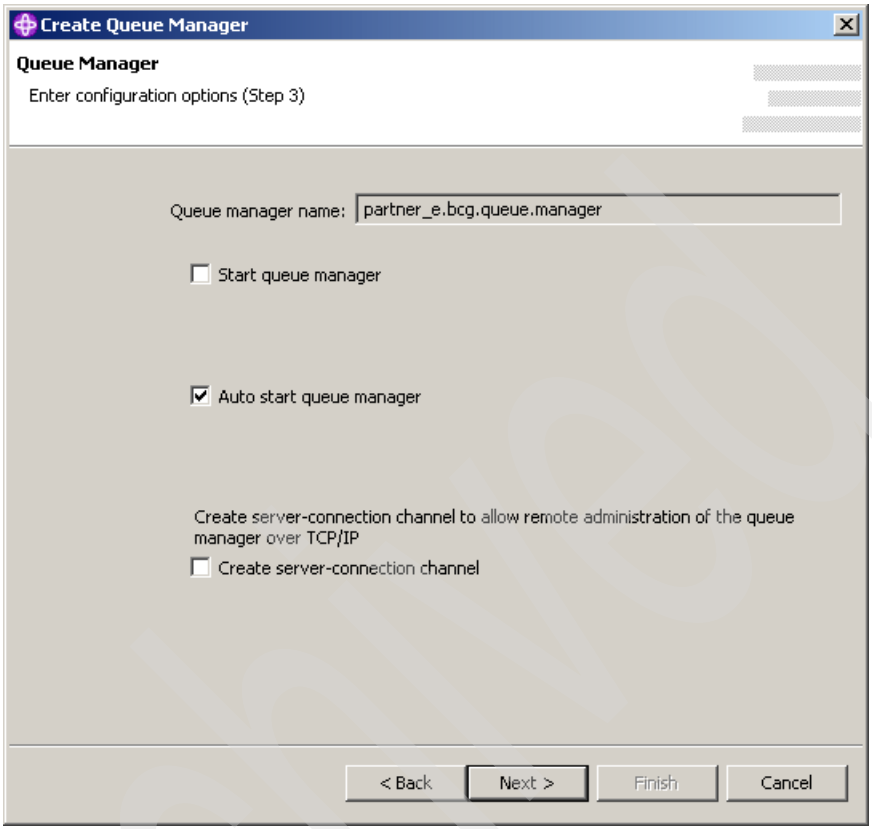


Figure 5-7 Create Queue Manager: Step 3

12. Click **Next**.
13. Deselect the option to create a TCP/IP listener. We perform this task later in this section, but we use another method. See Figure 5-8 on page 69.

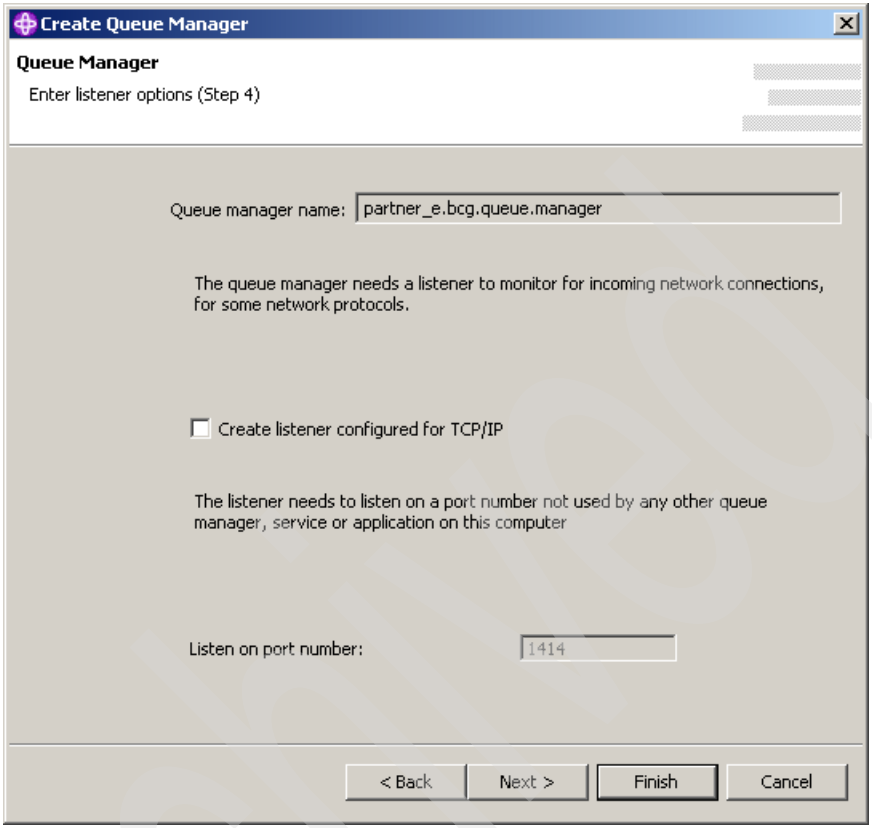


Figure 5-8 Create Queue Manager: Step 4

14. Click **Next.**
15. Accept the default settings for the auto connect parameters. See Figure 5-9 on page 70.

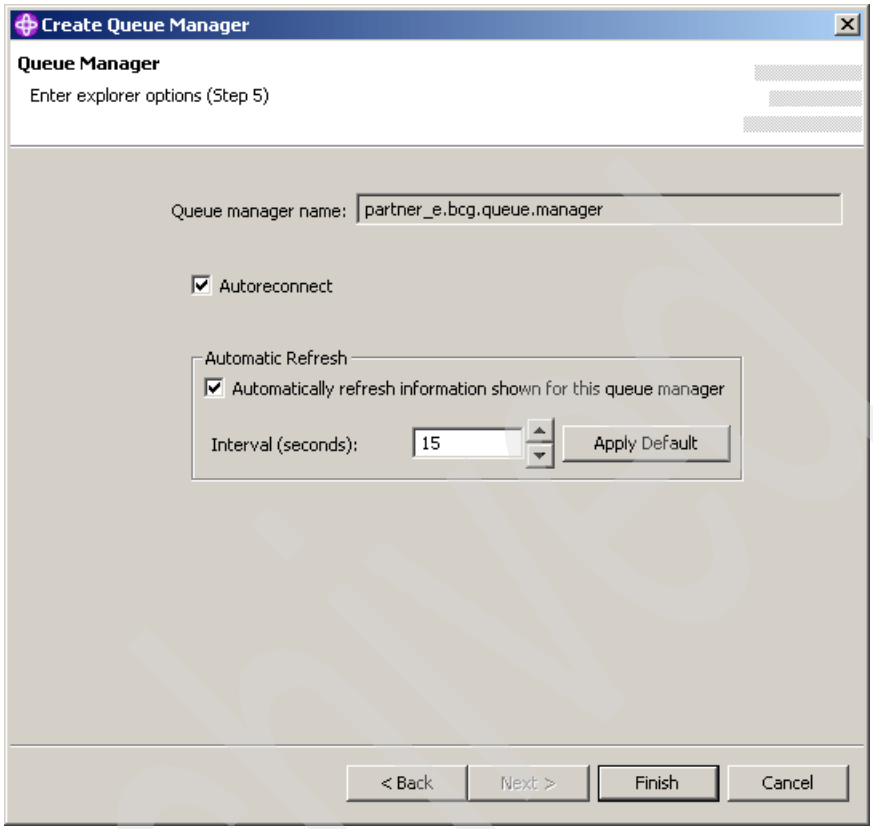


Figure 5-9 Create Queue Manager: Step 5

16. Click **Finish.**

17. The new queue manager is created. See Figure 5-10,

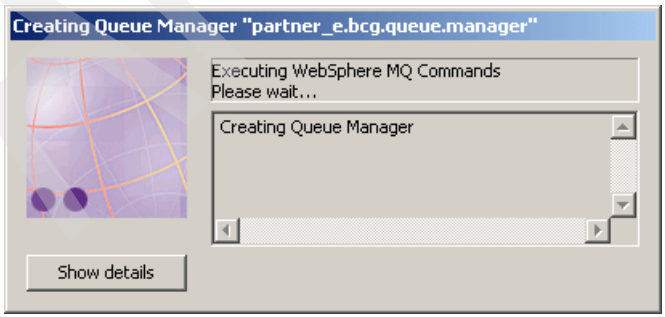
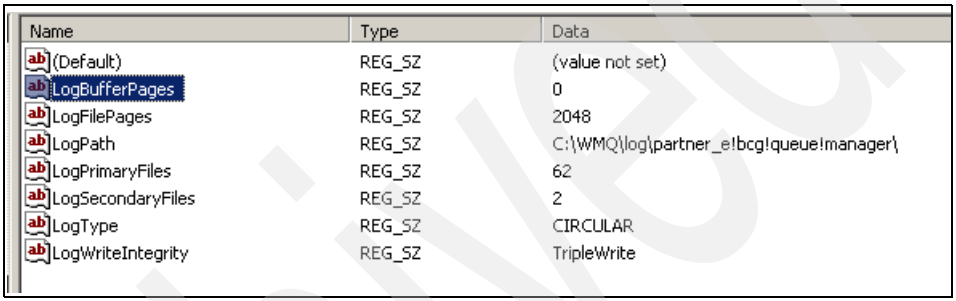


Figure 5-10 Creating Queue Manager

After the new queue manager has been created, we need to amend one more logging parameter before we use our queue manager. If your queue manager is running, shut it down.

Edit the registry values for the logs:

1. Open a command window and enter **regedit.**
2. Expand the registry.
3. Select **HKEY_Local_Machine** → **Software** → **IBM** → **MQSeries** → **CurrentVersion** → **Configuration** → **QueueManager** → **partner_e!bcg!queue!manager** → **Log**. See Figure 5-11.



Name	Type	Data
(Default)	REG_SZ	(value not set)
LogBufferPages	REG_SZ	0
LogFilePages	REG_SZ	2048
LogPath	REG_SZ	C:\WMQ\log\partner_e!bcg!queue!manager\
LogPrimaryFiles	REG_SZ	62
LogSecondaryFiles	REG_SZ	2
LogType	REG_SZ	CIRCULAR
LogWriteIntegrity	REG_SZ	TripleWrite

Figure 5-11 Edit the value for LogFilePages

4. Enter 2048 for the value of LogFilePages, as in Figure 5-12.

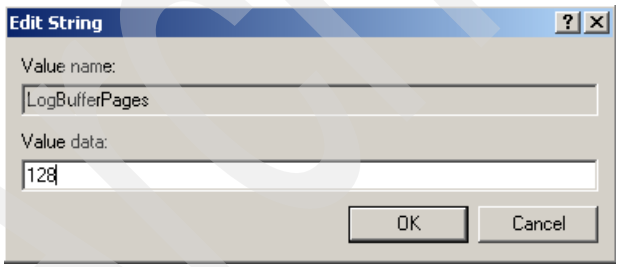


Figure 5-12 Enter value

5. Click **OK.**
6. Go back to the WebSphere MQ Explorer and open the properties of the newly created queue manager as shown in Figure 5-13 on page 72.
7. Select the **Channels** property.
8. Change the value for MaxChannels and MaxActiveChannels to 1000. This ensures that WebSphere Partner Gateway can establish sufficient MQ client sessions when required.

Archived

9. Click **Apply.**
10. Start the queue manager.

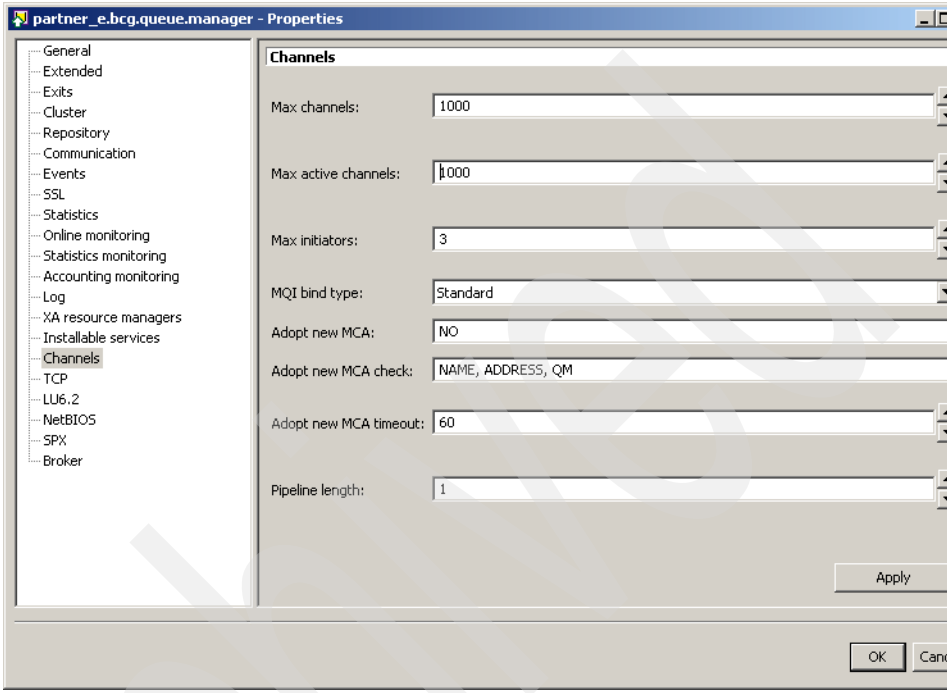


Figure 5-13 Alter channel properties

In addition to the queue manager we need a listener:

1. Expand the **Advanced** folder under the queue manager.
2. Right-click the **Listeners** folder and select **New** → **Listener**. See Figure 5-14 on page 73.

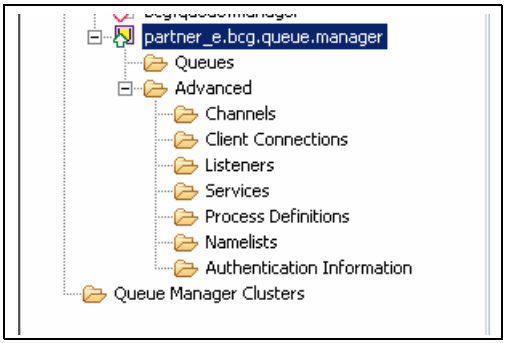


Figure 5-14 Advanced queue manager properties

3. Give the new listener a name. We chose `wpg` in Figure 5-17 on page 75.
4. Click **Next.**

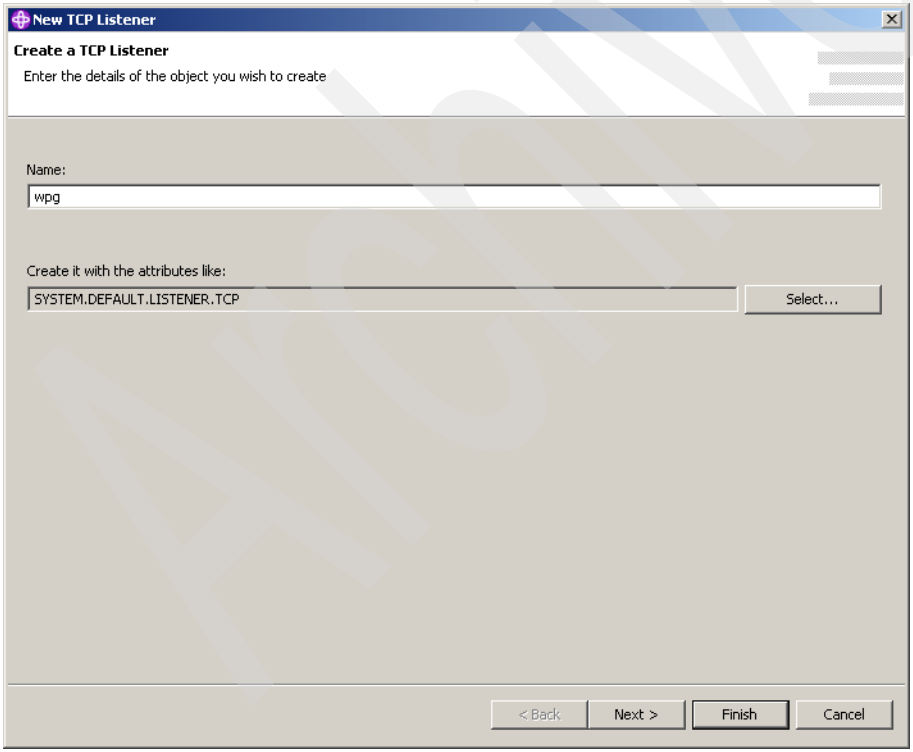


Figure 5-15 New listener

5. Using the menu box (Figure 5-16, “Enter listener properties” on page 74) select **Queue Manager Start** as the mode of control. This means that as part of the queue manager start up, it will start the listener.
6. Enter a Port number. We chose 9999, which is the recommended default for the WebSphere Partner Gateway queue manager.

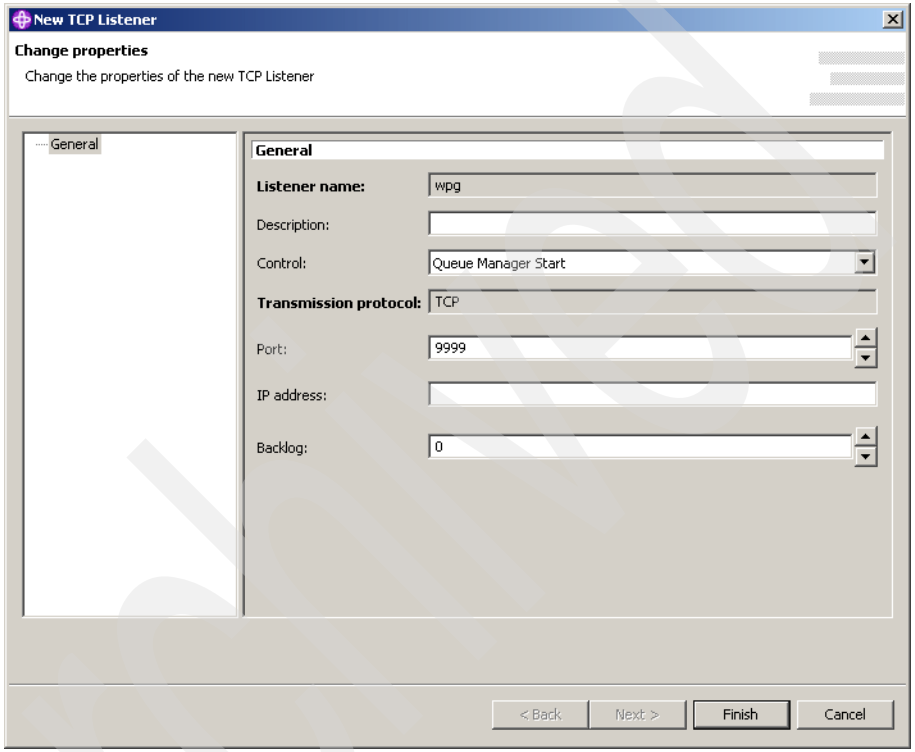


Figure 5-16 Enter listener properties

7. Click **Finish.**
8. When the listener has been created, you will receive a message similar to that seen in Figure 5-17 on page 75.

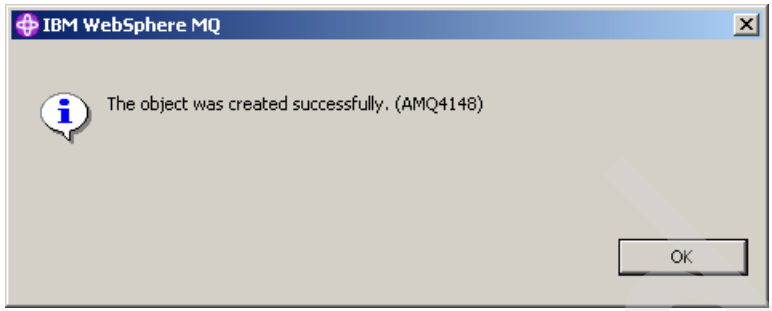


Figure 5-17 Successful creation

In addition to the queue manager, we also need a broker. This broker is used by components of WebSphere Partner Gateway that use publish/subscribe messaging. To ensure that the broker starts at the same time as the queue manager, we need to add the broker as a custom service to the service that represents the queue manager.

In the WebSphere MQ Explorer, follow these steps:

1. Right click the **Services** folder.
2. Select **New** → **Service**.
3. Give this service a name. We chose `wpg_broker` as shown in Figure 5-18 on page 76.
4. Click **Next.**

Archived

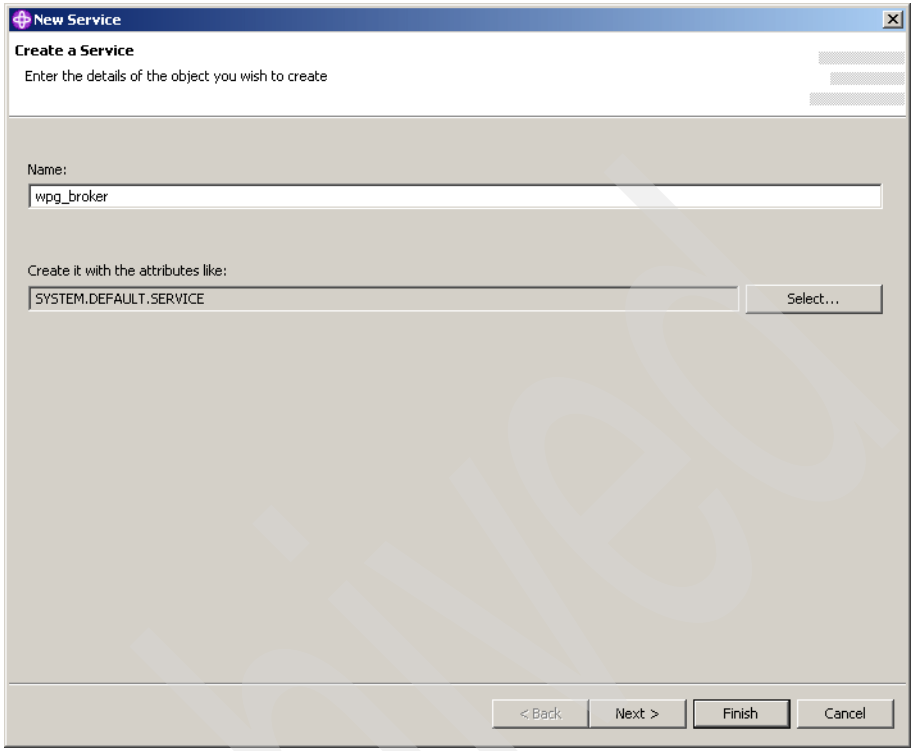


Figure 5-18 New service

5. As shown in Figure 5-19 on page 77:
   a. Using the menu box, set the Service control mode to **Queue Manager Start**.
   b. In the start command enter: `strmqbrk`

      This is the command to start the imbedded broker.
   c. In the Start args field, enter `-m partner_e.bcg.queue.manager.`

      This is the parameter required to indicate that it is the broker for the partner_e.bcg.queue.manager.

Archived

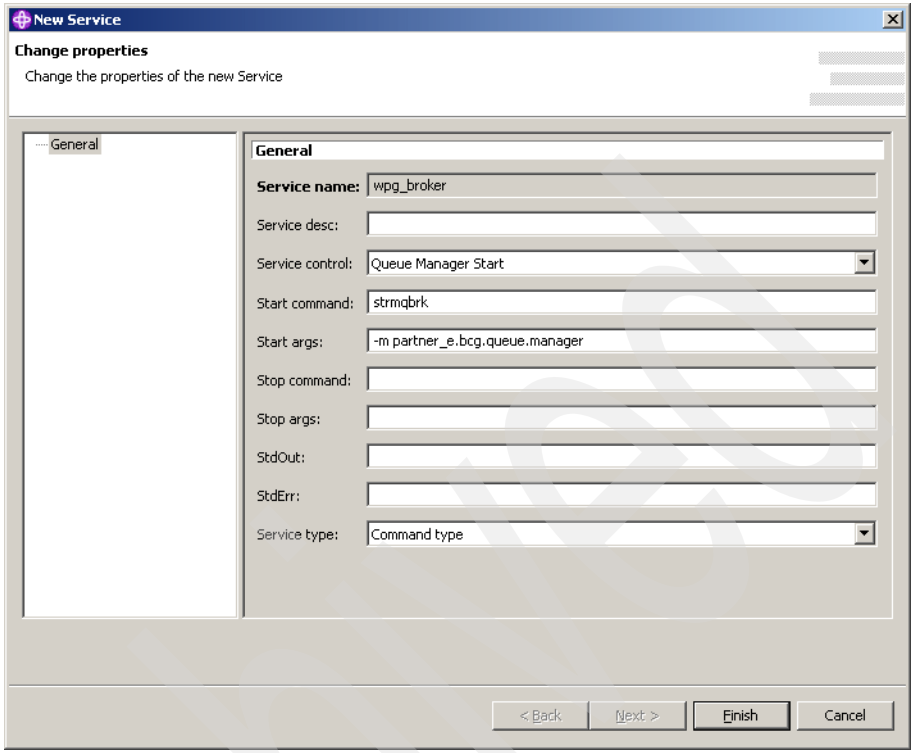


Figure 5-19 Enter service parameters

6. Enter `endmqbrk` as the Stop command, with the same Stop args as used for the Start args.
7. Click **Finish**.
8. Shutdown the queue manager to enable us to test the automatic start of our listener and broker. (Right-click the queue manager and select **Stop**.) See Figure 5-20 on page 78
9. **Select a Controlled shutdown.**
10. Click OK.

Archived

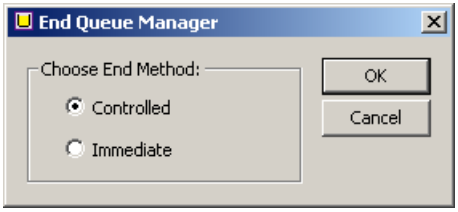


Figure 5-20 End queue manager

11. After the queue manager has ended, start the queue manager.
12. From the WebSphere MQ Explorer, you can see that the queue manager and listener are running.
13. Open a command window and check that the broker is running, as shown in Example 5-3.

Example 5-3 Display broker

```
C:\>dspmqbrk -m partner_e.bcg.queue.manager
WebSphere MQ Publish/Subscribe broker for queue manager
partner_e.bcg.queue.manager running.
```

We are almost completed with the MQ configuration. We need to issue two more commands to configure the resources.

1. Open a command window and change to the folder `\Tools\MQSeries` on the installation media. Enter:

   ```
   runmqsc partner_e.bcg.queue.manager < BCGCreate_Queues.mqsc
   ```

2. Navigate to the `\Java\bin` directory of the WebSphere MQ install directory. Enter:

   ```
   runmqsc partner_e.bcg.queue.manager < MQJMS_PSQ.mqsc
   ```

The first command defines a number of MQ resources used by WebSphere Partner Gateway. The second command defines a number of MQ resources which are used by JMS Publish/Subscribe.

Archived

5.3.3 Installing the database schema

Besides WebSphere MQ, WebSphere Partner Gateway relies on the services of a database manager. For this redbook, we use DB2. However, other database products are also supported. The database is used to store data that can be classified into two broad categories: profile information and logging information.

To simplify, the *profile information* is largely a read-only category when the system is configured. Changes occur only when new profiles are added or when an existing profile is updated, for example replacing an expired certificate.

The *logging information* is a highly active category of information. A single document exchange results in many logging events, which is useful in keeping track of what happens in the system or what has happened in the system. Thus, this category of information expands constantly and is consulted mostly by online users and administrators.

Given the different nature of the database access to both categories of information, it comes as no surprise that WebSphere Partner Gateway has grouped them into two different database tablespaces so that it becomes easy to manage them separately. This can mean storing the information on separate disks and implementing different backup and reorganization procedures.

1. Start the product installation program from the product CD.
2. You see several options. The two main tasks are:
   - Create the Database
   - Install the Product

   Select **Create the Database** (Figure 5-21 on page 80) to launch the database loader program. Be aware that the database loader program creates a database that is owned by the currently logged-on user. If you want the database to be owned by the database instance owner (for example, db2admin), log on to the system with this user ID.

   The database loader program does not perform a context switch, even though you might think it is going to do so, based on the information that you provide.

Archived

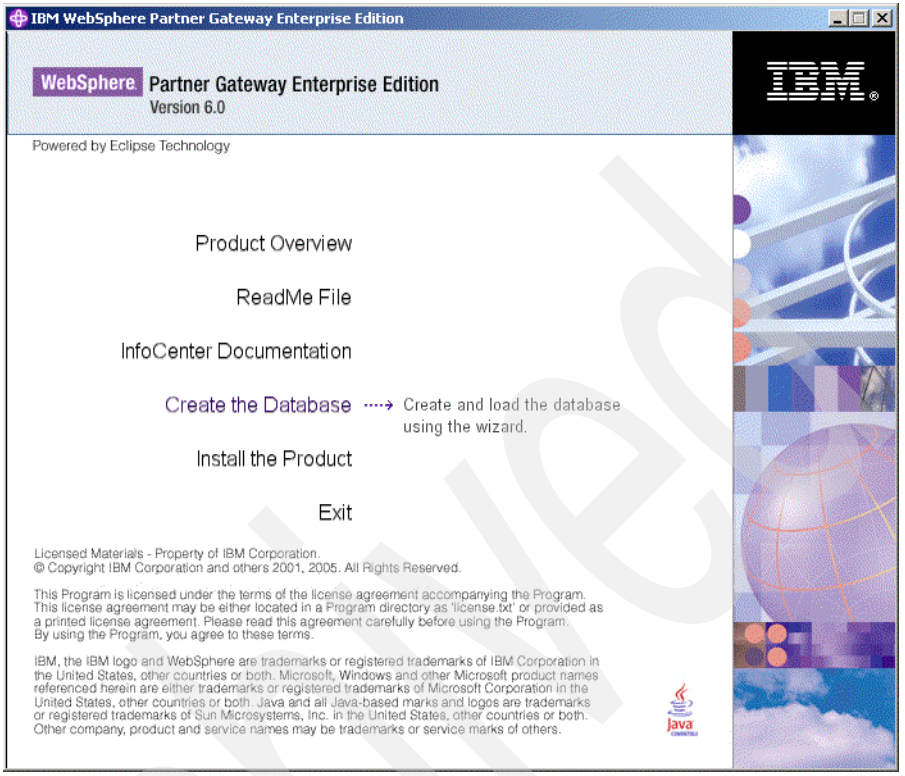


Figure 5-21 Select Create Database

3. When the database loader program is started, the Welcome window opens. Click **Next**.
4. Accept the software license. Then click **Next**.
5. Provide an installation directory, `D:\WPG6.0\bcgdbloader`, for example.
6. Select the database WebSphere Partner Gatewayserver product that is going to be used. This can be:
   - IBM DB2 8.2 or later
   - Oracle 9i 9.2.0.4 or later

   The following figures show the choices that we made for our environment and are specific for DB2. The product documentation explains this for Oracle as well.

Archived

7. In the Database Loader window (see Figure 5-22), enter the following information:
   a. Enter the name of the new database for use by WebSphere Partner Gateway, which by default is `bcgapps`.
   b. Type the name of the database instance. A standard Windows installation of DB2 has a default instance called `DB2`. Other instances can also be created. This instance *must* exist before you run the database loader program.
   c. Specify the name of the group to which the database owner user ID belongs. On Windows, this is usually the standard group `Administrators`.
   d. Enter the user ID and password of the database instance owner for the program.
   e. Click **Next** to proceed. See Figure 5-23 on page 82.

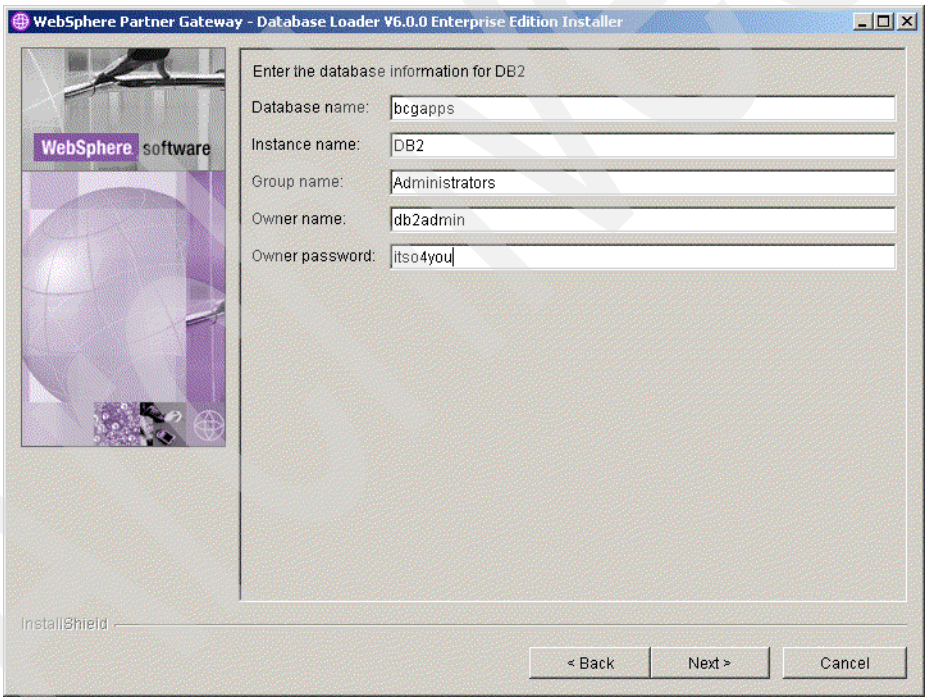


Figure 5-22 Provide the database system information

8. In the next window (Figure 5-23), you provide location information to store the data. As explained before, the WebSphere Partner Gateway tables are grouped into several categories that are mapped to tablespaces. For optimal performance, allocate directories of different disks for the main table spaces. The default values presented are fine for an initial setup of the product.

Note: The database loader program can be used so that the generated SQL statements are stored in a file and not executed. This enables you to review them with your database administrator and execute the SQL scripts manually

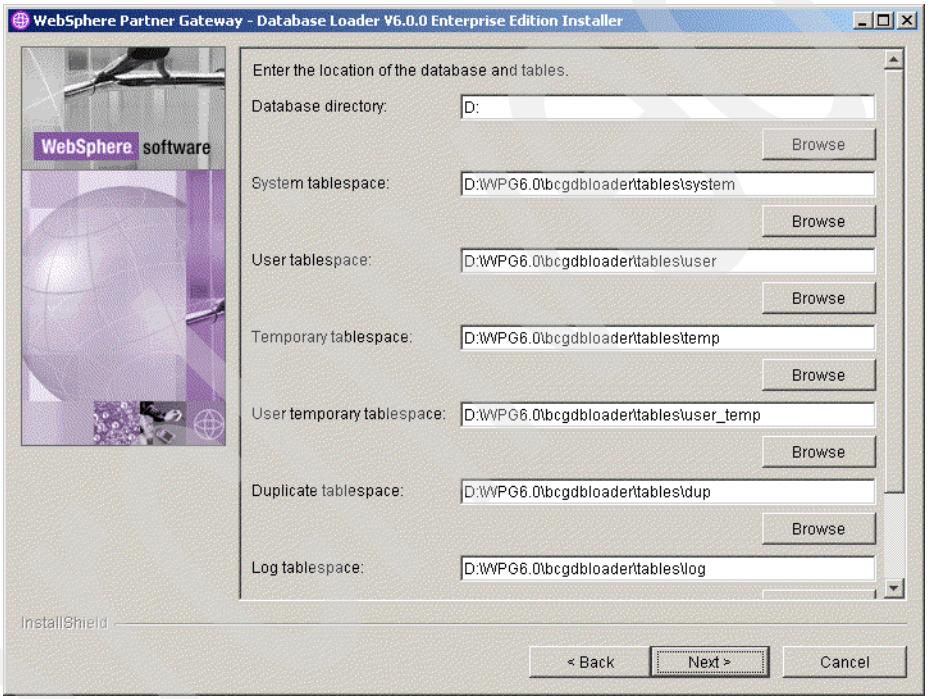


Figure 5-23 Location information for tablespaces

9. In the next window, provide the name of the WebSphere Partner Gateway group, as well as the user IDs and passwords. The values shown in Figure 5-24 are the default values and match the values used in 5.3.1, “Adding user IDs and a group” on page 63. These user IDs are used to access the database.

   Click **Next** to proceed.

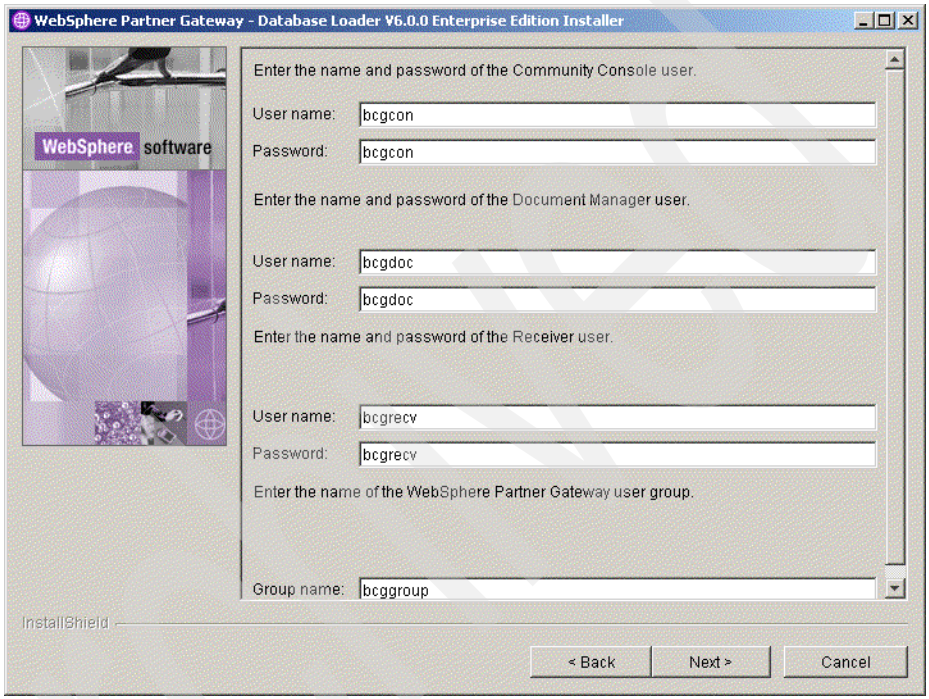


Figure 5-24 Provide user ID and password

10. In the next window (Figure 5-25 on page 84), provide the name of the location where the common data is stored. Common data is the data exchanged between the three different runtime components of the server. In an environment with multiple servers, this should be the name of the mount point on these servers. Assume that each of the three server components runs on a dedicated machine, and a network share must exist that allows the server components to pass information to each other.

    Provide the name of this network share as it is known on the server machines and not necessarily on the database server.

Note: The value used in Figure 5-25 implies that you will install WebSphere Partner Gateway on the drive that has been mapped as drive G: drive of the WPGV6DB machine.

Click **Next**.

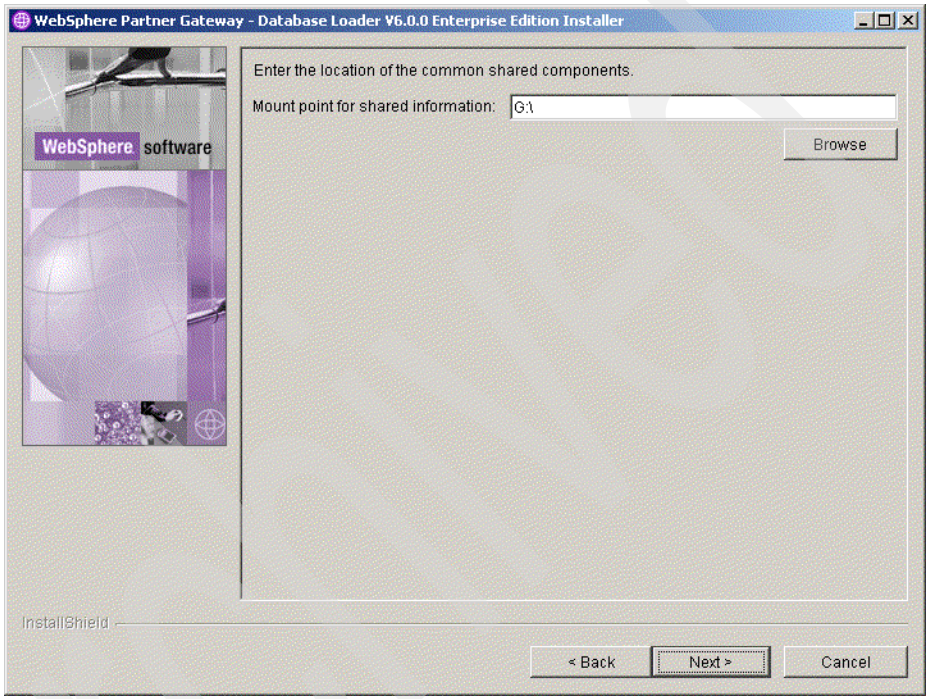


Figure 5-25 Mount point for common data

11. When all files are copied, you choose whether you want the database loader to run the generated SQL scripts or whether you want to run them manually, for example, after a detailed review of the scripts. See Figure 5-26 on page 85.

    Detailed steps on how to run the scripts manually are provided in a text file Instructions.txt that is stored in the \WPG6.0\DBLoader\scripts\DB2 folder. This text file also contains information about the process to delete the database, if you need to restart the process due to a failure, for example.

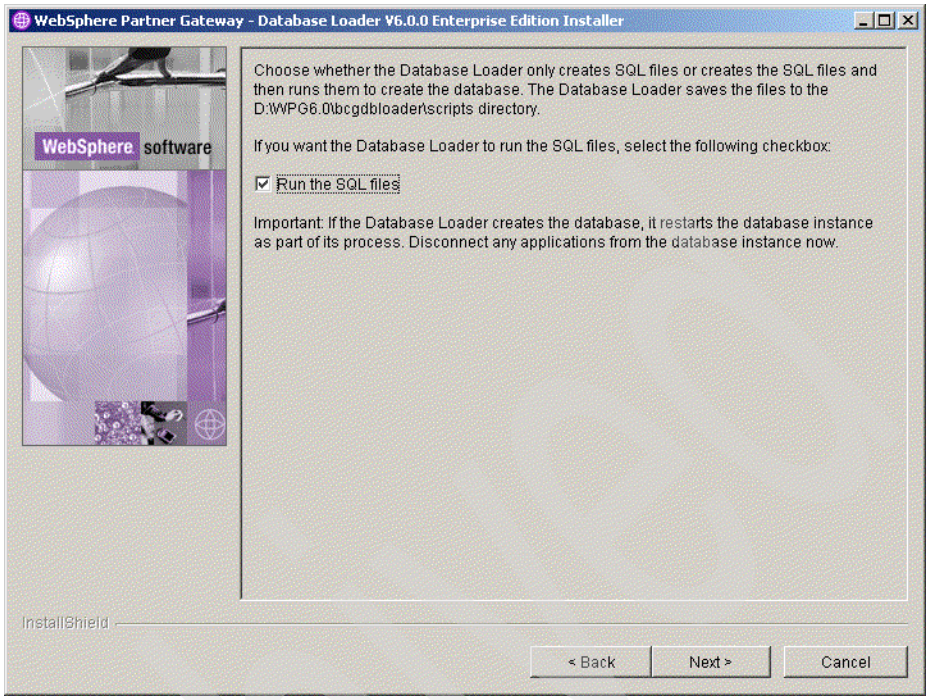


Figure 5-26 Choose to run scripts now or later

12. An installation summary window opens. Click **Next** to start the installation.

5.3.4 Local validation

Before proceeding, we recommend that you perform some validation. The database loader program creates many objects and the log file is quite large. The following steps perform some validation, but it is not conclusive. You must review the log file.

A quick validation can consist of the following actions:

1. Connect to the database with each user ID.
2. Count the stored procedures that were created. The creation of the stored procedures is the last step in the database loader program. As such, if all stored procedures are created, you can conclude that all previous steps were successful and that all objects used by the stored procedures are also created.

Example 5-4 shows the commands to perform this in a DB2 command window. The number of stored procedures that should be created is 643.

Example 5-4 Verify database

```
db2 => connect to bcgapps user bcgcon using bcgcon

   Database Connection Information

 Database server        = DB2/NT 8.2.1
 SQL authorization ID   = BCGCON
 Local database alias   = BCGAPPS

db2 => select count(*) from syscat.procedures where procschema='DB2ADMIN'

1
-----------
        643

  1 record(s) selected.
db2 => connect reset
DB20000I  The SQL command completed successfully.
```

5.4 Install WebSphere Partner Gateway

The previous section discusses the installation and configuration on the machine wpgv6db, which is used as the MQ and database server. This section explains how to install and configure WebSphere Partner Gateway on the machine wpgv6hub. You perform the following tasks:

1. Add a user and group.
2. Install WebSphere Partner Gateway Enterprise Edition.

At the end of this section, you perform a simple validation to make sure that WebSphere Partner Gateway is operating correctly before you perform any customization.

5.4.1 Adding a user and group

An additional user ID is required. The default name is `bcguser`, which is also a member of the group `bcggroup`. Because our environment does not have a common security system, such as a Windows domain, we create this user ID and the group on the machine `wpgv6hub` that will host the WebSphere Partner Gateway run time.

Use the same steps as explained in 5.4.1, “Adding a user and group” on page 86 to define the user and the group. Be sure to *deselect* the **User must change password at next logon** option and *select* the **Password never expires** option.

5.4.2 Installing the product code

To install the product code, follow these steps:

1. Launch the main installation program from the product CD. This time select the **Install the Product** option, as shown in Figure 5-27.

Figure 5-27 Launchpad

2. In the Welcome window, click **Next**.
3. Accept the license agreement, and click **Next**.
4. In the next window, provide the name of an installation folder, for example `D:\WPG6.0\bcghub`. Click **Next**.

Archived

5. Specify which components are going to run on this system (see Figure 5-28). Because we intend to create a single server, we select all components. Click **Next.**

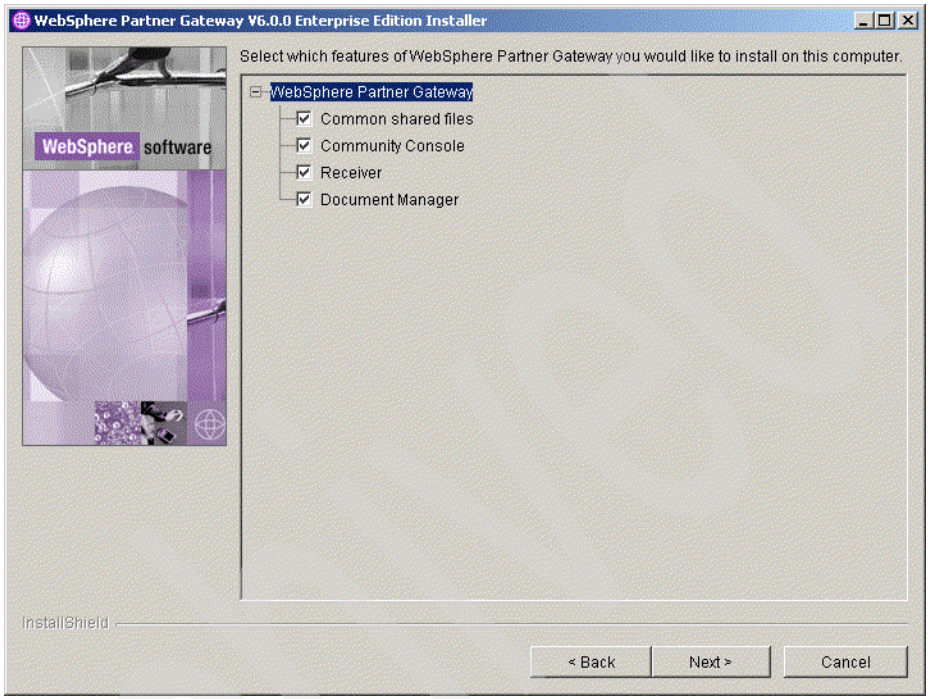


Figure 5-28 Select all required components

6. Specify the host name of the computer. Our hostname is: `WPGV6HUB.itso.ral.ibm.com`. Click **Next.** See Figure 5-29 on page 89.

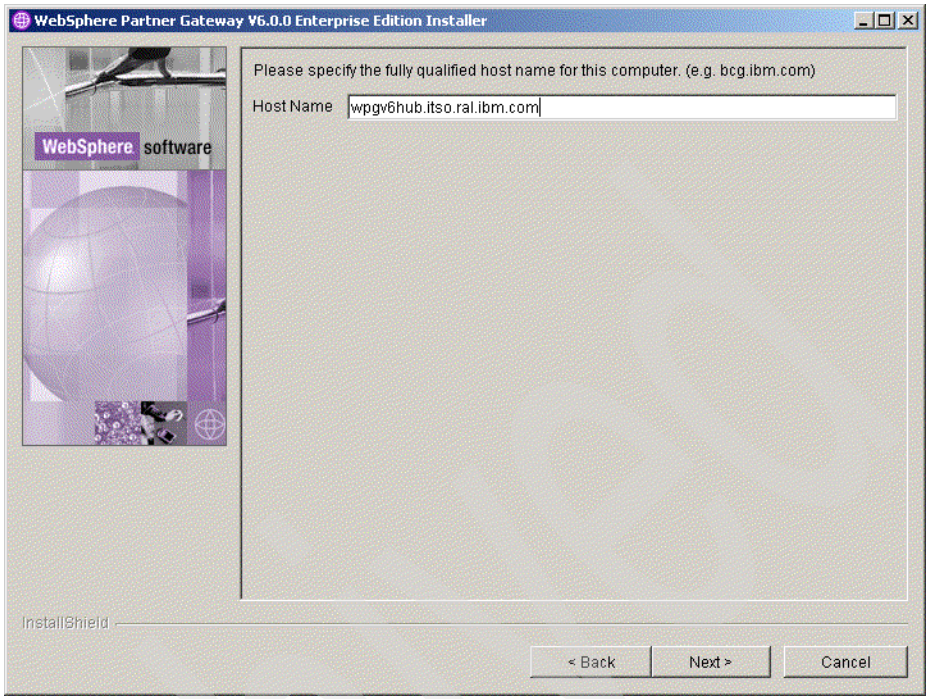


Figure 5-29 Specify hostname

7. Select the option **Use WebSphere Application Server v6.0 which is already installed on this Computer** as shown in the Figure 5-30 on page 90. Click **Next.**

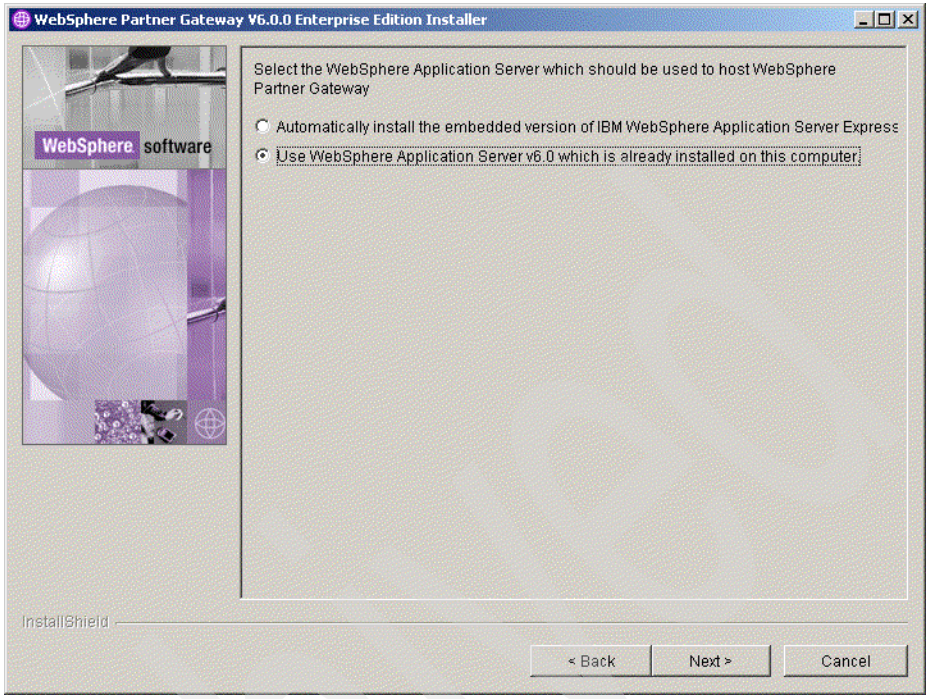


Figure 5-30 Use existing WebSphere Application Server installation

8. Specify the location of the directory where WebSphere Application Server is installed. We used `D:\was\bin`. Click **Next**. See Figure 5-31 on page 91.

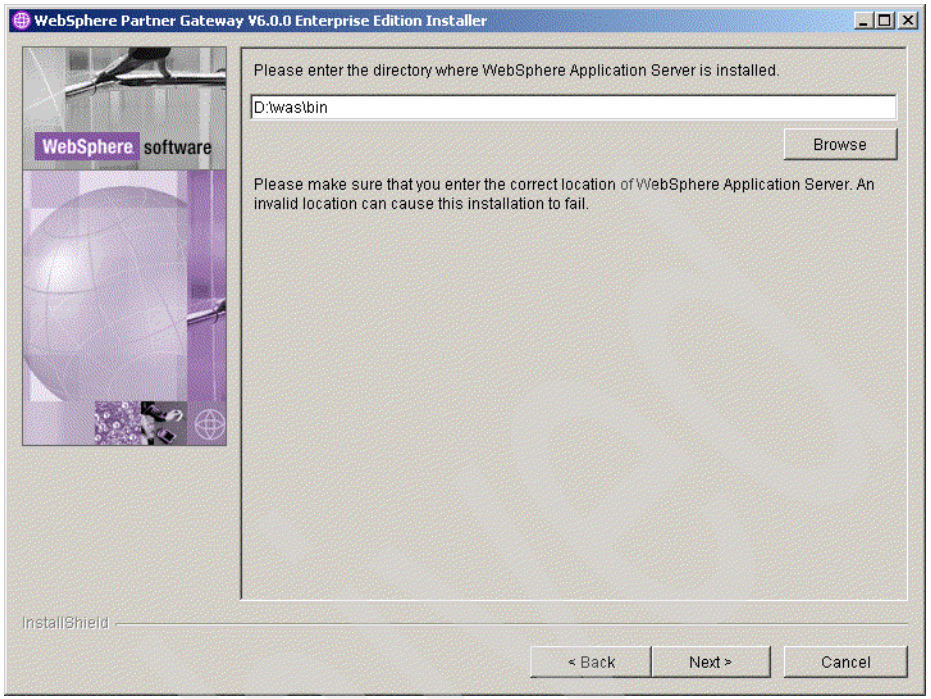


Figure 5-31 WebSphere Application Server root path

9. As in Figure 5-32 on page 92, Select the database product, either **IBM DB2** or **Oracle**.

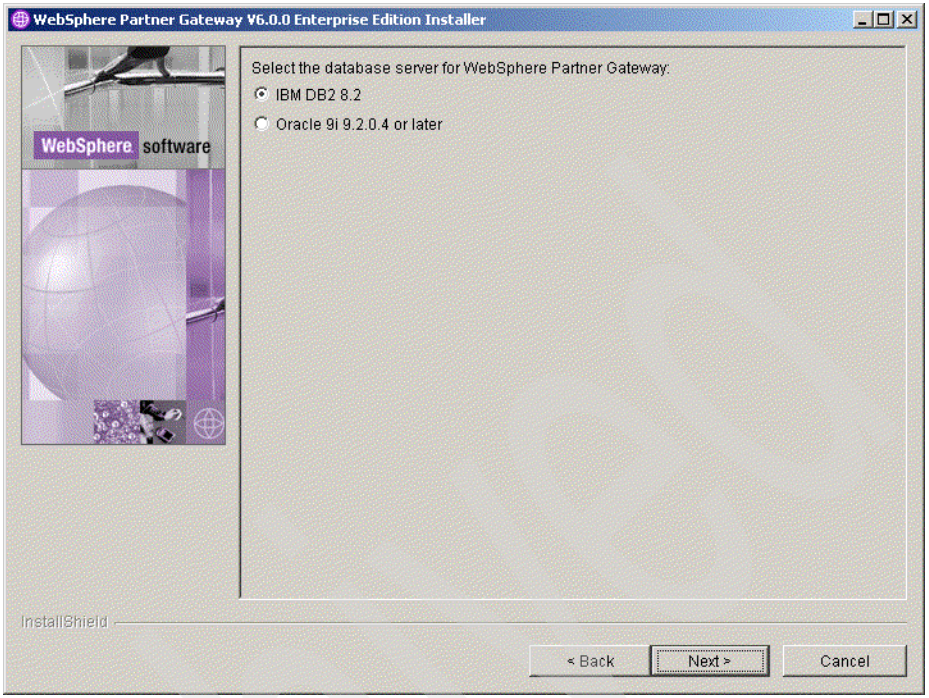


Figure 5-32 Select database type

10. Based on your selection, provide the DB2 or Oracle specific information. For DB2, WebSphere Partner Gateway needs to know the following information. The values we used are shown in Figure 5-33 on page 93:

    – The host name of the database server: `WPGV6DB`
    – The port number used by DB2, which is usually `50000`
    – The owner name and password: `db2admin`
    – The database name: `bcgapps`
    – The schema name: `db2admin`

    Click **Next** to continue to Figure 5-33 on page 93.

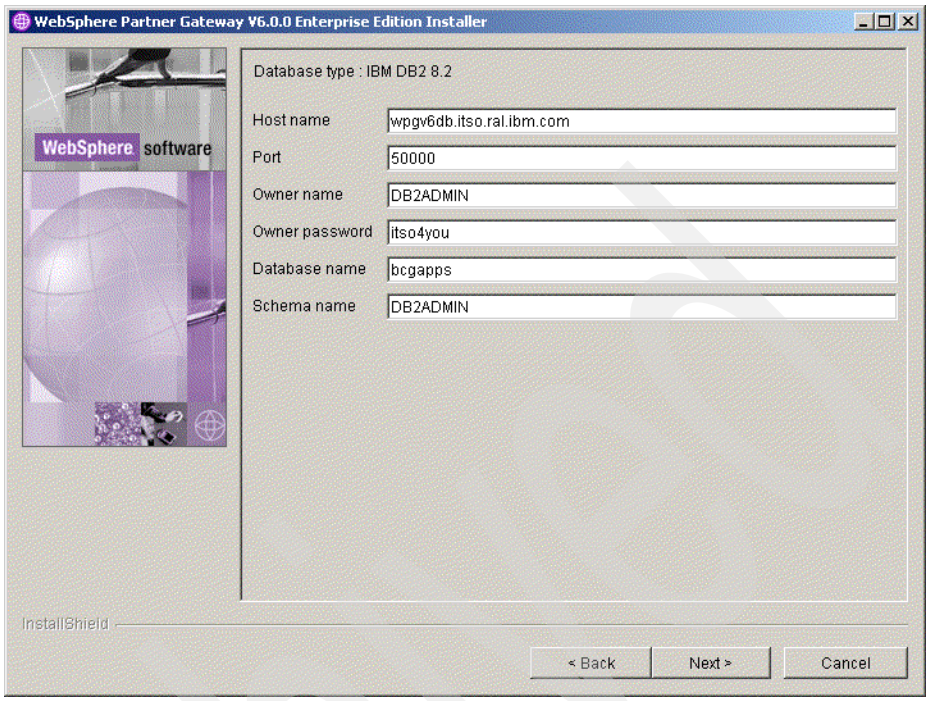


Figure 5-33 Database details

11. The installation program connects to the database server and retrieves information that the database loader program created. Figure 5-34 on page 94 shows the result of these queries. Click **Next** to proceed.

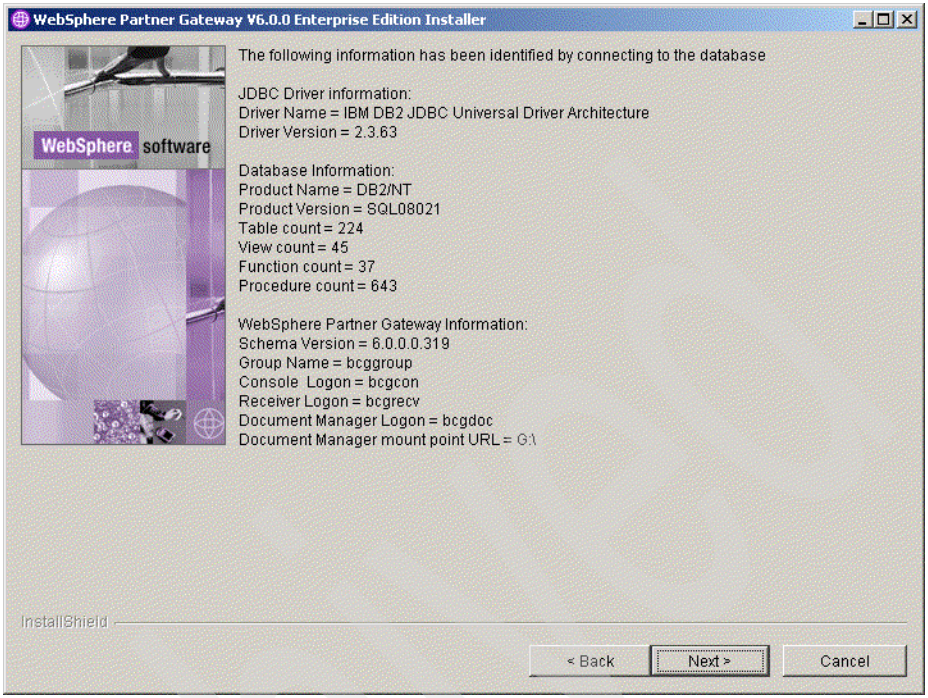


Figure 5-34 Database information

12. Provide the user and group information that is responsible for the installed files and executing the Server Processes.

    Fill in the following entries as shown in Figure 5-35 on page 95:

    – Enter the Username as `bcguser`

    – Enter the Password for the user `bcguser`

    – Enter the Group name as `bcggroup`

    Click **Next to** proceed.

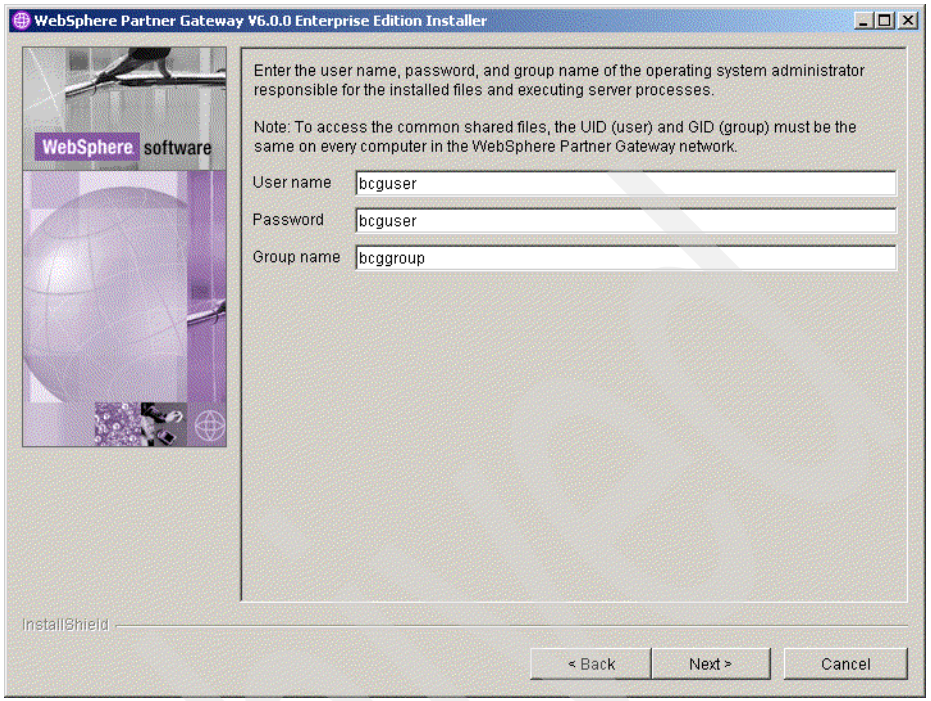


Figure 5-35 User details

13. Enter the directory location for the Common folder. We chose `D:\WPG6.0\bcghub\common` in Figure 5-36 on page 96.

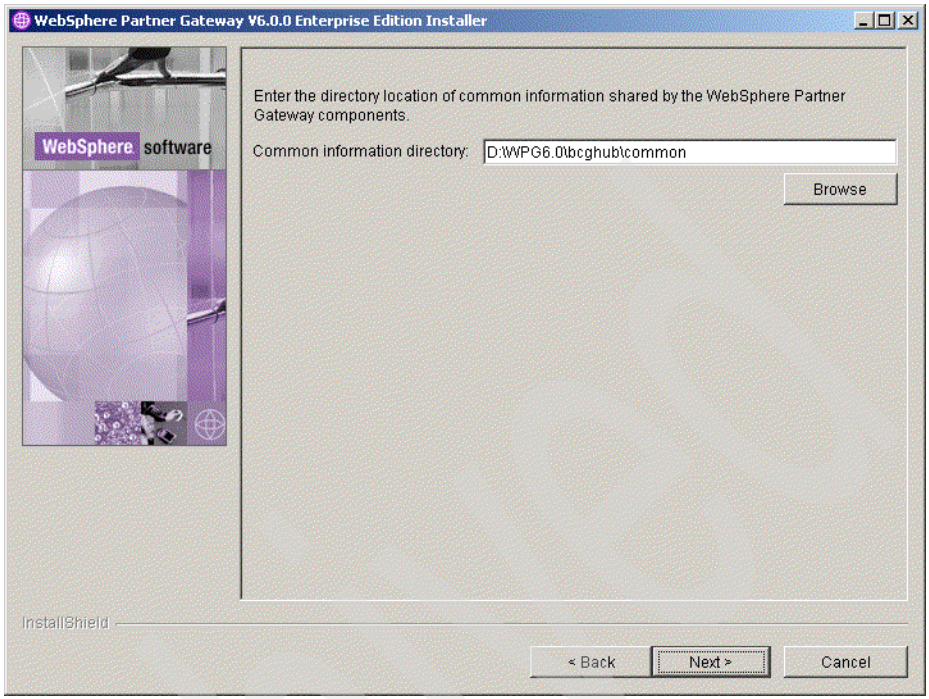


Figure 5-36 Common files directory

Important: If you recall, from Figure 5-25 on page 84, we have mapped a drive on the DB server to this common directory.

14. The installation program continues to request information about WebSphere MQ (Figure 5-37 on page 97). The installation program needs:
    - The host name of the MQ server: `wpgv6db.itso.ral.ibm.com`
    - The name of the queue manager: `partner_e.bcg.queue.manager`
    - The port number used by the MQ listener: `9999`

    Note: The installation program does not ask for the name of the server connection channel, which is another piece of information that is required to create an MQ client connection channel. WebSphere Partner Gateway uses a fixed name for this MQ object, `java.channel`. This object is listed in the MQ definition script provided by WebSphere Partner Gateway. It was created as part of the MQ setup in 5.3.2, “Configuring WebSphere MQ 6.0” on page 65.

    Click **Next**. See Figure 5-37 on page 97.

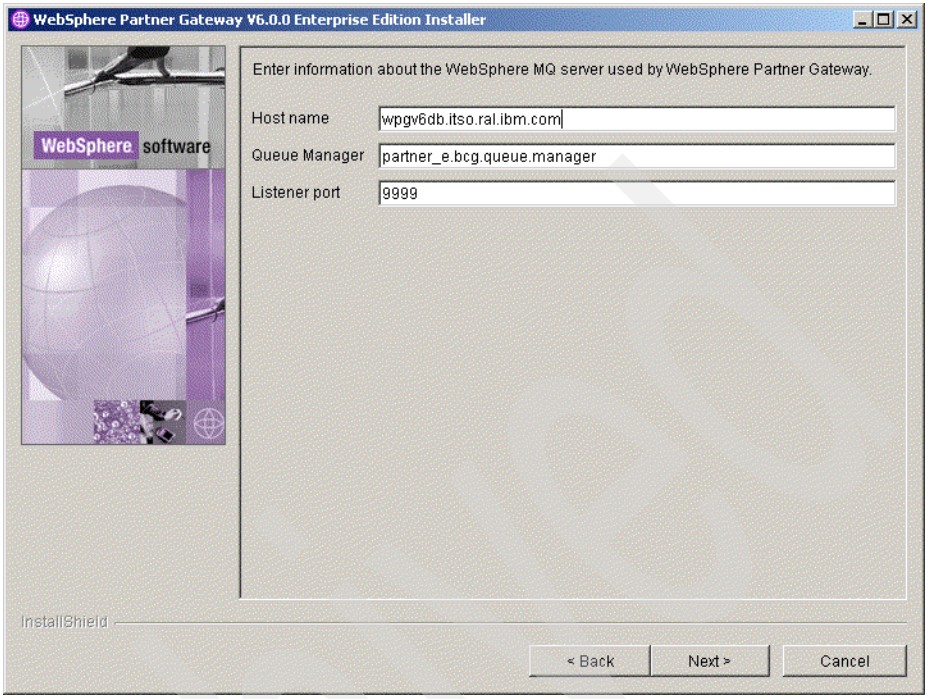


Figure 5-37 WebSphere MQ components

15. Indicate that the installation program should create Windows services for the three runtime components. Click **Next**.

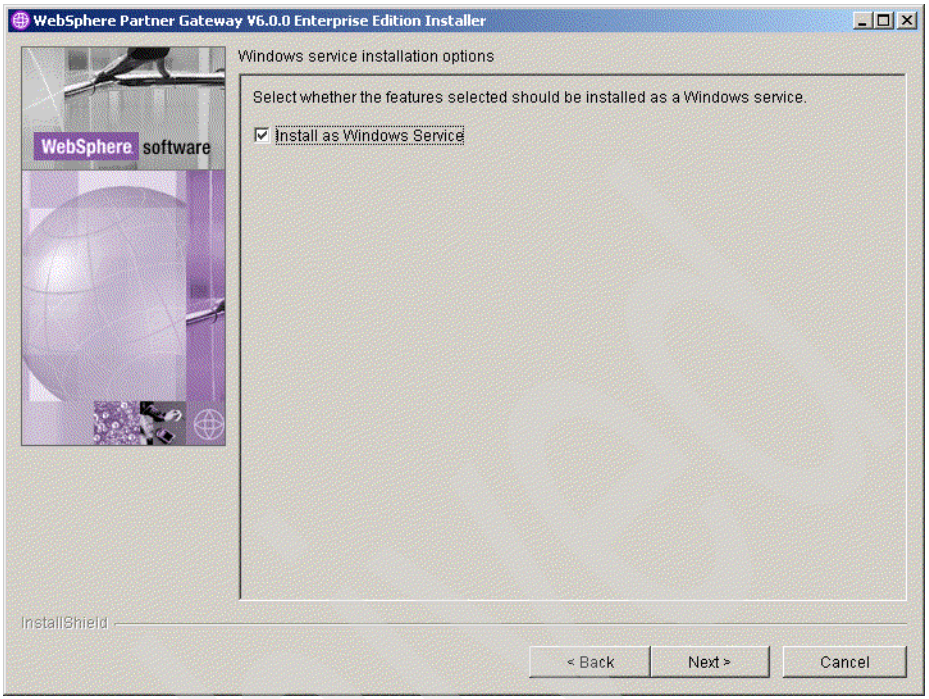


Figure 5-38 Create services

16. The installation program asks for configuration parameters for each of the three components. Figure 5-39 on page 99 shows these parameters forWebSphere Partner Gateway the Community Console component. A similar step follows for the Receiver and Document Manager components. The parameters are:
    - Name of the user ID that the component should use to access the database: `bcgcon`, `bcgrecv`, or `bcgdoc`
    - Password for that user ID
    - HTTP WebSphere Partner Gateway port to be used by the application server for this component
    - HTTPS port to be used by the application server for this component
    - Help system hostname: `wpgv6hub.itso.ral.ibm.com`
    - Help System port number: `58888`

    Click **Next**.

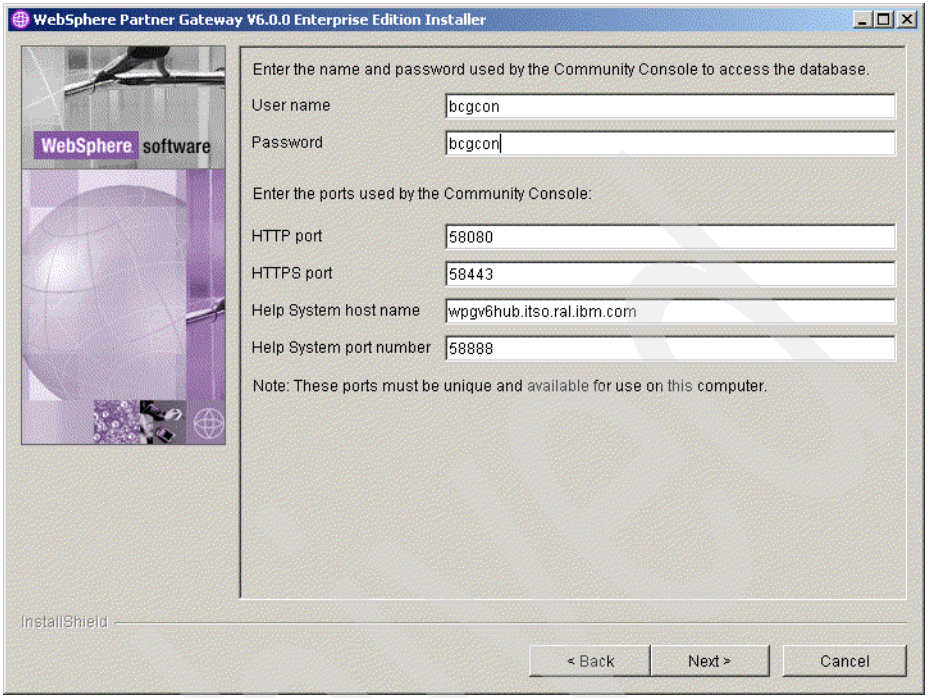


Figure 5-39 Community console access parameters

See Figure 5-40 on page 100.

Archived

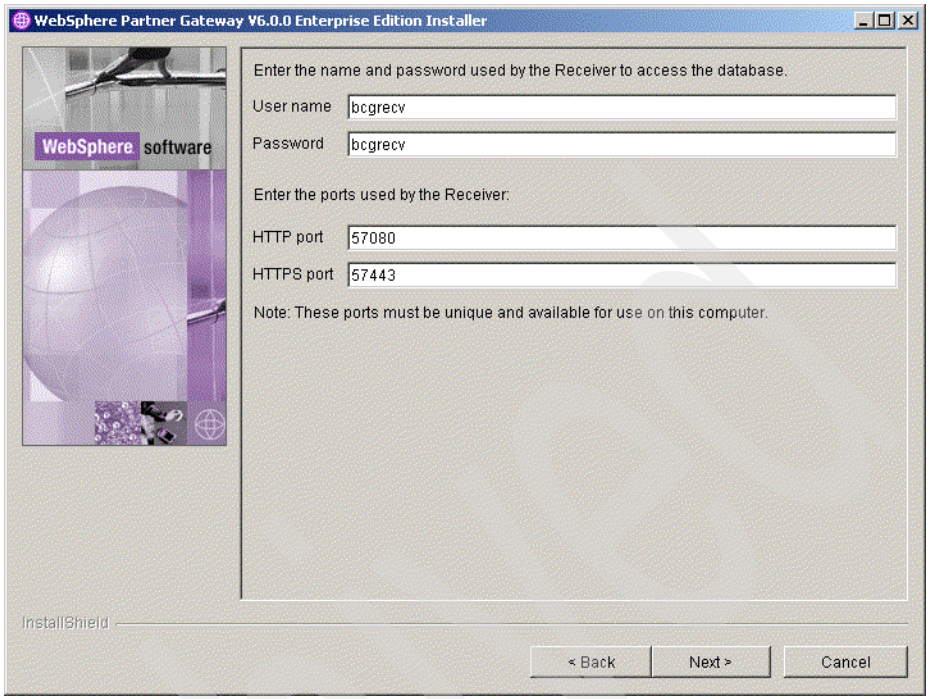


Figure 5-40 Receiver access parameters

See Figure 5-41 on page 101.

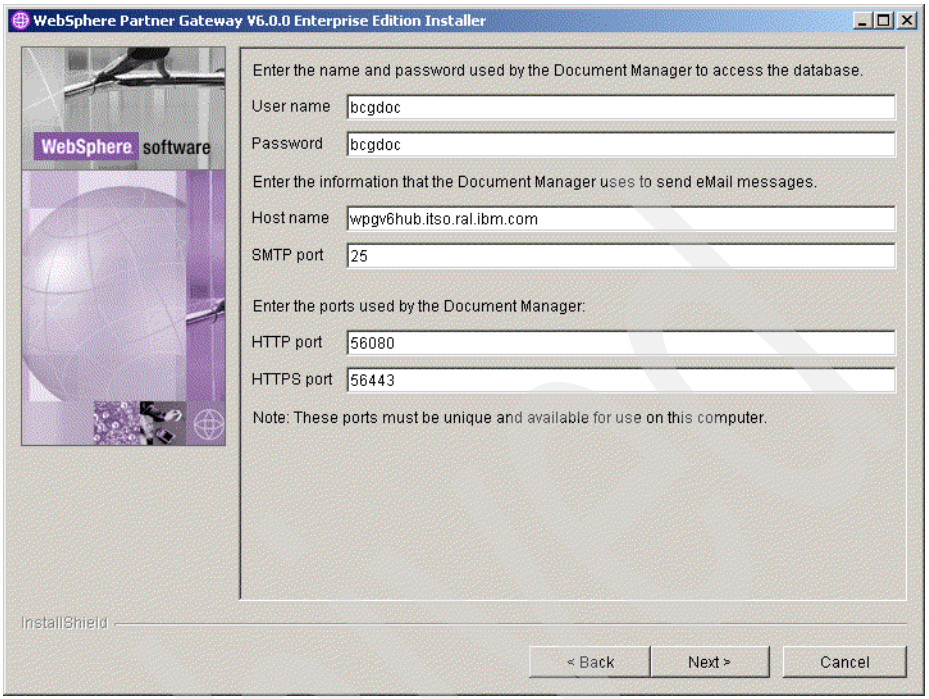


Figure 5-41 Document Manager access parameters

17. In the next two windows, you are prompted about the use of RosettaNet and SMTP, which are not used at this time.
18. Finally, you see a summary window (Figure 5-42 on page 102). Click **Next** to start the actual installation.

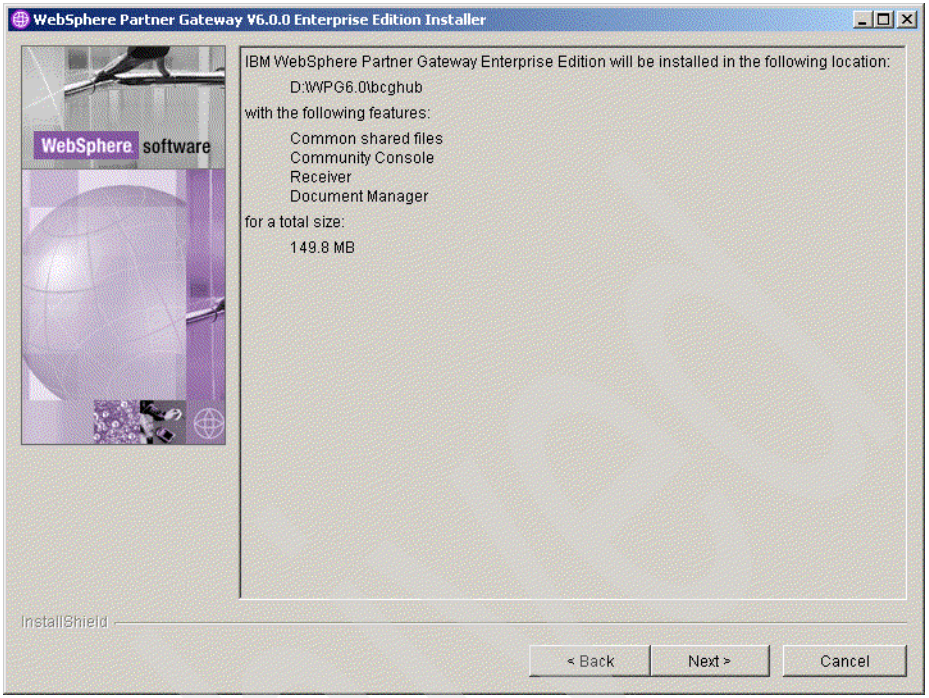


Figure 5-42 Start the install

5.4.3 Local validation

The simplest validation that you can perform is to start the components and verify the content of the log files. To start WebSphere Partner Gateway, you can either use the Windows Services application (Figure 5-43), or start the application servers from the command line.

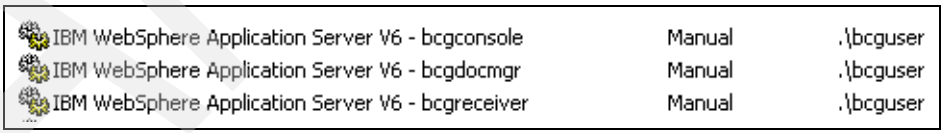


Figure 5-43 Start services

Tip: Remember, if you are going to start the servers as services, to ensure that the user bcguser has the correct authorization to do this. Go to the User Rights Assignment section of the Local Security Settings application in the Windows Control Panel to check or update as required.

You enter the following command, on a command line, for each component that is executed in the bin directory of that component, as shown in Example 5-5:

```
startserver bcgconsole
```

Example 5-5 Starting WebSphere Partner Gateway from a command line

```
D:\WPG6.0\bcghub\was\profiles\bcgconsole\bin>startServer bcgconsole
ADMU7701I: Because bcgconsole is registered to run as a Windows Service, the
           request to start this server will be completed by starting the
           associated Windows Service.
ADMU0116I: Tool information is being logged in file

D:\WPG6.0\bcghub\was\profiles\bcgconsole\logs\bcgconsole\startServer.
log
ADMU0128I: Starting tool with the bcgconsole profile
ADMU3100I: Reading configuration for server: bcgconsole
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server bcgconsole open for e-business; process id is 1488

D:\WPG6.0\bcghub\was\profiles\bcgdocmgr\bin>startServer bcgdocmgr
ADMU7701I: Because bcgdocmgr is registered to run as a Windows Service, the
           request to start this server will be completed by starting the
           associated Windows Service.
ADMU0116I: Tool information is being logged in file

D:\WPG6.0\bcghub\was\profiles\bcgdocmgr\logs\bcgdocmgr\startServer.lo
g
ADMU0128I: Starting tool with the bcgdocmgr profile
ADMU3100I: Reading configuration for server: bcgdocmgr
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server bcgdocmgr open for e-business; process id is 1808

D:\WPG6.0\bcghub\was\profiles\bcgreceiver\bin>startServer bcgreceiver
ADMU7701I: Because bcgreceiver is registered to run as a Windows Service, the
           request to start this server will be completed by starting the
           associated Windows Service.
ADMU0116I: Tool information is being logged in file

D:\WPG6.0\bcghub\was\profiles\bcgreceiver\logs\bcgreceiver\startServe
r.log
ADMU0128I: Starting tool with the bcgreceiver profile
ADMU3100I: Reading configuration for server: bcgreceiver
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server bcgreceiver open for e-business; process id is 1696
```

While the output in Example 5-5 already gives an indication that the startup was fine, we recommend that you verify the actual log files. Some errors are not

sufficient to cause a failure of the startup of the application server, but will result in error situations later when using WebSphere Partner Gateway.

The log files are called SystemOut.txt, while the error files are called SystemErr.txt. For each component, such a file is created and can be found in `D:\WPG6.0\bcghub\was\profiles\`*`component name`*`\logs\componentname`, where *`component name`* is one of the following names:

- bcgconsole
- bcgreceiver
- bcgrouter

5.5 Initial configuration of the WebSphere Partner Gateway server

In this section, we perform the initial configuration of the server.

1. The main interface of WebSphere Partner Gateway is browser-based. Log on to WebSphere Partner Gateway. Enter the following address in the address bar:

   `http://wpgv6hub:58080/console`

   The console component of WebSphere Partner Gateway in our environment runs on the server with host name wpgv6hub. The port number 58080 was configured during the installation. Refer to 5.4.2, “Installing the product code” on page 87.

2. A default company called Operator is set up. One user ID `hubadmin` belongs to this company, and its expired password is `Pa55word`. Enter this information in the logon section. Then click **Login**. See Figure 5-44 on page 105.

Note: You can consider the company Operator and users belonging to this company as the system owners of a WebSphere Partner Gateway installation.

Archived

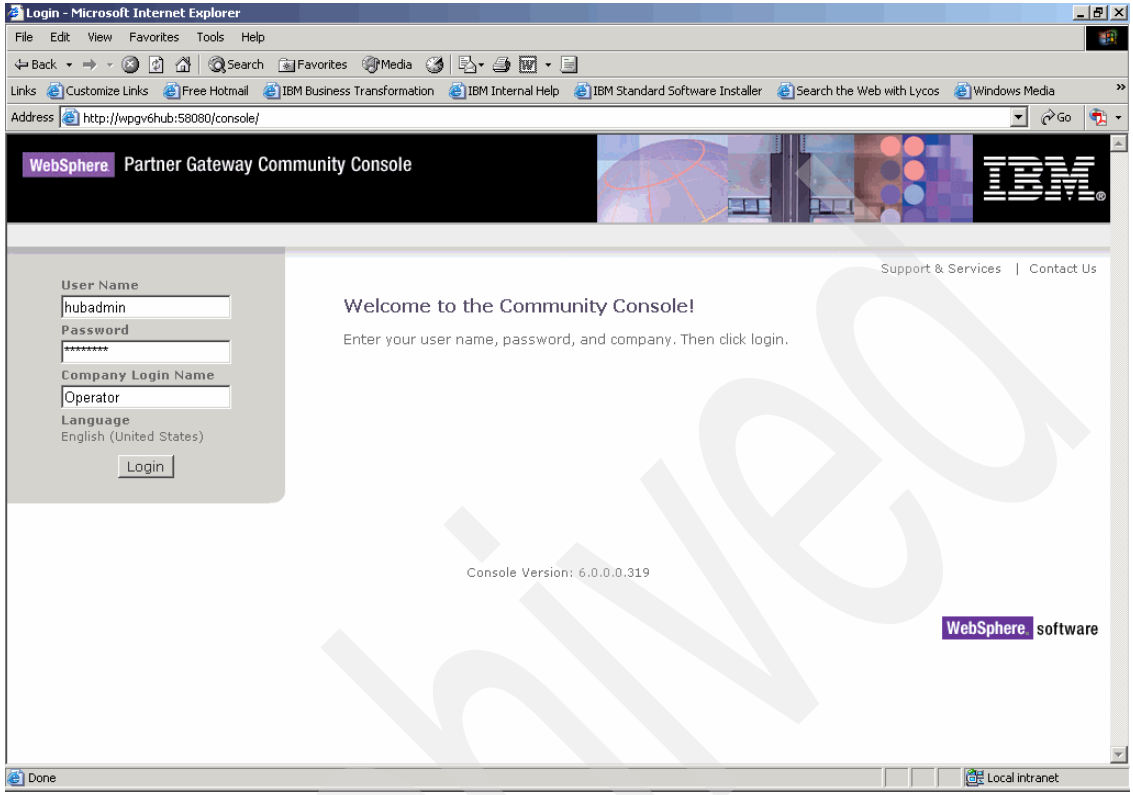


Figure 5-44 Community console login

3. Because the password is expired, you are forced to change it immediately. Enter a new password (Figure 5-45) and click **Save**.

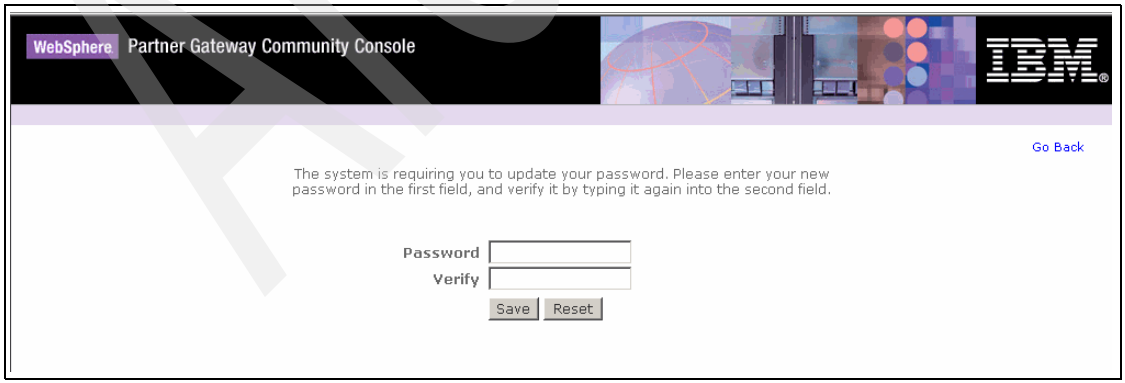


Figure 5-45 Change password

Archived

4. You see the main interface of WebSphere Partner Gateway. It consists of a three-level menu bar at the top. The bold items on each level of the menu tell you where you are currently working.

   The menu options depend on the profile of the current users. 7.2, “Role-based configuration” on page 138 explains how to add other users who have fewer options available in the menu.

   Under the menu, note the information for regional settings, such as language and time zone, as well as the name of the current user (Figure 5-46).

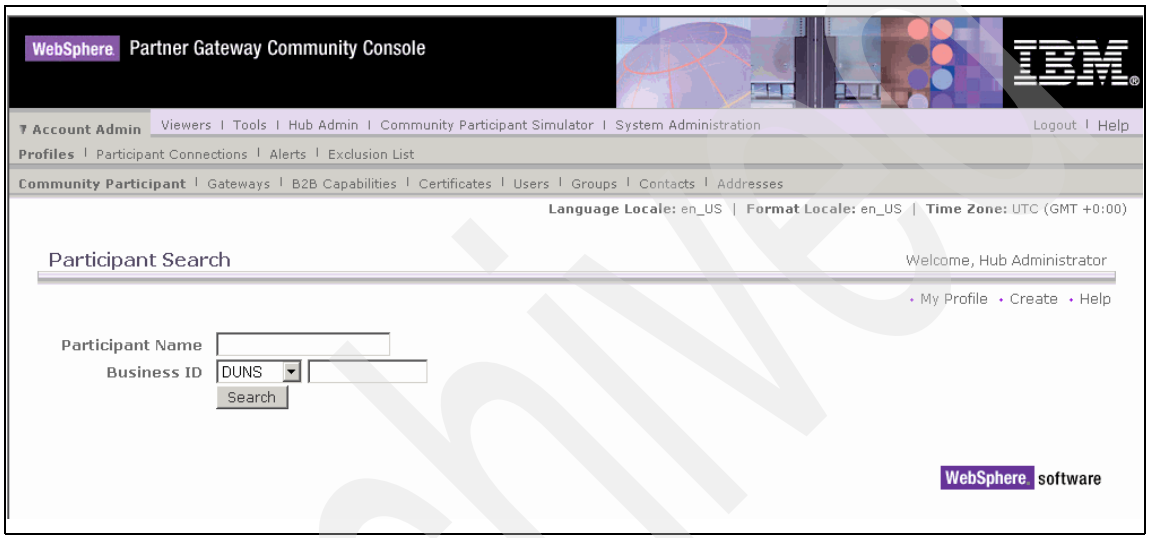


Figure 5-46 Main window

We explore and use most of the menu options in later chapters in this redbook.

The top-level menu option **Account Admin** has to do with managing accounts and profile information. Profiles exist for participants, gateways, and B2B capabilities. It is also possible, with this menu option, to add more users or groups to the current company. These additional users can be considered as additional system-level operators. The manager user of a community Participant should be the person who defines application-level users or business-level users.

If you select the top-menu option **Viewers**, you see the choice to view information about the system at several levels. The Event Viewer allows you to look for detailed information about ongoing events in the system. These events can be related to ongoing or past processing of documents or to actual system-related events. A higher-level viewer is provided for AS2 and RosettaNet. Here, you can search for information specific to AS2 exchanges. We use these

tools in later chapters to verify, for example, that a message disposition notification (MDN) has been received for a given document.

The Document Viewer, for which the search options are shown in Figure 5-47, allows the user to find documents independent of how these documents are exchanged. It is a great way for business-level people to determine the status of a document exchanged on a given date or time between two participants.

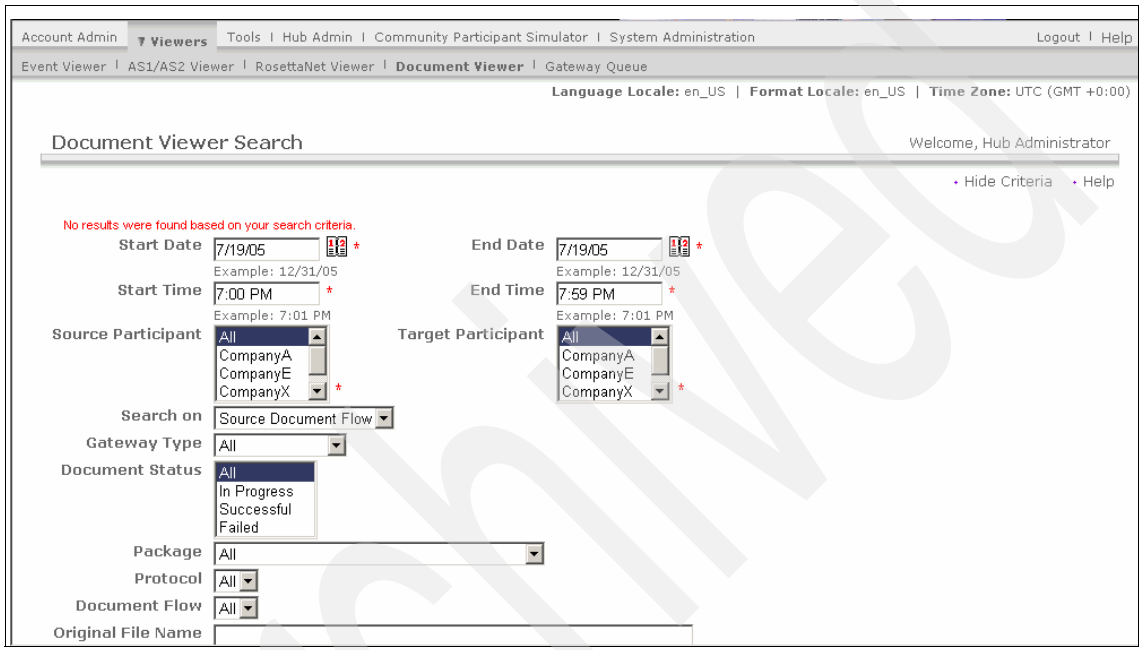


Figure 5-47 Document Viewer Search window

The **Hub Admin** menu option enables you to control system-wide settings and features. Under the submenu **Hub Configuration**, for example, you can find the option to manage XML formats and targets of document flows and document flow definitions themselves.

The submenu **Console Configuration** contains a few menu options that an administrator should review first before he or she performs any other configuration work. At this level, the hubadmin user can change locale information, permissions, and the password policy. Password policy settings are shown in Figure 5-48 on page 108. To make changes to the password policy, click the edit icon. After you enter the changes, click **Save**.

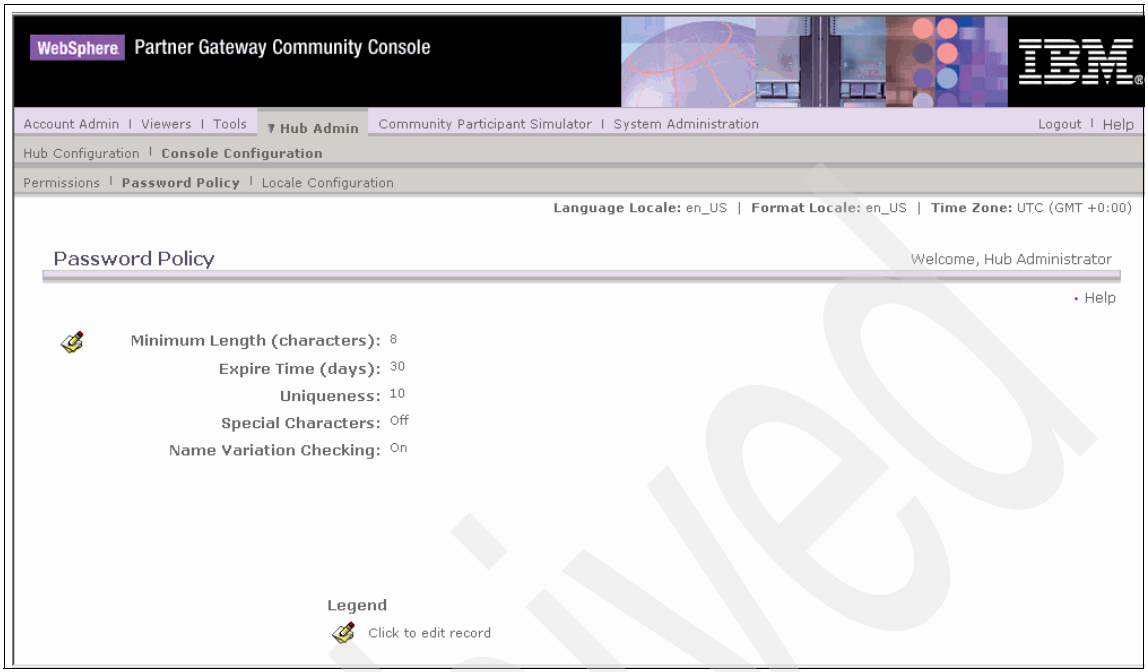


Figure 5-48 Review the password policy

More features of the product are demonstrated in later chapters. Chapter 6, “WebSphere Partner Gateway Advanced for AIX” on page 109 takes you through the implementation of WebSphere Partner Gateway Advanced on AIX.

Archived

6

WebSphere Partner Gateway Advanced for AIX

This chapter describes how to implement WebSphere Partner Gateway Advanced for AIX. It follows the same structure as Chapter 5, “WebSphere Partner Gateway Enterprise on Windows” on page 59.

Archived

© Copyright IBM Corp. 2005. All rights reserved.

6.1 Implementation Overview

In this chapter, all components including the database and MQ server are installed on the same physical AIX machine.

6.2 Verifying software levels on the AIX machine

As for any installation of a software product, you should understand the prerequisites for both the software and the hardware. You should also know how to verify that the prerequisites have been met.

WebSphere Partner Gateway relies on services and features of two other software products.

- A database such as DB2 or Oracle

  We use DB2 V8.2 in our environment.
- WebSphere MQ

  We use V5.3 with CSD8 in our environment.

The following instructions help you validate that these products are installed at the correct level and that the required features are available.

6.2.1 Verifying DB2

WebSphere Partner Gateway requires DB2 version 8.2 to be installed. To verify the version, open a shell and execute the **db2level** program as the database instance owner. The output of this program (see Example 6-1) tells you the specific DB2 information, as well as the FixPack level installed.

Example 6-1 Output of db2level command

```
# su - db2inst1
$ db2level
DB21085I  Instance "db2inst1" uses "32" bits and DB2 code release "SQL08021"
with level identifier "03020106".
Informational tokens are "DB2 v8.1.1.81", "special_13154", "U800400_13154", and
FixPak "8".
Product is installed at "/usr/opt/db2_08_01".
```

The DB2 code release level is `SQL8021` which shows DB2 is at the 8.2 level. It should be noted that many DB2 directories and signatures will still read as 8.1 and this can be confusing. A DB2 8.1 installation at or above FixPack 7 is considered DB2 V8.2.

6.2.2 Verifying WebSphere MQ

To verify the version of WebSphere MQ you can run the **mqver** command as the root or mqm user on the system. The output is shown in Example 6-2. CSD08 is required for WebSphere Partner Gateway. To obtain the required CSD, visit the following Web site which contains information about the latest CSDs :

http://www-306.ibm.com/software/integration/mqfamily/support/summary

Example 6-2 Output of mqver command

```
# mqver
Name:        WebSphere MQ
Version:     530.8  CSD08
CMVC level:  p530-08-L040921
BuildType:   IKAP - (Production)
```

Note: In previous versions of WebSphere Partner Gateway, the MQ Publish/Subscribe component needed to be installed seperately as a SupportPac™. CSD08 contains the Publish/Subscribe component, so no additional WebSphere MQ installation is needed.

6.3 Software Installation

This section explains how to install and configure WebSphere Partner Gateway Advanced. It includes these tasks:

- Adding the user accounts and group
- Configuring WebSphere MQ
- Installing the database component
- Installing WebSphere Partner Gateway

This section also covers a simple validation procedure to ensure that the WebSphere Partner Gateway product servers are up and running correctly.

6.3.1 Adding the user accounts

WebSphere Partner Gateway needs a set of operating system level users to function correctly. The names of these user IDs and group are free to choose, but it is recommended you use the default names.

For AIX, you can use the system management utility **smitty,** or use the following commands.

1. To add the group bcggroup, execute:

```
mkgroup -A bcggroup
```

2. To add the user bcguser that has bcggroup as its primary group, execute:

```
mkuser pgrp=bcggroup bcguser
```

3. To set the initial temporary password, execute:

```
passwd bcguser
```

Be aware that, by default, AIX has only set the temporary password at this point. To make the password for bcguser permanent, you should logon to the AIX system with Telnet or any other logon method. The system then prompts you to change the password. You can enter the same password if you choose.

Because this example is of a DB2 database installation, you need to also create userIDs for bcgcon, bcgrecv, and bcgdoc in the same manner as bcguser. An installation with an Oracle database does not require the additional users.

6.3.2 Configuring WebSphere MQ

There are several WebSphere MQ resources that need to be created and configured for WebSphere Partner Gateway. The instructions can be found in the product documentation Installation Guide, Chapter 2, section “Configuring WebSphere MQ” or use the following commands.

1. From a command line window as the root user, switch to the mqm user on the system as follows :

```
su - mqm
```

2. To create the queue manager, execute :

```
crtmqm -q partner_a.bcg.queue.manager
```

3. Now that the queue manager is created, you need to make some modifications to the qm.ini file that is associated with the newly created queue.

   Change to the directory where the new qm.ini file resides, with:

```
cd qmgrs/partner_a!bcg!queue!manager
```

4. Use an editor to modify the qm.ini file. It is good practice to change the following default log parameters to avoid potential process rollback errors :

   – `LogPrimaryFiles=62`
   – `LogSecondaryFiles=2`
   – `LogFilePages=2048`
   – `LogBufferPages=128`

5. At the bottom of the qm.ini file, add

```
Channels:
```

Underneath the Channels: stanza add two more lines that read:

- `MaxChannels=1000`
- `MaxActiveChannels=1000`

The adding of the channels parameters makes sure that WebSphere Partner Gateway can establish sufficient MQ client sessions when required.

Make sure that the file ends with a blank line, and save and close the file. Refer to Example 6-3 to see how the final qm.ini file should look:

Example 6-3 qm.ini file after modification

```
#*******************************************************************#
#* Module Name: qm.ini                                             *#
#* Type       : WebSphere MQ queue manager configuration file      *#
#  Function   : Define the configuration of a single queue manager *#
#*                                                                 *#
#*******************************************************************#
#* Notes      :                                                    *#
#* 1) This file defines the configuration of the queue manager     *#
#*                                                                 *#
#*******************************************************************#
ExitPath:
   ExitsDefaultPath=/var/mqm/exits/
#*                                                                 *#
#*                                                                 *#
Log:
   LogPrimaryFiles=62
   LogSecondaryFiles=2
   LogFilePages=2048
   LogType=CIRCULAR
   LogBufferPages=128
   LogPath=/var/mqm/log/partner_a!bcg!queue!manager/
   LogWriteIntegrity=TripleWrite
Service:
   Name=AuthorizationService
   EntryPoints=10
ServiceComponent:
   Service=AuthorizationService
   Name=MQSeries.UNIX.auth.service
   Module=/usr/mqm/lib/amqzfu
   ComponentDataSize=0
Channels:
   MaxChannels=1000
   MaxActiveChannels=1000
```

Archived

After the qm.ini file is successfully updated and saved, the next step is to set the the number of purchased licenses for the WebSphere MQ product. The number of required licenses depends on the number of processors in the AIX system.

1. In this example system we have one processor. From the command line (still as user mqm), execute :

   ```
   setmqcap 1
   ```

2. The final steps in the Websphere MQ configuration require that the queue be started. To start WebSphere MQ and the associated processes, execute the following commands :

   ```
   strmqm partner_a.bcg.queue.manager
   ```

3. Start the MQ listener. The recommended port for the listener is 9999. You can change this if you choose, but remember the port for the product installation steps later. Start the listener with :

   ```
   runmqlsr -t tcp -p 9999 -m partner_a.bcg.queue.manager &
   ```

4. To start the MQ broker, execute:

   ```
   strmqbrk -m partner_a.bcg.queue.manager
   ```

5. Start the MQ command services with :

   ```
   strmqcsv partner_a.bcg.queue.manager
   ```

There are two more commands needed to configure the MQ resources.

6. The first command refers to a definition file that is installed as part of the Java Messaging feature of Websphere MQ and contains definitions used by JMS Publish/Subscribe. This command is executed with :

   ```
   runmqsc partner_a.bcg.queue.manager < /usr/mqm/java/bin/MQJMS_PSQ.mqsc
   ```

   Note that `/usr/mqm` is the installation directory of WebSphere MQ and might be different on your system, depending where MQ was installed.

7. The second command defines a number of MQ resources that WebSphere Partner Gateway uses. It is available on the product CD (or in the uncompressed electronic image) of WebSphere Partner Gateway. Execute the command with :

   ```
   runmqsc partner_a.bcg.queue.manager <
   /cdrom/Tools/MQSeries/BCGCreate_Queues.mqsc
   ```

6.3.3 Installing the database schema

In addition to WebSphere MQ, WebSphere Partner Gateway relies on the services of a database manager. For this redbook, we used DB2. However, other database products are also supported. The database is used to store data that

can be classified in two broad categories: profile information and logging information.

The profile information is largely a read-only category when the system is configured. Changes occur only when new profiles are added or when an existing profile is updated, replacing an expired certificate, for example.

The logging information is a highly active category of information. A single document exchange results in many logging events, which are useful in keeping track of what happens in the system or what has happened in the system. Thus, this category of information expands constantly and is consulted mostly by online users and administrators.

Given the different nature of the database access to both categories of information, it comes as no surprise that WebSphere Partner Gateway has grouped them in two different database tablespaces so that it becomes easy to manage them separately. This can mean storing the information on separate disks and implementing different backup and reorganization procedures.

1. Start the product installation program from the product CD. To do this you need to execute the following command as the root user.

   ```
   # ./LaunchPad.sh
   ```

2. You are presented with several options, among which the two main tasks are:

   – Create the Database
   – Install the Product

   Select the **Create the Database** option to launch the database loader program (see Figure 6-1 on page 116). You must run this installation program as a root user. However, the database is created with the authority of the database instance owner, which is information that you need to provide during the installation.

Archived

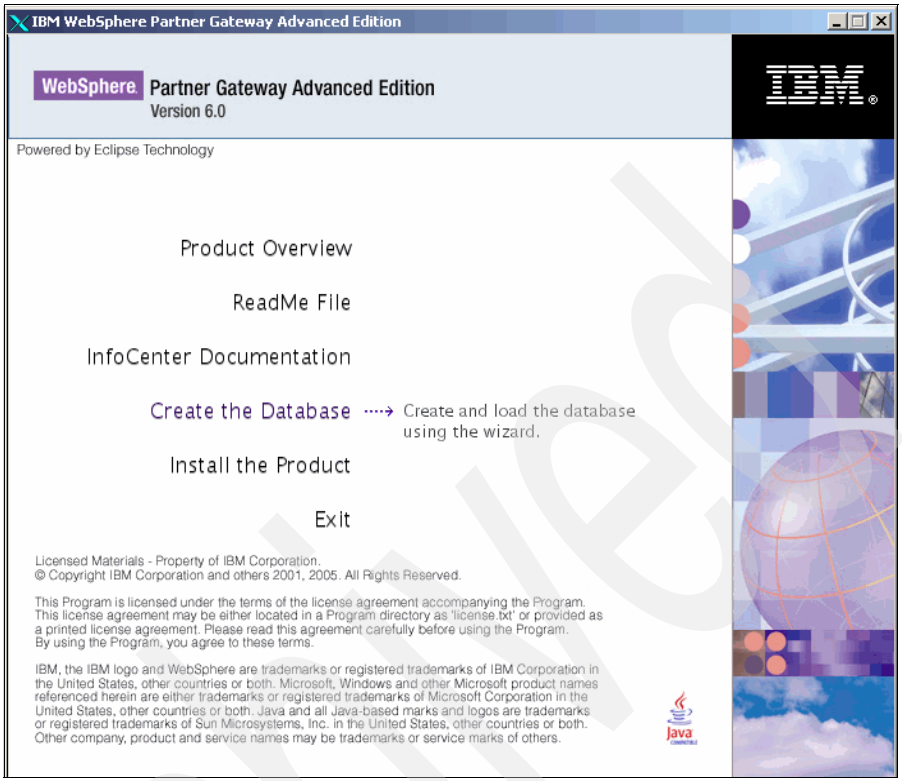


Figure 6-1 LaunchPad of WebSphere Partner Gateway

3. When the database loader program is started, the welcome window opens, click **Next**.
4. In the next window, accept the software license agreement. Click **Next**.
5. Provide an installation directory, for example `/opt/IBM/bcgdbloader`. See Figure 6-2 on page 117. Click **Next**.

Archived

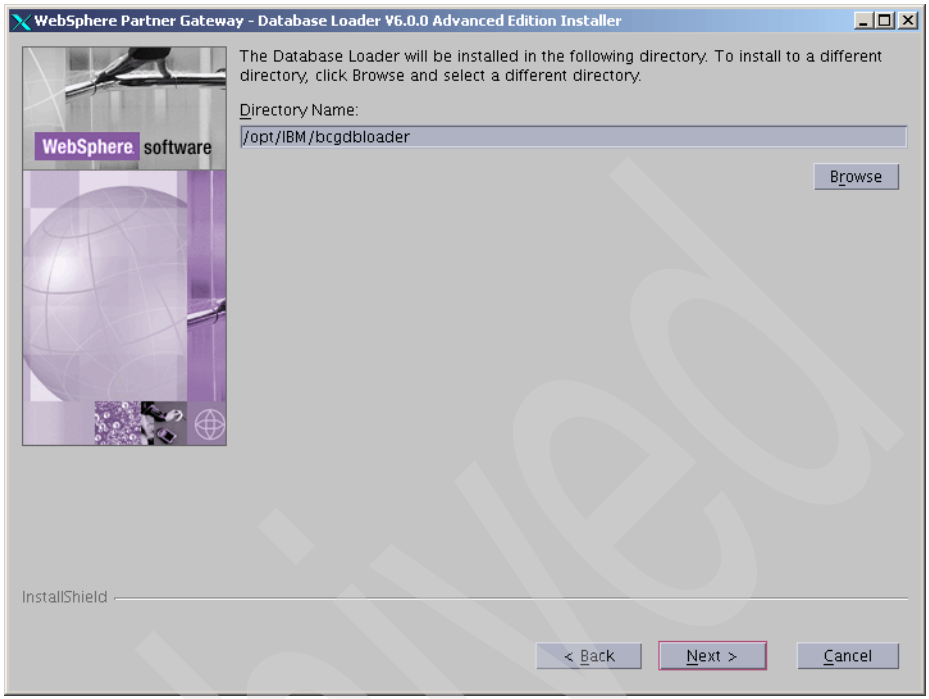


Figure 6-2 Selecting the installation directory of Database Loader

6. Select the database product that is going to be used. This can be :
   - IBM DB2 version 8.2
   - Oracle 9i 9.2.0.4 or later

   Click **Next**.

The following figures show the DB2 choices that we made for this redbook. The product installation guide explains the options for Oracle database.

7. In the Database Loader window that opens (see Figure 6-3 on page 118) enter the following information:
   a. The default name of the database is `bcgapps` and is entered already in the Database Name field.
   b. Next is the name of the database instance. A standard UNIX® installation of DB2, such as this one, has a default instance name of `db2inst1`. Other instances can also be created. This database instance *must* exist before you run the database loader program, and is usually done during installation of the database product.

Archived

c. Specify the name of the group to which the databse owner user ID belongs. On UNIX this is normally `db2grp1`.

d. Enter the user ID and password of the database instance owner.

e. Click **Next** to proceed.

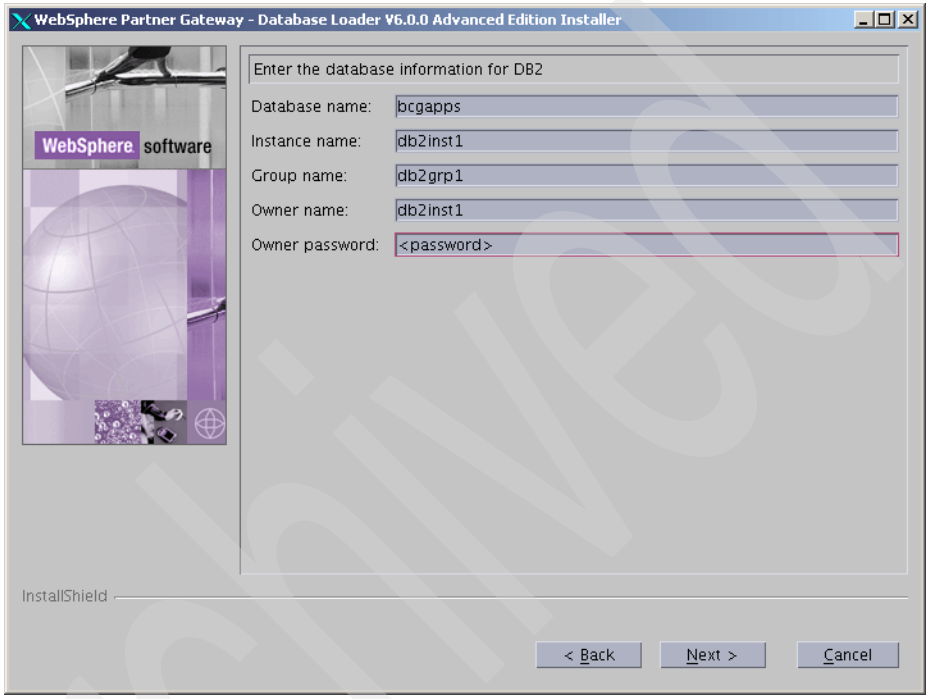


Figure 6-3 Database information

8. In the next window (Figure 6-4 on page 119), provide the location information to store the data. As explained earlier, the WebSphere Partner Gateway tables are grouped in categories that are mapped to tablespaces. For optimal performance, allocate directories of different disks for the main tablespaces. The presented default values are fine for an initial setup of the product. Click **Next**.

Note: You can use the database loader program so that the generated SQL statements are stored in a file and not executed automatically. That way, you can review them with your database administrator and execute the SQL statements manually.

Archived

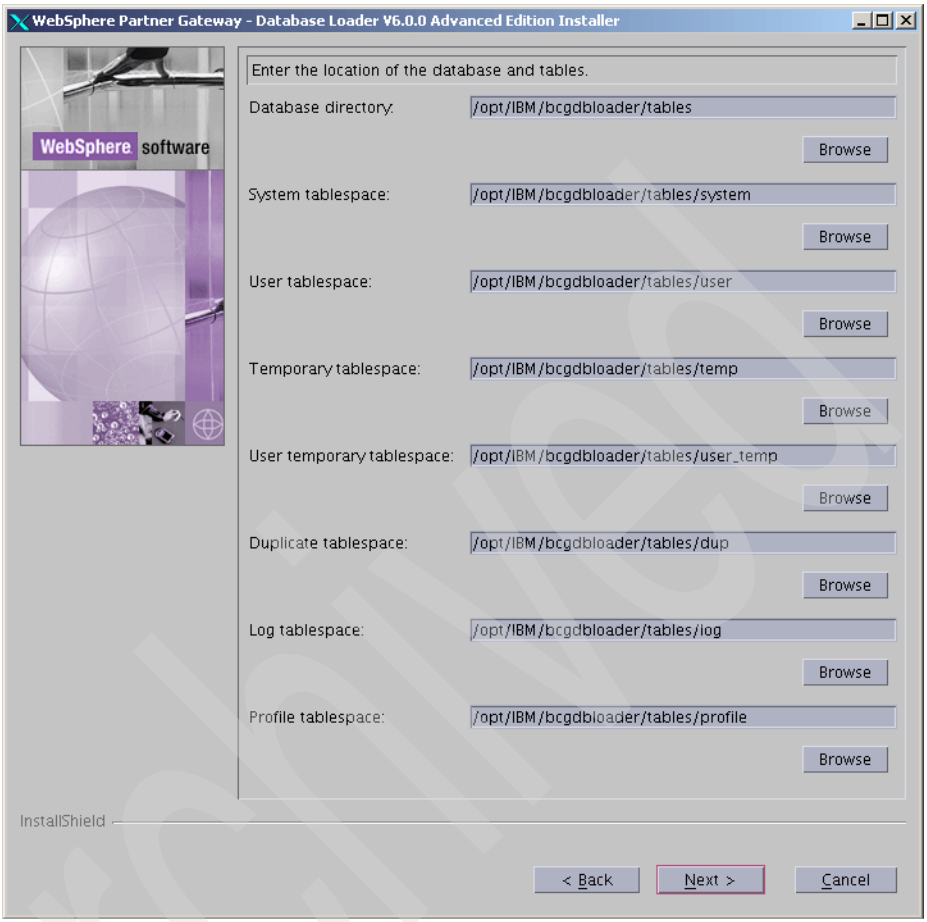


Figure 6-4 Tablespace location information

9. Provide the name of the WebSphere Partner Gateway group and user IDs and their passwords (Figure 6-5 on page 120). The values shown are the default values and match the values used in 6.3.1, “Adding the user accounts” on page 111. These user IDs are used to access the database.

   Click **Next** to continue.

Archived

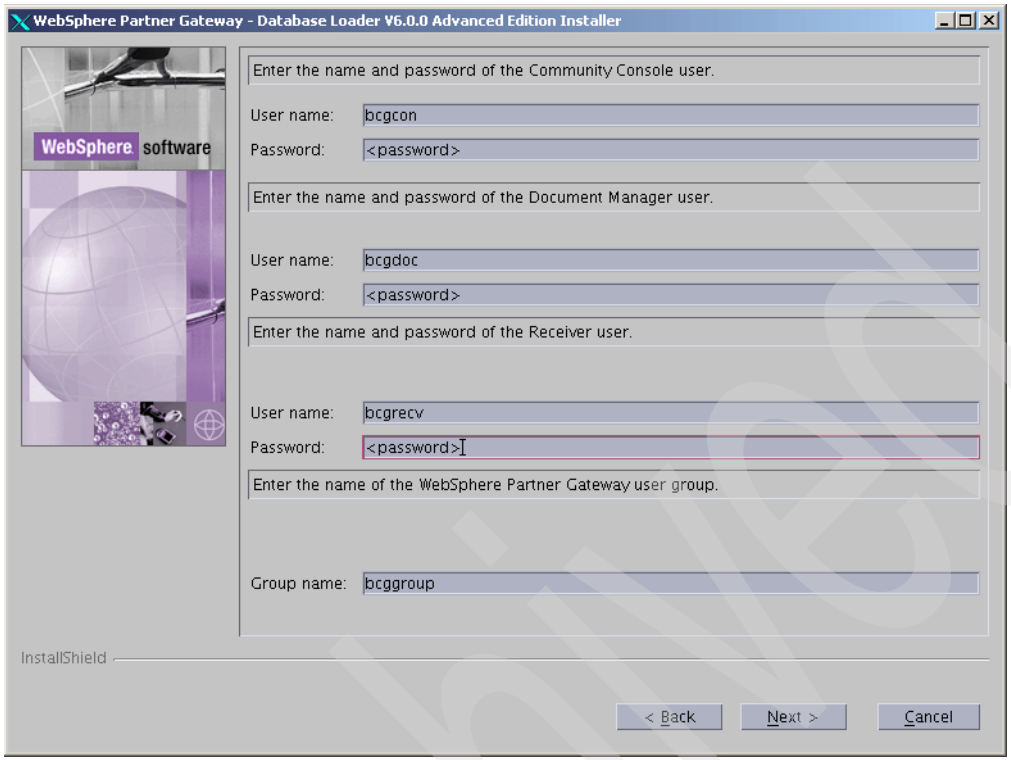


Figure 6-5 Providing user ID, password and group information

10. Name the mount point for the common data (see Figure 6-6 on page 121), which by default is `/opt/IBM/bcghub/common`,

    Click **Next**.

Archived

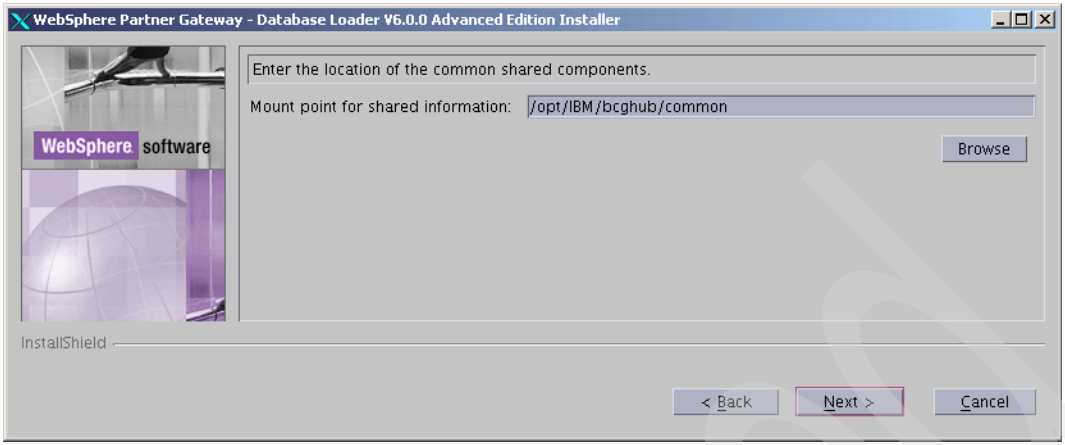


Figure 6-6 Setting the common data mountpoint

11. You can choose whether you want the database loader to execute the generated SQL scripts, or whether you want to run them manually after review by a databse administrator.

    You can find detailed instructions about how to run the scripts manually in the Instructions.txt file stored in /opt/IBM/bcgdbloader/scripts/DB2 directory. This text file also contains information about the process to delete the database, if you need to restart the process due to a failure, for example.

    Click **Next**.

12. Review the installation summary, and click **Next**.
13. Click **Finish.**

Before proceeding, it is recommended that you validate the Database Loader program installation. The databse loader program creates many objects, and the log files that are available in /tmp/logs/bcgdbloader are quite large. The following steps explain how to perform some simple validation. However, this check is inconclusive. You might still have to review the log files.

A quick validation can consist of the following tasks :

1. Connect to the database with each user ID.
2. Count the stored procedures that have been created.

Example 6-4 on page 122 shows the commands to perform this in a DB2 command session, executed while logged on as db2inst1. The number of stored procedures that should be created is 643.

Archived

Example 6-4 Simple Database Loader validation

```
db2 => connect to bcgapps user bcgcon
Enter current password for bcgcon:

   Database Connection Information

 Database server        = DB2/6000 8.2.1
 SQL authorization ID   = BCGCON
 Local database alias   = BCGAPPS

db2 => select count(*) from syscat.procedures where procschema='DB2INST1'

1
-----------
        643

  1 record(s) selected.

db2 => connect reset
DB20000I  The SQL command completed successfully.
```

6.3.4 Installing the Product Code

After you have installed the database schema, follow these steps.

1. The launchpad window started earlier should still be active. If not, restart it from the product CD or uncompressed electronic image. This time select the **Install the Product** (see Figure 6-7 on page 123).

Archived

Figure 6-7 Launching Product Installation

2. In the welcome window, click **Next.**
3. Accept the license agreement and click **Next.**
4. Provide the name of an installation folder, for example `/opt/IBM/bcghub`.
5. In the next window (Figure 6-8 on page 124), specify which components are going to run on this system. Because we are creating a single server in this example, we select all the components. Click **Next**.

Archived

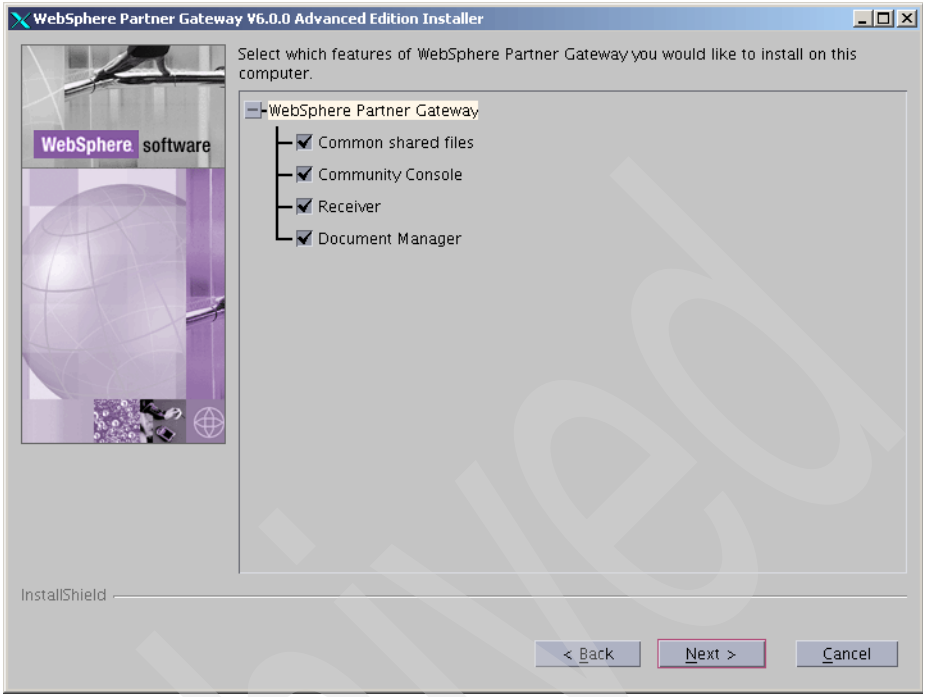


Figure 6-8 Selecting the product components to install

6. You are then asked to enter the fully qualified hostname of the machine running the WebSphere Partner Gateway product (see Figure 6-9 on page 125).

   Click **Next** to proceed.

Archived

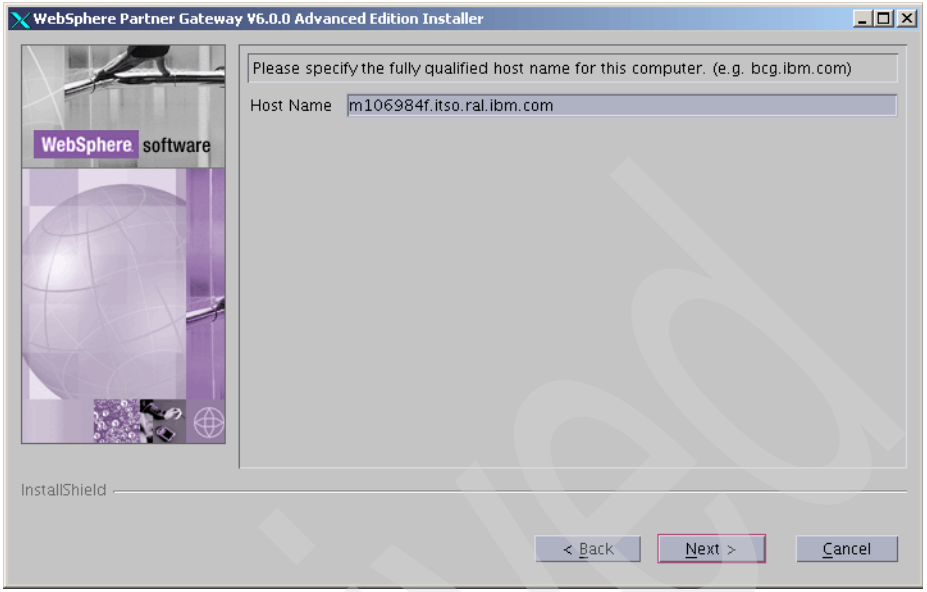


Figure 6-9 Entering the hostname

7. Select the version of WebSphere Application Server you would like to use. This option is primarily for users that already have an environment using a full version of WebSphere Application Server.

   You can install and use the embedded version of WebSphere Application Server or run on a full version that has already been installed. In this example, we use the embedded version because we do not have a currently running version of WebSphere Application Server on this machine. See Figure 6-10 on page 126.

   Click **Next**.

Archived

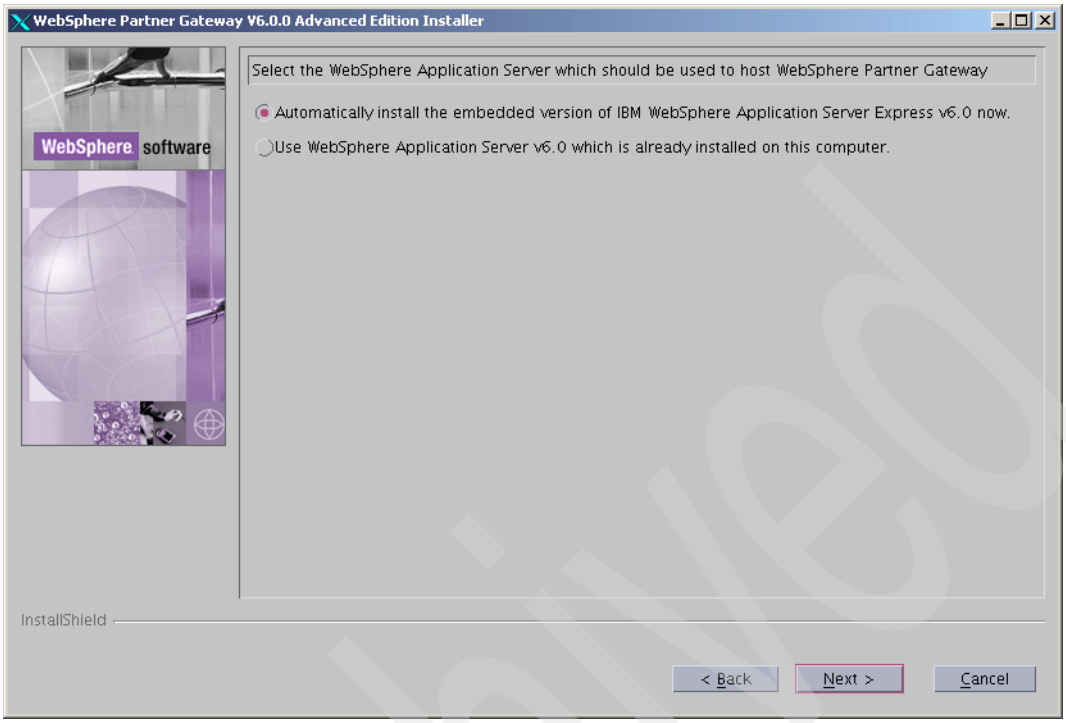


Figure 6-10 Websphere Application Server selection

8. Select the database product, either **IBM DB2** or **Oracle**, the click **Next**.
9. Based on your selection in the previous step, you are prompted for specific information about your selection. In this example for DB2, WebSphere Partner Gateway requests the following information, as shown in Figure 6-11 on page 127:
   - The hostname of the server
   - The port number used by DB2, which in most instances is `50000`
   - The owner name and password of `db2inst1`
   - The database name: `bcgapps`
   - The schema name: `db2inst1`

   Most of these values represent data you entered previously in the Database Loader program, so any changes in that installation will need to be reflected here.

   Click **Next** to continue.

Archived

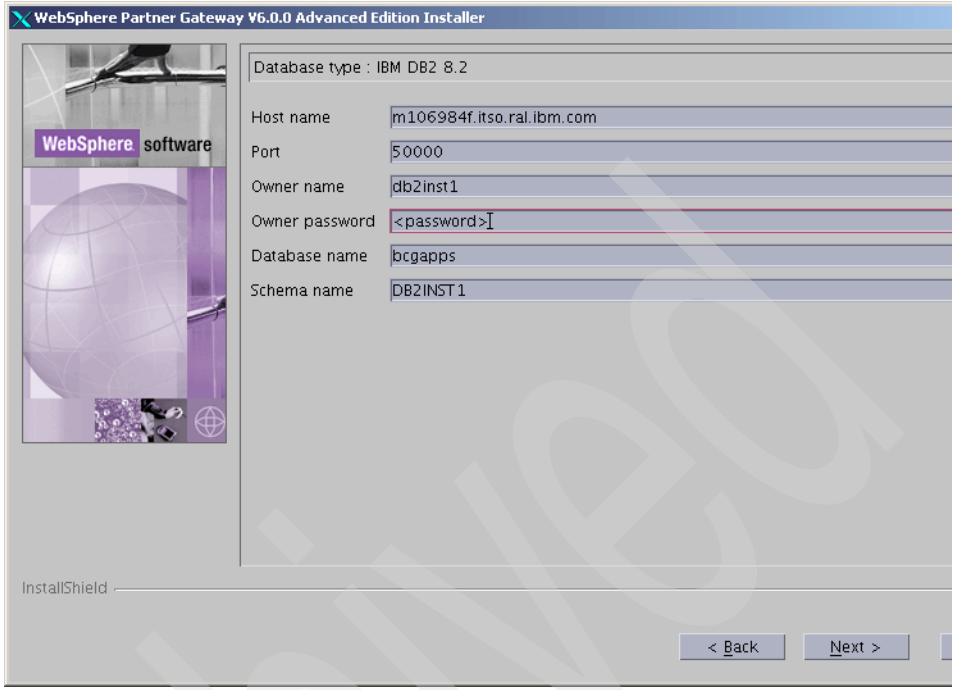


Figure 6-11 Providing Database server information

10. The installation program now connects to the database server and retrieves information that the Database Loader program created. Figure 6-12 on page 128 shows the result of these queries.

    This step is another simple validation step of the Database Loader program installation. Because the Product Install program is connecting to the database and retrieving information, this would be a likely place to receive an error if anything went wrong with the earlier Database Loader program.

    Click **Next** to continue.

Archived

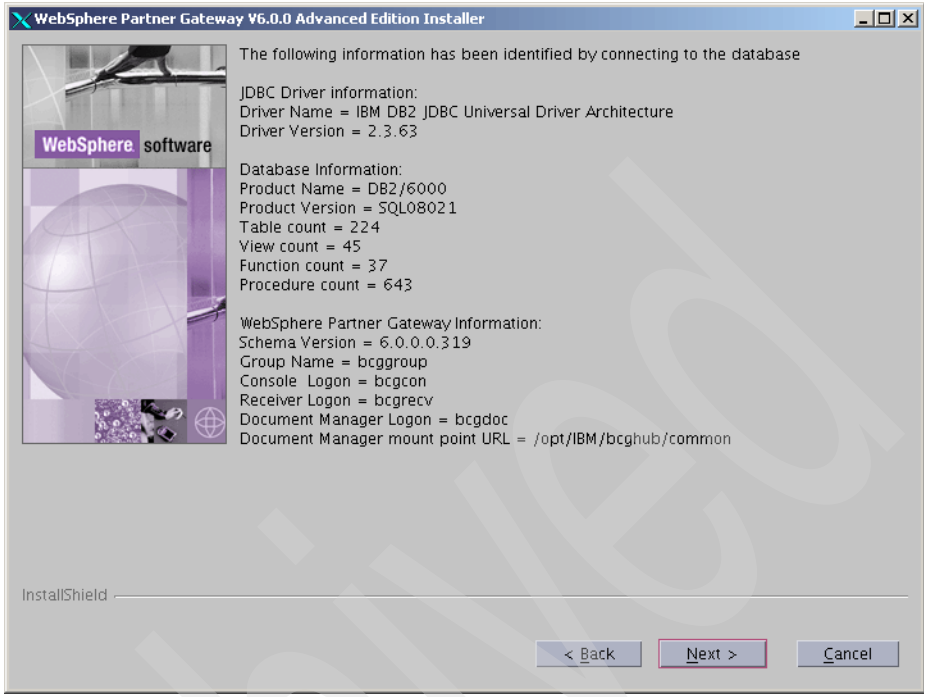


Figure 6-12 Retrieving information from the database

11. The installation program now asks for the user ID and password of the user who will own the product files and execute the processes, which is `bcguser` (Figure 6-13 on page 129). Click **Next** to proceed.

Archived

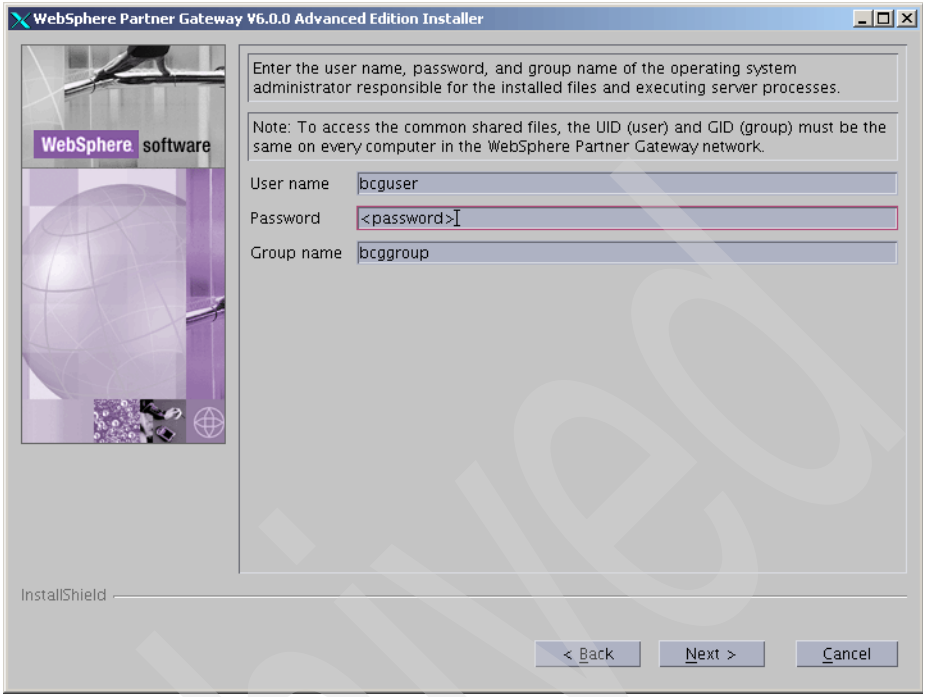


Figure 6-13 Providing user information for bcguser

12. The installation program now needs to know the location of the shared storage directory, similar to the Database Loader program did. See Figure 6-14 on page 130. This location must match the location entered earlier in the Database Loader Program in Figure 6-6 on page 121. Click **Next**.

Archived

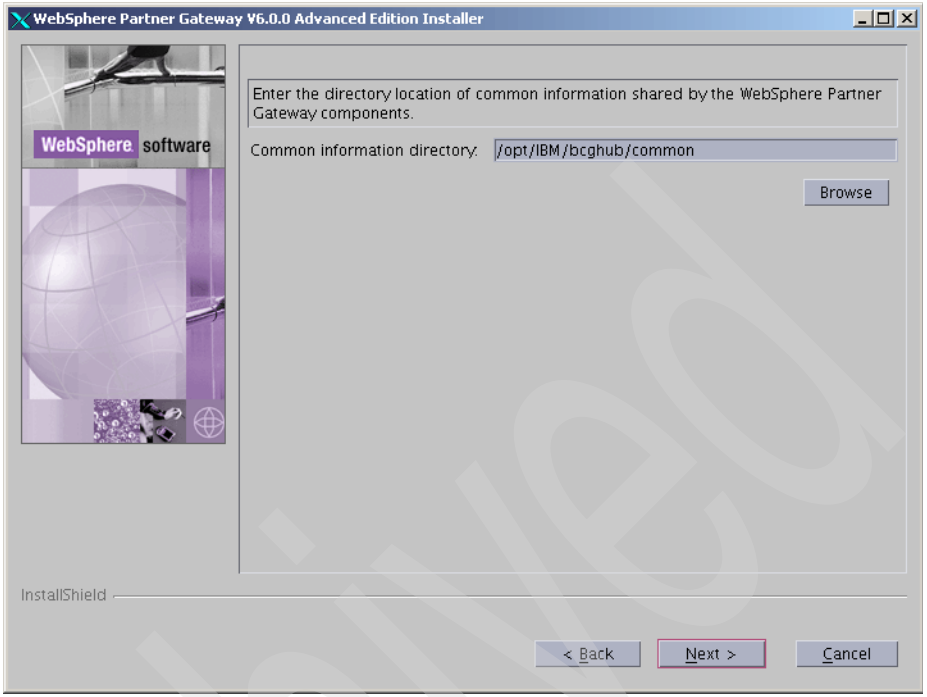


Figure 6-14 Common data location

13. Now that the installation program has verified that WebSphere Partner Gateway can access the database, it then requests information about WebSphere MQ (Figure 6-15 on page 131). The installation program needs:
    - The hostname of the MQ server
    - The name of the queue manager, which is `partner_a.bcg.queue.manager` in this example. This entry will have a default value that needs to be changed. Make sure it matches the name of the queue you created when configuring WebSphere MQ.
    - The port number used by the WebSphere MQ listener.

    When finished here, click **Next**.

Archived

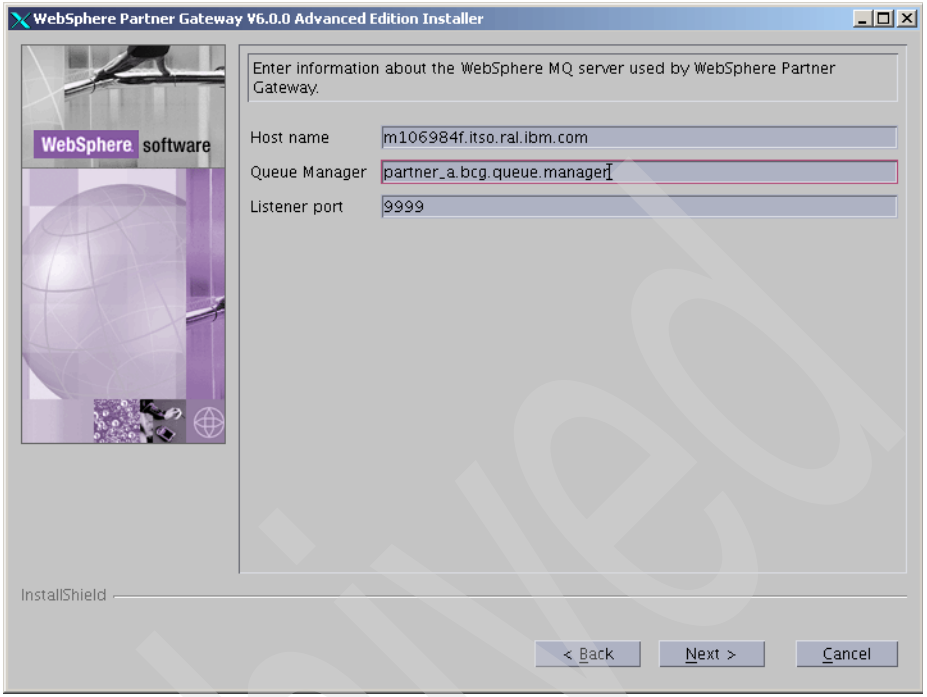


Figure 6-15 Providing MQ connection information

14. The next three windows prompt you for configuration parameters for each of the three components (bcgcon, bcgrecv, bcgdoc) In Figure 6-16 on page 132,you see these parameters for the Community Console component. A similar window will be next for the Receiver and then again for Document Manager components. The parameters that are requested are:
    - Name of the user ID that the component should use to access the database: bcgcon, bcgrecv, and bcgdoc
    - Password for that user ID
    - HTTP port to be used by the application server for this component
    - HTTPS port to be used by the application server for this component

> **Note:** The Community Console component shown in Figure 6-16 on page 132, has two additional parameters that Receiver and Document Manager components do not. These are Help System host name and Help System port number. These are parameters for the help system that runs on WebSphere Partner Gateway.

Archived

Click **Next** for each of the three windows when the above information has been entered correctly.

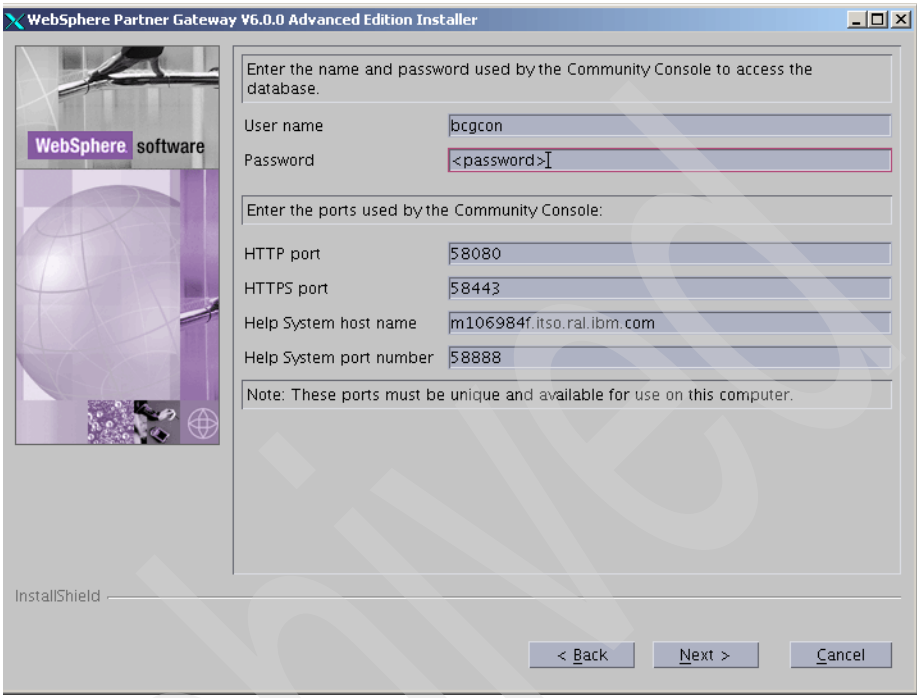


Figure 6-16 Providing information for the Community Console component

15. The next two windows, you are prompted to add contact information that will be included in RosettaNet and SMTP messages that are processed by WebSphere Partner Gateway. These will not be used for examples in this redbook, so default values are fine. Click **Next** on both of these windows.
16. You will now see a summary window. Clicking **Next** starts the installation.
17. When the installation has completed, you see a final window showing a successful installation message, click **Finish** to close the installation process.

6.3.5 Local Validation

The simplest validation that you can perform is to start the components and verify the contents of the log files. To start WebSphere Partner Gateway, you start the application servers from a command line.

Important: It is very important to start and stop the servers as user `bcguser` and not the root user. This can cause several problems in WebSphere Partner Gateway running on UNIX installations because file permissions are integral to the running of the product, and can be affected adversely by starting the servers as root.

To start the servers in WebSphere Partner Gateway, first change user to `bcguser` (for reasons noted above). Then navigate to the `/opt/IBM/bcghub/bin` directory. These steps are shown in Example 6-5.

Example 6-5 Changing user to bcguser and location of server startup directory

```
# su - bcguser
$
$ cd /opt/IBM/bcghub/bin
```

The commands to start the three application servers, and the output generated by them are shown in Example 6-6. The general format is `./bcgStartServer` followed by the name of the component you wish to start. The three component names are `bcgconsole`, `bcgreceiver`, and `bcgdocmgr`.

Example 6-6 Starting the application servers

```
$ ./bcgStartServer.sh bcgconsole
ADMU0116I: Tool information is being logged in file
/opt/IBM/bcghub/was/profiles/bcgconsole/logs/bcgconsole/startServer.log
ADMU0128I: Starting tool with the bcgconsole profile
ADMU3100I: Reading configuration for server: bcgconsole
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server bcgconsole open for e-business; process id is 21754
$
$ ./bcgStartServer.sh bcgreceiver
ADMU0116I: Tool information is being logged in file
/opt/IBM/bcghub/was/profiles/bcgreceiver/logs/bcgreceiver/startServer.log
ADMU0128I: Starting tool with the bcgreceiver profile
ADMU3100I: Reading configuration for server: bcgreceiver
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server bcgreceiver open for e-business; process id is 31230
$
$ ./bcgStartServer.sh bcgdocmgr
ADMU0116I: Tool information is being logged in file
/opt/IBM/bcghub/was/profiles/bcgdocmgr/logs/bcgdocmgr/startServer.log
ADMU0128I: Starting tool with the bcgdocmgr profile
ADMU3100I: Reading configuration for server: bcgdocmgr
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server bcgdocmgr open for e-business; process id is 31996
```

Archived

After starting WebSphere Partner Gateway for the first time, it is a good idea to check the log files. There are some errors that are not severe enough to cause a failure in starting of the application server, but that could become problematic later.

Each component has its own log directory. The directory has some dependency on what the product installation path is of course, but if you used the default path as we did in this example, then the three log directory locations are:

- `/opt/IBM/bcghub/was/profiles/bcgconsole/logs/bcgconsole`
- `/opt/IBM/bcghub/was/profiles/bcgreceiver/logs/bcgreceiver`
- `/opt/IBM/bcghub/was/profiles/bcgdocmgr/logs/bcgdocmgr`

In the three directories, the most helpful log file is the one named for the component. Example 6-7 shows the directory listing for the bcgreceiver component. The bcg_receiver.log should be empty after the initial start of the component. The SystemErr.log and SystemOut.log can also be helpful logs for debugging.

Example 6-7 bcgreceiver log file directory

```
$ pwd
/opt/IBM/bcghub/was/profiles/bcgreceiver/logs/bcgreceiver
$
$ ls -l
total 34
-rw-r--r--   1 bcguser  bcggroup          0 Jul 15 13:54 SystemErr.log
-rw-r--r--   1 bcguser  bcggroup      14781 Jul 15 16:10 SystemOut.log
-rw-r--r--   1 bcguser  bcggroup          0 Jul 15 13:54 bcg_receiver.log
-rw-r--r--   1 bcguser  bcggroup          5 Jul 15 13:54 bcgreceiver.pid
-rw-r--r--   1 bcguser  bcggroup          0 Jul 15 13:53 native_stderr.log
-rw-r--r--   1 bcguser  bcggroup          0 Jul 15 13:53 native_stdout.log
-rw-r--r--   1 bcguser  bcggroup       1880 Jul 15 13:54 startServer.log
```

It should be noted that the three component logs (bcg_console.log, bcg_receiver.log, and bcg_router.log) all located in their respective directories should be empty immediately after the initial start of the application servers. However, they can begin to collect error events as configuration and usage begins, but the events might be expected. For example, typing the console login password incorrectly after initial configuration causes a bcg_console.log event.

Generally, these three logs should be your first stop if you suspect any error activity in the installation, and can be very helpful in determining the problem.

6.3.6 Starting the help system

One final step that should be taken is to start the WebSphere Partner Gateway help system. This is slightly different than WebSphere Business Integration Connect 4.2.2 version when it was not necessary to start the help.

The Help Server in WebSphere Partner Gateway was separated out to make it possible to have the help pages and system running on a separate server. The default is to run on the same server as the console.

The help system is started similar to the application servers and is started from the same directory in /opt/IBM/bcghub/bin. Example 6-8 shows the command and its output.

Example 6-8 Starting the help server

```
$ cd /opt/IBM/bcghub/bin
$
$ ./bcgStartHelp.sh
Starting the IBM Eclipse Help System in the background...
IBM Eclipse Help System started.
Use bcgStopHelp.sh to stop the Help System.
$
```

Note: The help system should be started before logging into the WebSphere Partner Gateway console the first time so that the help links in the console work correctly. If for some reason it was not, then starting the help system will take effect after logging in and out of the console.

6.4 Initial configuration of the WebSphere Partner Gateway server

When the WebSphere Partner Gateway console server is running, the difference between an AIX instance of WebSphere Partner Gateway and a Windows-based installation of the product becomes small. The interface is always browser-based. Therefore in this chapter, we do not repeat the initial logon process on AIX.

Archived

Archived

7

Creating a basic B2B exchange

This chapter describes the required setup to enable a basic AS2 communication between the WebSphere Partner Gateway installation created in Chapter 5, “WebSphere Partner Gateway Enterprise on Windows” on page 59, and in Chapter 6, “WebSphere Partner Gateway Advanced for AIX” on page 109.

In this chapter, we limit the communication to the exchange of electronic data interchange (EDI) documents retrieved from a local file system and then stored on a local file system. As we progress through the book, we discuss adding additional source and target destinations between both partners in an exchange.

© Copyright IBM Corp. 2005. All rights reserved.

7.1 Scenario overview

This scenario entails working with the two WebSphere Partner Gateway instances that were implemented in the previous two chapters. The first is running WebSphere Partner Gateway Enterprise on Windows. The second instance is running WebSphere Partner Gateway Advanced on AIX.

This chapter explains how to implement the exchange of EDI documents in AS2 packaging between the two partners (Figure 7-1). For both partners, the source and destination of the EDI documents will be on their respective file systems.

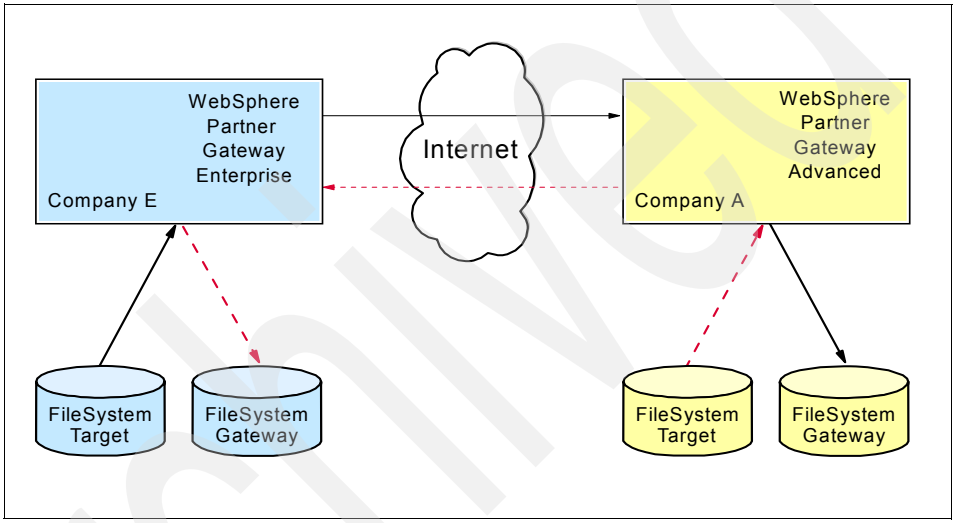


Figure 7-1 Scenario overview

7.2 Role-based configuration

When working with WebSphere Partner Gateway, you work with roles. A hub administrator needs to set up the profiles of the participants. A participant always has at least one administrator user and can have additional users added by the administrator for that profile.

To illustrate the concept of roles, suppose we have a simple implementation of WebSphere Partner Gateway that is ready for business. In this system, we have the minimum three profiles needed on the machine. They can be described as follows:

- Hub Operator Participant type

  This is a system defined profile that will be included on the machine directly from install. The Hub Operator profile has one defined user named hubadmin, that is the super-user of the system and can accomplish any configuration task. You can think of this role as the IT group that runs the actual WebSphere Partner Gateway server, but that is not actively sending documents back and forth. There can be only one Hub Operator type participant.

- Community Manager Participant type

  This Participant is created by the hubadmin user. You can think of this user as the company that bought the WebSphere Partner Gateway and is running the system. There should be only one Community Manager.

  It should be noted that some businesses act as both the Hub Operator and the Community Manager roles if they do not delegate the task of configuring and monitoring the WebSphere Partner Gateway system to some internal IT group or a third-party company.

- A Community Participant type

  This is the partner with which the Community Manager role wants to communicate. There can be multiple partners of this type of course. The Participant can have his own implementation of WebSphere Partner Gateway, but if he does he is the Community Manager on his own system, and a Community Participant on this one.

Each of these profiles has at least one user ID. As mentioned above, Hub Operator profile has the hubadmin super-user of the system. The other two profiles will each have an admin user assigned to them upon initial creation. These users, in turn, can create other users with equal or less abilities.

Each of these admin users has certain configuration abilities. For example, the the hubadmin user can create any object on the system such as the Community Manager itself or load system-wide security certificates. The Community Manager role can create participants or connections. The Community Participant role is the most limited in scope and basically only can view his own documents and configure the local destinations to which he wants the Community Manager to deliver documents.

This gives you two approaches when configuring WebSphere Partner Gateway. To promote self-management and teamwork between yourself and your partners you can use the role-based configuration and give users the ability to configure their end of the communication. Alternatively, if you are prefer to keep full control and maintain your own environment, the hubadmin user can be used to configure every aspect of the system.

Either approach is valid. In this example we demonstrate the different roles and how their abilities can be used. This example also discusses how the hubadmin user could be used to configure everything.

7.2.1 Outbound flow

For the outbound flow, an EDI document is stored for delivery in some shared file system directory that the WebSphere Partner Gateway can access. The arrival of the file is detected by the Receiver component of WebSphere Partner Gateway, and is retrieved from the file system and passed on the Document Manager component of the system.

The Document Manager component is where the business processing takes place. It parse the EDI document to find the source and target business identifiers in the document. These identifiers, which are part of the ISA segment (UNB segment for UN/EDIFACT or STX segment for TRADACOMS) of the EDI document, will need to be defined on WebSphere Partner Gateway.

Given the type of document (EDI) and the source and target business identifiers, the Document Manager then tries to find a defined connection between the two partners on the system. That means that the two participants on the system should have a connection between them that supports EDI documents. Or, in WebSphere Partner Gateway terms, the Document Manager reviews the B2B capabilities of the two participants. These B2B capabilities need to be defined for both the Community Manager and Community Participant.

Assuming the system finds a match between their B2B capabilities, the Document Manager then looks for a defined document flow that details what needs to be done to send the EDI document from the source partner to the target partner. This is called an *interaction*. In this interaction, you describe how an EDI document should be packaged in the AS2 envelope.

Finally, the Document Manager needs to know where the document is destined to go. This information is called the *gateway*. The target partner needs to have a gateway destination defined, which is an HTTP gateway in this scenario.

As we discussed in 7.2, “Role-based configuration” on page 138, there are three different users that can provide the information to configure the outbound flow (see Figure 7-2 on page 141).

- The hubadmin user is responsible for the correct operation of WebSphere Partner Gateway server. This person needs to perform the following tasks:
  - Define the location where the WebSphere Partner Gateway server will look for EDI documents that need to be sent out from the Community Manager (target definition).

- Define the interaction to package EDI documents in the AS2 format (document flow definition).
- Define the profile of the Community Manager who owns the server and include the business identifier in the profile.
- Define the profile of the Community Participant who is the recipient of the exchange and provide his business identifier in the profile.
- Define the connection between the two participants (participant connections).

► The Community Manager role will have an administrator user. This user will will have the responsibility to :

- Define the B2B capabilities of the Community Manager participant.

► The Community Participant role will also have an administrator user. Because the Participant is the target of this communication, this person must :

- Define the B2B capabilities of the Community Participant.
- Provide information about the target URI, called a gateway, which is the location he wants to receive the documents from the Community Manager.

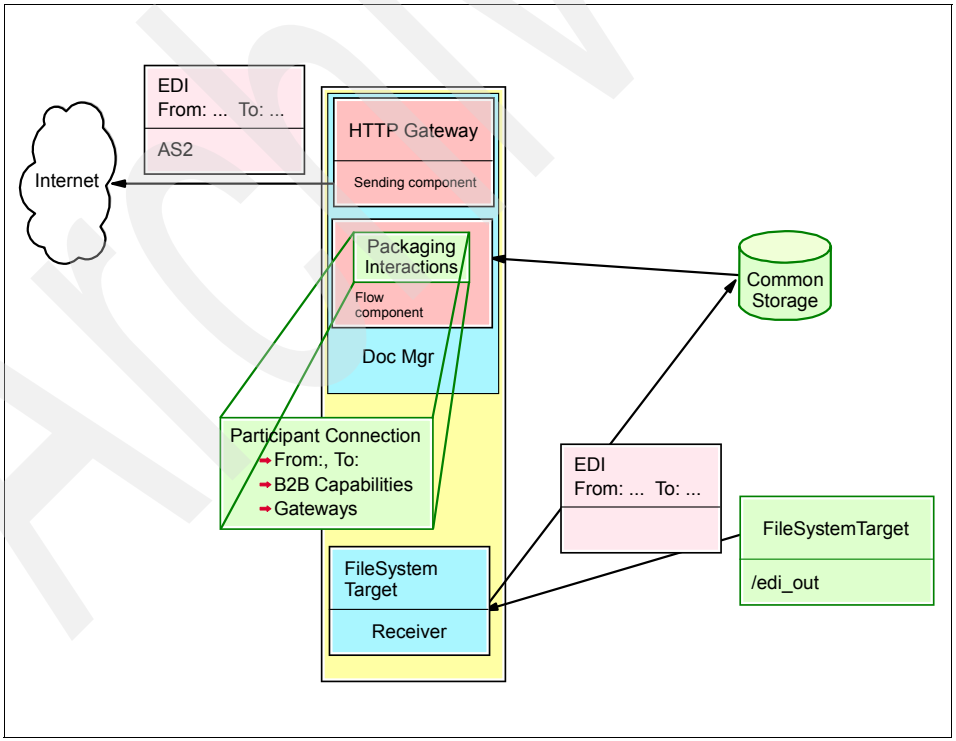


Figure 7-2 Outbound flow

Archived

7.2.2 Inbound Flow

The inbound flow is the logical opposite of the outbound flow. The outbound flow we described in 7.2.1, “Outbound flow” on page 140 is picking up an EDI file from the file system, packaging this document in AS2, and sending it outbound to the partner. We would also like to be able to do the reverse for inbound. That is, receive an AS2-packaged EDI document from the trading partner, unpackage the AS2 part of the document, and pass the remaining EDI document back to our own file system.

To accomplish this, we need our own URL where the server is listening for incoming HTTP data streams. This URL is called a *target* and needs to be defined on WebSphere Partner Gateway.

Upon arrival of the EDI document that is packaged in AS2 format, the Receiver stores the document in common storage. The Document Manager picks it up from the common storage and parses the document. Document Manager again looks for the business identifiers in that document and locates the connection between the two participants that are named in the EDI document.

If the connection is found, the WebSphere Partner Gateway server performs the interaction that is defined for the connection. This means the AS2 packaging is removed and the unpackaged EDI document passes to the gateway that was setup for the connection.

For the role-based configuration, there are three different users that can provide information to the WebSphere Partner Gateway for the inbound flow (Figure 7-3 on page 143):

- The hubadmin that is responsible for the correct operation of the WebSphere Partner Gateway needs to perform the following tasks:
  - Define the Community Manager profile and his business identifier, this has already been done for the outbound flow. The profile only needs to be created once.
  - Define the profile of the Community Participant who is the recipient of the exchange and provide his business identifier in the profile. Again this was done for the outbound flow. The profile only needs to be created once.
  - Define the location that the partner will send documents to, which will be an HTTP target.
  - Define the interaction that will be used to unpackage the AS2 EDI documents.
  - Define the connection between the two participants.

- The Community Manager role user still needs to perform these tasks:
  - Define the B2B capabilities of receiving an AS2 packaged EDI document.
  - Create the gateway to which the inbound file should be delivered, which is a file system gateway.
- The Community Participant who is the target partner in the outbound flow is now the role that is sending the inbound flow to the Community Manager. For inbound documents, he just needs to define B2B capabilities to send AS2 packaged EDI documents.

This chapter describes the steps each user of a defined role has to perform. The assumption is that the EDI documents will be sent in both directions. Therefore in the discussion about the configuration, you learn how to set up dual communication between the two partners without necessarily making a distinction between what is required for only outbound or inbound flows.

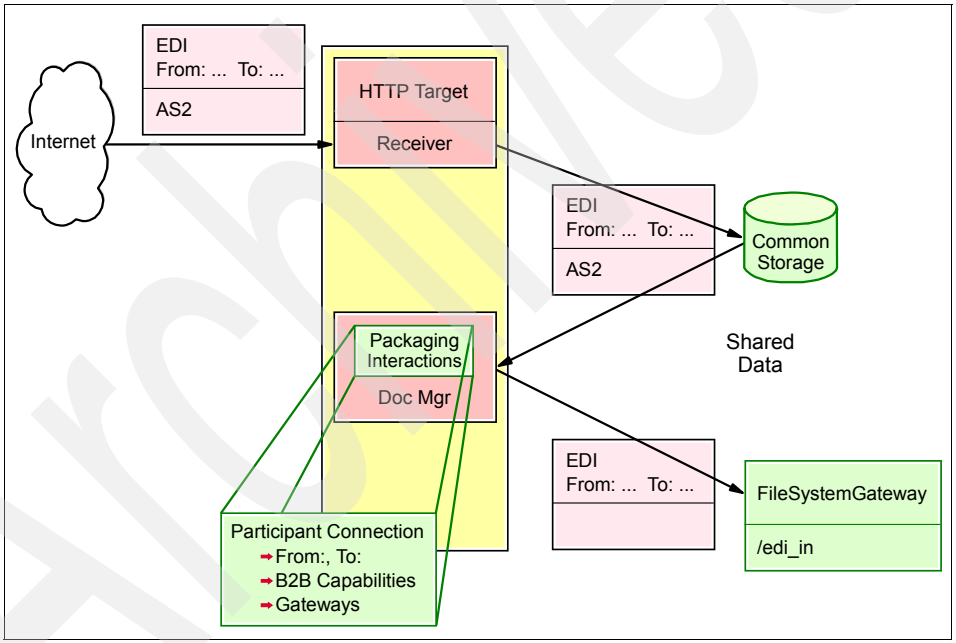


Figure 7-3 Inbound flow

7.3 Configuration tasks for hubadmin of Company E

The hub administrator has the responsibility to look after the overall configuration of the WebSphere Partner Gateway server. This user has the task to create the profiles of the Community Manager and the Community Participant. The hub

administrator also need to handle configuration tasks that involve global settings which would be applicable to all participants in the communications.

Figure 7-4 shows the scenario we intend to configure.

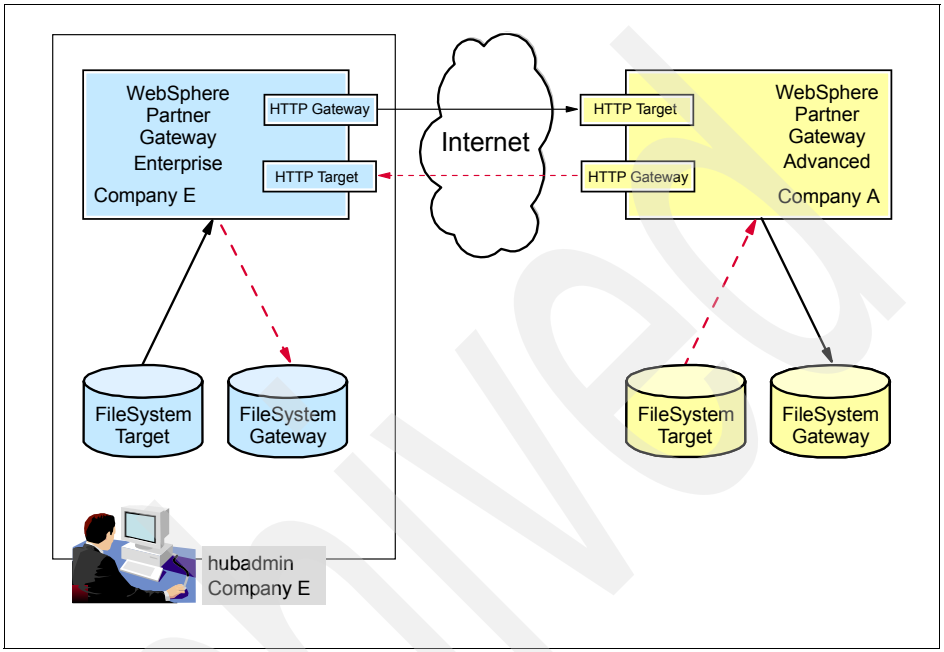


Figure 7-4 Logging on as hubadmin to server of Company E

7.3.1 Creating targets

Targets are the entry points into the WebSphere Partner Gateway server. These *entry points* can be a file directory, a queue, an HTTP URI, or some other type of resource. *Targets* are the entry points into the server regardless of their final destination. For example, the File System Target for company E in Figure 7-4 is the target that picks up a file from a local file system when the file is going outbound to the partner. The HTTP Target for Company E is the target that is externally available, and is awaiting files from Company A.

Therefore, Company E needs two targets for this dual communication scenario. You create them with these steps:

1. Log on to the hub administrator (User Name is `hubadmin`, Company Login Name is `Operator`) From the WebSphere Partner Gateway console menu select **Hub Admin** → **Hub Configuration** → **Targets.**

2. The browser displays a list of currently defined targets (Figure 7-5), which is currently empty. Click **Create Target**.

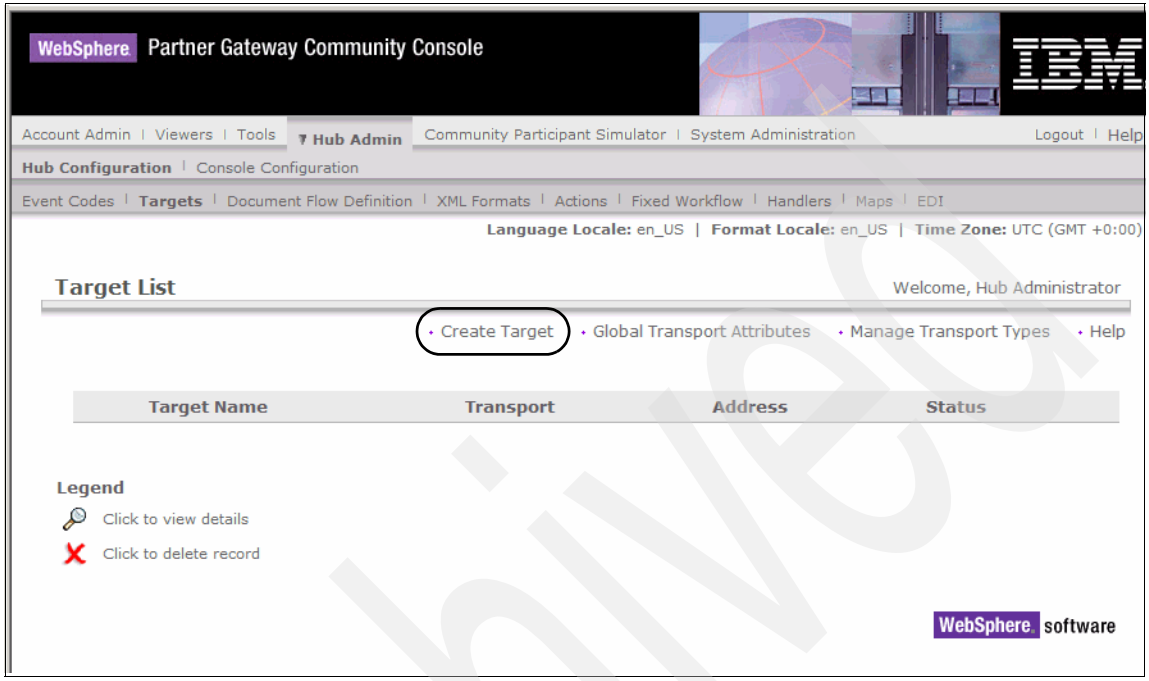


Figure 7-5 List of targets

3. In the Target Details window, follow these steps:

   a. Enter a name, and for the transport, select **HTTP/S.**

   A target is not really linked to a specific community participant. You can use specific names to make it clear why a certain target was created. Throughout this redbook, we make the name explicit to assist you in understanding where an object was defined and for what purposes.

   After you select a transport, the interface displays additional properties that are specific to that transport selection.

   b. Specify whether this target will be used for test or production purposes.

   c. An HTTP/S type target requires a URI to be provided. This URI must start with `/bcgreceiver`, but can be named freely after that.

   d. Click **Save**. See Figure 7-6 on page 146.

Archived

Figure 7-6 Creating an HTTP Target

4. After the new target has been created, click **List** to return to the list of targets (Figure 7-7 on page 147).

Archived

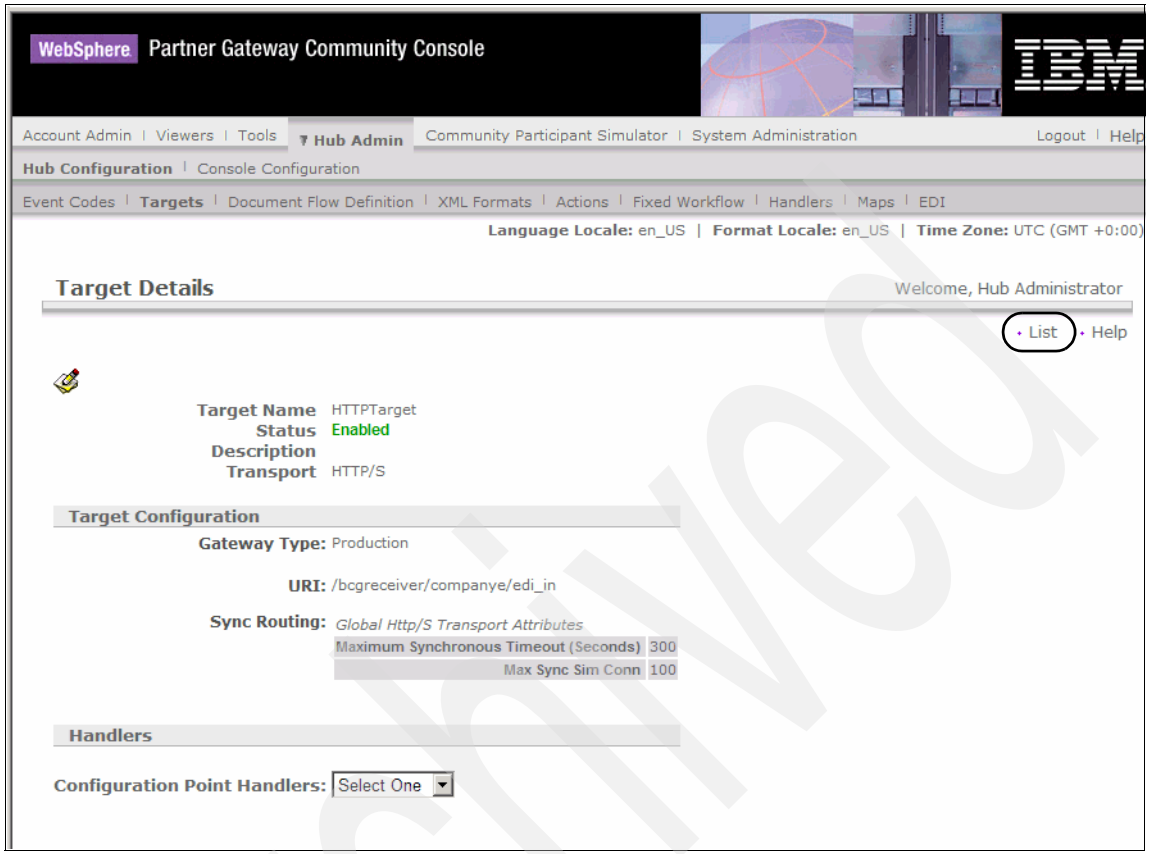


Figure 7-7 HTTP Target created

5. When you see the list of targets, click **Create Target** again to create a second target, this time for picking up the EDI documents locally that are to be sent out to the partners of Company E.
6. In the Target Details panel (Figure 7-8 on page 148), complete these tasks:
   a. Provide a name for the target.
   b. For Transport, select **File Directory**. The name of the directory can again provide an indication of its usage, but it is not required. Note that the WebSphere Partner Gateway will create the final directory edi_out on the path, but the other directories will have to exist before the target is created.
   c. Optionally, you can provide values to control the polling frequency of this directory. Also increasing the number of threads can be useful for systems where many files are being processed simultaneously.
   d. Click **Save** to store this new target to the database.

Archived

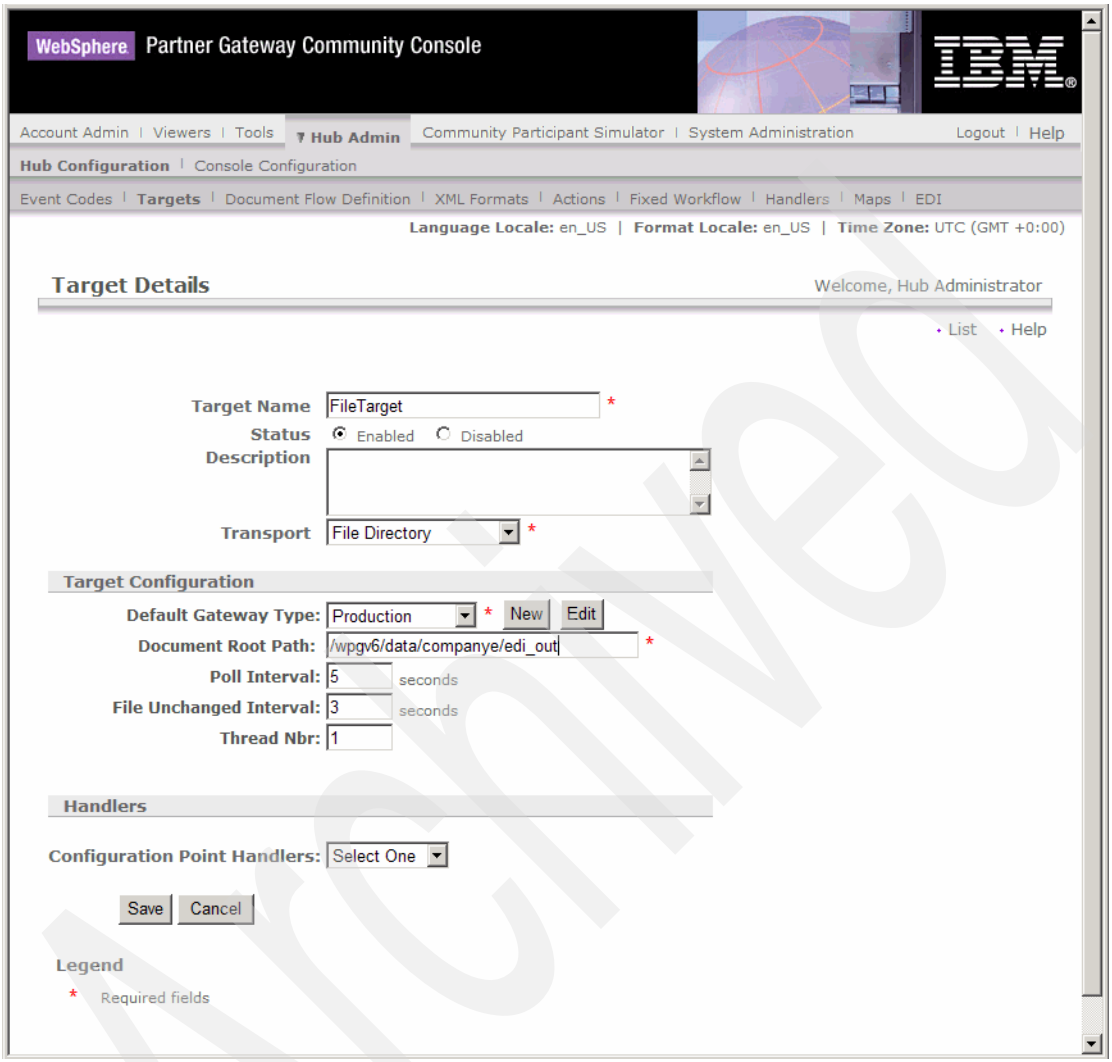


Figure 7-8 Creating a File Directory type target

This completes the creation of the HTTP Target that is needed to receive files from Company A and that is depicted in Figure 7-3 on page 143. It also created the FileSystemTarget that is needed to pickup files from the local directory of Company E and that is depicted in Figure 7-2 on page 141.

7.3.2 Creating interactions

Document flow definitions are the definitions of standards of the many types of formatting, protocols and documents that WebSphere Partner Gateway can understand. There are several levels of document flow definitions, such as Package level, Protocol level, Document Flow level, and even lower levels such as Action level and Signal Level. A *Package level definition* defines an enveloping standard such as AS2, where as a *Protocol Level* might define a standard such as EDI-X12. A *Document flow level definition* can define specifics or variations of the protocol that you may see at the document level.

The *interaction* that we want to create is a combination of the document flow definitions and the action that exists between them. The action can be something simple such as Pass-Through, or more complex, such as transformation, validation, or several actions simultaneously. The more complex actions are presented later in this redbook.

For now, we need to create two interactions on WebSphere Partner Gateway of Company E. The first needs to be able read an AS2 package and extract an EDI document. The second needs to be able to take an EDI document and package it in an AS2 format.

To create these interactions, follow these steps:

1. Select **Hub Admin** → **Hub Configuration** → **Document Flow Definition** (Figure 7-9 on page 150).

> **Note:** The hub administrator is creating the interaction for the entire system. Later in this chapter when you act as the Community Manager or Community Participant administrator, you activate the B2B capabilities for those profiles that essentially makes this interaction available to them for use.

2. By default, there are several Document Flow Definitions loaded on the system already. If you click the folder icon next to a package name, you see the protocols that are available for the package underneath. This will tell you if WebSphere Partner Gateway server has the document flow definitions that you need to create the interaction. For this example, AS Package already exists. Underneath that, you can see that Protocol : EDI-X12 and Document Flow : ISA also exists, so no action is needed on this particular page.

   There are examples later in this book where will you will create your own Package, Protocol, and Document flow Definitions.

   For now, click **Manage Interactions** to see the currently defined interactions.

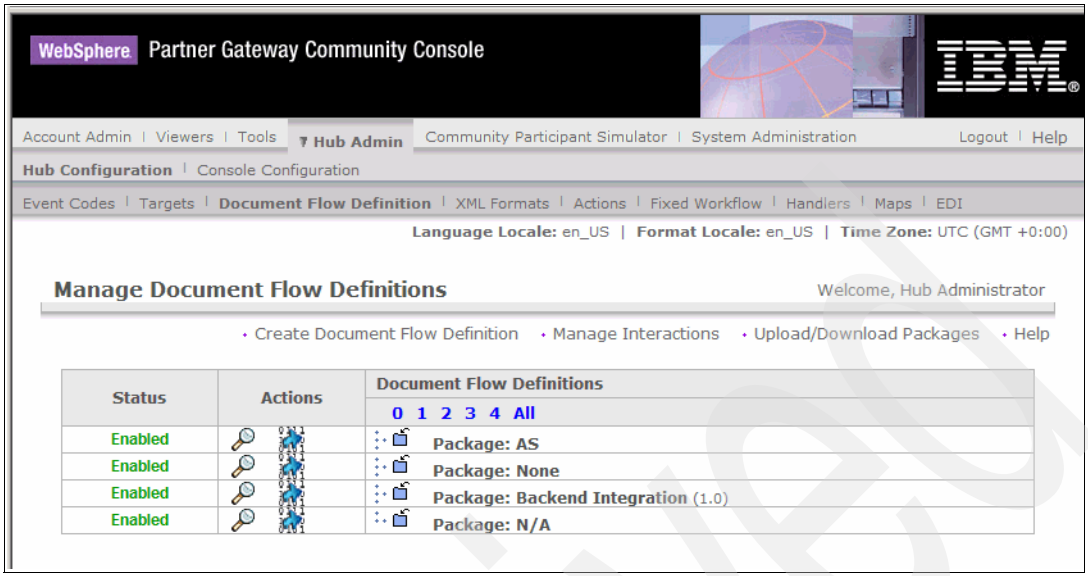


Figure 7-9 List of loaded document flow definitions

3. In the Manage Interactions panel (Figure 7-10 on page 151), click **Search** to see the interactions that currently exist. For this example, we would like to see a Package:None, Protocol:EDI-X12,Document Flow:ISA for the source side and Package:AS, Protocol:EDI-X12, Document Flow:ISA for the target side. The other interaction we need is the same with the source and target reversed. Remember for our outbound flow we want to pick up a plain EDI document from our local file system, which is what None packaging is. It means there is no transport packaging for the Document Manager to deal with at that point.

   The four interactions that are predefined deal with AS packaging of binary documents, and Back-end Integration packaging with binary documents. Back-end Integration is essentially JMS packaging for a back-end queue.

Because our desired interactions do not exist yet, we need to create them. Click **Create Interaction** to add the first one.

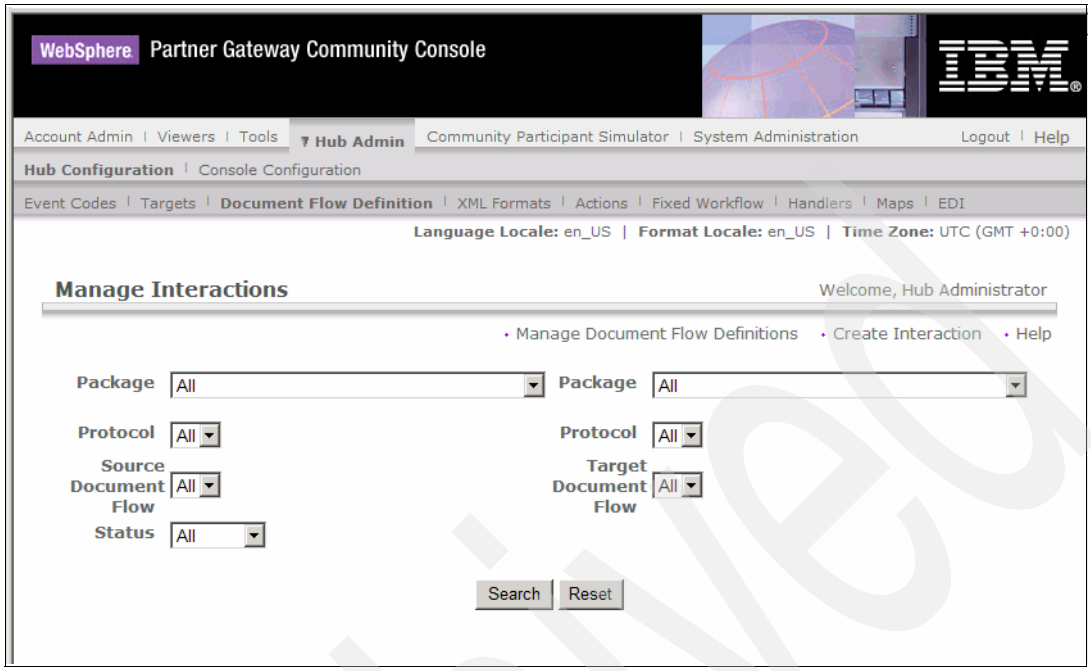


Figure 7-10 Searching for interactions

4. Figure 7-11 on page 152 shows the starting point to create a new interaction. You see the existing packages under the headings Source and Target. For our example, we need the following interactions:
   - AS2 packaged EDI documents to unpackaged EDI documents for the inbound flow
   - Unpackaged EDI documents to AS2 packaged documents for the outbound flow

   To do this, follow these steps:

   a. For inbound flow, under Source click the folder icon next to **Package: AS**.

Archived

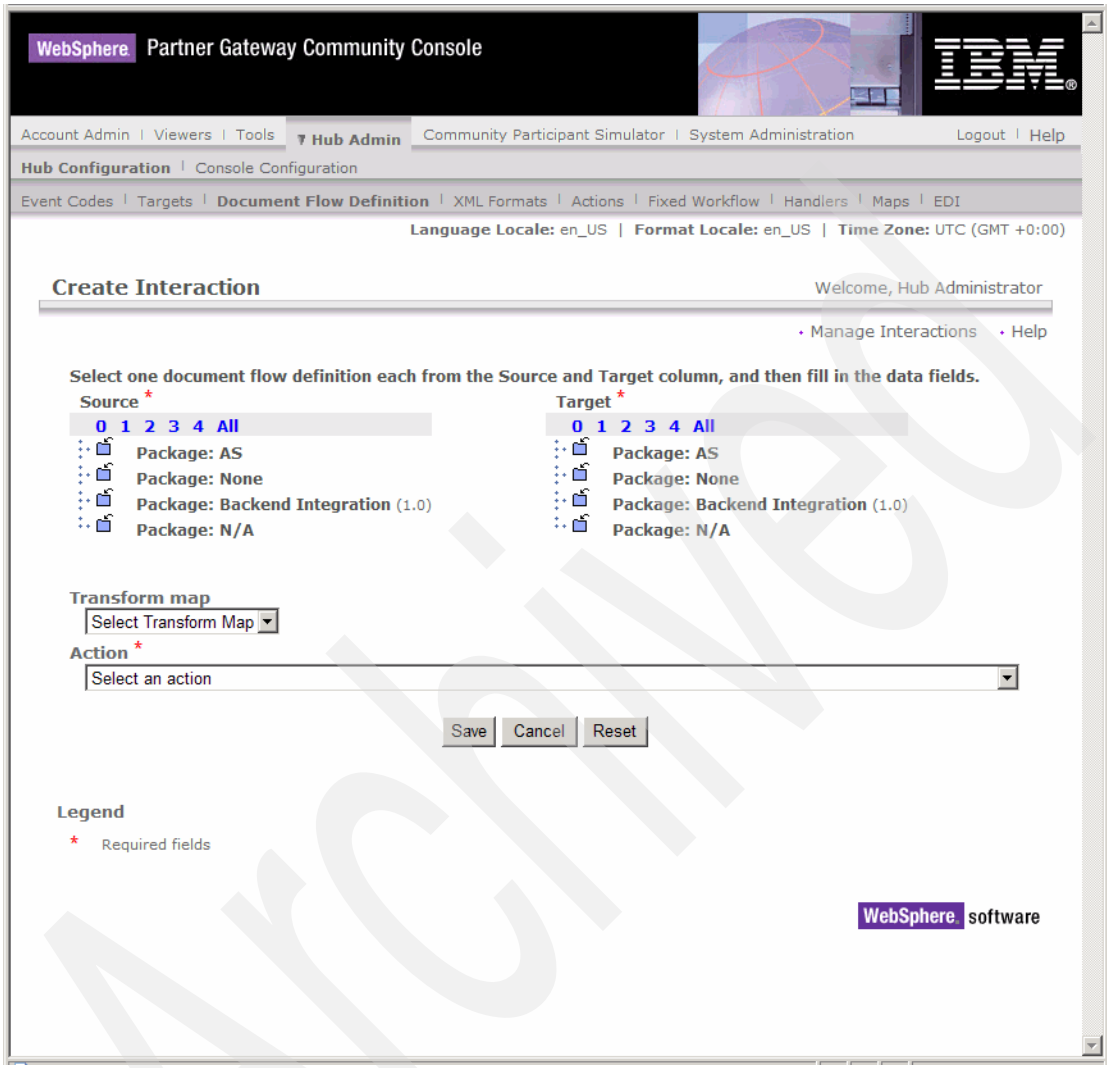


Figure 7-11 Create interaction start point

b. Now you see several protocols (Figure 7-12 on page 153), such as EDI-X12. Click the folder icon next to **Protocol: EDI-X12**. Then select the radio button for **Document Flow: ISA**.

c. Under Target, click the folder icon next to the **Package: None**. Under that click the folder icon next to **Protocol**: **EDI-X12** and select the radio button for **Document Flow**: **ISA**.

d. For Action, select **Pass Through**.

e. Click **Save**.

There are several actions predefined on the system. These are related to Custom XML, Rosetta Net, transformation and validation. You can also define your own actions. For this example, we are just passing the document through with no additional processing.

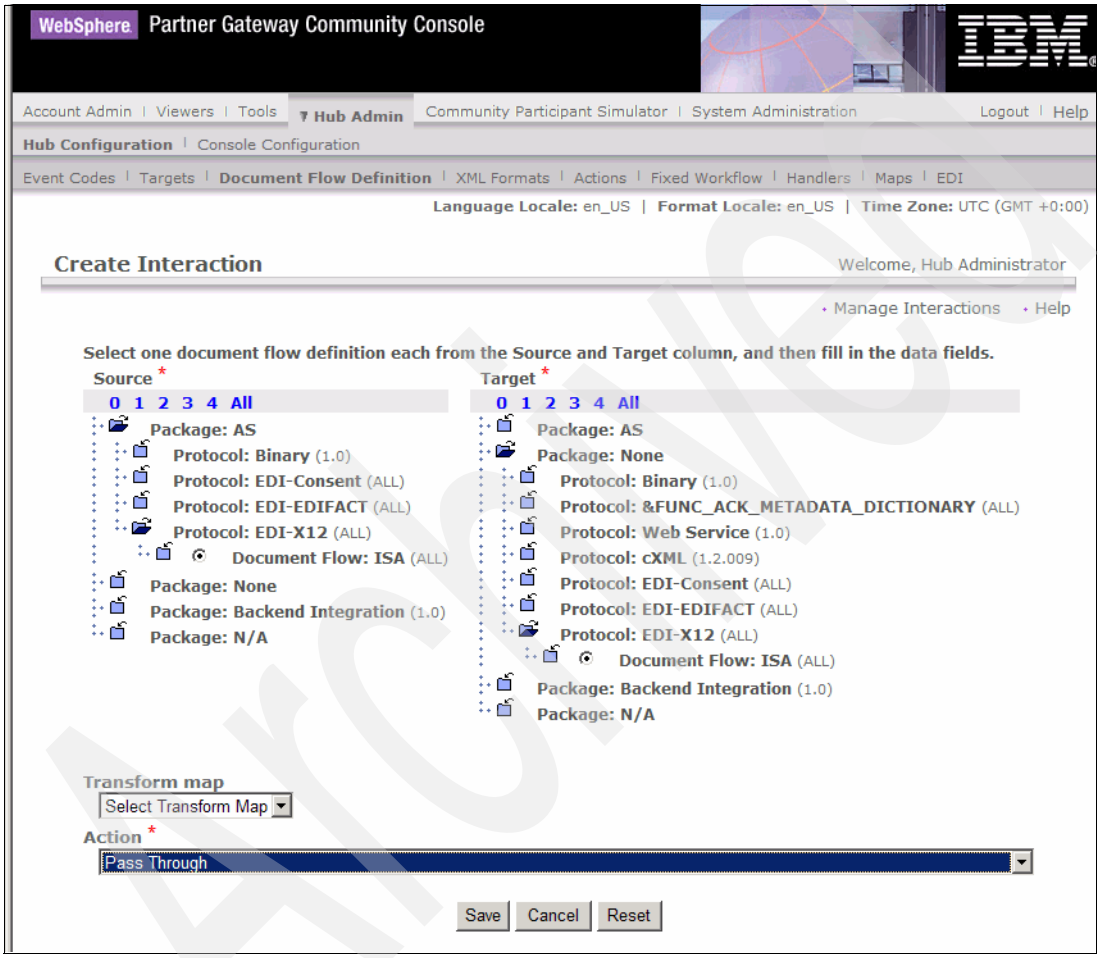


Figure 7-12 AS2 packaged EDI to unpackaged EDI

5. Using similar steps, create another the interaction to go from unpackaged EDI-X12 (None) documents to AS2 packaged EDI documents for our outbound flow. This is shown in Figure 7-13 on page 154.

   a. Select the **Pass Through** action.
   b. Click **Save**.
   c. Then click **Manage Interactions**.

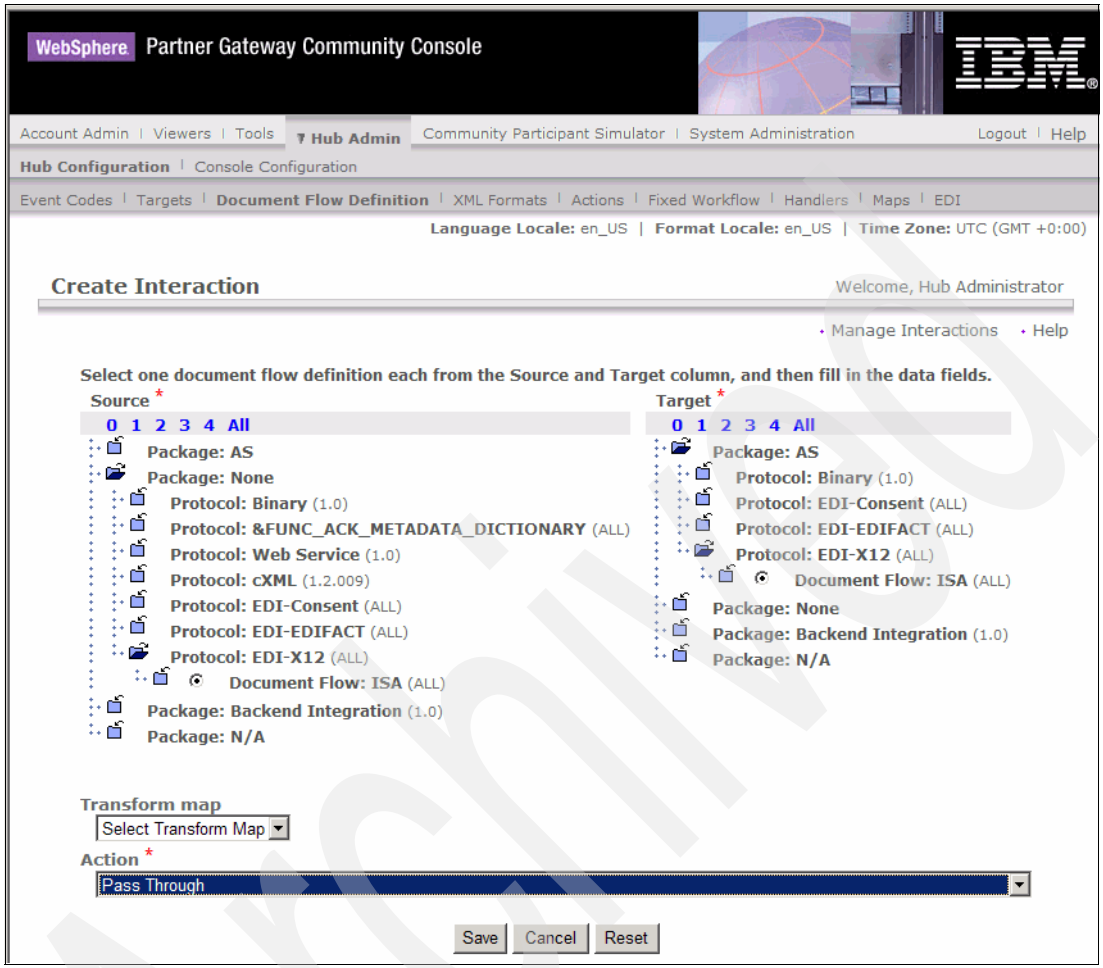


Figure 7-13 Unpackaged EDI-X12 to AS2 packaged EDI-X12

Return to the search window (Figure 7-10 on page 151) and search the interactions again. You see that are two new interactions at the bottom of the list (Figure 7-14 on page 155).

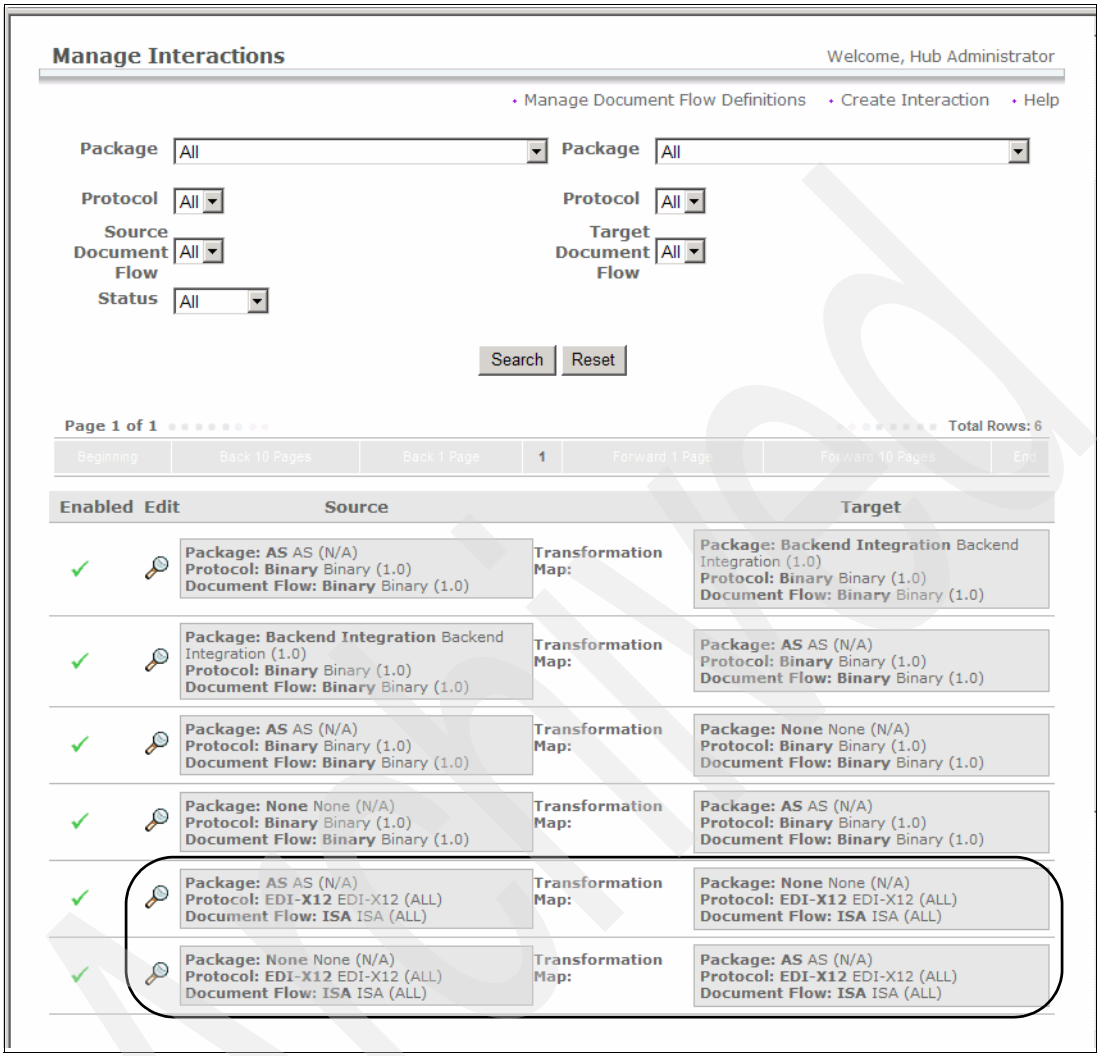


Figure 7-14 Two new interactions

You have now created the two objects labeled Interactions in Figure 7-2 on page 141 and Figure 7-3 on page 143.

7.3.3 Creating a Community Manager

As previously discussed, the Community Manager is essentially the company who purchased the WebSphere Partner Gateway to accomplish a certain task. It is this company (Company E here), that wishes to communicate with a Participant who in this case is Company A.

To create the community manager, follow these steps:

1. While logged on as the hub administrator, select **Account Admin** → **Profiles** → **Community Participant**.
2. Click **Create** to add a new participant, as shown in Figure 7-15.

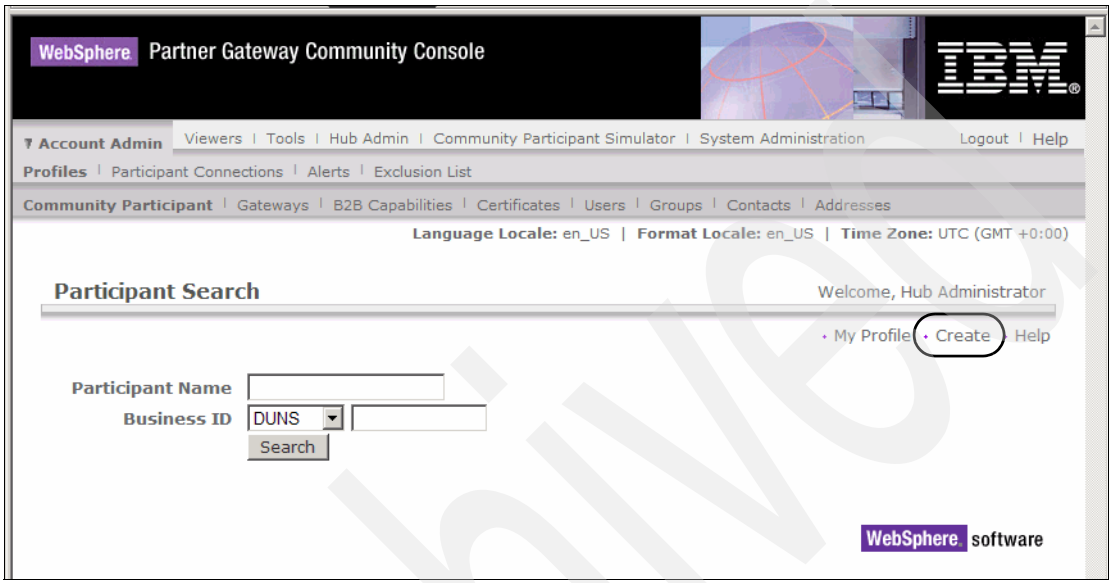


Figure 7-15 Creating new participant: Selecting Create

3. To create a new Participant (Figure 7-16 on page 157) you must:
   a. Provide a Participant login name, which is case-sensitive and cannot contain blanks. This name is used as the Company Login Name at the console login window.
   b. Provide a Participant name, which can be any text.
   c. Provide a Participant type, which can be Community Manager or Community Participant type. For this example, we want to select **Community Manager**.
   d. Optionally, you can provide a vendor type, which can have values such as Supplier or Distributor.
   e. Optionally, you can provide a Web Site. This value does *not* refer to any site used for WebSphere Partner Gateway processing, just a quick reference to the company Web site for instance.

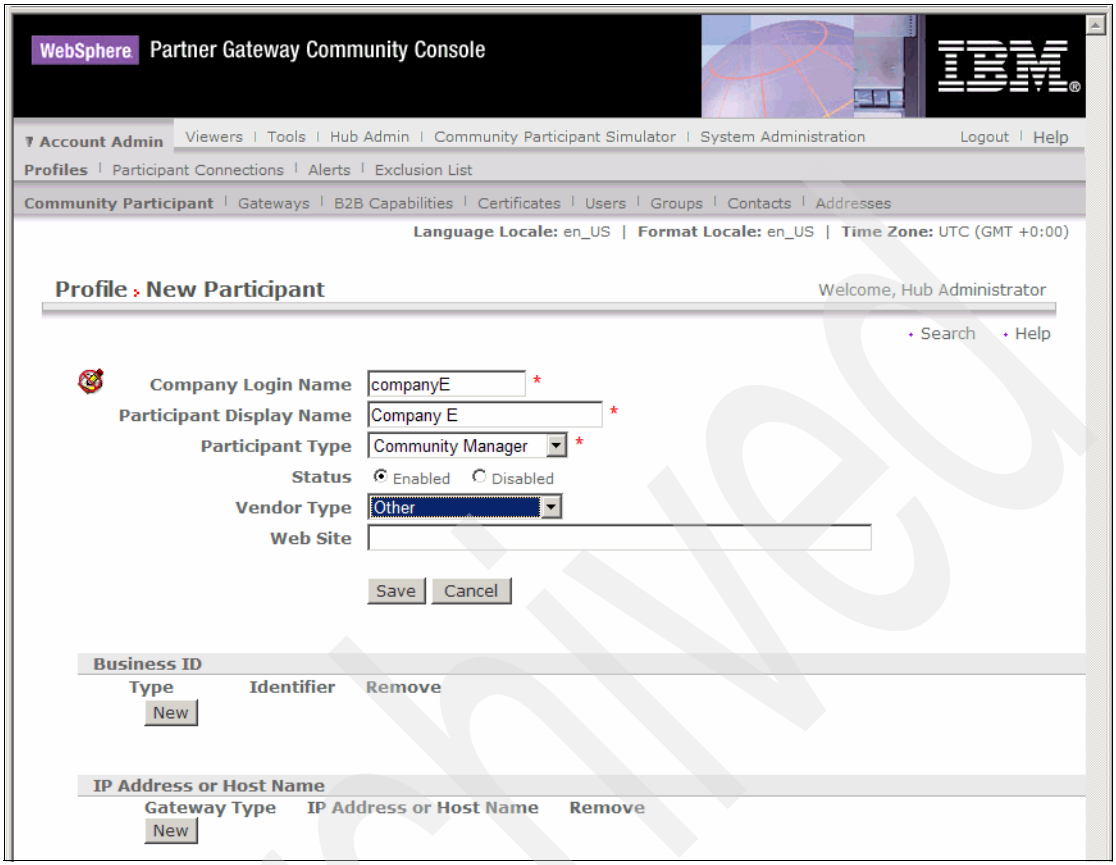


Figure 7-16 Creating a new participant: Providing information

Within B2B communication, several techniques exist to provide unique or pseudo-unique identifications. An example of unique identifications are the Data Universal Numbering System (DUNS) numbers. DUNS numbers are assigned by Dun & Bradstreet Corporation. For more information about DUNS numbers, refer to their Web site:

http://www.dnb.com

While DUNS numbers can be unique, other schemas exist that are only pseudounique, which means that there is no global organization that manages the assignment of the identifiers. An example of pseudounique identifiers are the ones used in an ISA segment of an EDI document. These identifiers tend to be unique within a given EDI network of business partners only. The EDI network provider makes sure that the identifiers are unique within the scope of their network.

Archived

a. To add an EDI identifier to a Participant profile, under Business ID, click **New**.

b. For Type, select **Freeform**.

c. For identifier, specify a value. We use the value `companye`, which is used in the following situations (Figure 7-17 on page 159).

   - The `companye` value is going to recognized as part of the ISA segment in an EDI document. This means that WebSphere Partner Gateway will parse an EDI document and locate the EDI identifiers to look up participants and participant connections internally.
   - The `companye` identifier will also be used to create the AS2 package to set the value in the AS2 From field. For incoming AS2 documents, the value for the AS2 From and To fields will be used to retrieve the participants and the participant connections.

   Note: It is possible to configure the WebSphere Partner Gateway so that the AS2 IDs can be different from the EDI identifiers. You do this by setting the appropriate attributes in the participant connection.

d. Under the IP Address or Host Name, click **New**.

e. Provide the name or IP address for the gateway that is used for the Community Manager.

f. Click **Save** in the middle of the page to store this new profile to the database.

Archived

Figure 7-17 Creating a new participant: Business ID and host name

When the new profile is created, WebSphere Partner Gateway generates a temporary password. This password must be passed to the administrator for the Community Manager company. (Figure 7-18 on page 160).

Archived

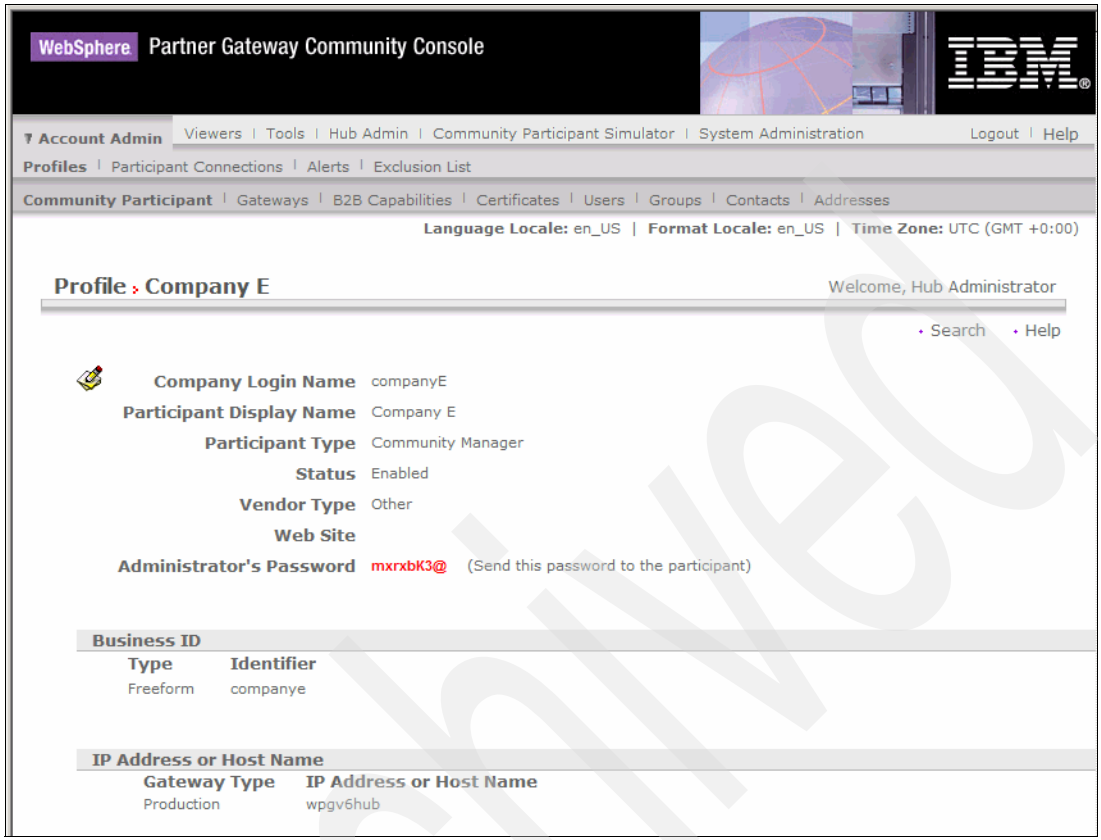


Figure 7-18 Community Manager profile created

You have completed the creation of the Community Manager. This object is the From in Figure 7-2 on page 141 and the object labeled To in the Figure 7-3 on page 143. These objects interpret the EDI documents in our example.

Resetting passwords

If a Community Participant password becomes lost, the hubadmin user can reset it.

1. Log on as `hubadmin`.
2. Select **Account Admin** → **Profiles** → **Community Participant**. In the Participant Search window, click **Search**.
3. From the list of the company profiles, open the details of the company to which the user belongs. From the menu select **Users.**

4. From the list of users who belong to this company profile, open the details of that user, for example **admin**. Click the **edit record** icon.
5. Set the password or generate a new password for this user, Figure 7-19.

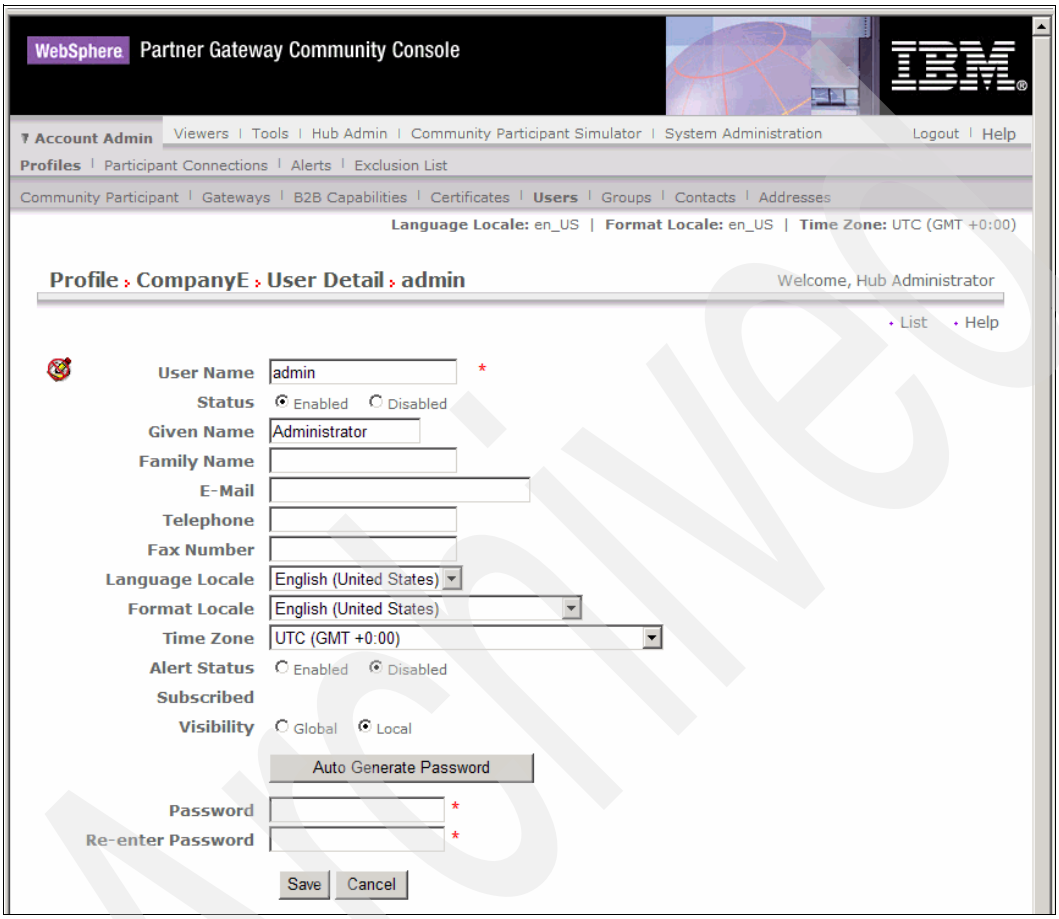


Figure 7-19 Changing the password of a user

7.3.4 Creating a Community Participant

The same steps used to create the Community Manager are used to create the Community Participant. The Participant in this case is Company A, which uses WebSphere Partner Gateway Advanced. The login name is `companyA`. Refer to Figure 7-20 on page 162 for a reminder of the options available.

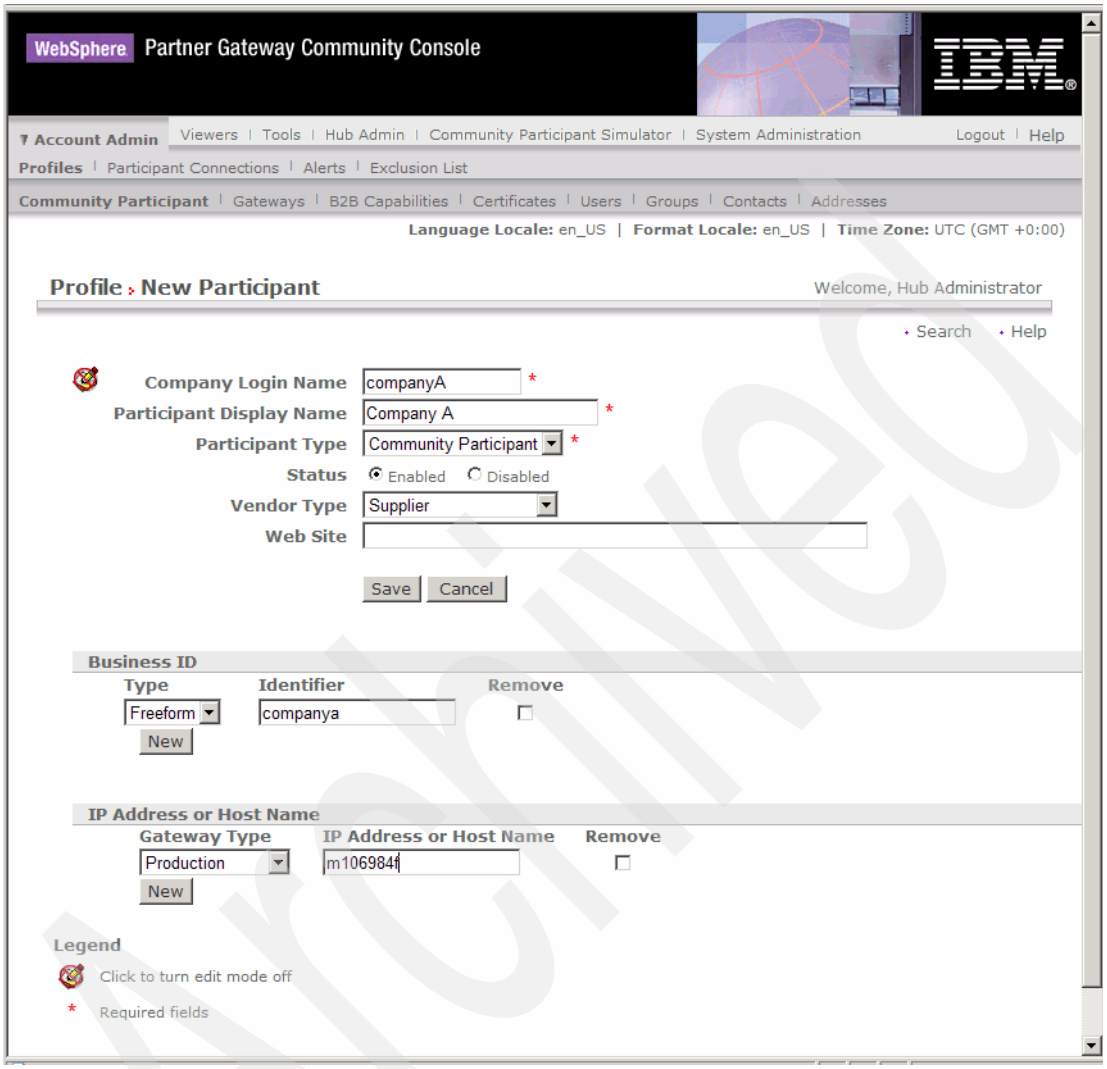


Figure 7-20 Creating new Participant

The business identifier companya is going to be used as the value for the object labeled To in Figure 7-2 on page 141 and the object labeled From in Figure 7-3 on page 143.

> **Note:** The hub administrator might not necessarily have information such as the business identifier of the Community Participant. However, when the Community Participant logs on the first time, he or she can update his or her own profile to add information.

Before switching roles, we can summarize what we have done so far (Figure 7-21). For the inbound flow, we have defined:

- The target, HTTPTarget that is awaiting documents from the participant
- The interaction that defines AS2 packaged EDI documents to unpackaged EDI documents
- The IDs, From and To, as used in the EDI document and the AS2 packaging

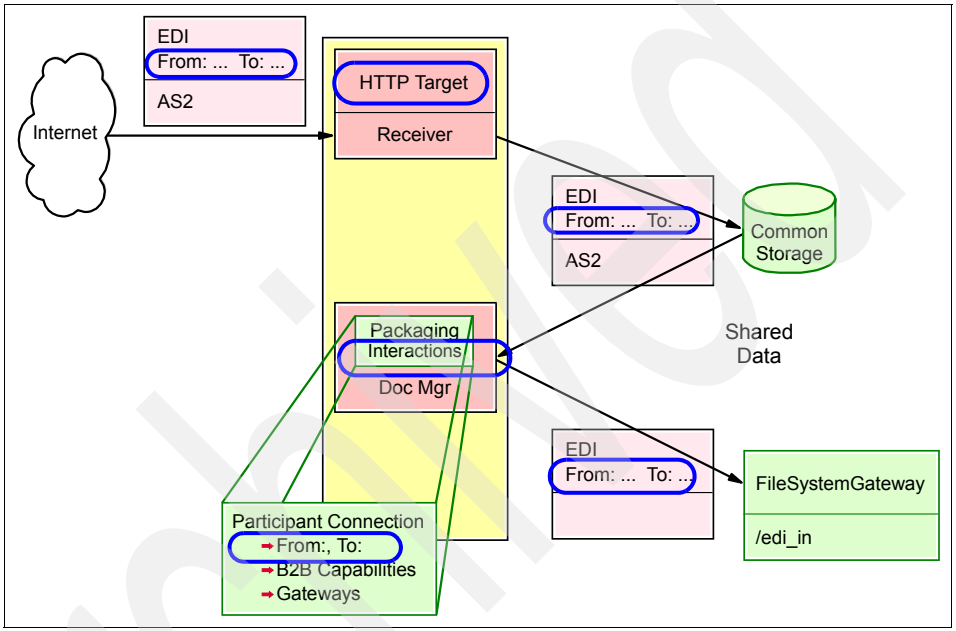


Figure 7-21 Status of configuration for inbound flow

For the outbound flow (Figure 7-22 on page 164), we have defined:

- The target FileSystemTarget for picking up unpackaged EDI documents from the file system
- The interaction from unpackaged EDI to AS2 packaged EDI documents
- The IDs, From and To, as used in the EDI document and the AS2 packaging

Archived

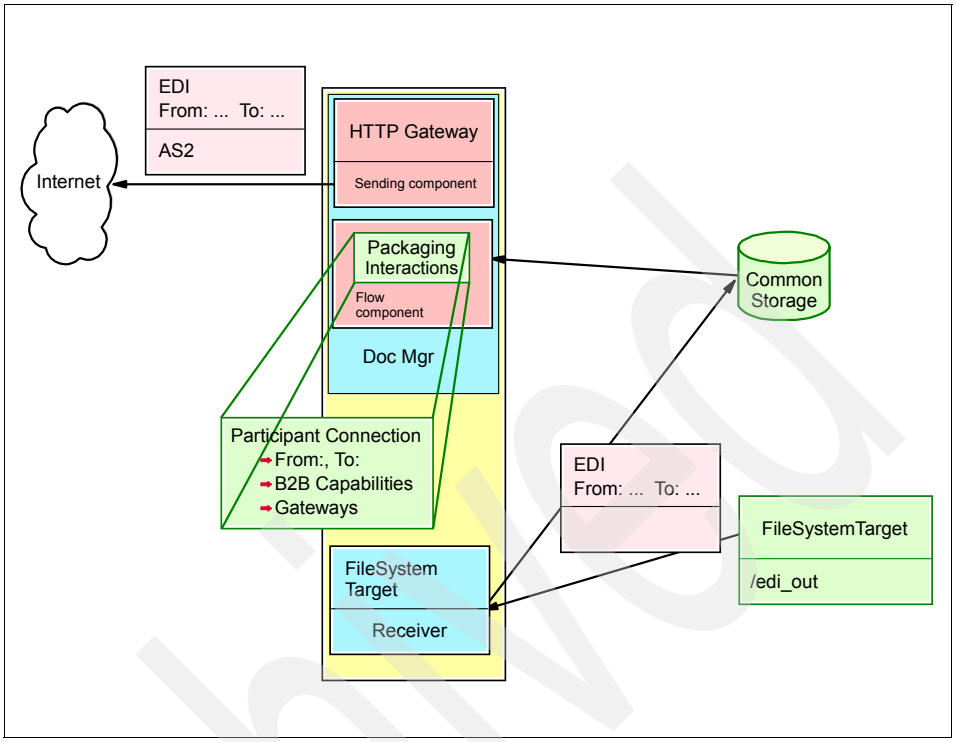


Figure 7-22 Status of configuration for outbound flow

So far the hub administrator has performed all the tasks. By creating the Community Manager and Community Participant (and by default, their admin user IDs) he has created the objects needed for those users to help with the configuration. This makes it possible for them to provide some information about themselves as discussed in 7.2, “Role-based configuration” on page 138.

7.4 Configuration tasks for Company E administrator

The previous sections explain how to create the Community Manager, which we named Company E. The administrator for Company E now can perform some of his own configuration work (Figure 7-23 on page 165).

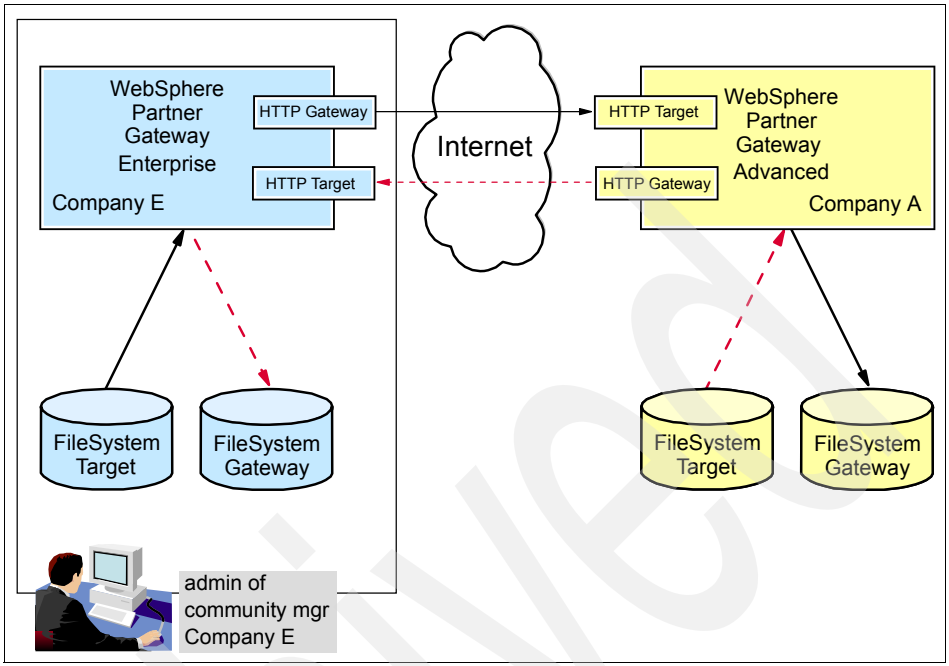


Figure 7-23 logging on as admin of Community Manager to Company E

7.4.1 Initial logon by the Community Manager

The login name defined in 7.3.3, “Creating a Community Manager” on page 155, is not the User Name. It is the value used in the Company Login Name field in the welcome window of the Community Console. For each new Participant, a predefined user name is created, which is admin.

Using the User Name **admin**, the Company Login Name **companyE**, and the generated password (see Figure 7-24 on page 166), log on to WebSphere Partner Gateway.

Archived

Figure 7-24 Logging on as admin of Company E

You are prompted to change the autogenerated password you just entered before being taken to the first console screen.

If required or desired, the Company E admin can now add additional users and groups within Company E that would like to interact with the Community Console.

7.4.2 Creating a gateway

To proceed with our example of sending EDI documents over AS2 between Company E and Company A, the administrator of Company E needs to create a gateway. A gateway is the *exit point* out of WebSphere Partner Gateway and into the world of Company E. We covered how files enter into the system through targets, when the Document Manager is finished processing the documents, they are sent out of the system through gateways.

For this example, we use a file-based gateway. The assumption is the gateway already exists as an entry point back into the systems behind the WebSphere Partner Gateway product. Therefore, make sure the complete directory structure exists before you define the gateway.

Archived

Attention: This is slightly different from the file system target you created previously where WebSphere Partner Gateway created the last directory in the path. This time you must make sure every level of the path is already defined.

1. Using Explorer, create a directory to store incoming EDI documents. For Company E, we used the value `\wpgv6\data\companye\edi_in`.
2. Return to the Community Console and select **Account Admin** → **Profiles** → **Gateways**.
3. You see an empty list of gateways. Click **Create**.
4. In the gateway List panel (Figure 7-25 on page 168), complete the following tasks:
   a. Provide a name for the gateway and select the transport, which is a **File Directory** for our scenario.
   b. Enter the complete directory structure in the URI format. In this case, there are three slash (/) characters following the colon character. If a disk letter is required, add C: between the second and third slash character. In the URI format, the Windows directory separator backslash (\) is not used. Also note that if you use the format like this example with three slash characters, it is assuming the drive is the same drive as the installation of WebSphere Partner Gateway.
   c. The Use Unique File Name check box is checked by default, and we will keep it this way. There are some configurations where you might want to overwrite files of the same name as you write them out to a gateway.
   d. Click **Save** to store the new gateway in the database.
   e. Click **List** to return to the list of defined gateways.

Archived

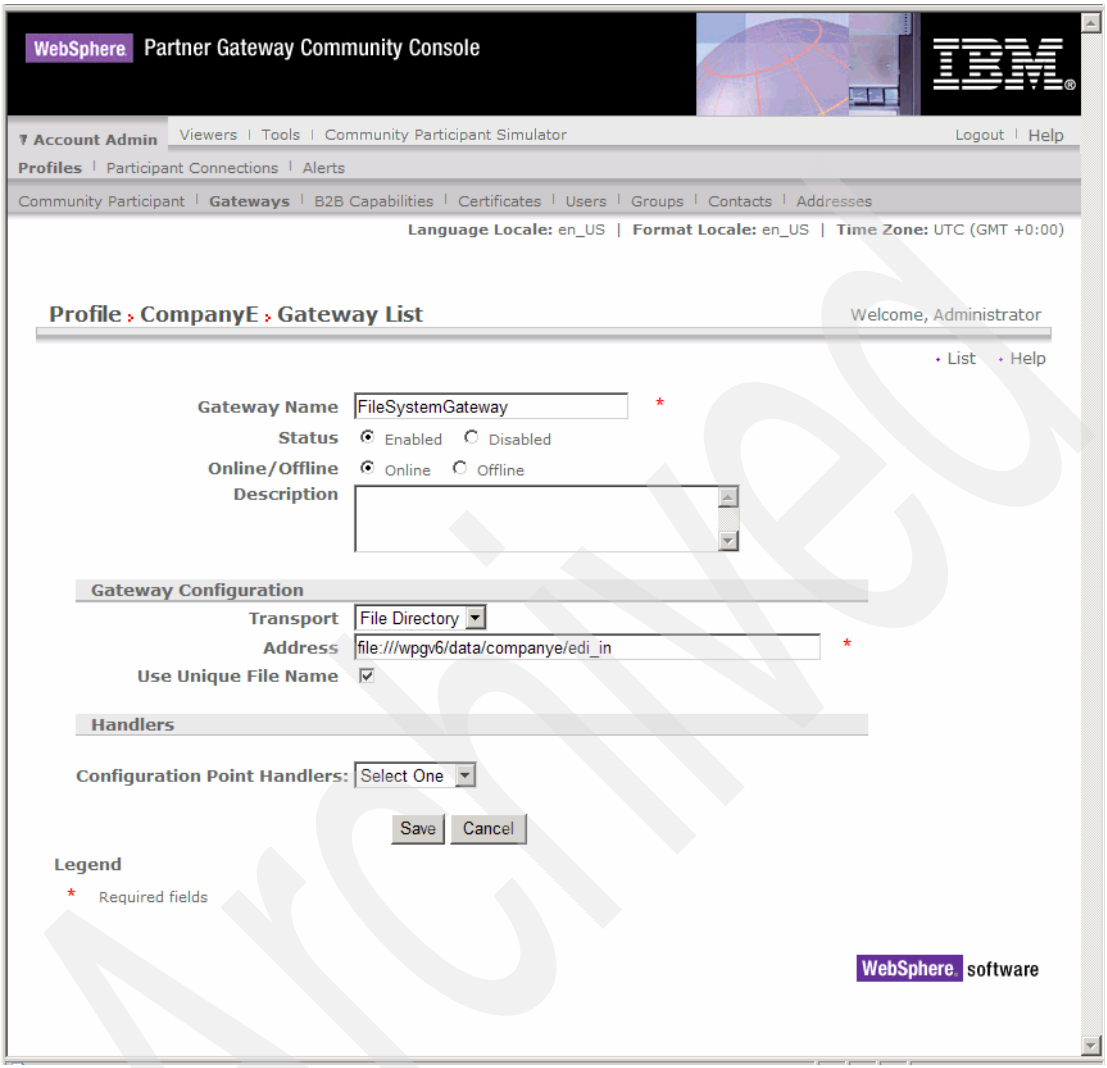


Figure 7-25 Creating a new gateway

Note: If you compare the top menu options for the hub administrator in (Figure 7-15 on page 156). with the menu options for this admin user of the Community Manager in (Figure 7-25). You can see that the option for Hub Admin is unavailable to this user, preventing him from creating targets and document flow definitions.

Archived

5. Figure 7-26 lists the new gateway. It also shows that a default gateway has not been provided. There is no checkmark under Default. To label a gateway as the default gateway, click **View Default Gateways**.

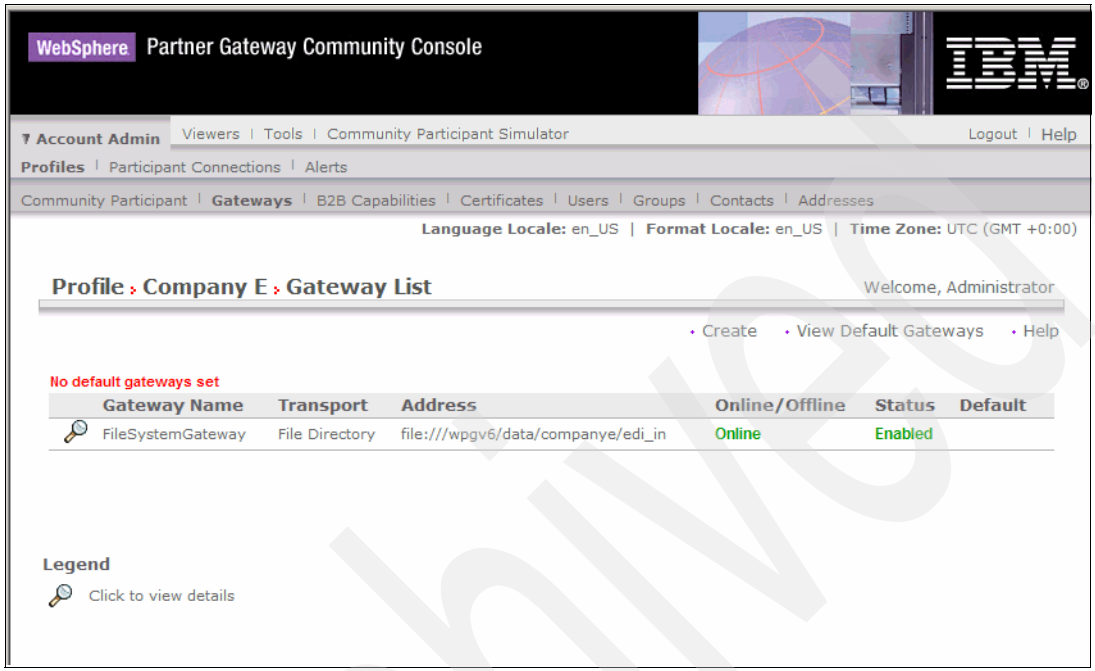


Figure 7-26 List of defined gateways

6. Figure 7-27 on page 170 shows how to select a default gateway for each type of gateway. Click **Save** to store your selection, then click **List**.

Archived

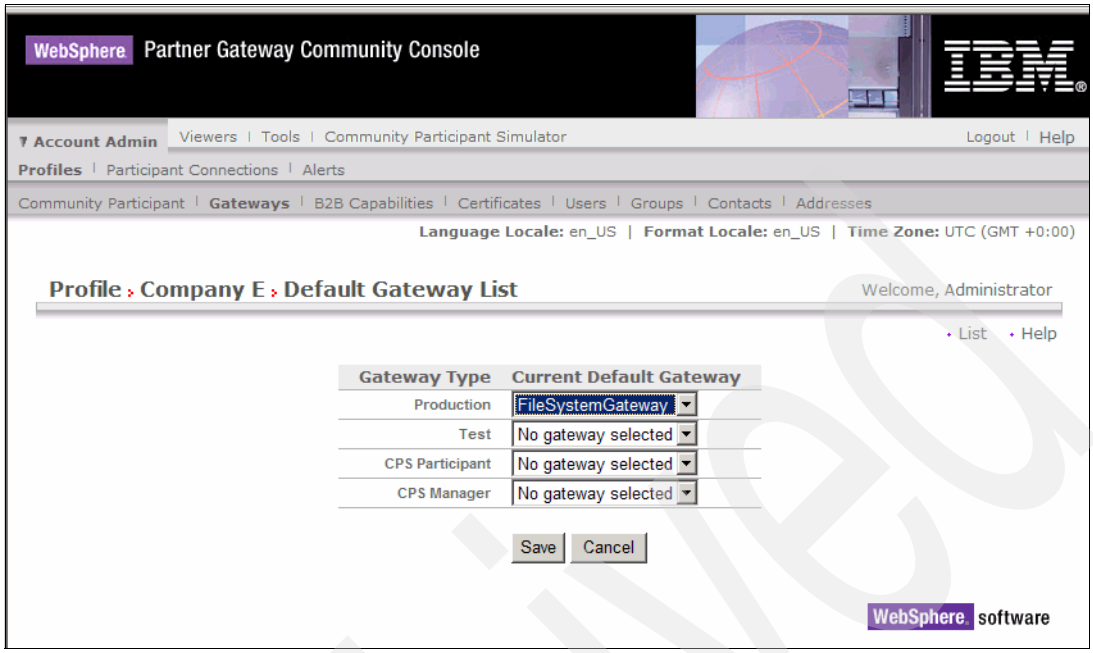


Figure 7-27 Setting the default gateway

You return to the list of gateways (Figure 7-28 on page 171). Notice that there is red checkmark labelling this gateway as the default gateway. Later, when configuring channels you will see this as the default gateway for this system.

Archived

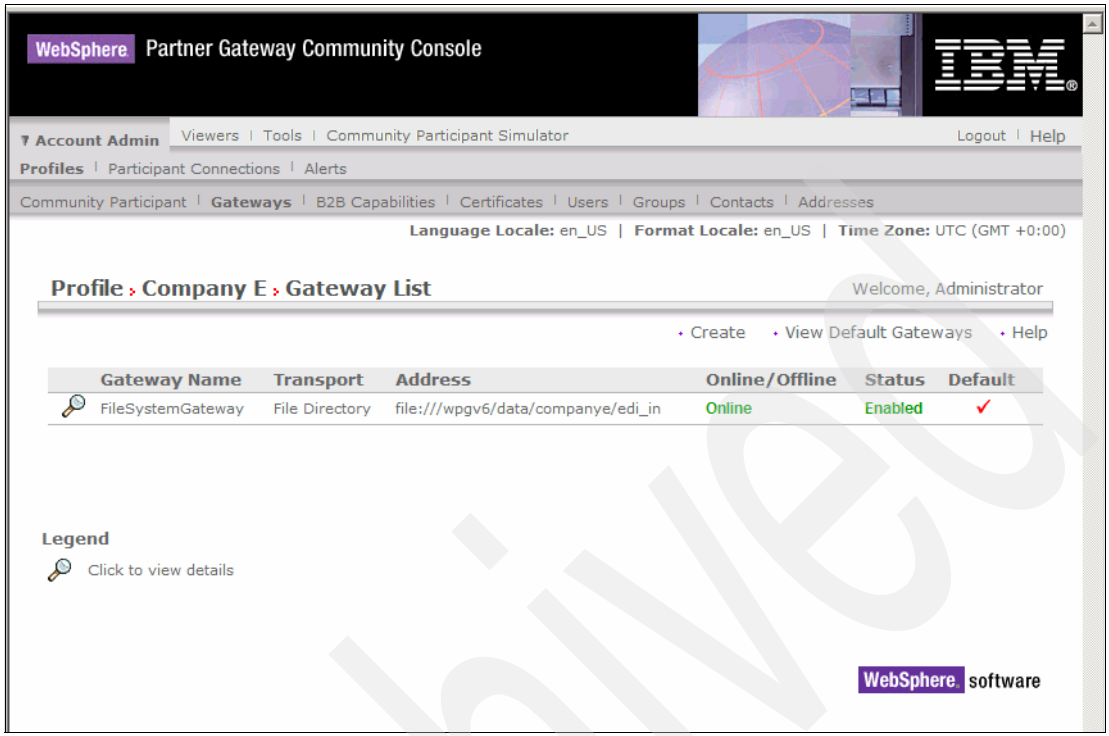


Figure 7-28 List of defined gateways

Referring to Figure 7-28, we have now completed the setup for the object labeled FileSystemGateway.

7.4.3 Providing B2B capabilities

Earlier we configured the interactions of the WebSphere Partner Gateway server. Because the Community Manager wants to make use of those interactions, he needs to modify his B2B capabilities match the definition of the interaction he wishes to use. To do this :

1. While logged on as the administrator of company E, select **Account Admin** → **Profiles** → **B2B Capabilities** (Figure 7-29 on page 172).

Archived

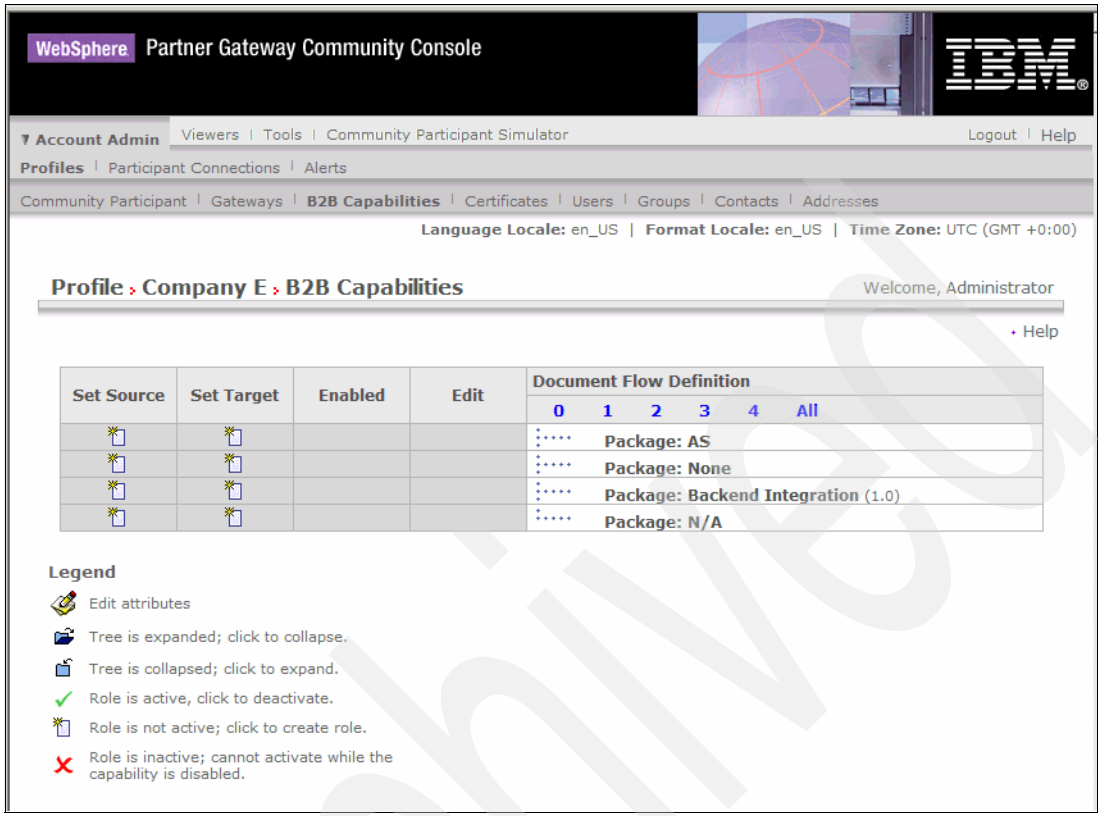


Figure 7-29 Providing B2B capabilities for Community Manager (Company E)

2. As shown in Figure 7-30 on page 173, complete these tasks:
   a. Click the check box icons underneath **Set Source** and **Set Target** for **Package: AS**.
   b. Click the folder icon next to **Package: AS** to expand it. Then click the check box icon for **Protocol: EDI-X12** for both **Set Source** and **Set Target**.
   c. Click the folder icon next to the **Protocol: EDI-X12** to expand that folder. Then click the check box icon for **Document Flow: ISA** for both the **Set Source** and **Set Target** columns.
   d. Repeat this sequence for the **Package: None**.

   When you have finished, your display should look similar to the example in Figure 7-30 on page 173.

Archived

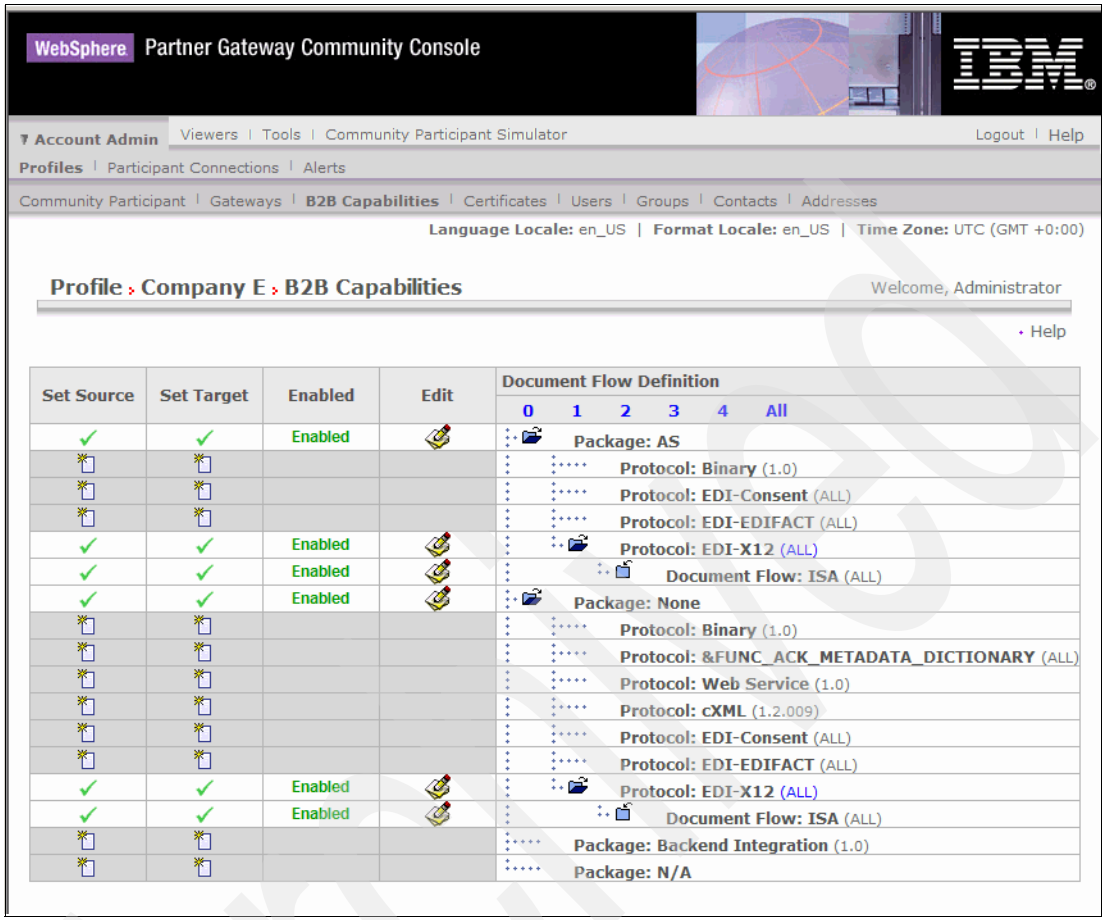


Figure 7-30 Completed B2B capabilities

The Community Manager has basically told the hub administrator that he wants to be enabled for these B2B capabilities.

If you select the edit icon next to **Package: AS**, you can set several AS-related parameters, such as encryption or custom AS2 identifiers. If you set them here as in Figure 7-30, they become the default values for all AS2 connections that involve Company E. However, you can also set them later at the participant connection level, so that the parameters will only apply to that specific Participant pair. We will be configuring it in that manner later as the hubadmin user.

This completes the tasks for the administrator user of Company E.

7.5 Configuration tasks for partner Company A

In 7.3.4, “Creating a Community Participant” on page 161, you created a Participant profile for Company A. A representative of Company A can now login to the Community Console at Company E’s site. Similar to the last section where the Community Manager configured some of his own parameters, the Company A Participant will do the same.

Note: Because we are using role-based configuration in this example, it is important to realize that Company A is *not* configuring his own implementation of WebSphere Partner Gateway that we installed in Chapter 6, “WebSphere Partner Gateway Advanced for AIX” on page 109. He is logging on to Company E’s implementation as a Participant to complete some of the configuration tasks for the hub administrator of Company E.

Figure 7-31 illustrates what happens after you log on as the administrator of the Community Participant to Company E.

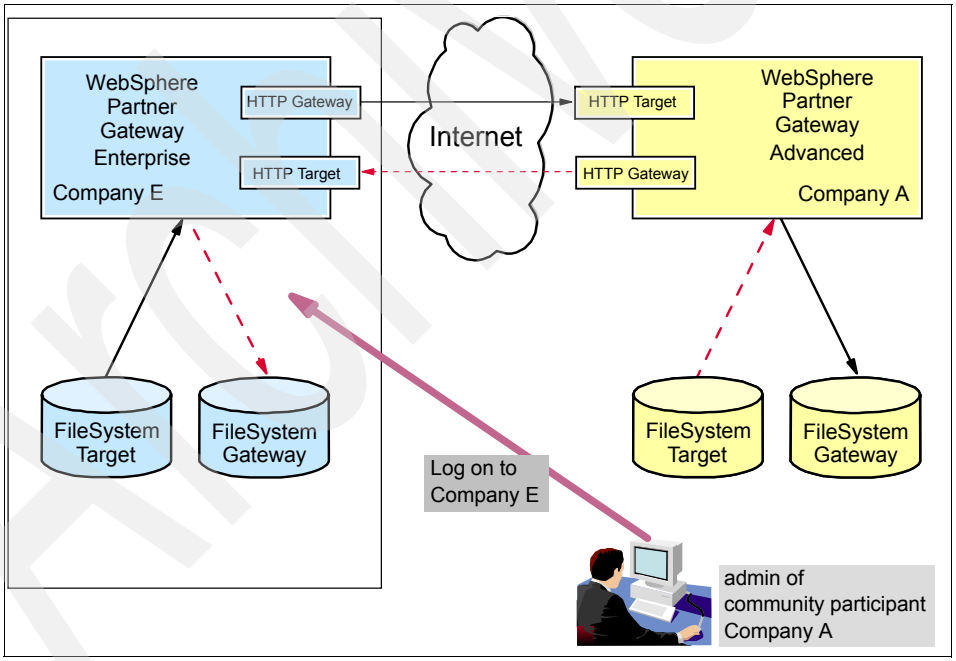


Figure 7-31 Logging on as admin of community Participant to Company Server

1. Log on to the hub of Company E, as depicted in Figure 7-31. Use these values:

- User Name: `admin`
- Password: was generated by WebSphere Partner Gateway when the hubadmin user created Community Participant named `companyA`.
- Company Login Name: `companyA`, as defined in 7.3.4, “Creating a Community Participant” on page 161.

2. After the initial logon, the administrator of Company A again must provide a new password. Similar to what the Community Manager needed to do, the Community Participant for Company A must now create a gateway, and enable their B2B capabilities.
3. Select **Account Admin** → **Profiles** → **Gateways**.
4. When you see the empty list of gateways, click **Create**.
5. In the Gateway List panel (Figure 7-32 on page 176), complete these tasks:
   a. Provide a name for the gateway.
   b. Select the transport, which is **HTTP/1.1**
   c. In our network environment we do not have a proxy server so In the Forward Proxy List, pull-down the menu and select **Use no forward proxy**. If you do have a forward proxy server, please see the product documentation.
   d. Provide the target URI.

   > **Note:** Because our example is between two implementations of WebSphere Partner Gateway, the target URI on this gateway (exit point) needs to match the target (entry point) on Company A’s implementation. So when Company A’s server is configured, you already know where his HTTP target should be listening.

   e. Click **Save**.

Archived

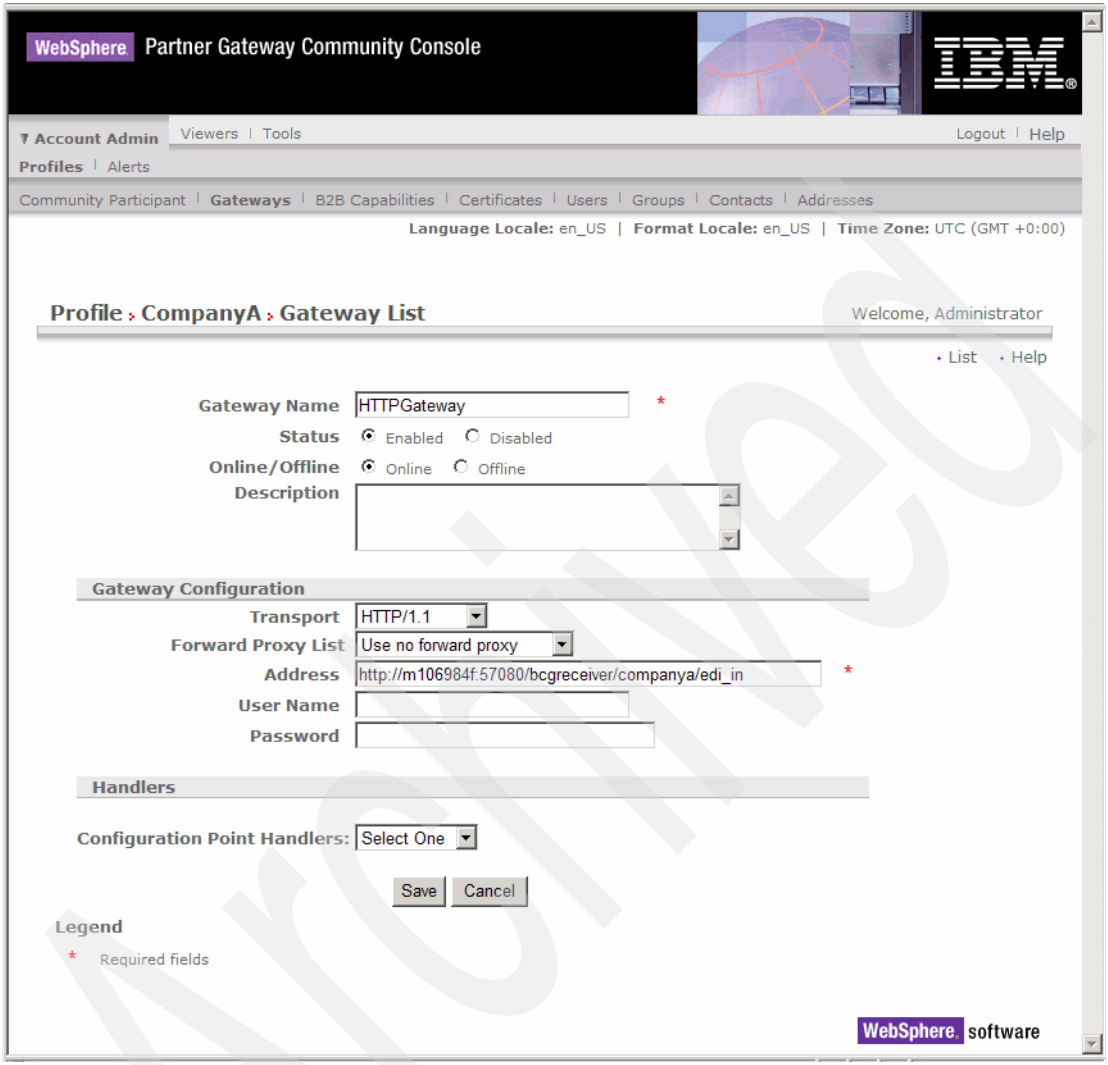


Figure 7-32 Creating a gateway for Company A

The WebSphere Partner Gateway for Company A runs on an AIX machine. The configuration to enable this dual communication with Company E is discussed starting with 7.7, “Configuration tasks for the Company A hubadmin” on page 188.

6. After the gateway object is stored in the database, click **List** to return to the list of gateways.

7. The list of gateways does not have a default gateway assigned. Click **View Default Gateways** to assign the HTTPGateway object as the default gateway for gateway type production.

   The list of gateways for Company A should now look like the example in Figure 7-33.

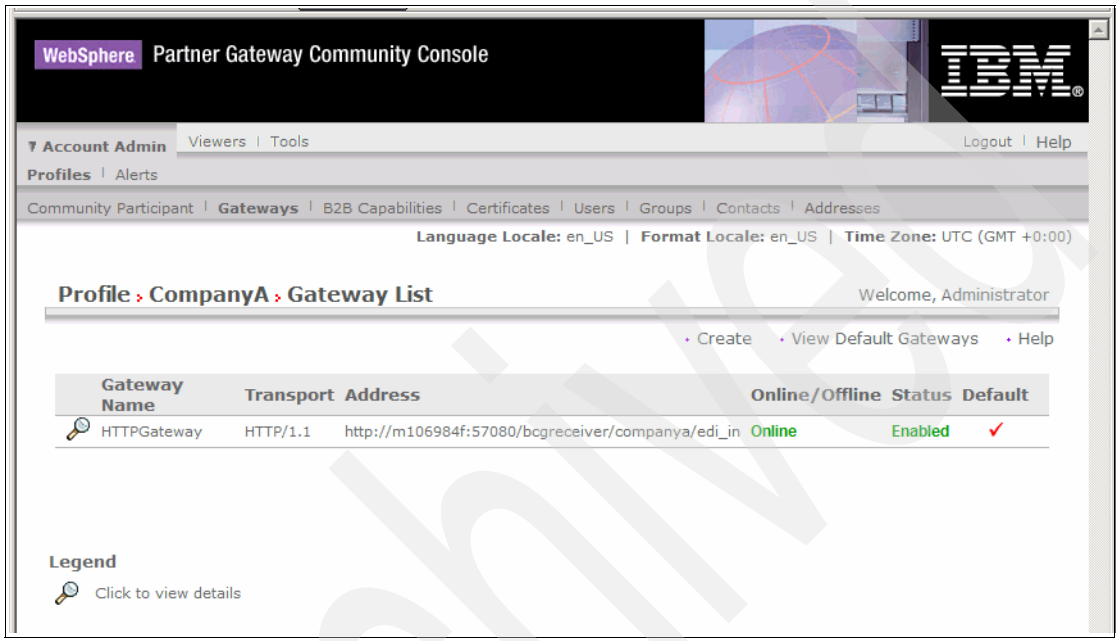


Figure 7-33 List of gateways for Company A participant

8. Select **Account Admin** → **Profiles** → **B2B Capabilities** so that Company E's WebSphere Partner Gateway server knows that Company A partner is enabling the same definition's that the Community Manager is. The procedure is the same as enabling the B2B capabilities of the Community Manager. The result is shown in Figure 7-34 on page 178.

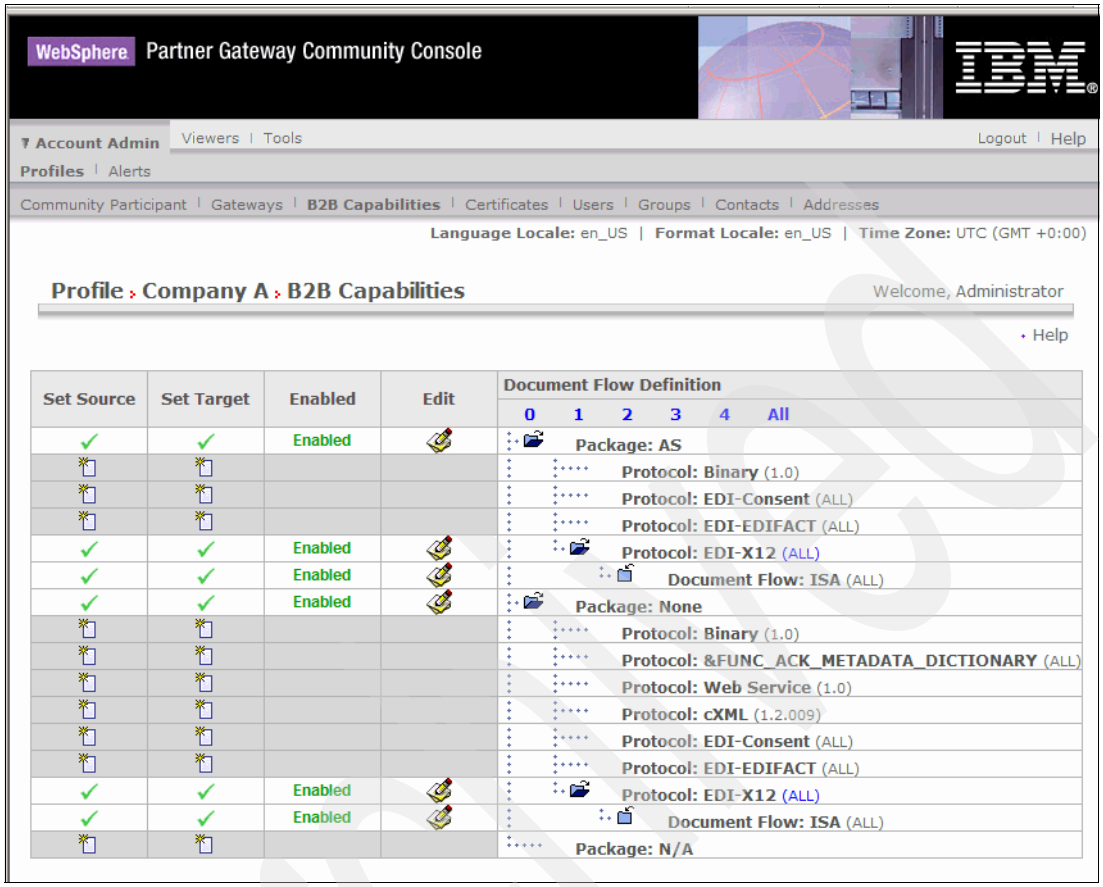


Figure 7-34 B2B capabilities for Company A participant

Before you proceed to the last step, again refer to the overview pictures for the outbound flow. In 7.4, “Configuration tasks for Company E administrator” on page 164 we defined the B2B capabilities for Company E, which are referenced in Figure 7-35 on page 179 and in Figure 7-36 on page 180. We also defined the FileSystemGateway object, which is also marked in Figure 7-35 on page 179.

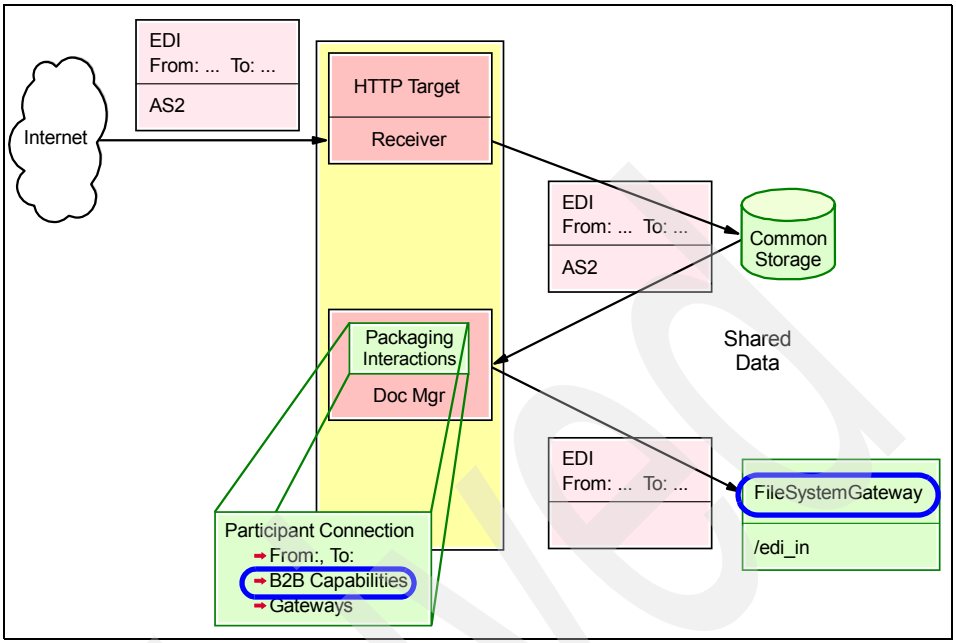


Figure 7-35 Status of configuration for Inbound flow

In 7.5, “Configuration tasks for partner Company A” on page 174, we defined the B2B capabilities for Company A, which is referenced in Figure 7-35 and in Figure 7-36 on page 180. We also defined the HTTPGateway object which is marked in Figure 7-36 on page 180.

Combining the previous summary with this one, there is only one remaining concept to complete in each picture: the participant connection that ties together the Community Manager and Community Participant, the interactions, and the different targets and gateways.

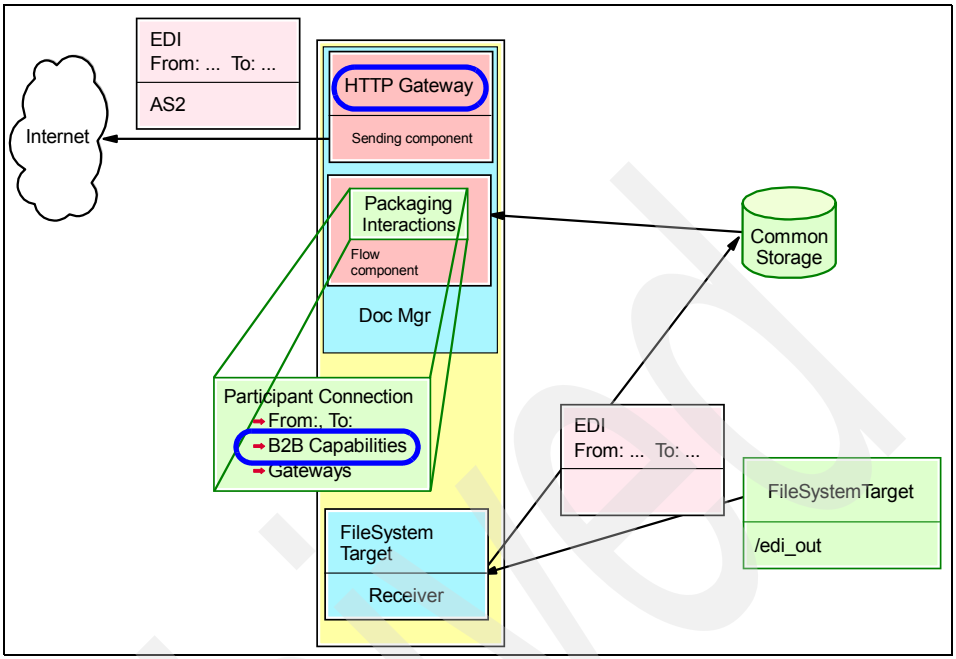


Figure 7-36 Status of the configuration of the outbound flow

7.6 Connecting Company E to Company A

At this point, the administrators of Company E and A have performed their tasks on the WebSphere Partner Gateway server residing at Company E's site. Now the hub administrator can tie the pieces together and create the participant connection.

This task could be performed by the admin user of the Community Manager. He also has the authority to do this. However, from a role-based view, it might be better to assign this task to the hubadmin. As soon as the participant connection is created, communication activity can begin on the server. This gives the hubadmin user the opportunity to review the configuration that was created by the admin users of both Company E and Company A.

The following discussion describes the required steps while logged on as the hubadmin user (Figure 7-37 on page 181).

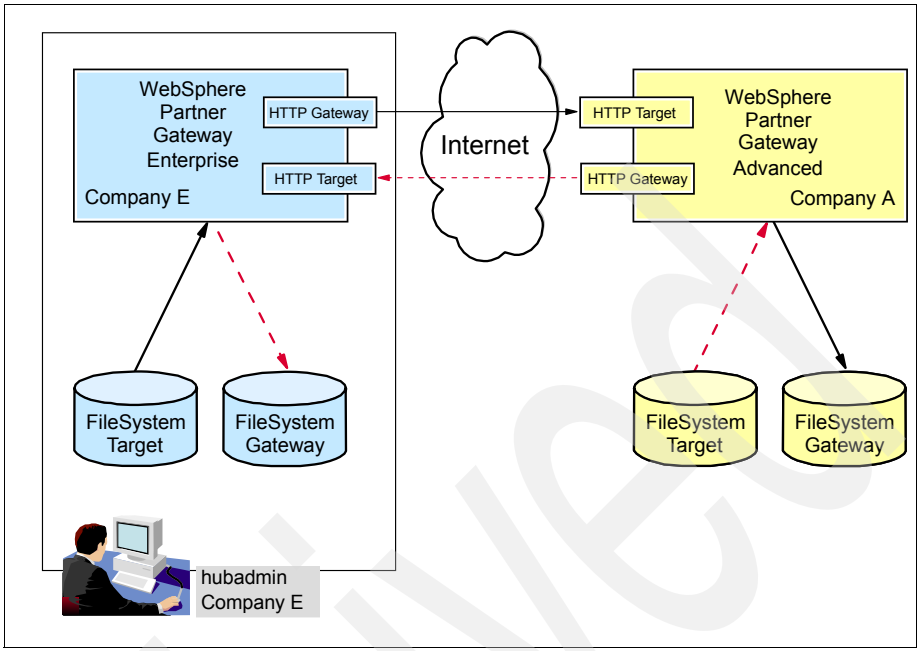


Figure 7-37 Logging on as hubadmin to the server of Company E

1. Log on as **`hubadmin`** user.
2. Select **Account Admin** → **participant connections**. See Figure 7-38 on page 182.
3. The browser presents the option to manage connections. For Source, select **Company E**, and for Target, select **Company A**. Click **Search**.

Archived

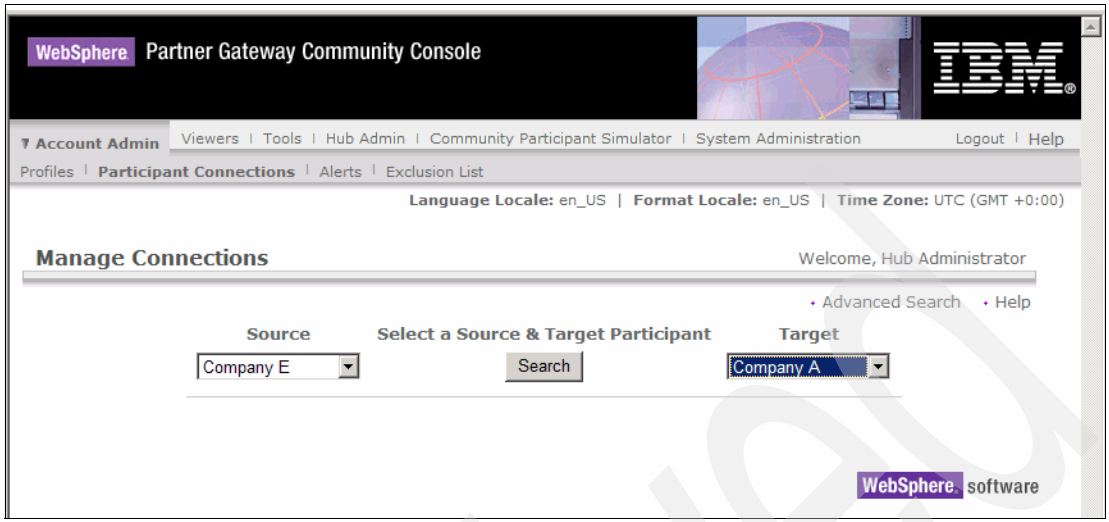


Figure 7-38 Searching for a connection between Companies E and A

4. WebSphere Partner Gateway now looks for matching B2B capabilities between Company E and Company A. Figure 7-39 on page 183 shows the matches. Click **Activate** on both to enable the matching connections.

> **Note:** The results of this search are based on two criteria.
>
> - One, the interaction has to exist on the system, which we created with the hub admin user.
> - Two, the B2B capabilities for the Source and Target participants that you choose must have been activated.
>
> You can easily see how a system that has many interactions and participants will only show the most useful ones between any pair of participants.

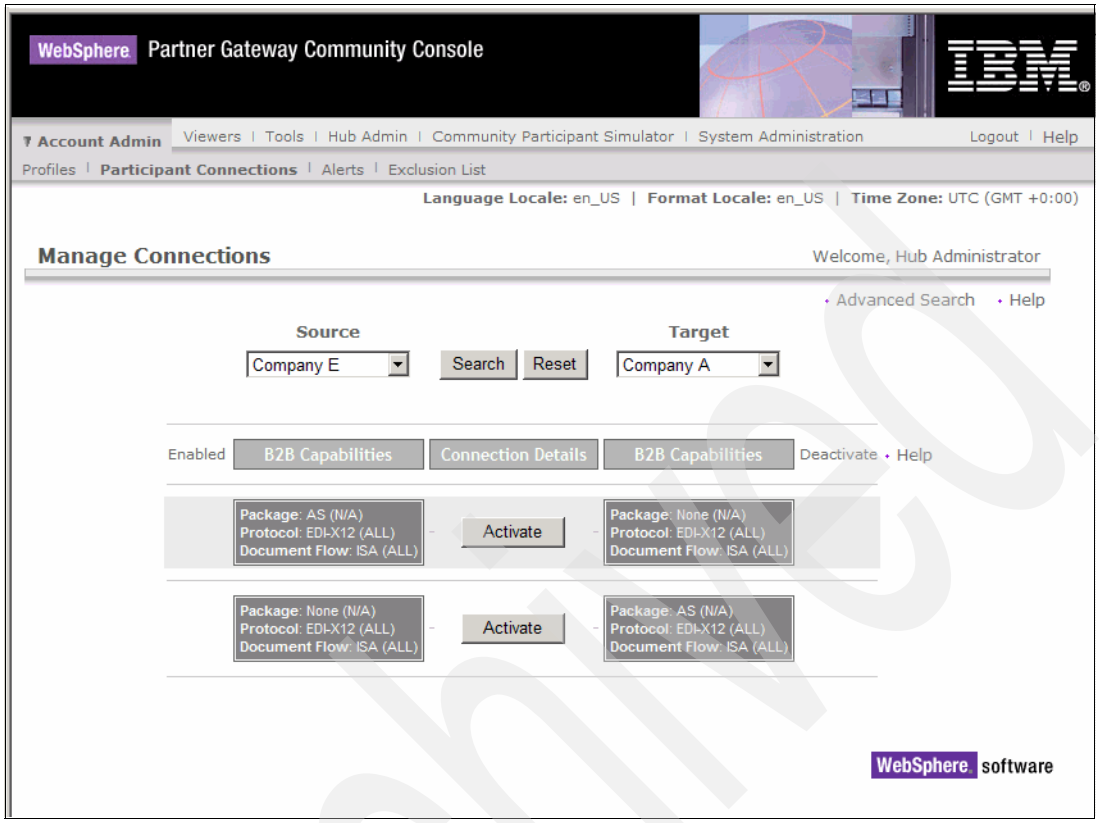


Figure 7-39 Matching B2B capabilities

5. When the connections are activated, more options are shown (Figure 7-40 on page 184). Click **Gateways** to select the correct gateway for this connection.

Archived

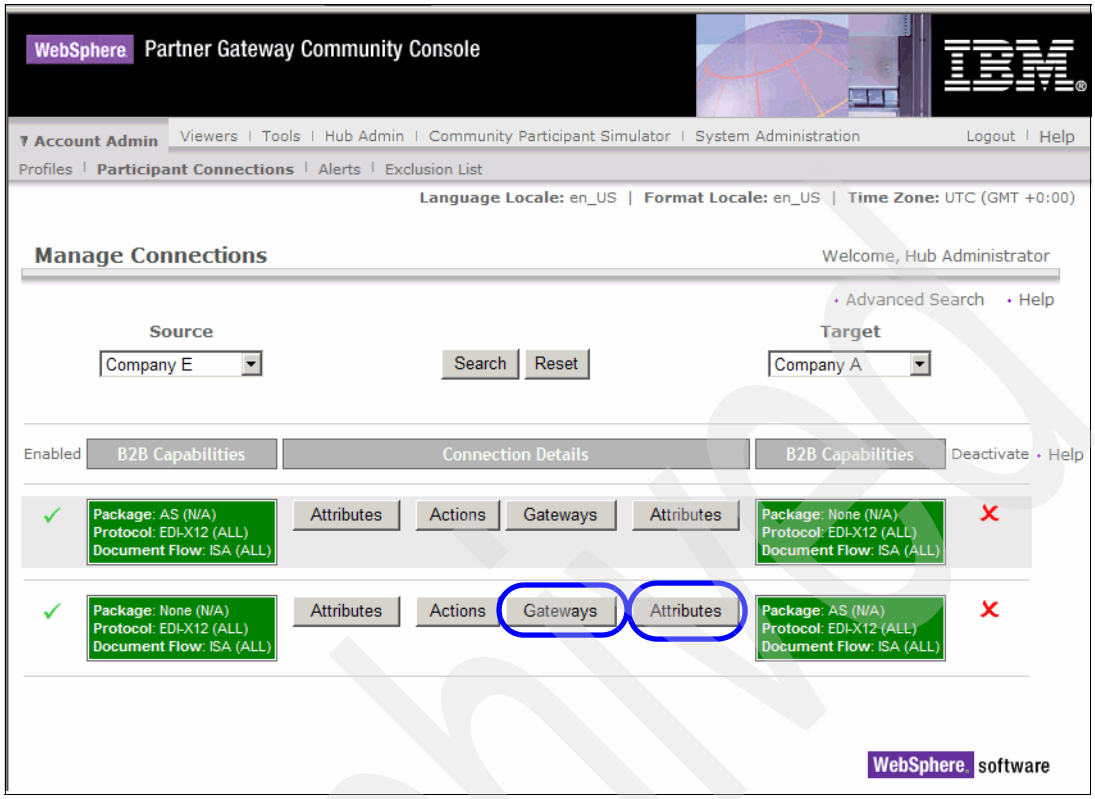


Figure 7-40 Connection details

A window opens where the correct gateways are already selected, because we defined the gateways as default gateways (Figure 7-41 on page 185). Review the gateway information and click **Save**.

Archived

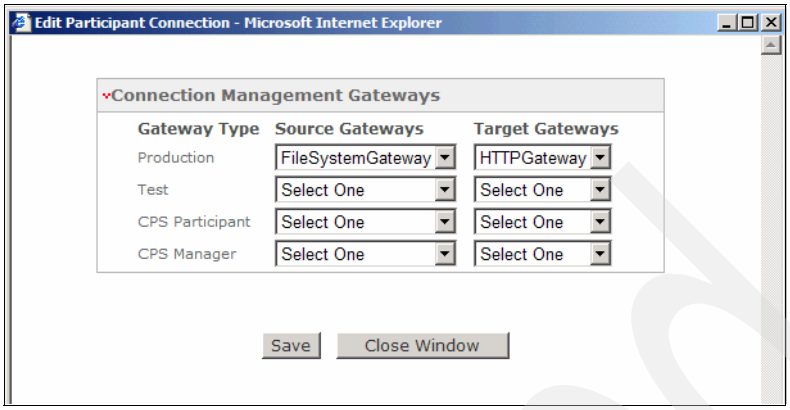


Figure 7-41 Confirming selected gateways

Click **Attributes** next to the AS2 package for the target partner (Figure 7-40 on page 184). This allows you to set a number of AS2-related parameters.

6. Figure 7-42 on page 186 shows the current connection attributes. As you can see, there is currently no value set for the parameter AS MDN Http Url. However under that parameter, you can see another parameter called AS MDN Requested. which is set to Yes. Clearly this cannot work. Requesting a message disposition notification (MDN) means that you need to provide a location to which it can be delivered.

   To update the parameter AS MDN Http Url, click the folder icon next to **Package: AS (N/A)** circled in Figure 7-42 on page 186. This opens a folder where the AS2-related parameters can be changed.

Archived

Connection Attributes Welcome, Hub Administrator

• Return • Help

Source: **Company E**
Package: None (N/A)
Protocol: EDI-X12 (ALL)
Document Flow: ISA (ALL)

Target: **Company A**
Package: AS (N/A)
Protocol: EDI-X12 (ALL)
Document Flow: ISA (ALL)

Scope: Connection

Connection Summary

Attribute	Description	Current Value	Inheritance
Time To Acknowledge	Time To Acknowledge	30	Inherited from: Scope: Global Type: Time To Acknowledge
Retry Count	Retry Count	3	Inherited from: Scope: Global Type: Retry Count
AS Compress Before Sign	AS Compress Before Sign	Yes	Inherited from: Scope: Global Type: AS Compress Before Sign
AS Compressed	AS Compressed	No	Inherited from: Scope: Global Type: AS Compressed
AS Encrypted	AS Encrypted	No	Inherited from: Scope: Global Type: AS Encrypted
AS MDN Http Url	AS MDN Http Url		No value provided
AS MDN Email Address	AS MDN Email Address		No value provided
AS MDN Asynchronous	AS MDN Asynchronous	Yes	Inherited from: Scope: Global Type: AS MDN Asynchronous
AS MDN Requested	AS MDN Requested	Yes	Inherited from: Scope: Global Type: AS MDN Requested
AS Message Digest Algorithm	AS Message Digest Algorithm	sha1	Inherited from: Scope: Global Type: AS Message Digest Algorithm
AS MDN Signed	AS MDN Signed	No	Inherited from: Scope: Global Type: AS MDN Signed
AS Signed	AS Signed	No	Inherited from: Scope: Global Type: AS Signed
AS Business Id	AS Business Id		No value provided
Allow documents with duplicate document ids	Allow documents with duplicate document ids	No	Inherited from: Scope: Global Type: Allow documents with duplicate document ids
Allow a TA1 request	Allow a TA1 request	Yes	Inherited from: Scope: Global Type: Allow a TA1 request
Discard Envelope if any errors	Discard Envelope if any errors	No	Inherited from: Scope: Global Type: Discard Envelope if any errors

Package: AS (N/A)

Figure 7-42 Current Connection attributes

7. Figure 7-43 on page 187 shows all the AS2 related parameters that can be updated. Provide a URL for the parameter AS MDN Http Url, for example:

```
http://wpgv6hub:57080/bcgreceiver/companye/edi_in
```

Also provide a value for the parameter AS MDN Email Address. Because we are doing AS2 and not AS1 communication, the e-mail address will not be used. But it is required to be entered in this field. Click **Save**.

Source

Company E

Package: None (N/A)
Protocol: EDI-X12 (ALL)
Document Flow: ISA (ALL)

Target

Company A

Package: AS (N/A)
Protocol: EDI-X12 (ALL)
Document Flow: ISA (ALL)

Scope Connection

Connection Summary

Package: AS (N/A)

Attribute	Description	Current Value	Inheritance	Update	Reset
Time To Acknowledge	Time To Acknowledge	30	Inherited from: **Scope:** Global **Type:** Time To Acknowledge		
Retry Count	Retry Count	3	Inherited from: **Scope:** Global **Type:** Retry Count		
AS Compress Before Sign	AS Compress Before Sign	Yes	Inherited from: **Scope:** Global **Type:** AS Compress Before Sign	Select one to update	
AS Compressed	AS Compressed	No	Inherited from: **Scope:** Global **Type:** AS Compressed	Select one to update	
AS Encrypted	AS Encrypted	No	Inherited from: **Scope:** Global **Type:** AS Encrypted	Select one to update	
AS MDN Http Url	AS MDN Http Url	No value provided	No value provided	ceiver/companye/edi_in	
AS MDN Email Address	AS MDN Email Address	No value provided	No value provided	mdn@companye.com	
AS MDN Asynchronous	AS MDN Asynchronous	Yes	Inherited from: **Scope:** Global **Type:** AS MDN Asynchronous	Select one to update	
AS MDN Requested	AS MDN Requested	Yes	Inherited from: **Scope:** Global **Type:** AS MDN Requested	Select one to update	
AS Message Digest Algorithm	AS Message Digest Algorithm	sha1	Inherited from: **Scope:** Global **Type:** AS Message Digest Algorithm	Select one to update	
AS MDN Signed	AS MDN Signed	No	Inherited from: **Scope:** Global **Type:** AS MDN Signed	Select one to update	
AS Signed	AS Signed	No	Inherited from: **Scope:** Global **Type:** AS Signed	Select one to update	
AS Business Id	AS Business Id	No value provided	No value provided	Select one to update	

Save Close

Protocol: EDI-X12 (ALL)

Figure 7-43 AS Connection attributes

Archived

Note: Figure 7-43 on page 187 shows a number of parameters to control encryption and digital signatures. These features are currently disabled in our setup. Do not activate them at this time, because we have not discussed the use of certificates and encryption keys. Chapter 8, “Securing the B2B exchange” on page 207 explains how to use these security features.

8. Click **Return** at the top to return to the window to manage connections.
9. Back in the manage connections window, select the reverse combination. That is, select **Company A** for Source and select **Company E** for the Target.
10. Activate these combinations and review the gateway settings and connection attributes. Because AS2 connection attributes are determined from the sender, you do not need to configure the same MDN Http Url and e-mail setting, that will have to be done on the side of Company A when he decides what he is sending to you. It is enough to just activate the channel so that this server knows to expect AS2 documents from Company A.

This completes the steps to set up WebSphere Partner Gateway at Company E. You must perform a similar setup for WebSphere Partner Gateway Advanced at Company A, as discussed in the following section.

7.7 Configuration tasks for the Company A hubadmin

The setup of WebSphere Partner Gateway Advanced on AIX is no different from the setup of WebSphere Partner Gateway Enterprise on Windows. In this section, we do not provide the detailed instructions on how to configure the server to complete our dual-communication example (Figure 7-44 on page 189). The interface and menu options are exactly the same. However, this section still explains the scenario from Company A’s point of view.

Archived

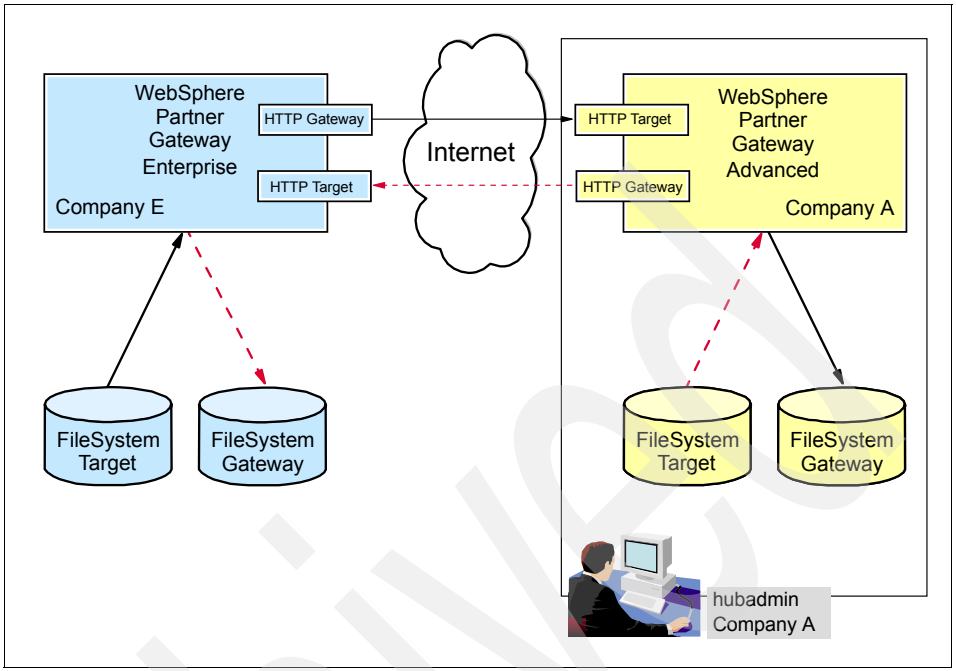


Figure 7-44 Logging on to hubadmin of Company A

As we did for Company E, the hub administrator of Company A now needs to log on to the WebSphere Partner Gateway server at his own site. The hubadmin of Company A also needs to complete the same actions, which are:

- Define Targets.
- Create interactions.
- Define the Community Manager and Community Participant.
- Define the participant connections between the two participants.

Two targets need to be created, an HTTP target and a file system target. The HTTP Target needs to listen at the same location that the administrator of Company A configured on his HTTP Gateway on Company E's server. That is, `http://m106984f:57080/bcgreceiver/companya/edi_in`. So the HTTP Target on this server would have a URI of /bcgreceiver/companya/edi_in. Refer to Figure 7-32 on page 176.

The second target is the directory that Company A's implementation of WebSphere Partner Gateway monitors to send EDI documents. When creating this target, make sure that the bcguser user has full access to this directory. You must also ensure that all higher level directories exist and are accessible to the user bcguser.

For example, if you choose the `/home/bcguser/wpgv6/data/companya/edi_out` directory for sending EDI documents, then you make sure that the `/home/bcguser/wpgv6/data/companya` directory exists and is accessible to bcguser. To avoid any security problems, create the directory structure you wish to use while logged onto the AIX system as bcguser.

After you create the target FileTarget a list of targets is displayed as shown in Figure 7-45.

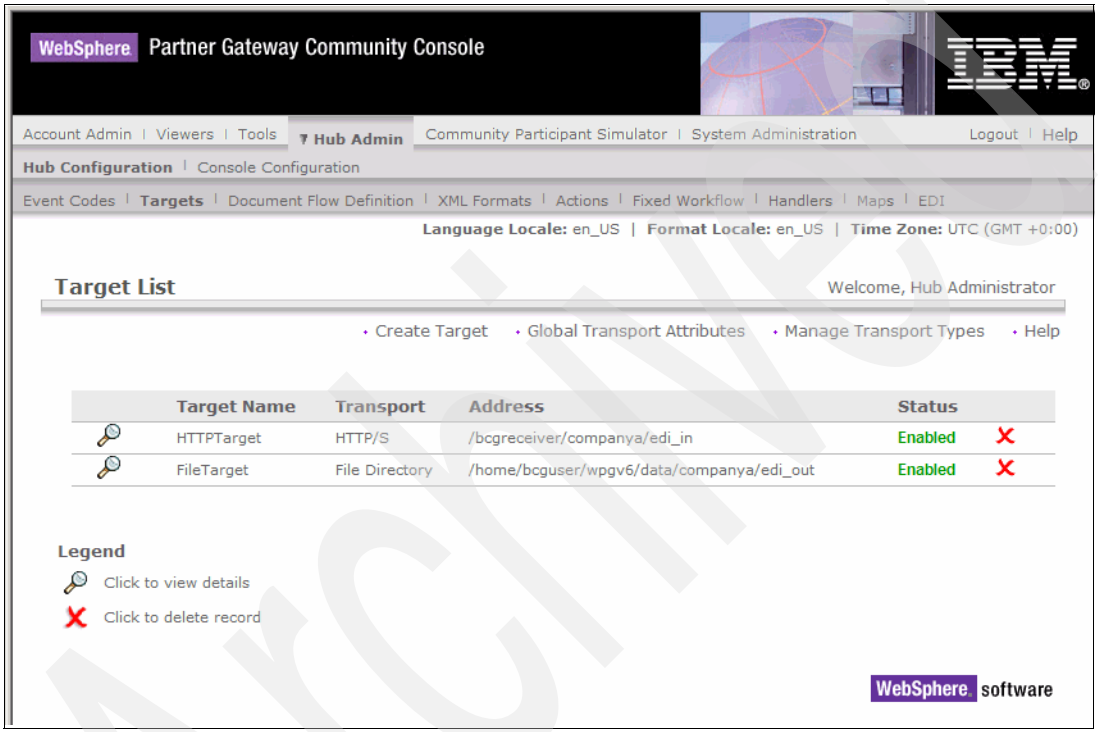


Figure 7-45 List of targets at company A

The creation of interactions is exactly the same as explained in 7.3.2, “Creating interactions” on page 149. Both implementations will have interactions that define AS2 packaged to unpacked EDI and unpackaged EDI to packaged AS2.

When that is complete, the hub administrator at Company A can then define the Community Manager for his implementation of WebSphere Partner Gateway.

The Company Login Name should be set to `companyA` and the business identifier should be set to `companya`. Remember that the `companya` identifier was already used on the setup of Company E’s server so the identifier needs to match.

After creating the Community Manager, the hub administrator creates the Community Participant that represents Company E. Here again, you must set the business identifier to `companye`, because that is already the identifier with which the Company E participant sees himself.

At this point we now have a reversal of the roles on the two implementations of WebSphere Partner Gateway. On Company E's server, Company E is the Community Manager and Company A is the Community Participant. On Company A's server it is the reverse; Company A is the Community Manager and Company E is the Community Participant. Figure 7-46 below shows how the participants are defined on the server at Company A.

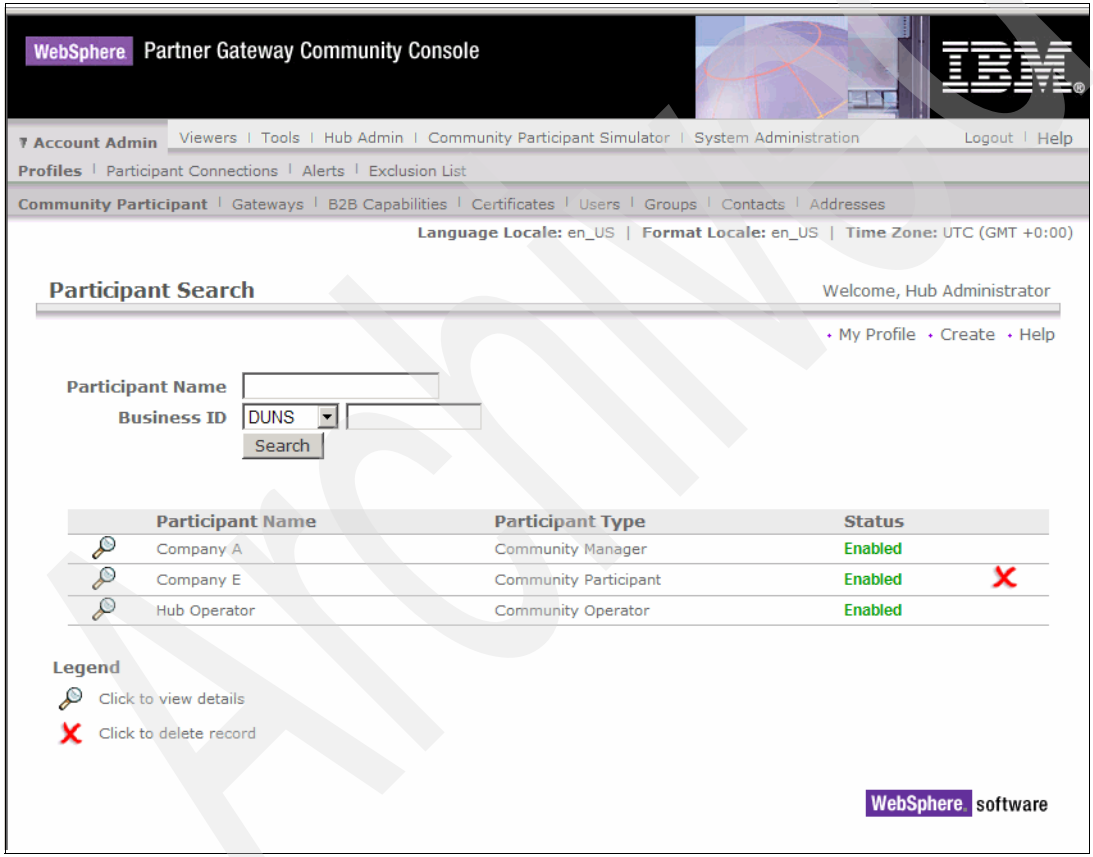


Figure 7-46 List of participants on Company A implementation

Still adhering to our role-based configuration model, the administrator for Companyt A Community Manager needs to log on and configure her B2B capabilities.

She also needs to create a gateway where the server can deliver incoming documents (Figure 7-47). Remember that this directory needs to exist, including the lowest level directory (`edi_in`, in this example). As before, the directory needs to be accessible to bcguser. The administrator of Company A creates the file system gateway with the value `file:///home/bcguser/wpgv6/data/companya/edi_in`. Make sure to set this gateway as the default gateway as well.

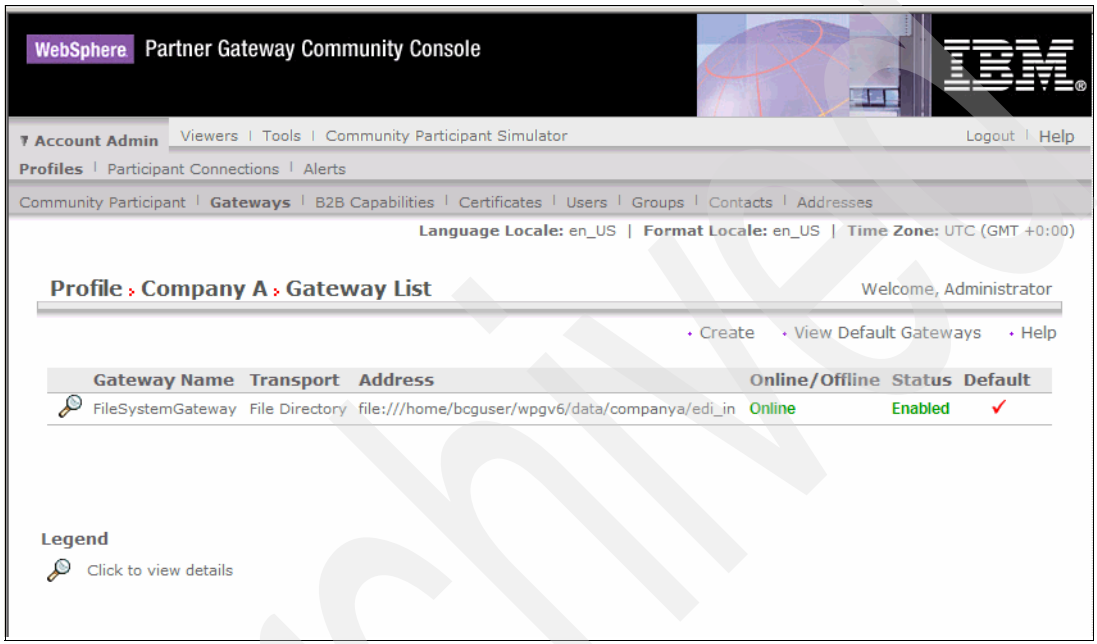


Figure 7-47 Creating the File System Gateway

Now it is the turn of the administrator of Company E to log on to Company A's server as the Community Participant. After he is asked to change his password, he needs to complete two actions:

- Provide his B2B capabilities.
- Create the gateway that points back to his own implementation of WebSphere Partner Gateway.

The value of the gateway, listed in Figure 7-48 on page 193, needs to be the same as the value for the HTTP Target of his own server. (see Figure 7-6 on page 146).

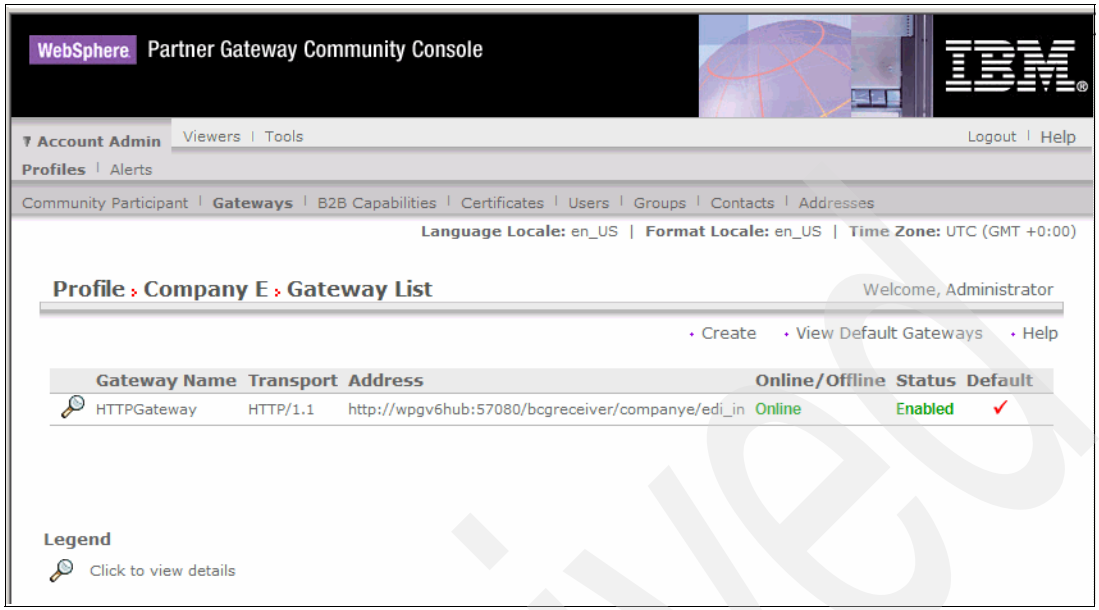


Figure 7-48 Company E as participant, creating the HTTP Gateway

After the administrators of the Community Manager and Community Participant have completed their tasks above, the hub administrator can log on again to perform the final step of tying the configuration pieces together with the participant connections.

Figure 7-49 on page 194 shows the activated connections where Company A is the source. You also need to activate the reverse connections as explained in 7.6, “Connecting Company E to Company A” on page 180.

Archived

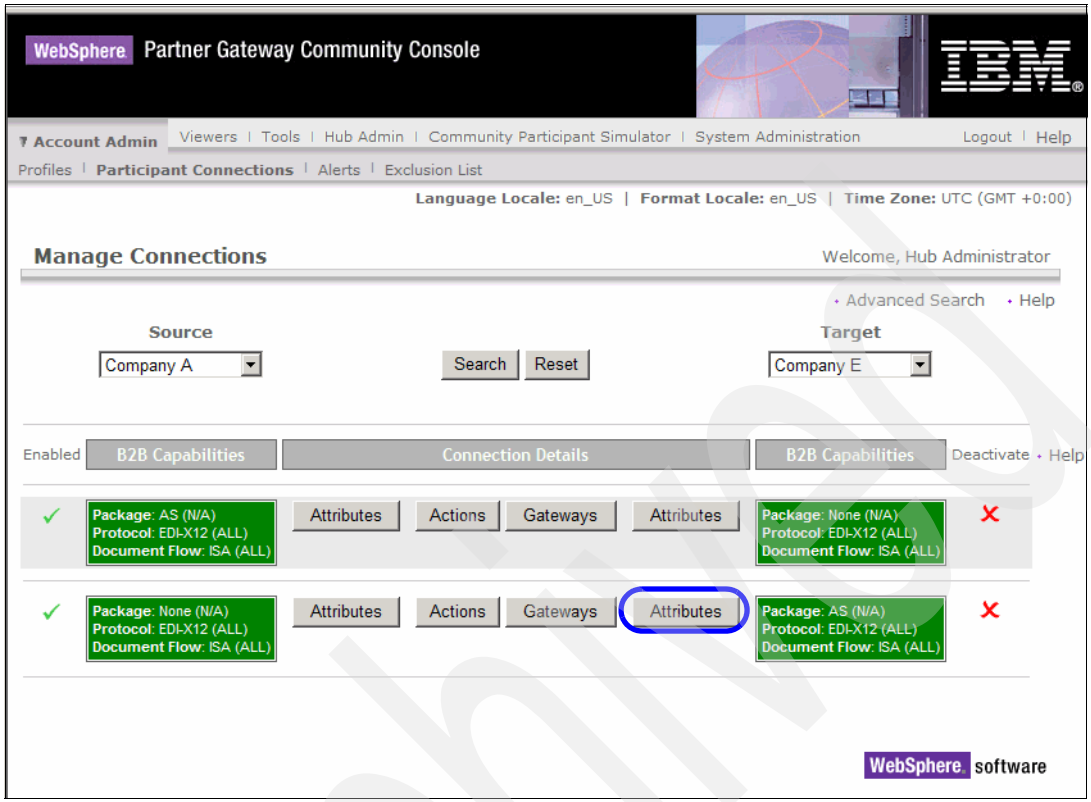


Figure 7-49 Activate connections at Company A between Company A and E

After activating the connections, review the gateways and update the following connection parameters for AS2 using the **Attributes** button in Figure 7-49.

- AS MDN Http Url
- AS MDN Email Address

Figure 7-50 on page 195 shows the update process for these attributes.

Archived

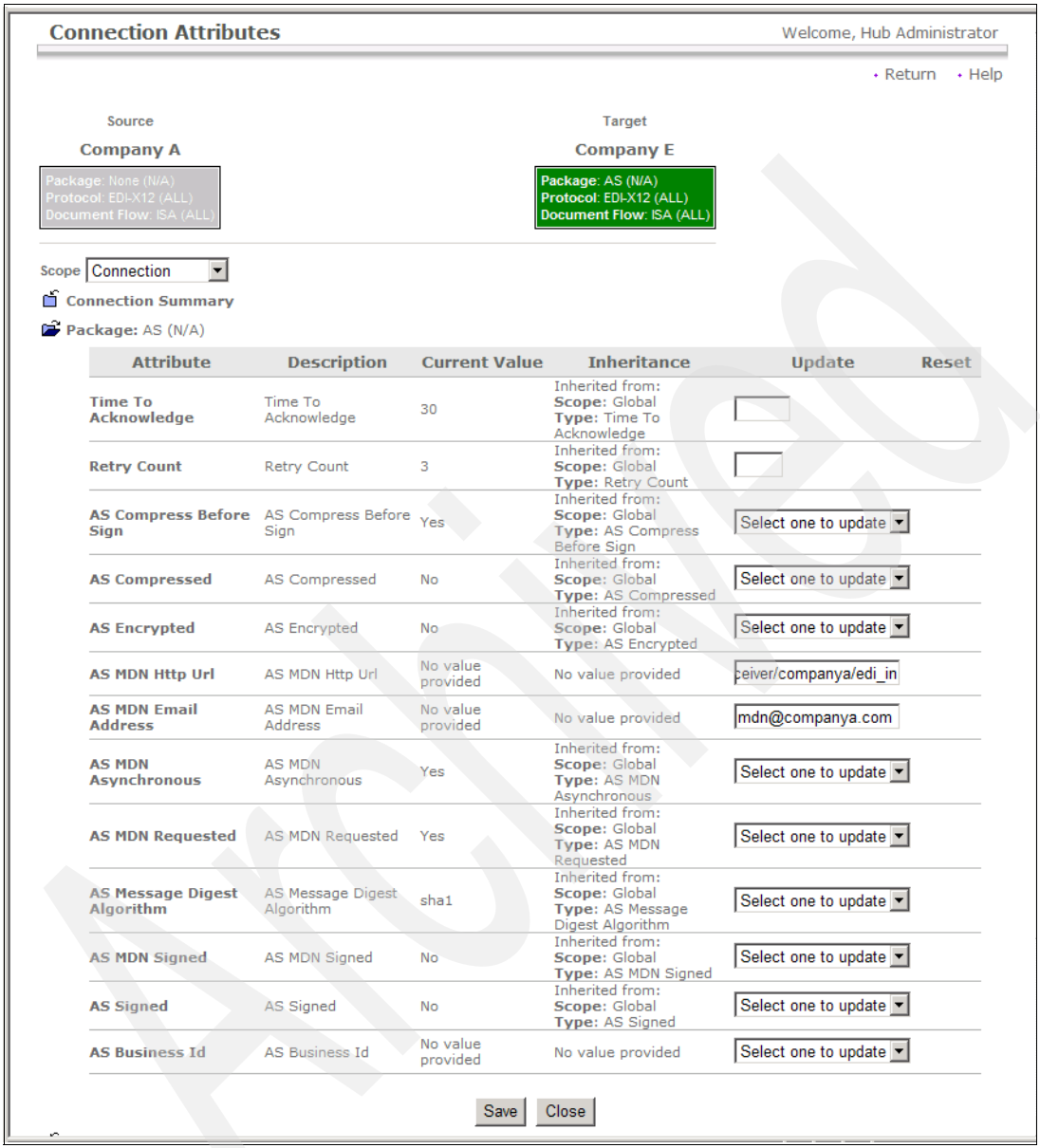


Figure 7-50 Connection attributes

This completes the setup for WebSphere Partner Gateway server owned by Company A. Now you can test the connection and begin to exchange documents between these two partners.

Archived

7.8 Validating communication

Before trying a sample exchange with a real document between the two companies, the hubadmin has one more tool that you can use to perform some validation. This tool can verify if the WebSphere Partner Gateway is able to reach a partner's configured gateway.

To access this tool:

1. Log on as hubadmin to Company E's server.
2. Select **Tools** → **Test participant connection.**
3. A number of selection options are shown in this tool (Figure 7-51). Select the Target Participant (Company A in this example) and the gateway you wish to test.
4. Select the **Command: POST** because it is an HTTP Gateway.
5. WebSphere Partner Gateway should load the URL. Click **Test URL** to verify the connection.

   If the test is successful, an HTTP code of 200 should be returned

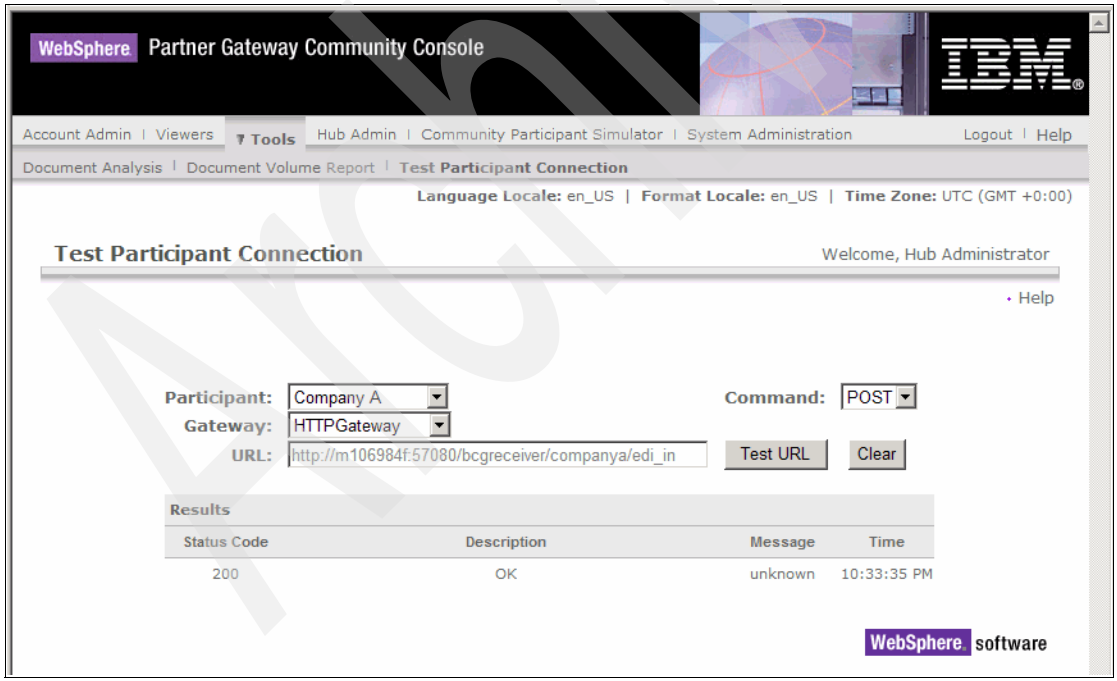


Figure 7-51 Test participant connection Tool

Archived

End-to-end validation is only possible by dropping an EDI file in the correct directory, verifying that it arrives at the partner machine, and verifying that the MDN was received.

Example 7-1 shows a sample EDI document. the Interchange Sender ID and Interchange Receiver ID in the ISA segment match the Business IDs as defined in the Participant profiles.

> **Note:** The EDI document in the example might look different from your common EDI file. In this pass through example, WebSphere Partner Gateway only looks at the business IDs to determine the sender and receiver, no other ISA parameters are evaluated at this time.

Example 7-1 Sample EDI file to be sent to Company A

```
ISA*  *          *  *          *  *companye      *  *companya      *050722*1517* *
*000000004* *P*:!
GS*PO*  *  *20050722*1517*4* *004010!
ST*850*0004!
BEG*06*SA*P345322**20050722!
N1*18*Ronan Dalton!
N2*Deliver To:*IBM Internationl Holdings Ireland!
N3*Damastown Industrial Estate!
N4*Dublin!
PO1**128*EA*10**BP*00021PO241!
PID*M**01*F*SCSI Card!
PO1**100*EA*10**BP*00031PO241!
PID*M**01*F*IDE Card!
PO1**128*EA*10**BP*00087P6821!
PID*M**01*F*PCI Card!
PO1**128*EA*10**BP*00021PO241!
PID*M**01*F*Deck of Cards!
SE*15*0004!
GE*1*4!
IEA*1*000000004!
```

When an ANSI X12 EDI file is placed in the directory \wpgv6\data\companye\edi_out\Documents\Production on the server of company E, the following events take place:

1. The WebSphere Partner Gateway Enterprise Receiver polls this directory for new messages.
2. When the message arrives, it is picked up by the Receiver and placed elsewhere in the file system for processing by the WebSphere Partner Gateway Document Manager.

3. The Document Manager parses the EDI envelope details to identify the participants in the transaction.
4. When the sender and receiver identifiers are determined, the Document Manager checks to see if a participant connection (channel) exists between the two identifiers it parsed out. If successful, it takes the appropriate action defined (in this case Pass through) and begins building the AS2 headers it needs to package the document in.
5. The message is then sent to the gateway defined on the target side of the participant connection.
6. Our gateway is defined as using HTTP, and points to the HTTP target of the WebSphere Partner Gateway Advanced installation at Company A's site.
7. The HTTP Target on the WebSphere Partner Gateway Advanced of Company A is listening for inbound messages sent over HTTP.
8. Again the Receiver component will pick up the message and place it on the file system for processing for by the Document Manager.
9. The Document Manager parses the MIME headers of the AS2 packaging and the EDI envelope data to determine the participants involved.
10. When the participants are known, the Document Manager determines the correct participant connection to use
11. In our case, the participant connection instructs the Document Manager to strip the AS2 packaging off and place the EDI document on the file system
12. The EDI transaction should now be seen in the /home/bcguser/wpgv6/data/companya/edi_in directory. The document name is not preserved in this type of transmission. so a typical filename upon receipt might be 23904832049323908234098232 4.vcm

In the real world, you probably will not have viewing access to their private file system in which they chose to have the document delivered. In the absence of physically checking the final destination directory, there are several tools within WebSphere Partner Gateway to review the communications and the exchange of documents between partners. Chapter 11, "Managing the B2B exchange" on page 307, discusses the options that are available to review communications.

For now, we focus on the AS1/AS2 viewer. This tool is available for the hub admin user, the admin user of the Community Manager and the admin user of the Community Participant. The Community Participant however would only be able to view documents that involved his own company. He could not view, for example, a communication between the Community Manager and another Trading Partner.

Note: This is one of the advantages of using role-based configuration. The admin users of the Community Participant have access to the tool in the Community Console to see transactions that involve themselves. So they can monitor their own documents. If you do not give them permission to log on to your implementation, they will have to monitor only the endpoint (gateway) they expect to receive documents on and might not be able to help with troubleshooting or delivery issues.

To use the AS1/AS2 viewer, select **Viewers** → **AS1/AS2 Viewer** in the menu bar. First you can provide filters to limit the output of the views. For example, you can specify a time range, limit the search to combination of partners (a Community Participant admin user will already be limited to only his own) or limit it to a certain gateway type. The default search options result in a list of all AS1/AS2 exchanges in the one-hour block you currently view.

Click **Search**, and you see a list of the of the AS1/AS2 communications that match the search arguments from the last page. Click the magnifying glass icon next to the AS1/AS2 communication that interest you. This results in the details of the exchange. An example is shown in Figure 7-52 on page 200.

The top section lists the main details about this exchange:

- Message ID
- Source and Target Participant
- Time Stamp
- Gateway Information
- MDN Information

Below that, you can see the incoming EDI document (package None) and packaged as AS. The lowest section covers the MDN, which is sent by Company A to Company E. Therefore, the roles of Company A and Company E are now reversed and Company A is listed as the source.

Archived

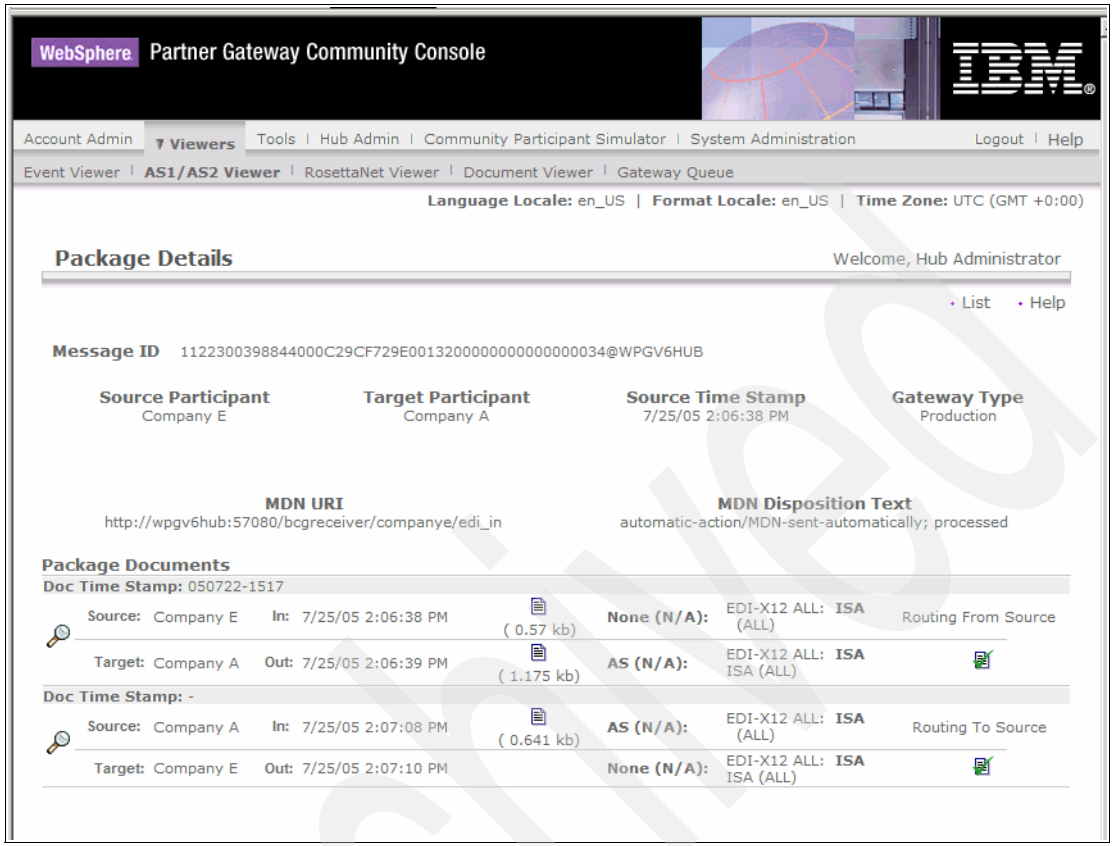
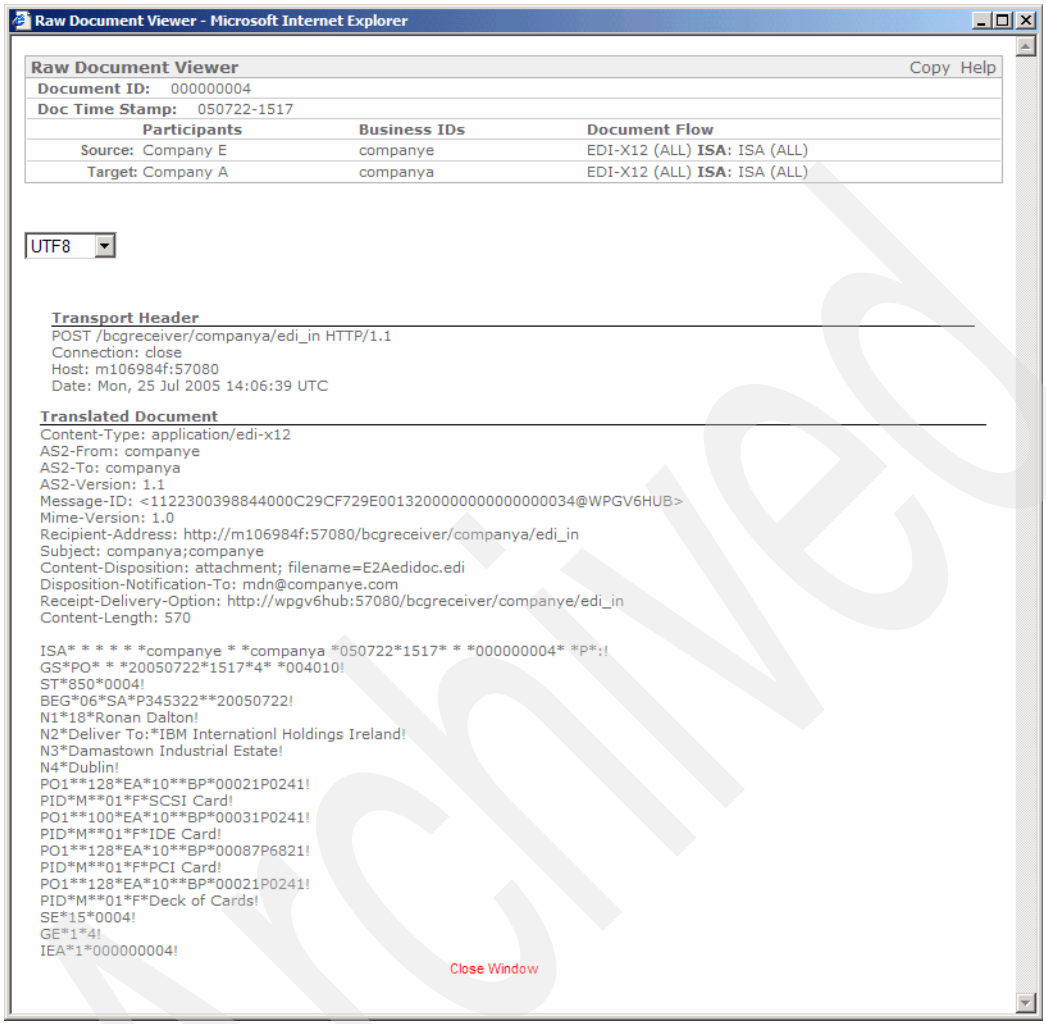


Figure 7-52 Details of an AS2 exchange

From within Figure 7-52, you can review the source document, the document as it is sent to the partner, and the MDN document. To review the HTTP and AS2 header information, click the document icon for the AS packaged document, the middle one of the three document icons. A window opens that shows these details. See Figure 7-53 on page 201.

Archived



Raw Document Viewer - Microsoft Internet Explorer

Raw Document Viewer Copy Help

Document ID: 000000004

Doc Time Stamp: 050722-1517

Participants	Business IDs	Document Flow
Source: Company E	companye	EDI-X12 (ALL) **ISA:** ISA (ALL)
Target: Company A	companya	EDI-X12 (ALL) **ISA:** ISA (ALL)

UTF8

Transport Header

```
POST /bcgreceiver/companya/edi_in HTTP/1.1
Connection: close
Host: m106984f:57080
Date: Mon, 25 Jul 2005 14:06:39 UTC
```

Translated Document

```
Content-Type: application/edi-x12
AS2-From: companye
AS2-To: companya
AS2-Version: 1.1
Message-ID: <1122300398844000C29CF729E0013200000000000000034@WPGV6HUB>
Mime-Version: 1.0
Recipient-Address: http://m106984f:57080/bcgreceiver/companya/edi_in
Subject: companya;companye
Content-Disposition: attachment; filename=E2Aedidoc.edi
Disposition-Notification-To: mdn@companye.com
Receipt-Delivery-Option: http://wpgv6hub:57080/bcgreceiver/companye/edi_in
Content-Length: 570

ISA* * * * * *companye * *companya *050722*1517* * *000000004* *p*:!
GS*PO* * *20050722*1517*4* *004010!
ST*850*0004!
BEG*06*SA*P345322**20050722!
N1*18*Ronan Dalton!
N2*Deliver To:*IBM Internationl Holdings Ireland!
N3*Damastown Industrial Estate!
N4*Dublin!
PO1**128*EA*10**BP*00021P0241!
PID*M**01*F*SCSI Card!
PO1**100*EA*10**BP*00031P0241!
PID*M**01*F*IDE Card!
PO1**128*EA*10**BP*00087P6821!
PID*M**01*F*PCI Card!
PO1**128*EA*10**BP*00021P0241!
PID*M**01*F*Deck of Cards!
SE*15*0004!
GE*1*4!
IEA*1*000000004!
```

Close Window

Figure 7-53 AS2 packaged EDI document

In the same way, you can inspect the MDN that was returned by Company A. See Figure 7-54 on page 202 for the MDN that was sent in response to the AS2 document we just looked at in Figure 7-53.

Archived

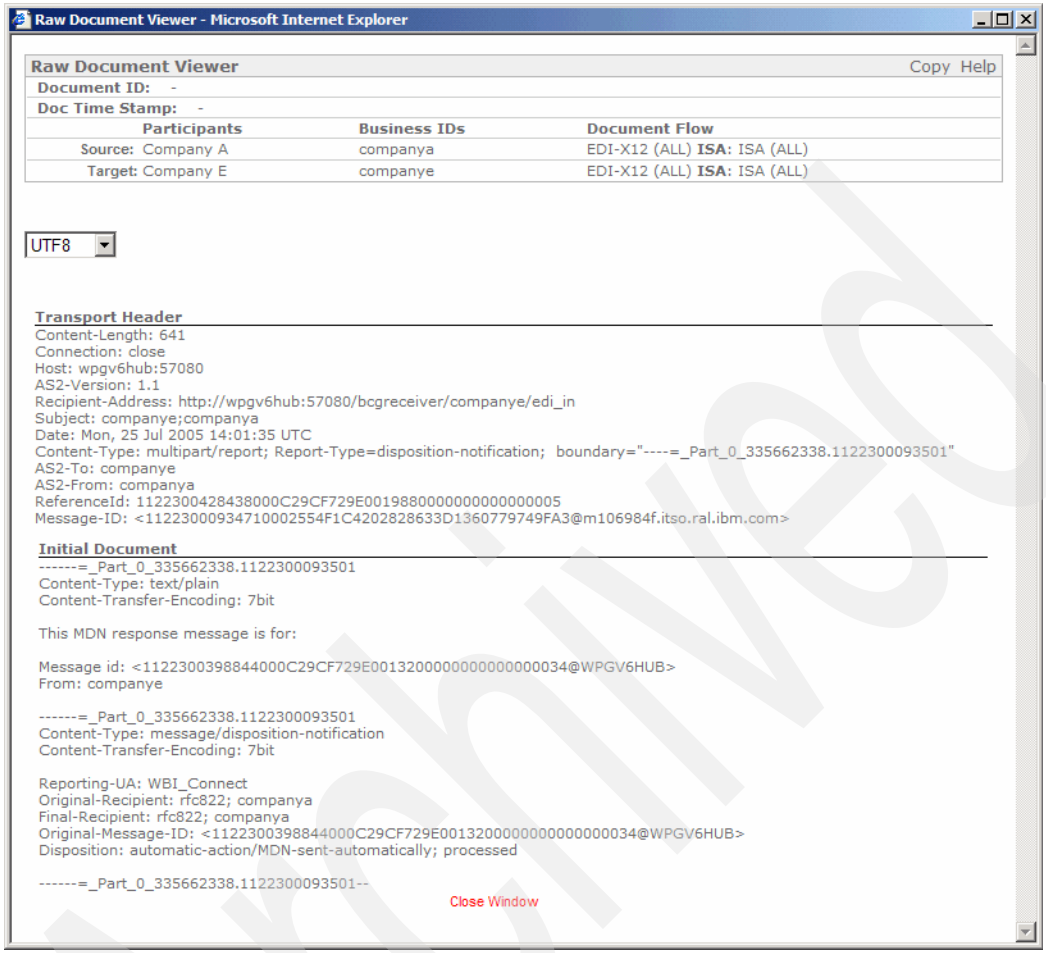


Figure 7-54 MDN received from Company A

7.9 Revisiting role-based configuration

During the configuration of WebSphere Partner Gateway on both of the machines in the previous sections, we stressed the role-based approach and what can be done by each user. However, this is not the only approach that you can use. The hubadmin user of the system can perform *all* the roles we described, and edit the profile of both participants himself.

After you log on as hubadmin, you see the Participant Search window. Click **Search** to display an unfiltered list of defined participants (Figure 7-55 on page 203).

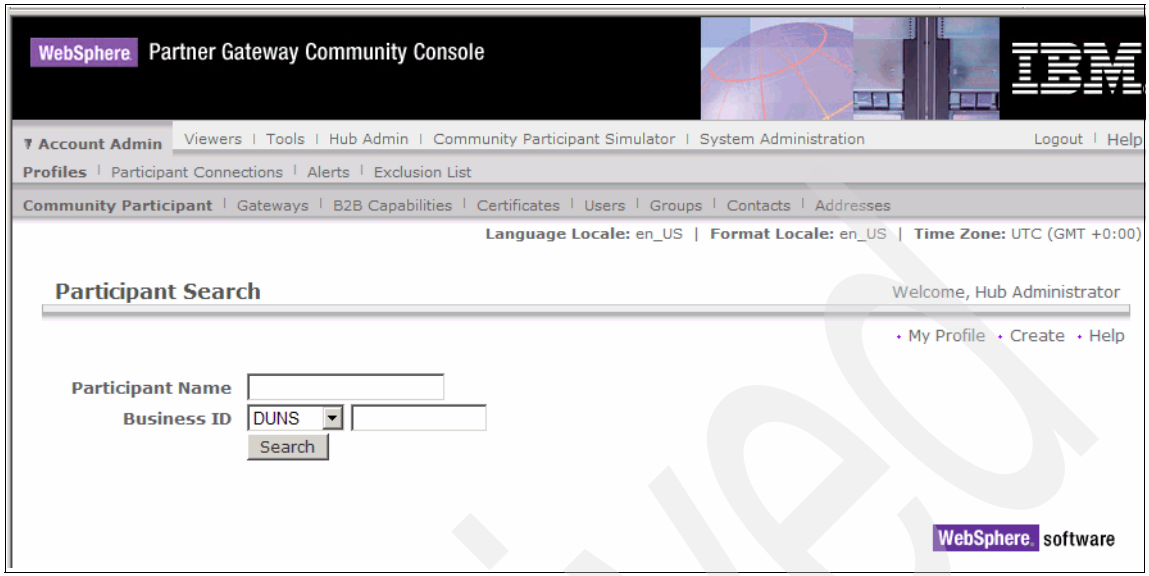


Figure 7-55 Searching for participants

When the list of defined participants is shown, click the **magnifying glass** icon. In Figure 7-56 on page 204 next to the profile of Company A, for example. From that point on, the hubadmin works with the profile of that Participant as though the hubadmin was the admin user of Company A.

Archived

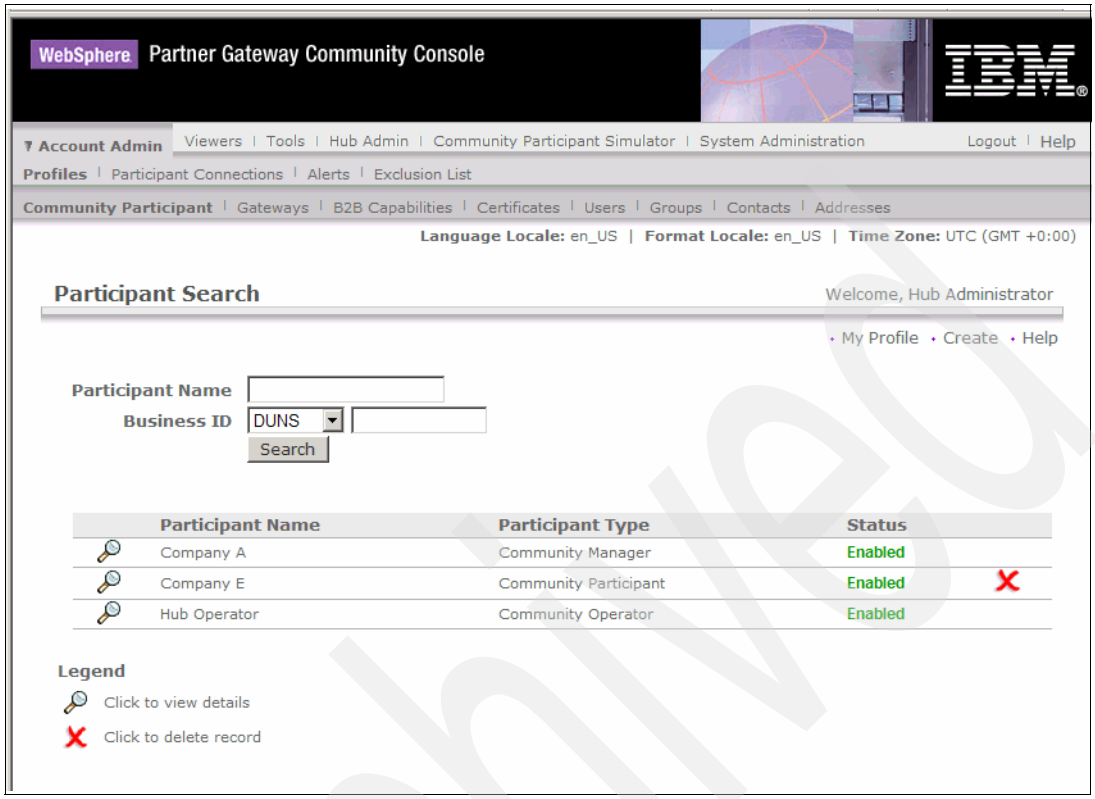


Figure 7-56 List of participants

When the profile is shown (Figure 7-57 on page 205), the hubadmin can now view and edit all elements of the profile of Company A. For example, as the admin user of Company A, you had to define your gateway.

Archived

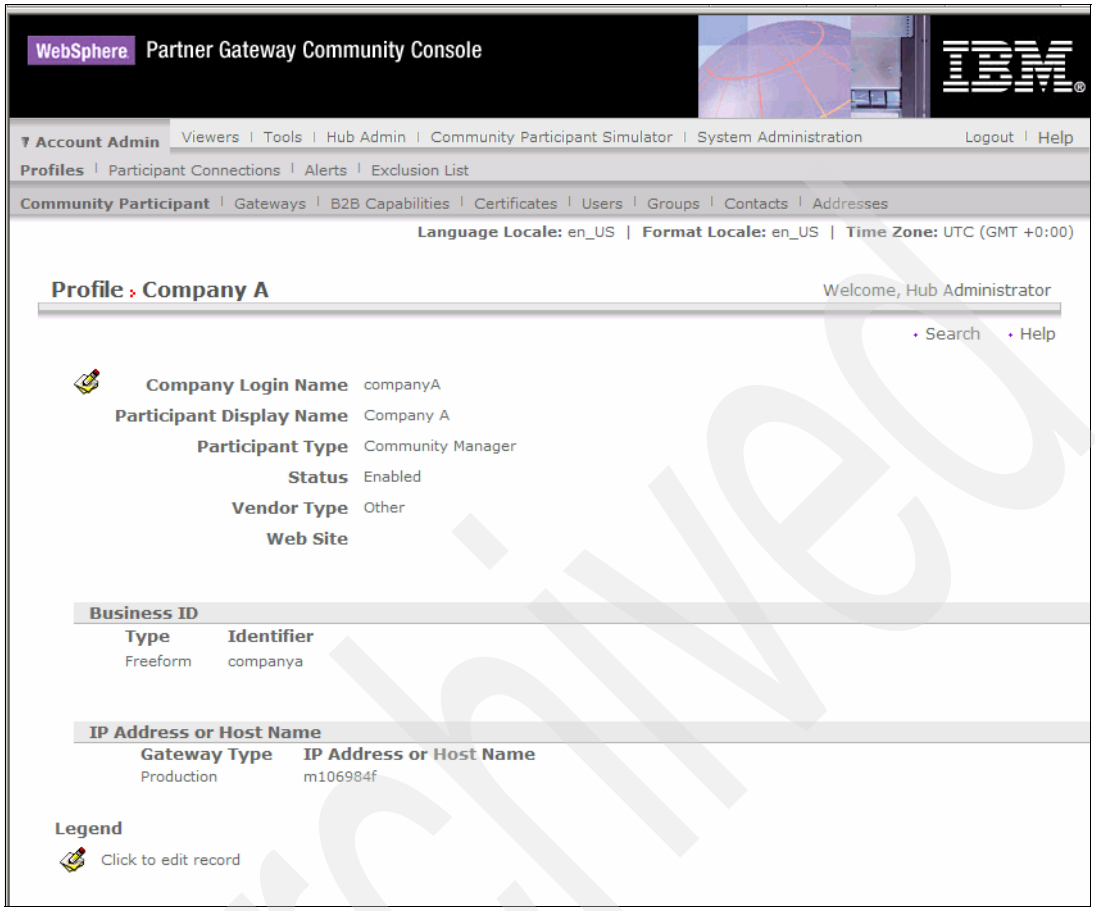


Figure 7-57 Profile of Company A inspected by the hubadmin user

If the user hubadmin selects the **Gateways** menu option in Figure 7-57, they see the gateway that was defined previously by the admin user of Company A when he was logged on as the Community Participant. The hubadmin user can create additional gateways, or change the properties of this existing gateway by clicking the **magnifying glass** icon next to the gateway (see Figure 7-58 on page 206).

Archived

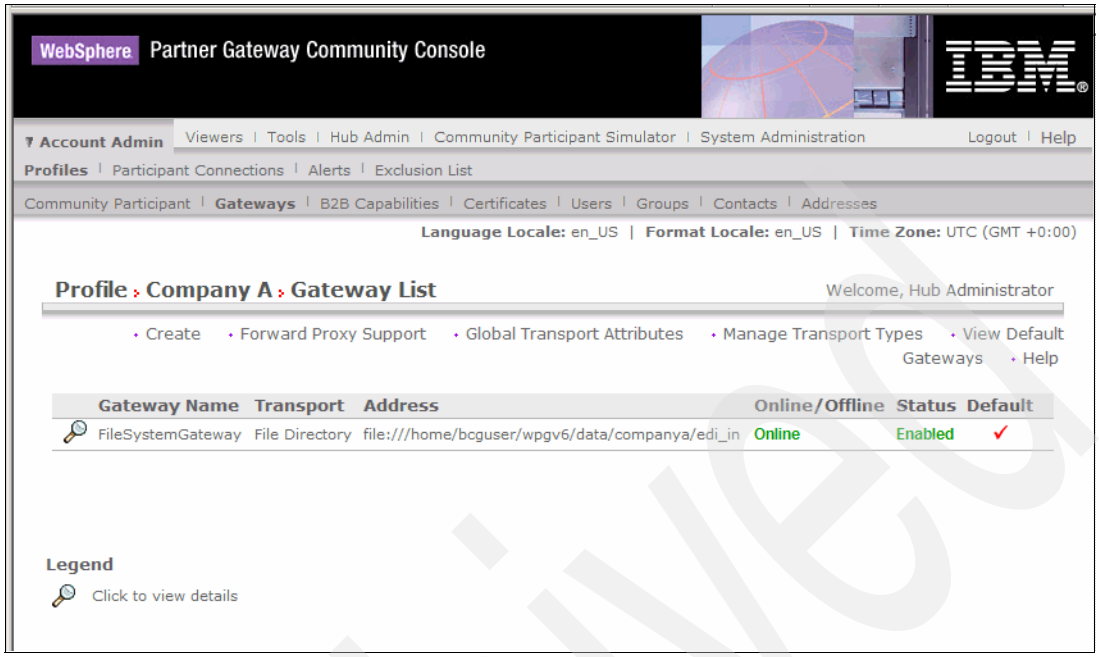


Figure 7-58 Gateways of Company A reviewed by user hubadmin

However, the **Gateways menu option** you arrived at in Figure 7-58 is different from the one you would arrive at if you click the **Gateways tab** without going through the process of selecting a Participant first as in Figure 7-55 on page 203. If hubadmin selects the Gateways link without first selecting an individual Participant, it will default to the Operator Participant to which the hubadmin belongs. This Operator profile should not have any gateways defined because it does not interact with any Business Partners or internal systems. It is a special case that houses the hubadmin user, and manages some other aspects of the overall server.

Archived

8

Securing the B2B exchange

Chapter 7, “Creating a basic B2B exchange” on page 137, describes the implementation of the exchange of EDI documents using AS2. However, this exchange lacked the usual security aspects of a typical AS2 configuration. This chapter shows you how to add encryption and digital signatures so that the exchange of the documents is performed in a secure way.

© Copyright IBM Corp. 2005. All rights reserved.

8.1 What is needed to perform encryption an decryption

Encryption is a generic term describing a collection of algorithms that scramble data so that only the intended receiver of the data can unscramble or *decrypt* the data. As a basic requirement, encryption is a reversible process. This is not a characteristic of digital signatures. The creation of a digital signature results in a stream of bytes from which you cannot rebuild the original text.

Encryption algorithms can roughly be categorized into two classes. The first class uses symmetric algorithms. Symmetric key algorithms require the key to be shared between two parties before encryption can take place.This creates an opportunity for the key to be intercepted and used by a third party to eavesdrop on the secure communications. A good example of a symmetric algorithm is DES or Triple DES.

The second class uses asymmetric algorithms or public key algorithms. Public key cryptography, by contrast, allows the public part of the key pair to be publicly available. This is because it is impossible to determine the private key based on the public key.

For example, if Alice wants to send an encrypted message to Bob:

1. Alice looks up Bob's public key, encrypts the message, and sends it to Bob.
2. Bob then decrypts the message using his private key.

Because no secret information needs to be shared between the two parties, public key cryptography is considered more secure. On the down side, public key cryptography is much more computation-intensive than symmetric key cryptography. Typically, a mixture of the two is used in practice.

A public key system is used to exchange a one-off symmetric key that the parties use to encrypt their subsequent conversation. A temporary session key is generated by the sender. It is sent to the receiver after encryption with the public key of the receiver using public key cryptography. Thus, the session key can only be obtained by the receiver, who is the owner of the matching private key. The actual data is encrypted using the temporary key using a symmetric algorithm. This two-step process is shown schematically in Figure 8-1 on page 209. The fact that the use of different keys still results in a reversible process is a consequence of the fact that these algorithms are built upon a prime number theory. The generation of public and private keys relies on the knowledge of large prime numbers.

The main advantage of public key cryptography is its level of security. The parties that exchange documents using these algorithms do not share a common piece of data, which reduces the risk of security breaches. The main disadvantage of public key cryptography is performance. In a B2B environment, it is not

uncommon to send large documents of several megabytes. Encrypting these kinds of documents using public key cryptography requires a significant amount of CPU resources.

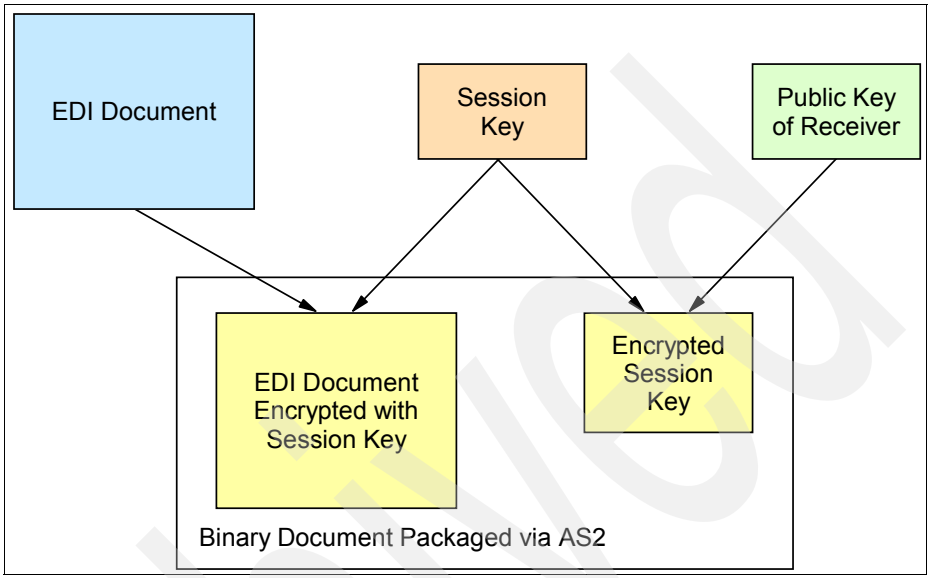


Figure 8-1 Encrypting EDI document using symmetric/asymmetric algorithms

When the Trading Partner receives the encrypted document:

1. First the Trading Partner extracts the encrypted session key and decrypts it with his private key.
2. When he obtains the session key, he uses it to decrypt the actual EDI document, as shown schematically in Figure 8-2 on page 210.

Because the session key can be obtained only by the owner of the private key, the EDI document can be decrypted only by the intended receiver.

Archived

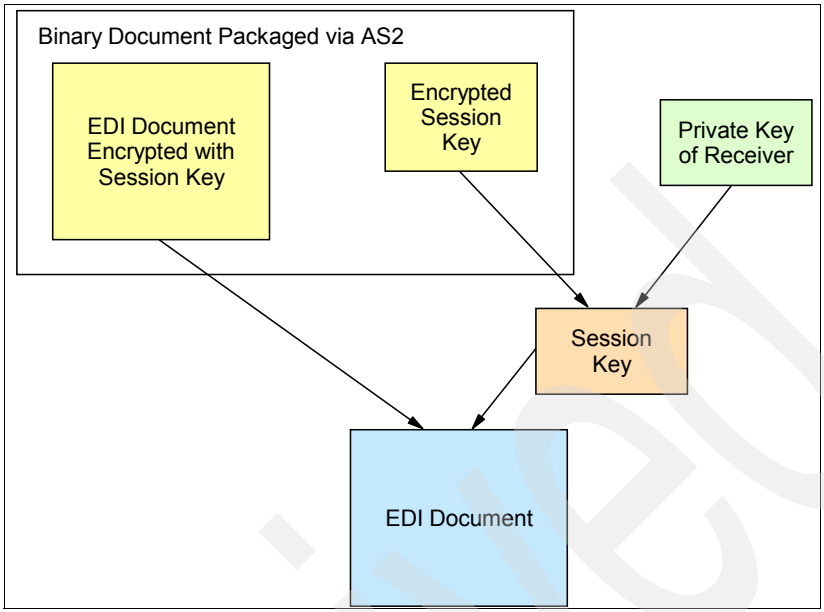


Figure 8-2 Decrypting the session key and EDI document

We now return to the configuration of WebSphere Partner Gateway for Company A and Company E (Figure 8-3 on page 211). When sending an EDI document over AS2 from Company E to Company A, the following arrangement is required:

- Company E needs the public certificate of Company A to encrypt.
- Company A uses its own private certificate to decrypt.

Archived

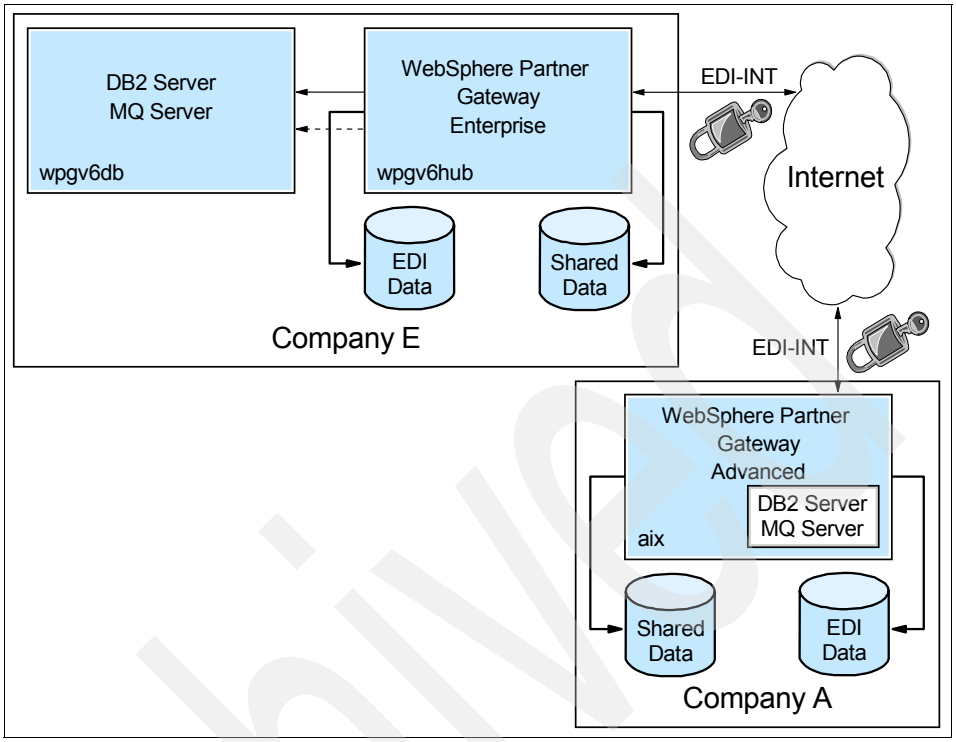


Figure 8-3 Scenario Overview

8.2 Enabling encryption

Enabling encryption for AS2 consists of several steps:

1. Generate a private/public key pair.
2. Upload the private key to the server that is owned by the private key owner, so that the server can decrypt documents.
3. Upload the public certificate to the server of the trading partner, so that the documents can be encrypted before being sent to the owner of the public certificate.
4. Update the AS2 connection parameters to request encryption.

These steps are required for both partners if you want to send and receive encrypted documents over AS2 between the two partners.

Archived

8.2.1 Company E generates a public/private key pair

WebSphere Partner Gateway provides a tool to generate a public/private key pair. This tool, called *ikeyman,* can be used to create self-signed certificates. However, it is quite common to request certificates from an external organization that is considered trust worthy by the business partners. Such an external organization is sometimes called a *certificate authority* or *root certificate authority.*

To open the ikeyman tool, navigate to the `install_dir\was\bin` directory and execute the ikeyman.bat file.

When the ikeyman tool opens, generate a new self-signed certificate:

1. From the Key Database File menu, click **Open**.
2. In the open window (Figure 8-4) complete these tasks:
   a. Click **Browse**.
   b. Navigate to the `install_dir\common\security\keystore` directory and select the **receiver.jks** file. Click **Open**.
   c. In the Open window click **OK**.

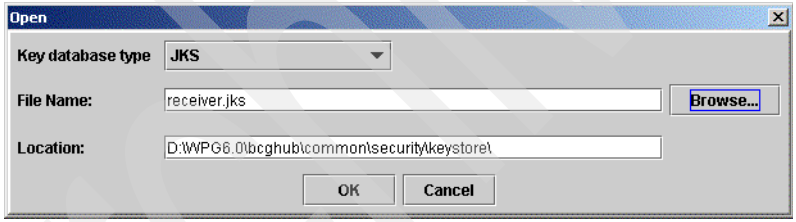


Figure 8-4 Opening the existing keystore

3. At the Password prompt enter `WebAS`.
4. In the IBM Key Management window (Figure 8-5 on page 213), delete the WebSphere dummy server certificate.

Archived

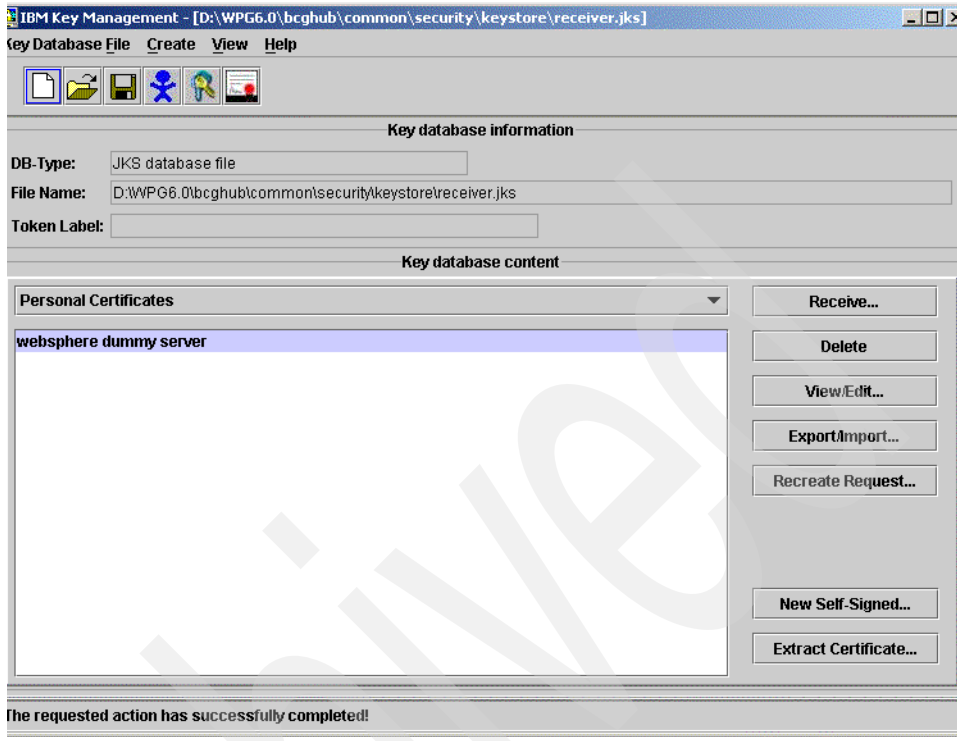


Figure 8-5 Key store containing dummy WebSphere key

5. Click **New Self-Signed**.
6. In the Create New Self-Signed Certificate window (Figure 8-6 on page 214), complete the details of the new certificate:
   a. Provide a label for the key, for example, `CompanyE_SelfSigned`.
   b. Provide a common name, for example, `wpgv6hub.itso.ral.ibm.com`.
   c. Provide a name for the organization, for example, `Company E`.
   d. Provide values for optional fields if applicable.
   e. Click **OK** to generate the certificate.

Archived

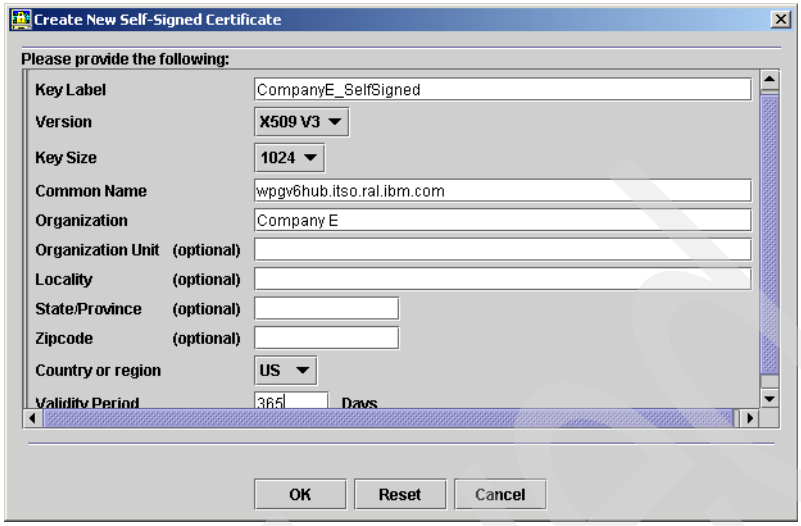


Figure 8-6 Creating a new self signed certificate

Next we must extract a public certificate from the new self-signed certificate. The WebSphere Partner Gateway Advanced server of Company A uses this public certificate to encrypt messages that are sent to theWebSphere Partner Gateway Enterprise of Company E. This means that at some point we need to upload this public certificate to the server of our partner. Because only the owner of the private key, Company E, can decrypt the messages sent by Company A, we know then that confidentiality is assured (Figure 8-7 on page 215).

1. In the IBM Key Management window (Figure 8-7 on page 215), click **Extract Certificate**.

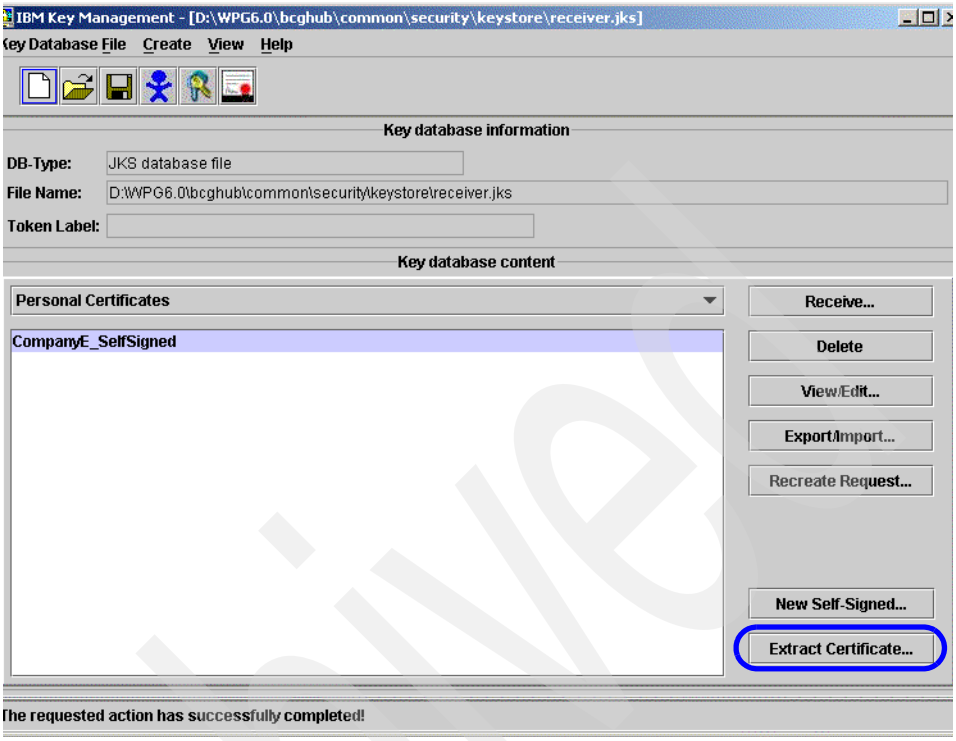


Figure 8-7 Extract certificate of key store (1 of 2)

2. In the Extract Certificate to a File window (Figure 8-8), complete the following tasks:

   a. For Data type, select **Binary DER data**.
   b. For certificate file name, type `CompanyEPublicCertificate.der`
   c. Click **OK**.

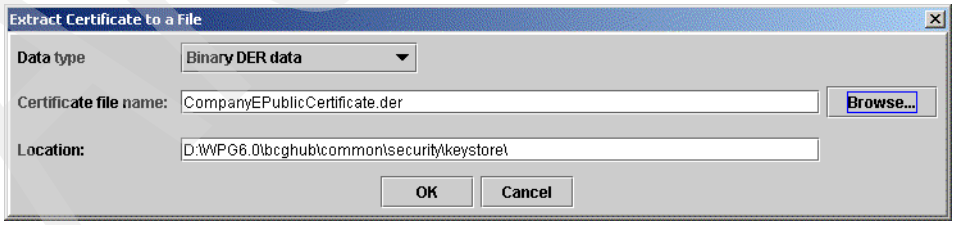


Figure 8-8 Extract certificate of key store (2 of 2)

The public certificate is extracted in a form that can be used by WebSphere Partner Gateway.

Now we must export a private key from the newly created self-signed certificate. WebSphere Partner Gateway Enterprise of Company E uses this to decrypt encrypted transactions.

1. From the ikeyman tool (Figure 8-5 on page 213), click **Export/Import**.
2. In the Export/Import Key window (Figure 8-9), complete the following tasks:
   a. For Key file type, select **PKCS12**.
   b. For the File name, type `CompanyEPrivateCertificate.p12`.
   c. Click **OK**.

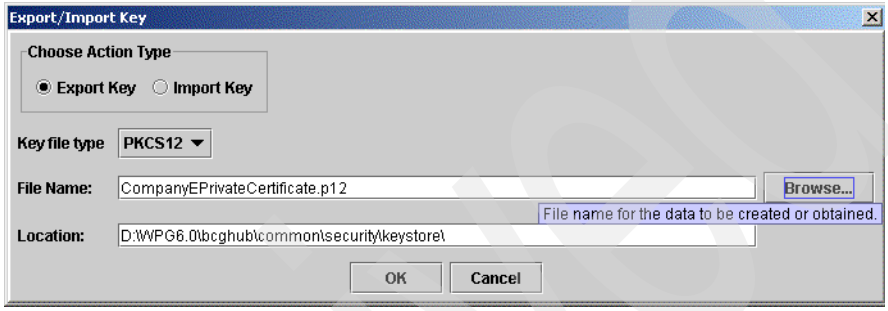


Figure 8-9 Export a key

3. Because this file contains our private key, we must protect it well. To avoid unauthorized use, the contents of this file are protected by a password. Any person who wants to use this file in an encryption/decryption solution has to know this password before such an encryption/decryption solution accepts the file.

   In the Password Prompt window (Figure 8-10), enter a password to protect the target PKCS12 file and click **OK**. Be sure to make a note of this password, because it is required later.

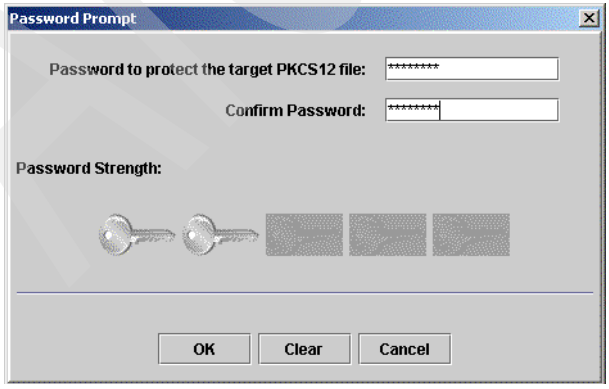


Figure 8-10 Provide a password to protect the private key

We are now finished working with the key management utility and can use the exported certificates with WebSphere Partner Gateway.

8.2.2 Company E uploads private key to its own server

We must now import the private key file into the WebSphere Partner Gateway Enterprise server that is owned by Company E. Remember that this private key is used when decrypting documents sent by the partners of Company E, such as Company A.

To upload the PKCS12 file, follow these steps:

1. Log in to WebSphere Partner Gateway as the Hub Admin.
2. Select **Account Admin** → **Profiles** → **Certificates**.
3. Select **Load PKCS12** as shown in Figure 8-11.

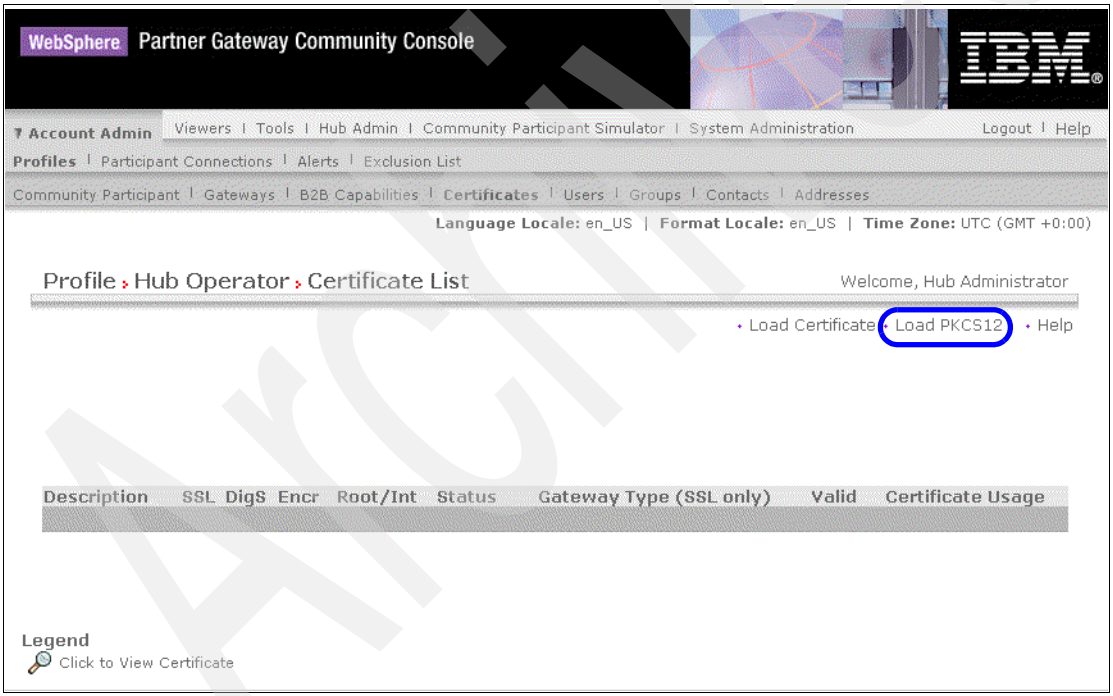


Figure 8-11 List of Certificates

4. In the Create New PKCS12 Certificate panel (Figure 8-12 on page 218) complete the following tasks:
   a. For Certificate Type, select **Encryption**.

b. Provide a description for the certificate.

c. For Status, select **Enabled**. Do not forget this, by default the status is set to Disabled.

d. Click **Browse** and navigate to the directory in which the PKCS12 file is stored, for example installL_Dir\common\security\keystore. Select the file and click **Open**.

e. Enter the password used when exporting the PKCS12 file earlier (see Figure 8-10 on page 216).

f. Click **Upload**.

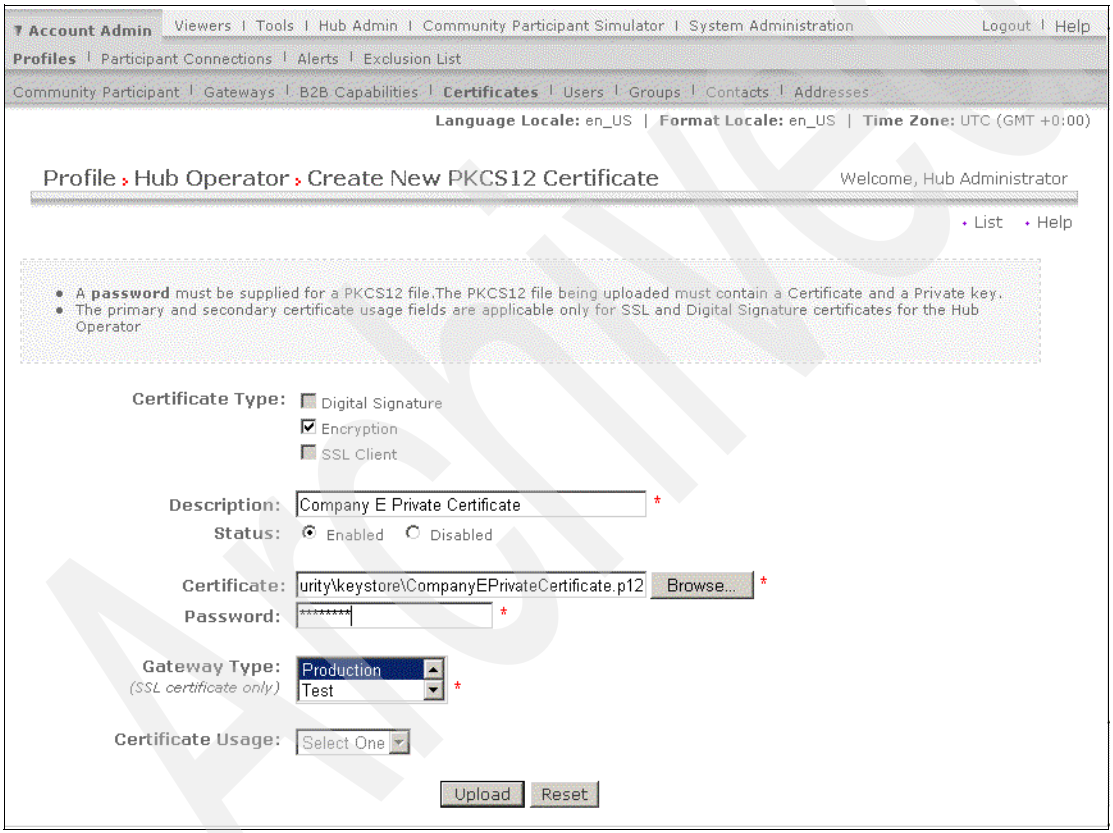


Figure 8-12 Loading the private key file

5. The server now reviews the certificate and presents information about it, as shown in Figure 8-13 on page 219 and Figure 8-14 on page 219.Review this to make sure that the information is correct. Then click **Save**.

Profile › Hub Operator › Create New PKCS12 Certificate Welcome, Hub Administrator

• List • Help

Certificate Type: ☐ Digital Signature
☑ Encryption
☐ SSL Client
☐ Root and Intermediate

Description: Company E Private Certificate

Status: Enabled

Gateway Type: Production

Save Cancel

Certificate Information

General Information

Status: This certificate is valid

Issued To: wpgv6hub.itso.ral.ibm.com

Valid: 7/25/05 through 7/25/06

Figure 8-13 Saving the uploaded certificate after review (1 of 2)

Detail Information

Version: 3

Serial Number: 42E5 670D

Valid From: Monday, July 25, 2005 10:26:21 PM GMT+00:00

Valid To: Tuesday, July 25, 2006 10:26:21 PM GMT+00:00

Fingerprint Algorithm: MD5

Fingerprint: 7203 3FDF 936E C045 EC15 10ED 7469 0F2C

Public Key:
3081 9F30 0D06 092A 8648 86F7 0D01 0101 0500 0381 8D00 3081 8902 8181 008E 588C 628A F15B 730F
4C3A 5A52 1DFF 3A62 82D4 0FAF 8D82 E482 A27D 9092 16B6 6CD3 1725 E7E2 5E3F B0A2 EAE5 A580 89B7
90E6 02EE 5AA1 7FD3 1FBB CF9D 19F9 328D 451F 911F 786E 5C22 6747 9BBA B9C7 E23F 2100 EA04 1315
0EAB EEAF 1FBA 6AC3 CA2C 7916 FBDA E6C3 DC0B 32AE 02D8 837C A601 6B99 25EB FF1A 1A28 5323 1E11
C07F D51A 3102 0301 0001

Subject Information

Common Name (CN): wpgv6hub.itso.ral.ibm.com

Organization Unit (OU):

Organization (O): Company E

Locality (L):

State (ST):

Country (C): US

Figure 8-14 Saving the uploaded certificate after review (2 of 2)

6. Click **List** to return to the list of certificates that now contains the new entry. We see the list in Figure 8-15 on page 220.

Archived

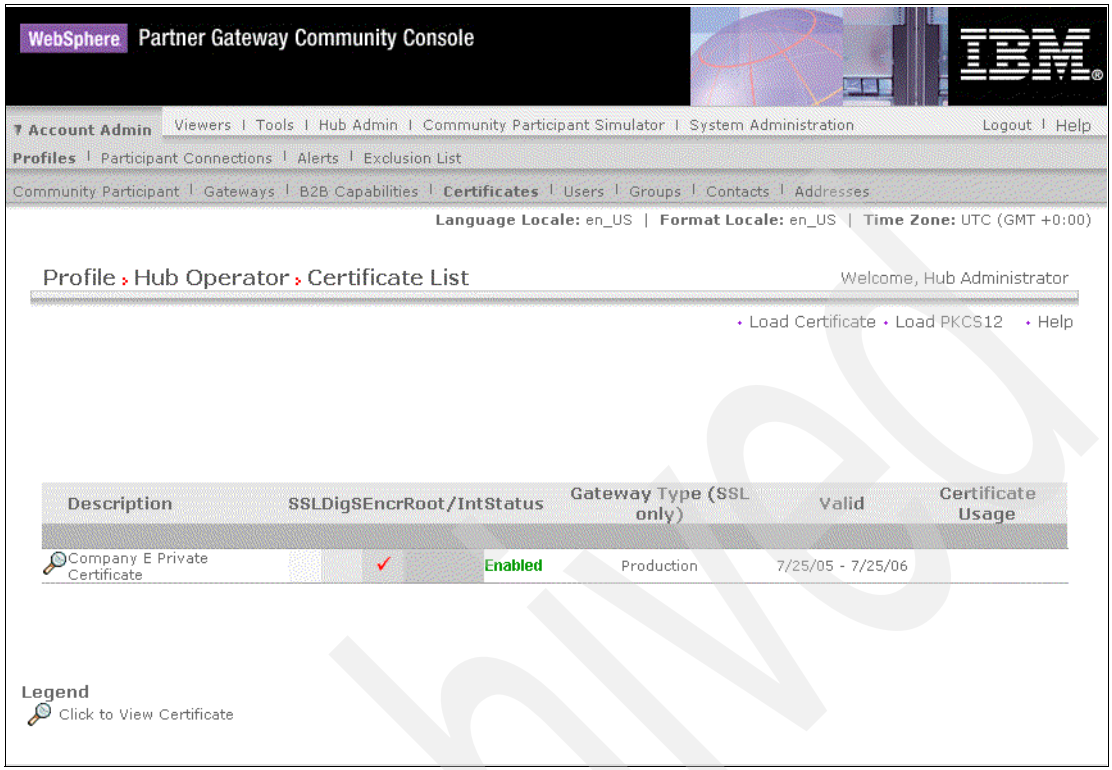


Figure 8-15 Certificate uploaded

At first, it might seem odd that the hub administrator has to upload the private key of Company E. Intuitively, we would expect that the administrator of Company E performs that task. However, if we think about the processing that happens when an encrypted file is received, it sounds logical that the hub administrator owns the private certificate.

When an encrypted document is received on an HTTP target, the WebSphere Partner Gateway server cannot know the Target Participant because the document is encrypted. The server needs to look for the certificate in a generic place, such as the operator profile. When the document is decrypted, the server determines what the target business entity is.

8.2.3 Company E uploads public certificate to partner's server

The next step is to upload the public certificate to the partner's server. This time the administrator of Company E logs on to the server of Company A. Company A uses this key to encrypt any documents targeted for Company E.

1. Log in as the Community Participant on the partner's server of Company A.
2. Select **Account Admin** → **Profiles** → **Certificates**.
3. In the Certificate List panel (Figure 8-16), click **Load Certificate**.

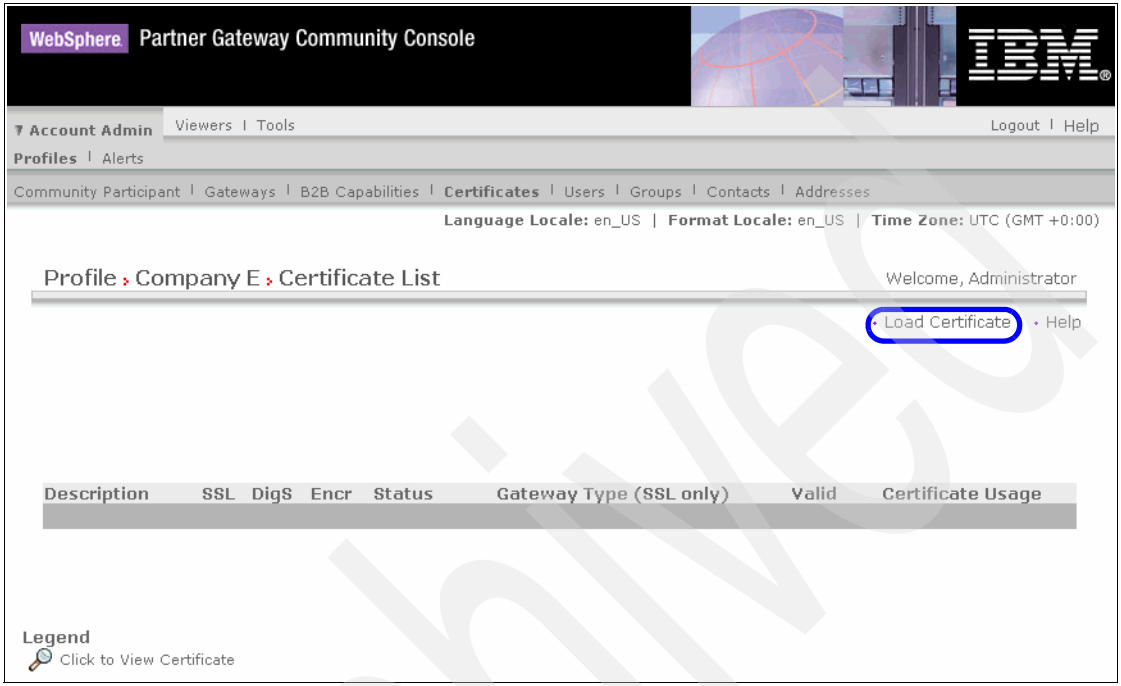


Figure 8-16 List of your certificates on your partner's system

4. In the Create New Certificate panel (Figure 8-17 on page 222), complete the following tasks:
   a. For Certificate Type, select **Encryption**.
   b. Provide a description for the certificate.
   c. For Status, select **Enabled**.
   d. Click **Browse** and navigate to the directory where you have stored the public certificate, for example `install_Dir\common\security\keystore\`. Note that this is a folder on the machine where you use the browser, not on the partner's machine.
   e. Certificate Usage indicates the use of multiple certificates. It will declare the certificate that is being uploaded as Primary or Secondary (backup). Primary certificate is the first and default certificate that is used for secure exchange of data. If the primary certificate under consideration is found invalid or not fit to be used, the server picks up the secondary certificate to enable the secure exchange.
   f. Click **Upload**.

Archived

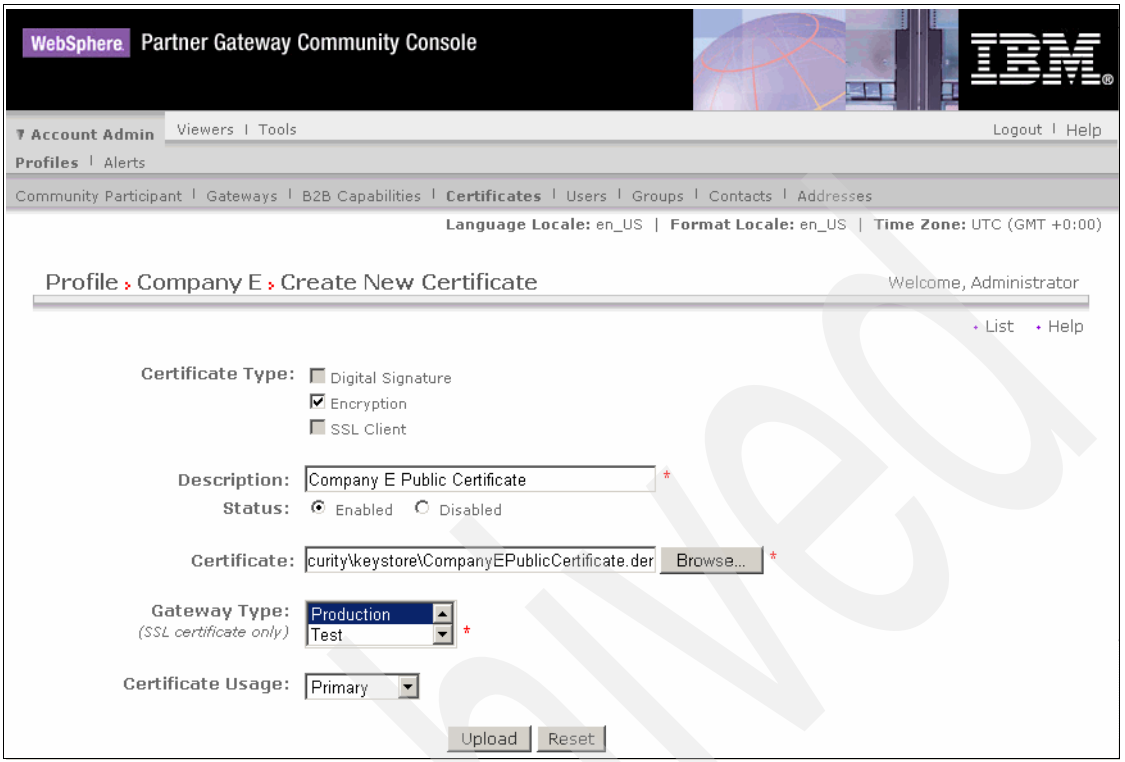


Figure 8-17 Uploading the certificate to the server of our partner

5. The certificate is now analyzed by the server of our partner. Review this analysis as shown in Figure 8-18 on page 223 and Figure 8-19 on page 224, and click **Save**.

Archived

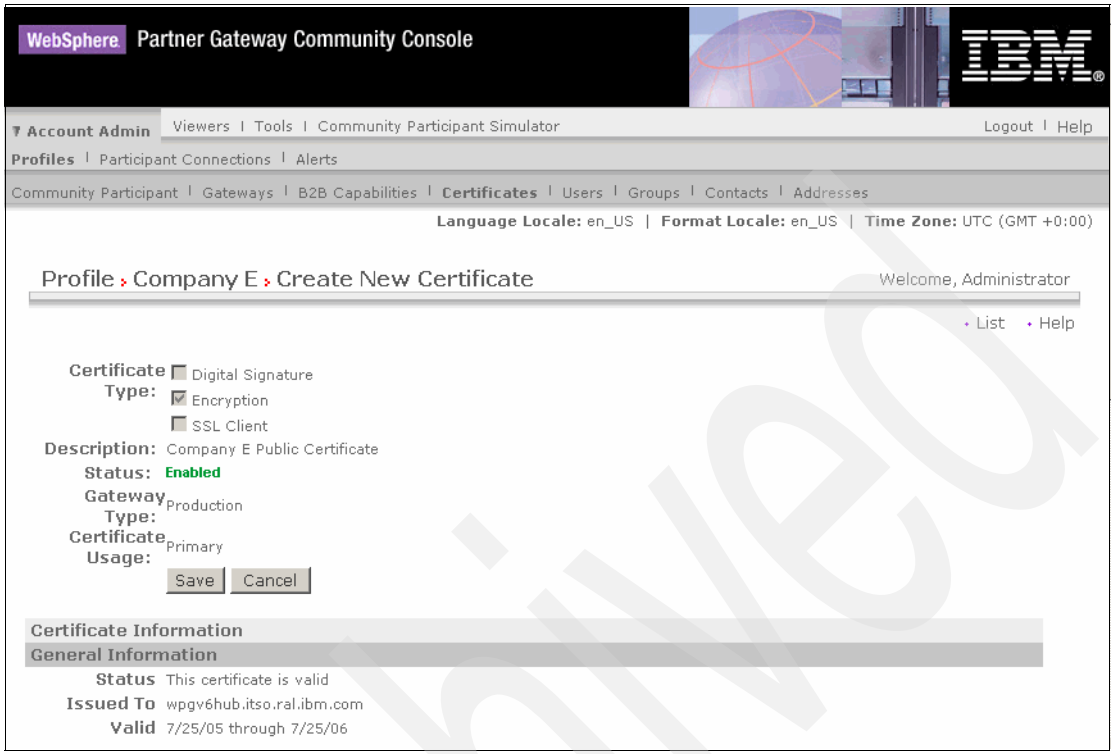


Figure 8-18 Saving the uploaded certificate (1 of 2)

Archived

Detail Information

Version	3
Serial Number	42E5 670D
Valid From	Monday, July 25, 2005 10:26:21 PM GMT+00:00
Valid To	Tuesday, July 25, 2006 10:26:21 PM GMT+00:00
Fingerprint Algorithm	MD5
Fingerprint	7203 3FDF 936E C045 EC15 10ED 7469 0F2C
Public Key	3081 9F30 0D06 092A 8648 86F7 0D01 0101 0500 0381 8D00 3081 8902 8181 008E 588C 628A F15B 730F 4C3A 5A52 1DFF 3A62 82D4 0FAF 8D82 E482 A27D 9092 16B6 6CD3 1725 E7E2 5E3F B0A2 EAE5 A580 89B7 90E6 02EE 5AA1 7FD3 1FBB CF9D 19F9 328D 451F 911F 786E 5C22 6747 9BBA B9C7 E23F 2100 EA04 1315 0EAB EEAF 1FBA 6AC3 CA2C 7916 FBDA E6C3 DC0B 32AE 02D8 837C A601 6B99 25EB FF1A 1A28 5323 1E11 C07F D51A 3102 0301 0001

Subject Information

Common Name (CN)	wpgv6hub.itso.ral.ibm.com
Organization Unit (OU)	
Organization (O)	Company E
Locality (L)	
State (ST)	
Country (C)	US

Figure 8-19 Saving the uploaded certificate (2 of 2)

6. Figure 8-20 on page 225 shows the result of saving the certificate. Here because we had no certificates loaded prior to our example, for the partner, the certificate we loaded will be considered as the primary certificate, as indicated by the highlighted text in the figure.

Archived

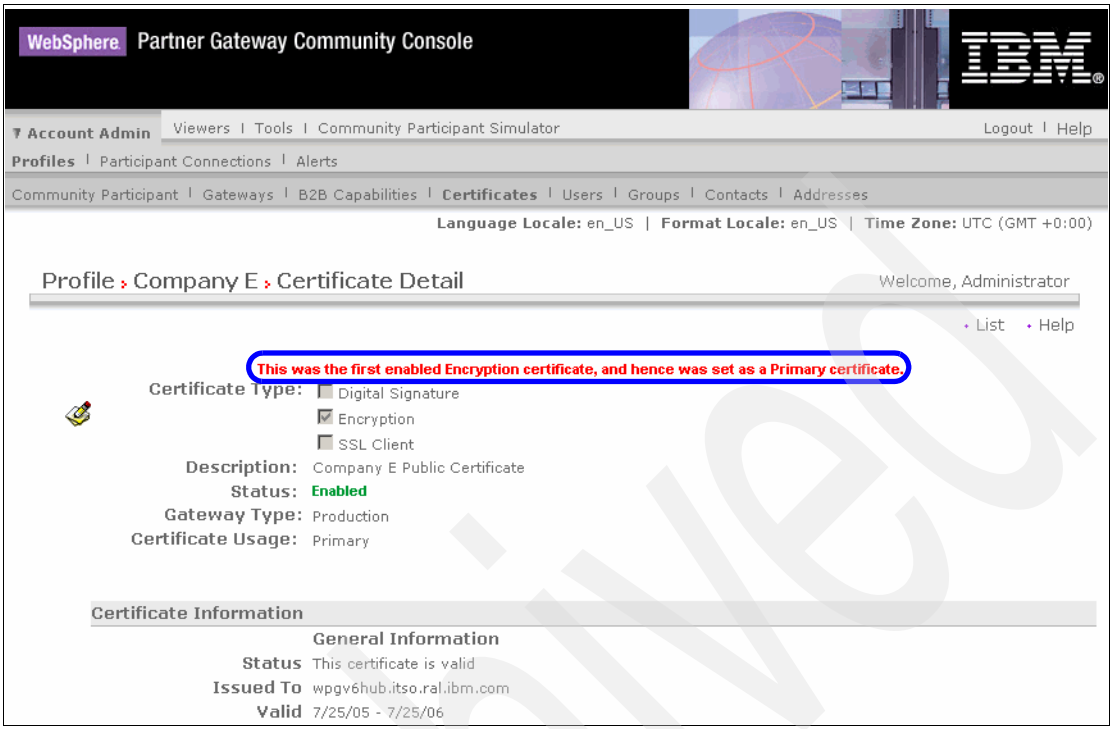


Figure 8-20 Message on saving the certificate

7. On selecting List, to display the list of certificates uploaded for Company E, we see a screen as shown in Figure 8-21 on page 226. The red text indicates a warning message, which is to inform the user that only a primary certificate has been uploaded, it is advised to upload a secondary certificate in order to avoid failure of transaction or transfer of data securely, due to invalidity or non-usability of the primary certificate.

Archived

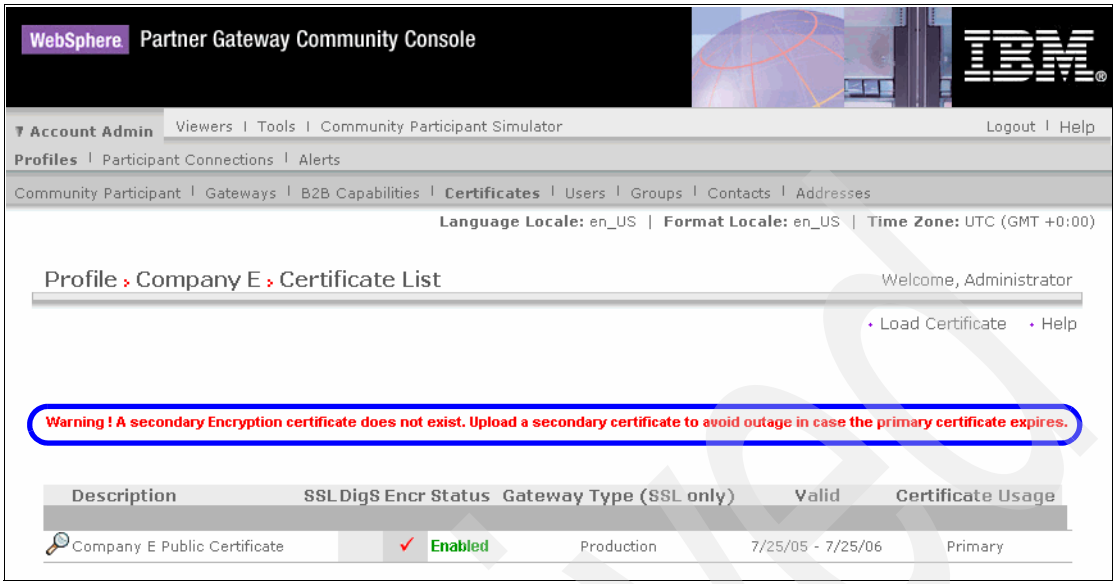


Figure 8-21 List of certificates uploaded

8.2.4 What happens next?

At this moment, Company A can send an encrypted document to his partner Company E. At this stage, we can amend the participant connection between Company A and Company E on the server of Company A and indicate that encryption is required. However, given that we want encryption in both directions for EDI documents, we enable encryption from the perspective of Company A.

8.2.5 Company A generates a private/public key pair

The process that needs to be executed from a Company A perspective is similar to the process for Company E. For completeness, this section explains these steps.

First, the files that we need to use (create, update, or both) are either owned by the user ID bcguser or should be owned by bcguser. Therefore, it is important that you run the key management tool while logged on with this user.

1. Launch the key management tool using the script ikeyman.sh, which can be found in /opt/IBM/bcghub/was/profiles/bcgreceiver/bin
2. When the key management tool is launched, select **Key Database File** → **Open**.

3. In the Open window (Figure 8-22), click Browse and navigate to the directory /opt/IBM/bcghub/common/security/keystore/ and open the receiver.jks file. Click **OK**.

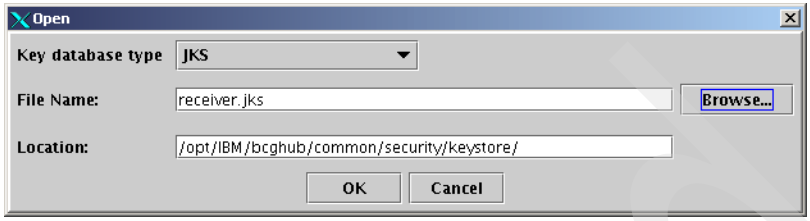


Figure 8-22 Opening the existing key store

4. At the password prompt, enter `WebAS`.
5. Delete the WebSphere dummy server certificate.
6. In the IBM Key Management window (Figure 8-5 on page 213), click **New Self Signed**.
7. In the Create New Self-Signed Certificate window (Figure 8-23 on page 228), complete the details of the new certificate:
   a. Provide a label for the key, for example, `CompanyA_SelfSigned`.
   b. Provide a common name, for example, `m106984f.itso.ral.ibm.com`.
   c. Provide a name for the organization, for example, `Company A`.
   d. Click **OK**.

Archived

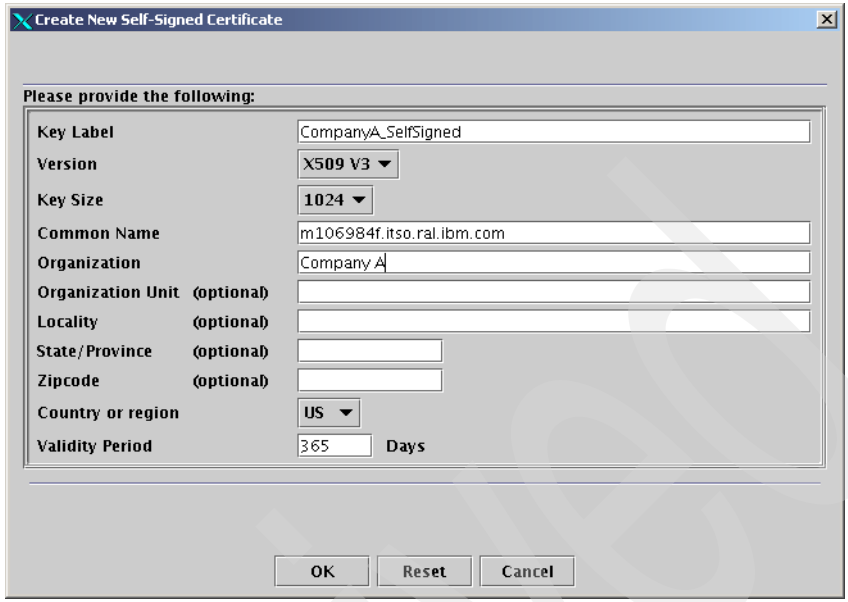


Figure 8-23 Creating a new self-signed certificate

The next step is to extract a public certificate from the new self-signed certificate. The WebSphere Partner Gateway Enterprise server of Company E uses this public certificate to encrypt messages that are sent to the WebSphere Partner Gateway Advanced of Company A. This means that at some point, we need to upload this public certificate to the server of the partner. Because only the owner of the private key,Company A, can decrypt the messages sent by Company E, we know then that confidentiality is assured (Figure 8-24 on page 229).

1. In the IBM Key Management window (Figure 8-24 on page 229), click **Extract Certificate**.

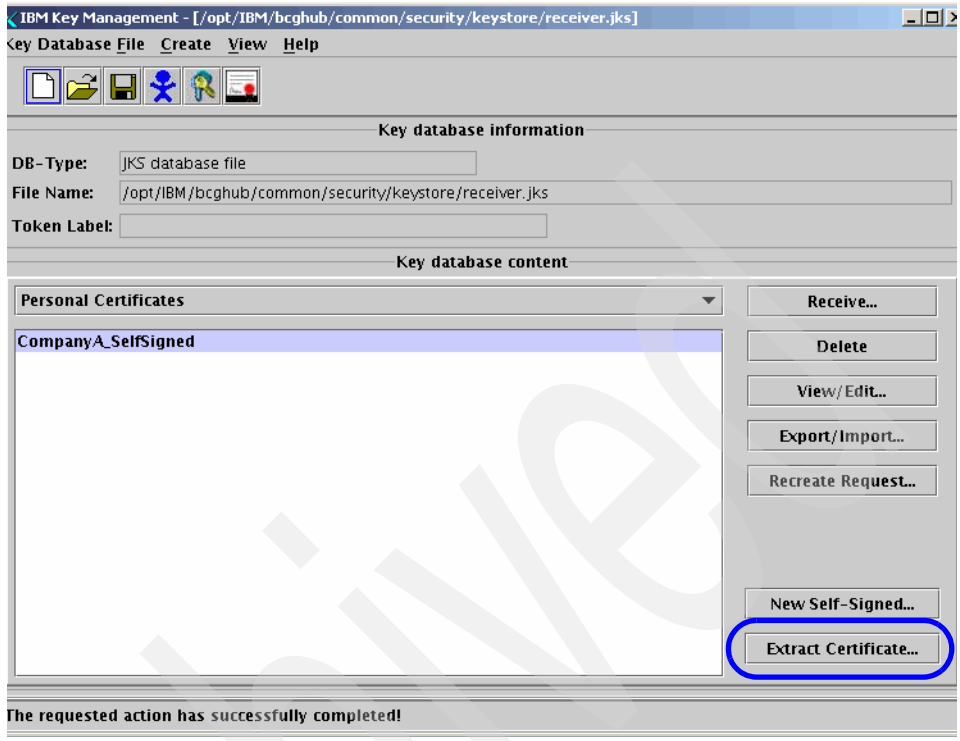


Figure 8-24 Selecting Extract Certificate

2. In the Extract Certificate to a File window (Figure 8-25), complete these tasks.
   a. For Data type, select **Binary DER data**.
   b. For Certificate file name, type `CompanyAPublicCertificate.der`.
   c. Click **OK**.

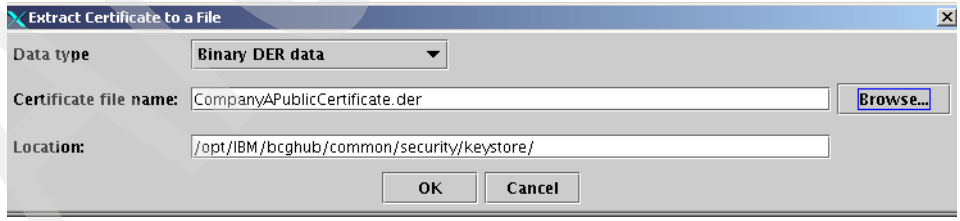


Figure 8-25 Extracting the certificate of key store

This public certificate is extracted in a form that can be used by WebSphere Partner Gateway.

Next, we generate a private certificate from the newly created, self-signed certificate. This is used by Company A's WebSphere Partner Gateway Advanced to decrypt encrypted transactions.

1. From the ikeyman tool, click **Export/Import**.
2. In the Export/Import key window (Figure 8-26), complete these tasks:
   a. For Key file type, select **PKCS12**.
   b. For File name, type `CompanyAPrivateCertificate.p12`.
   c. Click **OK**.

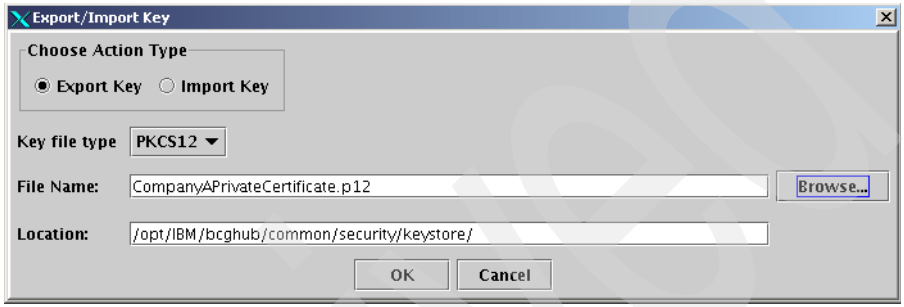


Figure 8-26 Export a key

3. This file contains our private key, so we must protect it well. To avoid unauthorized use, the contents of this file are protected by a password. Any person who wants to use this file in a encryption/decryption solution must know this password before such an encryption/decryption solution accepts the file.

   In the Password Prompt window (Figure 8-27),enter a password to protect the target PKCS12 file and click **OK**. Be sure to make a note of this password, because it is required later.

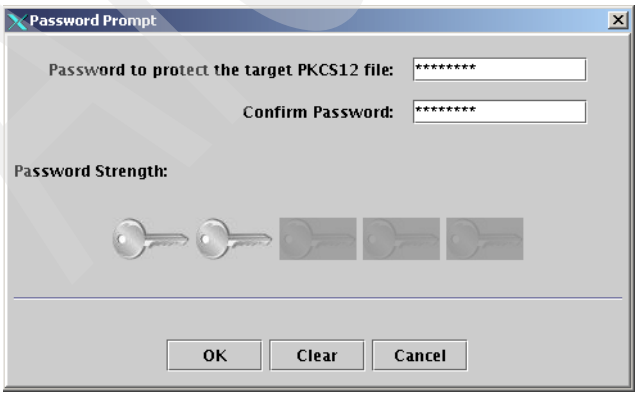


Figure 8-27 Providing a password to protect the private key

8.2.6 Company A uploads a private key to its own server

Now we import the private key file into Company A's WebSphere Partner Gateway Advanced server. Remember, Company A uses this private key when decrypting documents sent by their partners, such as Company E.

When using a browser to upload a file stored on AIX to the WebSphere Partner Gateway server running on AIX, the browser should be running on AIX as well. The browser opens the file and sends it to the Web server and to WebSphere Partner Gateway. It is not WebSphere Partner Gateway that reads the file based on the contents of the entry field in the browser.

Given that, we either need to use a browser running on AIX, or download the certificates from AIX to a PC platform and run the browser on the PC. The following instructions assume that the files were downloaded to a PC and stored in C:\WPGTempAIX.

To upload the PKCS12 file, follow these steps:

1. Log in to WebSphere Partner Gateway as the Hub Admin.
2. Select **Account Admin** → **Profiles** → **Certificates**.
3. Select **Load PKCS12**.
4. In the Create New PKCS12 Certificate window (Figure 8-28 on page 232), do these tasks:
   a. For Certificate Type, select **Encryption**.
   b. Provide a description for the certificate.
   c. For Status, select **Enabled**.
   d. Click **Browse** and navigate to the directory in which the PKCS12 file is stored, for example `C:\WPGTempAIX`. Select the file and click **Open**.
   e. Enter the password used when exporting the PKCS12 file earlier (see Figure 8-27 on page 230)
   f. Click **Upload**.

Archived

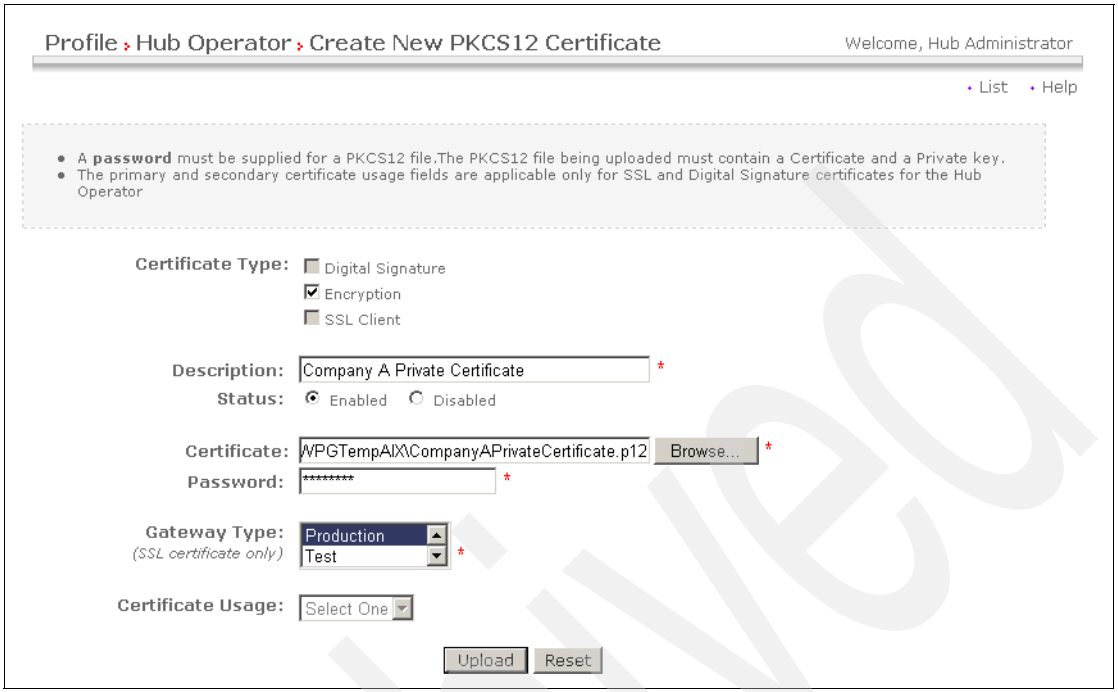


Figure 8-28 Uploading the private key to our own server

5. The server now reviews the certificate and presents us with information about it as shown in Figure 8-29 on page 233 and Figure 8-30 on page 234. Review this to make sure that we upload to the server what should be uploaded. Then click **Save**.

Archived

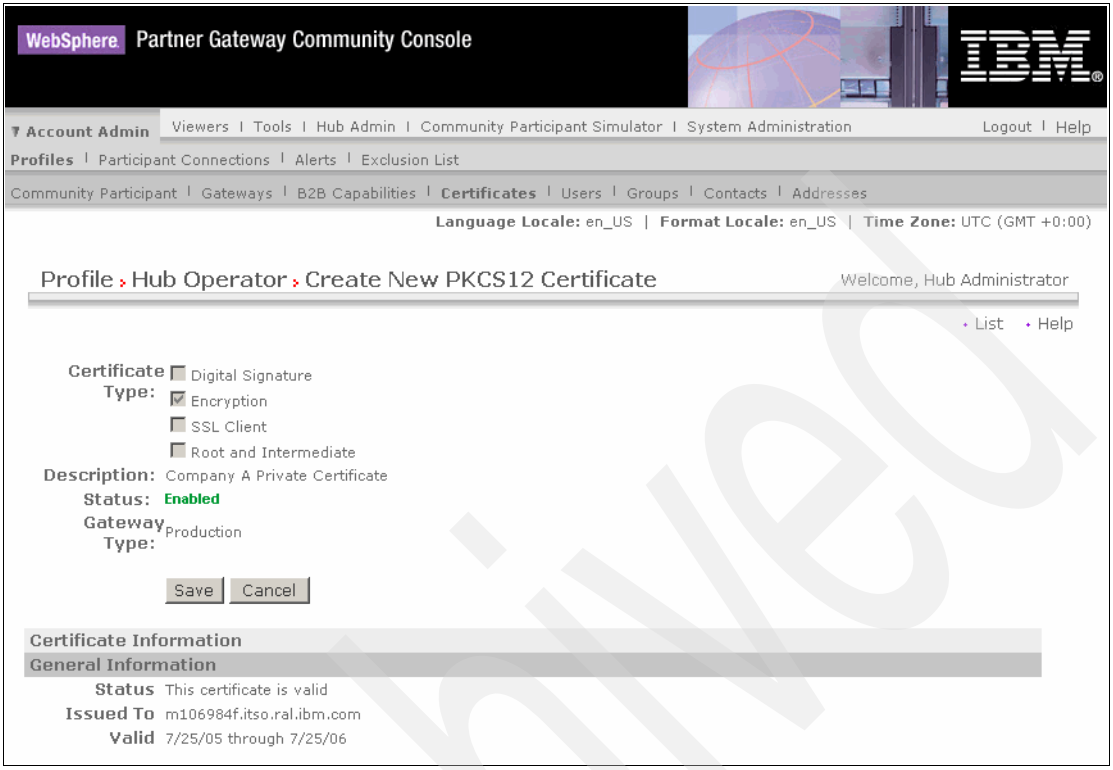


Figure 8-29 Saving the uploaded certificate (1 of 2)

Archived

Detail Information

Version	3
Serial Number	42E5 6DA7
Valid From	Monday, July 25, 2005 10:54:31 PM GMT+00:00
Valid To	Tuesday, July 25, 2006 10:54:31 PM GMT+00:00
Fingerprint Algorithm	MD5
Fingerprint	F202 1EC2 C2C3 9F2C 8AA9 5EC9 A29C 0655
Public Key	3081 9F30 0D06 092A 8648 86F7 0D01 0101 0500 0381 8D00 3081 8902 8181 00C7 1FCE C937 2972 DA30 4968 CA69 E2E2 F03B 10CA 7E3F F4FC 80CE 06CF 85CA 0B58 495A 68DA 96CF BF6A D618 AA7F A93B 8CD6 3384 D9AE 75F4 8077 977D 03DF 52E6 9697 AF65 F538 D687 0A06 28DC 0855 D4C2 ED8B 3505 98BB EA33 9EF2 FF0D D507 08BF 7B4C F30C C7A9 FFA7 73ED A093 A56E FB5F ACE1 C254 A019 3D17 6055 5F58 5998 9643 2C31 F302 0301 0001

Subject Information

Common Name (CN)	m106984f.itso.ral.ibm.com
Organization Unit (OU)	
Organization (O)	Company A
Locality (L)	
State (ST)	
Country (C)	US

Figure 8-30 Saving the uploaded certificate (2 of 2)

6. Click **List** to return to the list of certificates that now contains the new entry. See Figure 8-31.

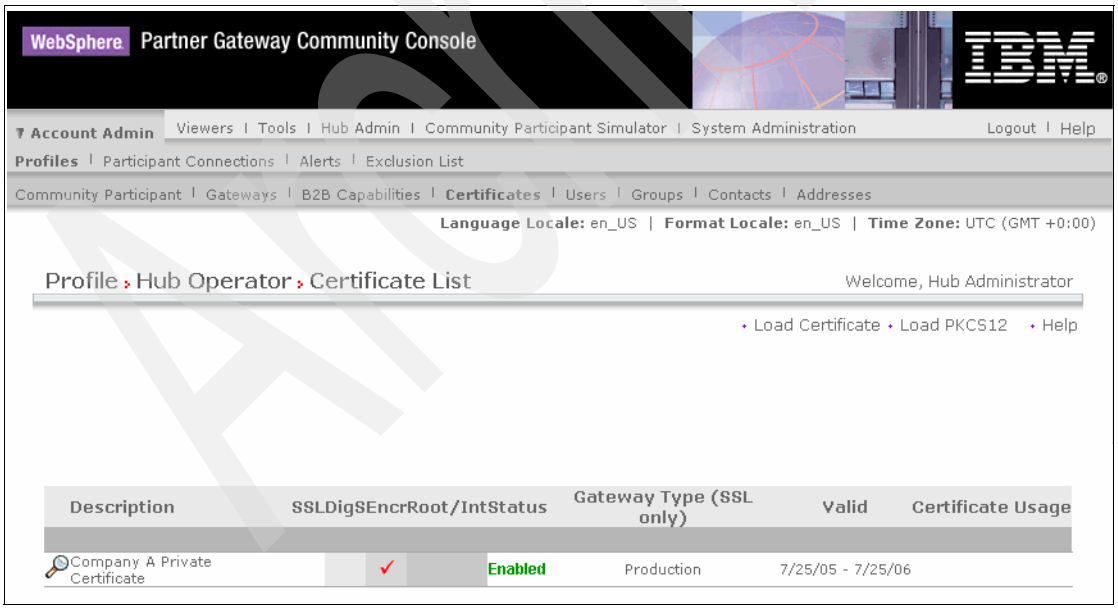


Figure 8-31 Certificate uploaded

Archived

8.2.7 Company A uploads their public key to their partner's server

Upload the public key to the partner's server. This time, the administrator of Company A logs on to the server of Company E. Company E uses this key to encrypt any documents targeted for Company A.

1. Log in as the Community Participant on Company E's partner server.
2. Select **Account Admin** → **Profiles** → **Certificates**.
3. Click **Load Certificate**.
4. In the Create New Certificate window (Figure 8-32 on page 236), complete these tasks:
   a. For Certificate Type, select **Encryption**.
   b. Provide a description for the certificate.
   c. For Status, select **Enabled**.
   d. Click **Browse** and navigate to the directory where we have stored the public certificate, for example `C:\WPGTempAIX`. This is a folder on the machine where we use the browser, not on the partner's machine.
   e. Certificate Usage indicates the use of multiple certificates, it will declare the certificate that is being uploaded as Primary or Secondary (backup). Primary certificate is the first and default certificate that is used for secure exchange of data. If the primary certificate is found invalid or not fit to be used, the server picks up the secondary certificate to enable the secure exchange.
   f. Click **Upload**.

Archived

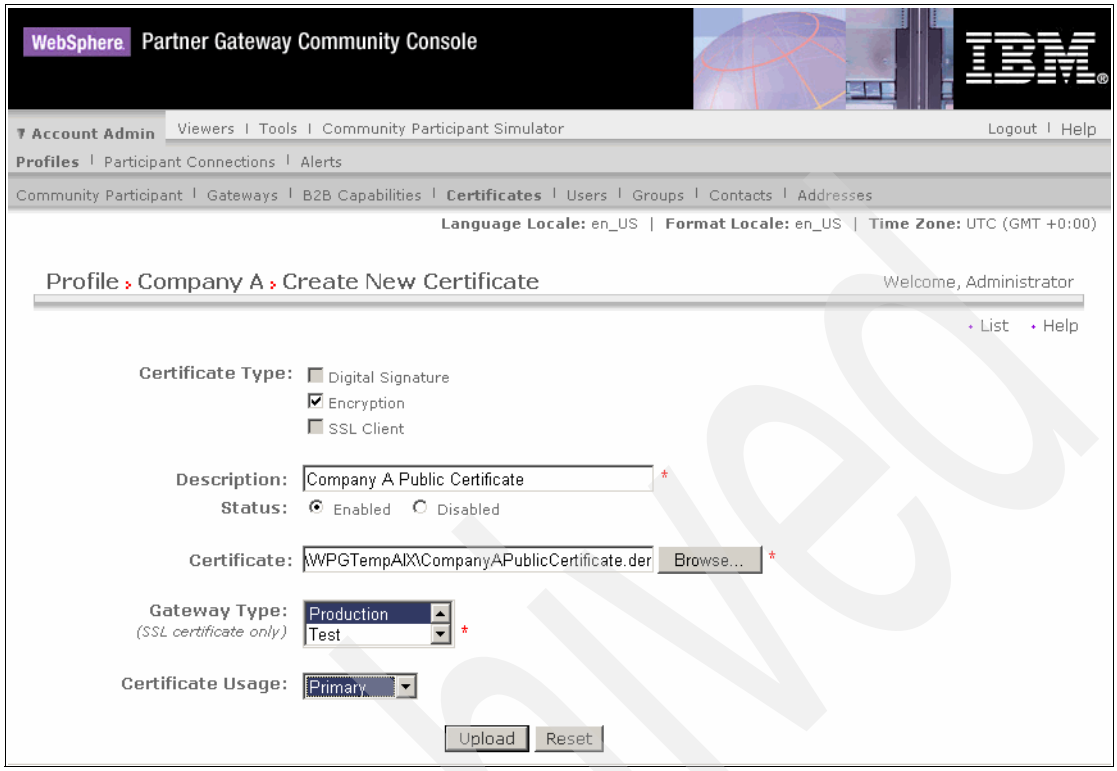


Figure 8-32 Uploading the certificate to the server of our partner

5. The certificate is now analyzed by the server of our partner. Review this analysis, which is shown in Figure 8-33 on page 237 and Figure 8-34 on page 238, and click **Save**.

Archived

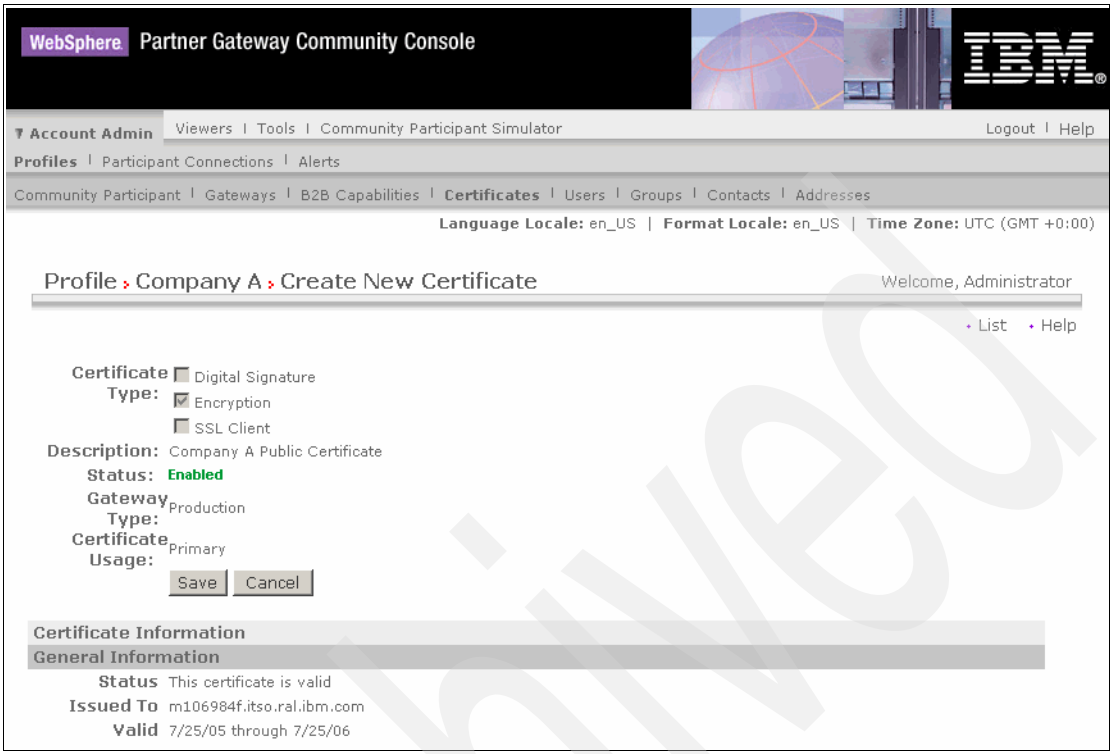


Figure 8-33 Saving the uploaded public certificate

Archived

Detail Information

Version	3
Serial Number	42E5 6DA7
Valid From	Monday, July 25, 2005 10:54:31 PM GMT+00:00
Valid To	Tuesday, July 25, 2006 10:54:31 PM GMT+00:00
Fingerprint Algorithm	MD5
Fingerprint	F202 1EC2 C2C3 9F2C 8AA9 5EC9 A29C 0655
Public Key	3081 9F30 0D06 092A 8648 86F7 0D01 0101 0500 0381 8D00 3081 8902 8181 00C7 1FCE C937 2972 DA30 4968 CA69 E2E2 F03B 10CA 7E3F F4FC 80CE 06CF 85CA 0B58 495A 68DA 96CF BF6A D618 AA7F A93B 8CD6 3384 D9AE 75F4 8077 977D 03DF 52E6 9697 AF65 F538 D687 0A06 28DC 0855 D4C2 ED8B 3505 98BB EA33 9EF2 FF0D D507 08BF 7B4C F30C C7A9 FFA7 73ED A093 A56E FB5F ACE1 C254 A019 3D17 6055 5F58 5998 9643 2C31 F302 0301 0001

Subject Information

Common Name (CN)	m106984f.itso.ral.ibm.com
Organization Unit (OU)	
Organization (O)	Company A
Locality (L)	
State (ST)	
Country (C)	US

Figure 8-34 Saving the uploaded public certificate

6. Figure 8-35 on page 239 shows the result of saving the certificate. Here because we had no certificates loaded prior to our example, for the partner, the certificate we loaded will be considered as the primary certificate, as indicated by the highlighted text in the figure.

Archived

Profile › Company A › Certificate Detail

Welcome, Administrator

• List • Help

This was the first enabled Encryption certificate, and hence was set as a Primary certificate.

Certificate Type: ☐ Digital Signature
☑ Encryption
☐ SSL Client

Description: Company A Public Certificate

Status: Enabled

Gateway Type: Production

Certificate Usage: Primary

Certificate Information

General Information

Status: This certificate is valid

Issued To: m106984f.itso.ral.ibm.com

Valid: 7/25/05 - 7/25/06

Detail Information

Version: 3

Serial Number: 42E5 6DA7

Valid From: Monday, July 25, 2005 10:54:31 PM GMT+00:00

Valid To: Tuesday, July 25, 2006 10:54:31 PM GMT+00:00

Fingerprint Algorithm: MD5

Fingerprint: F202 1EC2 C2C3 9F2C 8AA9 5EC9 A29C 0655

Public Key: 3081 9F30 0D06 092A 8648 86F7 0D01 0101 0500 0381 8D00 3081 8902 8181 00C7 1FCE C937
2972 DA30 4968 CA69 E2E2 F03B 10CA 7E3F F4FC 80CE 06CF 85CA 0B58 495A 68DA 96CF BF6A
D618 AA7F A93B 8CD6 3384 D9AE 75F4 8077 977D 03DF 52E6 9697 AF65 F538 D687 0A06 28DC
0855 D4C2 ED8B 3505 98BB EA33 9EF2 FF0D D507 08BF 7B4C F30C C7A9 FFA7 73ED A093 A56E
FB5F ACE1 C254 A019 3D17 6055 5F58 5998 9643 2C31 F302 0301 0001

Figure 8-35 Message on saving the certificate

7. On selecting **List**, to display the list of certificates uploaded for Company E, we see a screen as shown in Figure 8-36 on page 240. The red text indicates a warning message, which is to inform the user that only a primary certificate has been uploaded, it is advised to upload a secondary certificate in order to avoid failure of transaction or transfer of data securely, as a result of invalidity or nonusability of the primary certificate.

Archived

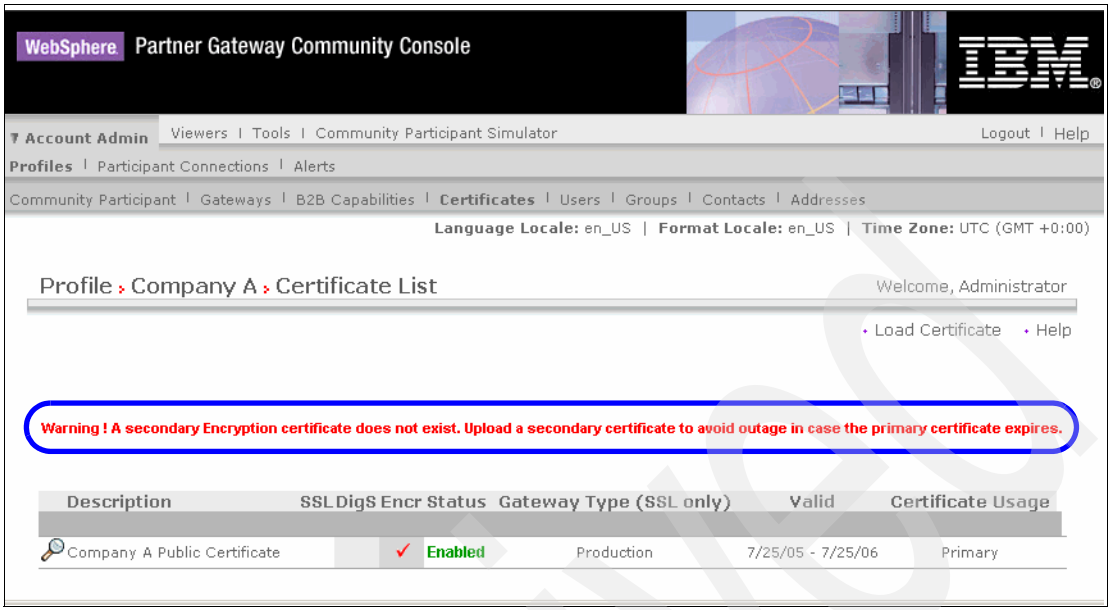


Figure 8-36 List of certificates uploaded

8.2.8 Updating the participant connections

To configure the use of encryption for an AS2 exchange between two partners, we need to change the settings of the participant connection. Earlier 7.6, “Connecting Company E to Company A” on page 180, explained how to inspect and change connection attributes for AS2 when sending documents from Company E to Company A.

Figure 8-37 on page 241 shows the change that is required. To enable encryption, select **Yes** next to the label AS Encrypted and click **Save**. From now on, WebSphere Partner Gateway generates a random session key, encrypts the session key using the public key of the receiver, and uses the session key in a symmetric encryption algorithm to encrypt the actual data.

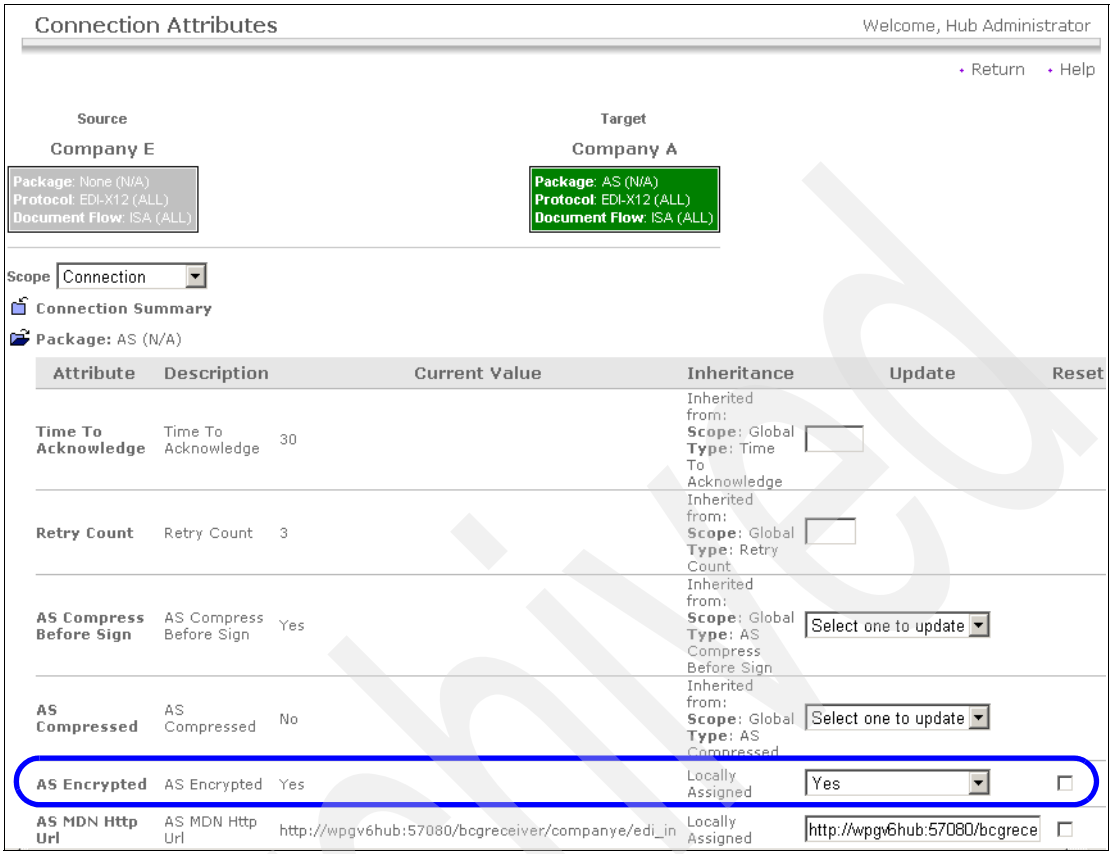


Figure 8-37 Connection attributes for AS2 from Company E to Company A

We must make the same changes to the participant connection from Company A to Company E. Refer to 7.7, “Configuration tasks for the Company A hubadmin” on page 188, which explains the setup on the server of Company A and how to access the AS2 connection attributes. Change the setting AS Encrypted to **Yes** for that connection as well.

8.2.9 Validating that encryption is enabled

We can validate the encryption in the same way as before. We drop an EDI file into the directory that is polled by the WebSphere Partner Gateway Enterprise FileSystem Target. It arrives in the corresponding FileSystemGateway on the WebSphere Partner Gateway server of Company A. The fact that the file arrives is an indication that the changes to the participant connection have not broken the setup from the end of Chapter 7, “Creating a basic B2B exchange” on page 137.

Archived

To verify that encryption has occurred, again we use the AS1/AS2 Viewer.

1. From the menu bar, select **Viewers** → **AS1/AS2 Viewer**.
2. Provide search arguments and click **Search**.
3. When we see the list of AS1/AS2 sessions matching the search criteria, click the icon to view the details of the selected AS2 exchange (Figure 8-38).

   Assuming that the same EDI file is used, we see that the file size quoted in Figure 8-38 is higher than the file size quoted in Figure 7-52 on page 200. This is already an indication that something different has happened.

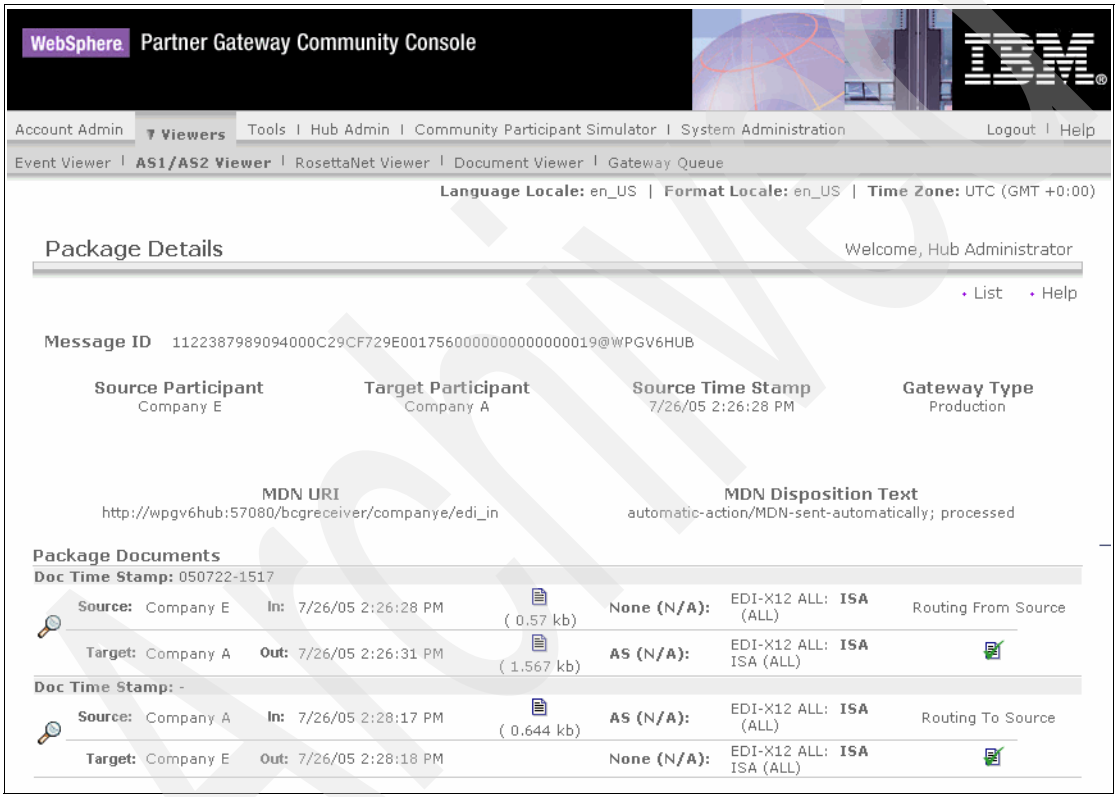


Figure 8-38 Details of an AS2 communication with encryption

To inspect the raw document, click the icon for the AS packaged document sent by Company E. This raw document is shown in Figure 8-39 on page 243.

Clearly, the EDI document is not readable anymore. The content type in Figure 8-39 on page 243 is set to application/pkcs7-mime. Compare this with Figure 7-53 on page 201, where the EDI document is shown in clear text and

the content type is set to application/edi-x12. This shows that the document is encrypted.

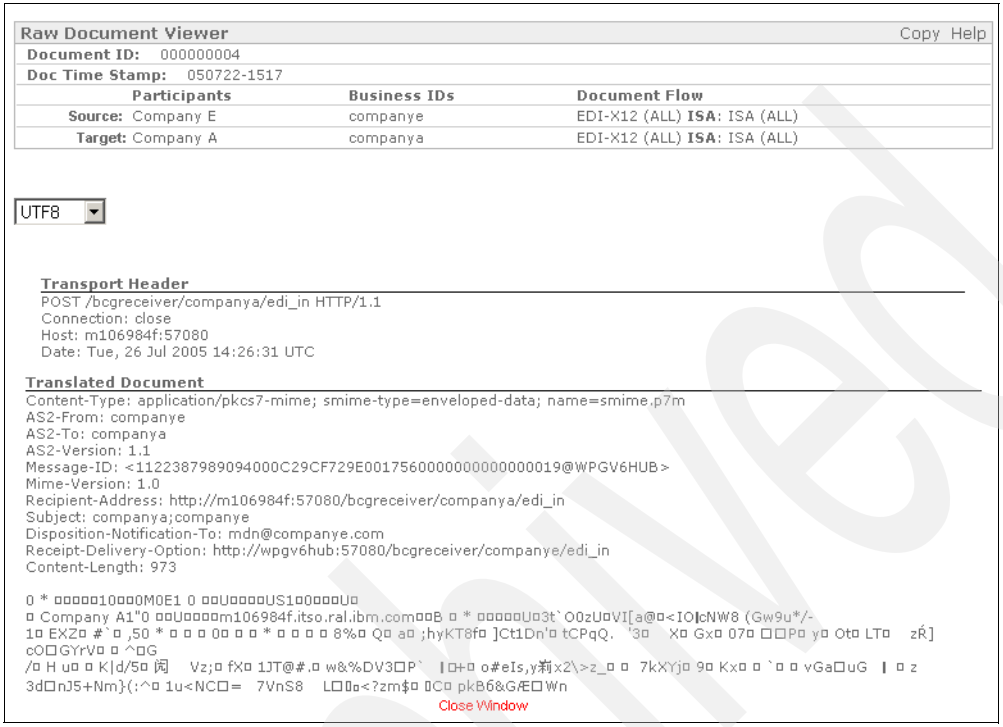


Figure 8-39 Raw document viewer for encrypted document

8.3 What is needed to digitally sign and verify the signature?

Encryption results in a confidential exchange of information. Only the owner of the private key can read the message.

However, as the recipient of that document, there are still two open issues:

- Is this document really sent by our trading partner? Anyone that has obtained the public key of the receiving company could have created this document.
- Is the received document exactly the same as the sent document? It could have been intercepted and altered somewhere between sender and receiver.

Archived

Adding a digital signature to the document provides nonrepudiation and message integrity. In general, digital signatures can be used independently of the encryption. But in many B2B environments, both technologies are combined.

First a message digest of the document is created using a one-way hashing algorithm such as SHA1. Then the message digest is encrypted using the sender's private key. This is a *signature*. The signature is sent with the document, as shown in Figure 8-40.

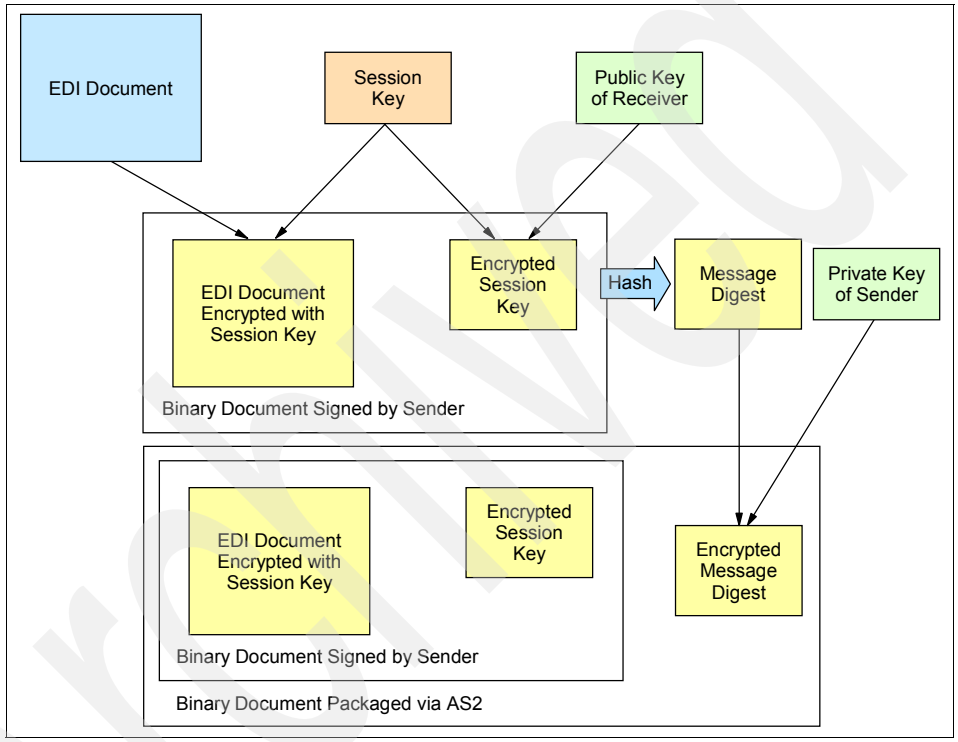


Figure 8-40 Signing a document by calculating a message digest

When the trading partner receives the document, they decrypt the message digest with the sender's public key and compare the decrypted message digest with one that is calculated from the actual document. If there is a match, the receiver knows that this is the document the sender intended to send and that the sender was the one that sent it. This process is shown schematically in Figure 8-41 on page 245.

Archived

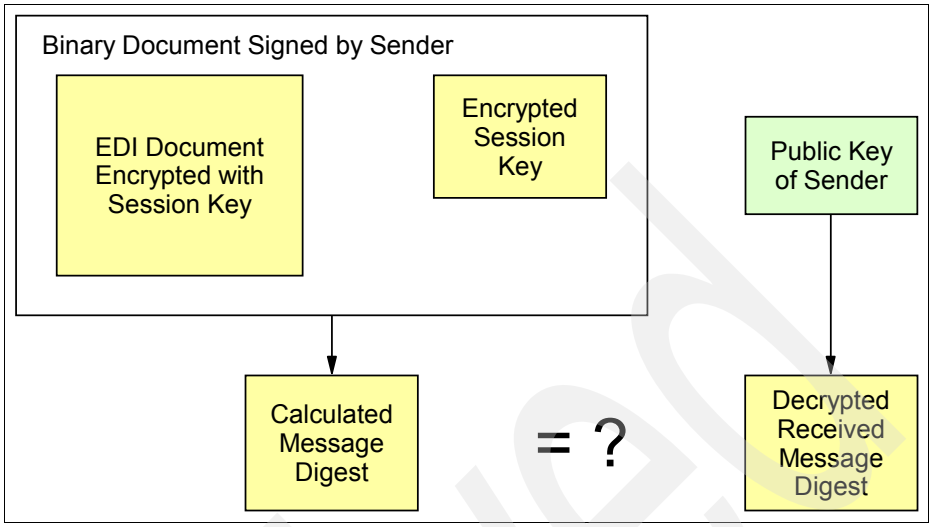


Figure 8-41 Verifying the digital signature

The authenticity of the received document relies on the authenticity of the public key, meaning, is the public key really owned by the organization that pretends to own it? That guarantee is provided by the signer of the public key, which is usually a root certificate authority. To be complete, we should add another step to Figure 8-41. The the sender's public key is validated by using the public key of the signing authority. When using the services of a certificate authority, we need the public key of that authority and to make that public key available to WebSphere Partner Gateway. For self-signed certificates, the public key of the sender is used in both roles:

- Validate itself.
- Calculate the message digest.

The fact that the public key for self-signed certificates is used twice has its impact on the configuration of WebSphere Partner Gateway.

When sending an EDI document over AS2 from Company E to Company A:

- Company E needs the public certificate of Company A to encrypt.
- Company E needs its own private key to sign it.
- Company A uses Company E's public certificate to validate Company E's signature.
- Company A uses its own private certificate to decrypt.

When sending a MDN from Company A to Company E:

- Company A needs its own private key to sign the MDN.
- Company E uses Company A's public certificate to validate the signature.

8.4 Enabling digital signatures

To enable digital signatures, again we need a public/private key pair. We can use the same key pair that was used for encryption. It is the sufficient to indicate in WebSphere Partner Gateway that the keys have an additional role.

8.4.1 Changes to be performed on the server of Company A

We must perform the following four tasks:

1. The role of the private key of Company A, that was uploaded earlier for decryption, cannot be updated to allow its use for digital signing. Hence we upload another private key for Company A, with its role selected as Digital Signature. The description for the new private key is retained the same as that of the key for decryption.

   Note: In WebSphere BI Connect 4.2.2, the same certificate or PKCS12 could be uploaded with the roles of Encryption and Digital Signature, but WebSphere Partner Gateway has a feature to support multiple certificates which is enabled by the Certificate Usage attribute (as seen in Figure 8-32), this enforces a limitation on the choice of the roles of a certificate being uploaded, there by making the roles Encryption and Digital Signature mutually exclusive , when a certificate or a PKCS12 is being uploaded.

2. The role of the public key of Company E that is stored in the profile of Company E, which was uploaded earlier for encryption, cannot be updated to allow its use for digital signature verification. Therefore, we upload another public key of Company E under the profile of Company E, with its role selected as Digital Signature. The description for the new public key is retained the same as that of the key for encryption.
3. Upload the public key of the signer of the public key of Company E to the profile of hubadmin. Because the public key of Company E is self-signed, we need to upload the public key of Company E in the profile of hubadmin. Thus, the public key for Company E is stored on its own profile on Company A and in the system profile of hubadmin.
4. Update the participant connection to activate signatures for AS2.

Follow the steps:

1. While logged on as hubadmin of Company A, select **Account Admin** → **Profiles** → **Certificates**.
2. The console should show the private key of Company A that was uploaded before and that has one single role: encryption. Click **Load PKCS12**.

3. Upload Company A's private key, select the role **Digital Signature** and click **Upload**.
4. Inspect the certificate details screen and click **Save**.
5. Click **List** to list the certificates for hubadmin on Company A as shown in Figure 8-42

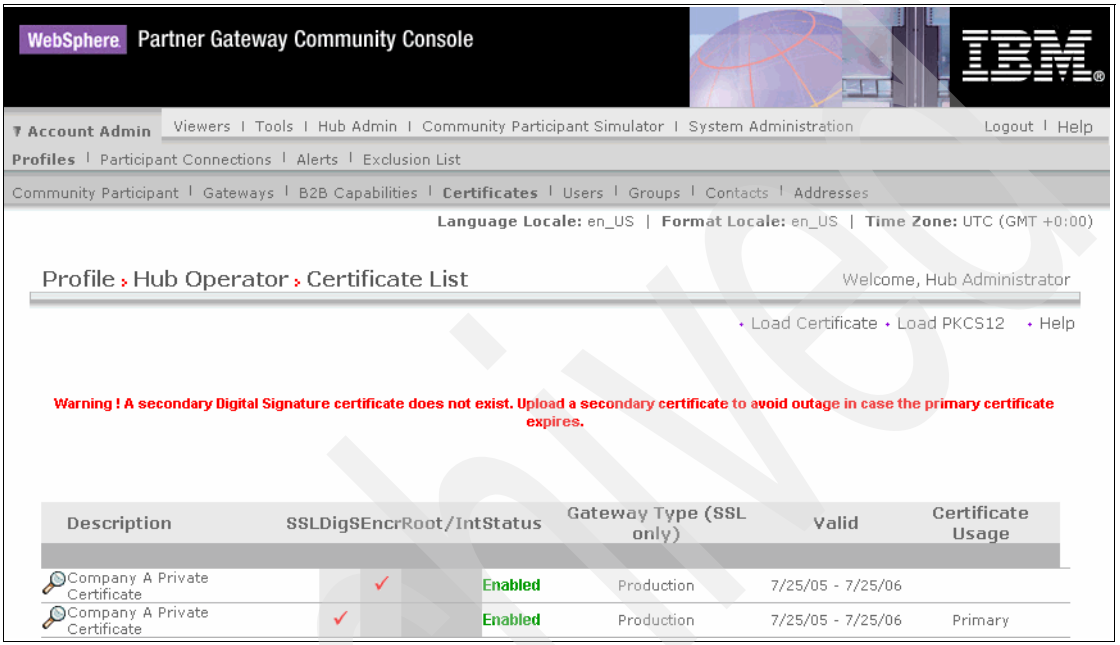


Figure 8-42 List of certificates for hubadmin on Company A

6. While logged on as hubadmin, select **Account Admin** → **Profiles** → **Community Participant**.
7. Click **Search** to display the list of profiles. Select the profile of Company E.
8. When the details of Company E's profile are shown, click **Certificates** in the menu bar. Click on **Load Certificate**.
9. Now upload the public certificate of Company E, which will be use to verify the signature of signed documents sent by Company E. While uploading the certificate select the role of the certificate as Digital Signature.

> **Note:** Uploading a new certificate to the profile of Company E on the server of Company A can also be performed by the admin user of Company E logging on the server of Company A.

Archived

10. Upload the public certificate of Company E as root authority. This task has to be performed by the hubadmin of Company A. Select **Account Admin** → **Profiles** → **Certificates**.

11. When the list of certificates is shown, click **Load Certificate**.

12. In the Create New Certificate window (Figure 8-43), complete these steps:

    a. For Certificate Type, select **Root Certificate**.
    b. Provide a description.
    c. Set Status to **Enabled**.
    d. Click **Browse** to navigate to the actual file that contains the public key.
    e. Click **Upload**.

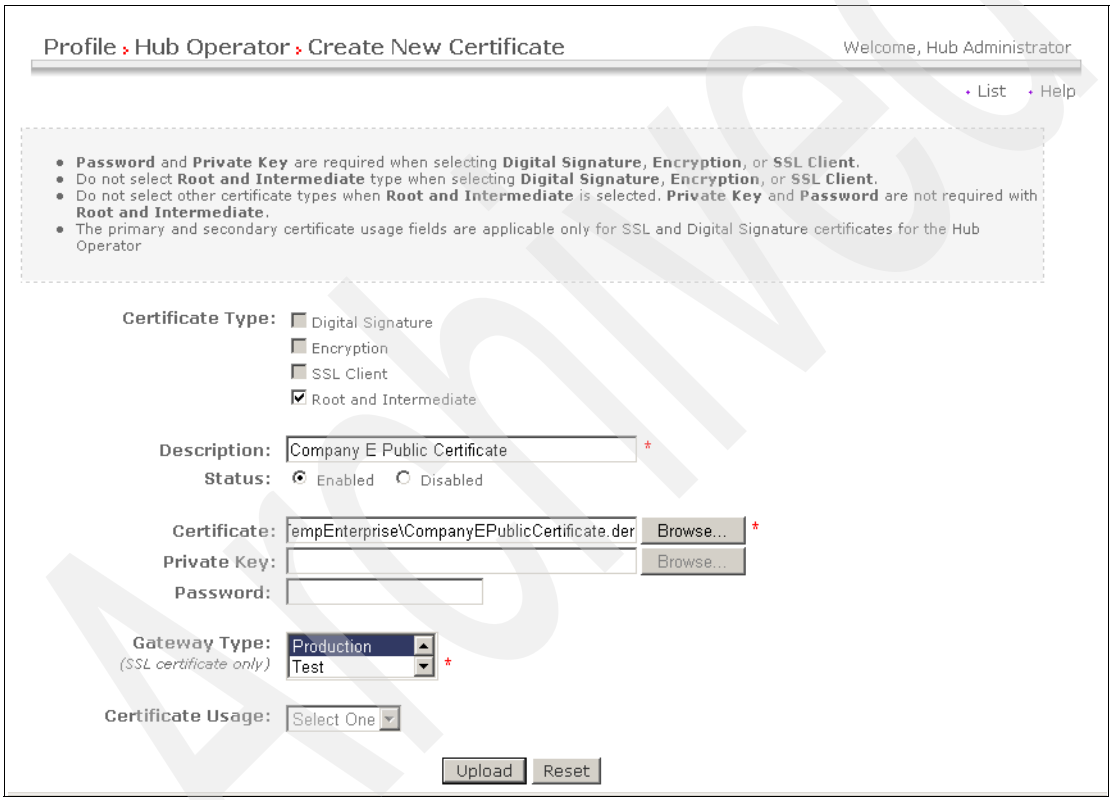


Figure 8-43 Uploading the public key of Company E to Company A

13. Review the analysis of the public key by WebSphere Partner Gateway. Click **Save**. We should see the list of certificates for hubadmin on Company A as shown in Figure 8-44 on page 249.

Archived

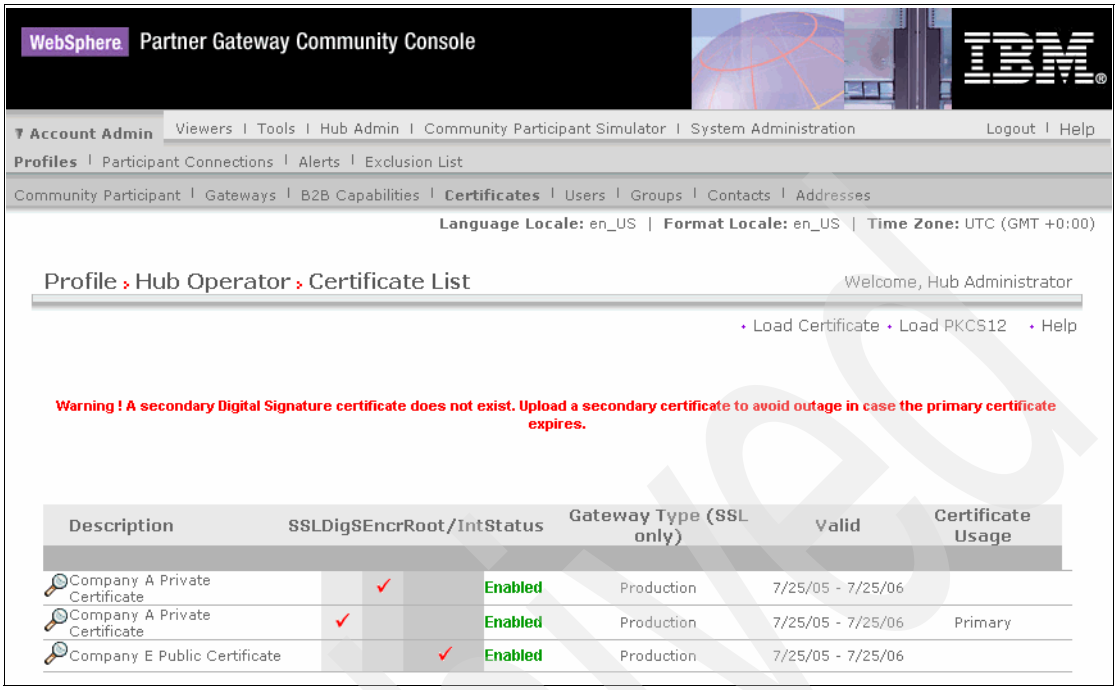


Figure 8-44 List of certificates for hubadmin on Company A

Archived

About public keys: You might think this last task is not in line with the role-based approach that we have used in this redbook so far. The hubadmin needs to have access to the file that contains the public key of their partner to upload it as a Root Certificate.Previously, when enabling encryption, the admin user of Company E could upload the file to the server of Company A and, as such, avoid a handover of their own public key file. This process is a direct result of the use of self-signed certificates.

In real-life situations, business partners usually require the use of certificates signed by a commercial certificate authority. In that situation, the hubadmin of Company A obtains the public key from the certificate authority. Thus the hubadmin user can upload that public key to the server. In real-life situations, Figure 8-44 on page 249 lists the private key of Company A and the public key of the certificate authority.

Anytime there is a certificate exchanged between communicating partners, it is the responsibility of the recipient partner to validate the incoming certificate. An End-Entity public certificate coming from a partner is validated for authenticity by looking up its signer certificate in the trust store, or in other words, among the list of certificates trusted by the recipient company. These trusted certificates are uploaded with the role of Root Certificate, from the console. The trusted certificates usually are the public certificates of the signer, who has signed the End-Entity certificate.

These public certificates are checked for their revocation status by looking up the Certificate Revocation List (CRL) repository.

The check for revocation status of the certificate has changed from WebSphere BI Connect 4.2.2 to WebSphere Partner Gateway 6.0, to the effect that now it is required to upload all the public certificates in the issue chain of the End-Entity certificate, until the Root Certificate, which is usually the certificate of a Certifying Authority (CA).

The future release of WebSphere Partner Gateway will have enhanced functionality features for revocation status check. One of them would be that the revocation check would be made configurable, so that a company can opt to validate its certificates using the published certificate revocation lists. The option to configure would be first made available with the bcg.properties file, before supporting the configuration option using the console.

Table 8-1 on page 251 summarizes what is uploaded in which profile and its use.

Table 8-1 Certificates on server of Company A

	Encryption	**Decryption**	**Digital Signature**	**Root**
hubadmin profile		Private key of Company A	Private key of Company A	Public key of Company E
Company A profile				
Company E profile	Public key of Company E		Public key of Company E	

14. While logged on as hubadmin, select **Account Admin** → **participant connections**.
15. Select the source and target participants and click **Search**. Locate the connection for sending AS2 documents and click **Attributes**.
16. In the window shown in Figure 8-45 on page 252 and Figure 8-46 on page 252, for AS Signed and AS MDN Signed, select **Yes**. Click **Save** to update the attributes.

It is possible to use only signatures on the actual AS document and not on the MDN. However, in practice, business partners require signatures on both the document and the MDN.

Archived

Connection Attributes

Welcome, Hub Administrator

• Return • Help

Source

Company A

Package: AS (N/A)
Protocol: EDI-X12 (ALL)
Document Flow: ISA (ALL)

Target

Company E

Package: None (N/A)
Protocol: EDI-X12 (ALL)
Document Flow: ISA (ALL)

Scope Connection

Connection Summary

Package: AS (N/A)

Attribute	Description	Current Value	Inheritance	Update	Reset
Time To Acknowledge	Time To Acknowledge	30	Inherited from: **Scope**: Global **Type**: Time To Acknowledge		
Retry Count	Retry Count	3	Inherited from: **Scope**: Global **Type**: Retry Count		
AS Compress Before Sign	AS Compress Before Sign	Yes	Inherited from: **Scope**: Global **Type**: AS Compress Before Sign	Select one to update	
AS Compressed	AS Compressed	No	Inherited from: **Scope**: Global **Type**: AS Compressed	Select one to update	
AS Encrypted	AS Encrypted	No	Inherited from: **Scope**: Global **Type**: AS Encrypted	Select one to update	
AS MDN Http Url	AS MDN Http Url	No value provided	No value provided		
AS MDN Email Address	AS MDN Email Address	No value provided	No value provided		

Figure 8-45 Updating AS2 connection attributes (1 of 1)

AS MDN Asynchronous	AS MDN Asynchronous	Yes	Inherited from: **Scope**: Global **Type**: AS MDN Asynchronous	Select one to update	
AS MDN Requested	AS MDN Requested	Yes	Inherited from: **Scope**: Global **Type**: AS MDN Requested	Select one to update	
AS Message Digest Algorithm	AS Message Digest Algorithm	sha1	Inherited from: **Scope**: Global **Type**: AS Message Digest Algorithm	Select one to update	
AS MDN Signed	AS MDN Signed	Yes	Locally Assigned	Yes	☐
AS Signed	AS Signed	Yes	Locally Assigned	Yes	☐
AS Business Id	AS Business Id	No value provided	No value provided	Select one to update	

Save Close

Figure 8-46 Updating AS2 connection attributes (2 of 2)

Archived

8.4.2 Changes to perform on the server of Company E

Again we must perform the same four tasks:

1. The role of the private key of Company E that was uploaded earlier for decryption, cannot be updated to allow its use for digital signing. Therefore, we upload another private key for Company E, with its role selected as Digital Signature. The description for the new private key is retained the same as that of the key for decryption.
2. The role of the public key of Company A that is stored in the profile of Company A, which was uploaded earlier for encryption, cannot be updated to allow its use for digital signature verify. Therefore, we upload another public key for Company A under the profile of Company A, with its role selected as Digital Signature. The description for the new public key is retained the same as that of the key for encryption.
3. Upload the public key of the signer of the public key of Company A to the profile of hubadmin. Because the public key of Company A is self-signed, we need to upload the public key of Company A in the profile of hubadmin. The public key for Company A is stored in his own profile on Company E and in the system profile of hubadmin.
4. Update the participant connections to activate signatures for AS2.

These tasks are exactly the same, so we only show the results. Figure 8-47 on page 254 shows the list of certificates for hubadmin after performing the first and last tasks.

Archived

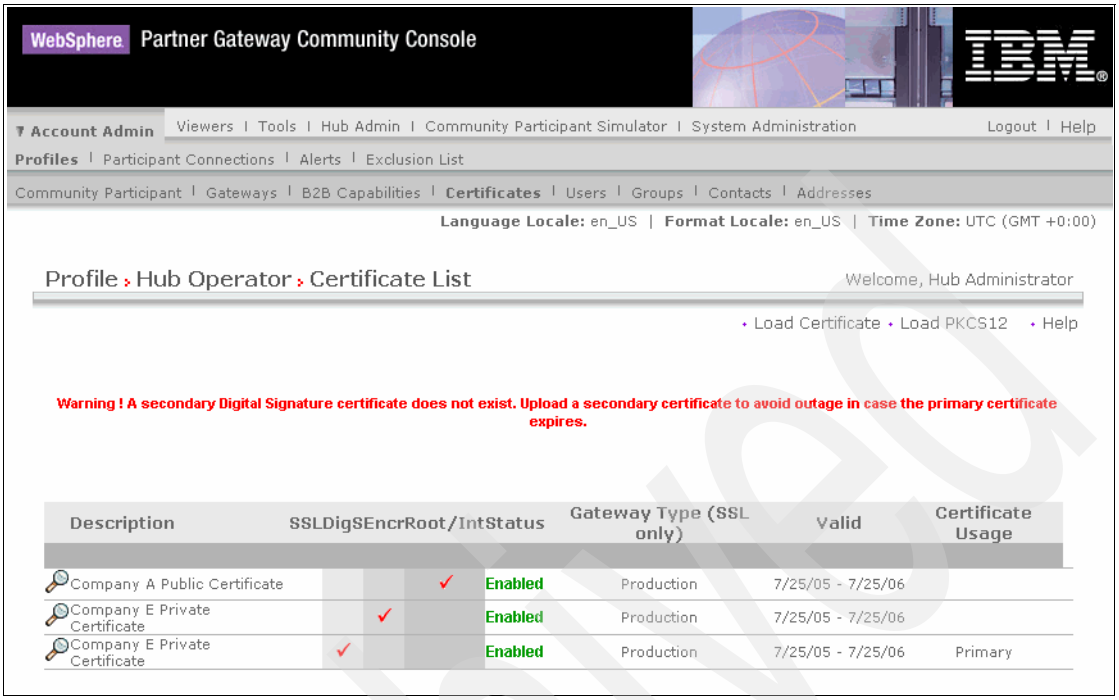


Figure 8-47 Uploading the public key of Company A to Company E

Figure 8-48 on page 255 shows the list of certificates of Company A on the server of Company E after updating the role of the public key, which is the second task. Again, the hubadmin of Company E or the admin user of Company A can perform this task.

Archived

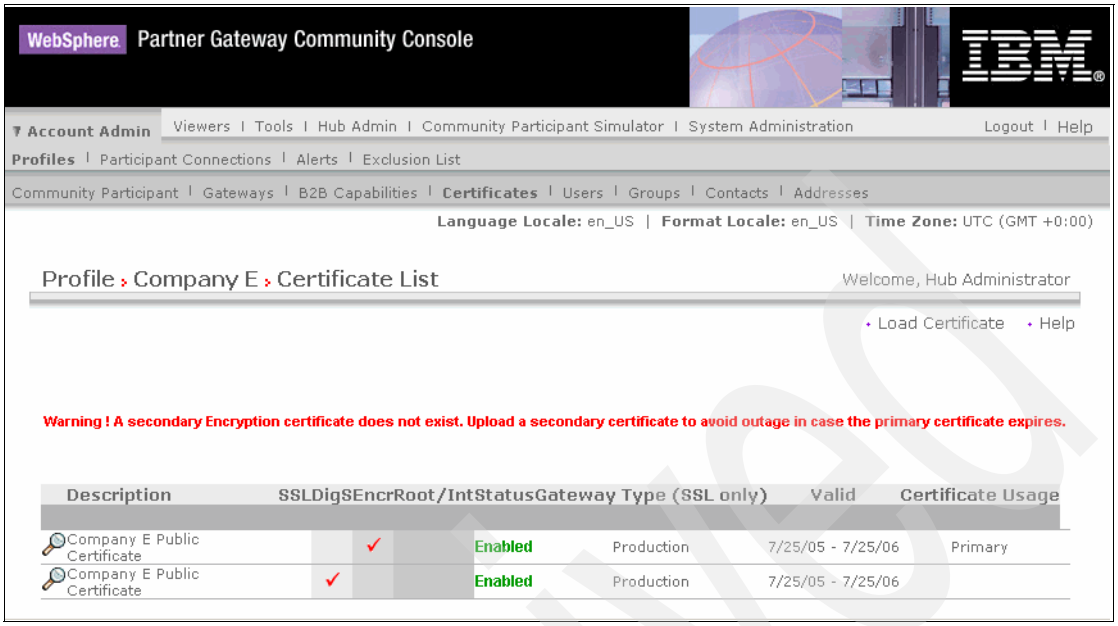


Figure 8-48 Upload public certificate, Company A, on server, Company E

Table 8-2 summarizes where the certificates are stored and the roles they have.

Table 8-2 Certificates on server of Company E

	Encryption	**Decryption**	**Digital signature**	**Root**
hubadmin profile		Private key of Company E	Private key of Company E	Public key of Company A
Company A profile	Public key of Company A		Public key of Company A	
Company E profile				

Figure 8-49 on page 256 and Figure 8-50 on page 257 show the changes that we need to make to the AS2 connection attributes.

Archived

Connection Attributes

Welcome, Hub Administrator

• Return • Help

Source

Company E

Package: None (N/A)
Protocol: EDI-X12 (ALL)
Document Flow: ISA (ALL)

Target

Company A

Package: AS (N/A)
Protocol: EDI-X12 (ALL)
Document Flow: ISA (ALL)

Scope Connection

Connection Summary

Package: AS (N/A)

Attribute	Description	Current Value	Inheritance	Update	Reset
Time To Acknowledge	Time To Acknowledge	30	Inherited from: **Scope**: Global **Type**: Time To Acknowledge		
Retry Count	Retry Count	3	Inherited from: **Scope**: Global **Type**: Retry Count		
AS Compress Before Sign	AS Compress Before Sign	Yes	Inherited from: **Scope**: Global **Type**: AS Compress Before Sign	Select one to update	
AS Compressed	AS Compressed	No	Inherited from: **Scope**: Global **Type**: AS Compressed	Select one to update	

Figure 8-49 Updating the AS2 connection attributes (1 of 2)

Archived

AS Encrypted	AS Encrypted	Yes	Locally Assigned	Yes
AS MDN Http Url	AS MDN Http Url	http://wpgv6hub:57080/bcgreceiver/companye/edi_in	Locally Assigned	http://wpgv6hub:57080/bcgrece
AS MDN Email Address	AS MDN Email Address	mdn@companye.com	Locally Assigned	mdn@companye.com
AS MDN Asynchronous	AS MDN Asynchronous	Yes	Inherited from: **Scope**: Global **Type**: AS MDN Asynchronous	Select one to update
AS MDN Requested	AS MDN Requested	Yes	Inherited from: **Scope**: Global **Type**: AS MDN Requested	Select one to update
AS Message Digest Algorithm	AS Message Digest Algorithm	sha1	Inherited from: **Scope**: Global **Type**: AS Message Digest Algorithm	Select one to update
AS MDN Signed	AS MDN Signed	No	Locally Assigned	Yes
AS Signed	AS Signed	No	Locally Assigned	Yes
AS Business Id	AS Business Id	No value provided	No value provided	Select one to update

Save Close

Figure 8-50 Updating the AS2 connection attributes (2 of 2)

8.4.3 Validating that digital signatures are enabled

After we make the changes on both servers, we perform the validation in the same way.

1. Drop an EDI file in the target directory and verify that it still arrives in the gateway directory on the partner machine.
2. To verify that signatures have been used, we can again use the AS1/AS2 Viewer.
3. Locate again the AS2 transaction and open the details. As shown in Figure 8-51 on page 258, the file sizes of the AS2 document and the MDN are again bigger than when using encryption only.

Archived

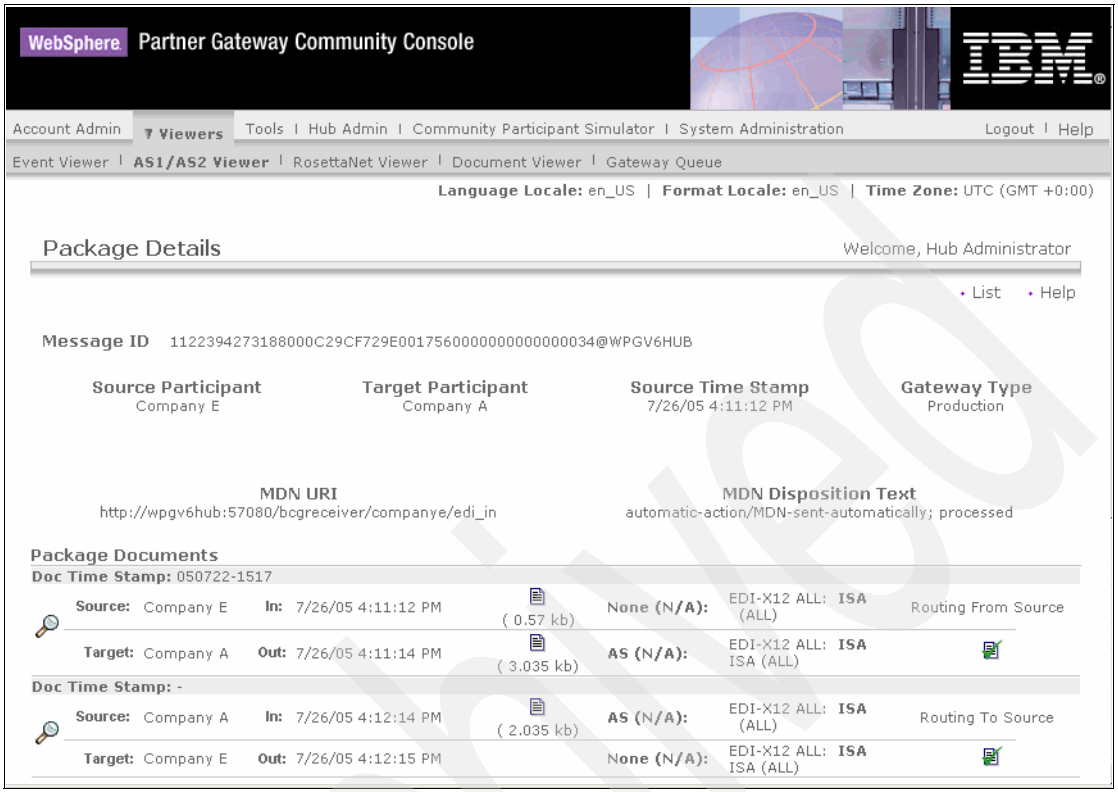


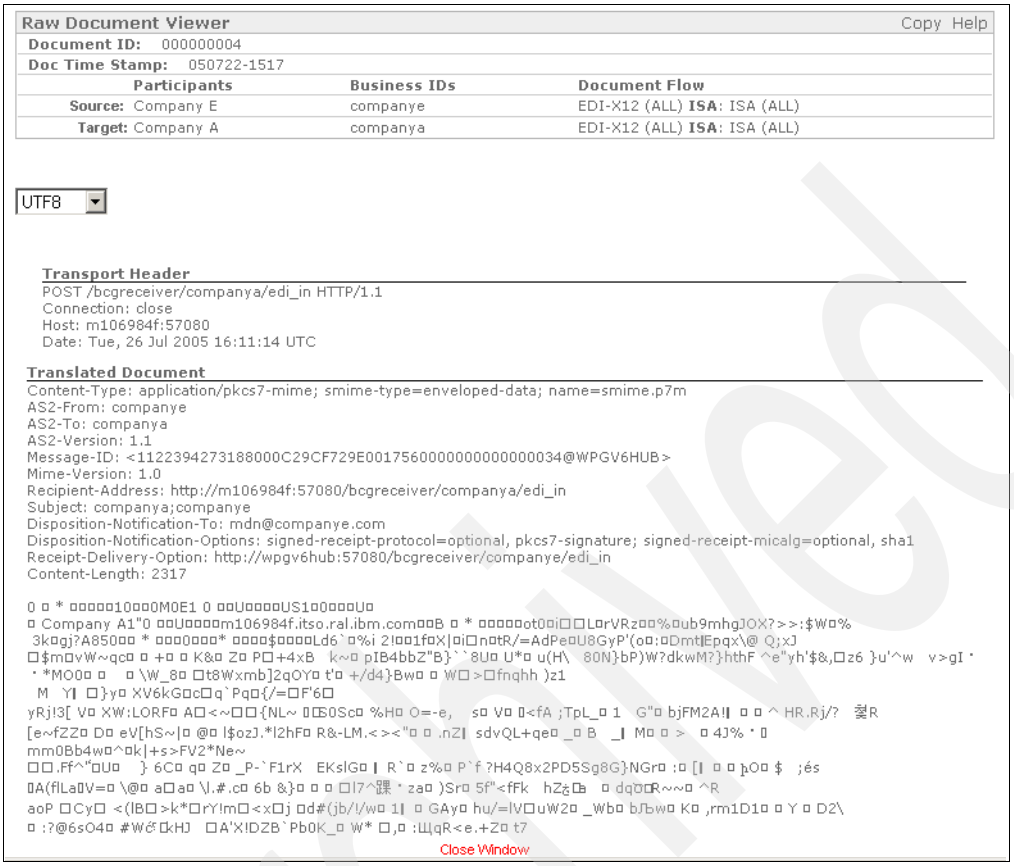
Figure 8-51 Details of an AS2 exchange with encryption and signatures

4. Select the AS2 document to open the Raw Document Viewer, shown in Figure 8-52 on page 259. Again compare the content type in Figure 8-52 on page 259 with the content type in Figure 8-39 on page 243.The content types are the same because the two document manipulation techniques are executed on top of each other. These techniques are executed in the following order:

   a. Sign
   b. Encrypt

   Thus, the encryption hides the signature.

Archived



Raw Document Viewer — Copy Help

Document ID: 000000004

Doc Time Stamp: 050722-1517

	Participants	Business IDs	Document Flow
Source:	Company E	companye	EDI-X12 (ALL) **ISA**: ISA (ALL)
Target:	Company A	companya	EDI-X12 (ALL) **ISA**: ISA (ALL)

UTF8

Transport Header

```
POST /bcgreceiver/companya/edi_in HTTP/1.1
Connection: close
Host: m106984f:57080
Date: Tue, 26 Jul 2005 16:11:14 UTC
```

Translated Document

```
Content-Type: application/pkcs7-mime; smime-type=enveloped-data; name=smime.p7m
AS2-From: companye
AS2-To: companya
AS2-Version: 1.1
Message-ID: <1122394273188000C29CF729E0017560000000000000034@WPGV6HUB>
Mime-Version: 1.0
Recipient-Address: http://m106984f:57080/bcgreceiver/companya/edi_in
Subject: companya;companye
Disposition-Notification-To: mdn@companye.com
Disposition-Notification-Options: signed-receipt-protocol=optional, pkcs7-signature; signed-receipt-micalg=optional, sha1
Receipt-Delivery-Option: http://wpgv6hub:57080/bcgreceiver/companye/edi_in
Content-Length: 2317
```

Close Window

Figure 8-52 Encrypted and signed AS2 document

5. Select the MDN document in Figure 8-51 on page 258 to view the raw document. Figure 8-53 on page 260 and Figure 8-54 on page 260 clearly shows a multipart message, where one part is clear text. Another part has as content type, the value message/disposition-notification. The last part has as content type, the value application/pkcs7-signature.

Archived

Raw Document Viewer Copy Help

Document ID: -

Doc Time Stamp: -

	Participants	Business IDs	Document Flow
Source:	Company A	companya	EDI-X12 (ALL) **ISA**: ISA (ALL)
Target:	Company E	companye	EDI-X12 (ALL) **ISA**: ISA (ALL)

UTF8

Transport Header

```
Content-Length: 2035
Connection: close
Host: wpgv6hub:57080
AS2-Version: 1.1
Recipient-Address: http://wpgv6hub:57080/bcgreceiver/companye/edi_in
Subject: companye;companya
Date: Tue, 26 Jul 2005 16:06:43 UTC
Content-Type: multipart/signed; micalg=sha1; protocol="application/pkcs7-signature"; boundary="----
=_Part_5_1671613720.1122394002822"
AS2-To: companye
AS2-From: companya
ReferenceId: 1122394334156000C29CF729E00152400000000000000013
Message-ID: <1122394002804000 2554F1C42028286D57433B89DD0C3EF@m106984f.itso.ral.ibm.com>
```

Figure 8-53 Signed MDN (1 of 2)

Initial Document

```
------=_Part_5_1671613720.1122394002822
Content-Type: multipart/report; Report-Type=disposition-notification;
 boundary="----=_Part_4_1696189720.1122394002819"

------=_Part_4_1696189720.1122394002819
Content-Type: text/plain
Content-Transfer-Encoding: 7bit

This MDN response message is for:

Message id: <1122394273188000C29CF729E00175600000000000000034@WPGV6HUB>
From: companye

------=_Part_4_1696189720.1122394002819
Content-Type: message/disposition-notification
Content-Transfer-Encoding: 7bit

Reporting-UA: WBI_Connect
Original-Recipient: rfc822; companya
Final-Recipient: rfc822; companya
Original-Message-ID: <1122394273188000C29CF729E00175600000000000000034@WPGV6HUB>
Disposition: automatic-action/MDN-sent-automatically; processed
Received-Content-MIC: JlNBbq2wW8hK96oCaxAh1ZY0EqI=, sha1

------=_Part_4_1696189720.1122394002819--

------=_Part_5_1671613720.1122394002822
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: binary

0 * □□□□□□1 0 □□+□□□□□0 □ * □□□□□□0□j□□□□□B □ * □□□□0E1 0 □□U□□□□US1□0□□□U□
□ Company A1"0 □□U□□□□m106984f.itso.ral.ibm.com0□ 050725225431Z□ 060725225431Z0E1
0 □□U□□□□US1□0□□□U□
□  Company A1"0 □ □ U□ □ □ □ m106984f.itso.ral.ibm.com00 □  * □ □ □ □ □ 0□ 7)rIh□ ?□ uXIZh□j3□ 斗u
□(U□ □ L□s?=□ `U_XYC,1□ □ □ 0 □  * □ □ □ □ □ I@_sXZ□ p9IM"\□ l□ / □ 7□ M p□ Q□ _l6□ =t□ 3a~□/(HT:1□ □ 0M0E1
0 □□U□□□□US1□0□□□U□
□ Company A1"0 □□U□□□□m106984f.itso.ral.ibm.com□□B □□+□□□□□0□□ * □ □1 □ * □□□0□ * □ □1□□
050726160642Z0□ * □ □1□□□unstructuredName0#□ * □ □1□□□Dk□* pyF □ * □□□□□\Hm"Ph\w!NwD□<8H:o
(< fe4I)bH
2□□{@□Ãth□[□F0□ * □ □1□□□unstructuredName
------=_Part_5_1671613720.1122394002822--
```

Close Window

Figure 8-54 Signed MDN (2 of 2)

9

WebSphere Partner Gateway Express on Windows

After finishing the base setup of a B2B exchange between two business partners, we now extend it to a third partner, who uses WebSphere Partner Gateway Express. This chapter focuses on the installation and initial configuration of the Express edition of WebSphere Partner Gateway.

© Copyright IBM Corp. 2005. All rights reserved.

9.1 Overview of the Express edition

WebSphere Partner Gateway Express is a lightweight, easy to use, cost-effective B2B connectivity tool. It leverages AS2 standards for transmitting documents securely over the Internet. It provides a browser interface to manage the connection with trading partners and to manage the exchange of documents between trading partners. The interface provides the same look and feel as the Advanced and Enterprise editions of WebSphere Partner Gateway.

Similar to the full versions, WebSphere Partner Gateway Express provides tools to analyze, track and investigate all aspects of theB2B exchange. You can review information about queued documents that are pending transmission and acknowledgements, and view historical information about successfully sent or failed documents. You can also view, add, and update public certificates and private keys that are used for encryption and digital signatures.

Sending and receiving documents is performed over either HTTP or HTTPS transport for Secure Sockets Layer (SSL) server-based authentication. The documents can either be sent without any further packaging or with the industry standard packaging AS2.

Sending documents can be performed in two ways. You can use the console interface and manually send a file by uploading it through the browser. Alternatively, WebSphere Partner Gateway Express can monitor a directory and send any files that are dropped there by other applications. Files that are received by the server are also stored in a directory on the file system. That means that any integration between WebSphere Partner Gateway Express and other internal applications is through the file system. The full version of WebSphere Partner Gateway offered has more options, such as Java Message Service (JMS) and Web services.

A wide range of document formats is supported by WebSphere Partner Gateway Express. Business formats such as electronic data interchange (EDI) documents (ANSI X12 and EDIFACT) and RosettaNet service content are supported by WebSphere Partner Gateway. Custom XML or standardized XML formats are also supported. In addition to structured data, WebSphere Partner Gateway can also process binary files.

To make sure that document exchange occurs in optimal and secure circumstances, WebSphere Partner Gateway Express supports standard algorithms for encryption, digital signatures, and compression.

Figure 9-1 on page 263 shows a schematic overview of the outbound flow. WebSphere Partner Gateway Express can monitor a folder on the file system for the arrival of new files that are generated by a back-end application. When the

file arrival is noticed, it is copied within the folder structure managed by WebSphere Partner Gateway Express. The process engine picks it up from there to perform document packaging before it is transmitted to the trading partner via the Internet. Sending files can also be performed using the Console, which allows a direct interaction with the process engine. The Console also interacts with the process engine to perform any real-time monitoring of the document flow between the trading partners.

The inbound flow is quite similar. However, the Console cannot be used to receive files directly. All inbound files are stored in the inbound folder on the file system.

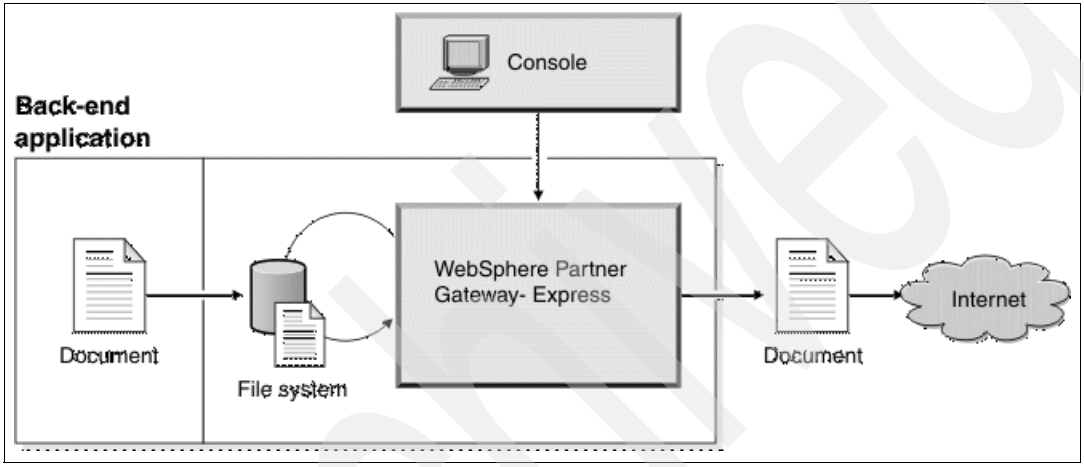


Figure 9-1 Outbound flow for WebSphere Partner Gateway Express

9.2 Software installation and configuration

This section discusses the installation of WebSphere Partner Gateway Express on a single machine (see Figure 9-2 on page 264).

There are a few options to configure or customize during the installation. Chapter 10, “Extending the B2B exchange” on page 277, uses this installation of WebSphere Partner Gateway Express on host name wpgv6x to interact with Company E.

Archived

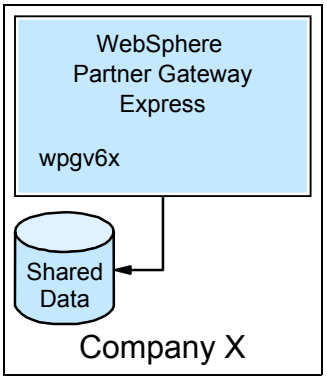


Figure 9-2 Single machine implementation

The installation of WebSphere Partner Gateway Express is a simple process:

1. Start the Launch Pad (see Figure 9-3 on page 265).
2. From here, you can read the readme file or connect to the documentation on the Internet or start the installation of the software itself.

Archived

Figure 9-3 Launching the installation of WebSphere Partner Gateway Express

3. Click **Install the product** to start the installation wizard, as shown in Figure 9-4 on page 266.

Archived

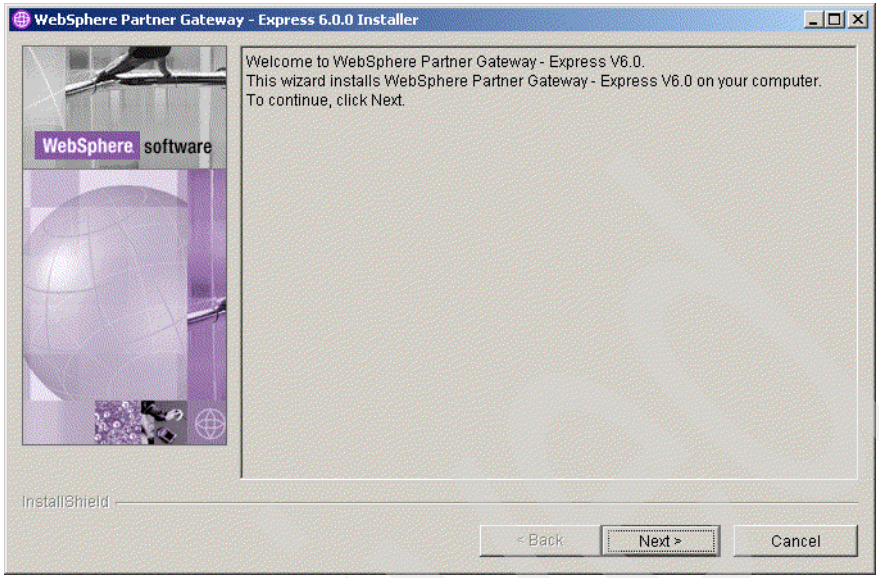


Figure 9-4 The installation wizard

4. Click **Next** to move to the license agreement screen.
5. Accept the license agreement.
6. Enter a complete path to where WebSphere Partner Gateway Express is to be installed, as shown in Figure 9-5 on page 267.

Archived

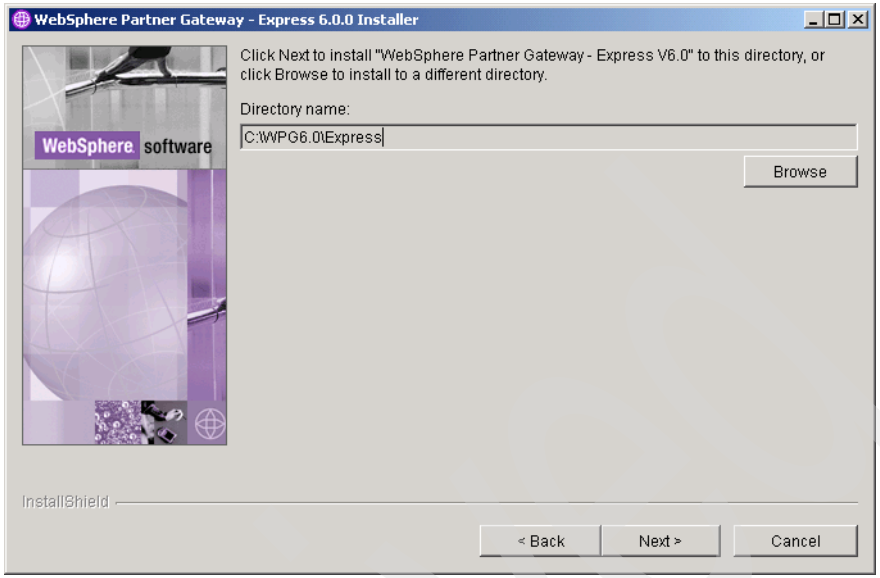


Figure 9-5 Enter the directory name for the installation

7. In the following windows, we specify the following details:
   - The port numbers for HTTP

     We accepted all of the suggested defaults.
   - The folder name in the Start Programs menu
   - The option to run WebSphere Partner Gateway Express as a service

     We chose **yes**.
   - The windows service name for WebSphere Partner Gateway Express

     We accepted the default.
8. After specifying this information, we see the summary window (Figure 9-6 on page 268), that lists all our selections.
9. Click **Next** to start the actual installation.

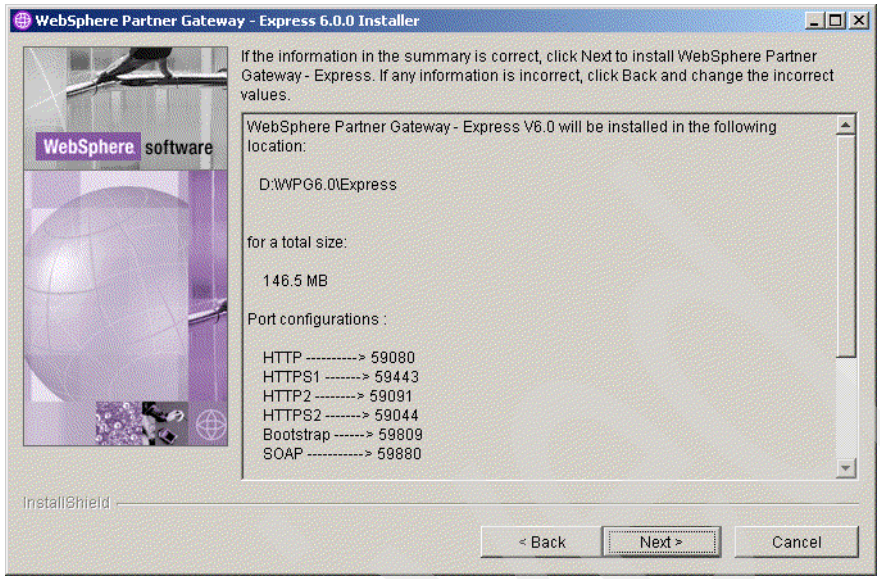


Figure 9-6 Installation summary

10. When the installation is complete, we are ready to start the server.
11. There are various methods to start and stop the server. Start the server using one of the methods outlined below:
    - Using the Windows Services application if you chose to create a service during the installation.
    - Using the shortcut in the Start Programs menu, select **Start → Programs → IIBM WebSphere partner Gateway - Express → Start Server.**
    - Using the **`bcgStartServer`** command as shown in Example 9-1.

Example 9-1 Start the server

```
C:\WPG6.0\Express\bin>C:\WPG6.0\Express\bin\bcgStartServer.bat

C:\WPG6.0\Express\bin>echo off
"Attempting to start bcgexpress server... this may take a few minutes."
ADMU0116I: Tool information is being logged in file

C:\WPG6.0\Express\was\profiles\bcgexpress\logs\bcgexpress\startServer.log
ADMU0128I: Starting tool with the bcgexpress profile
ADMU3100I: Reading configuration for server: bcgexpress
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server bcgexpress open for e-business; process id is 2580
```

Note: The stop server command can be issued by using the menu path or using the **bcgStopServer** command.

9.3 Initial configuration of the Express server

Complete the initial configuration of the Express server as follows:

1. If it is not already running, start the server.
2. Start the console application by selecting **Start** → **Programs** → **IBM WebSphere Partner Gateway - Express** → **Console.**

Note: This shortcut refers to a URL that includes a reference to the port number. If you set a nonstandard port in the installation, you might have to update this shortcut to use the custom port number. The URL behind this shortcut is :

```
http://localhost:59080/qc/index.jsp
```

3. For the initial logon, use the user ID of `admin` and the password of `admin`. Click **Login** (as shown in Figure 9-7).

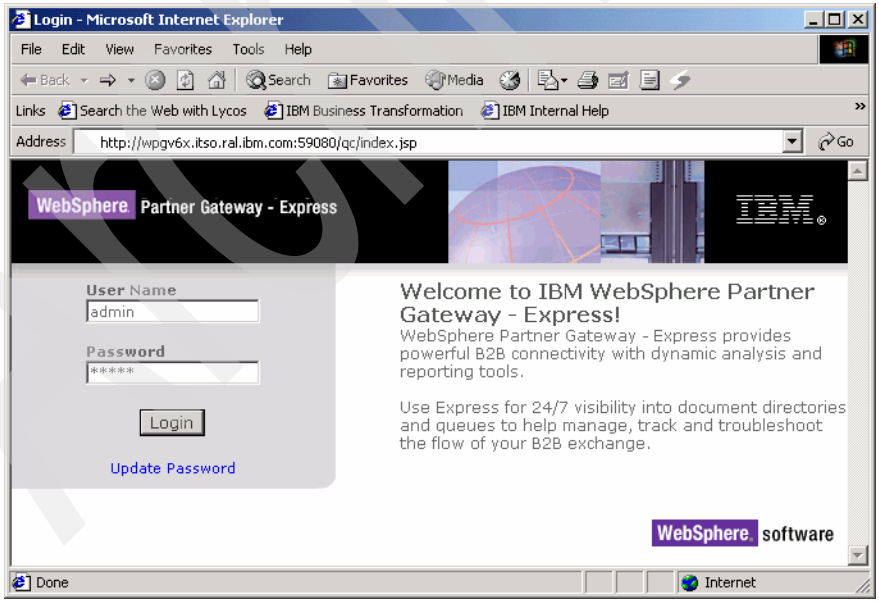


Figure 9-7 Initial logon to WebSphere Partner Gateway Express

4. During the first logon, you must change the password of the user ID admin. You must also change the password of another built-in user ID, Guest. Provide new passwords for both users and click **Save** (see Figure 9-8).

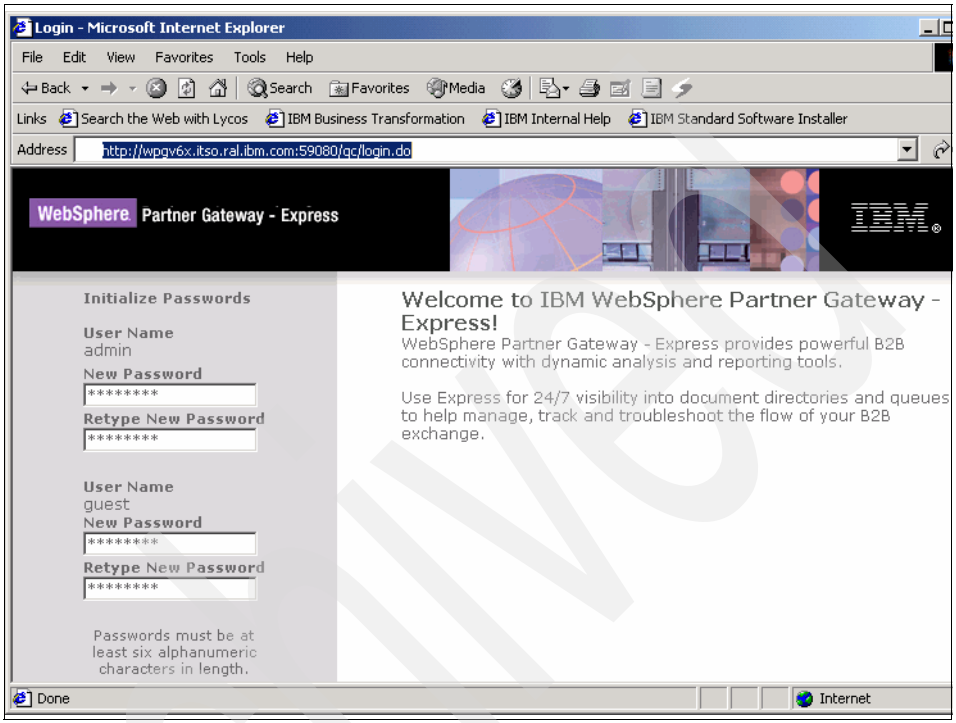


Figure 9-8 Initializing passwords during initial logon

5. After setting the new passwords, the application returns to the original main logon window.
6. Logon again, this time using the user ID `admin` and the new password. See Figure 9-9 on page 271.

Archived

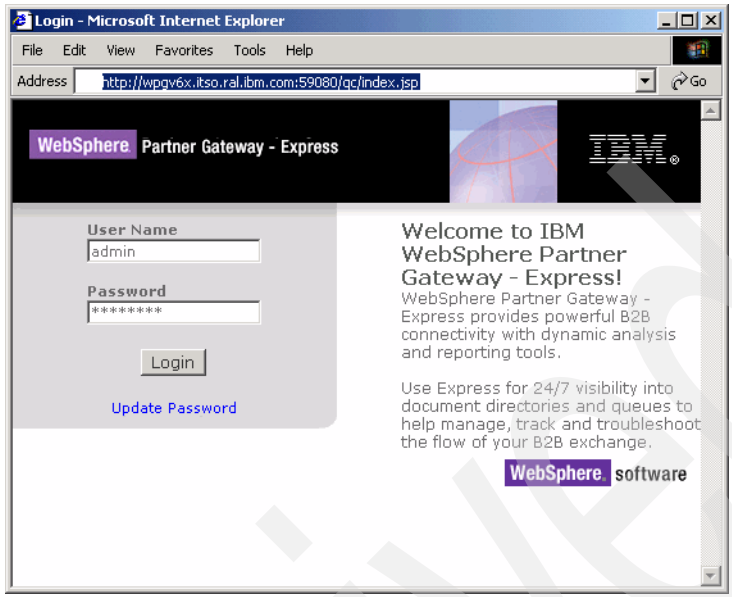


Figure 9-9 Logon using the new password

7. When the logon is complete, the Create Participant window is displayed. See Figure 9-10 on page 272.

Archived

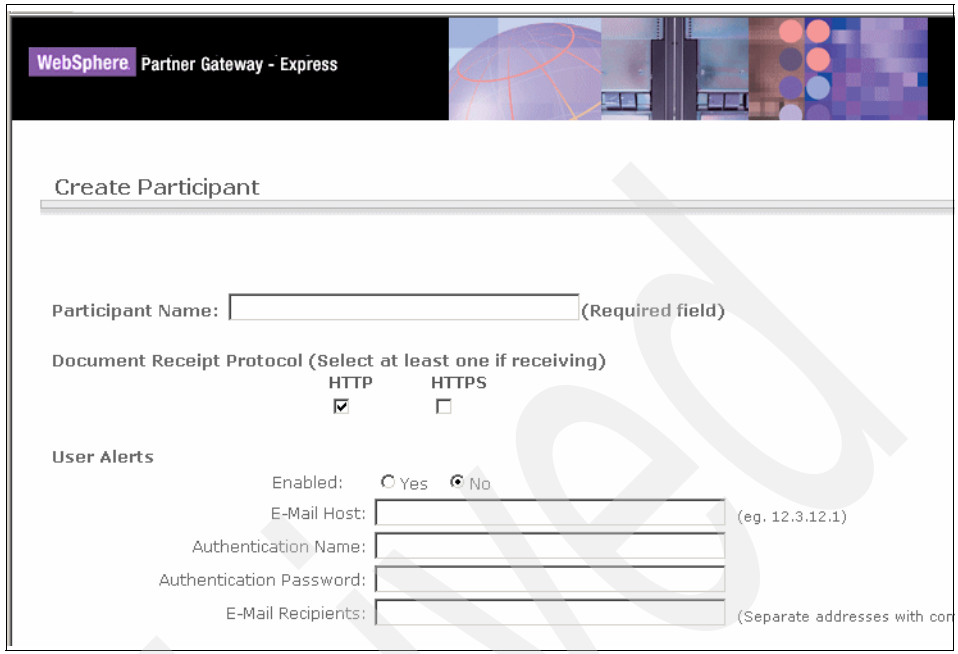


Figure 9-10 Create Participant initial screen

8. Provide basic information about the first WebSphere Partner Gateway Participant with whom the owner of this server of WebSphere Partner Gateway Express wants to communicate. Also specify the options or features that you want to use in that exchange.

   Except for the Participant Name field (the first one), you can change all parameters later:

   a. Provide a Participant name. We chose `companyE`.

   b. Select the inbound protocols you want to support: **HTTP**, **HTTPS** or both.

   c. If you want to activate user alerts, provide information about the e-mail server in your environment.

   See Figure 9-11 on page 273.

Archived

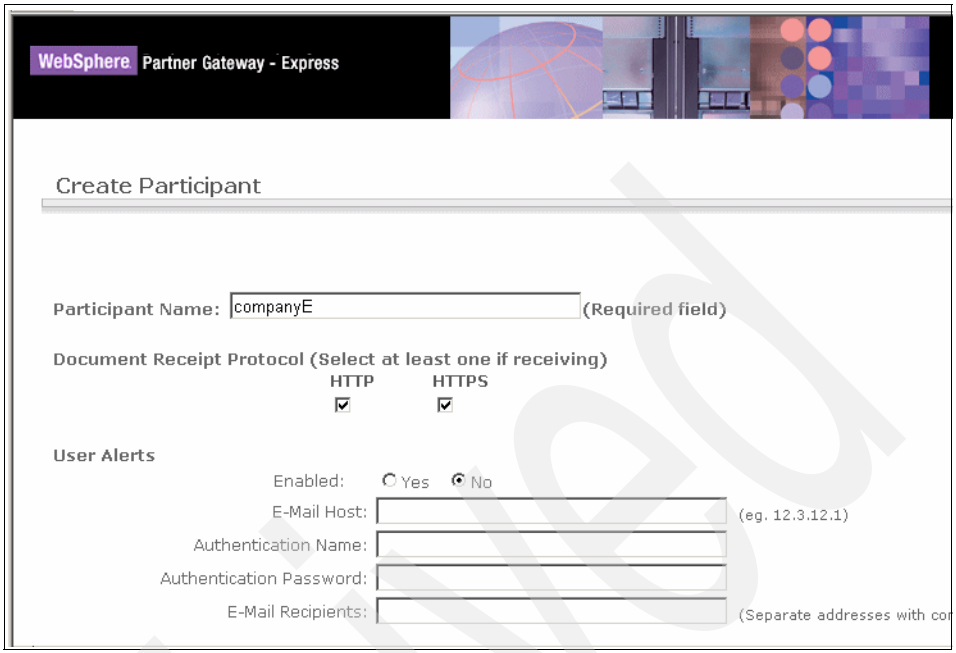


Figure 9-11 Create Participant (Part 1 of 2)

d. Scroll further down to the section labeled Capabilities (as shown in Figure 9-12 on page 274). Here, you indicate the kind of documents that can be received.

   Provide the AS2 Participant ID. You must provide this ID when you want to use AS2. It is used in the AS2 header information when sending documents to your partners.

e. You can transport many types of documents within an AS2 envelope. Some of these documents carry routing information and business identifiers, such as EDI. If you want to send or receive EDI documents, select the appropriate option to activate it and click **Save**.

Archived

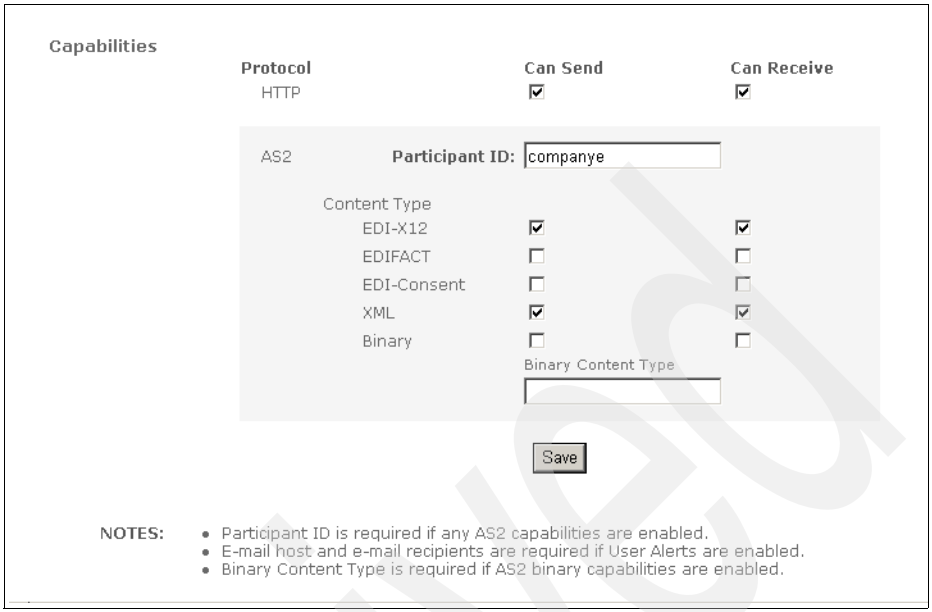


Figure 9-12 Create Participant (Part 2 of 2)

When the information entered in Figure 9-11 on page 273 and Figure 9-12 is stored, the normal interface of WebSphere Partner Gateway Express becomes available. First, in the Manage Participants window, shown in, you see that the configuration work is not yet finished. We complete this in Chapter 10, “Extending the B2B exchange” on page 277.

The first three menu options, **Reports**, **AS2**, and **HTTP**, provide several options for you to learn about the documents that are sent and received, AS2 exchanges and their state, and low-level HTTP information. The **Configuration** menu option provides you with all possible configuration settings, again grouped in logical units: participants, AS2 and HTTP. The **Security** menu option brings together all settings related to security including the management of certificates.

Archived

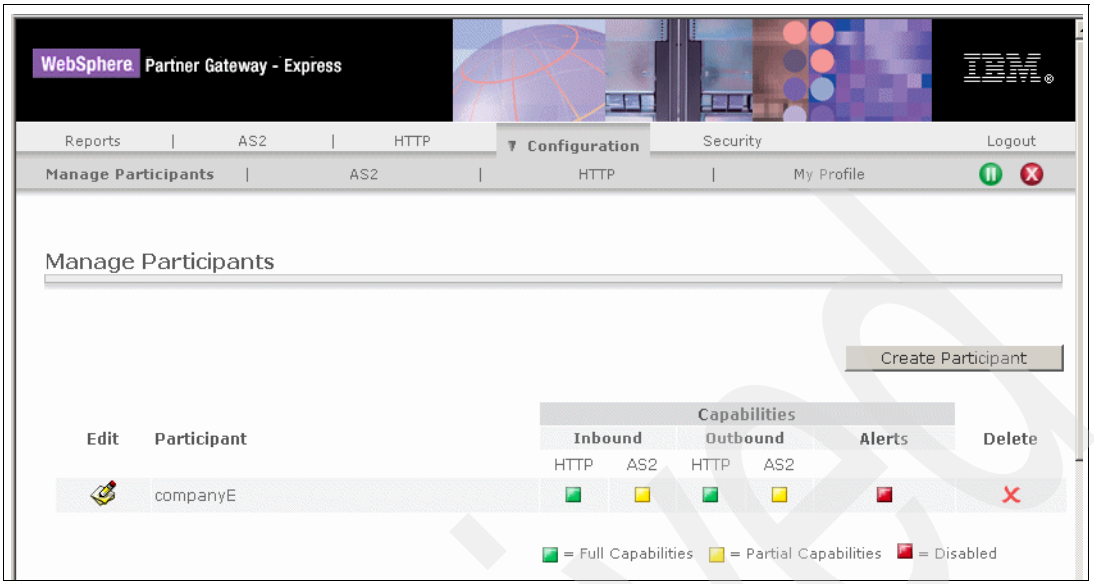


Figure 9-13 Initial configuration of WebSphere Partner Gateway Express

The initial configuration of the server is now complete and verified.

Upon saving the participant's profile, a number of directories are created on the file system. The directories are used to store both messages received from and sent to the participant. You can find the newly created directories in:

```
C:\WPG6.0\Express\data\FileSystemAdapter2\partners\companyE
```

This supposes that you have used the same installation directory and Participant name as our example.

Archived

Archived

10

Extending the B2B exchange

This chapter extends the base B2B environment by connecting the WebSphere Partner Gateway Express installation of Company X to the WebSphere Partner Gateway Enterprise installation of Company E. Company E and Company X exchange Custom XML documents over AS2.

Archived

© Copyright IBM Corp. 2005. All rights reserved.

10.1 Scenario overview

The XML documents that are exchanged are not in any specific industry or EDI standard. Company E and Company X have agreed upon a custom XML format that allows them to exchange business documents such as purchase orders, invoices, shipment notices and so on over the Internet. This custom XML format will be added to WebSphere Partner Gateway so that it can find the routing information in the XML document. When an XML document arrives at the receiver component of WebSphere Partner Gateway of Company E, WebSphere Partner Gateway looks for routing information in the XML document so that it can route the document to the trading partner for whom it is intended. This look-up process happens for both inbound and outbound documents.

Figure 10-1 shows a schematic overview of the B2B exchange among Company E, Company A, and Company X. This could give the impression that two different inbound streams are necessary or will be used.

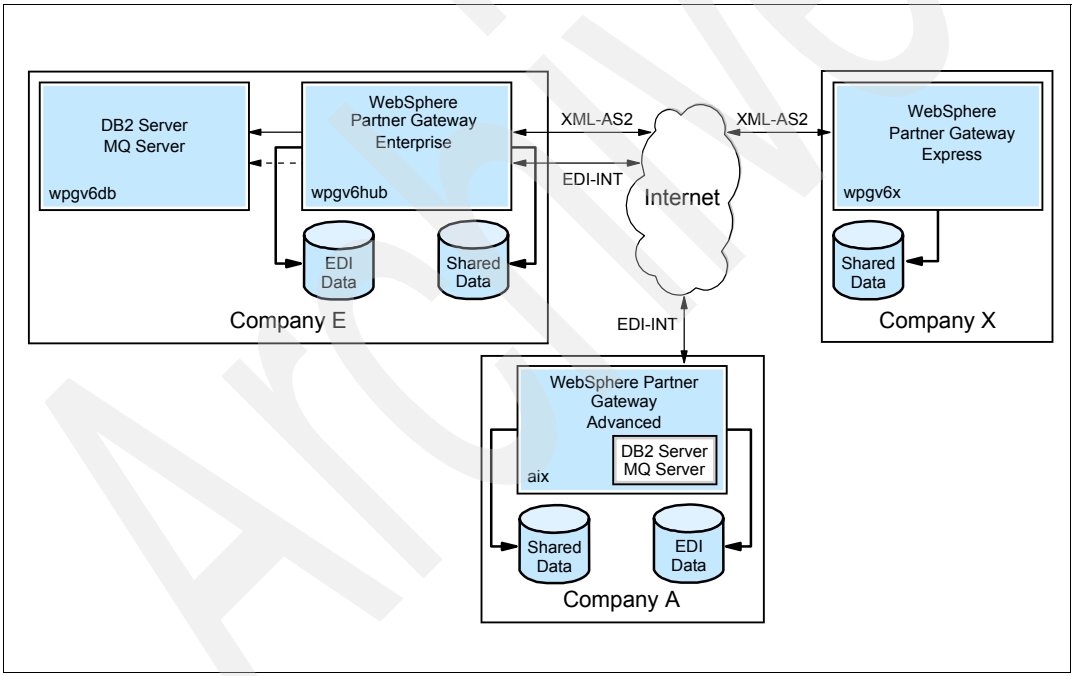


Figure 10-1 Scenario overview

In Chapter 7, “Creating a B2B exchange “on page 121, an HTTP Target was created to accept inbound electronic data interchange(EDI) documents from Company A. That object referred to the following URI.

```
http://wpgv6hub:57080/bcgreceiver/companye/edi_in
```

This does not necessarily mean that this Target will be used for receiving only EDI documents. For any document arriving at the Target URI, WebSphere Partner Gateway tries to find a matching definition based on the routing information in the document. We can create a separate Target for the reception of XML documents. However, there is no need for that.

This routing information usually will be present in the business documents. For example, in an EDI-X12 document, the routing information is present in the ISA segment. WebSphere Partner Gateway readily contains some of the EDI document flow definitions but does not yet contain a definition for the custom XML documents. By adding the Custom XML document flow definition to WebSphere Partner Gateway, we can make it ready for the routing of custom XML documents as well.

A similar discussion can be held about the file system gateway. Earlier, a file system gateway with the URI \wpgv6\data\companye\edi_in was defined. This folder was used by WebSphere Partner Gateway to store any received EDI documents. Depending on other processing that has to occur for incoming documents, we might prefer to store incoming XML documents in a different directory. Some back-office applications are easier to configure if XML and EDI documents are stored in a different folder. WebSphere Partner Gateway can do it either way.

The same applies for the file system target. EDI documents were picked up by WebSphere Partner Gateway from \wpgv6\data\companye\edi_out directory. We can, again, use this same folder again to send XML documents to Company X. If we chose to, we could create a new file system target if we chose.

10.2 Implementation steps

To enable the exchange of XML documents between Company E and Company X, we need to finalize the configuration of WebSphere Partner Gateway for Company X and perform additional configuration of WebSphere Partner Gateway Enterprise for Company E.

On the Company X server, we need to:

- Complete the profiles for Company X and Company E.
- Review the settings for HTTP and AS2.

Because WebSphere Partner Gateway Express does not really validate documents, it does not need to know the structure of the document is a custom XML format. The outbound routing is performed based on where the document is found on the file system.

On the Company E server, more work is needed:

1. Create a profile for the new partner - Company X.
2. Create a new document flow definition for the custom XML format.
3. Link this to the packaging models AS2, None, and Backend Integration.
4. Create an interaction for documents from None to AS2 packaging and vice versa.
5. When the new document flow and interactions are created, we can update the B2B Capabilities of the partners Company E and Company X.
6. Create and configure the participant connections for sending and receiving custom XML documents over AS2 between Company E and Company X.

10.3 Configuration of Company X

The tasks to perform by the administrator of WebSphere Partner Gateway Express can be divided into two categories. We need to update the Company X profile (known as Our Profile) and we need to update the profile of the Participant Company E that was created immediately after the initial logon.

10.3.1 Customizing My Profile

In configuring WebSphere Partner Gateway Express for both sending and receiving XML over AS2, we must review and edit My Profile. My Profile in WebSphere Partner Gateway Express contains both the properties of the machine on which WebSphere Partner Gateway Express is installed and the business details that relate to the organization that owns the installation.

1. To edit My Profile, select **Configuration** → **My Profile.** Click **Edit** to change the current values.
2. At a minimum, provide the host name and port for incoming AS2 documents and the AS2 identifier, for example, `companyx` (see Figure 10-2 on page 281). The complete URL for incoming AS2 documents is:

   `http://wpgv6x:59080/input/AS2`

   For plain HTTP, WebSphere Partner Gateway Express listens on:

   `http://wpgv6x:59080/input/HTTP`
3. Click **Save** to store the edited profile.

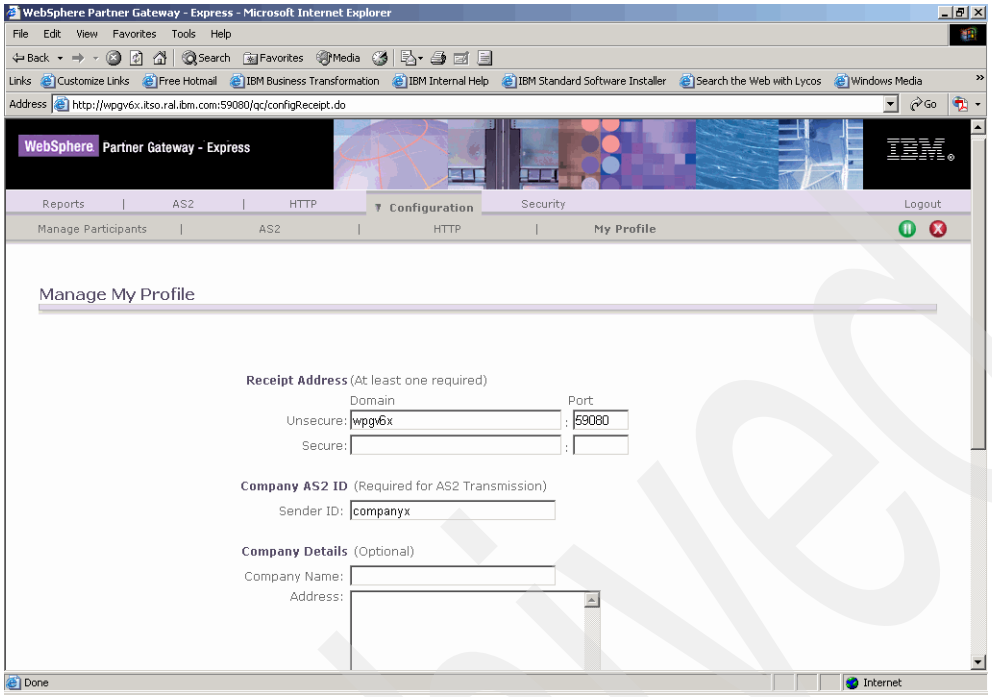


Figure 10-2 Managing My Profile for Company X

10.3.2 Customizing the profile of Participant Company E

During the initial startup, we created a partial profile for Company E, as the partner of Company X, which owns the WebSphere Partner Gateway Express server. We need to proceed by completing that profile.

To finish the setting for routing over HTTP, follow these steps:

1. Start the WebSphere Partner Gateway Express server and start the Console. Select **Start** → **Programs** → **IBM WebSphere Partner Gateway-Express** → **Console**.
2. Log on as admin user.
3. From the Main Menu of the console, select **Configuration** → **HTTP**.
4. In the Manage HTTP window, for the drop-down box, selected **Participant**, make sure that **Company E** is selected. Click **Edit.**
5. In the next window(Figure 10-3 on page 282), complete these tasks:
   a. For the inbound communication, provide a user name and password. For example, in Figure 10-3 on page 282, these entries were shown as

Archived

`compnaye,partner_e`. This needs to be used by the server of Company E when setting up a profile to send to Company X. WebSphere Partner Gateway Express uses this incoming user name and password to identify Company E and authenticate the sending server.

> **Note:** For inbound HTTP, the URL settings are copied from My Profile.

b. For the outbound communication, provide the URL on the server of Company E that is configured as a target. You do not need a user name and password for outbound communication, because WebSphere Partner Gateway uses other techniques for identifying the partner that sent the incoming document. If Company E was using WebSphere Partner Gateway Express as well, then it would be required to provide a user name and password. click **Save** to store the updated HTTP settings for communication with Company E.

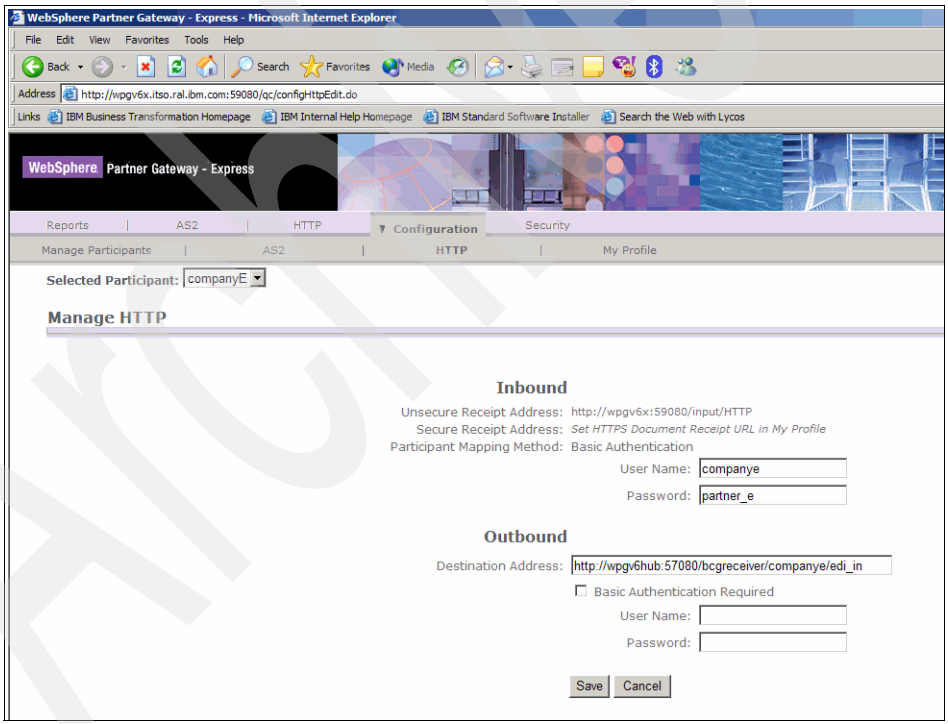


Figure 10-3 Updating the HTTP settings related to partner Company E

6. In addition to the HTTP settings, look at the settings for AS2. Select **Configuration** → **AS2**.

7. In the Manage AS2 window (Figure 10-4), for Selected Participant, select **Company E**. When you see the current settings, click **Edit**.
8. For the Inbound communication, select the **Identify Partner** radio button as **Using AS2ID**. For Outbound Communication, update the destination address for AS2 over HTTP. Set it to the target that is defined on the server of Company E. As no MDN is required for this communication, *deselect* the Request MDN box. For now, you can accept all other defaults and click **Save.**

Note: For inbound AS2, the settings are copied from My Profile.

.

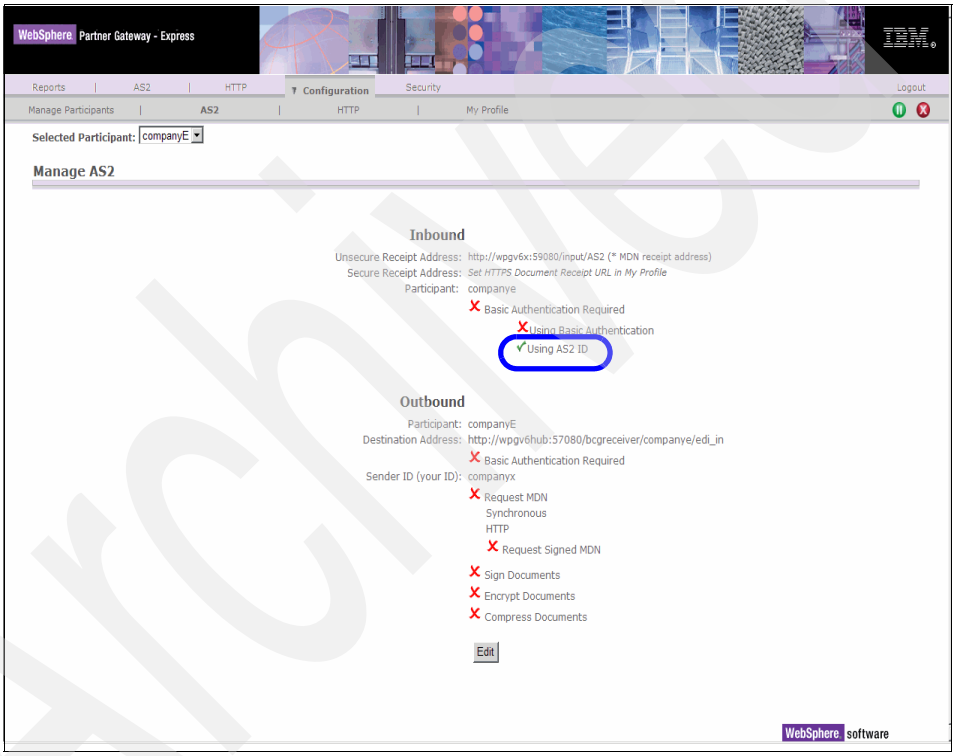


Figure 10-4 Editing the AS2 settings related to partner CompnayE

Tip: For testing purposes, you can enter the URLs in your browser. The browser prompts you for a user ID and password. Provide the values specified in Figure 10-3 on page 282. This is not a way to send documents, but only to see whether the server is active.

Archived

This finishes the Configuration required on the WebSphere Partner Gateway Express in order to communicate with WebSphere Partner Gateway Enterprise over Custom XML

10.4 Additional configuration of Company E

Given that WebSphere Partner Gateway Enterprise provides more features than WebSphere Partner Gateway Express, it comes as no surprise that the configuration of WebSphere Partner Gateway Enterprise includes a few more steps. Changes have to be made to the global profile of hubadmin and the profile of Company E. Also, a new profile for Company X has to be created.

The hubadmin must create the new partner and define a new protocol and document format.

10.4.1 Creating a new Community Participant

First, on the WebSphere Partner Gateway Enterprise hub, we must define a new Community Participant that identifies the WebSphere Partner Gateway Express server of Company X.

1. Log on as hubadmin to the console of WebSphere Partner Gateway Enterprise
2. Select **Account Admin** → **Profiles** → **Community Participant.** Click **Create.**
3. Use Table 10-1 as a guide to populate the fields in the New Participant window (Table 10-1). Remember that Participant Login Name is case-sensitive and is used as the company name in the logon window. The Freeform Business ID is used for routing purposes.
   a. Provide a login (or company) name for the new company, for example, `companyX`.
   b. Provide a Participant name, for example, `Company X`. This name is used in the interface.
   c. For Participant Type, select **Community Participant**.

Table 10-1 Attributes to populate when creating a new Participant

Attribute	Value
Participant Login Name	companyX
Participant Name	Company X
Participant Type	Community Participant

Attribute	Value
Freeform Business ID	companyx
Gateway	wpgv6x

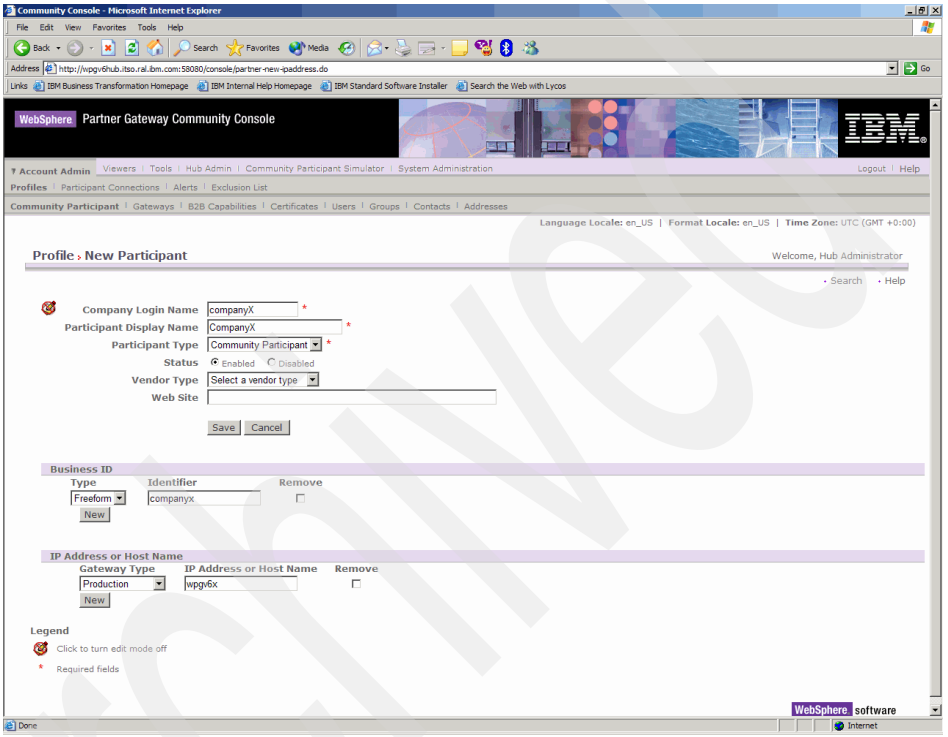


Figure 10-5 Creating a new Participant

10.4.2 Creating a new document flow definition

WebSphere Partner Gateway does not include a document flow definition for custom XML formats. Given that these XML formats are essentially freeform, each custom XML format to be processed by WebSphere Partner Gateway must first be defined as a document flow definition.

The hubadmin user must perform this task:

1. To create a new document flow definition for our custom XML format, select **Hub Admin** → **Document Flow Definition.**
2. When the existing document flow definitions are shown, click **Create Document Flow Definition.**

3. The document flow definition that we create is at the Protocol Level, which means that this XML document is defined at the same level as an EDI standard, for example. Use the values in Table 10-2 for this new document flow definition (Figure 10-6 on page 287).

Table 10-2 Values to use when creating a document flow definition at Protocol level

Attribute	Value
Document flow type	Protocol
Code	FREEFORM_XML
Name	FREEFORM_XML
Version	ALL
Document level	No
Status	Enabled

a. Set Visibility to **Yes** for the:

- Community Operator
- Community Manager
- Community Participant

b. Add this new protocol to:

- **Package: AS**
- **Package: None**
- **Package: Backend Integration**

c. After you specify all values, click **Save**.

Archived

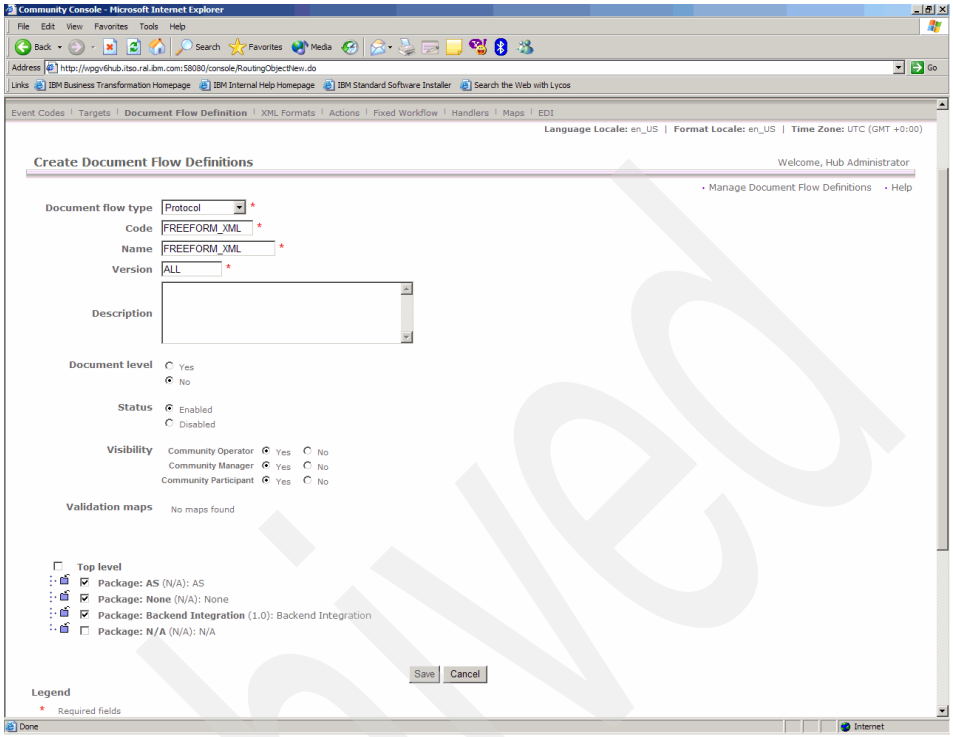


Figure 10-6 Creating a document flow definition at protocol level

4. We also need to create a document flow definition at the document flow level.
5. Click **Manage Document Flow Definitions** or select **Hub Admin Document Flow Definition** from the menu.
6. Click **Create Document Flow Definition**.
7. Use the values in Table 10-3 to complete the form shown in Figure 10-7 on page 289.

Table 10-3 Values to use when creating a document flow definition at doc-flow level

Attribute	Value
Document flow type	Document flow
Code	FORECAST
Name	FORECAST
Version	ALL

Archived

Attribute	Value
Document Level	Yes
Status	Enabled

a. Set Visibility to **Yes** for the three roles, as shown in Figure 10-7 on page 289.

b. At the bottom of Figure 10-7 on page 289 you see the expandable tree structure of packages and protocols. Expand this structure.

c. Add this new document flow to:

- **Package: AS**

  → **Protocol: FREEFORM_XML**

- **Package: None**

  → **Protocol: FREEFORM_XML**

- **Package: Backend Integration**

  → **Protocol: FREEFORM_XML**

Note: We can compare the definition of this document flow to adding the definition of an 850 purchase order document to the EDI standard. If other XML documents needed to be defined, we could add them directly at the document flow level. There is no need to create another document flow definition at the protocol level.

Archived

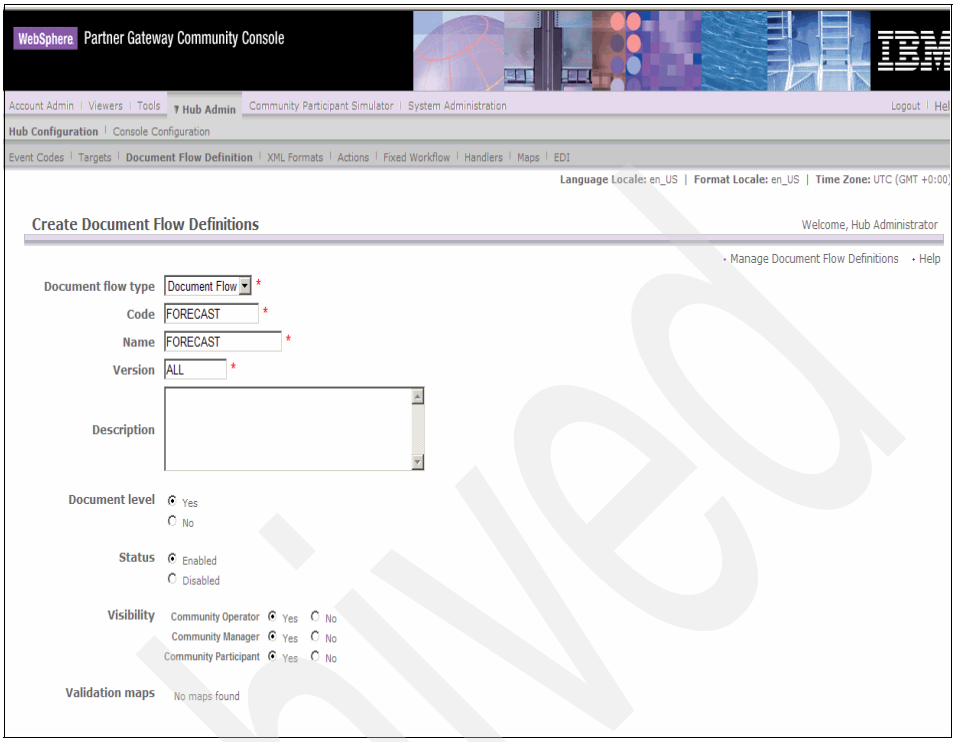


Figure 10-7 Creating a document flow definition at document flow level

d. Figure 10-8 on page 290 shows the completed structure. Click **Save.**

Archived

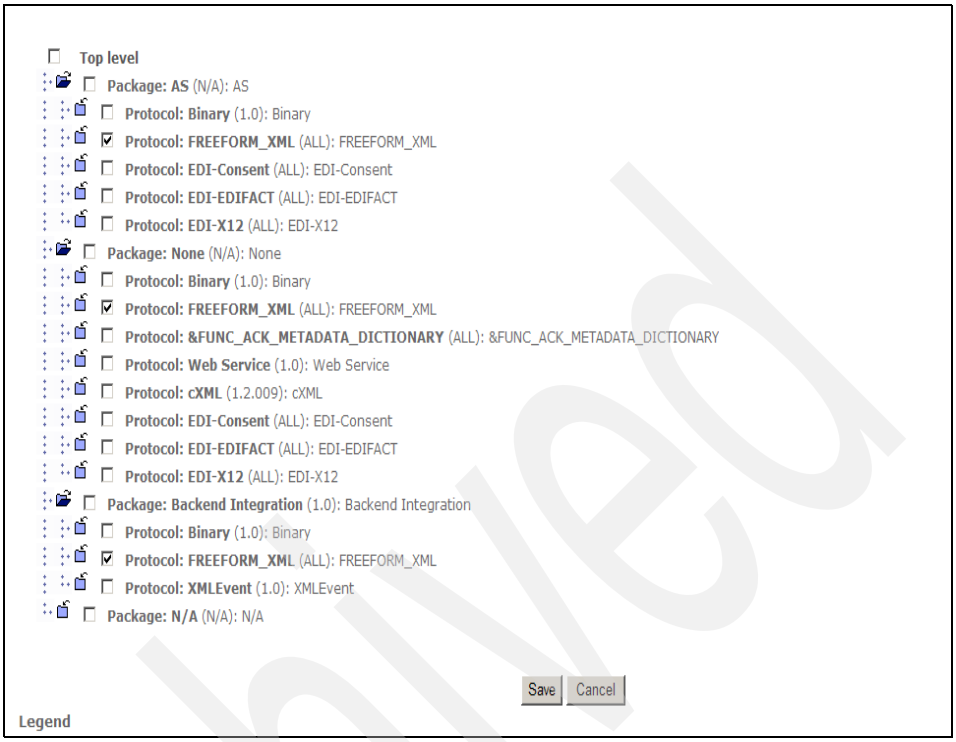


Figure 10-8 Link document flow to package and protocols

The next step is to create a new XML format and tie it to the document flow that we created.

10.4.3 Creating a new XML format

The main reason to define the XML format to WebSphere Partner Gateway is to make sure that the server knows how to route the XML document. It must use the data in the XML document to determine to which partner a document should be sent.

1. In the console of WebSphere Partner Gateway, while logged on as hubadmin, select **Hub Admin** → **Hub Configuration** → **XML Formats**.
2. The Manage XML Formats window (Figure 10-9 on page 291) opens. Click **Create XML Format.**

Archived

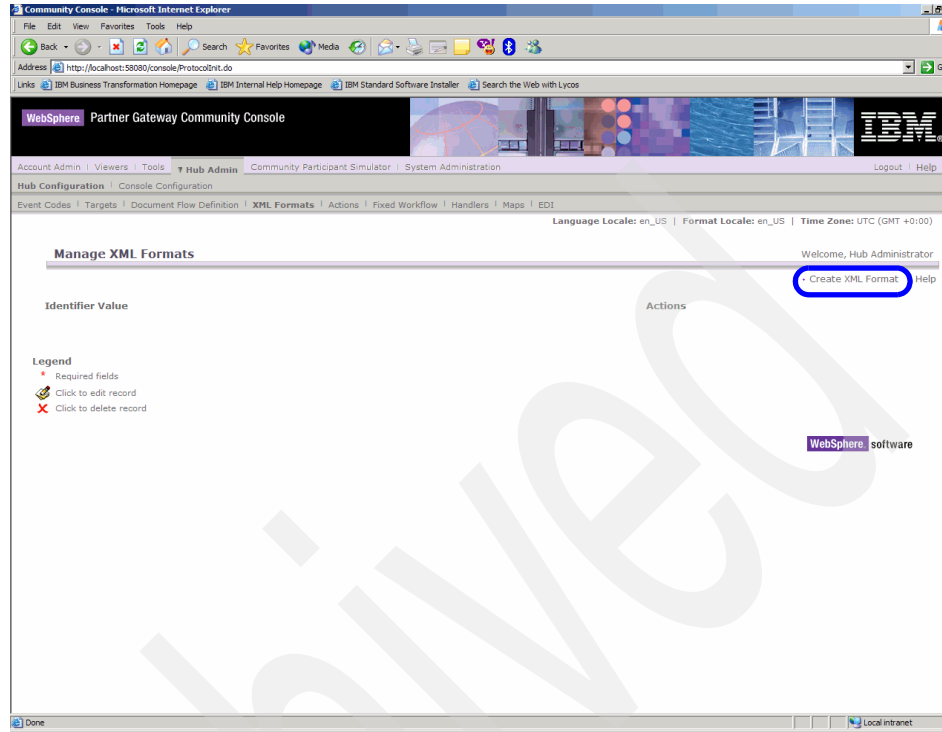


Figure 10-9 Manage XML formats in WebSphere Partner Gateway

3. We see the View XML Format window (Figure 10-10 on page 293). The new format needs to link to an existing protocol, which we defined earlier.
   a. For Routing Format, select **FREEFORM_XML.**
   b. For File Type, select **XML**.
   c. For Identifier Type, select **Root Tag**. The root tag in our XML documents is `forecast`. Another way to identify an XML document is to use a document type definition (DTD) or name space.
   d. In the Schema Attributes section, we tell WebSphere Partner Gateway how to identify the source and target business identifiers. These can be set as a constant.

Note: In most cases, you use an element in the XML document as the carrier for information about business identifiers. That way, it becomes easier to send this kind of XML document to multiple partners.

Archived

Example 10-1 shows a sample XML document that indicates the element names that store business identifiers.

Example 10-1 Sample XML forecast message

```
<?xml version="1.0"?>
<forecast>
 <forecastHeader>
  <customerNumber>companyx</customerNumber>
  <orderNumber>P4387654</orderNumber>
  <orderFlag>UO</orderFlag>
  <orderDate>26072005</orderDate>
  <supplierNumber>companye</supplierNumber>
  <supplierName>Company X</supplierName>
 </forecastHeader>
 <forecastDetail>
  <lineNumber>1</lineNumber>
  <partNumberRequired>00024L6654</partNumberRequired>
  <unitPrice>4.45</unitPrice>
  <quantity>150,000</quantity>
  <uom>EA</uom>
  <itemDesc>WIDGET_A</itemDesc>
  <period>31122005</period>
 </forecastDetail>
 <forecastTrailer>
  <totalDollars>667,500</totalDollars>
  <totalQuantity>150,000</totalQuantity>
 </forecastTrailer>
</forecast>
```

Figure 10-10 on page 293 shows the form for creating a new XML format.

e. In addition to information about business identifiers, WebSphere Partner Gateway needs to know which document flow to invoke for XML documents of this format. This information can again be carried in the XML document itself. However, we make this information constant. The source document flow is FORECAST and the version is ALL. The document flow was defined earlier (see Figure 10-7 on page 289).

f. Click **Save** to store the new format.

Archived

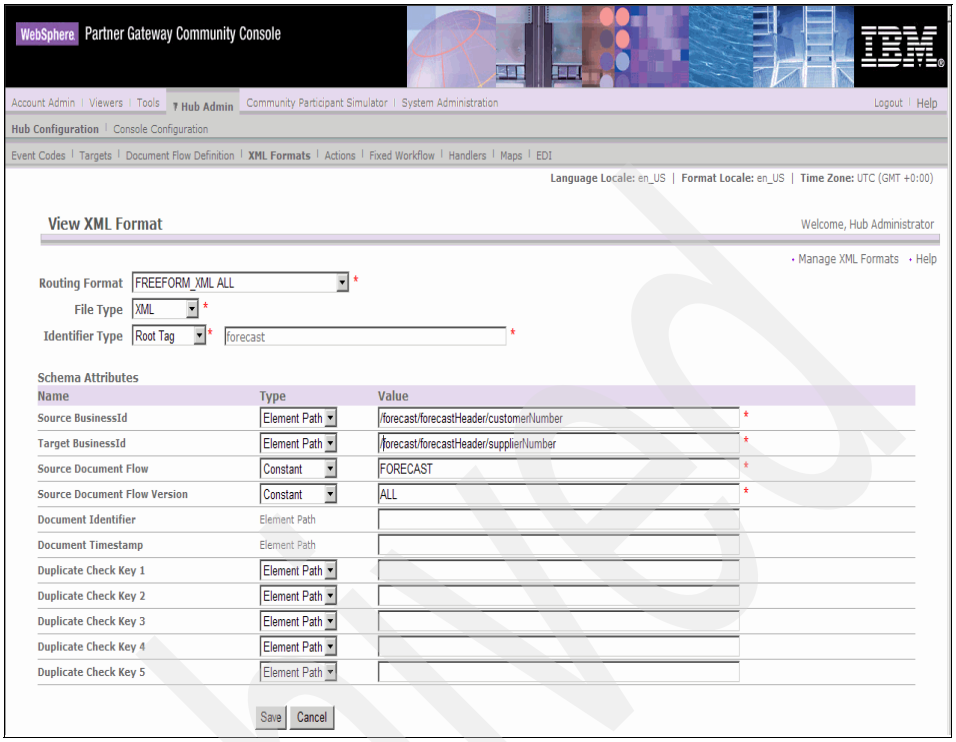


Figure 10-10 Creating a new XML format

10.4.4 Create an interaction

Earlier, we created the protocol FREEFORM_XML that can be packaged in AS2, None, or Backend Integration. Then we created the document flow FORECAST that was linked to the protocol FREEFORM_XML for each packaging method. Next we defined the XML format to link to that document flow. Now we create the interactions that tell WebSphere Partner Gateway how to move from one packaging, protocol, or document to another.

1. While logged on as hubadmin, select **Hub Admin** → **Hub Configuration** → **Document Flow Definition**.
2. Click **Manage Interactions**.
3. Click **Create Interaction**.
4. We need an interaction from AS/FREEFORM_XML/FORECAST to None/FREEFORM_XML/FORECAST, which is the interaction shown in Figure 10-11 on page 294. For Action, select **Pass Through** and click **Save**.

Archived

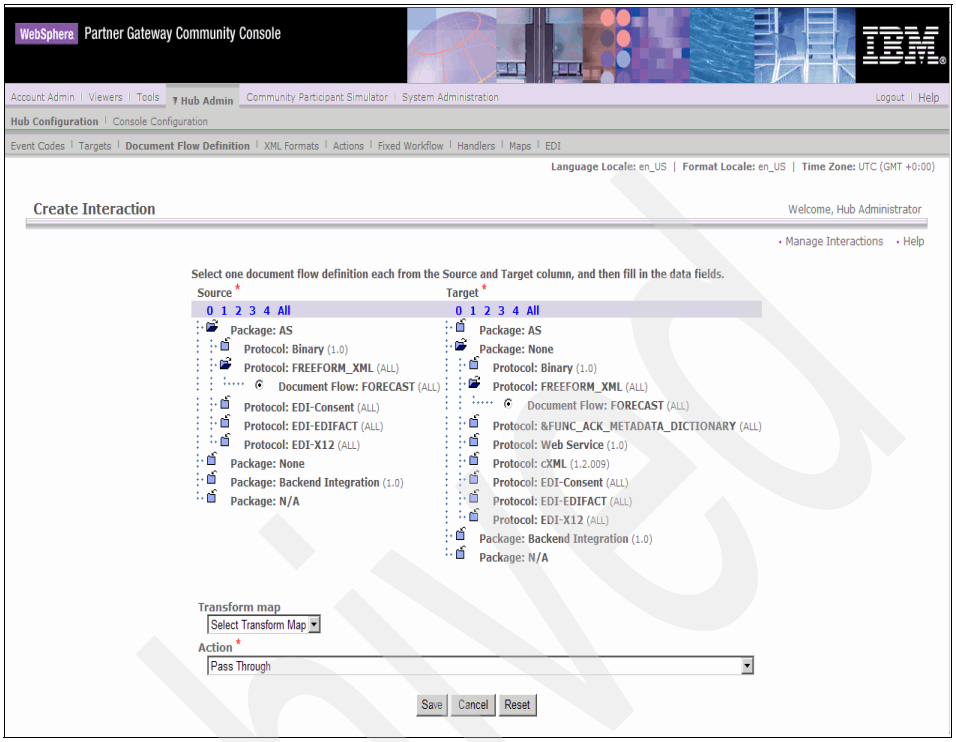


Figure 10-11 Creating an interaction from AS to None for FREEFORM_XML

5. Repeat the same steps to create the reverse interaction, from None/FREEFORM_XML/FORECAST to AS/FREEFORM_XML/FORECAST. This interaction is shown in Figure 10-12 on page 295. Again set Action to **Pass Through** and click **Save**.

Archived

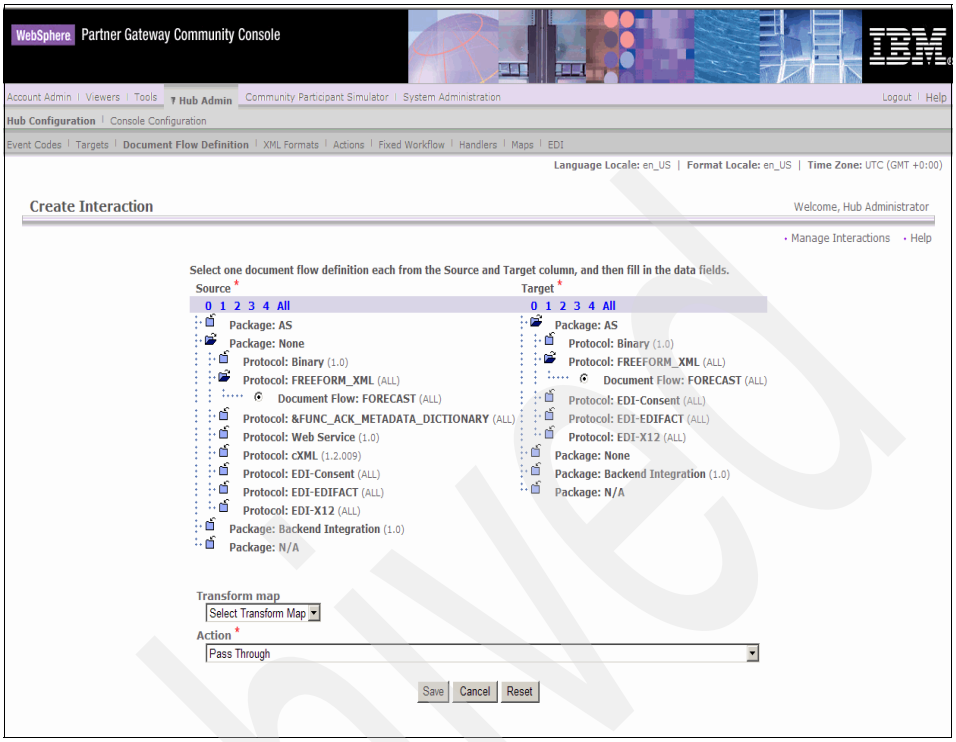


Figure 10-12 Creating an interaction from None to AS for FREEFORM_XML

10.4.5 Updating Company X's profile on Company E's server

When we configured the AS2 communication between Company E and Company A in Chapter 7, "Creating a basic B2B exchange" on page 121, we described the configuration process in a role-based way. The hubadmin and the administrators of Company E and Company A had their own tasks to complete. Remember that as an administrator of a company, you have to perform tasks on your own server and on the server of your partner.

In 7.8, "Revisiting role-based configuration" on page 184, we explained that the role-based approach is not required. For example, some companies might not feel comfortable about the fact that their partners log on directly to their systems. Another reason why you might not want to follow the role-based approach is the use of different software products by your partner. You might not want to educate your partners about WebSphere Partner Gateway if they use a different AS2 provider.

This is why we do not follow the role-based approach in this section. Company X uses the Express edition of WebSphere Partner Gateway. It is not familiar with

the console interface of the Advanced or Enterprise editions of WebSphere Partner Gateway. Therefore, the hubadmin updates the profile of Company X on behalf of the administrator of Company X.

1. While logged on as hubadmin, select **Account Admin** → **Profiles** → **Community Participant**.
2. In the Participant Search window (Figure 10-13), click **Search**. There is no need to provide a search argument because only four profiles are defined. When you see the list of profiles, click the icon next to Company X.

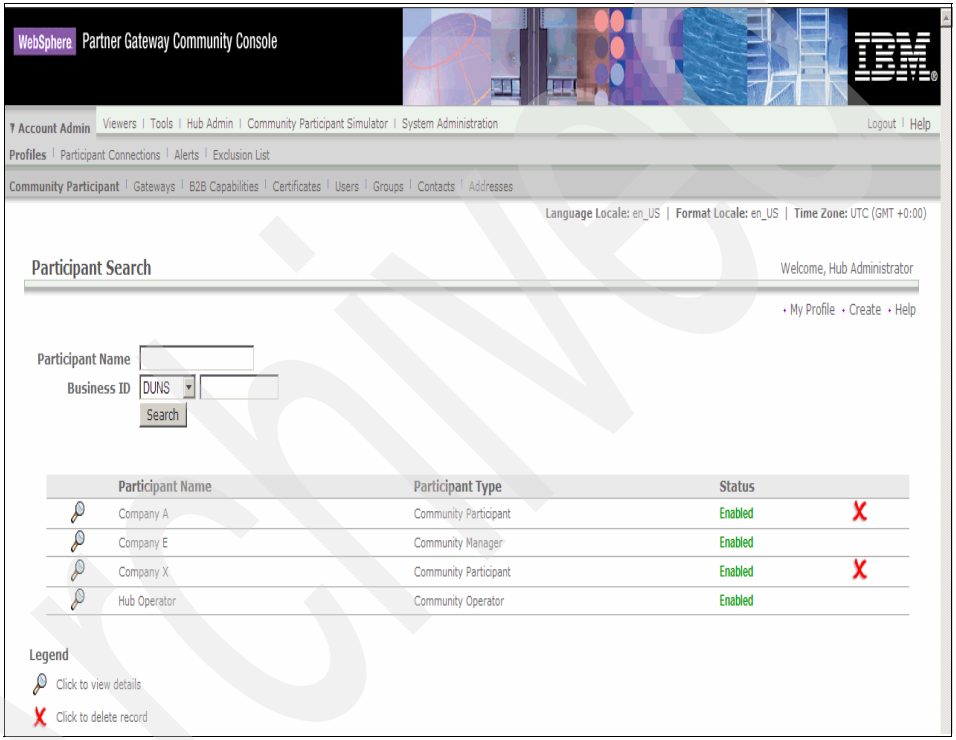


Figure 10-13 List of participants at Company E

3. Set the B2B capabilities of partner Company X as shown in Figure 10-14 on page 297.
   a. Select **B2B** Capabilities.
   b. Verify that you are working with the profile of Company X.
   c. Enable AS, FREEFORM_XML, and FORECAST as the source and target.
   d. Enable None, FREEFORM_XML, and FORECAST as the source and target.

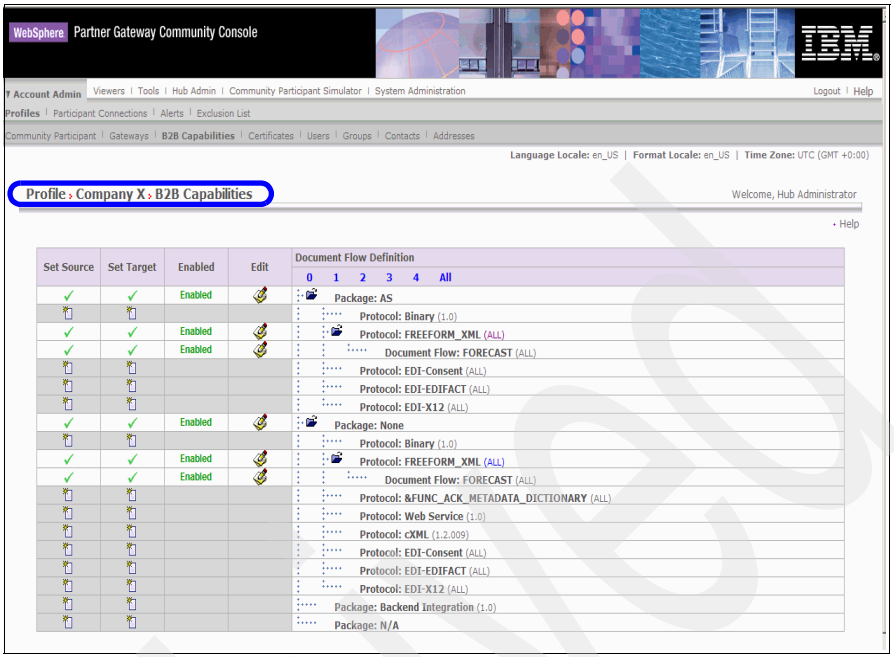


Figure 10-14 Setting the B2B capabilities of Company X on server Company E

4. Create the HTTP gateway that points to the WebSphere Partner Gateway Express server of Company X. While working with the profile of Company X, select **Gateways.**
5. When you see the list of gateways, click **Create**.
6. Use the values in Table 10-4 on page 298 to complete the form in the Gateway List window (Figure 10-15 on page 298).
   a. For the attribute Forward Proxy List, select **Use no forward proxy**.

   > **Note:** Routing through proxies is a new enhancement introduced with WebSphere Partner Gateway 6.0.

   b. The attributes User Name and Password need values, because Company X hosts WebSphere Partner Gateway Express. These attributes were not used when we created a gateway for Company A, for example. We created the user name companye and a password during the profile customizing of Company E on the WebSphere Partner Gateway Express server of Company X. This user name is not defined on the server of Company E and is not related to business identifiers.

Archived

c. For the attribute Configuration Point Handlers, Select **None** because no user Exit handlers will be used for this particular scenario

d. Restore the default values for the attributes, Retry Count, Retry Interval, Number of threads, Validate Client IP, Auto Queue and Connection Timeout. See figure Figure 10-15.

Table 10-4 Creating the default gateway for Company X

Attribute	Value
Gateway Name	AS2Gateway
Transport	HTTP/1.1
Forward Proxy List	Use no forward proxy
Target URI	http://wpgv6x:59080/input/AS2
User Name	companye
password	partner_e
Configuration point Handlers	Select **None**

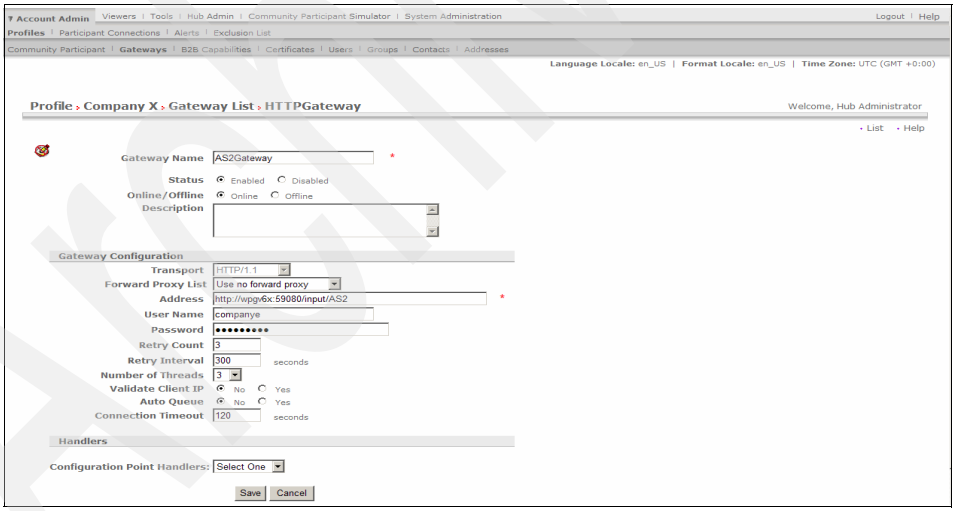


Figure 10-15 Creating a new gateway AS2Gateway

7. When the gateway is created, set it as the default gateway for Company X. Return to the list of gateways for Company X and click **View Default Gateways.**

8. For the production gateway, select **AS2Gateway** and click **Save**. You should see the list of gateways for Company X as shown in Figure 10-16 on page 299.

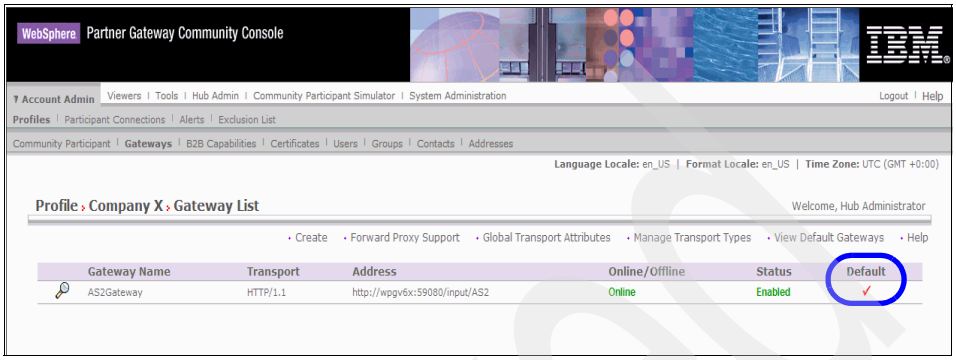


Figure 10-16 List of Gateways for Company X

10.4.6 Updating Company E's profile on the Company E server

In Chapter 7, "Creating a basic B2B exchange" on page 121, we set the B2B capabilities of Company E so that it could send and receive EDI documents over AS2. Now we must update these B2B capabilities to include custom XML documents.

1. While logged on as hubadmin, open the profile for Company E.
2. Select **B2B Capabilities.**
3. In the B2B Capabilities window (Figure 10-17), update the capabilities by enabling AS, FREEFORM_XML, and FORECAST, as well as None, FREEFORM_XML, and FORECAST.

Archived

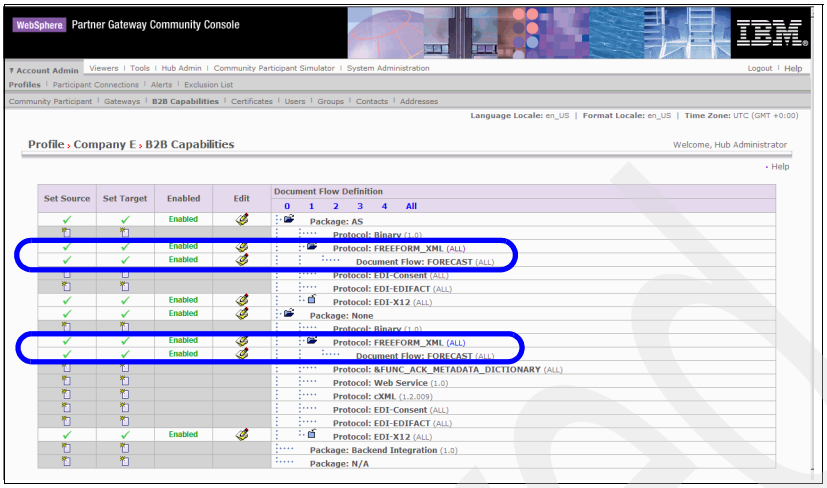


Figure 10-17 Updating the B2B capabilities of Company E

There is no need to define another gateway. While the existing gateway refers to edi_out in its URI, there is no need to create a specific gateway for XML. Now you must create new participant connections, the final step in the configuration.

1. While logged in as hubadmin, select **Account Admin** → **participant connections.**
2. Select **Company E** as the source and **Company X** as the target. Click **Search**.
3. Locate the connection that takes an unpackaged XML document and sends an AS2 packaged document. See Figure 10-18. Click **Activate**.

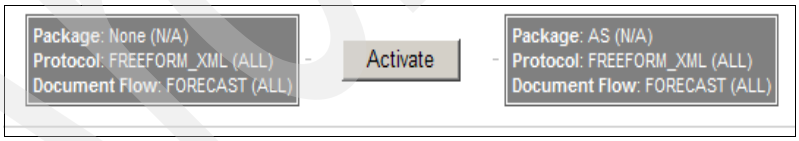


Figure 10-18 Connection to send AS2-XML to Company X

4. Click **Attributes** for this connection to set AS2 attributes in the same way as done before.
5. Set the **AS MDN Requested** attribute to **No.** Click **Save.**

This almost completes the configuration required on the WebSphere Partner Gateway Enterprise server to allow for the sending of a custom XML format over AS2 to a WebSphere Partner Gateway Express server.

Archived

Before we send a sample XML file over AS2 from the WebSphere Partner Gateway Express server to the WebSphere Partner Gateway Enterprise, we must activate another connection.

Return to the search window for participant connections. This time, set **Company X** as the source and **Company E** as the target. Click **Activate** for the connection that takes an AS packaged document and turns it to no-packaging (see Figure 10-19).

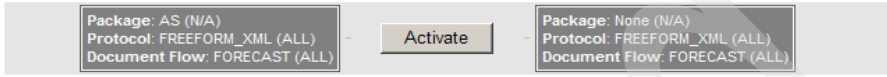


Figure 10-19 Connection to receive AS2-XML documents from Company X

Archived

10.5 Validating Communication

We are now ready to validate the configuration by sending XML documents between both companies.

10.5.1 Sending XML documents from Company E to Company X

All should be in place. To test the configuration, again we can use the test connection feature first. Assuming that this feature does not report an error, it is sufficient to drop an XML file in the edi_out folder used previously to send EDI documents to Company A. WebSphere Partner Gateway can determine that the document is an XML document. There is no need to sort any documents in format-specific directories or partner-specific directories.

Compared to the processing that happened between Company E and Company A, there are a few differences:

- Having picked up the file and wrapped for AS2, WebSphere Partner Gateway Enterprise must first establish an HTTP connection with WebSphere Partner Gateway Express. This connection is established only on the presentation of a valid user name and password by the WebSphere Partner Gateway Enterprise server. In essence, the WebSphere Partner Gateway Enterprise hub is logging into the servlet employed by WebSphere Partner Gateway Express for receipt of HTTP POSTs.
- When this connection is established, WebSphere Partner Gateway Express knows the source of the message. The user name and password should match the detail provided when creating the HTTP configuration for the Participant in WebSphere Partner Gateway Express.
- TheWebSphere Partner Gateway Express server determines whether packaging is to be stripped by the URI on which it receives.
- Messages sent to a URI with the extension /input/HTTP are received and placed in the file system in the folder HTTP. Messages sent to a URI with the extension input/AS2 are assumed to have AS2 MIME headers. These messages are stripped of any AS2 headers.
- The payload, in our case an XML file, is placed in the file system in the folder AS2. The complete directory is called:

```
<root dir>\data\FileSystemAdapter2\partners\companyE\received\AS2\XML
```

Besides looking at the destination folder, we can use the AS2 Viewer on the server of Company E to learn whether the transmission succeeded. Alternatively, we can use the viewers that are available in WebSphere Partner Gateway Express

1. Log into the console of WebSphere Partner Gateway Express as the admin user.
2. From the menu, select **AS2** → **Received.**
3. Enter any search parameters and click **Search**. You see a list of received documents that includes some statistical information, as shown in Figure 10-20.

Similar to the AS2 Viewer for WebSphere Partner Gateway Enterprise, you can obtain more details by selecting the **magnifying glass** icon, and look at the actual document.

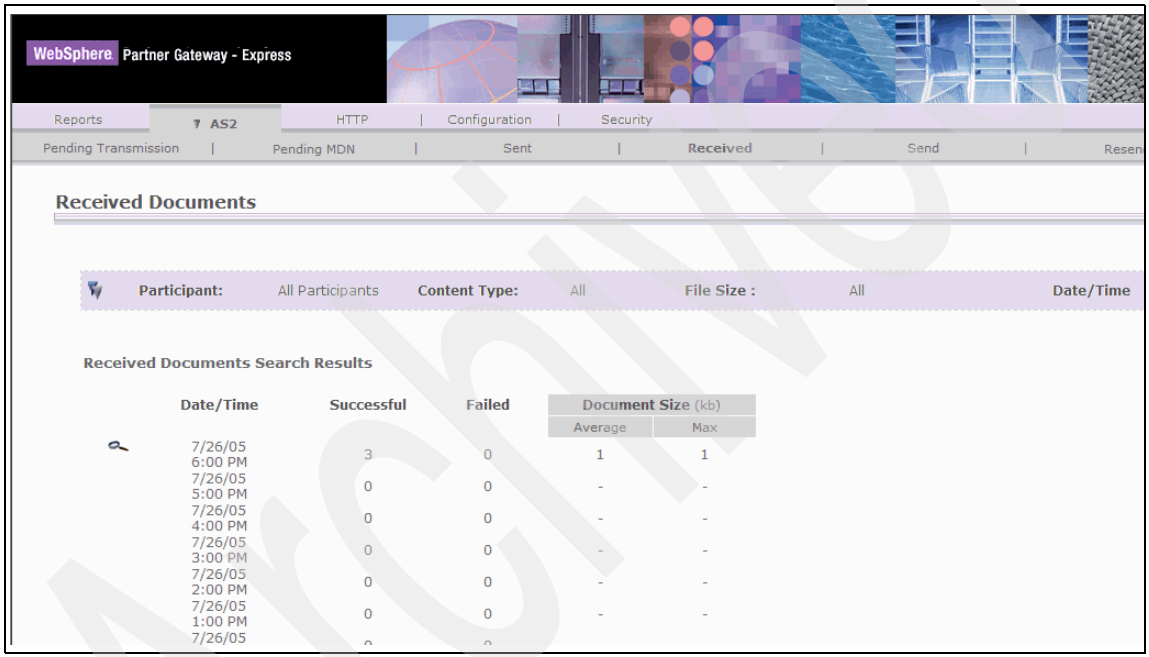


Figure 10-20 AS2 Received Documents Viewer

10.5.2 Sending XML documents from Company X to Company E

We can now send messages from the WebSphere Partner Gateway Express server by either using the WebSphere Partner Gateway Express console or by dropping a file in the right directory.

Manual sending

For manual sending, we use these steps:

1. While logged into the console as the admin user, select **AS2** → **Send**.

2. Select **Company E** as the participant. Set the content type to **XML**. Click **Browse** to locate the file being sent on the file system. Click **Send** to initiate the transmission (Figure 10-21).

The file is wrapped with AS2 MIME headers and sent to Company E over HTTP. We can see the message in the AS1/AS2 Viewer of Company E as before.

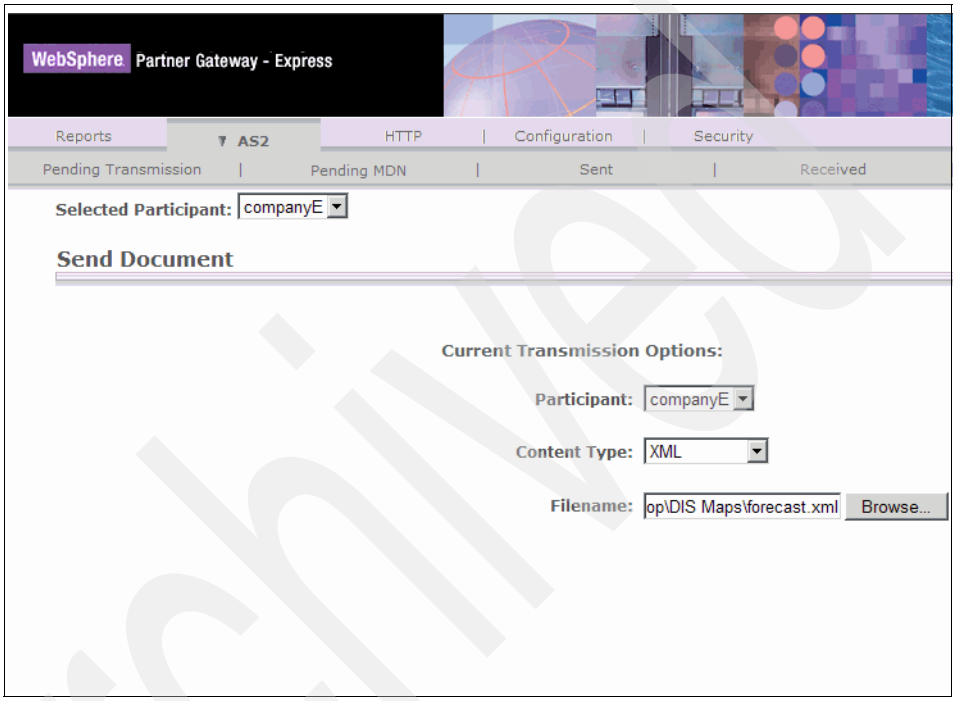


Figure 10-21 Manually sending an XML document

3. To verify the transmission, use the AS2 Viewer of WebSphere Partner Gateway Express to follow these steps:
   a. Select **AS2** → **Sent.**
   b. Provide any search arguments.
   c. Click **Search**.

We see a list of sent documents and some statistical information as shown in Figure 10-22 on page 305.

Archived

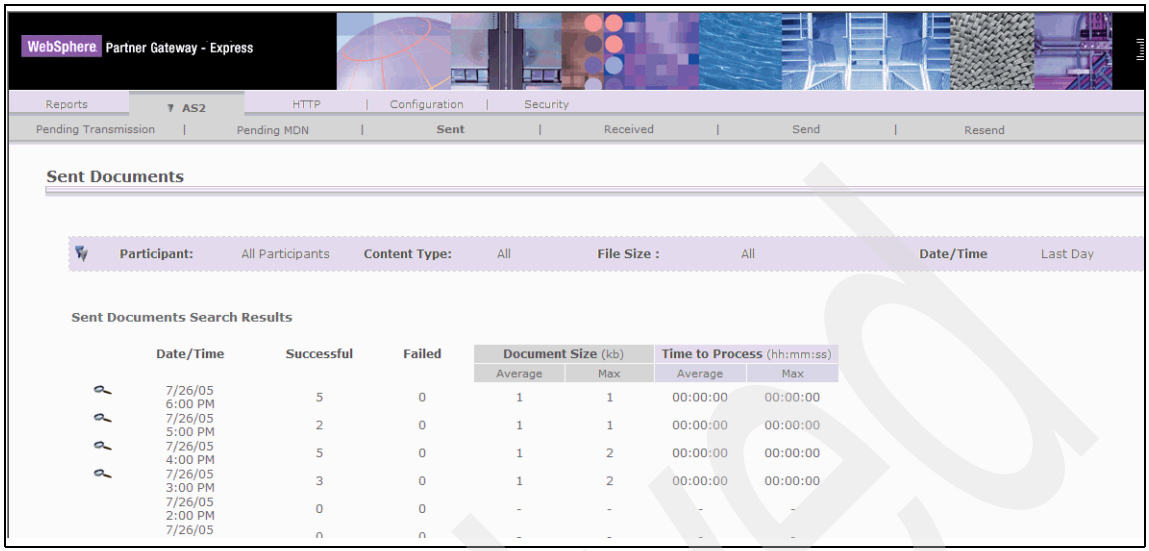


Figure 10-22 AS2 Sent Documents Viewer

Automatic sending

.The second means of sending files from the WebSphere Partner Gateway Express server is to use the file system. Similar to WebSphere Partner Gateway Enterprise, WebSphere Partner Gateway Express can read files from the file system and send to participants over HTTP or HTTP using AS2. Unlike WebSphere Partner Gateway Enterprise, it is not possible to define from which directory WebSphere Partner Gateway Express reads the files to be sent.

For each partner, for each protocol, and for each document format, there is a directory from which files are read by WebSphere Partner Gateway Express. For XML documents sent over AS2 to Company E, the name of the directory is:

```
<root dir>\data\FileSystemAdapter2\partners\companyE\send\AS2\XML.
```

Similar directories exist for HTTP and for document formats such as EDI.

To send a message to Company E in this manner, simply copy an XML file in this location. Via this directory, WebSphere Partner Gateway Express can be integrated with any back-end application systems.

Archived

Archived

11

Managing the B2B exchange

This chapter introduces you to some of the tools that are a part of WebSphere Partner Gateway and that can help the administrator to manage the WebSphere Partner Gateway system. The main tools are:

- Event Viewer
- Document Viewer
- AS2 Viewer

In this chapter, we also learn how to use these tools to investigate a number of problems. We see how these tools provide critical information to find the root cause of a problem.

© Copyright IBM Corp. 2005. All rights reserved.

11.1 Tools available to manage the exchange

There are many tools and sources of information within WebSphere Partner Gateway to manage the system. These tools are available at a number of levels. Low-level sources of information, such as WebSphere Application Server log files, can be useful to investigate low-level problems. Other tools provide a chronological list of events that have occurred, or are happened at a given time, in the system. Higher-level tools look at documents as they pass through WebSphere Partner Gateway and events as they are related to a certain document. And there are tools in the system that are specific to certain protocols, notably AS2 and RosettaNet.

All these tools are runtime and real-time tools. They show what happens in the system during a certain time interval, including real-time information. Tools are also available that can perform historical analysis or that can be used prior to going live with a connection. In earlier chapters, we use the Test Connection feature to validate up to some level that the configuration for a certain participant connection is working. You can also use this Test Connection feature as a quick test when problems are reported to the system administrator.

A more extensive test feature exists for RosettaNet interactions. WebSphere Partner Gateway provides a Participant simulator that allows you to validate the setup specifically for RosettaNet.

To perform historical analysis or to perform some type of charge back to business units using the WebSphere Partner Gateway system, the system administrator can generate a document volume report. These reports can be exported from WebSphere Partner Gateway as a comma-separated file, which can be imported into a spread sheet for further analysis.

In addition to wanting to know how many documents have been processed, you may also want to investigate archival options for all completed transactions. A trace of transactions, completed and on-going, is kept in the database and on the file system. For example, if a document was received by WebSphere Partner Gateway, the system maintains a record in the database about that document, and the actual document (or payload) is stored on the file system. On systems with a reasonable volume of transactions, this can result in a huge amount of data. A procedure to archive this data is documented in the *Administrator Guide*, which can be downloaded from:

http://www.ibm.com/software/integration/wspartnergateway/library/infocenter/

11.2 System log files

Two types of log files exist for each component of the product. The first type of log files are those related and managed by WebSphere Application Server. For each component directory (bcgreceiver, bcgconsole, bcgdocmgr), there is a folder:

- \was\profiles\bcgreceiver\logs\bcgreceiver
- \was\profiles\bcgconsole\logs\bcgconsole
- \was\profiles\bcgdocmgr\logs\bcgdocmgr

This folder contains the files SystemErr.txt and SystemOut.txt. Most of the information in these files relates to WebSphere Application Server itself. But, when investigating system related issues, it is worthwhile to check the contents of these files.

In the same folder structure you can find a log file managed by WebSphere Partner Gateway. The file names are:

- bcg_receiver.log
- bcg_console.log
- bcg_router.log.

The contents of this file can be helpful to locate configuration errors or resource problems. By default, the level of detail of these files is limited to errors. But if required, you can increase the output by updating the log4j properties in the properties file for each component. The properties file is located in the install_dir\receiver\lib\config for receiver component, in install_dir\console\lib\config for the console component and in install_dir\router\lib\config for the router component.

11.3 Event Viewer

The Event Viewer is probably the first tool that the hub administrator uses to learn how the system is performing and whether any errors have occurred. It is also a viewer that has an automatic refresh rate built-in.

Many filtering options are available in the main window of the Event Viewer. You can filter for several levels of event information, from debug level up to critical. You can also filter on participants, time and date intervals, event location, and so forth.

Events are normally shown in a chronological order and are not necessarily linked to a document flow or exchange. The default search arguments are always

Archived

set to show you all events of the last ten minutes as shown in Figure 11-1 on page 310.

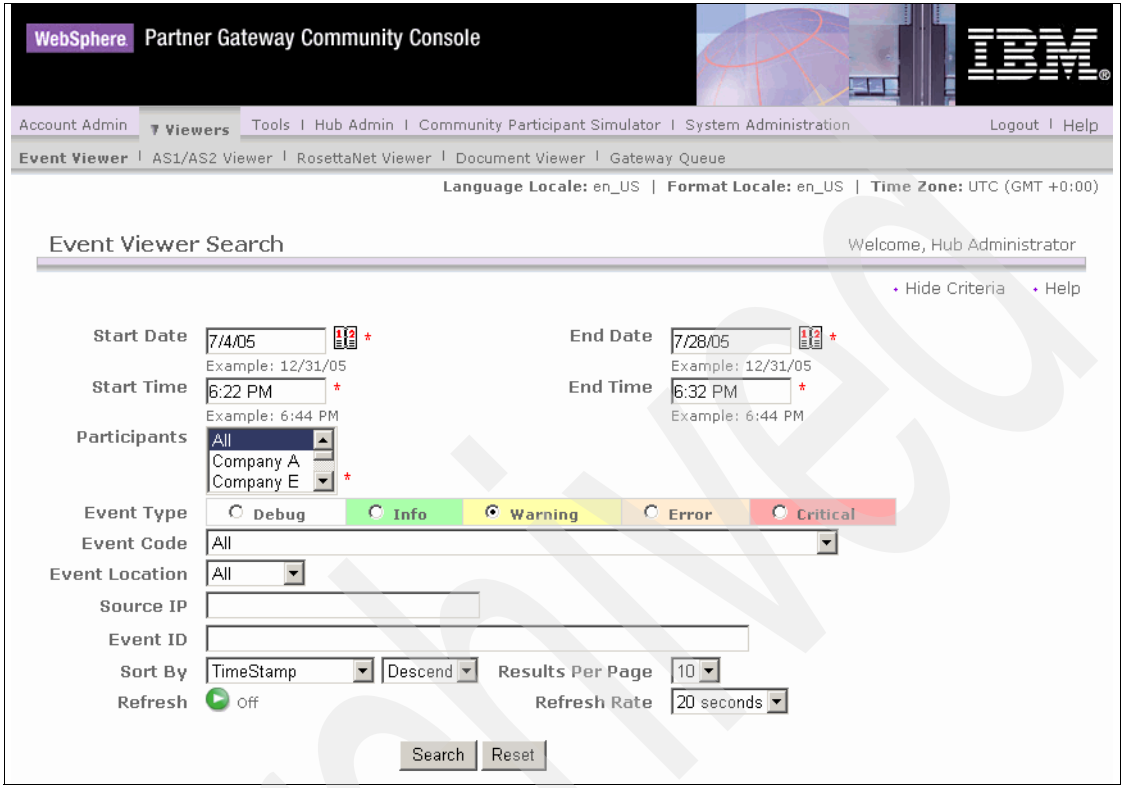


Figure 11-1 Event Viewer interface

When an event is found that needs further investigation, the Event Viewer can link to the Document Viewer. Figure 11-2 on page 311 shows a warning message indicating that a delivery has failed. A message was being sent from Company E to Company A, but it did not succeed.

Archived

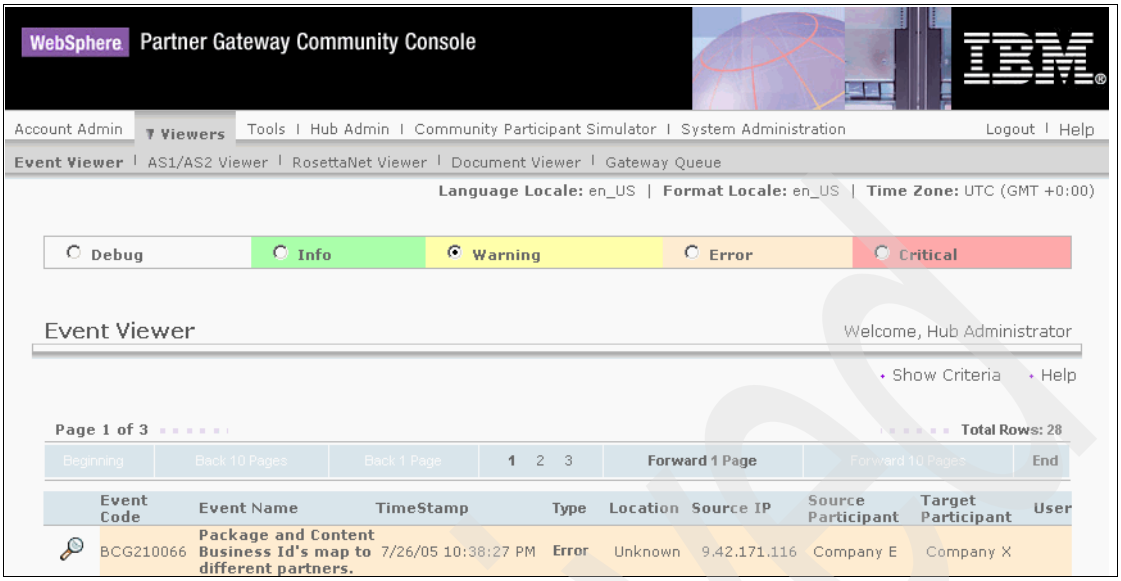


Figure 11-2 Example of an event which is of category Warning

Clicking the magnifying glass details icon for the event would result in Figure 11-3 and Figure 11-4 on page 312. You can now open the document that generated this event.

Figure 11-3 Event details and the associated document details (1 of 2)

Archived

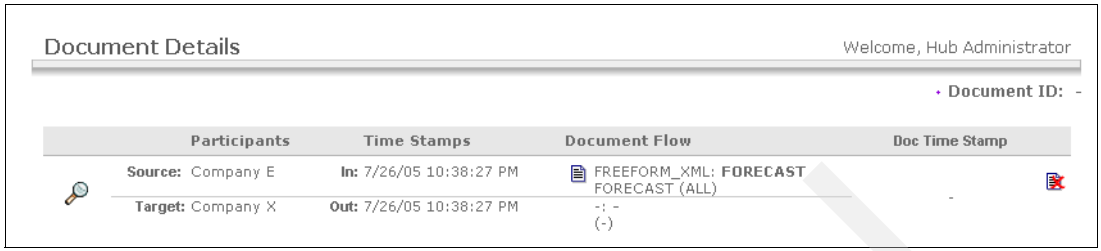


Figure 11-4 Event details and the associated document details (2 of 2)

In the Document Details section, in Figure 11-4, select the icon. This opens the window shown in Figure 11-5 and Figure 11-6 on page 313.

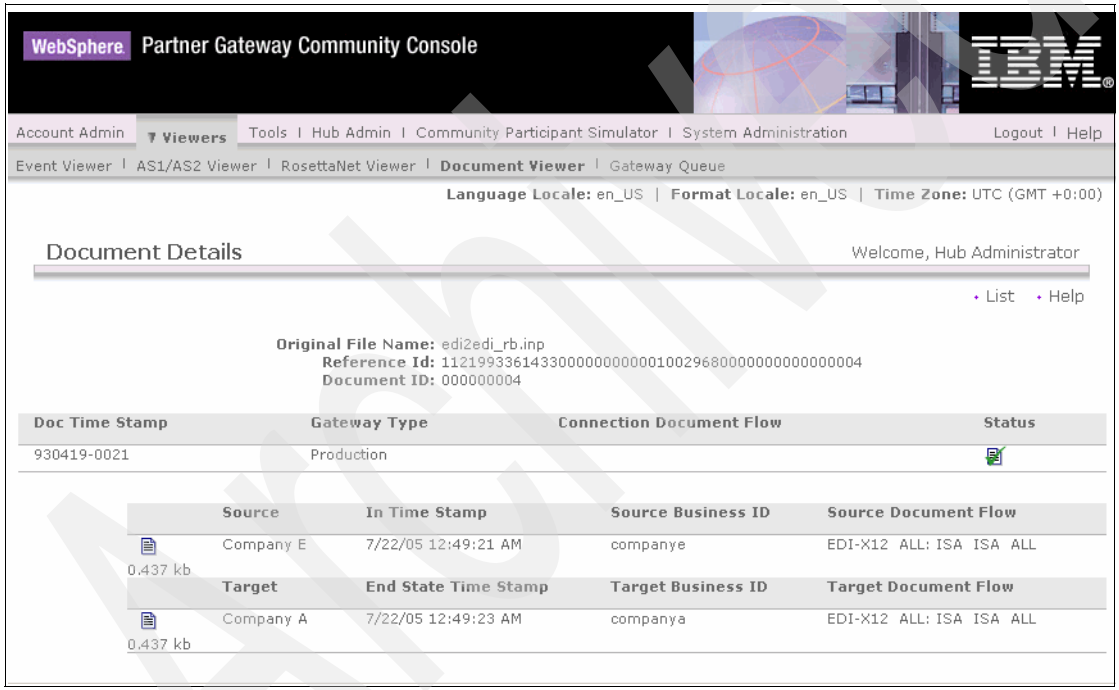


Figure 11-5 Details of a document with associated events (1 of 2)

Archived

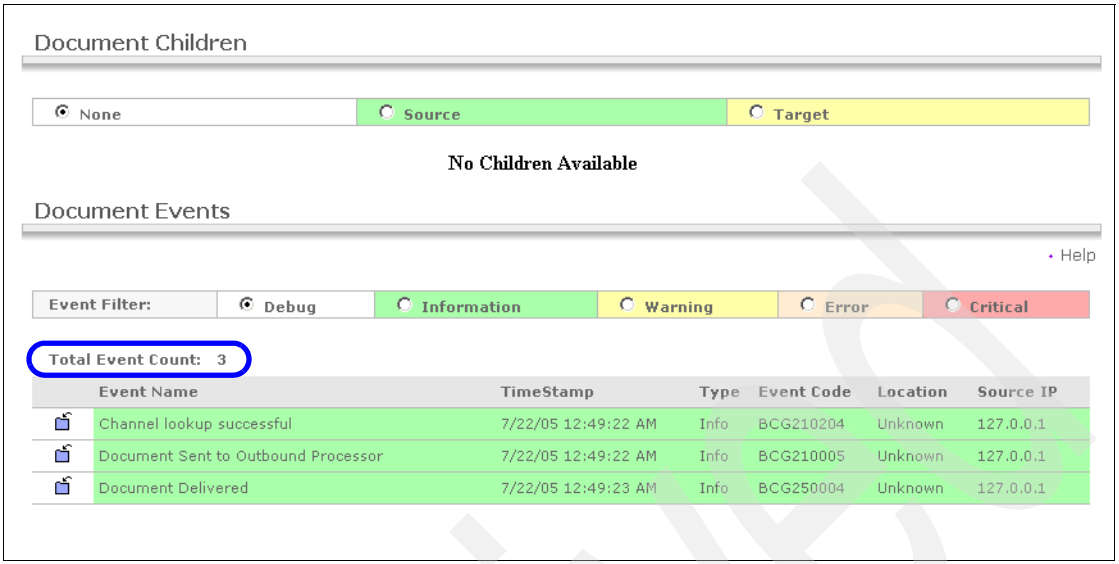


Figure 11-6 Details of a document with associated events (2 of 2)

Figure 11-3 on page 311 and Figure 11-4 on page 312 show the event and the associated document. Figure 11-5 on page 312 and Figure 11-6 show the document and its associated event count. Now it becomes apparent that there are other events linked to this same document. By changing the filter level to Information or Debug, you can display all the events.

11.4 Document Viewer

The Document Viewer is used to look at documents as they pass through the system. You can use it to look at documents that include specific partners, and many other filtering options, as shown in Figure 11-7 on page 314 and Figure 11-8 on page 314.

Archived

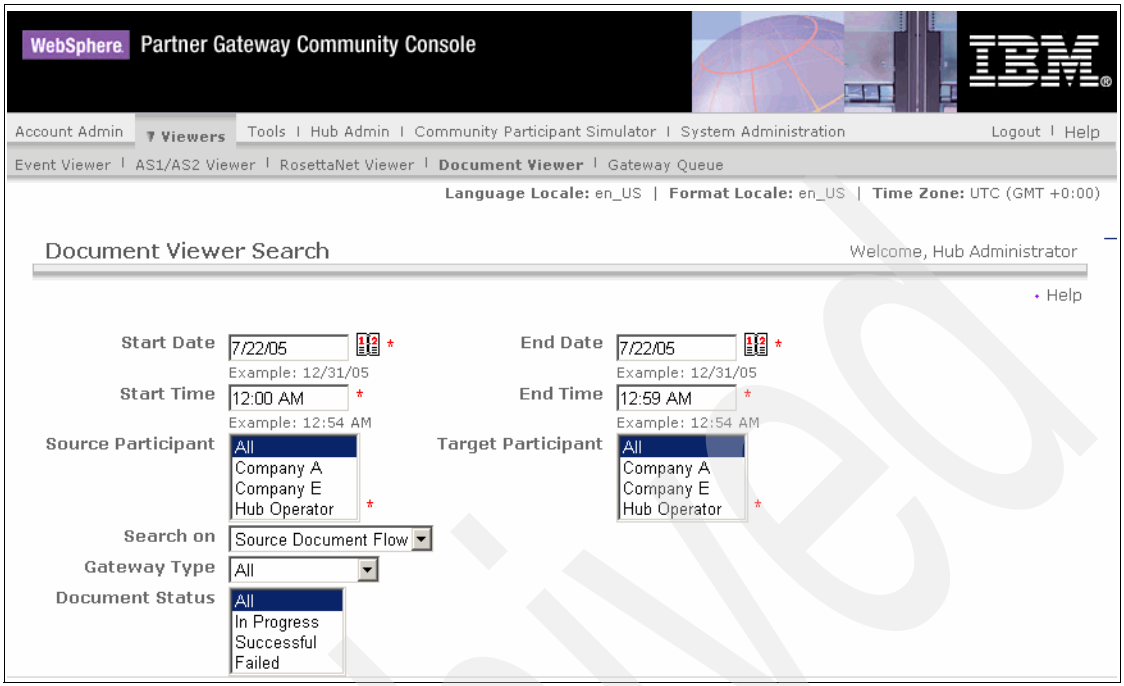


Figure 11-7 Search options for the Document Viewer (1 of 1)

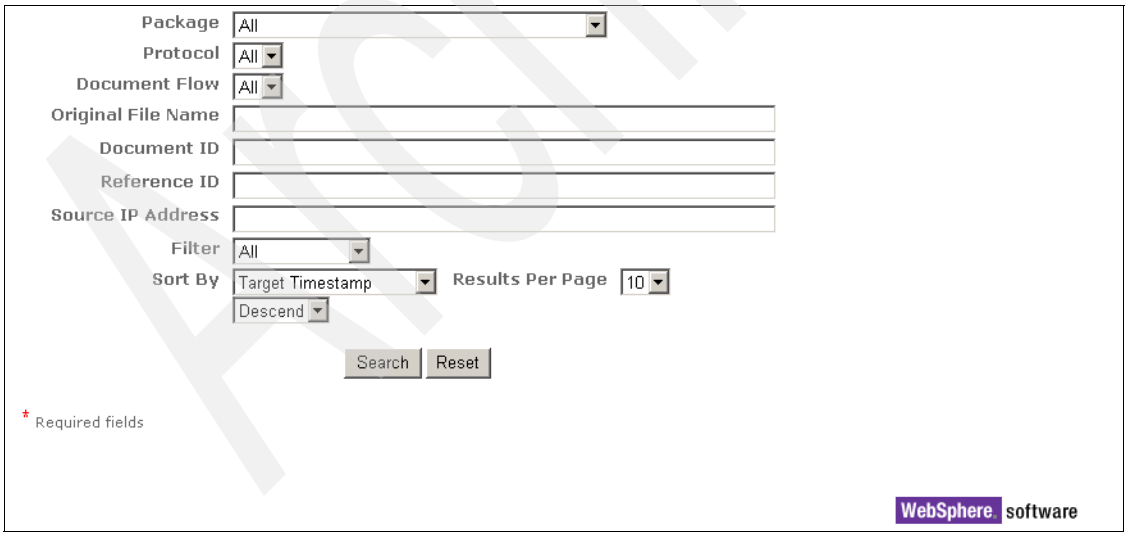


Figure 11-8 Search options for the Document Viewer (2 of 2)

Archived

Figure 11-9 shows the resulting list of documents for a particular time interval.

The first document in the list is sent from Company A to Company E and is unrelated to the second document shown in the list in Figure 11-9. If you click the icon, you see the details of the document and any events associated with that document.

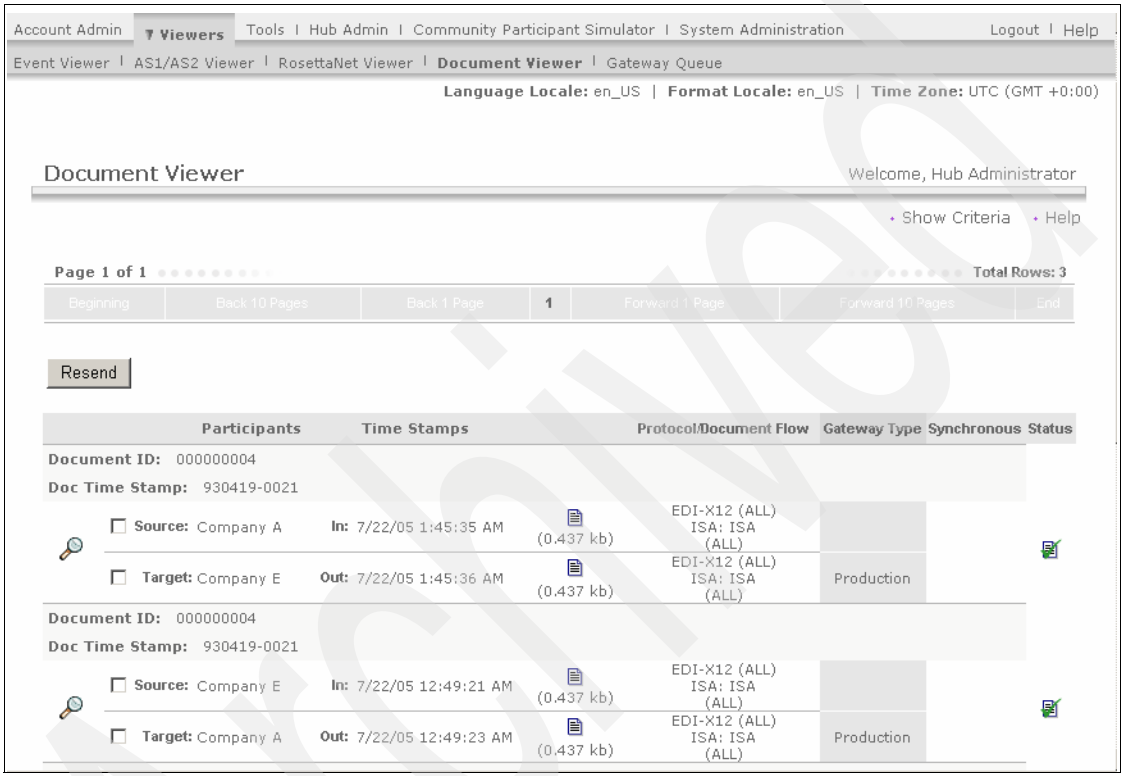


Figure 11-9 List of documents matching the search criteria

11.5 AS1/AS2 Viewer

Earlier in this redbook, we used AS1/AS2 Viewer when we described how to use it to validate the configuration and how to learn about the AS2 transactions. See 7.8, “Validating communication” on page 196.

The search options for the AS1/AS2 Viewer are more or less the same as the options for the Document Viewer. Any result is now specific to AS2 packaging. No other types of communication are shown in AS1/AS2 Viewer. A major differentiator is that now related documents are grouped together. Even more,

Archived

the initial list of results in the AS1/AS2 Viewer shows one item per AS1/AS2 transaction which includes the original EDI document and the MDN.

As shown in Figure 11-10, for each item in that list, there are the source and target partners, the message ID, and the MDN status. For all documents shown in Figure 11-10, WebSphere Partner Gateway received a corresponding MDN.

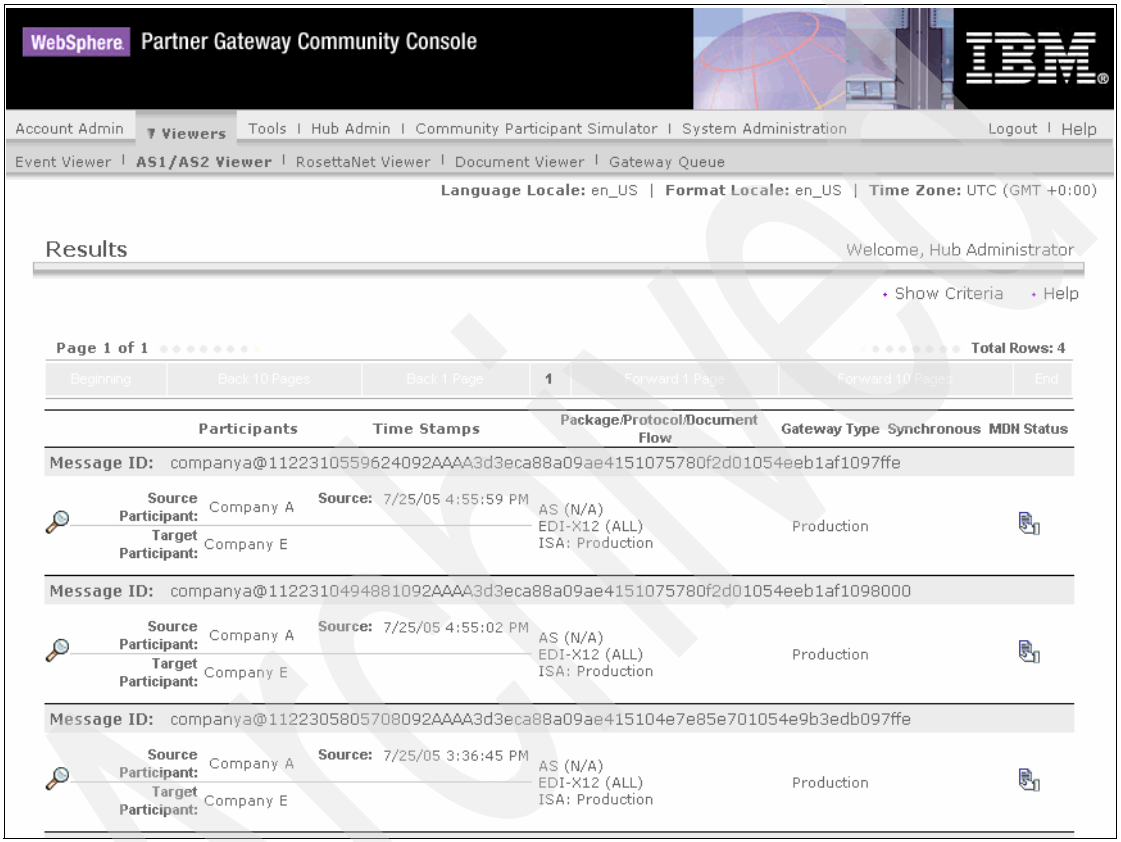


Figure 11-10 List of AS2 transactions

Click the icon next to a transaction and see all three documents that belong to this transaction. You can inspect the electronic data interchange (EDI) document that was picked up by WebSphere Partner Gateway in the edi_out directory. You can see the AS2 packaged document that was sent to the partner. This packaged document has the AS2 and HTTP header information.

Click the icon next to the document to see details about that document and packaging as shown in Figure 11-11 on page 317. In this window, you see information about the AS2 packaging and AS2 status, the EDI document, and the

events that relate to this transaction. When you change the event filter from Warning to Debug, you see all possible details, as shown in Figure 11-12.

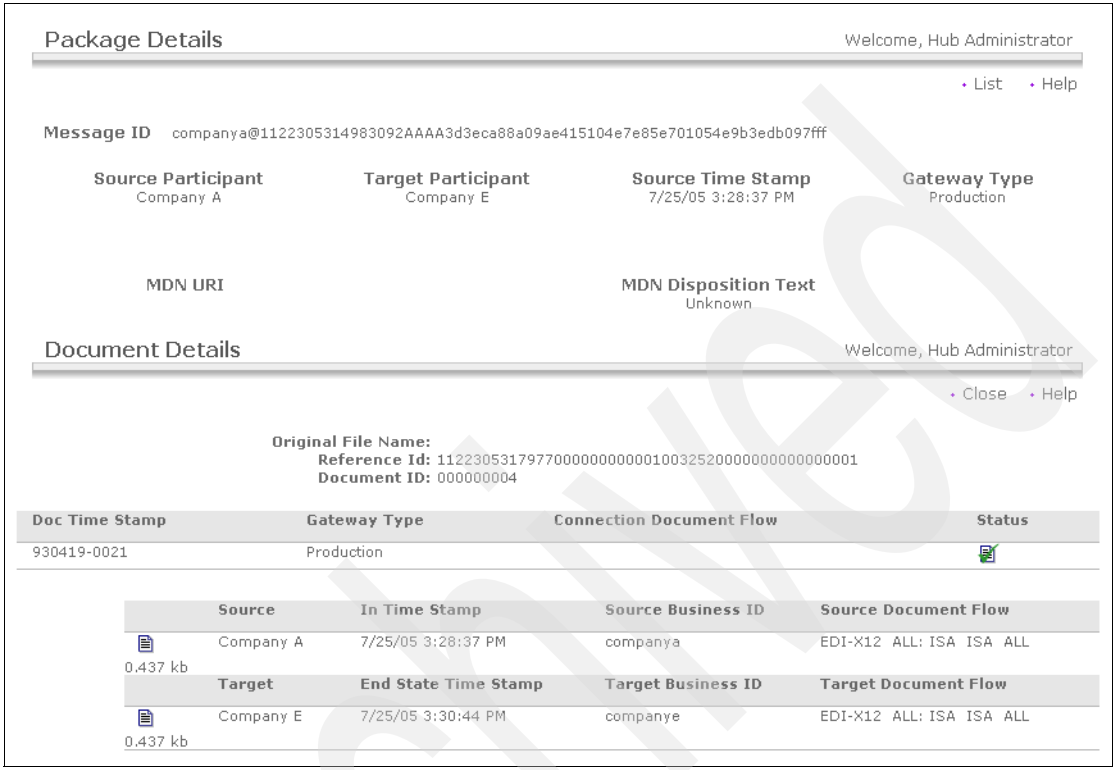


Figure 11-11 Event, document and packaging details for a single AS2 exchange (1 of 2)

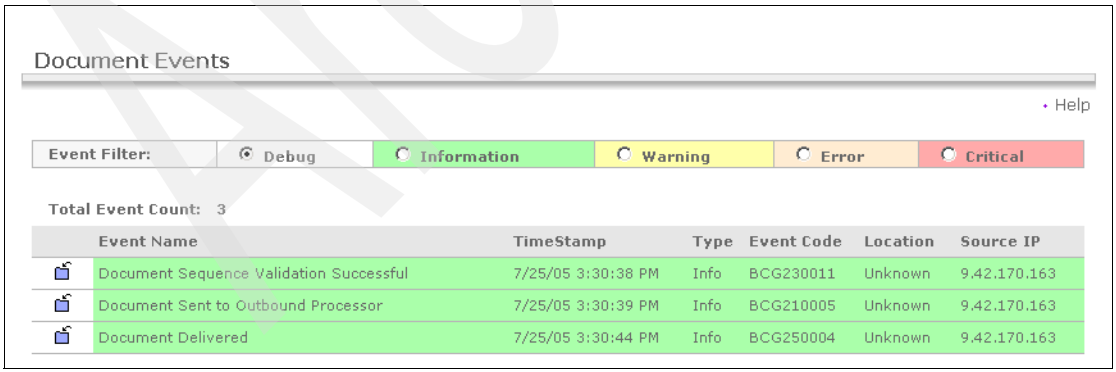


Figure 11-12 Event, document and packaging details for a single AS2 exchange (2 of 2)

Archived

Figure 11-10 on page 316 shows a list of completed AS2 transmissions. However, the AS2 Viewer can also help to determine which transactions are incomplete, because WebSphere Partner Gateway has not yet received an MDN for a given document. Figure 11-13 shows an example of such a document. The clock icon in the MDN status column indicates that MDN has not yet arrived for this document.

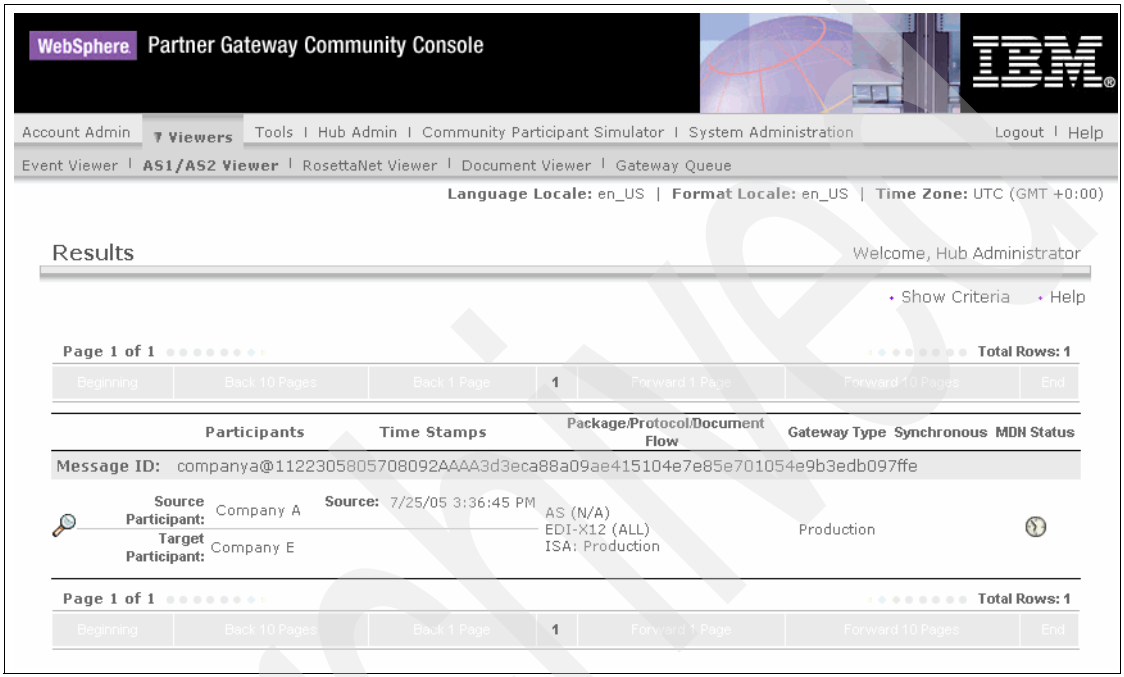


Figure 11-13 AS2 transmissions in progress Gateway Queue

11.6 Gateway queue

The next viewer utility allows you to inspect the delays or the queuing at a gateway. On a lightly-loaded system, you might not be able to find any queuing at a gateway, because the message can be processed before you reach this viewer.

The gateway queue viewer can be configured to refresh automatically. You can configure it also to only show documents with a delay that is longer than a certain number of minutes. This can be useful, for example, to see if service level agreements are being met.

Figure 11-14 shows that one single document is queued at the gateway called AS2Gateway, which is the HTTP gateway configured under the profile of Company X on Company E's server, to send documents that has AS2 packaging.

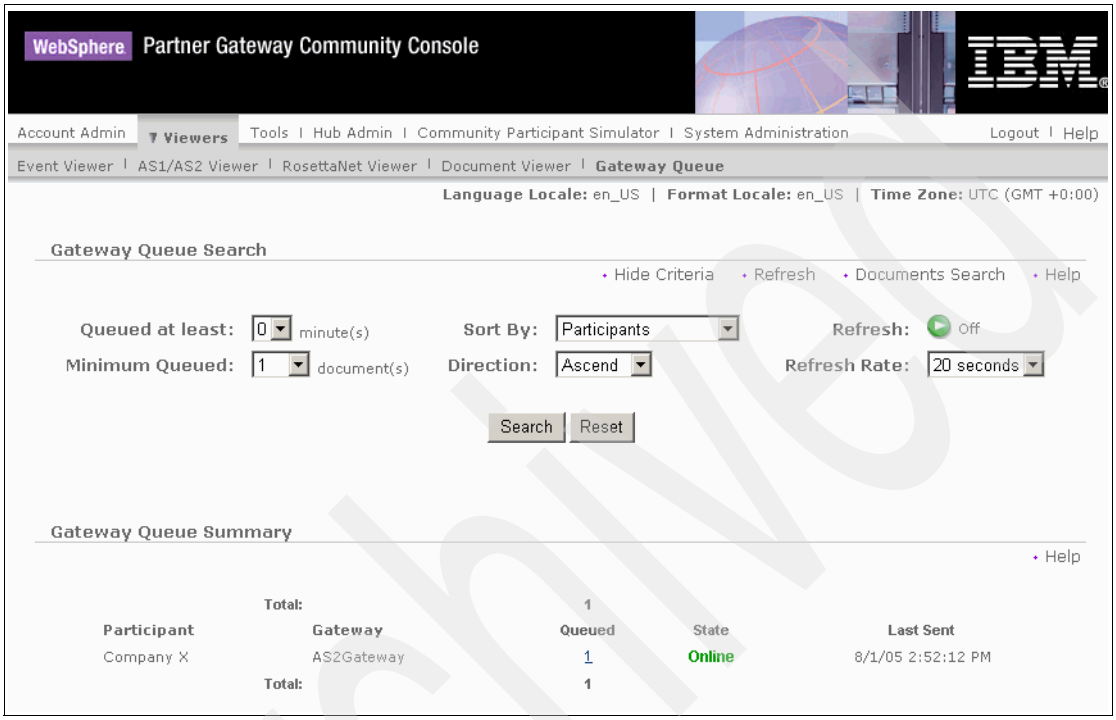


Figure 11-14 Gateway queue

When you click the number below the Queued label, you see additional configuration details about the gateway and information about the queued documents, as shown in Figure 11-15 on page 320.

Archived

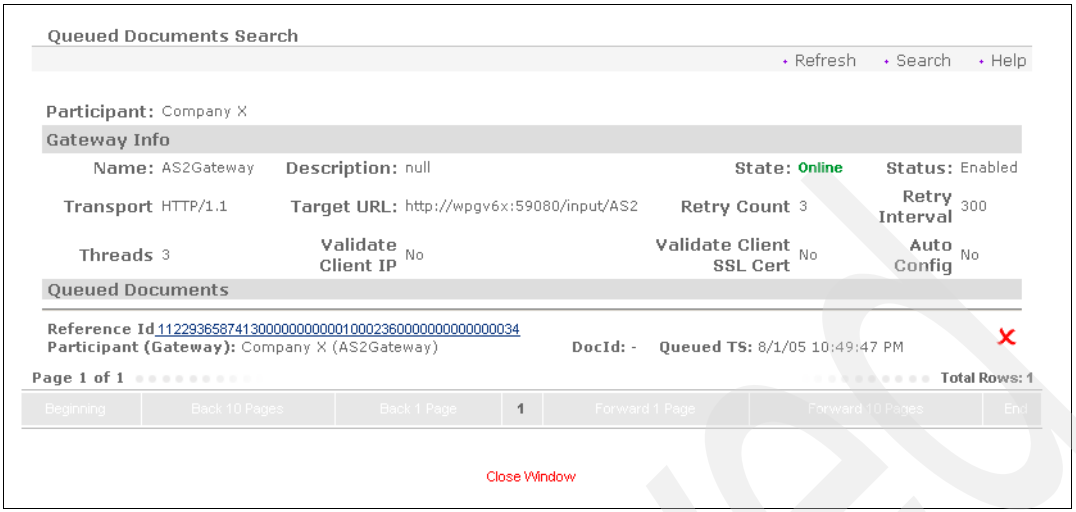


Figure 11-15 Details about the gateway and queued documents on that gateway

When you select the Reference ID in Figure 11-15, you see additional information about the ongoing processing and transfer. This window, shown in Figure 11-16 on page 321, gives you the option to stop the process. If you suspect that something is wrong with a given transmission and that, as a result of such a problem, other documents are queuing, you can click the **Stop Process** link in Figure 11-16 on page 321 to stop the processing of this document.

Archived

Gateway Details

• Help • Edit Gateway

Participant Info

Participant: Company X Participant Type: Community Participant Status: Enabled

Gateway Info

Name: AS2Gateway Description: null State: Online Status: Enabled

Transport HTTP/1.1 Target URL: http://wpgv6x:59080/input/AS2 Retry Count 3 Retry Interval 300

Threads 3 Validate Client IP No Validate Client SSL Cert No Auto Config No

Document Details Welcome, Hub Administrator

• Help

Original File Name: purchaseOrder.rod
Reference Id: 112293658305600000000000100500800000000000008
Document ID: -

Doc Time Stamp	Gateway Type	Connection Document Flow	Status
-	Production		(Stop Process)

	Source	In Time Stamp	Source Business ID	Source Document Flow
0.281 kb	Company E	8/1/05 10:49:43 PM	companye	CUSTOM_ROD ALL: PURCHASE_ORDER PURCHASE_ORDER ALL

	Target	End State Time Stamp	Target Business ID	Target Document Flow
0.281 kb	Company X		companyx	CUSTOM_XML ALL: PURCHASE_ORDER PURCHASE_ORDER ALL

Figure 11-16 Details about the gateway and queued documents on that gateway

11.7 Using the tools for problem determination

The previous sections demonstrate the use of a number of tools that a system administrator can use to find problems, look up documents in transit, or review the status of AS2 exchanges. This section describes the use of these tools to solve a few problems.

11.7.1 MDN HTTP URL not defined

An outbound AS2 document needs a URL for the MDN response. If an AS2 document should not have this, then the receiving partner does not know how to send an automated response. To avoid this situation and to make sure that the

packaged AS2 document adheres to the AS2 standard, WebSphere Partner Gateway verifies this MDN URL and other attributes of an AS2 connection.

Incorrect or incomplete settings such as that are considered critical errors. As such, the packaging within WebSphere Partner Gateway does not go far enough to have a document classified as an AS2 document. If you try to send a document over AS2 and nothing appears in the AS2 Viewer, try the Event Viewer, which is the first place where you can find information about such a problem. Figure 11-17 shows a critical event that refers to an AS Packager failure.

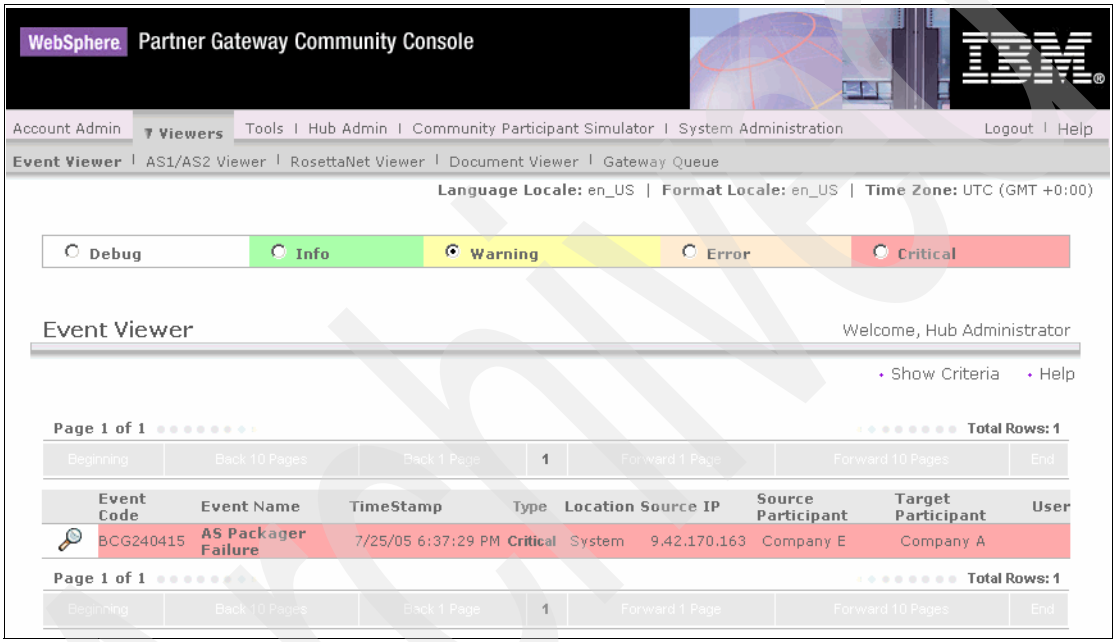


Figure 11-17 Critical event related to AS packaging

When opening the details of this event, you see a window similar to Figure 11-18 on page 323. The description field clearly explains what is wrong. The document details also provide information about the document flow (EDI-X12) and the participants. With this information, the hubadmin user or the administrator of Company E has sufficient information to look up the participant connection and to correct the value for the attribute AS MDN HTTP URL. Refer 7.6, “Connecting Company E to Company A” on page 180 and to Figure 7-43 on page 187.

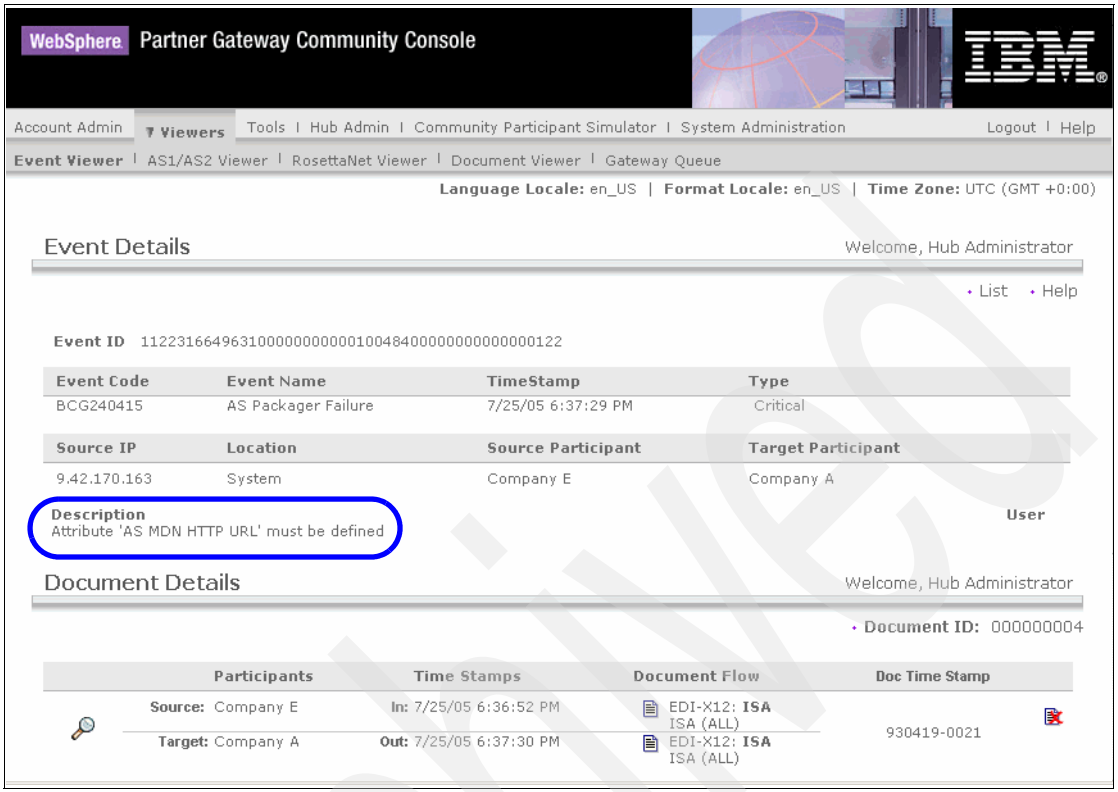


Figure 11-18 Details of the critical event

11.7.2 Problems with encryption

When implementing encryption in a WebSphere Partner Gateway environment, you might have to correct a few errors before you set it up correctly. Because encryption is performed at the AS2 packaging layer, you might find that the AS2 Viewer has no information about the document that you try to send. When encryption fails, it is considered a critical packaging error that appears only in the Event Viewer, as shown in Figure 11-19 on page 324.

Archived

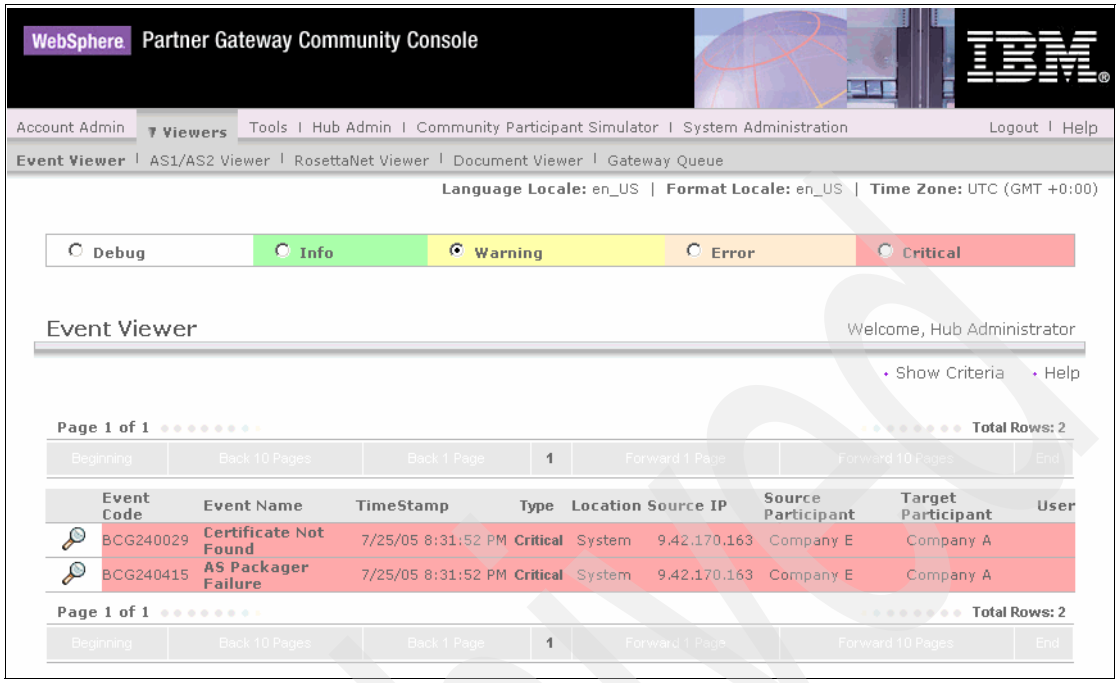


Figure 11-19 Critical events related to AS2 packaging

When opening up the details of this event, you see the description of the event, as shown in Figure 11-20 on page 325. The error text says that the certificate is null, which is caused by not having the right certificate in the right place. When Company E needs to send an encrypted document to Company A, the server looks for the public key of Company A in the profile of Company A. If you accidentally upload the public key in another profile, such as the hubadmin profile, you can see a similar error.

Archived

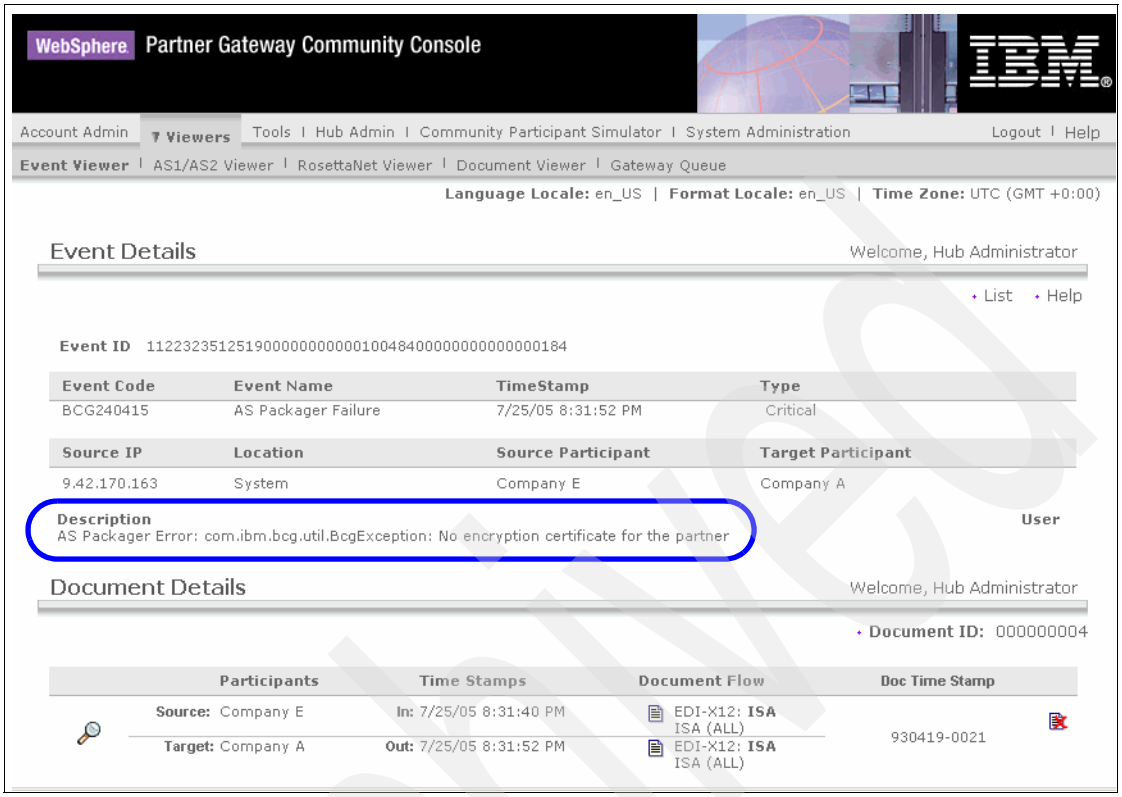


Figure 11-20 Details of the critical event

Besides the Event Viewer, this type of error is also reported in the Document Viewer. When searching for failed documents, this error is shown as well.

11.7.3 Problems with digital signatures

When implementing digital signatures, you might again experience a situation where documents are not processed the way you expect. Figure 11-21 on page 326 shows a warning message that at first seems not so critical. Company E receives an MDN from Company A that is not signed, while signed MDNs were requested.

Archived

Figure 11-21 Warning event about unsigned MDN

When you open the details of this event, shown in Figure 11-22 on page 327, you have access to the document that is linked to this event. It shows that the MDN was sent by Company A. Click the icon for that document to look at the MDN as it has been received by Company E.

Archived

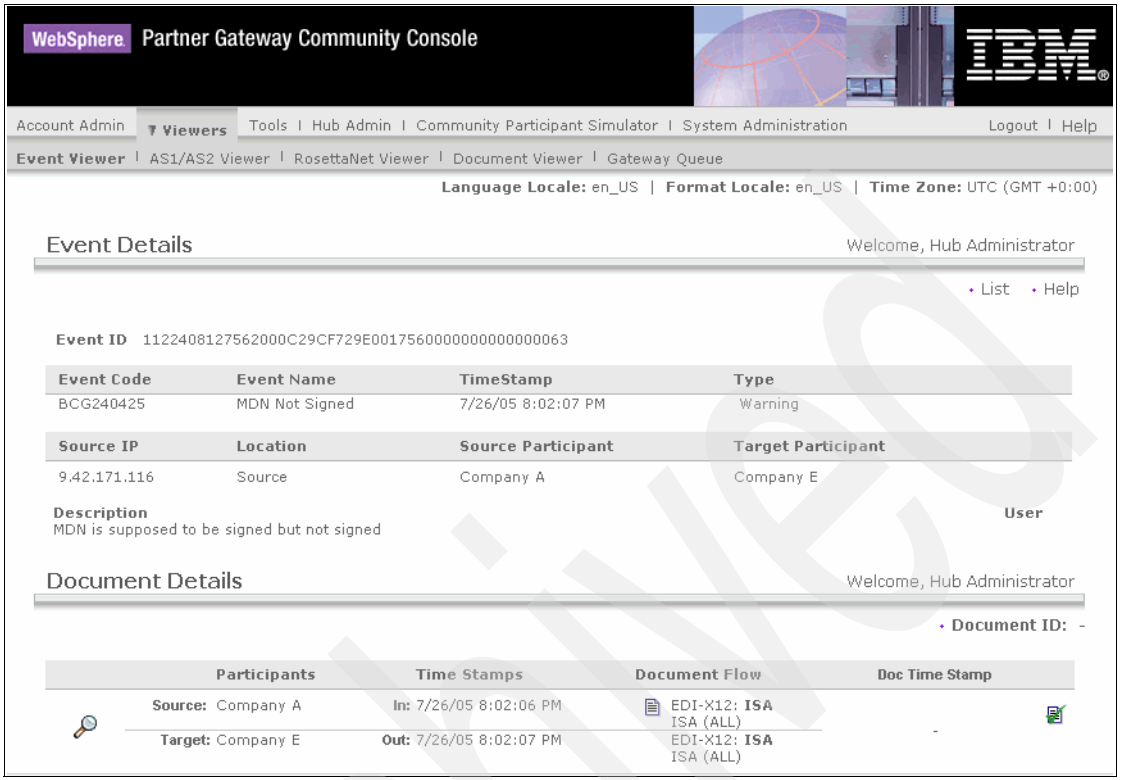


Figure 11-22 Details of the event and linked document

The actual MDN, shown in Figure 11-23 on page 328, contains a more interesting problem. The disposition message says that the authentication has failed. This can explain why Company A did not bother signing the MDN. Company A was unable to authenticate Company E as the sender of the document. What is this authentication to which the MDN is referring?

Archived

Raw Document Viewer Copy Help

Document ID: -

Doc Time Stamp: -

	Participants	Business IDs	Document Flow
Source:	Company A	companya	EDI-X12 (ALL) **ISA**: ISA (ALL)
Target:	Company E	companye	EDI-X12 (ALL) **ISA**: ISA (ALL)

UTF8

Transport Header

```
Content-Length: 728
Connection: close
Host: wpgv6hub:57080
AS2-Version: 1.1
Recipient-Address: http://wpgv6hub:57080/bcgreceiver/companye/edi_in
Subject: companye;companya
Date: Tue, 26 Jul 2005 19:56:45 UTC
Content-Type: multipart/report; Report-Type=disposition-notification;  boundary="----=_Part_8_587353444.1122407803925"
AS2-To: companye
AS2-From: companya
ReferenceId: 1122408126578000C29CF729E00152400000000000000019
Message-ID: <11224078039160002554F1C42028286273B05BECD9E1884@m106984f.itso.ral.ibm.com>
```

Initial Document

```
------=_Part_8_587353444.1122407803925
Content-Type: text/plain
Content-Transfer-Encoding: 7bit

This MDN response message is for:

Message id: <1122408097703000C29CF729E00175600000000000000056@WPGV6HUB>
From: companye

------=_Part_8_587353444.1122407803925
Content-Type: message/disposition-notification
Content-Transfer-Encoding: 7bit

Reporting-UA: WBI_Connect
Original-Recipient: rfc822; companya
Final-Recipient: rfc822; companya
Original-Message-ID: <1122408097703000C29CF729E00175600000000000000056@WPGV6HUB>
Disposition: automatic-action/MDN-sent-automatically; processed/Error: authentication-failed
Received-Content-MIC: JlNBbq2wW8hK96oCaxAh1ZY0EqI=, sha1

------=_Part_8_587353444.1122407803925--
```

Close Window

Figure 11-23 Details of the MDN that was unsigned

Archived

Now look at this same problem from an AS2 perspective. When using the AS2 Viewer, it lists this transaction with a status of MDN received, as shown in Figure 11-24. At first, there does not seem to be anything wrong with it.

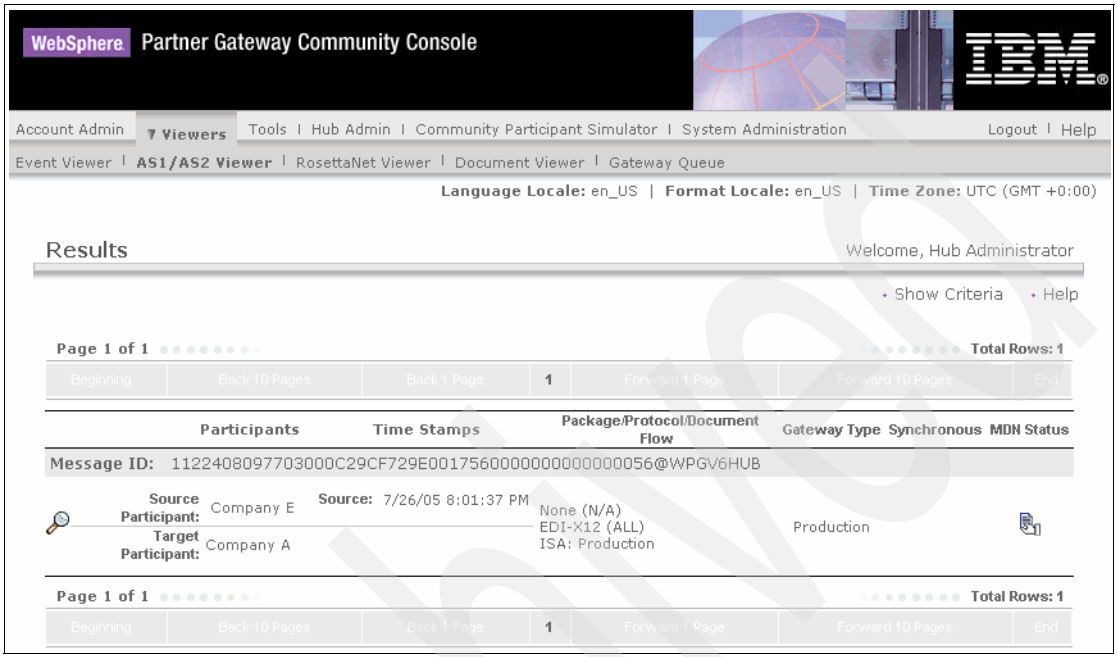


Figure 11-24 Overview of the AS2 exchange that caused the warning message

Archived

When you open the details for this specific AS2 exchange, you have the sent document and the received MDN in one browser window. This time, you see the disposition text in the MDN, which we saw earlier. The fact that the MDN has been received is not necessarily an indication of a successful transmission. There are positive MDNs and negative MDNs. Thus, the contents of the MDN must be inspected as well (Figure 11-25).

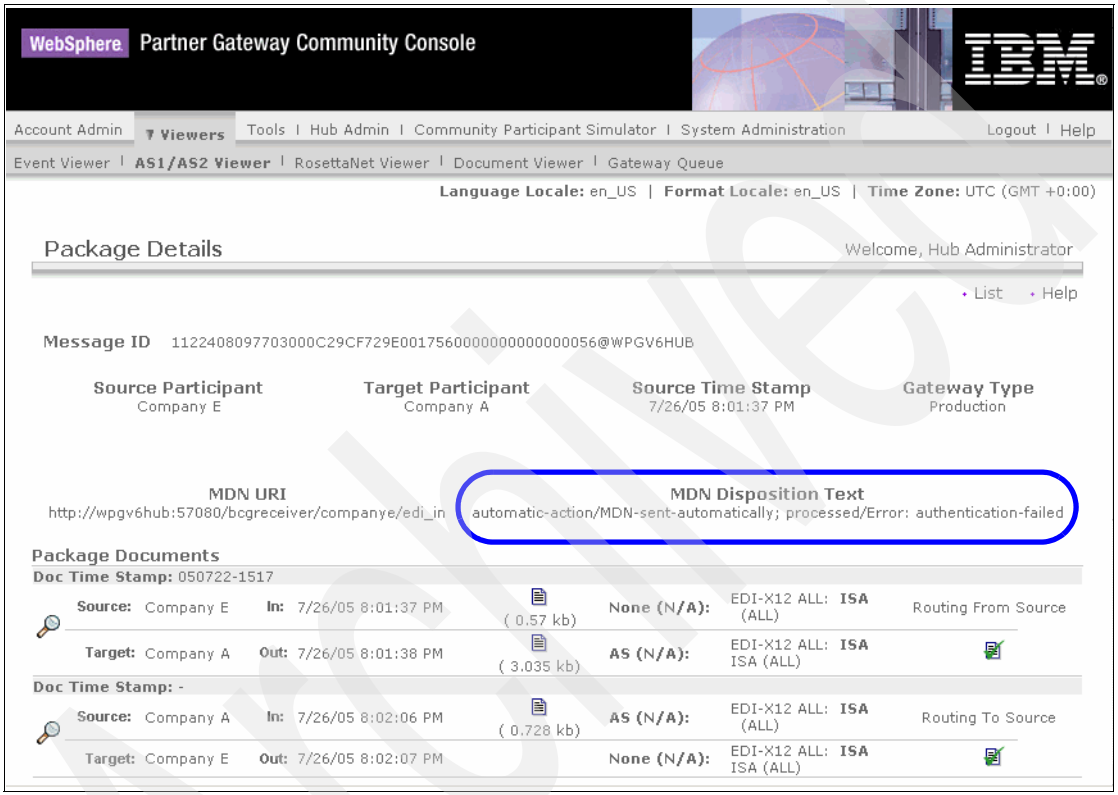


Figure 11-25 Details of the AS2 exchange as reported by the sender Company E

At this moment, we still do not know why authentication has failed. Also, if you use the Document Viewer of Company E (the sender), you do not see this document listed as a failed document. Because an MDN was received, WebSphere Partner Gateway assumes that the processing was successful.

We can look closer at the server of Company A and see how this problem might have been handled by the receiver of the document that failed authentication. Using the Event Viewer at Company A, an error event is being logged. When opening the details of that event, shown in Figure 11-25,the kind of authentication that has failed becomes clear. The signature that was included in the original AS2 document could not be verified (see Figure 11-26 on page 331).

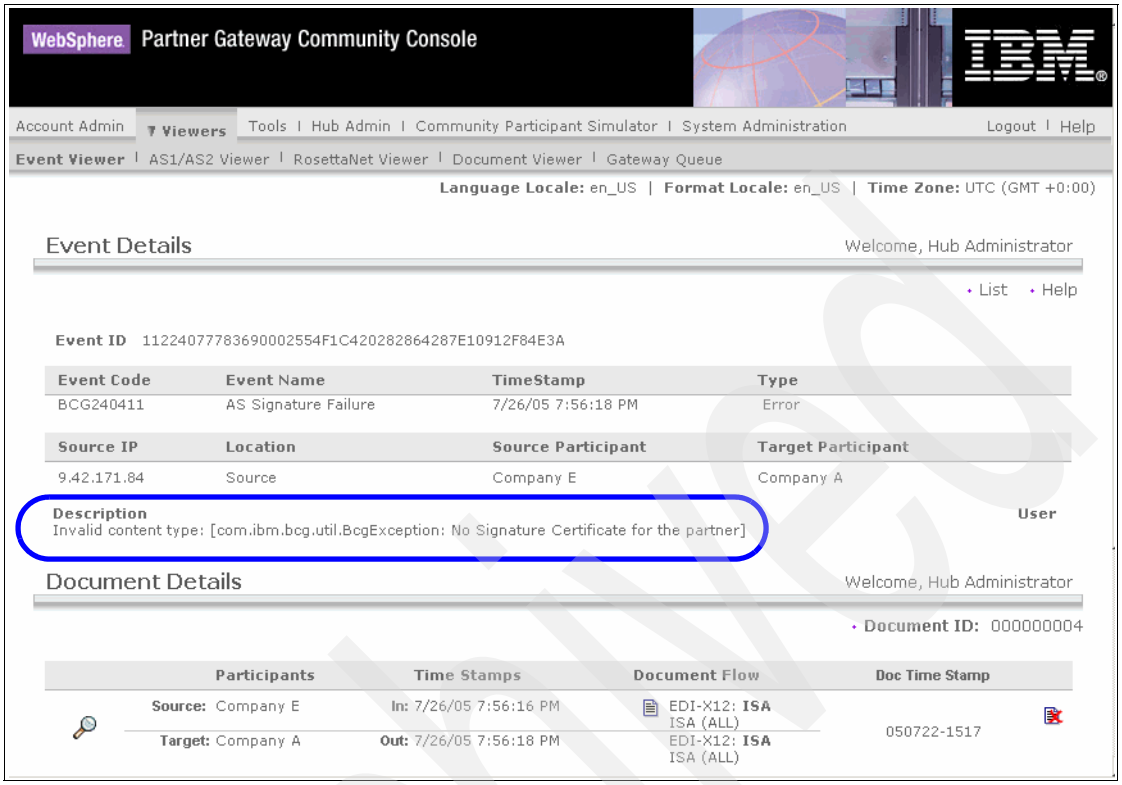


Figure 11-26 Details of the event as reported by the receiver Company A

When we use the AS2 Viewer on the server of Company A, we see that the AS2 transaction is listed as a transaction in error. The unencrypted EDI document is not available, only the encrypted document is as it was received at the HTTP protocol level. The MDN that was sent back to Company E is listed as well in Figure 11-27 on page 332.

Archived

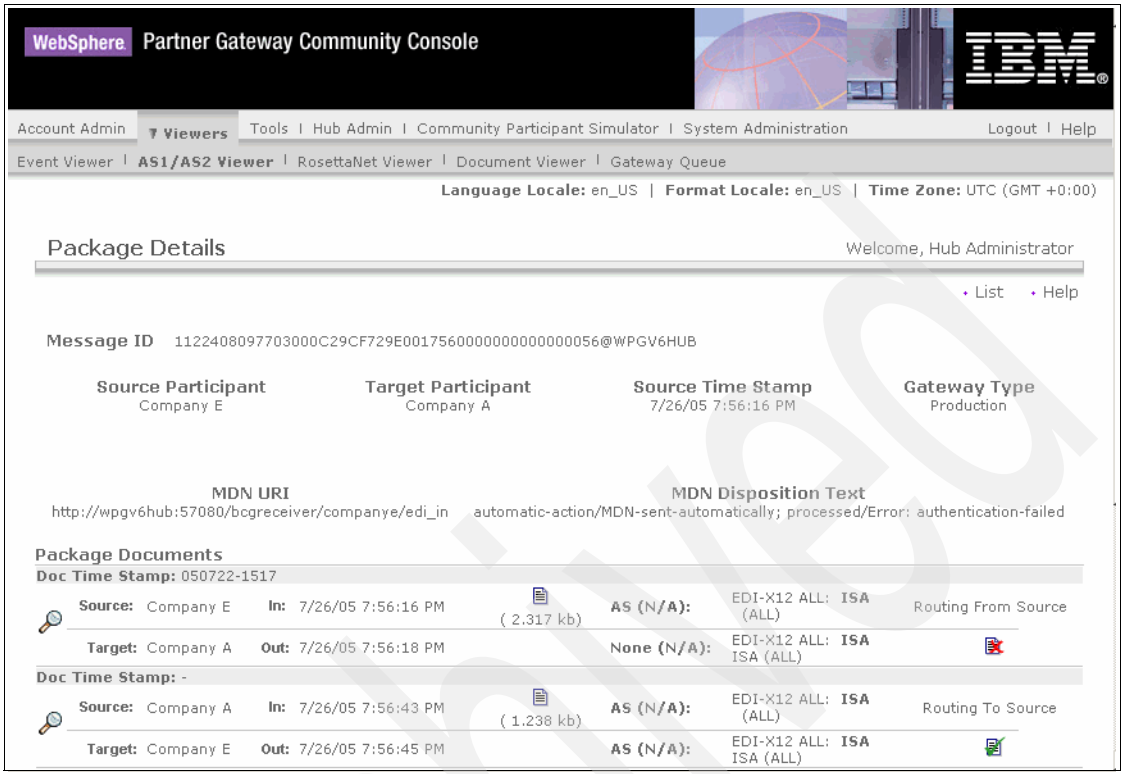


Figure 11-27 Details of the AS2 exchange as reported by the receiver Company A

If we open the Document Viewer on Company A, then we find that this specific document is categorized as a rejected document. So, what went wrong in this case? What is required to validate a digital signature? Two things are needed:

- The public key of the sender, Company E, needs to be available in their profile on the server of Company A. It must also be marked as used for digital signatures.
- The public key of the sender, Company E, needs to be uploaded to the profile of the user hubadmin and marked as used for CA Root.

Because this last step was not performed, the error condition discussed here was the result. This last step is required because we used self-signed certificates. For certificates issued by a commercial Certificate Authority, we would have uploaded the public key of the CA itself.

Archived

Part 3

FTP support

Archived

© Copyright IBM Corp. 2005. All rights reserved.

Archived

12

Integrating FTP servers

This chapter discusses an extension of the B2B solution for Company A. Company F and Company A want to establish communication. However, Company F is not ready to implement AS2. preferring an FTP-based solution.

WebSphere Partner Gateway provides a few methods for sending and receiving documents over FTP. We discuss the options available and describe the implementation of these methods in the context of Company A and Company F.

© Copyright IBM Corp. 2005. All rights reserved.

12.1 Overview of FTP and FTP Scripting scenarios

In a typical B2B environment, quite often not all business partners are aligned along a single communication mechanism. While AS2-based solutions are common and fairly easy to implement, there are always situations when a trading partner cannot implement a suitable solution in a certain time frame. Fortunately, WebSphere Partner Gateway can handle these situations.

This chapter considers a situation where Company A wants to extend its B2B infrastructure to include a document exchange with Company F. However, Company F is reluctant to jump so quickly to using a technology that is new to its IT department. At the same time, Company A wants to build a solution quickly, so they find a compromise by implementing an FTP-based solution. Company F already has extensive FTP experience, therefore they can accommodate Company A's wishes quickly.

WebSphere Partner Gateway Version 6 has added some additional FTP functionality with FTP Scripting. This is in addition to the FTP functionality that remains from earlier releases. In this chapter, we implement two methods of connecting Company A to Company F: FTP and FTP Scripting.

In a real-world configuration you would only need to use one of the two methods. We cover them both here only as a means of comparison. We also cover the advantages of one method over the other.

Figure 12-1 on page 337 shows a schematic overview of the B2B solution we have so far, with the addition of the FTP and FTP Scripting solutions we plan to implement in this chapter. It shows the existing AS2-based exchanges between Company E and Company A and the XML solution between Company E and Company X. Now Company A wants to have its own communication with Company F.

Archived

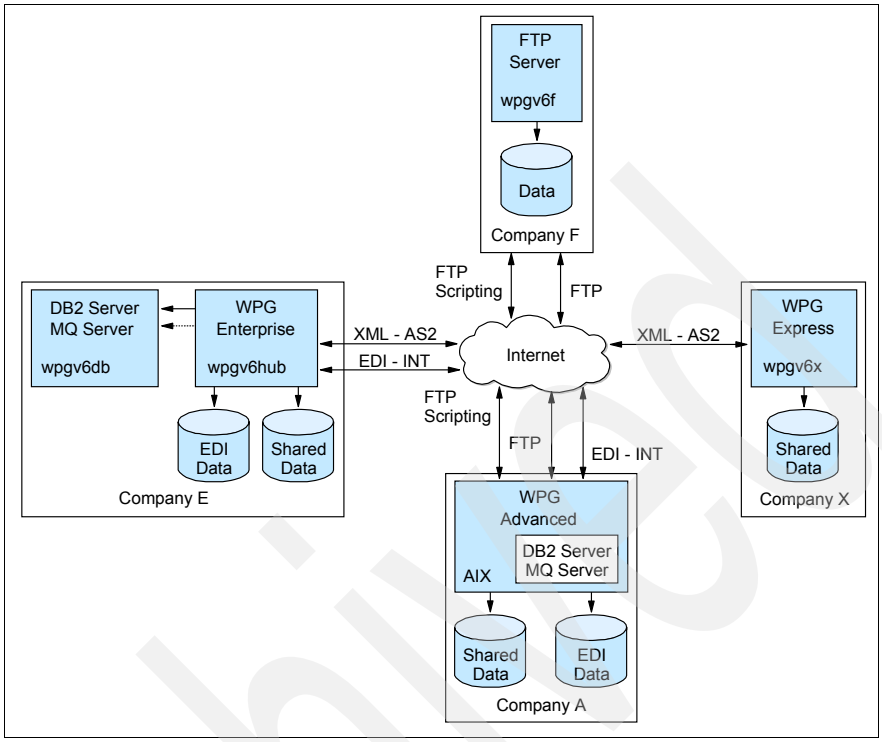


Figure 12-1 Scenario overview

12.1.1 FTP method one: FTP servers

The first method of communicating with FTP involves having FTP client and FTP server support at both companies. For documents outbound from Company A, this means the FTP client in WebSphere Partner Gateway contacts an FTP server at Company F's site and delivers a document. For inbound documents to Company A, it means the Company F FTP client contacts an FTP server at the Company A site to deliver the document.

The WebSphere Partner Gateway acts as an FTP client, but it does not have FTP server functionality. The FTP servers needed in this method can be almost any third-party FTP solution you choose. Integrating the FTP server with the FTP client functionality of WebSphere Partner Gateway is relatively simple and is consistent regardless of the FTP server used.

Implementing this first method is discussed in:

- 12.3, "Configuration of Company A for FTP Inbound" on page 360
- 12.4, "Implementing FTPS" on page 371

Archived

12.1.2 FTP method two: FTP Scripting

The second method of connecting Company A to Company F using FTP is with the new FTP Scripting functionality in WebSphere Partner Gateway Version 6.0. In this method, only a single FTP server located at the Company F site is needed for both outbound and inbound documents. This is one of the advantages of FTP Scripting.

With either method, FTP communication is established between Company A and Company F and integrated within WebSphere Partner Gateway. This means the documents exchanged can be viewed using the tools we discussed in Chapter 11, “Managing the B2B exchange” on page 307. Also, when Company F is ready to upgrade its solution to a full B2B hub, Company A will not have to make big changes to the routing logic in its back-end applications or to its B2B server.

Implementing FTP Scripting is discussed in:

- 12.5, “Configuration of FTP Scripting outbound” on page 384
- 12.6, “Configuration of FTP Scripting inbound” on page 403
- 12.7, “FTPS for FTP Scripting Gateways and Targets” on page 413

12.2 Configuration of Company A for FTP outbound

This section discusses the outbound flow in 12.1.1, “FTP method one: FTP servers” on page 337. That is, the documents sent from Company A to the FTP server of Company F. We will learn how the FTP client functionality is built into WebSphere Partner Gateway and the steps needed to implement this scenario.

This section, along with 12.3, “Configuration of Company A for FTP Inbound” on page 360 and 12.4, “Implementing FTPS” on page 371 discuss the *first method* in 12.1.1, “FTP method one: FTP servers” on page 337.

12.5, “Configuration of FTP Scripting outbound” on page 384, 12.6, “Configuration of FTP Scripting inbound” on page 403 and 12.7, “FTPS for FTP Scripting Gateways and Targets” on page 413 discuss the *second method* in 12.1.2, “FTP method two: FTP Scripting” on page 338.

You only need *one* of these implementations in a real-world environment, so reading through this entire chapter before following the examples, can be beneficial to determine which solution best fits your environment.

12.2.1 Outbound implementation steps

We can send any type of file that WebSphere Partner Gateway supports outbound from the system through an FTP-type gateway. However, because our example assumes Company F did not want to jump to any new technology, we are therefore assuming also that Company F are not ready to parse AS2 packaging or use the EDI protocol. For this implementation, we assume binary-type documents, which is another document type that the server supports.

When sending flat, binary files, WebSphere Partner Gateway does not inspect the document and retrieve the sending and receiving business identifiers from the document itself. We can setup a scenario in which the server can scan the document and read information from a Record Oriented Data (ROD) document, but it involves using mapping and transformation of ROD documents. That topic is addressed in Chapter 15, “Mapping” on page 473. For now, we are concerned about transferring plain, binary data from one participant to another.

Without opening the binary document, WebSphere Partner Gateway will need a way to determine the sending and receiving participants of the document. The sending participant is determined though the use of an FTP Directory type target. This target has a fixed directory structure and based on the directory the document is received from, it can determine the sending participant. The receiving business identifier is determined from the filename itself.

For this outbound example, Company A is sending a file to the FTP server of Company F.

First we need the sending business identifier. The back-end application of Company A (where the file originates) needs to put the file in a specific directory that has the login name of the participant, companyA, included. The fact that file is binary, and the type of transmission (production or test), is determined from the directory as well. In line with our scenario, the full directory structure would be:

```
/home /bcguser/wpgv6/data/ftp/companyA/binary/Production
```

The root directory that the FTP Directory target scans is:

```
/home/bcguser/wpgv6/data/ftp
```

The receiving business identifier is determined from the file name. Assuming that that company F has a business identifier of companyfm a sample file name would be:

```
companyf.123456789.bin
```

WebSphere Partner Gateway can now determine the sending and receiving business identifiers of the document. The server will then need an interaction defining our scenario. This case would be:

```
None (unpackaged) binary document to unpackaged binary
```

Both participants need the B2B capabilities updated to support this interaction. Finally the participant connection between Company A and Company F is configured, with Company F giving the destination of its FTP server as its gateway.

A summary for the configuration described above is shown in Figure 12-2 on page 341 and is listed here:

1. Create a folder that is used by the FTP Directory target on the Company A server:

   ```
   /home/bcguser/wpgv6/data/ftp
   ```

2. Create a fixed directory structure underneath that folder that starts with companyA.
3. Create the target FTP target of transport type FTP Directory that maps to the directory created in Step 1.
4. Create a participants profile for Company F, which has a business identifier of companyf,
5. Update the B2B capabilities of Company F to include **None/Binary/Binary** for source and target
6. Update the B2B capabilities of Company A to include **None/Binary/Binary** for source and target
7. Create an interaction between None/Binary/Binary and None/Binary/Binary with action as Pass-through.
8. Create a gateway within the Company F profile that points to the FTP server of Company F.
9. Create a participant connection that brings together the interaction between that partners and the destination gateway.

Archived

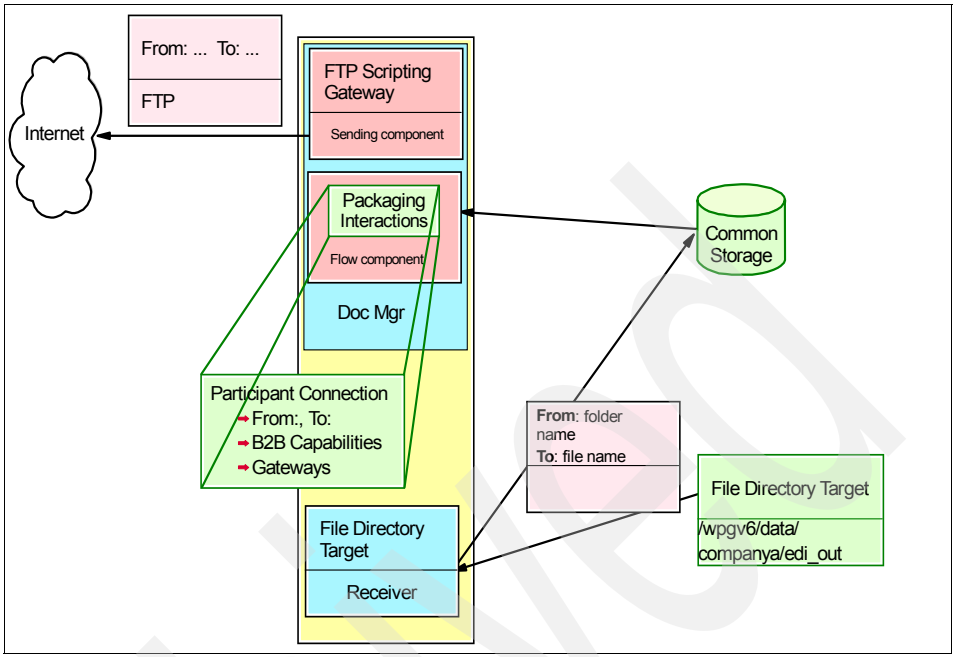


Figure 12-2 Outbound flow

12.2.2 Creating the directory structure

Company A is running the WebSphere Partner Gateway Advanced on an AIX installation. The steps to create the directory structure are as follows:

1. Create the directory that acts as the root directory for all outbound FTP communications using the AIX command line.

   In our setup we will create the directory `ftp` within the existing directory of `/home/bcguser/wpgv6/data` (see Example 12-1).

   You can create another directory if you choose, but it is important for the user `bcguser` to have read and write access to this directory and all its children.

Example 12-1 FTP Directory Target root directory creation

```
# su - bcguser
$
$ cd /home/bcguser/wpgv6/data
$
$ mkdir ftp
```

2. Within the folder `ftp`, create a new folder called `companyA`. This is the Company Login Name of Company A that was entered for the Community

Manager in 7.7, “Configuration tasks for the Company A hubadmin” on page 188. The directory name is case-sensitive. See Example 12-2.

Example 12-2 Create the companyA directory structure under ftp

```
$ pwd
/home/bcguser/wpgv6/data/ftp
$
$ mkdir companyA
```

3. Enter the companyA directory and create two directories, `binary` and `documents`. See Example 12-3.

Example 12-3 creating binary and documents directories

```
$ pwd
/home/bcguser/wpgv6/data/ftp/companyA
$
$ mkdir binary
$
$ mkdir documents
$
$ ls -l
total 2
drwxr-xr-x   2 bcguser  bcggroup        512 Jul 25 16:30 binary
drwxr-xr-x   2 bcguser  bcggroup        512 Jul 25 16:30 documents
```

4. In each of these two new folders, create two subdirectories: `Production` and `Test`. See Example 12-4.

Example 12-4 Creating Production and Test subdirectories

```
$ pwd
/home/bcguser/wpgv6/data/ftp/companyA
$
$ cd binary
$
$ mkdir Production
$ mkdir Test
$
$ cd ..
$ cd documents
$
$ mkdir Production
$ mkdir Test
```

A recursive list shown in Example 12-5 on page 343, lists the directory structure we now have, starting at the root directory that the FTP Directory target.

Example 12-5 Recursive List of directory structure.

```
$ pwd
/home/bcguser/wpgv6/data/ftp
$ ls -R
companyA
./companyA:
binary     documents

./companyA/binary:
Production  Test

./companyA/binary/Production:

./companyA/binary/Test:

./companyA/documents:
Production  Test

./companyA/documents/Production:

./companyA/documents/Test:
```

12.2.3 Updating the hubadmin profile

As the hubadmin user on Company A, we need to perform the following tasks:

1. Create a Participant profile for Company F.
2. Create the FTP Directory type target to point at the directory structure created in the last section.
3. Create an interaction for unpackaged binary (None/Binary/Binary) to unpackaged binary (None/Binary/Binary).

Creating the new community participant

To create the new partner profile for Company F, complete these steps:

1. Log on as the hubadmin to the WebSphere Partner Gateway Advanced server of Company A.
2. Select **Account Admin** → **Profiles** → **Community Participant**.
3. Click **Create**.
4. In the New Participant window (Figure 12-3 on page 344), fill in the following
   a. For Company Login Name, enter `companyF`.

Archived

b. For Participant Display Name, enter `Company F`. This is consistent with the naming convention used throughout the examples with Company E.

c. For Participant Type, select **Community Participant**.

d. Under Business ID, click **New**, and then select **Freeform**. For the identifier use `companyf`.

e. Under IP Address or Host Name, click **New**, add a hostname.

f. Click **Save**.

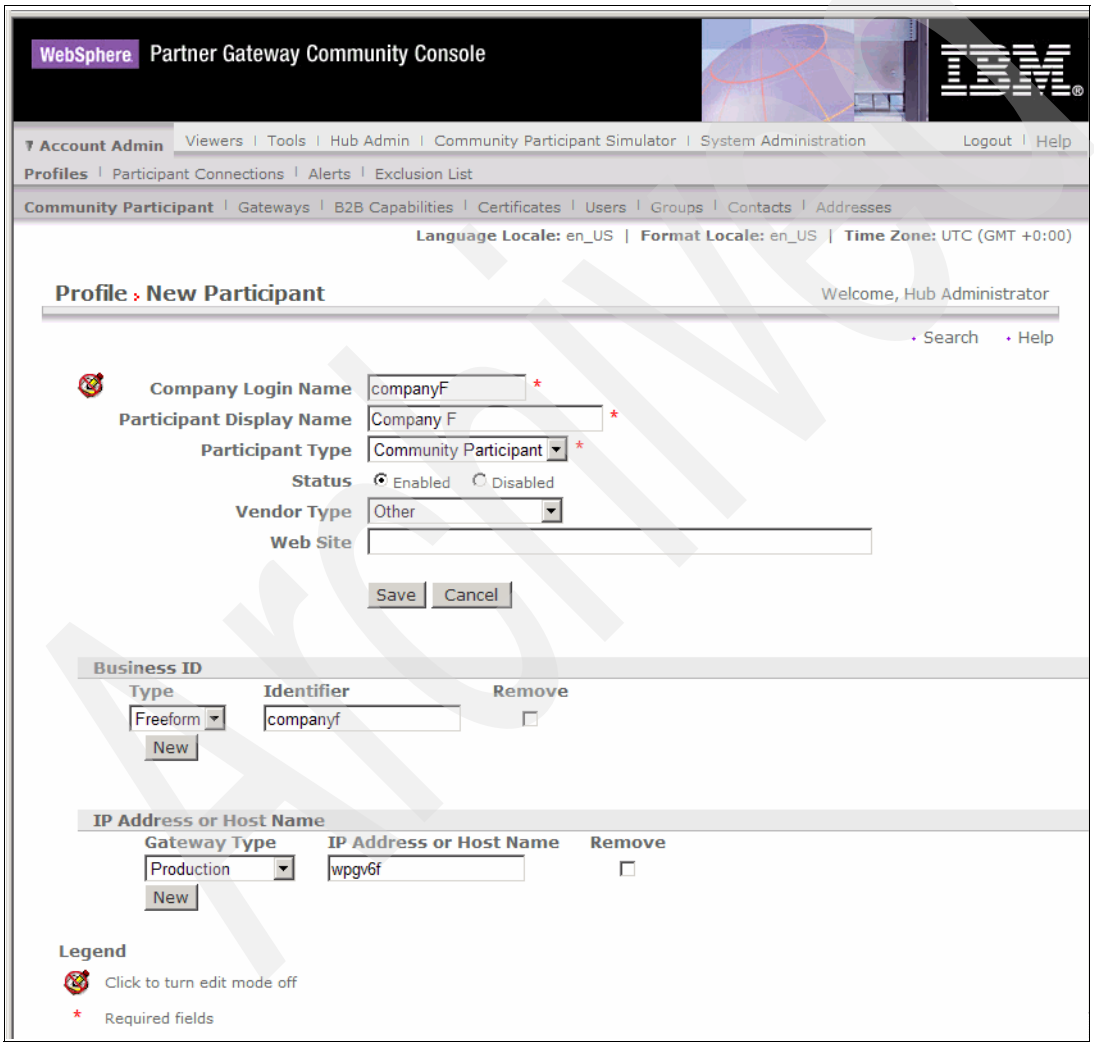


Figure 12-3 Creating a Participant profile for Company F

Creating a new target

A new target is needed to pick up the files that are destined for Company F. To integrate this for binary FTP documents, we use the FTP Directory type target that is able to determine binary document source and destination business identifiers.

1. Select **Hub Admin** → **Hub Configuration** → **Targets**.
2. When the list of currently defined targets loads, click **Create Target**.
3. In the Target Details window (Figure 12-4 on page 346), take these steps:
   a. Provide a name for the new target, for example `FTPTarget`.
   b. Set the transport to **FTP Directory**. This displays more attributes specific to the transport type.
   c. Set the FTP Root Directory to `/home/bcguser/wpgv6/data/ftp`, which is the folder that we created in 12.2.2, “Creating the directory structure” on page 341.
   d. The other attributes can have default values.
   e. Click **Save**.

   Two attributes on this screen are not used in this example:

   – The File Unchanged Interval is the amount of time the Receiver component of WebSphere Partner Gateway will wait before picking up the file from the directory. This is to make sure it is completely finished being written to the directory before processing.
   – The Exclude File Ext attribute is used to ignore certain file extensions that reside in the directory, meaning the Receiver will not pick them up at all.

Archived

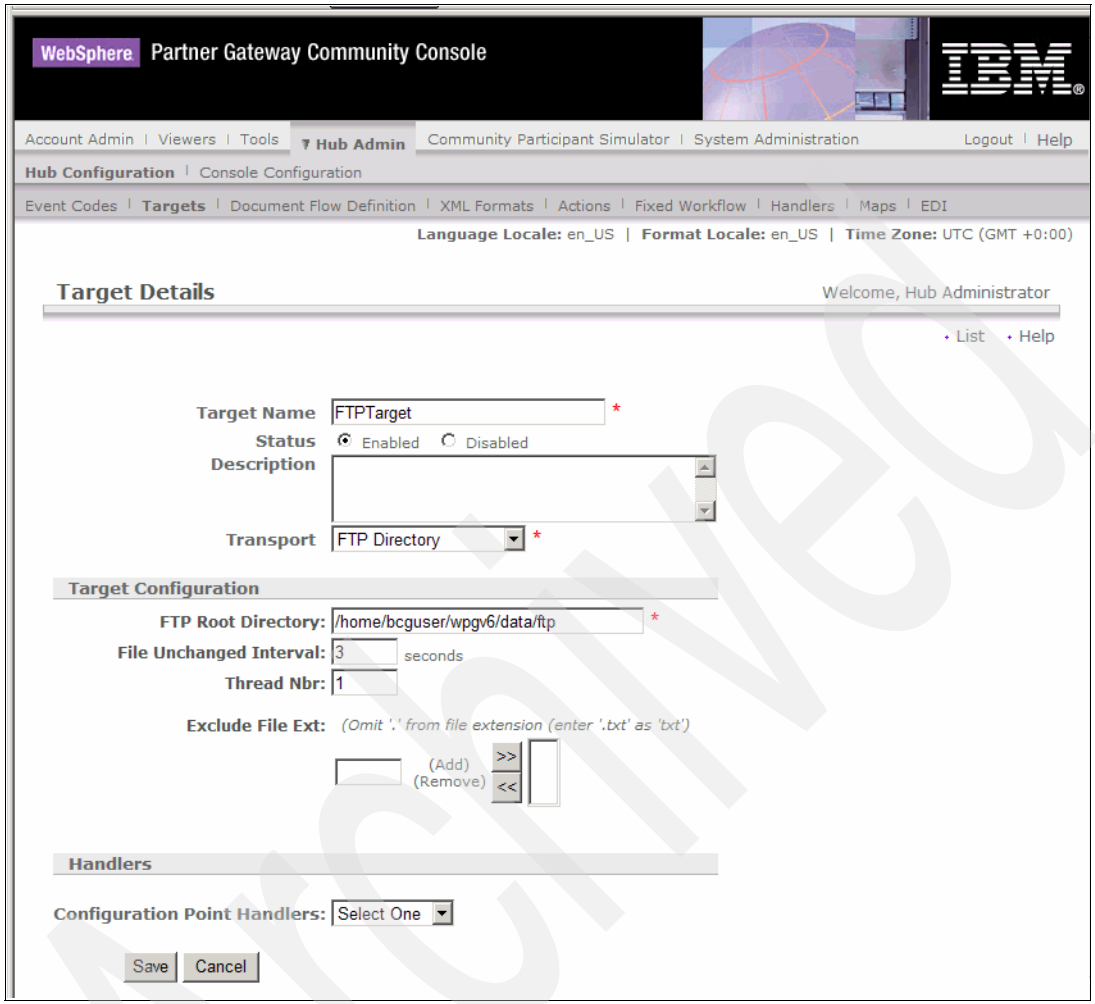


Figure 12-4 Creating a new FTP Directory type target.

4. After the **Save** action, we are presented with a confirmation screen that looks like Figure 12-5 on page 347.

Archived

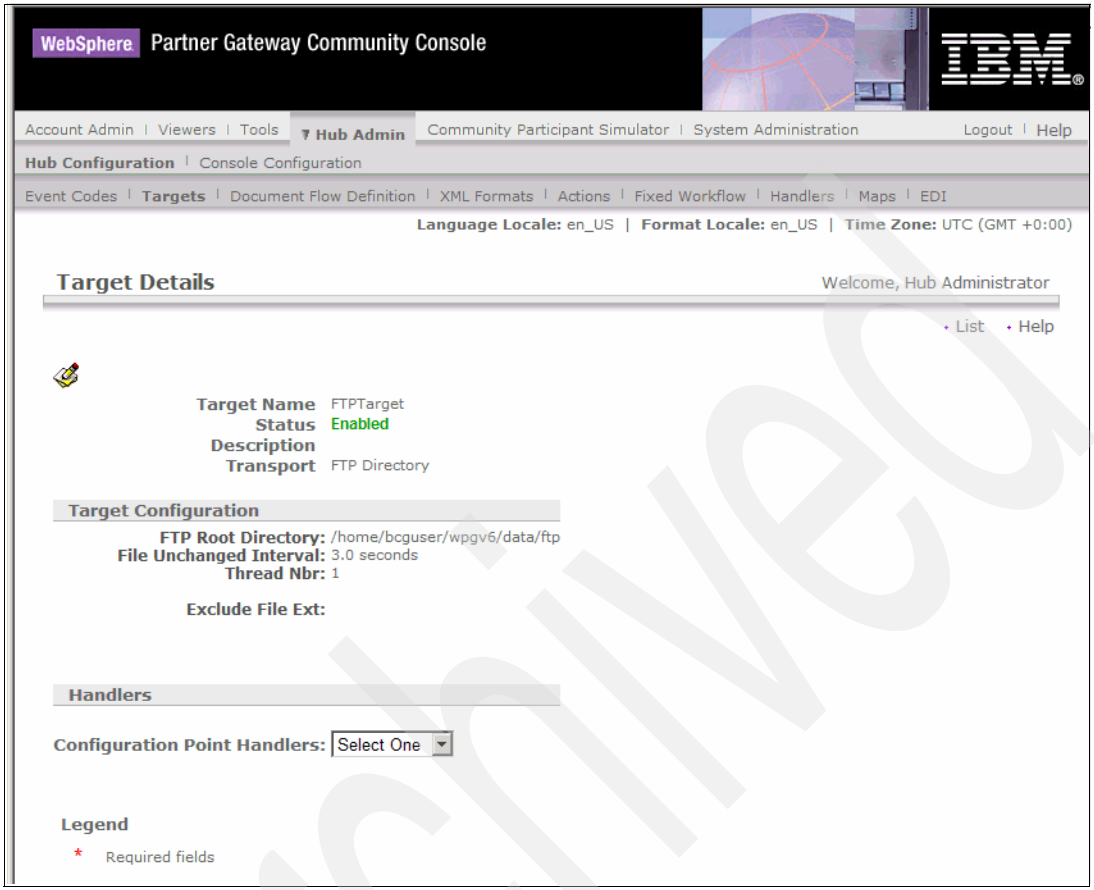


Figure 12-5 Newly created target confirmation

Creating the interaction

The required interaction is one of the simplest that can be configured within WebSphere Partner Gateway. However, it still must be created and enabled before it will work correctly. Follow these steps:

1. While logged on as hubadmin, select **Hub Admin** → **Hub Configuration** → **Document Flow Definitions**.
2. In the Manage Document Flow Definitions window (Figure 12-6 on page 348), expand the folder next to **Package: None** and then expand the **Protocol: Binary**. No specific action is needed here other than to verify that the Status on the left side of the screen in Figure 12-6 on page 348 is *enabled*.

Archived

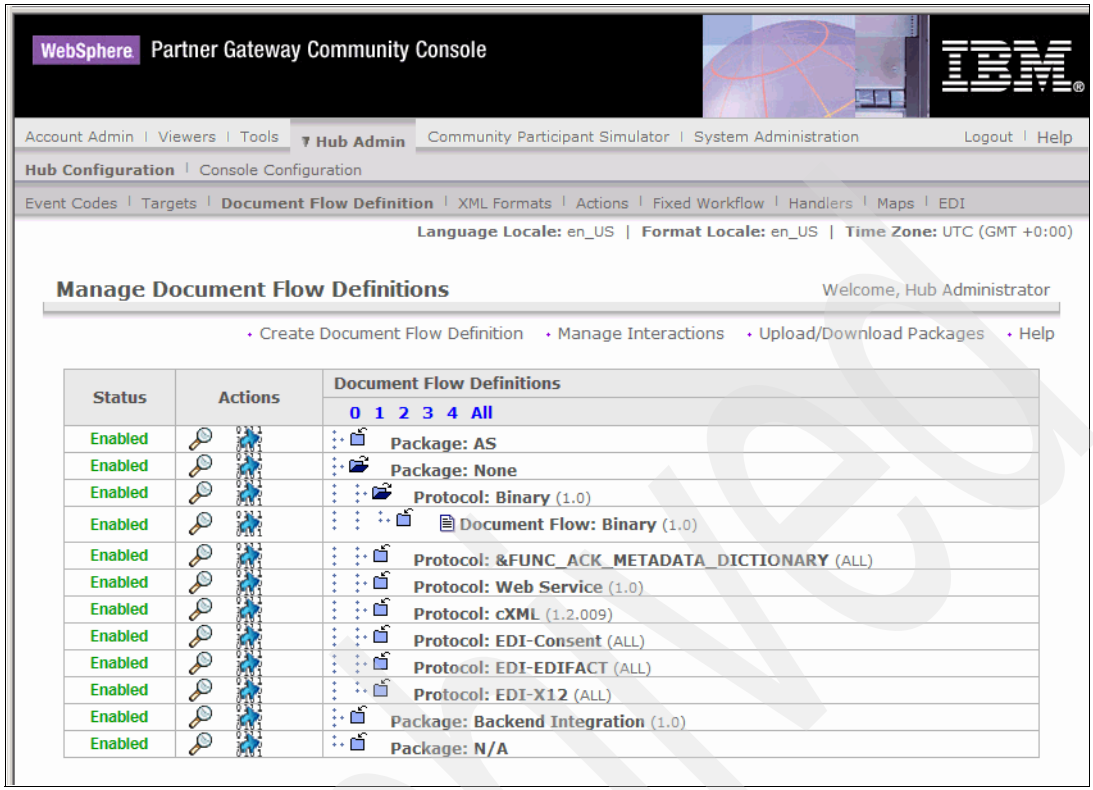


Figure 12-6 Enabled document flow definitions

3. To review the interactions, select **Manage Interactions**.
4. Click **Create Interaction**.
5. In the Create Interaction window (Figure 12-7 on page 349), complete these tasks:
   a. Expand the tree structure under the Source and Target headings for **Package: None.**
   b. Locate **Protocol: Binary** under the None folders, then click the radio button for **Document Flow: Binary** under both Source and Target as shown in Figure 12-7 on page 349.
   c. For Action select **Pass Through**.
   d. Click **Save**.

Archived

Figure 12-7 Interaction for None/Binary/Binary to None/Binary/Binary

6. We are returned to the Manage Interactions screen. Click **Search** here to see the new interaction that was created (Figure 12-8 on page 350).

Archived

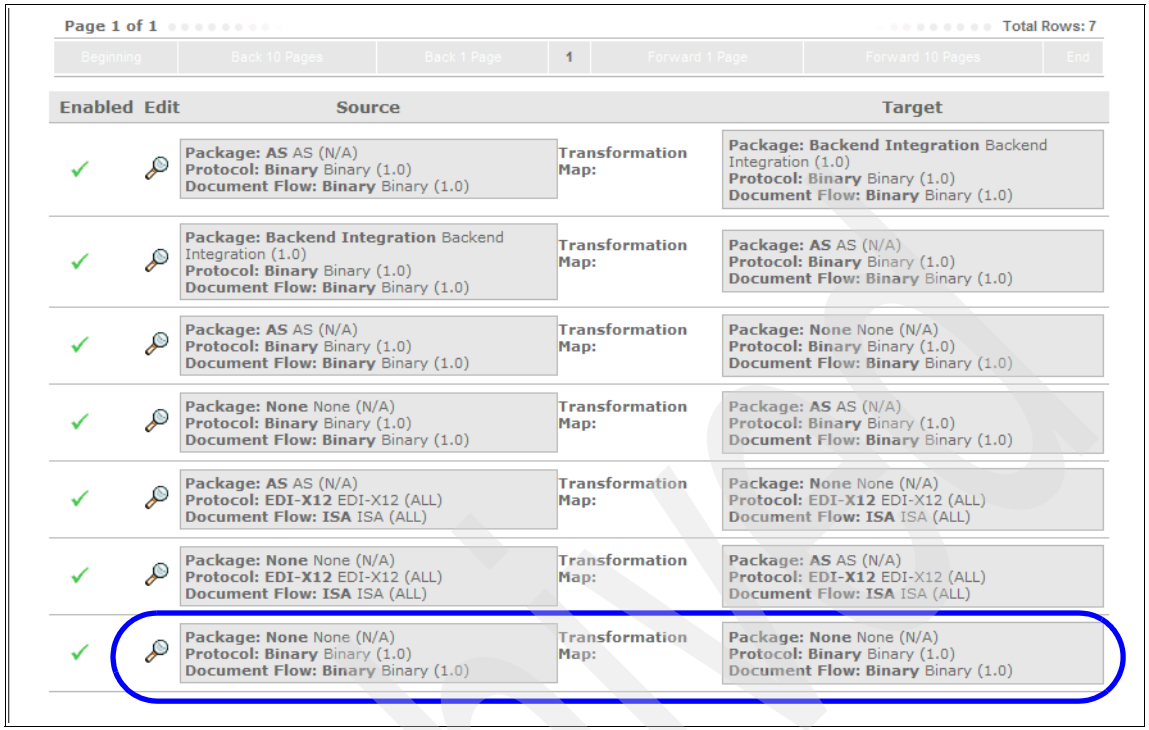


Figure 12-8 New interaction displayed on Manage Interaction screen

12.2.4 Updating the profile for Company F

In the previous section, we created a Community Participant profile for Company F, and created the system-wide target and interactions needed for the example. The Company F profile still needs some configuration, namely:

- The B2B capabilities for Company F need to be activated to match the interactions we plan to use.
- The FTP Gateway that points to the Company F FTP server needs to be created.

To accomplish this, follow these steps:

1. While logged on as the hub administrator, select **Account Admin** → **Profiles** → **Community Participant**.

Note: Notice we are configuring the profile of Company F as the hubadmin user and *not* by logging on as the admin user of Company F itself as described in 7.2, “Role-based configuration” on page 138. If Company F had its own WebSphere Partner Gateway implementation, it would be possible for an individual at Company F to login and create Company F’s own B2B capabilities and gateway at this point. However, because Company F does not have its own WebSphere Partner Gateway implementation, the more likely scenario would be the hubadmin user of Company A defining the necessary configuration for Company F.

2. Click **Search**.
3. Locate the profile of Company F and click the **magnifying glass** icon next to it. Select **B2B Capabilities** to indicate Company F wishes to handle binary files that have no packaging.
4. In the B2B Capabilities window, enable **Protocol** and **Document Flow: Binary** under packaging and **None** for source and target. The configuration should look like Figure 12-9 on page 352.

Archived

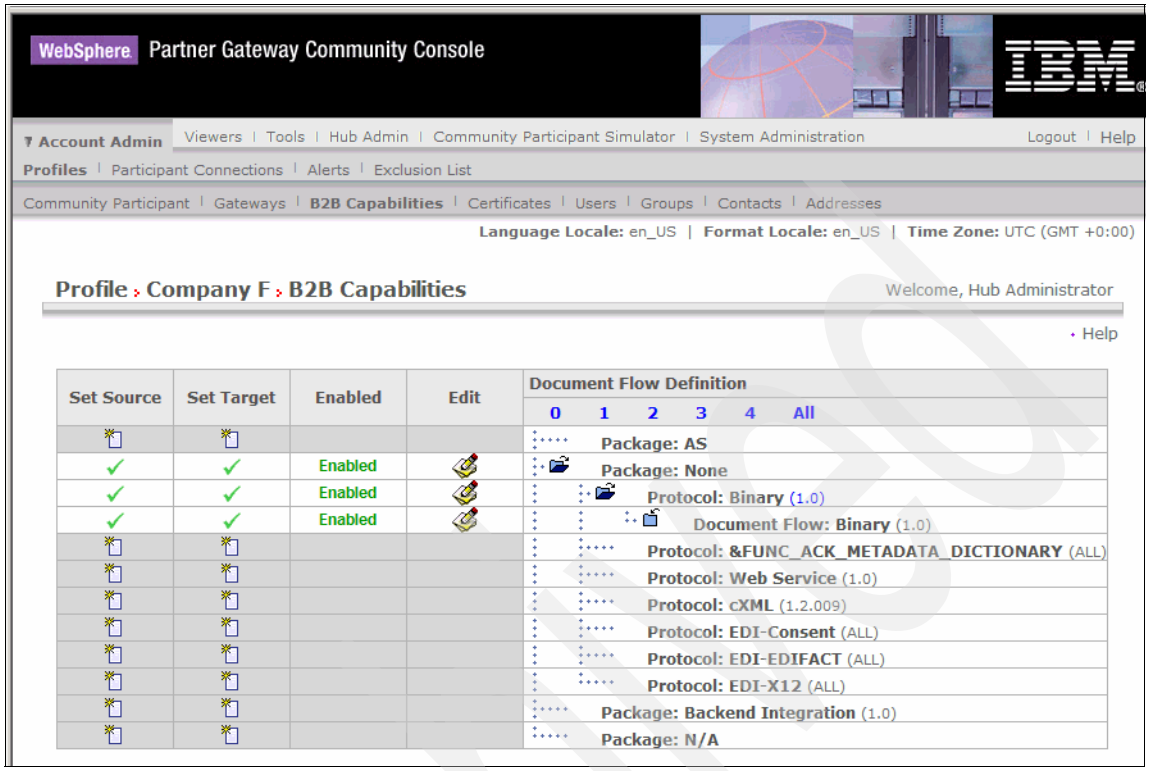


Figure 12-9 Setting the B2B Capabilities for Company F

5. Create the FTP gateway. While still working with the profile of Company F, select **Gateways**. We see an empty list of gateways. Click **Create**.
6. In the Gateway List window (Figure 12-10 on page 353), enter the following:
   a. For Gateway Name, type `FTPGateway`, for example.
   b. For Transport, select **FTP**.
   c. For Address, type `ftp://wpgv6f`. The value `wpgv6f` is the hostname of the FTP server that resides at Company F.
   d. Supply the User Name and Password to access the Company F FTP server.

      This would be given by Company F. The home directory that the account maps to would be the destination where Company F wants to receive the files from Company A.
   e. All other attributes can be left as default.
   f. Click **Save**.

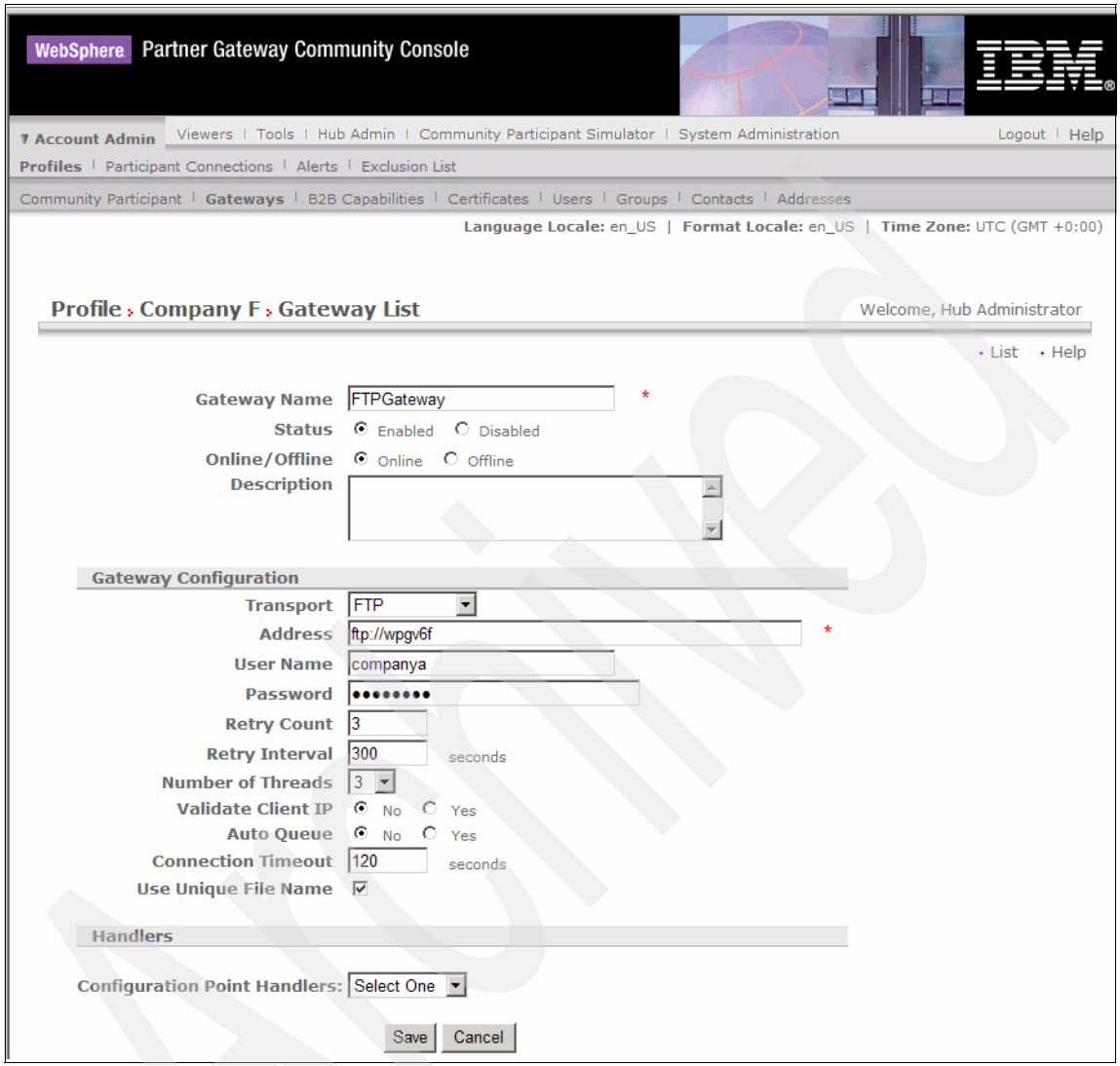


Figure 12-10 Creating the FTP Gateway within the Company F profile

7. Remember each partner profile needs a default gateway. Return to the list of gateways for Company F by clicking **List**, and then **View Default Gateways**.
8. Select **FTPGateway** as the production gateway and click **Save**. We see a list of gateways for Company F as shown in Figure 12-11 on page 354.

Archived

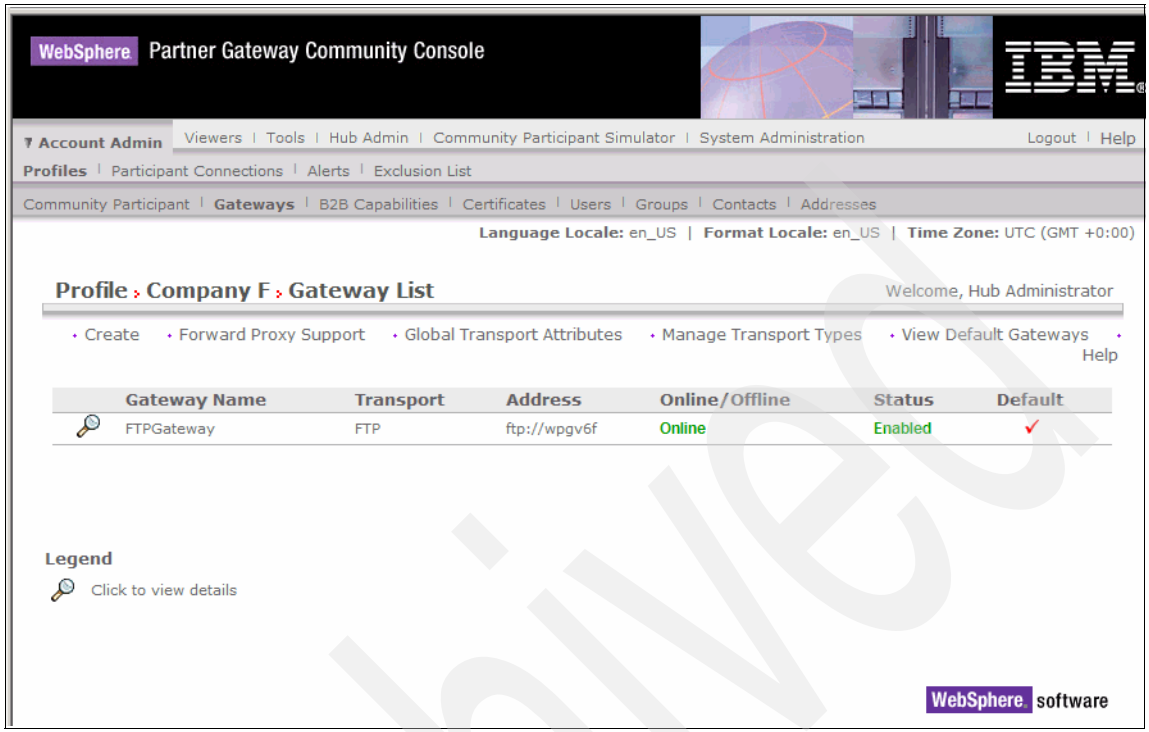


Figure 12-11 List of Gateways for Company F

12.2.5 Updating the profile for Company A

So far, we have created the new target, new interactions, and a new participant. For that new Participant (Company F) we have activated the B2B capabilities and created a gateway that points to the location Company F want to receive the files. However, we have yet to activate the B2B capabilities of Company A to match the interaction as well. To do this, follow these steps:

1. Select **Account Admin** → **Profiles** → **Community Participant**.
2. Click **Search**, and then click the **magnifying glass** icon next to **Company A**.
3. While working with the profile of Company A, select **B2B Capabilities**.
4. In the B2B Capabilities window (Figure 12-12 on page 355), enable the None/Binary/Binary for source and target. Figure 12-12 on page 355 shows the B2B capabilities of Company A.

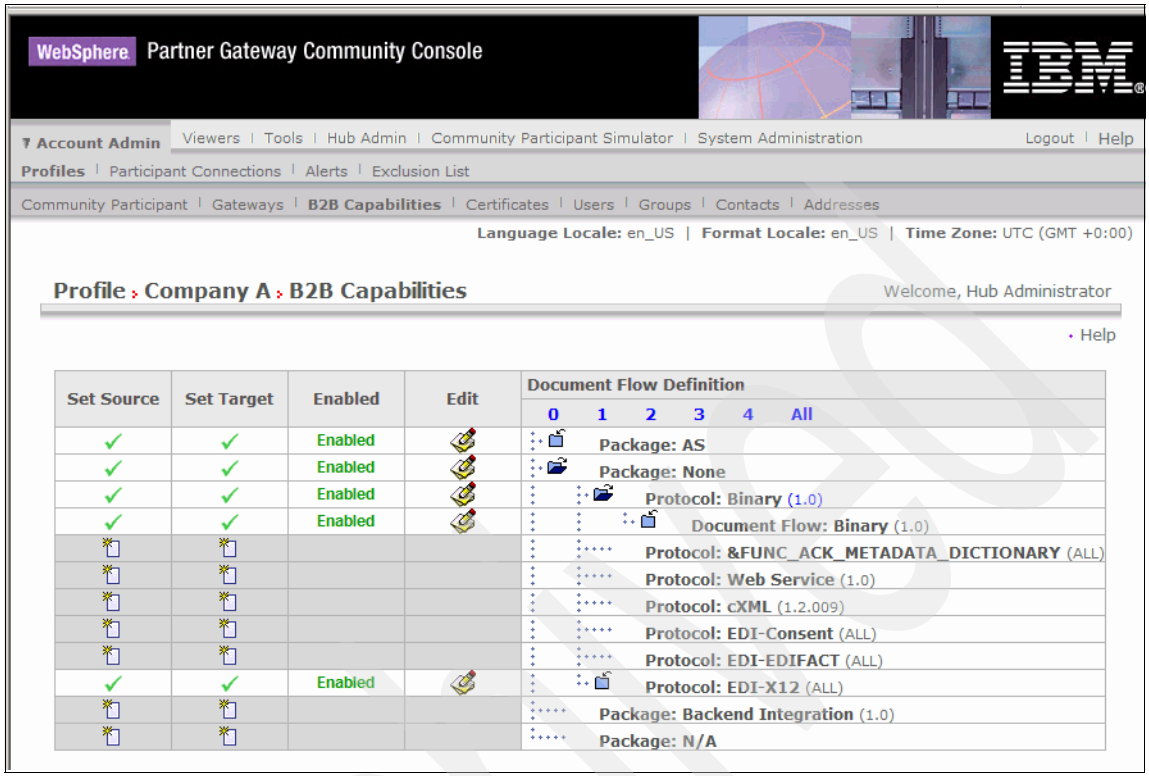


Figure 12-12 Updating the B2B capabilities of Company A

12.2.6 Creating a participant connection

The final step in this outbound configuration is to activate the needed participant connection between Company A and Company F.

1. While logged on as the hub administrator, select **Account Admin** → **participant connections**.
2. Select **Company A** as the source and **Company F** as the target. Click **Search** (Figure 12-13 on page 356).
3. We now see the interaction we created. It is based on the common B2B capabilities of companies A and F.

   Click **Activate** for this connection.

Archived

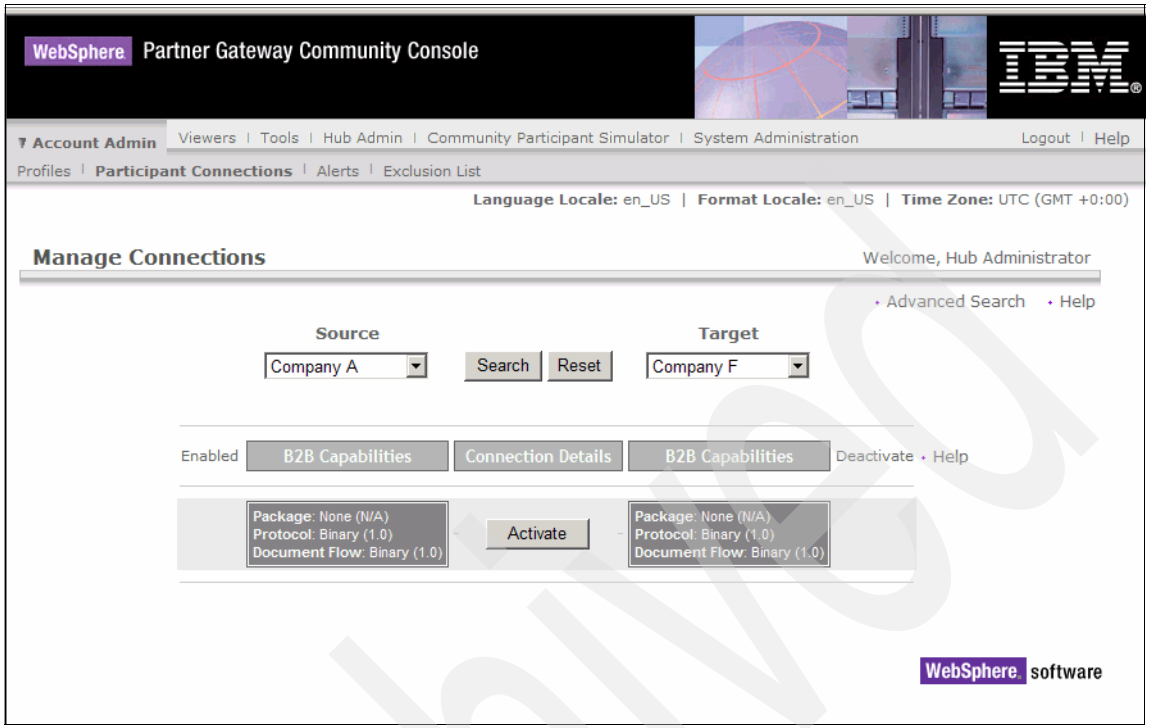


Figure 12-13 Unactivated possible connections between Company A and Company F

Figure 12-14 on page 357 shows the resulting activated connection. We can click **Gateways** to review the gateway selection. However, because we already made the FTPGateway as the default gateway for Company F, the correct information should already be configured.

Archived

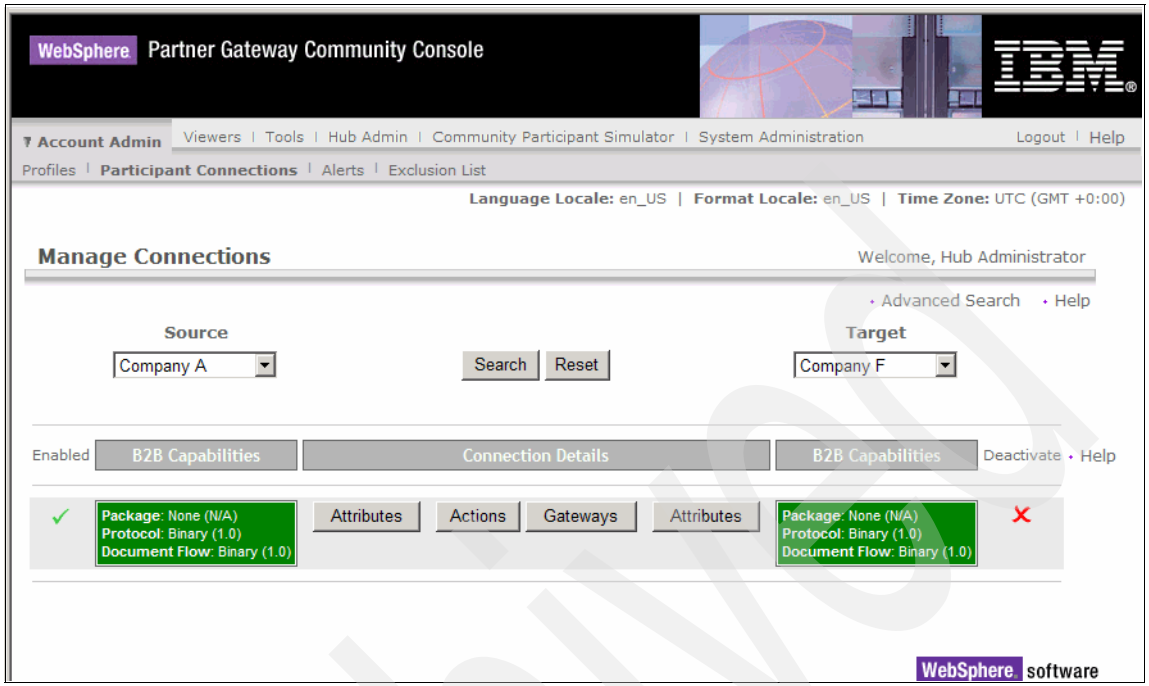


Figure 12-14 List of activated connections between Company A and Company F

12.2.7 Validating outbound communication

The configuration of the FTP server that resides at Company F is not important to the scope of this redbook. As long as Company F has an FTP server, the gateway that we configured at Company A points to that FTP server, and has the correct username and password, then WebSphere Partner Gateway will deliver the document to that location.

In the next section, for the inbound flow, we discuss how WebSphere Partner Gateway integrates with the default FTP server on the AIX machine of Company A. For the outbound flow, it is sufficient to say Company F has an FTP server that we can reach.

To validate the transfer of binary files, copy a file with a name that starts with `companyf` (for example `companyf.123456789.bin`) in the directory `/home/bcguser/wpgv6/data/ftp/companyA/binary/Production`. At the next polling cycle, the file is picked up by the FTPTarget target and passed to the Document Manager component of the server.

The Document Manager which knows the sending business identifier from the directory it picked up from, and the receiving business identifier from the file

name, will use the built-in FTP client functionality to deliver the file to company F. We can use the document viewer to monitor the progress and result of the transmission.

Figure 12-15 shows a successful exchange between Company A and Company F with binary files. Click the **magnifying glass** icon to view any events associated with this communication.

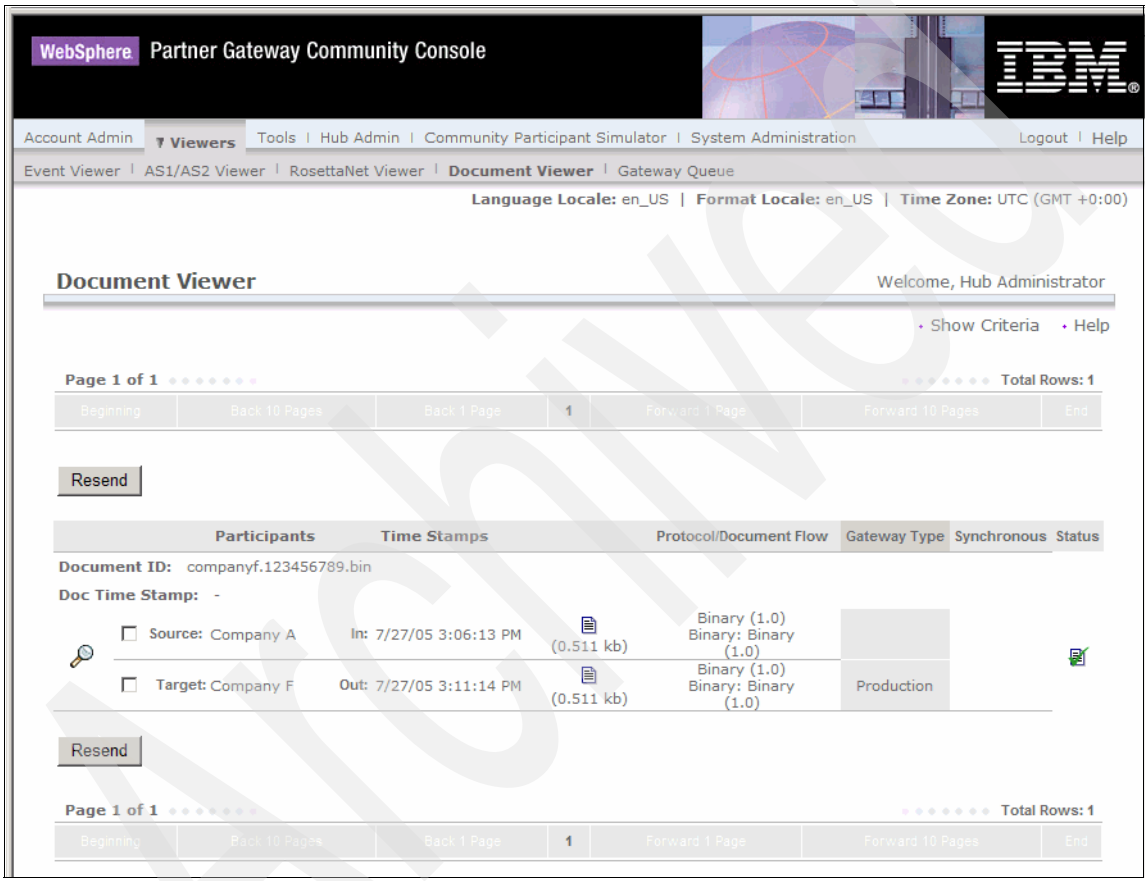


Figure 12-15 Using the document viewer for binary FTP exchange

Figure 12-16 on page 359 shows all the events associated with an exchange. As we can see, there is one warning event in the details. The first delivery attempt failed because the FTP server residing at Company F was intentionally shut down. After starting it, the document was successful. The number of retry attempts, and the time it would wait between attempts can be configured on the gateway.

Document Details

Welcome, Hub Administrator

• List • Help

Original File Name: companyf.123456789.bin
Reference Id: 1122476772746000255 4F1C42036544472B26A2C21126EE
Document ID: companyf.123456789.bin

Doc Time Stamp	Gateway Type	Connection Document Flow	Status
-	Production		

	Source	In Time Stamp	Source Business ID	Source Document Flow
0.511 kb	Company A	7/27/05 3:06:13 PM	companya	Binary 1.0: Binary Binary 1.0
	Target	**End State Time Stamp**	**Target Business ID**	**Target Document Flow**
0.511 kb	Company F	7/27/05 3:11:14 PM	companyf	Binary 1.0: Binary Binary 1.0

Document Children

None | Source | Target

No Children Available

Document Events

• Help

Event Filter: Debug | Information | Warning | Error | Critical

Total Event Count: 4

Event Name	TimeStamp	Type	Event Code	Location	Source IP
Channel lookup successful	7/27/05 3:06:13 PM	Info	BCG210204	Unknown	9.42.171.84
Document Sent to Outbound Processor	7/27/05 3:06:14 PM	Info	BCG210005	Unknown	9.42.171.84
First Delivery Attempt Failed	7/27/05 3:06:14 PM	Warning	BCG250011	System	9.42.171.84
Document Delivered	7/27/05 3:11:14 PM	Info	BCG250004	Unknown	9.42.171.84

Figure 12-16 Detailed events of an FTP binary exchange.

Figure 12-17 on page 360 shows the home directory of the user companya. This is the view the recipient of the document (in this case Company F) would see.

As we can see, the file was uploaded successfully. The name of the file has been altered by appending a unique document identifier to the end of the original file name. By deselecting the User Unique Filename parameter (shown in Figure 12-10 on page 353), we can have WebSphere Partner Gateway preserve the original filename after transmission.

Archived

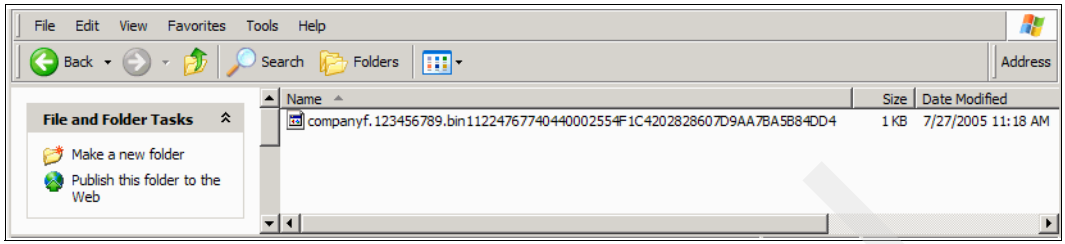


Figure 12-17 Contents of the home directory of user companya

During the configuration of this example, we created an interaction for None/Binary/Binary at both source and target. However, we can send other types of documents out to Company F over FTP as well. Files that are not binary, such as XML or EDI documents, can be placed in this directory for sending:

```
/home/bcguser/wpgv6/data/ftp/companyA/documents/Production
```

Remember that the Custom XML and EDI documents contain information that the server does read, unlike binary files. The document we send from the ../documents/.. directory will route based on those attributes and would not need the business identifier of the recipient in the file name.

12.3 Configuration of Company A for FTP Inbound

This section explains how to enable inbound FTP communication for Company A. We discuss how to integrate an FTP server with WebSphere Partner Gateway so that we can continue to use the features of WebSphere Partner Gateway, while using an FTP server to receive documents.

As mentioned earlier, WebSphere Partner Gateway itself does not have FTP server capability, but it is possible to integrate any FTP server with the product. The main concern is to make sure the FTP Directory type target has access to the files that the FTP server receives. The server does not actually listen on the FTP port, the server lets the FTP server handle the communication that Company F initiates. After the file is stored at some location that Company A can access, it pulls the file into the FTP Directory type target and passes it to the Document Manager for processing.

The Company A WebSphere Partner Gateway implementation is installed on an AIX machine. AIX has a built-in FTP server that we can use to illustrate this concept.

12.3.1 Integrating with the AIX FTP server

Because Company A happens to have a basic FTP server already installed on the same machine WebSphere Partner Gateway is installed on, it makes the integration relatively simple. We first need a location that both the FTP server can write *to* and that WebSphere Partner Gateway can read *from*.

Creating a directory structure

For simplicity, we can use the same root directory structure that we used for outbound communication. For outbound communication we used the directory:

```
/home/bcguser/wpgv6/data/ftp/companyA
```

For inbound we need to create a directory for Company F. Namely:

```
/home/bcguser/wpgv6/data/ftp/companyF
```

The command to execute this on AIX is shown in Example 12-6. As before, the name of this folder should match the Company Login Name when the Participant was created. In this case it was **`companyF`**. We should be user **`bcguser`** to execute this command.

Example 12-6 Creating companyF folder

```
$ whoami
bcguser
$ pwd
/home/bcguser/wpgv6/data/ftp
$
$ mkdir companyF
```

To create the subdirectory structure, follow these steps:

1. Within the directory `companyF`, create the same structure that we described in the outbound flow example. This includes two subdirectories, `binary` (to receive binary documents) and `documents` (to receive XML and EDI documents)
2. For each of these directories, create subdirectories for `Production` and `Test`.

   Example 12-7 depicts the commands from the above steps.

Example 12-7 Creating sub-directory structure for companyF

```
$ pwd
/home/bcguser/wpgv6/data/ftp/companyF
$
$ mkdir binary
$ mkdir documents
$
```

Archived

```
$ cd binary
$
$ mkdir Production
$ mkdir Test
$
$ cd ..
$ cd documents
$
$ mkdir Production
$ mkdir Test
```

The results of a recursive list command is shown in Example 12-8, showing the full structure that was just created.

Example 12-8 Result of a recursive list command

```
$ pwd
/home/bcguser/wpgv6/data/ftp/companyF
$
$ ls -R
binary     documents
./binary:
Production  Test

./binary/Production:

./binary/Test:

./documents:
Production  Test

./documents/Production:

./documents/Test:
```

Creating the companyF user on AIX

Just as Company A needed a user ID and password to log on to the Company F FTP server. Company F now needs access to the Company A FTP server simulation. The AIX default FTP server uses the user ID access list of the AIX machine itself.

To give Company F a user ID to use FTP we can use the **smitty** utility as follows:

1. From a command line as the root user, enter **smitty.**
2. We are presented with the smitty System Management screen (Figure 12-18 on page 363). Move the cursor to **Security and Users** and press Enter.

```
                                        System Management

Move cursor to desired item and press Enter.

  Software Installation and Maintenance
  Software License Management
  Devices
  System Storage Management (Physical & Logical Storage)
  Security & Users
  Communications Applications and Services
  Print Spooling
  Problem Determination
  Performance & Resource Scheduling
  System Environments
  Processes & Subsystems
  Applications
  Installation Assistant
  Cluster System Management
  Using SMIT (information only)
```

Figure 12-18 smitty System Management interface

3. At the Security and Users screen, select **Users**, then press Enter.
4. On the Users screen, select **Add a User**, then press Enter (Figure 12-19).

```
                                              Users

Move cursor to desired item and press Enter.

  Add a User
  Change a User's Password
  Change / Show Characteristics of a User
  Lock / Unlock a User's Account
  Reset User's Failed Login Count
  Remove a User
  List All Users
```

Figure 12-19 smitty Add a User screen

5. We now need to add the specific characteristics of the user. Use these settings:
   - User NAME is `company_f`
   - Primary GROUP is `bcggroup`
   - HOME directory is `/home/bcguser/wpgv6/data/ftp/companyF`

   The rest of the settings can be left blank or default (Figure 12-20 on page 364). Press Enter to execute the command.

Archived

Note: Notice that we set the home directory for the `companyf` user to a directory that begins with `/home/bcguser` and the primary group to `bcggroup`. To make the integration with WebSphere Partner Gateway simple, we are using the same FTP Directory Target to pickup files for inbound communication that we did for outbound communication. The target itself does not distinguish between outbound and inbound, it is only an entry point to the Document Manager component of the server. Because /companyF folder resides under /home/bcguser/wpgv6/data/ftp, it knows when there is a file from companyF that requires processing.

There are many ways to integrate an FTP server with the directory structure. The only requirement is that user `bcguser` has access to whatever location the FTP server writes the file to, that the directory structure that we created is preserved, and that a target scans the root directory of that structure.

```
                                    Add a User

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                   [Entry Fields]
* User NAME                                            [companyf]
  User ID                                              []                                  #
  ADMINISTRATIVE USER?                                  false                              +
  Primary GROUP                                        [bcggroup]                          +
  Group SET                                            []                                  +
  ADMINISTRATIVE GROUPS                                []                                  +
  ROLES                                                []                                  +
  Another user can SU TO USER?                          true                               +
  SU GROUPS                                            [ALL]                               +
  HOME directory                                       <home/bcguser/wpgv6/data/ftp/companyF>
  Initial PROGRAM                                      []
  User INFORMATION                                     []
  EXPIRATION date (MMDDhhmmyy)                         [0]
  Is this user ACCOUNT LOCKED?                          false                              +
  User can LOGIN?                                       true                               +
  User can LOGIN REMOTELY(rsh,tn,rlogin)?               true                               +
  Allowed LOGIN TIMES                                  []
  Number of FAILED LOGINS before                       [0]                                 #
      user account is locked
  Login AUTHENTICATION GRAMMAR                         [compat]
  Valid TTYs                                           [ALL]
  Days to WARN USER before password expires            [0]                                 #
  Password CHECK METHODS                               []
  Password DICTIONARY FILES                            []
  NUMBER OF PASSWORDS before reuse                     [0]                                 #
  WEEKS before password reuse                          [0]                                 #
  Weeks between password EXPIRATION and LOCKOUT        [-1]
  Password MAX. AGE                                    [0]                                 #
  Password MIN. AGE                                    [0]                                 #
  Password MIN. LENGTH                                 [0]                                 #
  Password MIN. ALPHA characters                       [0]                                 #
  Password MIN. OTHER characters                       [0]                                 #
  Password MAX. REPEATED characters                    [8]                                 #
  Password MIN. DIFFERENT characters                   [0]                                 #
  Password REGISTRY                                    []
  Soft FILE size                                       [2097151]                           #
  Soft CPU time                                        [-1]
  Soft DATA segment                                    [262144]                            #
  Soft STACK size                                      [65536]                             #
[MORE...11]

Esc+1=Help             Esc+2=Refresh          Esc+3=Cancel           Esc+4=List
Esc+5=Reset            Esc+6=Command          Esc+7=Edit             Esc+8=Image
Esc+9=Shell            Esc+0=Exit             Enter=Do
```

Figure 12-20 User settings on the smitty Add a User screen

6. We see a command completion message from smitty. Exit out of the smitty program using the **Exit** option at the bottom of the screen.
7. Returning back to the root prompt, we need to create an initial password for the user companyf that was just created. Type `passwd companyf`. We are prompted to enter a the password twice (Example 12-9).

Example 12-9 creating initial password for companyf user

```
# passwd companyf
Changing password for "companyf"
companyf's New password:
Enter the new password again:
```

8. Remember that the password we created is only the initial password on an AIX installation. To make it permanent, we need to telnet to the AIX machine and enter the username and password we just created. We are prompted one more time to change the password. By default, AIX will accept the same password if you choose.
9. Finally, we need to alter the permissions of the directories now that Company F has a defined user. We created all the directories as bcguser so that user currently owns the entire directory structure. To do this, follow these steps:
   a. Remain as root user and traverse to the `/home/bcguser/wpgv6/data/ftp` directory, then execute **`chown -R companyf:bcggroup companyF.`** This turns over ownership of the directory structure to the companyf user.
   b. However, the bcguser user still needs write access to this directory. To correct this, execute **`chmod -R 775 companyF`**, which gives group write access to the directory structure and since bcguser and companyf users are in the same group, they will both have access.

      Example 12-10 shows these commands executed.

Example 12-10 Altering permissions of companyf user directory structure

```
# cd /home/bcguser/wpgv6/data/ftp
#
# chown -R companyf:bcggroup companyF
#
# chmod -R 775 companyF
#
```

We have now created a user **`companyf`** that can ftp to the default FTP server of the AIX machine from Company F's site. The subdirectory structure is already created underneath the `/home/bcguser/wpgv6/data/ftp/companyF` directory with the correct permissions. The next step is to configure the WebSphere Partner Gateway server to be aware of inbound FTP connections from Company F.

12.3.2 Updating the configuration of WebSphere Partner Gateway

Now that Company F can contact Company A's FTP server and send a file, WebSphere Partner Gateway will need a way to retrieve that file. The manner in which we created the home directory for companyf user under `/home/bcguser/wpgv6/data/ftp` means that the FTPTarget that already exists on the system will pick up these files, and because we used the required directory structure for FTP Directory targets, it will know the inbound file is from Company F.

As part of the outbound flow, we configured the B2B capabilities of Company A and Company F so that both could send and receive binary files. We also created an interaction that defined this flow in both directions. However we did not activate the participant connection in the reverse direction, Company F to Company A. To do so, follow these steps:

1. Log on to the console of WebSphere Partner Gateway as hubadmin and select **Account Admin** → **Participant Connections**.
2. Select **Company F** as the source and **Company A** as the target. Click **Search**.
3. We see the available connection in Figure 12-21 on page 367.

   Click **Activate** for the connection between None/Binary/Binary and None/Binary/Binary.

Archived

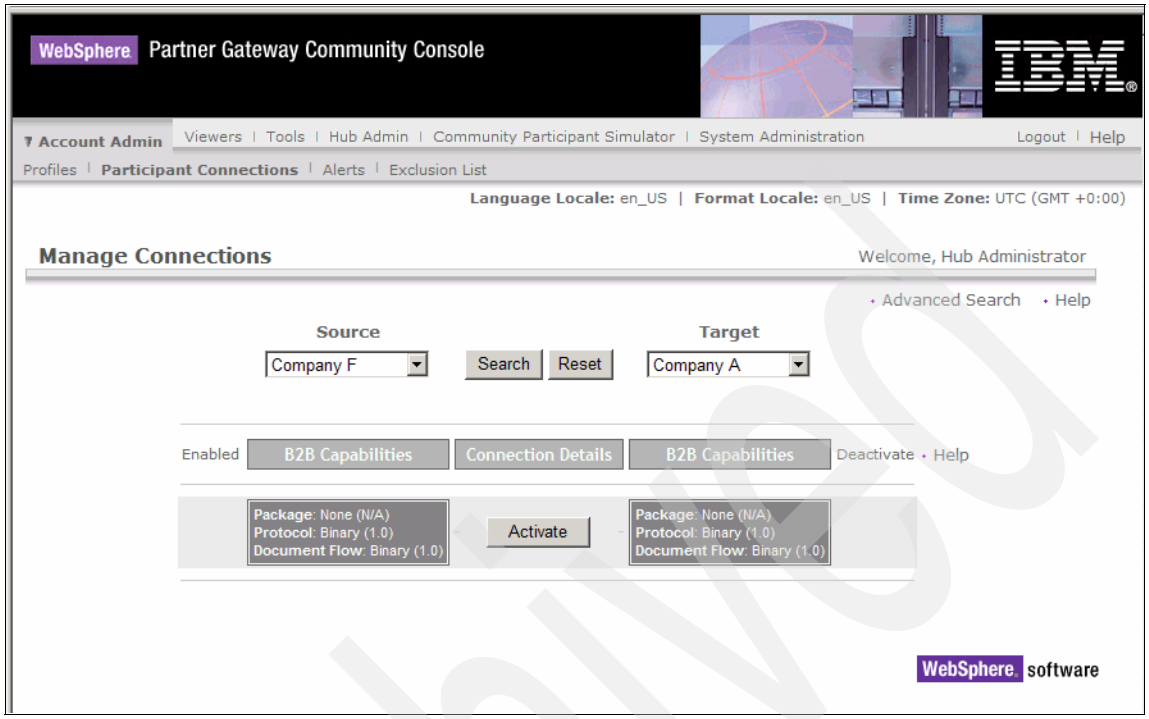


Figure 12-21 List of possible participant connections

We see the result of the activation in Figure 12-22 on page 368.

Archived

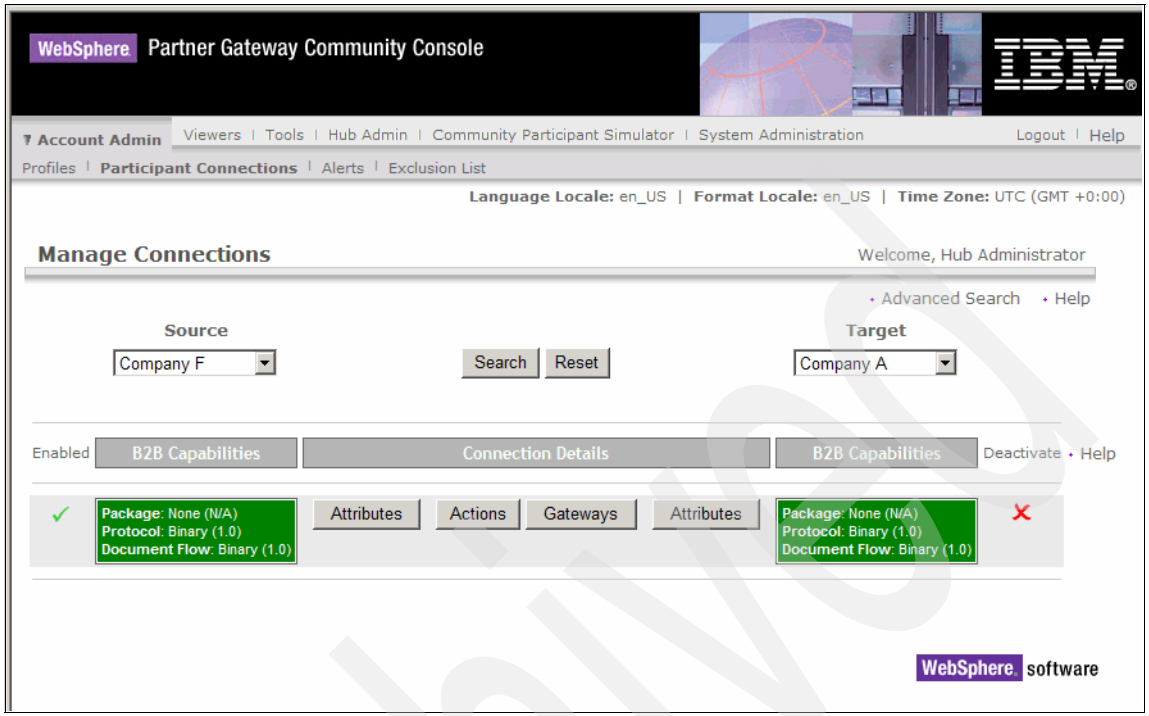


Figure 12-22 List of activated connections between Company F and Company A

12.3.3 Validating Inbound communication

To validate this setup for inbound communication, we need to send a file over FTP from Company F to Company A. As before, the file name must start with `companya`, which is the business identifier of Company A. It should also have some unique name following the companya by a `.`*nnnnn*`.`*xxx* format, where *nnnnn* and *xxx* represent the uinque ID and the file extention, for example:

```
companya.1234567890.bin
```

We send the file using FTP to the FTP server that resides on the AIX machine at Company A. The file is sent to the /binary/Production directory under the home directory of user companyF. This gives the WebSphere Partner Gateway server all the necessary information it needs to route the document successfully.

Example 12-11 shows a portion of the dialog between a standard Windows FTP client and the FTP server hosted by Company A.

Example 12-11 FTP Client uploading a binary file to Company A

```
230 User companyf logged in.
```

```
ftp> pwd
257 "/home/bcguser/wpgv6/data/ftp/companyF" is current directory.
ftp>
ftp> cd binary
250 CWD command successful.
ftp>
ftp> cd Production
250 CWD command successful.
ftp>
ftp> put companya.1234567890.bin
200 PORT command successful.
150 Opening data connection for companya.1234567890.bin.
226 Transfer complete.
ftp: 19030 bytes sent in 0.00Seconds 19030000.00Kbytes/sec.
ftp>
ftp> bye
221 Goodbye.
```

After uploading the file to this directory:

```
/home/bcguser/wpgv6/data/ftp/companyF/binary/Production
```

The arrival of the file is noticed by the FTP Directory target. The target is part of the Receiver component of the WebSphere Partner Gateway server. The receiver picks up the file and passes it to the Document Manager component. In this case, the Document Manager has very little work to do because there is no packaging on the document. It sends the file along to the destination, FileSystemGateway. This is the same local directory that we have used several times in this redbook:

```
/home/bcguser/data/wpgv6/data/companya/edi_in
```

This demonstrates that WebSphere Partner Gateway can write any kind of file to a gateway, so there is no real need to name or create gateways for a specific document type or source (as we did for the /edi_in directory).

Open this directory:

```
/home/bcguser/wpgv6/data/companya/edi_in
```

Now we can see a new file with the name:

```
companya.1234567890.bin<uniqueString>.
```

This `<uniqueString>` gets added to the name to file by default, but it is a configurable option. This was mentioned in 12.2.7, “Validating outbound communication” on page 357.

Another way to verify the successful processing of the file received by Company A's FTP server is by using the Document Viewer. The Document Viewer shows the details of the document sent. In Figure 12-23, the source and target business identifiers have been recognized correctly by the WebSphere Partner Gateway server.

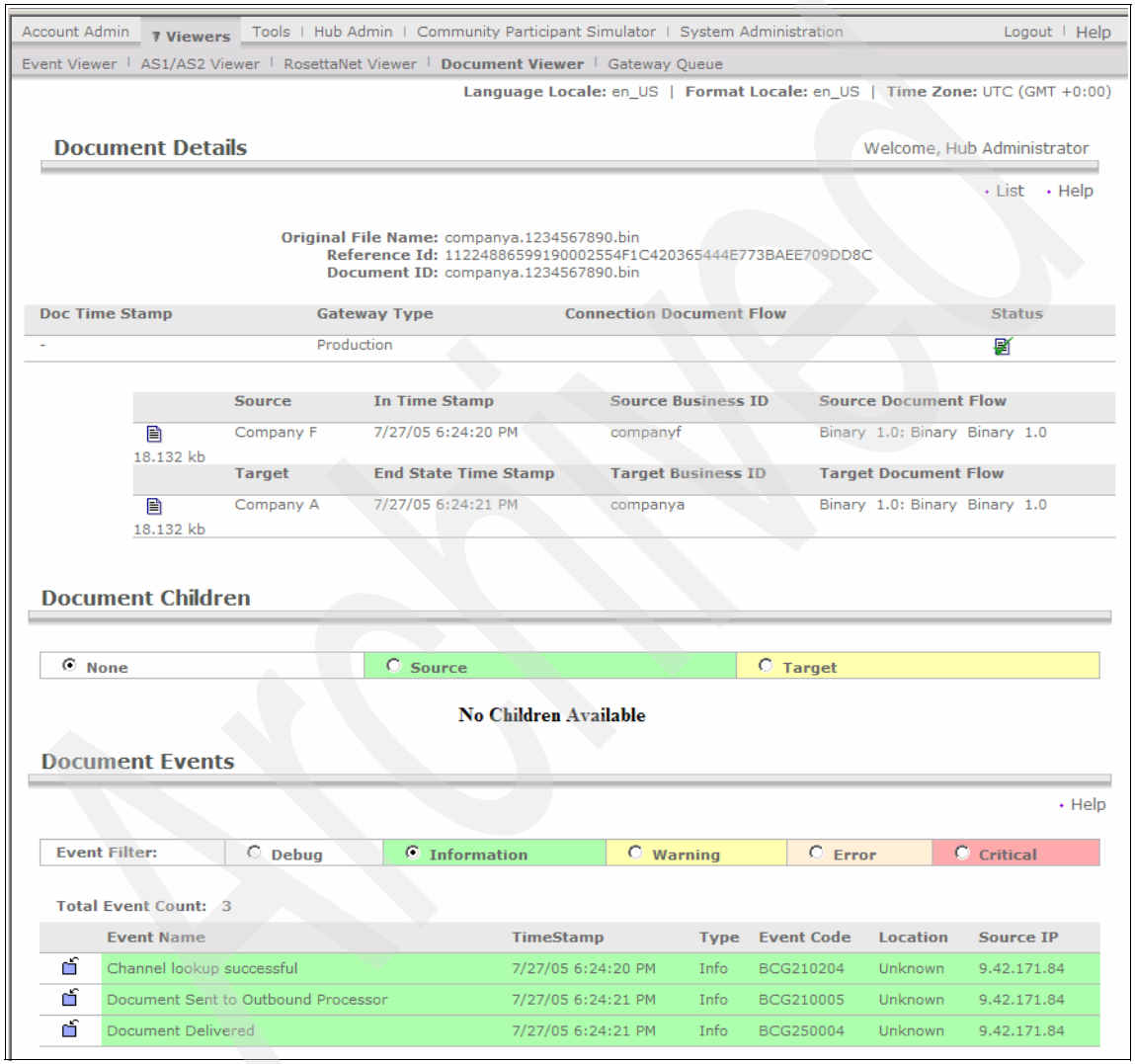


Figure 12-23 Using document viewer on Inbound FTP files

In the same manner as inbound FTP, we could exchange Custom XML or EDI documents. In those cases, we would have to have a CustomXML or EDI channel defined between Company A and Company F. We would also have to

FTP the file into the `../documents/Production` directory instead of the `/binary/Production` directory. If the interaction and connection were correctly implemented though, we would not have to name the file with any special restrictions. WebSphere Partner Gateway would then determine the source and target business identifiers by the packaging.

So far in this chapter, we have executed outbound and inbound FTP communication between two participants, Company A and Company F. One of the benefits of integrating FTP communication with WebSphere Partner Gateway is the ability to view all the transactions with the Document Viewer. In the outbound example, the server acted as an intelligent FTP client with built-in retry ability. In the inbound example, WebSphere Partner Gateway allowed the transactions to be viewed in the Document Viewer, but only moved the received file from one directory to another within the local file system. However, WebSphere Partner Gateway can provide more value to inbound transactions by directly bridging the FTP server with other gateway types, such as a JMS message queue. The only configuration we would need to perform in that case would be to switch the destination gateway from FileSystemGateway to any other defined gateway.

12.4 Implementing FTPS

One obvious disadvantage of using FTP to exchange critical business documents is the low level of security. Basic FTP transmissions are not secure and it is inherent to the standard. Fortunately, the Internet Engineering Task Force has accepted a standard to extend standard FTP with a number of security extensions. For more information about the RFC2228 standard, go to:

http://www.ietf.org/rfc/rfc2228.txt?number=2228

Basically this standard allows for a Secure Sockets Layer (SSL)-like extension of FTP, similar to HTTP and HTTPs. This means we can add an SSL handshake to the FTP communication between the FTP client and FTP server. This allows for the authentication of the client, server or both and encryption on the data sent.

12.4.1 FTPS outbound

FTPS and the use of certificates is highly dependent on the FTP server. As mentioned earlier, WebSphere Partner Gateway does not have FTP server functionality so it has limited interaction with FTPS and certificates. In 12.2, “Configuration of Company A for FTP outbound” on page 338. Company A sent a document to the FTP server of Company F. Company A started with a file that it put in a local file system, it was picked up, processed and WebSphere Partner Gateway handled the communication with the FTP server at Company F. Now if

Company F decides it wants more protection on this transmission, we will need to adjust the configuration so that the server will establish communication with Company F's FTP server in a secure manner.

What exactly does this mean to the configuration of the WebSphere Partner Gateway? Company A still needs to pick up the file it wants to send. This is done by the FTP Directory target. However the target does not care where the file is going, it only exists to pass the file into the Document Manager component of the server. The Document Manager processes the document, then delivers the document to Company F by using the FTPGateway. If the FTP server at Company F's site becomes an FTPS server and wants only secure communications we are going to need a new gateway first.

Configuring an FTPS gateway

The configuration of an FTPS gateway is exactly the same as a regular FTP gateway. Only the pull-down menu option of Transport Type is going to change. To do this, follow these steps:

1. Log on to the Community Console of Company A as hubadmin.
2. Click **Search** on the Participant Search screen.
3. Locate the profile of Company F and click the **magnifying glass** icon next to it. Select **Gateways**. We should see the currently configured FTP Gateway (Figure 12-24).

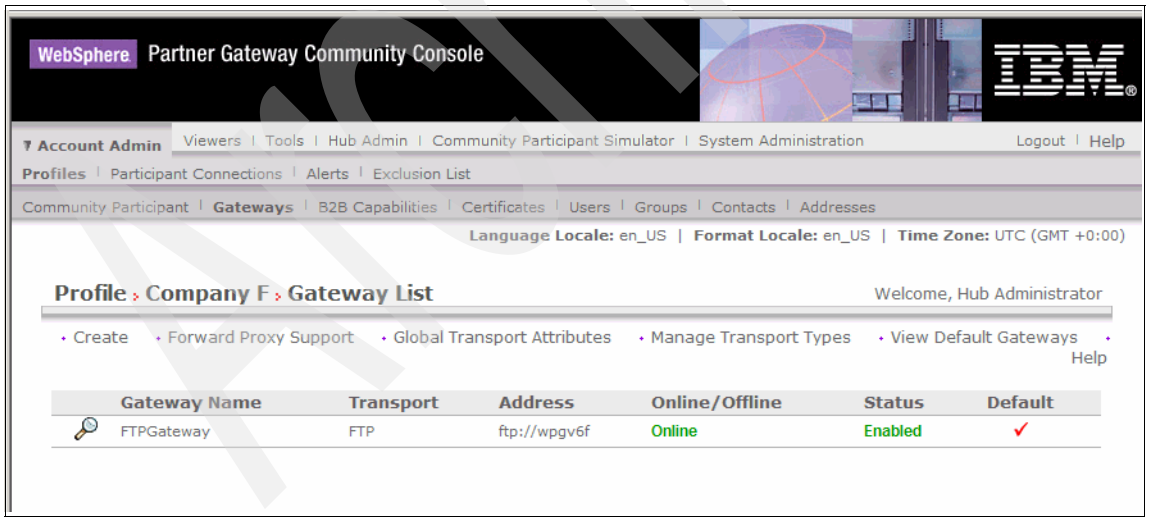


Figure 12-24 List of Company F's defined gateways

4. Click **Create**.

5. Enter a Gateway Name, `FTPSGateway` for example, and select **FTPS** as the Transport. Then enter the following information (Figure 12-25 on page 374):
   a. Address is the hostname of the FTP server at Company F, in our scenario it is `wpgv6f`.
   b. Enter the User Name and Password (that would have been given to you from Company F).
   c. Leave the rest as default.
   d. Click **Save**.

> **Note:** The User Name and Password that are needed will really depend on the situation encountered with the partner company. You notice we are using the same User Name and Password as with the FTPGateway. If Company F added a completely new server, this ID would most likely change. It is also possible that they extended and reconfigured its existing FTP server to only accept secure connections.

Archived

WebSphere Partner Gateway Community Console

Account Admin | Viewers | Tools | Hub Admin | Community Participant Simulator | System Administration | Logout | Help

Profiles | Participant Connections | Alerts | Exclusion List

Community Participant | Gateways | B2B Capabilities | Certificates | Users | Groups | Contacts | Addresses

Language Locale: en_US | Format Locale: en_US | Time Zone: UTC (GMT +0:00)

Profile > Company F > Gateway List

Welcome, Hub Administrator

List | Help

Gateway Name: FTPSGateway *

Status: Enabled / Disabled

Online/Offline: Online / Offline

Description:

Gateway Configuration

Transport: FTPS

Address: ftp://wpgv6f *

User Name: companya

Password: ••••••••

Retry Count: 3

Retry Interval: 300 seconds

Number of Threads: 3

Validate Client IP: No / Yes

Auto Queue: No / Yes

Connection Timeout: 120 seconds

Use Unique File Name: ☑

Handlers

Configuration Point Handlers: Select One

Save | Cancel

Figure 12-25 Creating the FTPS gateway within the Company F profile

6. Click **List**. The resulting list of gateways now configured for the Company F profile should look like Figure 12-26 on page 375.

Archived

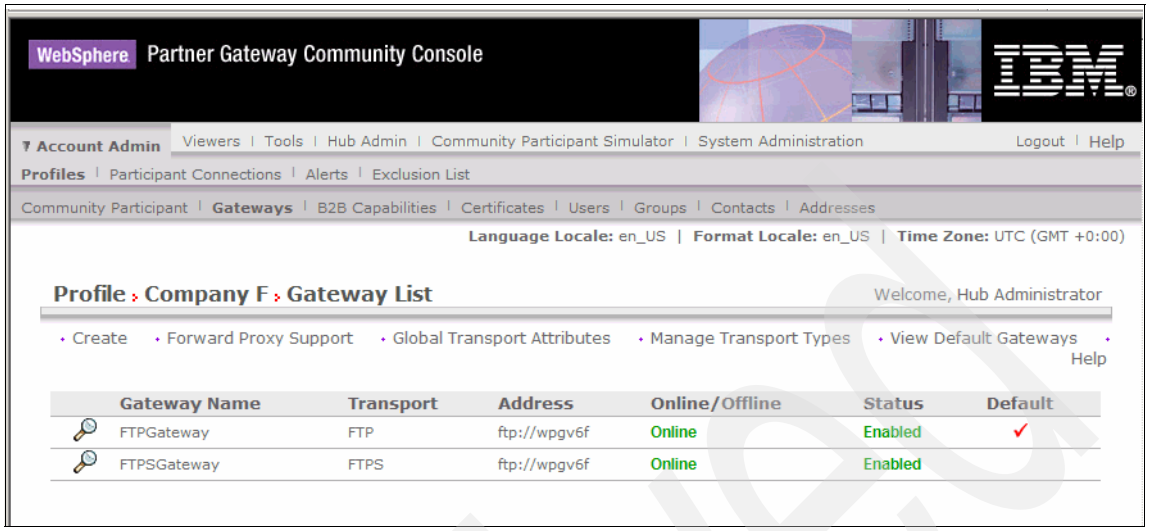


Figure 12-26 List of gateways for Company F

We have now added a new gateway for Company F with a transport type of FTPS. However, the participant connection that we configured in 12.2.6, “Creating a participant connection” on page 355 still defines the previous FTPGateway as the gateway to use. We update that in this next section.

Updating the participant connection

Although we have created a new gateway for the communication between Company A and Company F, we have not configured WebSphere Partner Gateway to use it. To do this, follow these steps:

1. While logged on as hub administrator, select **Account Admin** → **participant connections**.
2. For Source, select **Company A** and for Target, select **Company F**.
3. Click **Search**.
4. When presented with the list of Active connections between Company A and Company F, click the **Gateways** button (Figure 12-27 on page 376).

Archived

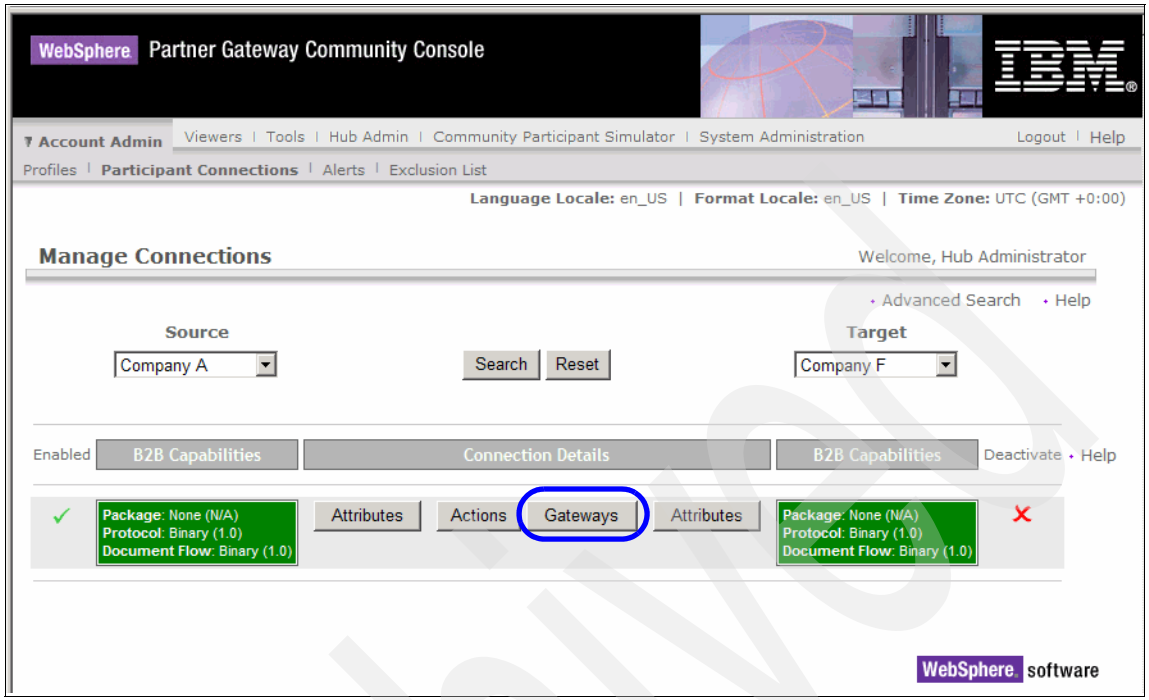


Figure 12-27 List of participant connections between Company A and Company F

5. We are presented with a new window named Edit participant connection. Under Target Gateways, select the new **FTPSGateway** that was just created (Figure 12-28 on page 377).

> **Note:** Now that Company F has multiple gateways defined, we did *not* set the new FTPS gateway to the Default Gateway. A Participant profile does require one Default Gateway, but it is this assignment (Figure 12-28) that ultimately determines the gateway WebSphere Partner Gateway will use. Having said that, the gateway most commonly used should be set to the default gateway.

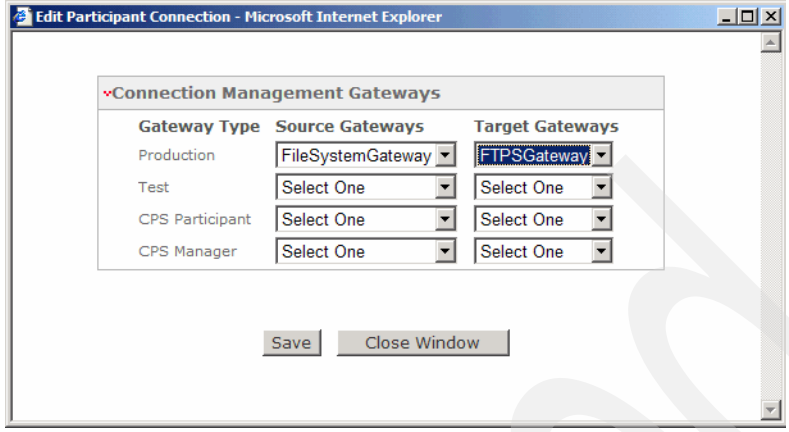


Figure 12-28 Changing the gateway in the participant connection

At this point, we have defined the new gateway (destination) and updated the participant connection to use this gateway. A file destined for Company F, can now be picked up, processed and sent to the correct gateway. The only step that remains is to update WebSphere Partner Gateway with the correct certificate that it will need to complete the SSL handshake.

Uploading a new certificate to Company A

In Chapter 8, “Securing the B2B exchange” on page 207 we examined certificates and some of their application in WebSphere Partner Gateway. We used certificates for digital signatures and encryption. For outbound server authentication over FTPS we are concerned with only one certificate. The certificate the FTPS server of Company F is going to present to us during the SSL handshake.

To implement FTPS, Company F will have already created its own certificate. Many FTPS servers have included functionality that creates certificates for use. Most certificate formats can be converted to each other. The creation of the certificate at Company F’s site is outside the scope of this redbook. The important point to remember is WebSphere Partner Gateway will need Company F’s public certificate in DER format to upload.

During the SSL handshake, that certificate is presented to Company A and will need to trust that certificate or terminate the handshake. To trust the certificate, it needs to load the public portion of that certificate as a root certificate under the Hub Operator User.

Follow these steps:

1. As the hubadmin user on Company A Community Console, select **Account Admin** → **Profiles** → **Community Participant**.
2. Click **Search**.
3. Click the **magnifying glass** icon next to the **Hub Operator** profile.
4. Within the Hub Operator Profile, click **Certificates**. We should see a view similar to (Figure 12-29).

> **Note:** You can safely ignore the warning message about a secondary digital certificate. This is explained in Chapter 8, "Securing the B2B exchange" on page 207. Secondary certificates are usually needed when the primary certificate is about to expire to ensure continuous secure operation.

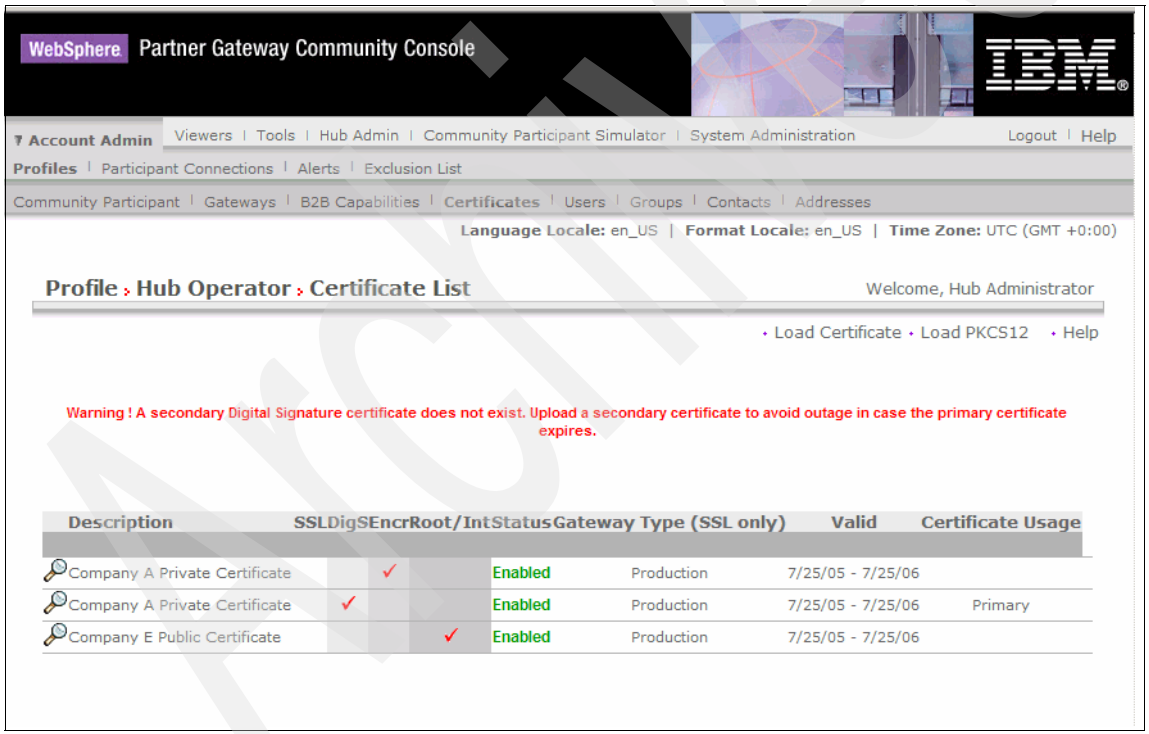


Figure 12-29 Company A Hub Operator certificate view

5. Click **Load Certificate**.
6. On the Create New Certificate screen (Figure 12-30 on page 380), enter the following:

a. Select the check box next to **Root and Intermediate.**

b. Enter a Description, `Company F Public Certificate`, for example.

c. Select the radio button for **Enabled** next to **Status**.

d. For Certificate, **Browse** to the location the certificate for Company F's FTPS server resides. The browser opens a window to the machine running the Community Console Web interface.

e. Leave the rest as default.

f. Click **Upload**.

Archived

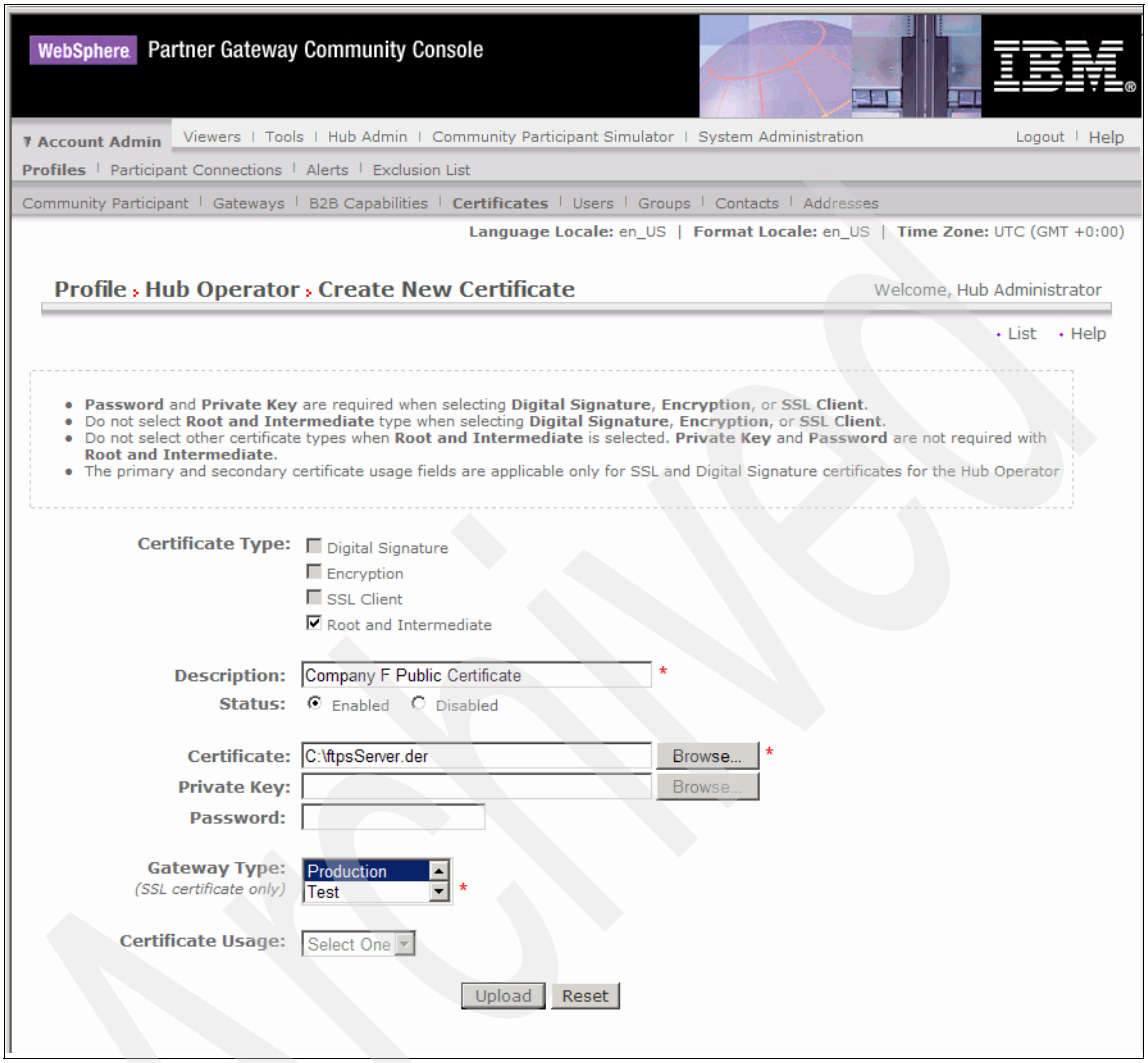


Figure 12-30 Uploading Company F public certificate

7. When the Certificate is uploaded, we need to press the **Save** button highlighted in Figure 12-31 on page 381 to confirm the configuration.

Archived

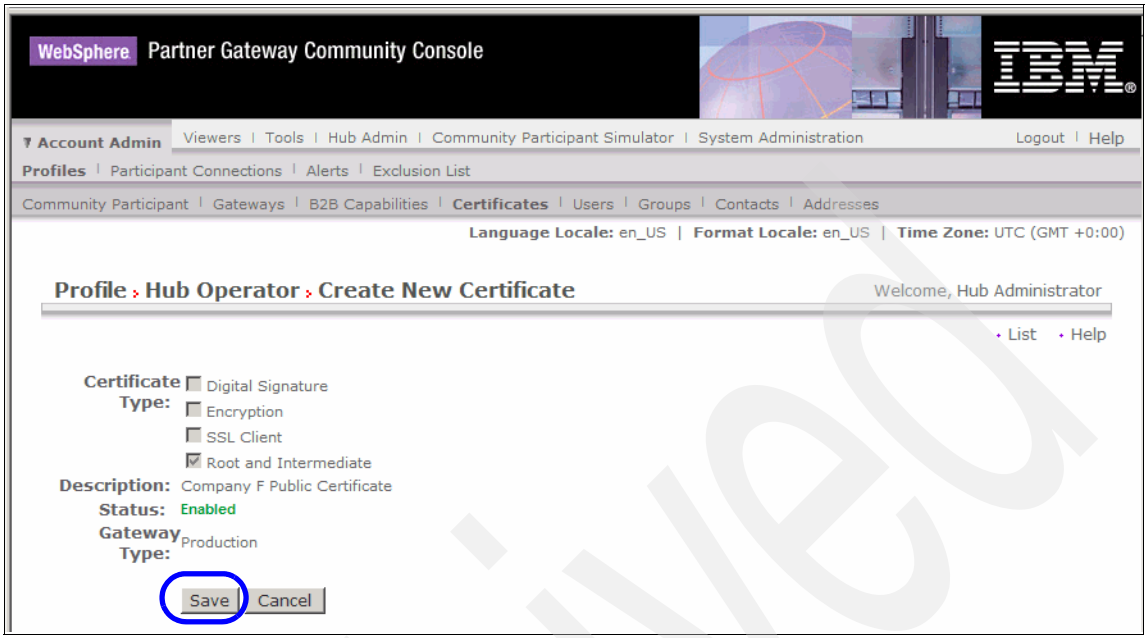


Figure 12-31 Confirming the upload

8. Click **List**. This will take us back to the uploaded certificates for the Hub Operator Profile. It should look like Figure 12-32 on page 382.

   Again, ignore the Secondary certificate warning message.

Archived

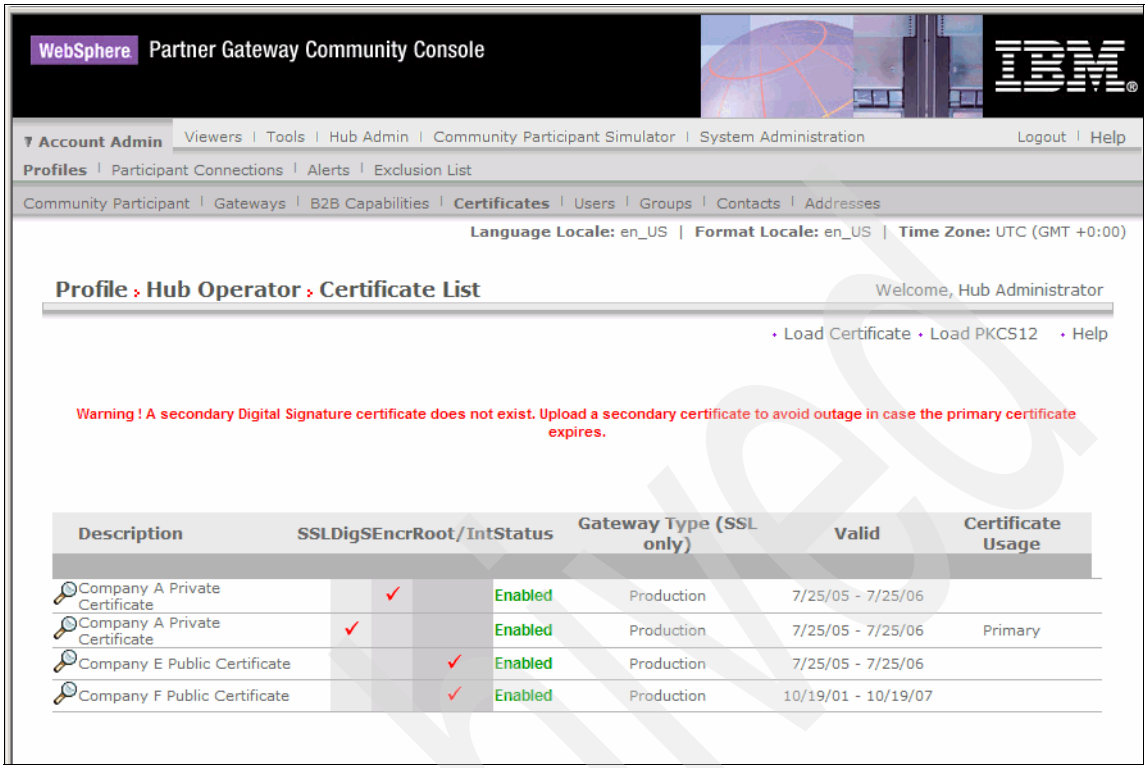


Figure 12-32 New List of Certificates for Company A

Now Company A is ready to attempt communication over the new gateway. The configuration is essentially the same as the outbound example in 12.2, “Configuration of Company A for FTP outbound” on page 338. We have created a new gateway for a destination, updated the participant connection, and loaded the certificate it needs for the SSL handshake.

Validating FTPS outbound

Because the configuration is, the same, for the most part, to validate the new configuration we only need to put the companyf.123456789.bin file into the /home/bcguser/wpgv6/data/ftp/companyA/binary/Production directory. This will initiate the send process from WebSphere Partner Gateway.

This time, WebSphere Partner Gateway uses an FTPS client to contact the FTPS server at Company F. It sends the **`auth`** command during the initial contact. The FTPS server responds with its certificate. The certificate is validated by the WebSphere Partner Gateway server. This is why the upload of the Company F Public Certificate was critical. The server uses this certificate to encrypt any further communication with the FTPS server at Company F, including the **`user`**

and **pass** commands. In this way, only the Company F FTP server can decrypt the data successfully.

At the end of the transmission the file is no different than the upload over the standard FTPGateway. The file is available to Company F in the home directory of the companya user.

We can see the results in Figure 12-33.

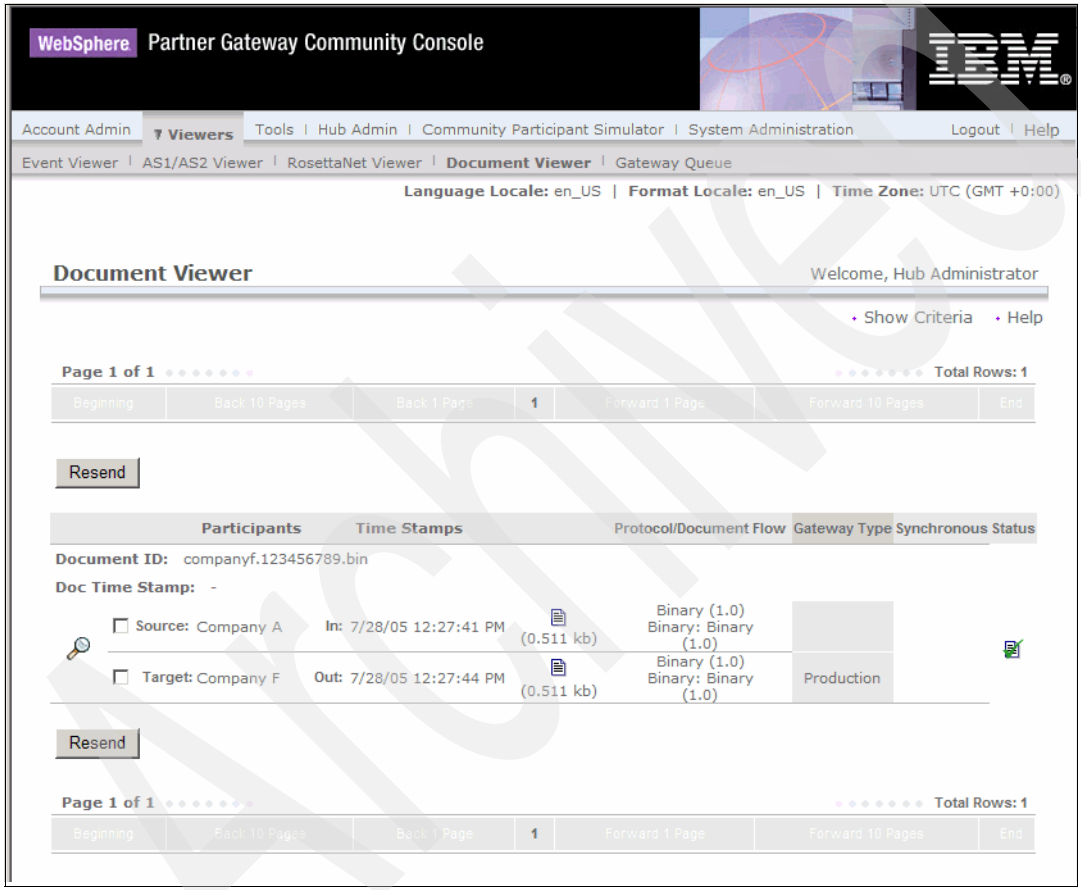


Figure 12-33 Document View of successful FTPS outbound transfer

12.4.2 FTPS client authentication

In the FTPS outbound example in 12.4.1, “FTPS outbound” on page 371 the SSL handshake was using what is called server authentication. That is, Company A was presented with the Company F Public Certificate which was validated and then used to encrypt the FTP communication. However there is one further step

that can be taken, extend the SSL handshake to use client authentication. WebSphere Partner Gateway also supports this optional version of the SSL handshake.

In client authentication when Company A contacts the FTPS server of Company F, the server certificate is still sent by Company F as before. In addition to that certificate, the FTPS server requests a certificate from the client, which is Company A.

The actual configuration of client authentication would be completely dependant on Company F's FTP implementation. Nothing on the WebSphere Partner Gateway configuration would change about the participant connection, but it would need a certificate to provide to the FTPS server.

We have already created a private certificate for Company A in 8.2.5, "Company A generates a private/public key pair" on page 226. Just as we loaded the Company A Private Certificate for Encryption we could upload the certificate (or another private certificate) for SSL. That certificate would be provided to Company F during the SSL handshake when requested. Company F would need the public certificate of Company A so he could validate it. No additional configuration on the WebSphere Partner Gateway side is needed.

12.4.3 FTPS inbound

For FTPS inbound communication, Company A would need an FTP server that supports FTPS. The basic FTP server that the AIX installation has, does not support these security extensions. Downloading a new FTP server for Company A and configuring it for FTPS is outside the scope of this redbook. If you are interested in performing this task, there are several FTP servers available on the Internet that support FTPS. There is also a way to accomplish FTPS inbound communication without Company A having an FTP server at all. That is discussed in 12.5, "Configuration of FTP Scripting outbound" on page 384, and is the second method we mentioned at the beginning of this chapter.

12.5 Configuration of FTP Scripting outbound

FTP Scripting is a new type of gateway and target in WebSphere Partner Gateway Version 6. It is briefly described in 12.1, "Overview of FTP and FTP Scripting scenarios" on page 336. The addition of this feature in WebSphere Partner Gateway is basically to allow greater flexibility in FTP communications. In this section. we configure FTP communication between Company A and Company F outbound, this time using the new FTP Scripting gateway. Keep in mind that this is just another approach to enabling FTP between Company A and Company F. It covers the same ground as 12.2, "Configuration of Company A for

FTP outbound" on page 338.You would not need *both* of these configurations in a real-world production environment.

12.5.1 Outbound FTP Scripting overview

12.2, "Configuration of Company A for FTP outbound" on page 338 discussed one method of doing this for straight binary files. This time, assume Company F has started to implement an EDI solution at its site, and wants EDI documents to be sent, but still over FTP transport methods. In addition, Company F wants these new files in a separate directory of its FTP server. This will illustrate some of the additional capabilities of the FTP Scripting Gateway that we are using.

Here is a summary of what is needed:

1. A Target to pickup EDI files for transmission outbound, this exists from Chapter 7, "Creating a basic B2B exchange" on page 137.
2. Update the B2B capabilities of Company F to include None/EDI-X12/ISA for source and target.
3. Update the B2B capabilities of Company A to include None/EDI-X12/ISA for source and target. This is already done in Chapter 7, "Creating a basic B2B exchange" on page 137.
4. Create an interaction between None/EDI-X12/ISA and None/EDI-X12/ISA with action Pass Through.
5. Create a new FTP Scripting Gateway within Company F's profile. We need to provide a script that can alter the destination directory.
6. Update the participant connection between Company A and Company F that uses the new interaction and new gateway.

This summary is depicted in Figure 12-34 on page 386.

Archived

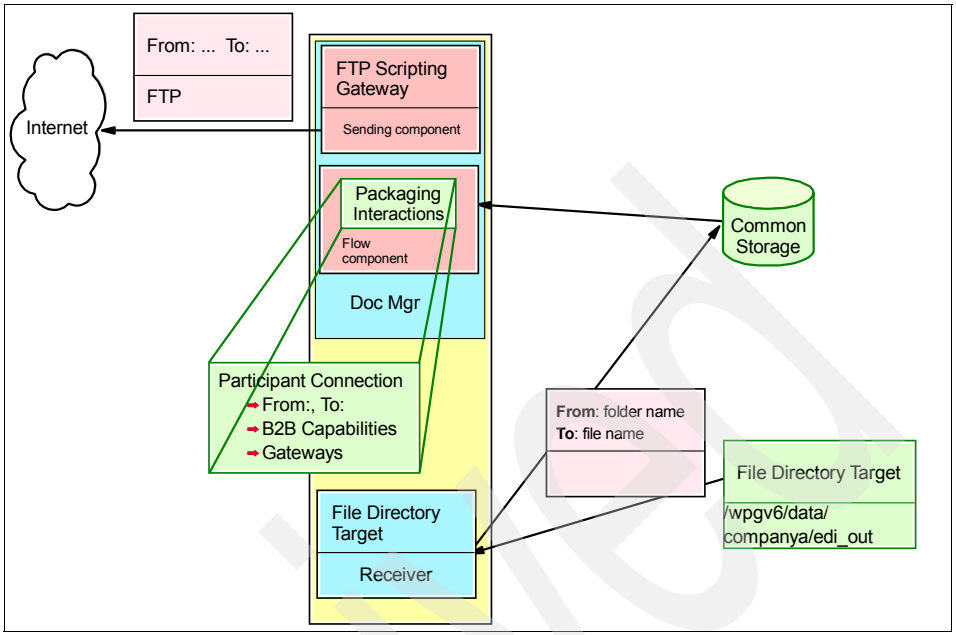


Figure 12-34 Outbound Flow with FTP Scripting gateway

12.5.2 Updating the B2B Capabilities of Company F

By now, we have updated B2B capabilities of Participant profiles several times. Here are the quick steps for enabling Company F to understand None/EDI-X12/ISA:

1. Log on to Company A Community console as hubadmin.
2. On the Participant Search screen, click **Search**.
3. Click the **magnifying glass** icon next to the **Company F Participant** profile.
4. Now within the profile of Company F, click **B2B Capabilities**.
5. Expand the **Package: None** by clicking the **folder** icon.
6. Click the check box under **Set Source** and **Set Target** for **Protocol: EDI-X12.**
7. Next, expand the folder next to **Protocol: EDI-X12**, and click **Set Source** and **Set Target** check boxes for **Document Flow: ISA**. Our B2B capabilities screen should look like Figure 12-35 on page 387.

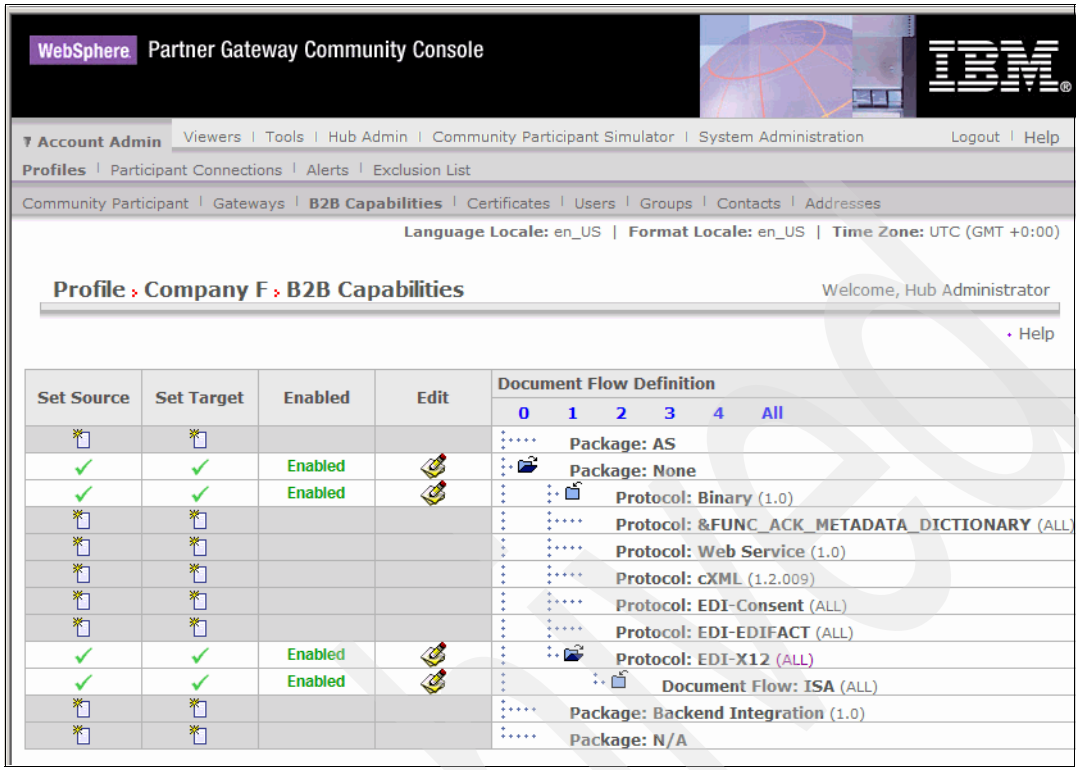


Figure 12-35 Company F new B2B capabilities

12.5.3 Creating the new interaction

Company A is enabled for None/EDI-X12/ISA from Chapter 7, “Creating a basic B2B exchange” on page 137 and we just enabled Company F for the same capabilities. To define the interaction between them, follow these steps:

1. Still logged on as the hubadmin user on the Community Console of Company A, Click **Hub Admin** → **Hub Configuration** → **Document Flow Definition**.
2. Click **Manage Interactions**.
3. From there, click **Create Interaction**.
4. On the Source side, expand **Package: None**, under that expand **Protocol: EDI-X12**, and then click the radio button next to **Document Flow: ISA**
5. On the Target side, expand **Package: None**, under that expand **Protocol: EDI-X12** and then click the radio button next to **Document Flow: ISA**.
6. For Action, select **Pass Through.**

Archived

Figure 12-36 shows the Create Interaction screen. If it is correct, click **Save**.

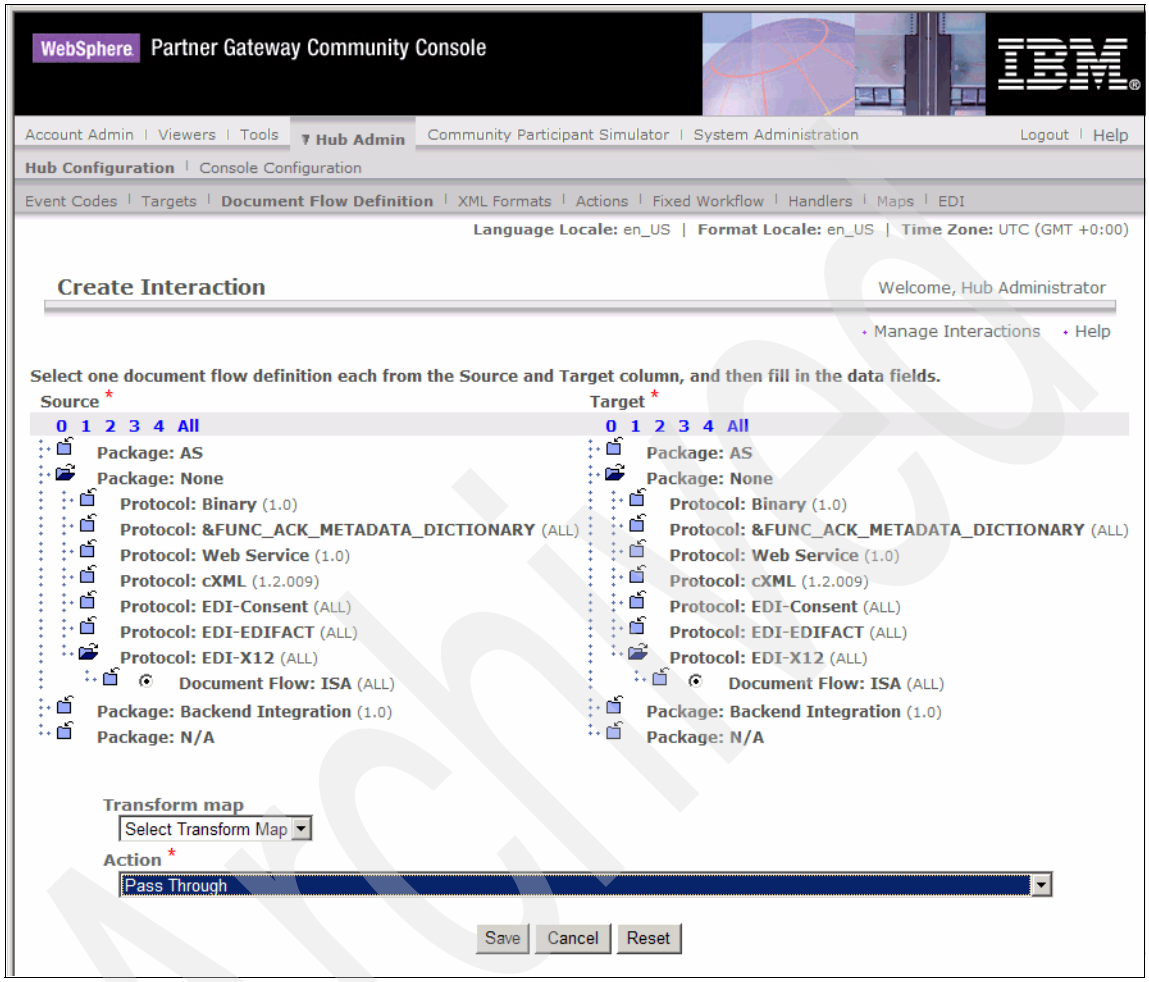


Figure 12-36 Creating the interaction for None/EDI-X12/ISA to None/EDI-X12/ISA

7. We are returned to the Manage Interactions screen, click **Search**.

   The new interaction should now be visible in the list. Your screen should look similar to Figure 12-37 on page 389.

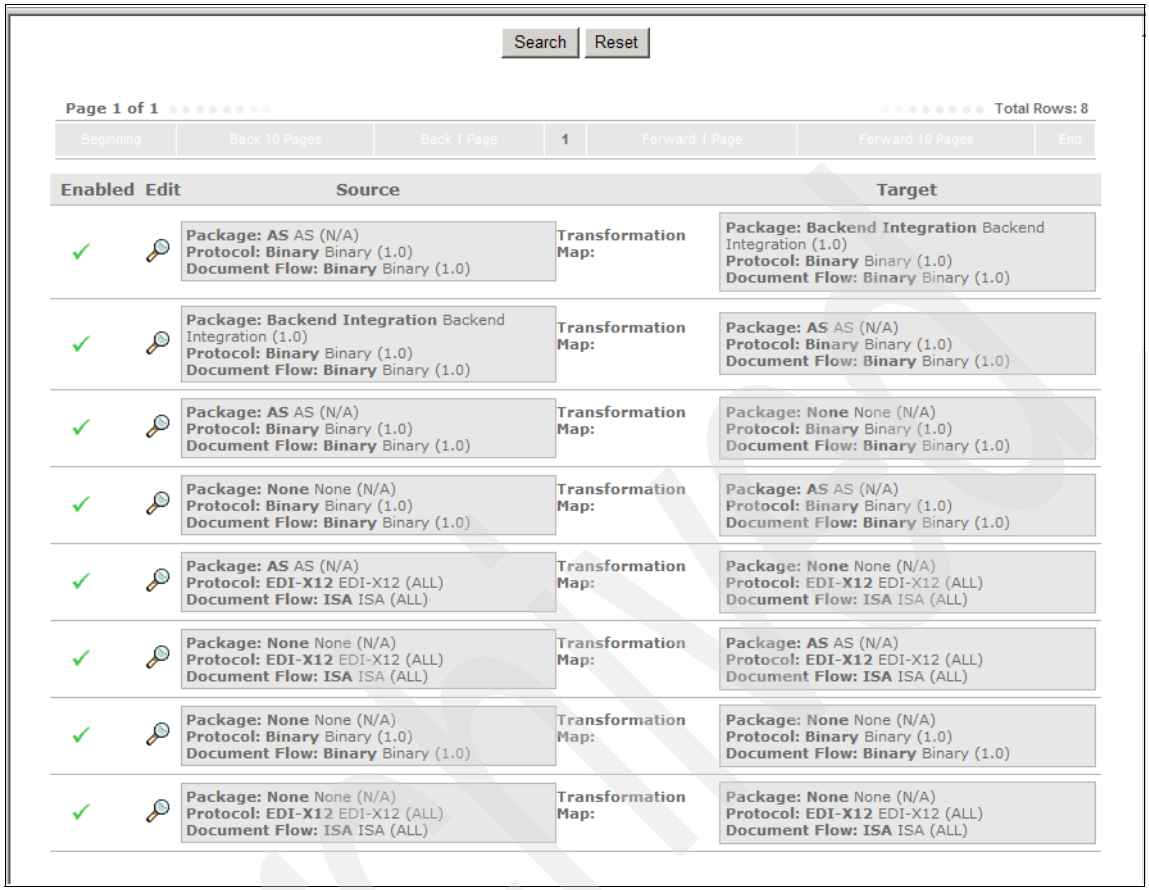


Figure 12-37 Interaction List

12.5.4 Create an FTP Scripting Gateway for Company F

So far in this chapter, we have created an FTPGateway and an FTPSGateway for Company F. In this section, we create a third, FTPScriptGateway. Because this gateway type is new and has some additional features, there are some additional configuration steps to perform first.

We have mentioned that the FTP Scripting gateway has greater flexibility with FTP communications. That greater flexibility is transferred to the user of WebSphere Partner Gateway by allowing us to use a script to define specifics of the FTP transfer. This script is executed by the WebSphere Partner Gateway server when it has a document for Company F's FTP server.

For this example, we know that Company F wants to receive the new EDI files in a directory separate from where the two earlier gateways wrote files. Those two gateways dropped all outbound files into the home directory of the `companya` user. The new goal is to put these outbound files somewhere besides this home directory.

Writing the script file

To create a script that can accomplish this simple task, follow these steps:

1. On the machine that is running the Community Console, open a Text editor, such as Notepad on Windows.
2. The script allows comments using the # character, so for the first line of the script type `# FTPScriptGateway Script File`.

> **Tip:** You can add comments using the # character at the beginning of the line to make WebSphere Partner Gateway ignore the line.

3. Next, type the first active command which must be:

   ```
   open %BCGSERVERIP% %BCGUSERID% %BCGPASSWORD%
   ```

   The **open** command is what initiates the connection to the FTP server. The next three parameters are variables that get assigned their values from the gateway definition that we will configure in a moment. By making these parameters variables we can update the gateway configuration to point to a different FTP server without having to update and reload the script file.

   Our script file currently looks like Figure 12-38.

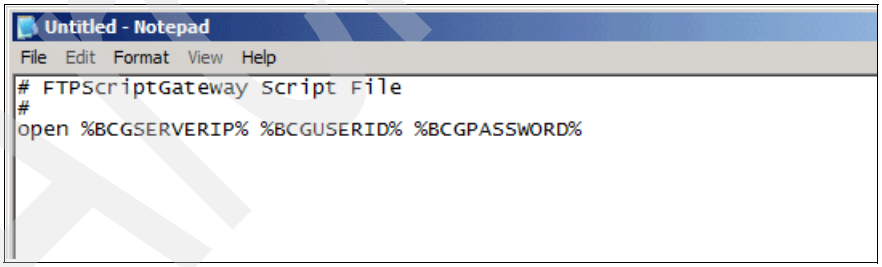


Figure 12-38 Creating script file for the gateway

4. Right now, this will connect us to the FTP server of Company F but we are in the home directory. To change to another directory, add the line:

   ```
   cd edi_in.
   ```

   This directory will need to be created for use by the FTP server administrator at Company F.

5. The delivery of the file is now handled by the script, so we must add a command to tell WebSphere Partner Gateway to deliver the file. Enter

```
mput *
```

The **mput** command is used instead of **put** because we will not know the name of the file at time of delivery. The Document Manager component renames the file with a unique identifier when it processes the document.

The Script file should look similar to what is in Figure 12-39.

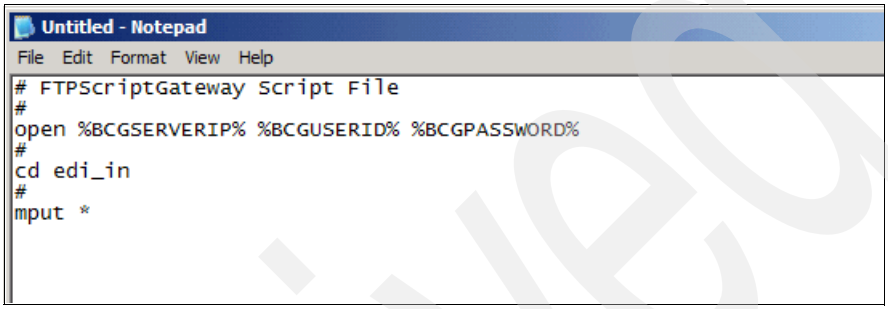


Figure 12-39 Changing directory and delivering the file in the script

6. The final command to add is: **quit**. The script controls the interaction with the FTP server, so exiting out of the connection cleanly is important.

Our final script looks like Figure 12-40.

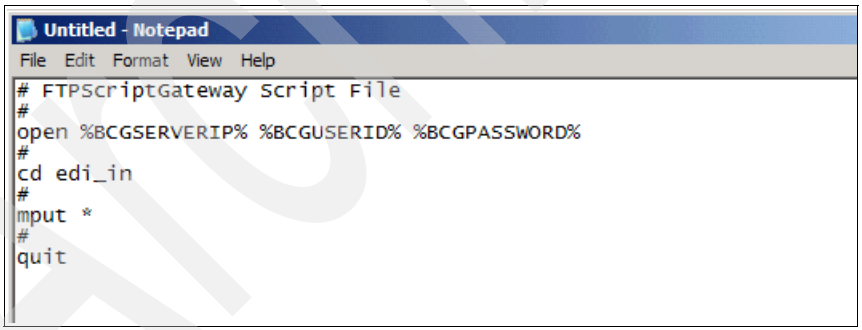


Figure 12-40 Basic script file completed

7. **Save** the file, giving it a name such as **FTPScriptFile.txt** and store it somewhere on the machine that is running the Community Console.

Note: There are other commands that can be used inside an FTP Scripting type gateway. For a full list of commands that would apply, see the Participant Guide documentation of the product.

Archived

Creating the FTP Scripting Gateway

With the script file defined, we can now create the real FTP Scripting Gateway. The steps to create it are the same as the steps other two gateways for Company F, FTPGateway and FTPSGateway.

Here are the steps:

1. As the hubadmin user, click **Account Admin** → **Profiles** → **Community Participant**.
2. Click **Search**.
3. Click the **magnifying glass** icon next to Company F to enter the profile of Company F.
4. Click **Gateways**, we should see the two previously defined gateways, FTPGateway and FTPSGateway.
5. Click **Create**.
6. On the Gateway List screen we need to enter the following information for this new gateway (Figure 12-41 on page 393):
   a. Enter a Gateway Name, `FTPScriptGateway` for example.
   b. For Transport, select **FTP Scripting**. The options will change to reflect FTP Scripting related choices.
   c. For Server IP, enter the IP Address or host name of the FTP server and Company F. In this example it is `wpgv6f`. Notice this name is not prefaced with ftp:// as the other two gateways were.
   d. Enter the User ID and Password of the account Company A has on Company F's FTP server. In this case, we use the same account, `companya`, because we are using the script to change the directory location.
   e. For FTPS Mode, select the **No** radio button.

> **Note:** For now we are leaving FTPS off, it is clear however that enabling FTPS functionality does not require a new gateway type when using FTP Scripting as it did with standard FTP. It becomes an attribute of the gateway itself. We discuss this fact in more detail in 12.7, "FTPS for FTP Scripting Gateways and Targets" on page 413.

The Configuration to this point is shown in Figure 12-41 on page 393.

Archived

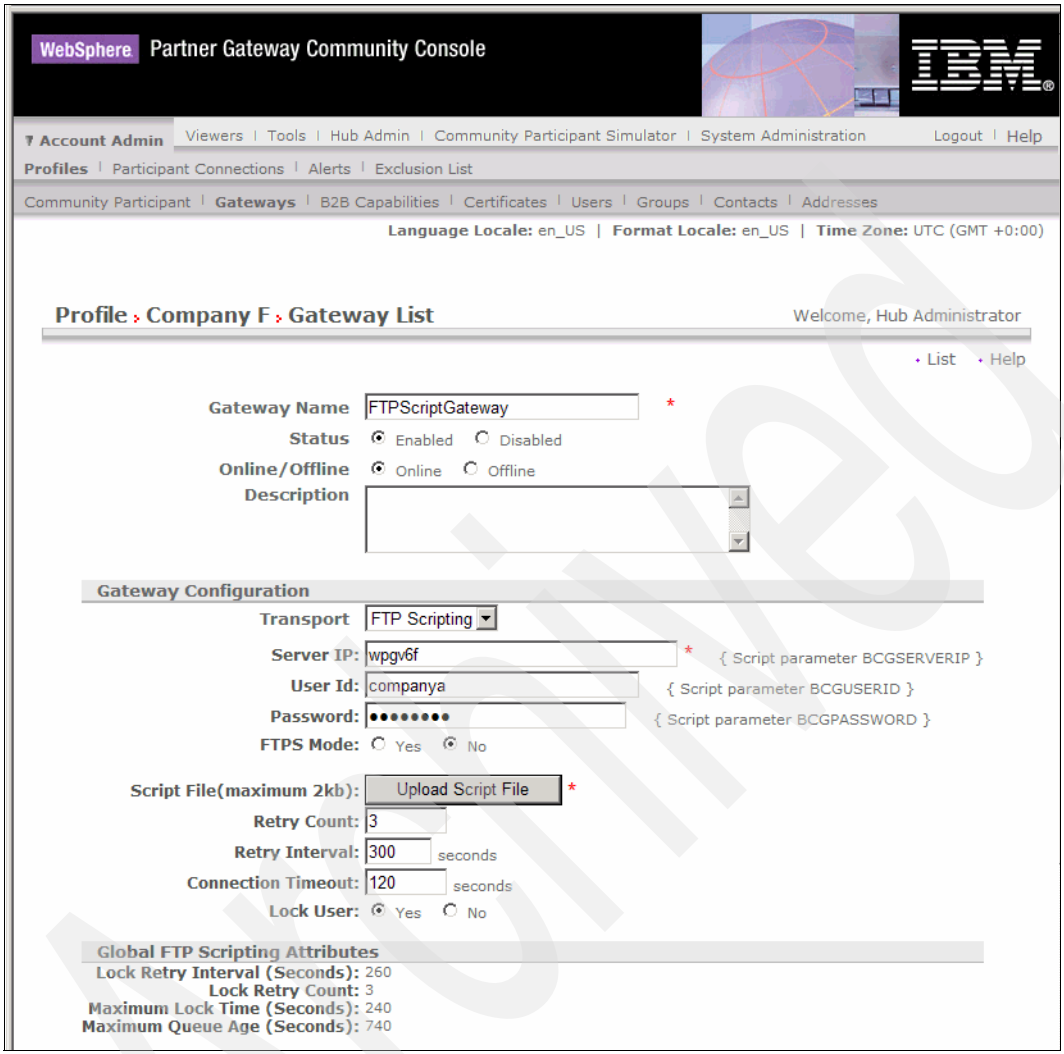


Figure 12-41 Entering the FTP Scripting gateway attributes

f. Click the **Upload Script File** button (visible in Figure 12-41). This opens a new Upload FTP Script File window. In this window, click **Browse**.

g. In the Browse Window, locate the FTPScriptFile.txt file we created earlier and click **Open**.

h. When the file name is in the Script File field, click the **Load File** button (Figure 12-42 on page 394). This load file action allows us to preview the file that we are about to assign to the gateway.

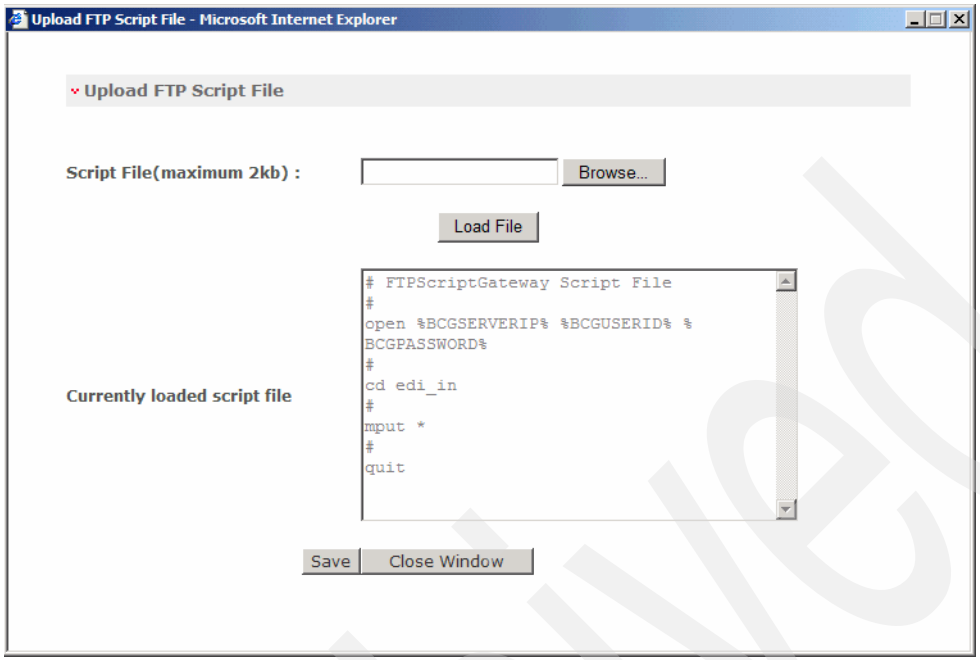


Figure 12-42 Uploading the script file

i. Click **Save** on this pop-up window to assign this file as the script to the gateway.

j. The remaining parameters can be left at the default setting. See Figure 12-43 on page 395 showing the complete FTP Scripting Gateway Configuration. Click **Save**.

Archived

Profile › Company F › Gateway List Welcome, Hub Administrator

• List • Help

Gateway Name: FTPScriptGateway *
Status: Enabled Disabled
Online/Offline: Online Offline
Description:

Gateway Configuration

Transport: FTP Scripting
Server IP: wpgv6f * { Script parameter BCGSERVERIP }
User Id: companya { Script parameter BCGUSERID }
Password: •••••••• { Script parameter BCGPASSWORD }
FTPS Mode: Yes No

Script File(maximum 2kb): Upload Script File *
Retry Count: 3
Retry Interval: 300 seconds
Connection Timeout: 120 seconds
Lock User: Yes No

Global FTP Scripting Attributes

Lock Retry Interval (Seconds): 260
Lock Retry Count: 3
Maximum Lock Time (Seconds): 240
Maximum Queue Age (Seconds): 740

User defined attributes

New

Schedule

Interval Based Scheduling : Calendar Based Scheduling :
Interval : 60.0 seconds

Handlers

Configuration Point Handlers: Select One

Save Cancel

Figure 12-43 Saving the complete FTPScriptGateway configuration

k. When the Save action completes, we see the FTPScriptGateway configuration summary. Click **List** to see the current configured gateways for Company F(Figure 12-44 on page 396).

Archived

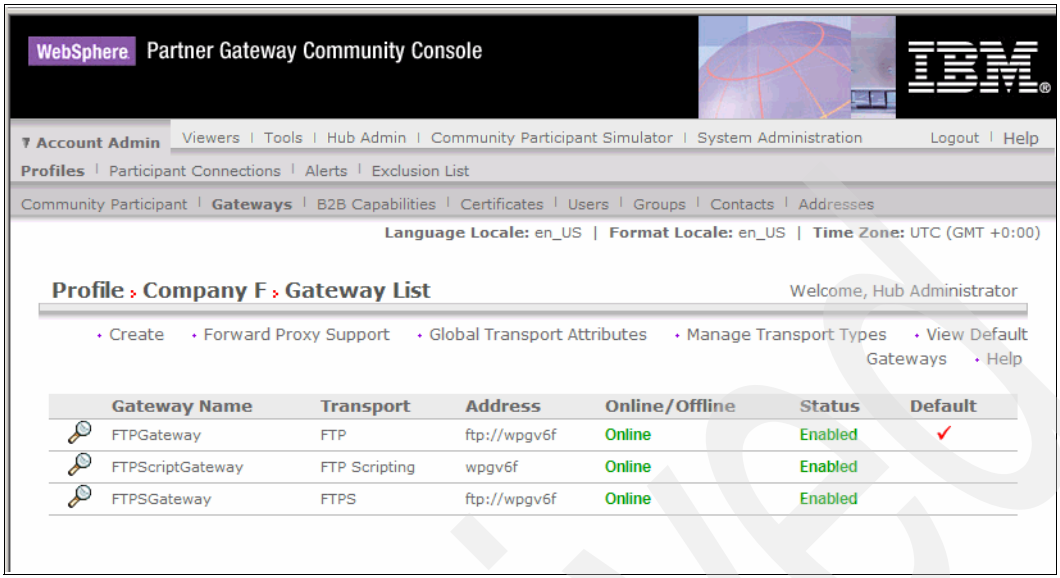


Figure 12-44 New Company F gateway list including the FTPScriptGateway

Back in Figure 12-43 on page 395, notice that several new parameters were left as default. These parameters are new and can be very useful in the correct scenarios. A brief description of their uses is as follows:

- The Lock User setting defaults to **Yes**. This parameter is used mainly when there are multiple FTP Scripting gateways accessing the same user account. This is common in a scenario where VAN services are being provided between two participants. This is covered in more detail in Chapter 13, “Enabling VAN connectivity using FTP Scripting” on page 419.
- The four Global FTP Scripting Attributes are used to tune the lock settings such as length of lock, lock expiration time, time before retry. Again these are discussed in Chapter 13, “Enabling VAN connectivity using FTP Scripting” on page 419, and are normally applicable in VAN Service connections.
- The Schedule section at the bottom of Figure 12-43 on page 395 can be important in any scenario. It allows us to configure the schedule we want to use to have documents delivered.
  - For **Interval Based Scheduling** for example, we configure WebSphere Partner Gateway to wait a certain amount of seconds before sending the documents. If this setting were 600 seconds, the server would receive and process documents normally, but it would only contact the FTP server of Company F and deliver the documents once every 10 minutes. While the documents are accumulating they display as In Process in the Document Viewer.

– **Calender Based Scheduling** works in the same manner, but allows us to configure the time and day of week that we want documents to be delivered. We could for receive documents for an entire week or month that are destined for Company F, but only deliver them at midnight on Tuesday for instance.

In most cases of normal operation, you will probably want to deliver the files after they are processed. We wanted to point out that the scheduling feature does allow you to deliver files determined by your preference and is another advantage of FTP Scripting-based gateways.

We have completed the creation of the FTP Scripting Gateway for Company F. All that is needed now is to update the participant connection to use the new interaction and gateway and validate that it works.

12.5.5 Update the participant connection

There are two things we need to accomplish when updating the participant connection:

- Activate the new None/EDI-X12/ISA to None/EDI-X12/ISA connection between Company A and Company F.
- Configure the connection to use the newly created FTPScriptGateway.

To do this, follow these steps:

1. As the hubadmin user, click **Account Admin** → **Participant Connections**.
2. Select **Company A** as the Source, and **Company F** as the Target, then click **Search**.
3. We will see some new connections available based on the updated B2B capabilities. Click **Activate** on the None/EDI-X12/ISA to None/EDI-X12/ISA channel. It is highlighted in Figure 12-45 on page 398.

Archived

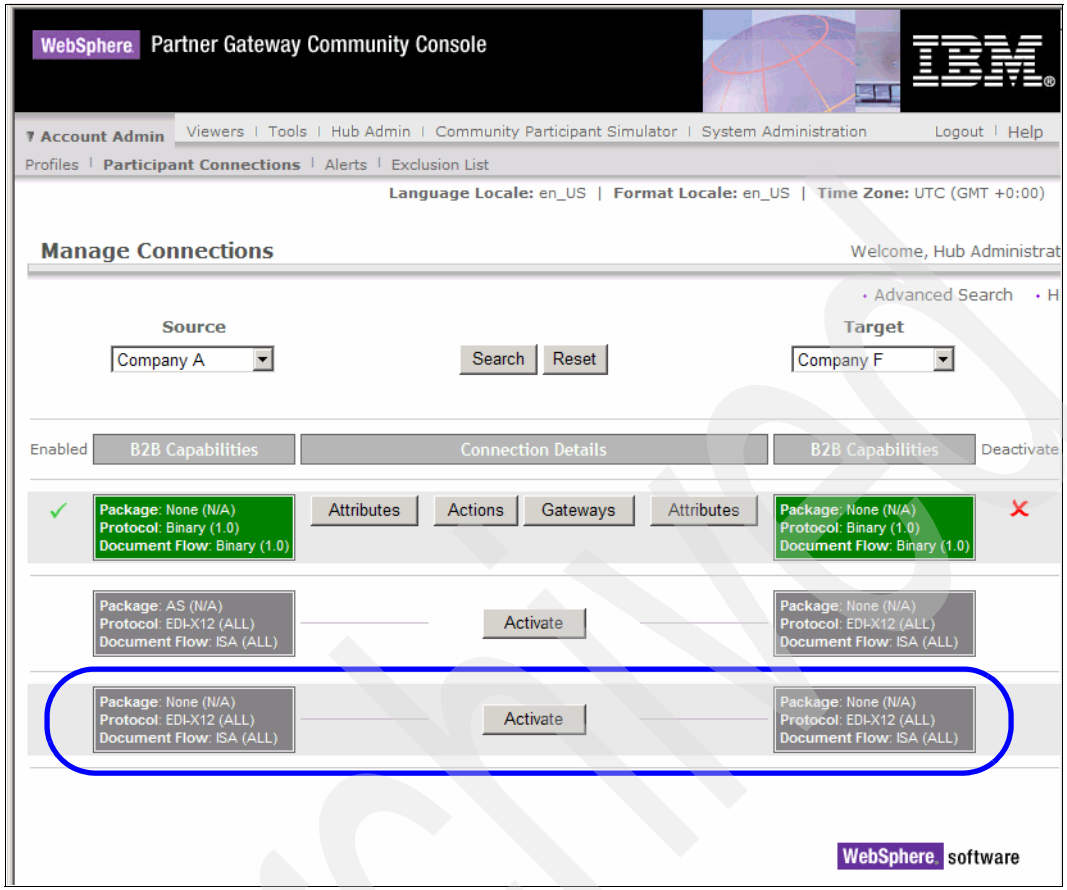


Figure 12-45 Activating the new connection

4. After the channel is activated, we need to set the update the gateways it uses. To do this, click the **Gateways** button.
5. In the Edit participant connection pop-up window, under Target Gateways, select the **FTPScriptGateway** (Figure 12-46 on page 399). Then click **Save**.

Archived

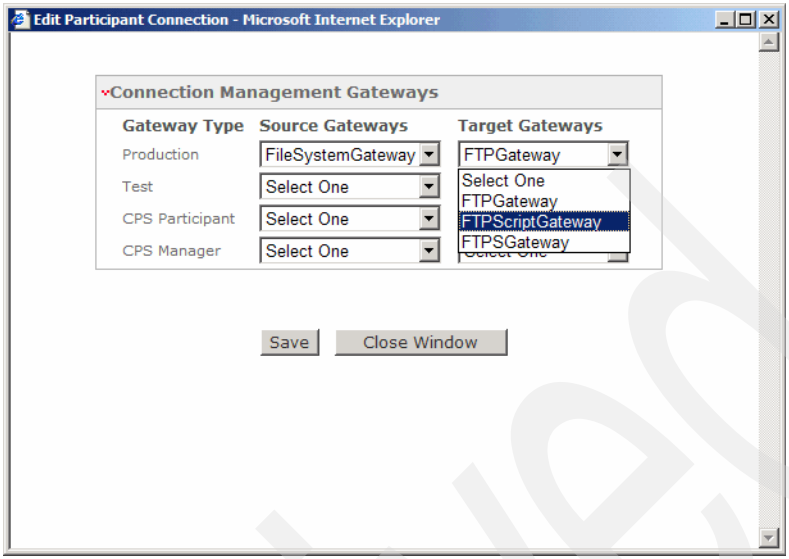


Figure 12-46 Configuring the connection with the new gateway

We now have a complete configuration of an outbound connection from Company A to Company F using the FTP Scripting Gateway.

12.5.6 Validating the connection

To validate this new connection, we need to have an EDI document with the correct business identifiers present in the document. The File Directory target picks up the file from local file system of Company A (the /edi_out directory). The Receiver component passes that file to the Document Manager for processing. There is no packaging on the document so the Document Manager sends the file to the FTP Scripting Gateway. When WebSphere Partner Gateway initiates the connection it uses **`open %BCGSERVERIP% %BCGUSERID% %BCGPASSWORD%`** and replaces the variables with what was configured in the FTPScriptGateway. When a connection is established, it will change directory to the edi_in directory on the FTP server of Company F using the command entered when we created the script. It then delivers the file using the **`mput *`** command. It closes the open connection with the **`quit`** command given in the script as well.

A sample EDI file with the correct business identifiers is shown in Example 12-12.

Example 12-12 Sample EDI document set for Company A to Company F

```
ISA*  *          *  *          *  *companya      *  *companyf
*050722*1517* *     *000000004* *P*:!
```

```
GS*PO*  *  *20050722*1517*4* *004010!
ST*850*0004!
BEG*06*SA*P345322**20050722!
N1*18*Ronan Dalton!
N2*Deliver To:*IBM Internationl Holdings Ireland!
N3*Damastown Industrial Estate!
N4*Dublin!
PO1**128*EA*10**BP*00021P0241!
PID*M**01*F*SCSI Card!
PO1**100*EA*10**BP*00031P0241!
PID*M**01*F*IDE Card!
PO1**128*EA*10**BP*00087P6821!
PID*M**01*F*PCI Card!
PO1**128*EA*10**BP*00021P0241!
PID*M**01*F*Deck of Cards!
SE*15*0004!
GE*1*4!
IEA*1*000000004!
```

When the sample EDI file in Example 12-12 on page 399 is placed in the /home/bcguser/wpgv6/data/companya/edi_out/Documents/Production directory, it is picked up and begins the transaction.

To view the document ,use the Document Viewer function of the Community Console of Company A. The result looks like Figure 12-47 on page 401.

Archived

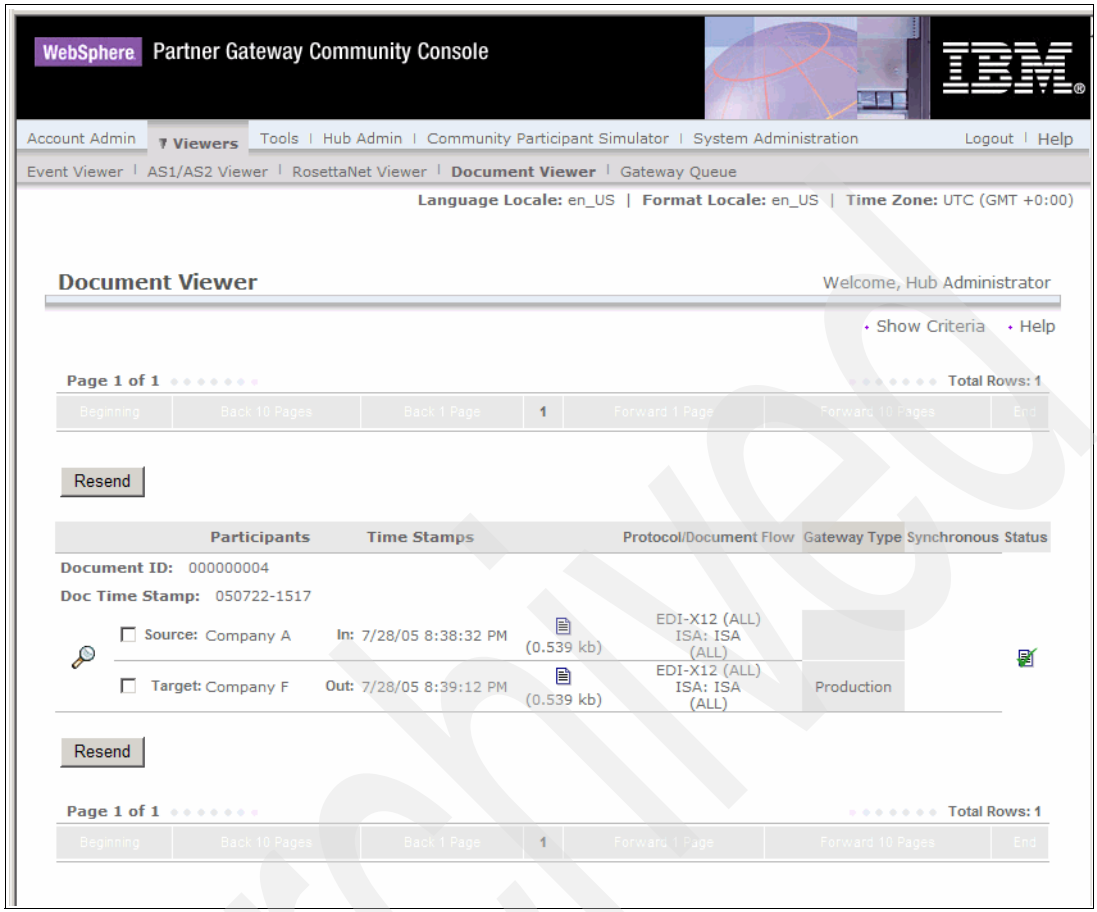


Figure 12-47 Successful document sent from Company A to Company F

Note: If you traverse to the Document Viewer quickly, or send another document while you have the Document Viewer screen up, you might see the transaction in a Document is being processed status as seen in Figure 12-48 on page 402. This is a result of the 60-second Interval Based Scheduling parameter mentioned in 12.5.4, “Create an FTP Scripting Gateway for Company F” on page 389. WebSphere Partner Gateway is waiting this interval before initiating the connection to deliver the document. With a much higher setting you can see several documents in this state, waiting for the interval to trigger and send all the waiting documents with a single **`mput *`** command issued from the script.

Archived

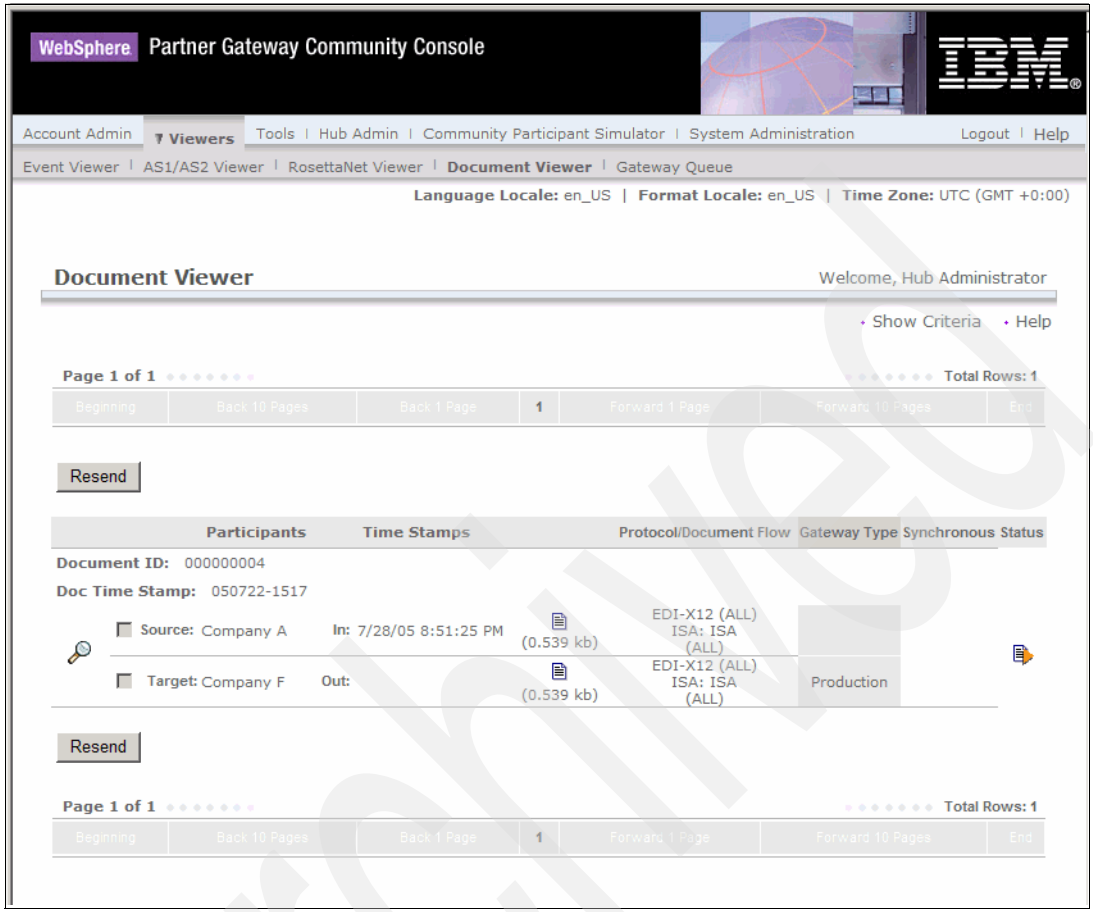


Figure 12-48 Document being processed state

The final destination of the file delivered is in the \edi_in directory under the home directory of the companya user. A quick look at that directory should show a view similar to Figure 12-49 on page 403.

> **Note:** Notice there the filename in this case is completely unique, and the original file name is not included in the file (Figure 12-49 on page 403). There is no Use Unique Filename attribute on the FTP Scripting Gateway as there is on a standard FTP Gateway.

Archived

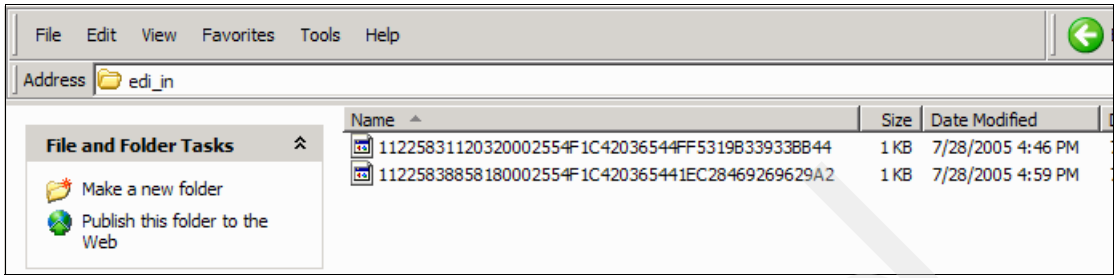


Figure 12-49 edi_in folder view at the Company F FTP server

To review, the successful transmission of a document outbound over an FTP Scripting gateway demonstrates some of the flexibility of using this type of gateway. By writing a script to handle the FTP connection, we were able initiate the connection using variable replacement, alter the destination directory, deliver the file, and close the connection yourself.

We also discussed some of the tuning parameters available to an FTP Scripting gateway, such as the delivery schedule. Another key advantage to the new FTP Scripting functionality in WebSphere Partner Gateway is discussed in 12.6, “Configuration of FTP Scripting inbound” on page 403.

12.6 Configuration of FTP Scripting inbound

The inbound flow example in 12.3, “Configuration of Company A for FTP Inbound” on page 360 requires Company A to have an FTP server located on its side of the transaction. In that example, we use the basic FTP server on the AIX installation to show the inbound flow. One of the greatest advantages of using the new FTP Scripting target functionality is that now the inbound flow can be completed *without* an FTP server present on the side of Company A.

This is done by using an FTP Scripting Target. Remember that *targets* are the entrance points to the WebSphere Partner Gateway server. However, in this special case, the FTP Scripting Target has some gateway type characteristics. Essentially, the FTP Scripting Target contacts the FTP server of Company F, using a script similar to what we configured in the outbound flow example. In this case it uses the **`mgetdel *`** command to pull the files directly from Company F into the server and remove them afterwards. If the files were left in the directory, they would be picked up and processed again during the next connection. This eliminates the need for Company A to maintain an FTP server, and it also eliminates the need for Company F to use an FTP client to deliver the files to Company A.

This is a great benefit to WebSphere Partner Gateway implementations in which a Participant only has an FTP server and no true B2B gateway product. The company that does own a WebSphere Partner Gateway can retrieve all the files that Participant has waiting for delivery by using this FTP Scripting target to access an FTP server at the remote site. They can even pull-down the files on a custom-defined schedule.

The configuration of the FTP Scripting target is similar to any other target configuration. The sole difference is the files it retrieves are physically located at a remote site.

12.6.1 FTP Scripting inbound overview

In this inbound flow example we would like to use the new FTP Scripting Target to contact the FTP server of Company F. After a connection is established, change to a directory where Company F is storing the EDI files it wishes Company A to process. The WebSphere Partner Gateway retrieves the file or files and passes them to the Document Manager. The Document Manager then stores the files to gateway local to Company A, completing the inbound flow.

A summary of what is needed to configure this scenario is as follows (Figure 12-50 on page 405):

1. Create a new FTP Scripting target that can remotely access the FTP server of Company F and retrieve the files. A script has to be created to guide the target's actions.
2. The None/EDI-X12/ISA to None/EDI-X12/ISA flow was defined in 12.5.3, "Creating the new interaction" on page 387.
3. The B2B capabilities of both Company A and Company F need to be enabled for None/EDI-X12/ISA document flows. This was done in 12.5.2, "Updating the B2B Capabilities of Company F" on page 386.
4. The participant connection between Company F and Company A needs to be activated.
5. Create a final destination in the form of a FileSystemGateway for Company A to write the files locally. This was created back in Chapter 7, "Creating a basic B2B exchange" on page 137.

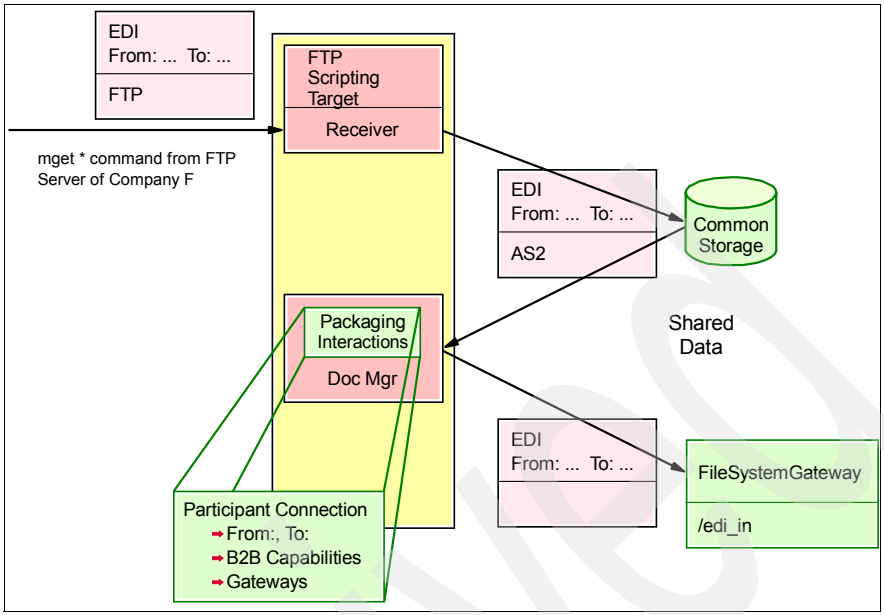


Figure 12-50 Inbound flow using FTP Scripting type target

12.6.2 Creating the FTP Scripting Target

The creation of the FTP Scripting type target is very similar to targets that have been created throughout the redbook. A script file is needed for the FTP Scripting target just as one was needed for the FTP Scripting gateway.

Continuing with our assumption that Company F wishes to exchange EDI documents, it would need to provide a new directory for Company A where it stores the EDI files that it wishes Company A to retrieve. That directory would be the equivalent of the /edi_out file directory where Company A stores its documents that it is ready to ship outbound.

Creating a script for use by the target

First, we need to create a script for use by the FTP Scripting target. The steps to this are very similar to the steps outlined in 12.5.4, “Create an FTP Scripting Gateway for Company F” on page 389. A quick review of the steps are covered here:

1. On the machine that is running the Community Console, open a text editor.
2. Create a basic text file and label it using a # character to denote a comment.
3. Enter the first active command, that makes the initial connection using variable replacement from the target configuration.

```
open %BCGSERVERIP% %BCGUSERID% %BCGPASSWORD%
```

4. Enter **cd edi_out**. This changes the directory from the home directory of the account into another directory where Company F is storing the files for delivery to Company A.
5. To actually retrieve the files, use the **mgetdel *** command. This command, issued to the FTP server of Company F, retrieves all the files and then deletes them.
6. Enter the **quit** command to close the connection.

   The script file should look similar to Example 12-13.

Example 12-13 FTP Scripting target script file

```
# FTPScriptTarget Script File
#
open %BCGSERVERIP% %BCGUSERID% %BCGPASSWORD%
#
cd edi_out
#
mgetdel *
#
quit
```

7. **Save** and close the file, storing it in a location we can access from the Community Console. In this example, we named it `FTPTargetScript.txt`.

Configuring the target

To configure an FTP Scripting Target for Company A, follow these steps:

1. Log on to the Community Console as the hubadmin user.
2. Click **Hub Admin** → **Hub Configuration** → **Targets.**
3. A list of the currently configured targets is displayed. Click **Create Target**.
4. On the Target Details screen, the attributes are set as follows:
   a. Name the target, `FTPScriptingTarget`, for example.
   b. For Transport, select **FTP Scripting** from the menu.
   c. The Server IP is the hostname or IP address of the Company F FTP server. In this case, `wpgv6f`.
   d. Enter the User ID and Password. We will continue to use the same User ID and password, `companya`, because we are again switching into a new directory during the connection.
   e. For FTPS mode, select the **No** radio button.

   The configuration should currently look like Figure 12-51 on page 407.

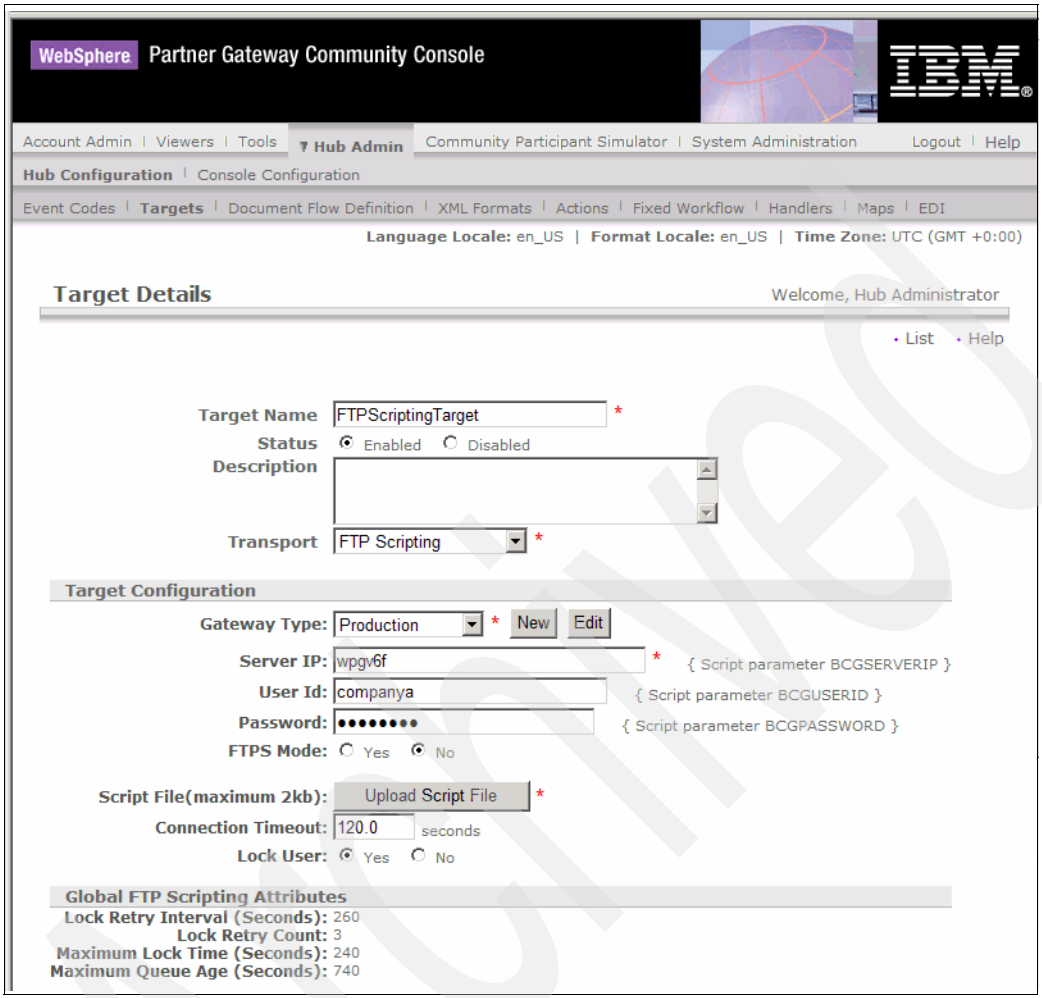


Figure 12-51 FTP Scripting target Attributes

f. Click the **Upload Script File** button (visible in Figure 12-51). This opens a new Upload FTP Script File window. In this window, Click **Browse**.

g. In the Browse Window, locate the FTPTargetScript.txt file we created earlier and Click **Open**.

h. When the file name is in the Script File field, click the **Load File** button. This load file action allows us to preview the file that we are about to assign to the gateway.

i. Click **Save**.

Archived

j. Under Schedule, change the Interval: from 60.0 to 300.0 seconds (shown in Figure 12-52) This tells WebSphere Partner Gateway to contact the Company F server every five minutes instead of every minute.

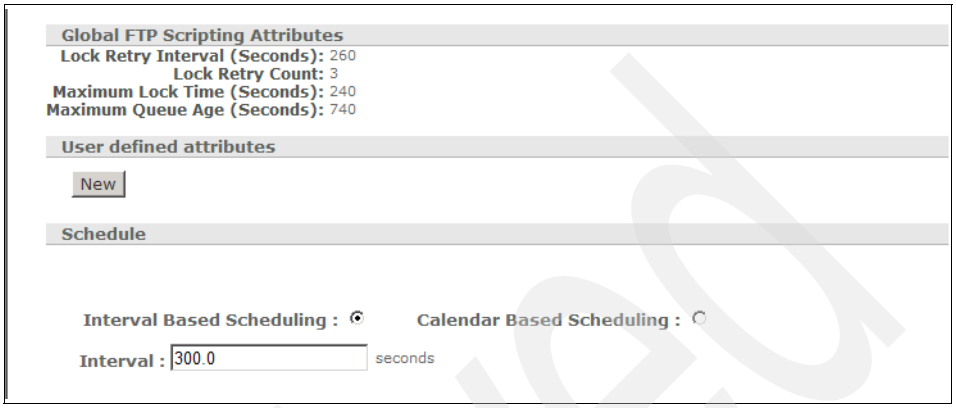


Figure 12-52 Altering the schedule of the FTP Scripting target

k. The final configuration can be seen in Figure 12-53 on page 409. Click **Save** on the Target Details screen to save this target.

Important: Clicking **Save** on this target configuration immediately starts the Interval timer WebSphere Partner Gateway uses regulate the connection made to Company F's FTP server. So 300.0 seconds after this action, it will attempt to connect to the FTP server. If there are no files ready to be picked up there will not be a problem, but you might want to confirm with the Participant you are connecting to that they are ready to begin accepting connections on your configured schedule.

Archived

Target Details

Welcome, Hub Administrator

List Help

Target Name: FTPScriptingTarget *

Status: Enabled Disabled

Description:

Transport: FTP Scripting *

Target Configuration

Gateway Type: Production * New Edit

Server IP: wpgv6f * { Script parameter BCGSERVERIP }

User Id: companya { Script parameter BCGUSERID }

Password: •••••••• { Script parameter BCGPASSWORD }

FTPS Mode: Yes No

Script File(maximum 2kb): Upload Script File *

Connection Timeout: 120.0 seconds

Lock User: Yes No

Global FTP Scripting Attributes

Lock Retry Interval (Seconds): 260

Lock Retry Count: 3

Maximum Lock Time (Seconds): 240

Maximum Queue Age (Seconds): 740

User defined attributes

New

Schedule

Interval Based Scheduling : Calendar Based Scheduling :

Interval : 300.0 seconds

Handlers

Configuration Point Handlers: Select One

Figure 12-53 Final FTP Scripting target configuration

l. Click **List** to return to the list of configured targets (Figure 12-54 on page 410).

Archived

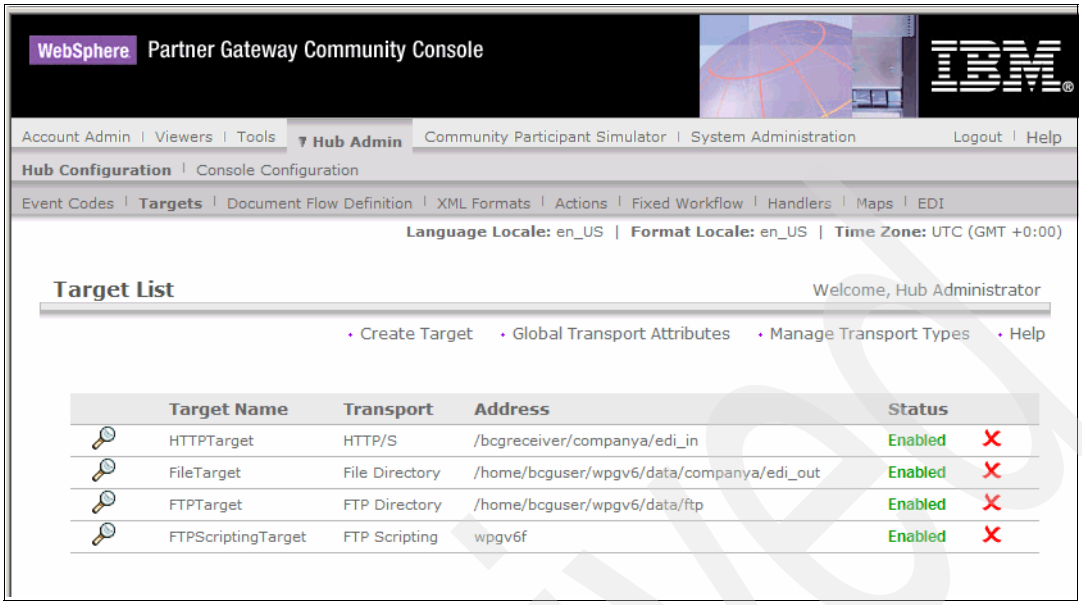


Figure 12-54 List of targets for Company A

The new target is now configured. We now need to update the participant connection and define it to use this new target.

12.6.3 Updating the participant connection

To this point, most of the configuration has been done in the previous examples in the chapter. To update the participant connection for this flow, inbound using the FTP Scripting type target, follow these steps:

1. As the hubadmin user, click Account Admin → participant connections.
2. Select **Company F** as the source and **Company A** as the target. Remember that this an inbound flow *to* Company A, even though Company A is essentially going to the remote site itself to pick up the documents.
3. Click **Activate** for the None/EDI-X12/ISA to None/EDI-X12/ISA flow (Figure 12-55 on page 411).

Figure 12-55 Activating the inbound FTP Scripting flow

4. Click **Gateways** (highlighted in Figure 12-55) on the new connection.

> **Note:** The resulting pop-up window is shown in Figure 12-56 on page 412. In the special case of an FTP Scripting Target, this is a confusing view. The Source Gateway is listed as FTPGateway, which is the default gateway on Company F. However, because Company A is picking up the file through the new target, the Source Gateway is inconsequential. The Target Gateways setting remains important, but is already set to the correct location, namely the local FileSystemGateway.
>
> The end result is Company A picks up the files from Company F, then pulls them into the Document Manager using the FTPScriptingTarget. It then processes them, and stores them in it local file system.

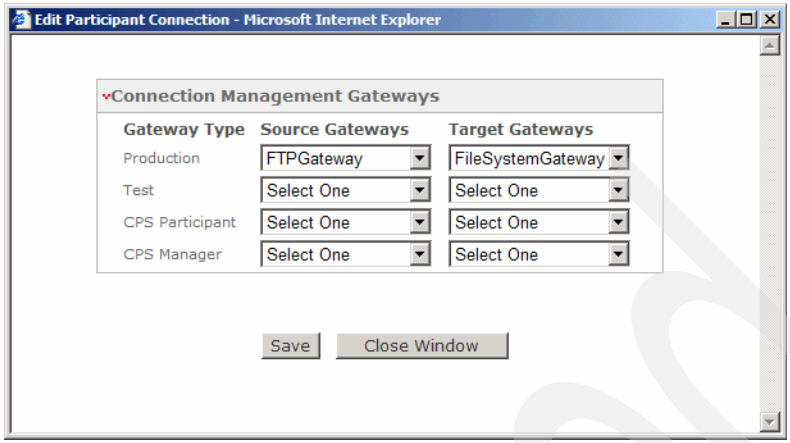


Figure 12-56 Gateways definition for inbound flow participant connection

This completes the update of the participant connection.

12.6.4 Validating inbound communication with FTP Scripting target

As was noted in 12.6.2, “Creating the FTP Scripting Target” on page 405, the FTPScriptingTarget began contacting the FTP server of Company F 300.0 seconds after the configuration was saved. WebSphere Partner Gateway has probably contacted the Participant FTP server several times in the intervening time. The easiest way to validate the example scenario, is to place an EDI file with the correct business identifiers (Company F as the source, Company A as the target) and put it into the `edi_out` directory on the FTP server. The directory is located immediately under the home directory of the `companya` user that we have been using throughout this chapter.

The Document Viewer will again be our main interface point to monitor the transmissions. Keep in mind that the documents will only be received at the interval that is defined on the schedule, so there might be a delay before we see the document appear in the Document Viewer.

Archived

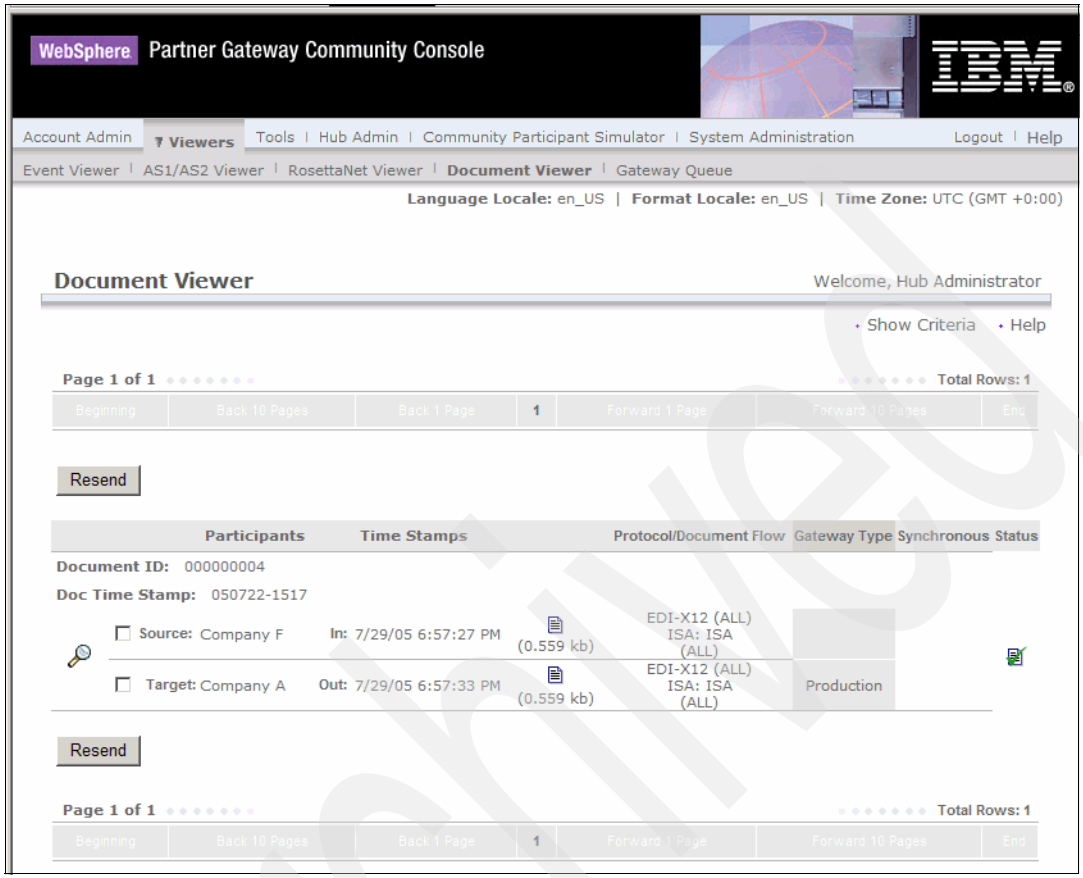


Figure 12-57 Successful FTP Scripting Inbound flow shown in Document Viewer

12.7 FTPS for FTP Scripting Gateways and Targets

In 12.4, “Implementing FTPS” on page 371 we discussed securing the outbound and inbound flows when using the first method of FTP communication. For the outbound flow a new gateway (FTPSGateway) needed to be created. For the inbound flow, WebSphere Partner Gateway did not need any configuration because the SSL handshake and transmission was solely dependent on the FTPS server implementation at the site of Company A.

The FTP Scripting functionality of WebSphere Partner Gateway has a slightly different implementation of FTPS. There are two cases to consider:

- FTP Scripting Gateway for outbound flow
- FTP Scripting Target for inbound flow

12.7.1 FTPS for the FTP Scripting Gateway

As noted in 12.5.4, “Create an FTP Scripting Gateway for Company F” on page 389, the FTP Scripting Gateway has a radio button parameter that enables or disables FTPS. This is an improvement over the older FTPGateway where an entirely new gateway of transport FTPS was needed. For the newer FTP Scripting Gateway, FTPS can be turned into a simple edit of the gateway to enable FTPS (see Figure 12-58 on page 415). However, a root certificate for Company F is still needed under Hub Operator Certificates. Also, if client authentication is used, as discussed in 12.4.2, “FTPS client authentication” on page 383, an SSL certificate will have to be loaded as well.

Archived

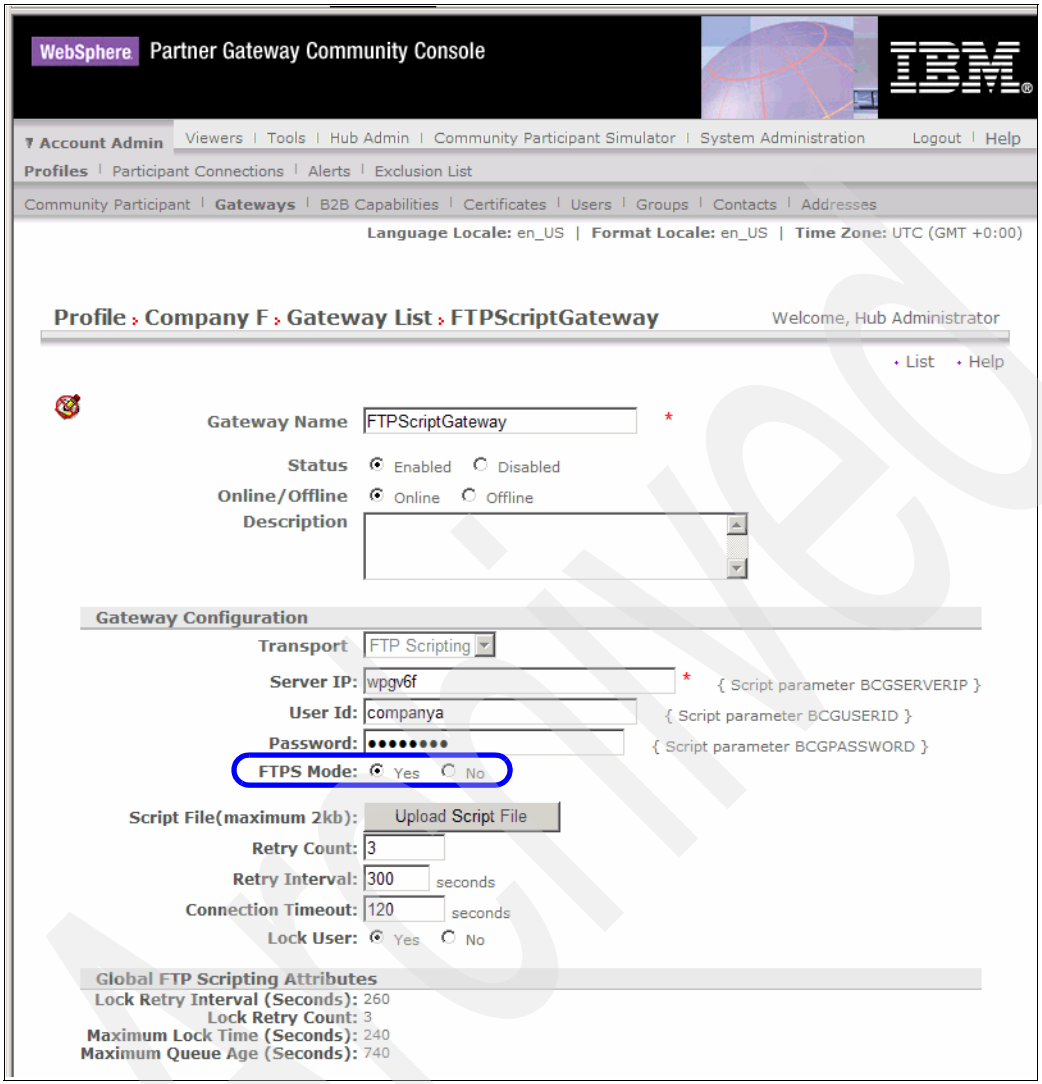


Figure 12-58 FTP Scripting Gateway update for FTPS

12.7.2 FTPS for the FTP Scripting Target

For the inbound flow using FTP Scripting, there is no longer an FTP server residing at the site of Company A, so the FTPS implementation must have changed. To enable FTPS for the FTP Scripting Target, it is again done with a radio button parameter in the FTP Scripting Target configuration screen (see Figure 12-59 on page 416). The certificates when using server or client

authentication will also need to be loaded correctly under the Hub Operator profile.

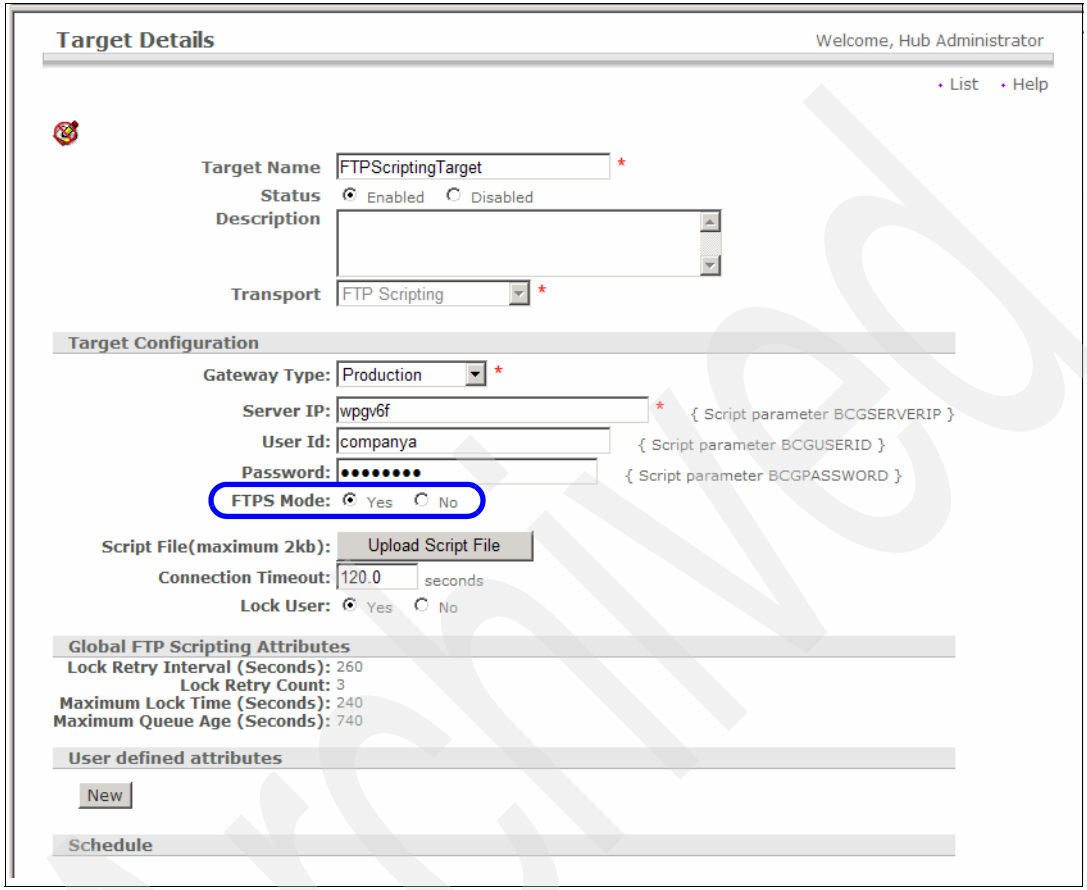


Figure 12-59 FTP scripting target update for FTPS

Securing FTP communication when FTP Scripting gateways or targets are in use is a matter of setting the radio button parameter for FTPS to **On**, and having the correct certificates loaded under the Hub Operator profile. This means:

- The public certificate of the FTPS server we are contacting has to be loaded as a Root and Intermediate certificate when implementing server authentication.
- A private p12 type certificate is loaded and set as an SSL certificate, when utilizing client authentication.

The easier configuration steps to upgrade to FTPS communication when using the FTP Scripting functionality of WebSphere Partner Gateway Version 6, are

another of the benefits to using this method. We highlight the similarities and differences of the two methods of general FTP communication in 12.8, “Comparing FTP and FTP Scripting” on page 417.

12.8 Comparing FTP and FTP Scripting

There are two distinct methods of configuring FTP communication between Company A (using a WebSphere Partner Gateway implementation) and Company F (using an FTP only), discussed in this chapter.

The first method uses an FTP Directory target to pickup files locally, and passes those files to FTP gateway that delivers them to Participant FTP server. To receive files it uses the same FTP Directory target to pull the files in after the Participant had delivered them to a local FTP server.

The second method uses the new FTP Scripting functionality. The outbound flow picks up a file locally, and delivers it to the Participant FTP server through the use of a user-defined script. The inbound files are received remotely, also using a user-defined script on an FTP Scripting target.

Throughout the chapter, several key differences are highlighted between the two methods. A real-world scenario *could* involve using both methods simultaneously, but it is more likely that one method would be used consistently throughout the configuration. A more detailed summary focusing on the differences may help you make a decision on how FTP communication would best be configured in your environment.

For the first method (FTP gateway and target, FTPS gateway) we determined that:

- We could send and receive binary files, made possible by the special directory structure of the FTP Directory target.
- FTP server functionality was required on both sides, which can be preferable when each participant company already has a FTP server or wants to maintain its own server.
- Little or no FTP knowledge is needed, other than the user ID and password of the account.
- A new gateway is needed to secure outbound transmissions.

For the second method (FTP Scripting gateway and target) we learned:

- Only a single FTP server was needed in a two-way communication, because the FTP Scripting target can pickup remote files.

- Because it lacks a special directory structure, it cannot receive binary files, although it possible to deliver them outbound.
- The FTP communication is user-defined and therefore more flexible. You can switch directories, use variable replacement, and use special FTP commands.
- Securing the connection with FTPS is possible through simple edit of the gateway or target, a new object is not required.

FTP Scripting functionality in WebSphere Partner Gateway has one final advantage over the older FTP method. Due its greater flexibility, FTP Scripting is able to accommodate a Value Added Network (VAN) scenario where a third-party service provider resides between the two companies. This scenario is detailed in Chapter 13, “Enabling VAN connectivity using FTP Scripting” on page 419.

Archived

13

Enabling VAN connectivity using FTP Scripting

This chapter discusses another extension of the B2B solution for Company E. Company V and Company E want to establish a B2B exchange. However, Company V is not ready to implement AS2 or communicate over the Internet. Company V uses its own Value Added Network (VAN) to send Electronic Data Interchange (EDI) documents outbound to its partners and also to receive EDI documents inbound from its partners.

WebSphere Partner Gateway provides an FTP Scripting solution that can communicate with companies that are on a VAN. The FTP Scripting solution uses script files to execute a batch of proprietary FTP commands, which can be tailored to enable communication with VANs. Using this solution, Company E can now send and receive documents from Company V, and Company V can use its own private VAN for communication.

© Copyright IBM Corp. 2005. All rights reserved.

13.1 Scenario overview

In a typical B2B environment, it is common that not all business partners can be aligned along a single communication mechanism. While AS2-based solutions are common and not that difficult to implement, there are always situations and many reasons where a trading partner cannot implement an AS2 solution. However, a B2B software product, such as WebSphere Partner Gateway, can handle these situations.

This chapter considers the situation where Company E wants to extend its B2B infrastructure to include a document exchange with Company V. However, Company V is reluctant change its infrastructure.

Figure 13-1 on page 421 shows Company V availing VAN services offered by a certain EDI services company, this EDI services company enables Company V to communicate with other companies which use their own private VANs, similar to Company V. The EDI services company uses an FTP Gateway in front of it and, hence, uses FTP to communicate with the external world. The EDI interconnect services company provides mailboxes to all its users, so when one trading partner communicates with another through the EDI services, it actually would be connecting to the respective mailboxes on the EDI services company, to read off the file or to place a file. The EDI services company itself would then be routing the files to appropriate trading partners on their respective private VANs.

WebSphere Partner Gateway has the ability to execute script files that consist of FTP commands. These commands need not necessarily be pure FTP commands. They can consist of special FTP commands that can only be understood by a VAN interconnect. This is the solution that would facilitate communication between Company E and Company V.

Figure 13-1 on page 421 shows the overview of the scenario where we have implemented a B2B solution for Company E with Company V. We show an FTP Scripting solution for Company E and Company V. This solution requires an FTP server support for the EDI services company acting as an intermediary, facilitating the communication of Company V with the outside world. It is worthwhile to notice here that, with this solution neither Company V nor Company E needs to have an FTP server. This solution uses the services offered by the EDI services company.

To exchange documents we use the **`mget`** and **`mput`** commands, in the script files, which is explained further in this chapter.

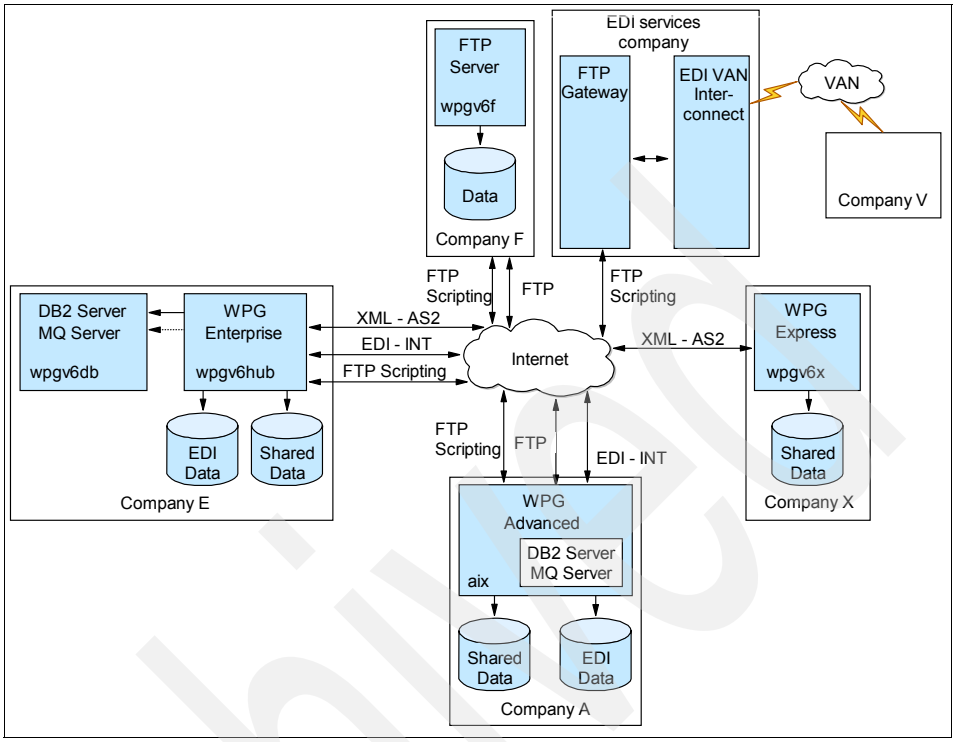


Figure 13-1 Scenario Overview

13.2 Configuration of Company E for outbound

This section explains how to configure Company E so that it can send documents outbound to a partner such as Company V, that communicates only using its private VAN. We see how WebSphere Partner Gateway can enable Company E to build a solution.

13.2.1 Implementation steps

Because VAN is mostly used for EDI communication, in our example scenario we consider sending an EDI-X12 outbound from Company E to Company V. Similarly, Company E receives an EDI-X12 inbound from Company V. This is similar to the Company A to Company F example using FTP Scripting in Chapter 12, “Integrating FTP servers” on page 335.

Figure 13-2 on page 422 summarizes the configuration points for this scenario:

1. Use the existing File Directory Target .

2. Create a partner profile for Company V with business ID companyv.
3. Update the B2B capabilities of Company V to include None/EDI-X12/ISA as source and target.
4. Update the B2B capabilities of Company E to include None/EDI-X12/ISA as source and target.
5. Create an interaction between None/EDI-X12/ISA and None/EDI-X12/ISA for the action Pass-Through.
6. Create a gateway of type FTP Scripting within the profile of Company V.
7. Create a participant connection that brings together the interaction and the partners.

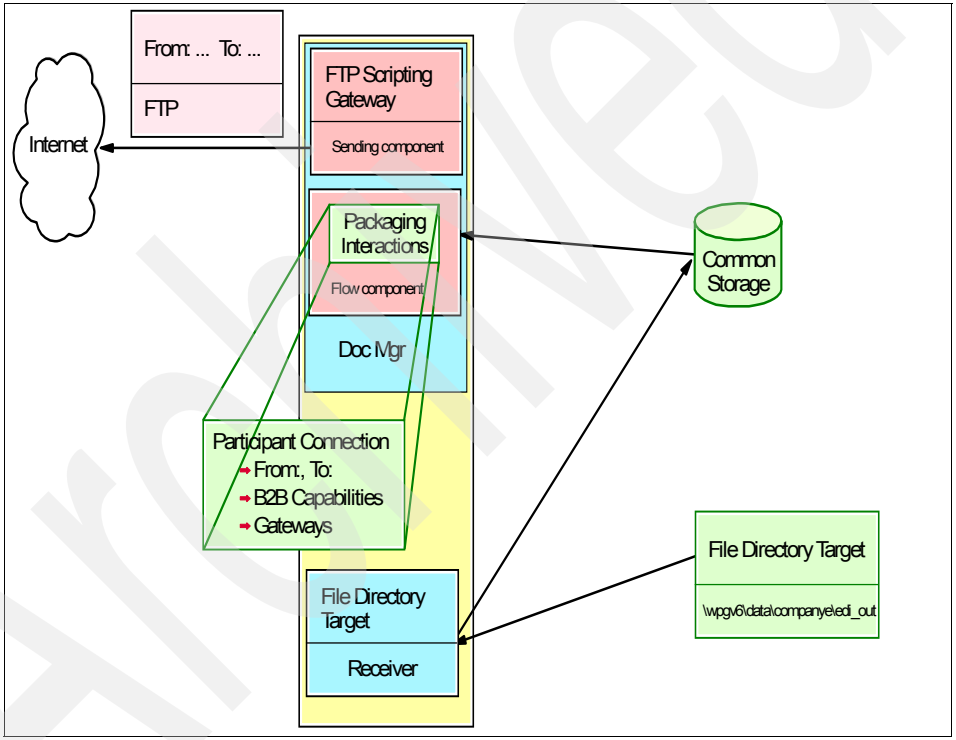


Figure 13-2 Outbound flow

13.2.2 Updating the profile of hubadmin

As the hub operator, you perform the following tasks:

1. Create a partner profile for Company V.

2. Create an interaction from unpackaged EDI-X12 (None/EDI-X12) to unpackaged EDI-X12 (None/EDI-X12).

Creating a new Community Participant

To create the new partner profile for Company V, complete the following steps:

1. Log on as hubadmin to Company E's WebSphere Partner Gateway server.
2. Select **Account Admin** → **Profiles** → **Community Participant**.
3. Click **Create**.
4. In the New Participant window (Figure 13-3 on page 424), perform the following tasks:
   a. For Participant Login Name, type `companyV`.
   b. For Participant Name, type `Company V`. This is consistent with the profiles that we created earlier for Company A and Company X .
   c. For Participant Type, select **Community Participant**.
   d. Under Business ID, for Type, select **Freeform** and for identifier, select **companyv**.
   e. Click **Save**.

Archived

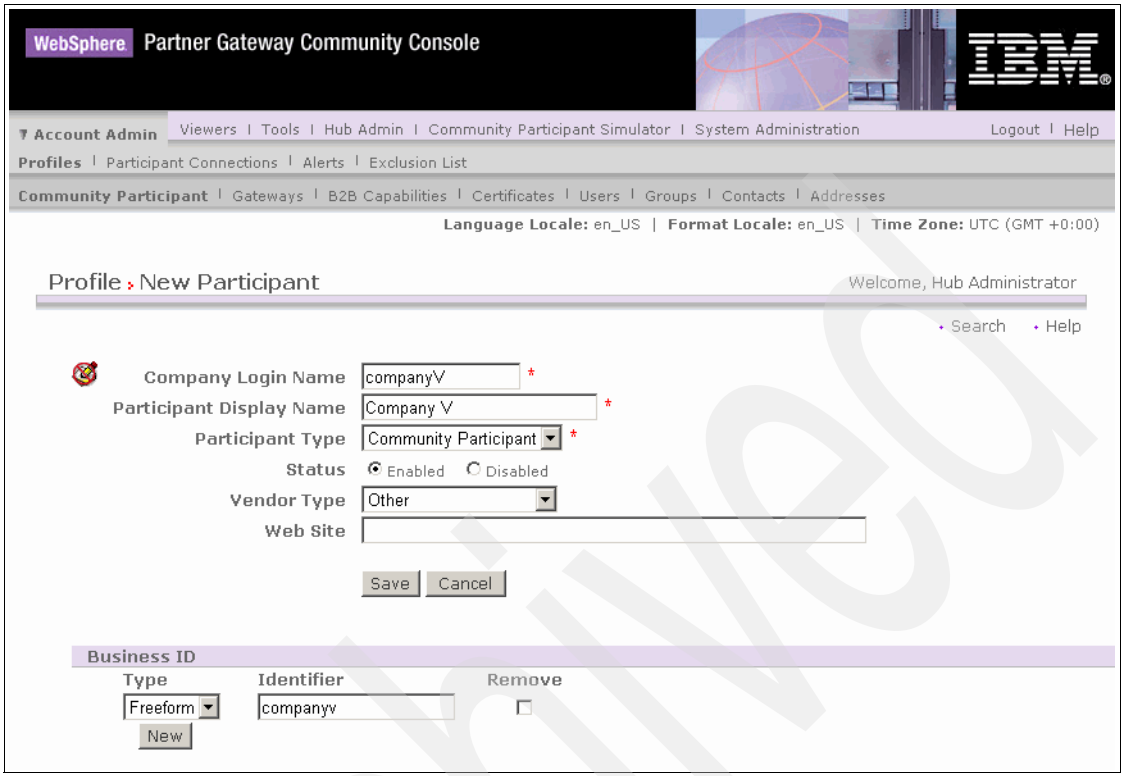


Figure 13-3 Creating a Participant profile for Company V

Creating an interaction

We can now see how we can enable a simple interaction by configuring the same in WebSphere Partner Gateway.

1. While logged in as hubadmin, select **Hub Admin** → **Hub Configuration** → **Document Flow Definitions**.
2. In the Manage Document Flow Definitions window (Figure 13-4 on page 425), verify that the combination None/EDI-X12 is enabled.

Archived

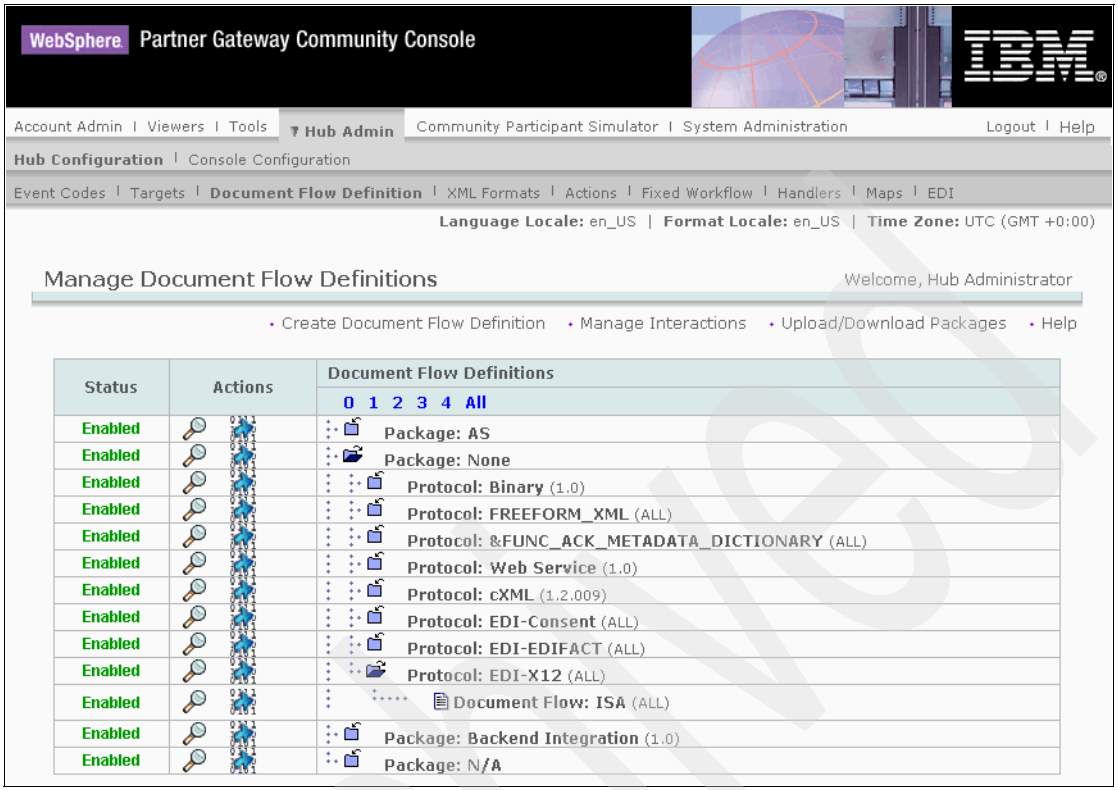


Figure 13-4 Enabled document flow definitions

3. To review the interactions, select **Manage Interactions**.
4. Click **Create Interaction**.
5. In the Manage Interactions window (Figure 13-5 on page 426), complete these steps:
   a. Expand the tree structure under the Source and Target headings.
   b. Locate **None**, **EDI-X12** and **ISA** as the source and target.
   c. For Actions, select **Pass Through**.
   d. Click **Save**.

Archived

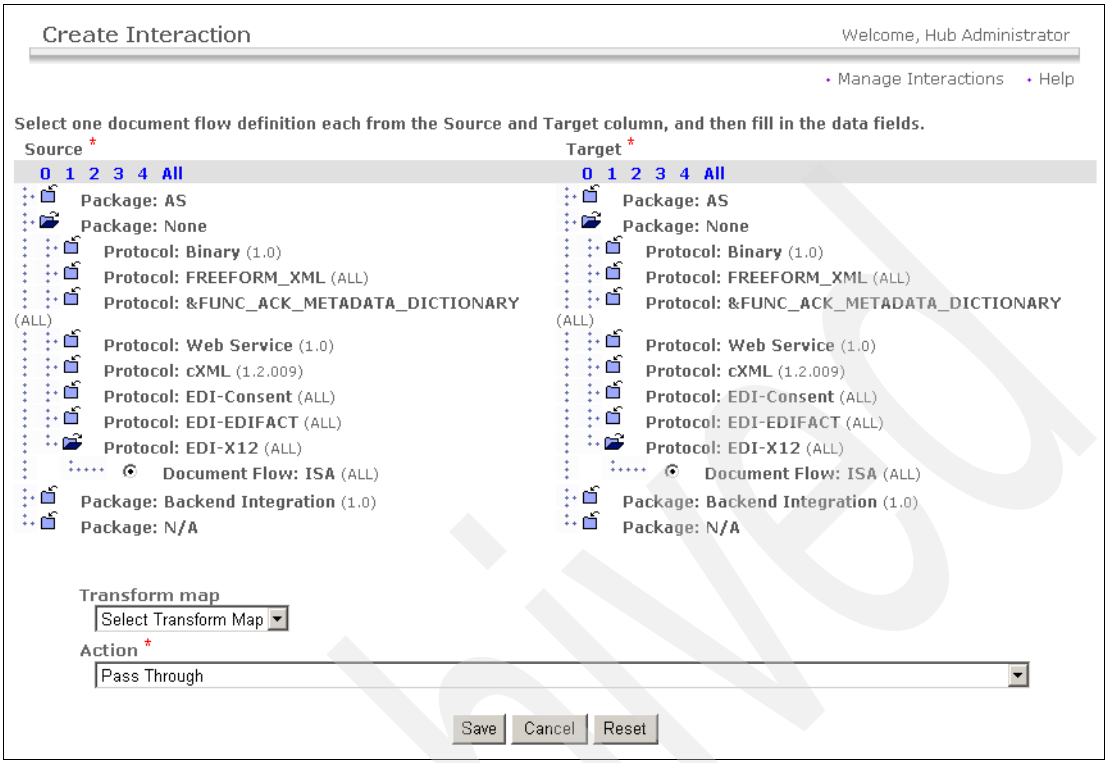


Figure 13-5 Interaction for None/EDI-X12 to None/EDI-X12

13.2.3 Updating the profile of Company V

In the previous section, we created a profile for Company V. This profile needs further configuration. Perform the following tasks:

1. Set the B2B capabilities of Company V.
2. Create the FTP Scripting gateway that points to the FTP server of the FTP gateway in the EDI services company.

Follow these steps:

1. While logged in as the hub administrator, select **Account Admin** → **Profiles** → **Community Participant**.
2. Click **Search**.
3. Locate the profile of Company V and click the **magnifying glass** icon next to it as shown in Figure 13-6 on page 427. Select **B2B capabilities** in order to enable Company V to handle EDI X12 files with no packaging.

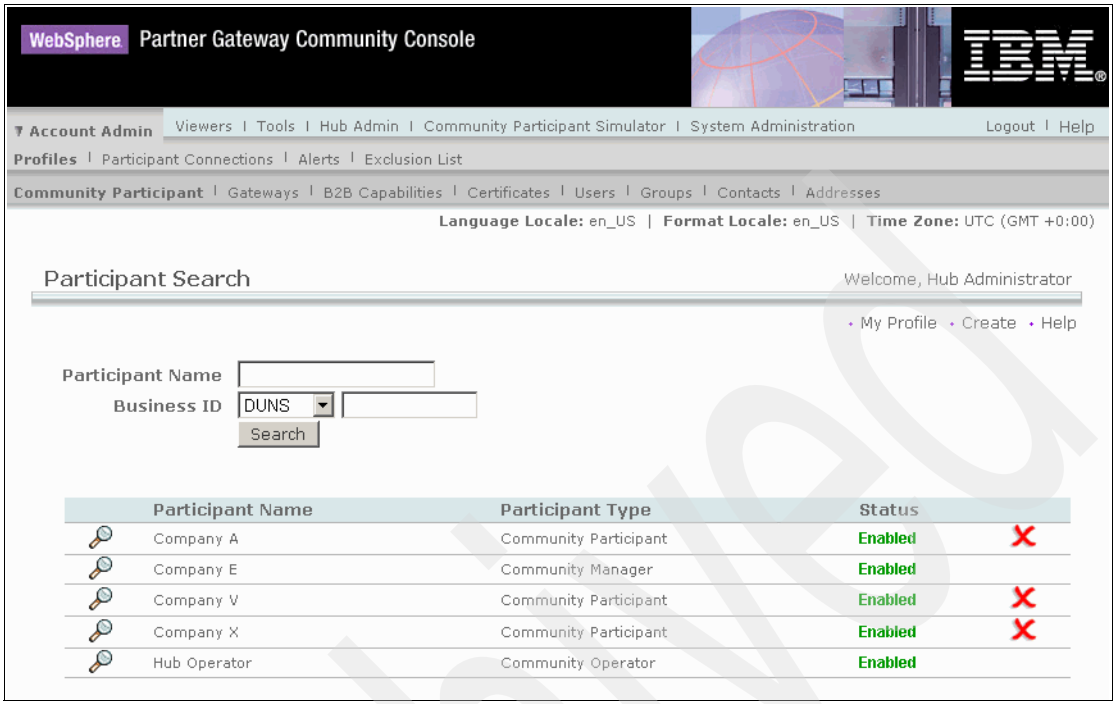


Figure 13-6 Selecting the Participant Company V

4. In the **B2B Capabilities** window, enable **EDI-X12** and **ISA** under packaging None, as the source and target.
5. Create the FTP Scripting gateway.
   a. While working with the profile of Company V, select **Gateways.**
   b. You see an empty list of gateways. Click **Create**.
   c. In the GatewayList window (shown in Figure 13-7 on page 428 and Figure 13-9 on page 430), complete these items:
      i. For Gateway Name, type `FTPScriptingGateway`, for example.
      ii. For Transport, select **FTP Scripting**.
      iii. For Server IP, type `edi.interconnect.com`, for example.
      iv. For User Id field, type the user ID for the FTP account on the server the IP of which was provided in the previous step, `accountv.companyv`, for example. For communication with VAN, the user IDs are in the specific format `account.userid`, because every trading partner registered with the EDI services will be assigned a mailbox which is specific to the

Archived

account on which it has been created. Hence, the account information is also important to enable login to the trading partner mailbox.

v. The user account is protected by a password that has to be provided in the Password field.

vi. FTPS mode radio button provides us with the choice of securing the communication. Enabling FTPS by clicking **Yes**, means that the exchange of data will be performed using Secure Socket Layer (SSL) with FTP.

For the current scenario however, we select **No**.

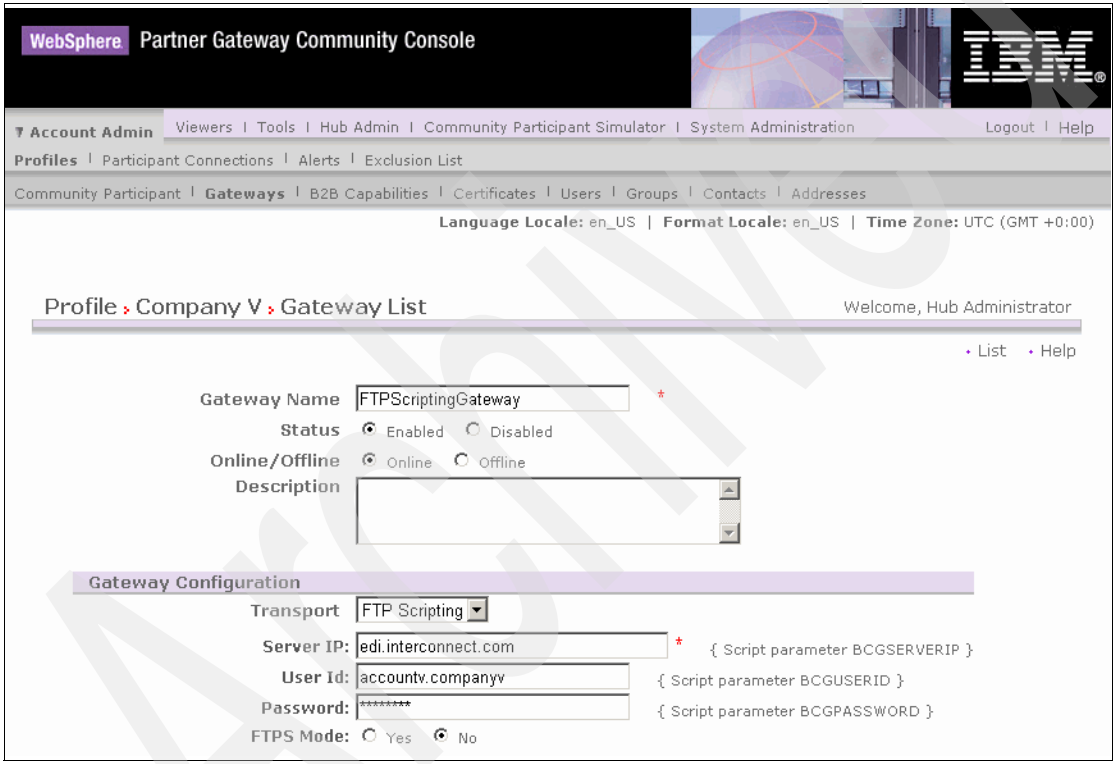


Figure 13-7 Create FTP Scripting Gateway within the profile of Company V: 1 of 2

vii. Script File is where we upload the text file with the commands that need to be interpreted by the FTP Gateway on the EDI services end, so that the desired document exchange can happen.

Click the **Upload Script File** button. Because we are configuring a gateway for outbound, the script file shown in Example 13-1 on page 429 will be loaded as shown in Figure 13-8 on page 429.

Note: The size of the script file should not exceed 2 KB, to avoid truncation when it persists in WebSphere Partner Gateway.

Everything between the %% tags (Example 13-1) in the script file is substituted at run time by the values provided during the configuration of the FTP Scripting gateway.

Example 13-1 Sample script file for outbound

```
open %BCGSERVERIP% %BCGUSERID% %BCGPASSWORD%
cd %BCGOPTION1%
mput *
quit
```

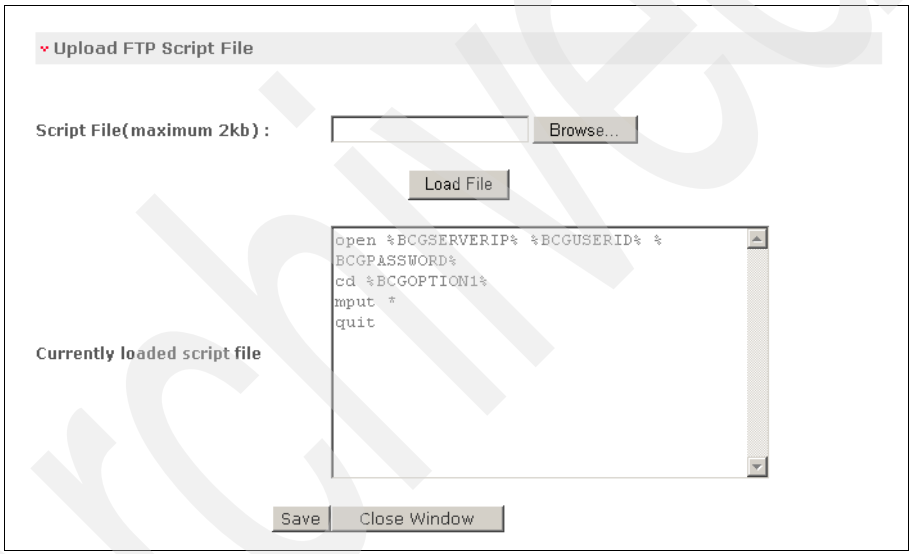


Figure 13-8 Uploading the script file

Note: Use the **cd** command in the script file for FTP scripting outbound when communicating with a VAN. Why? The argument of the **cd** command is the destination where the file has to be dropped.

The format for the command is: **cd** `[destination][/class]`. The `destination` could be accountid.userid of the trading partner local to EDI services company, for example. The `class` specifies the classification of the message. This command is important for communication to VAN because all the messages in the mailboxes are classified.

Archived

viii. The Retry Count indicates the number of times retries have to be attempted in case of a failure.

ix. Retry Interval specifies the time interval between each of the retries. We can use the default value here.

x. Connection timeout is the duration in seconds during which an FTP Socket is open in case of inactivity. We can use the default value here.

xi. Lock User attribute controls the locking of user accounts, to prevent other users from logging into an account when the owner of that account is already logged in. We can use the default option here.

xii. In addition to these attributes, we have User defined attributes, wherein users can define and use their own attributes. These are in the script file as substitute values for arguments between the %% tags . In the script file shown in Example 13-1 on page 429, `BCGOPTION1` is one such user defined attribute. This will be replaced by the value that is provided against the user defined attribute Option 1, shown in Figure 13-9.

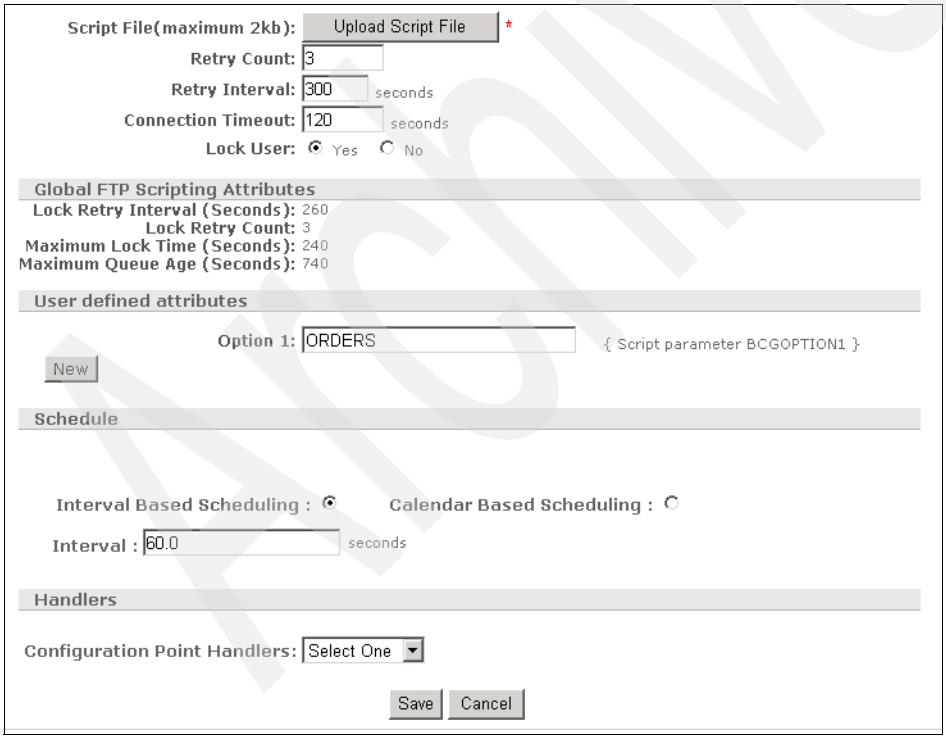


Figure 13-9 Create FTP Scripting Gateway within the profile of Company V: 1 of 2

xiii. Click **Save**.

6. Each partner profile needs a default gateway. Return to the list of gateways for Company V and click **View Default Gateways**.
7. Select **FTPScriptingGateway** as the production gateway and click **Save**. You should now be able to see this gateway listed under the list of gateways for Company V as shown in Figure 13-10.

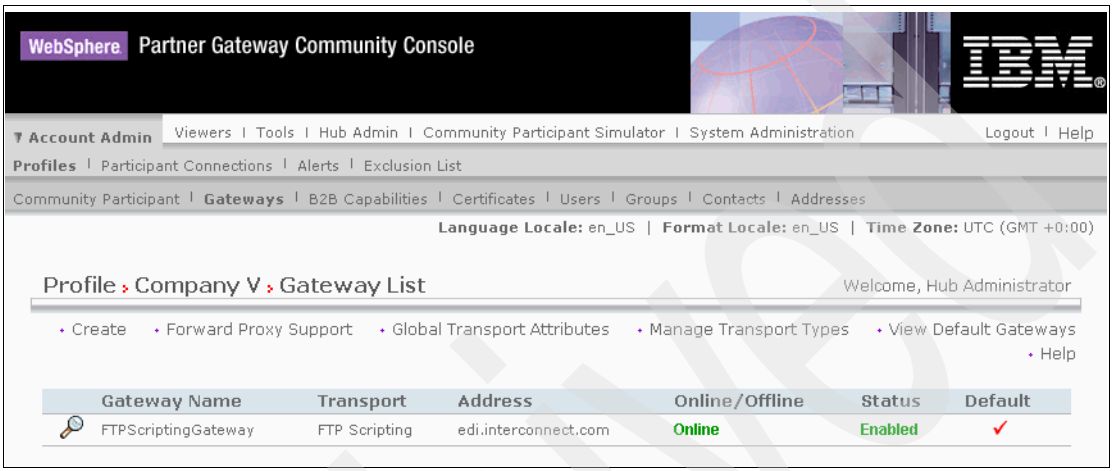


Figure 13-10 List of gateways for Company V

13.2.4 Updating the profile of Company E

The changes to the profile of Company E are limited to reviewing the B2B capabilities of Company E.

1. Select **Account Admin** → **Profiles** → **Community Participant**.
2. Click **Search** and then select Company E by clicking the **magnifying glass** icon next to **Company E**.
3. While working on Company E's profile, select **B2B capabilities**.
4. In the B2B Capabilities window, enable None/EDI-X12/ISA for the source and target.

13.2.5 Creating a participant connection

The last step in the configuration of the WebSphere Partner Gateway server of Company E is to activate the required participant connections.

1. While logged in as the hub administrator, select **Account Admin** → **Participant Connections**.

2. Select **Company E** as the source and **Company V** as the target. Click **Search**.
3. You now see several possible connections (see Figure 13-11), based on the common B2B capabilities of Company E and Company V. The required connection for these two companies is:
   – From None/EDI-X12/ISA to None/EDI-X12/ISA
4. Click **Activate** for this connection.

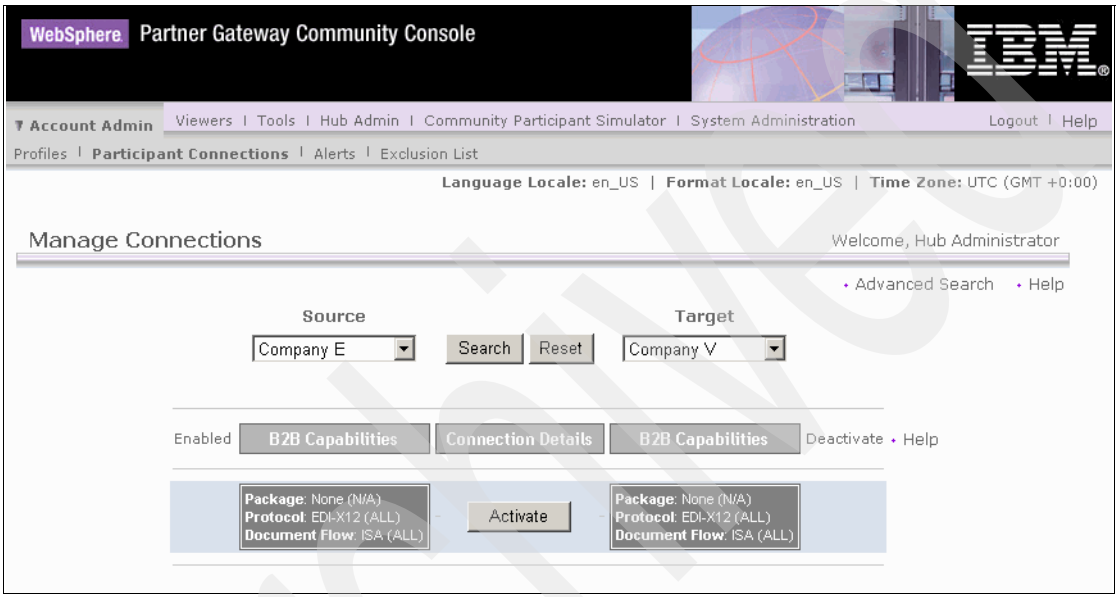


Figure 13-11 List of possible connections between Company E and Company V

Figure 13-12 on page 433 shows the resulting list of active connections. You can click **Gateways** to review the gateway selection. However, by default, FTPScriptingGateway should be the only defined gateway in the profile of Company V.

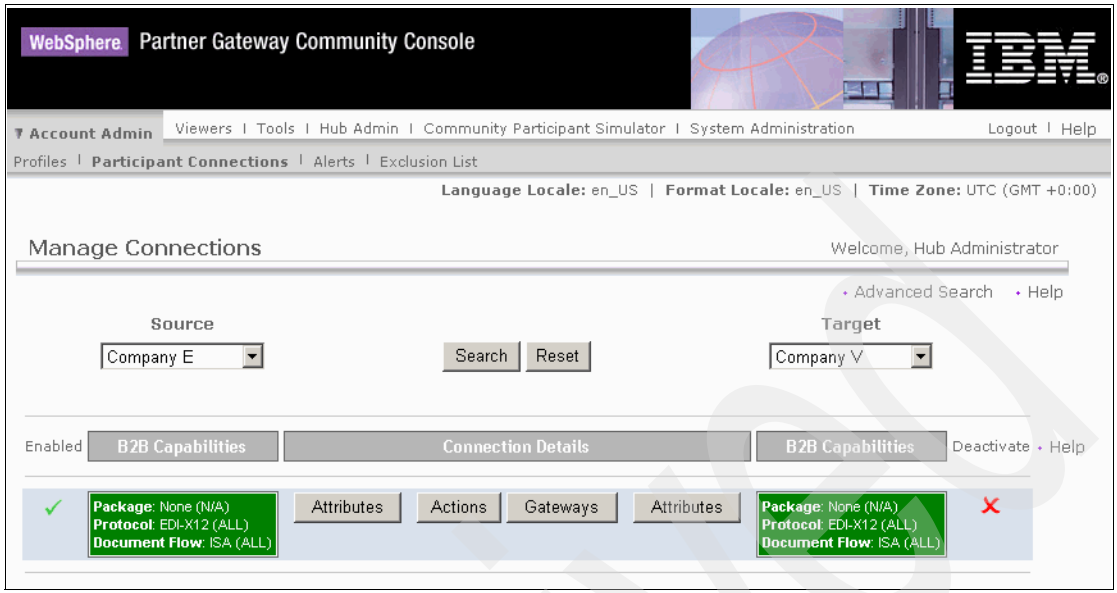


Figure 13-12 List of activated connections between Company E and Company V

13.2.6 Validating communication

It is beyond the scope of this redbook to discuss the configuration of the FTP Gateway in the EDI services company. The only information needed here is that the FTP Gateway in the EDI services company is operational with the user ID `accountv.companyv`.

To validate the transfer of EDI-X12 files, copy a file into the directory `\wpgv6\data\companye\edi_out\Production`. At the next polling cycle, this file should be picked up by the FileDirectoryTarget and handed over to the Document Manager. Also make sure that the EDI file has the business identifiers for the sender and receiver as `companye` and `companyv`, respectively.

The Document Manager should then use its built-in FTP Scripting function to send the file to Company V. To be more specific, the file is sent from Company E to the FTP Gateway on the EDI services company, which then converts the FTP commands it received to appropriate VAN commands. The EDI services company then sends the files to Company V over its private VAN. The details of communication between the EDI services company and Company V are outside the purview of this redbook.

You can again use the document viewer in WebSphere Partner Gateway to monitor the progress and the result of the communication.

13.3 Configuration of Company E for inbound

This section explains how the FTP Scripting capability of WebSphere Partner Gateway can be used to receive or retrieve files from trading partners like Company V that communicate to the outside world using their private VANs and an EDI services company.

In this redbook, going forward we make an assumption that Company V has an active mailbox in the EDI services company. Because Company V is registered with it, the mailbox is used by Company V to communicate with its trading partners, using its private VAN.

13.3.1 Creating a FTP Scripting Target for Company E

To retrieve files from Company V using FTP Scripting, Company E should have an FTP Scripting Target configured.

Configuring an FTP Scripting Target

To configure a target, follow these steps:

1. Log in to WebSphere Partner Gateway server as hubadmin, then select **Hub Admin** → **Hub Configuration** → **Targets.**
2. In the Target Details window (shown in Figure 13-13 on page 435), type `FTPScriptingTarget`, for example, in the Target Name field.
3. Select the Transport Type as **FTP Scripting**.
4. For the Server IP type `edi.interconnect.com`, for example.
5. Provide the user ID as `accountv.companyv`, in the User ID field.
6. Also provide the password required to log in to the user account in the Password field.
7. FTPS Mode radio button provides an option of choosing to secure the connection. Selecting Yes enables the transfer on FTP using SSL. For this scenario, we select **No**.

Archived

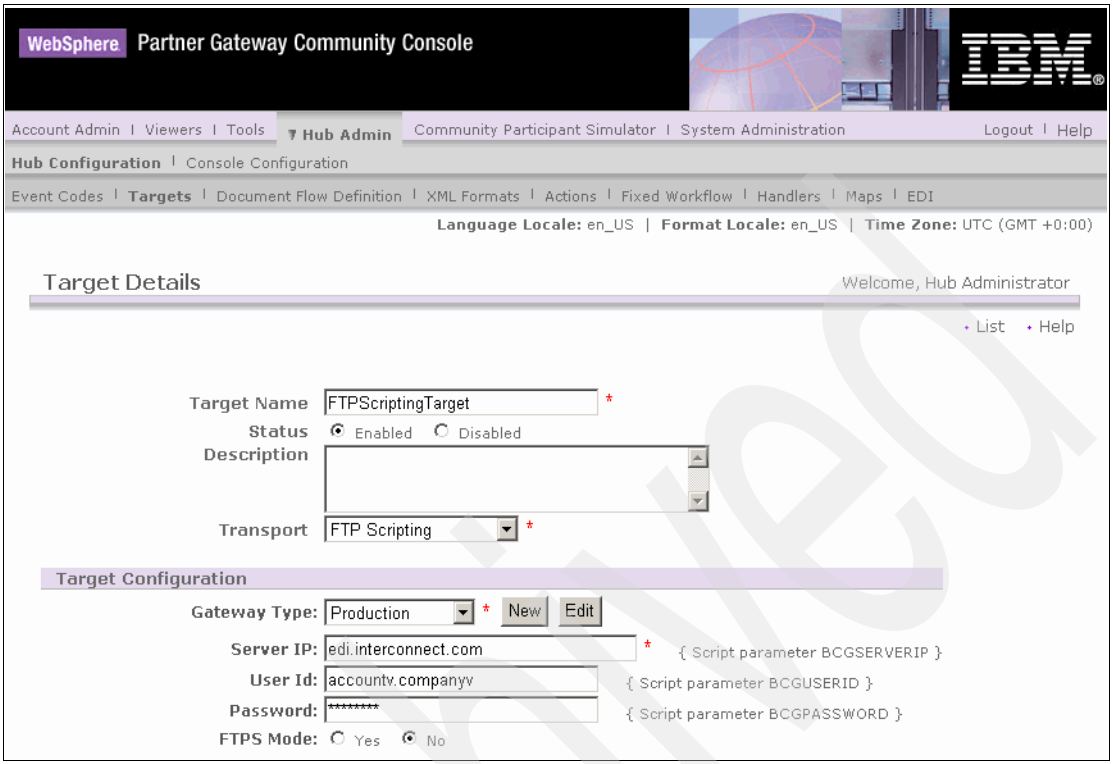


Figure 13-13 Configuring FTP Scripting Target: 1 of 2

8. Script File is where we upload the text file that is executed at the EDI services company, so that the document exchange can happen.

   Because we are configuring a target for inbound, the script file shown in Example 13-2 is uploaded as shown in Figure 13-14 on page 436, by clicking the **Upload Script File** button.

   All that is included between the %% tags in the script file will be substituted at run time by the values provided during the configuration of the FTP Scripting target.

Example 13-2 Sample script to receive files to Company E

```
open %BCGSERVERIP% %BCGUSERID% %BCGPASSWORD%
mget *
quit
```

Archived

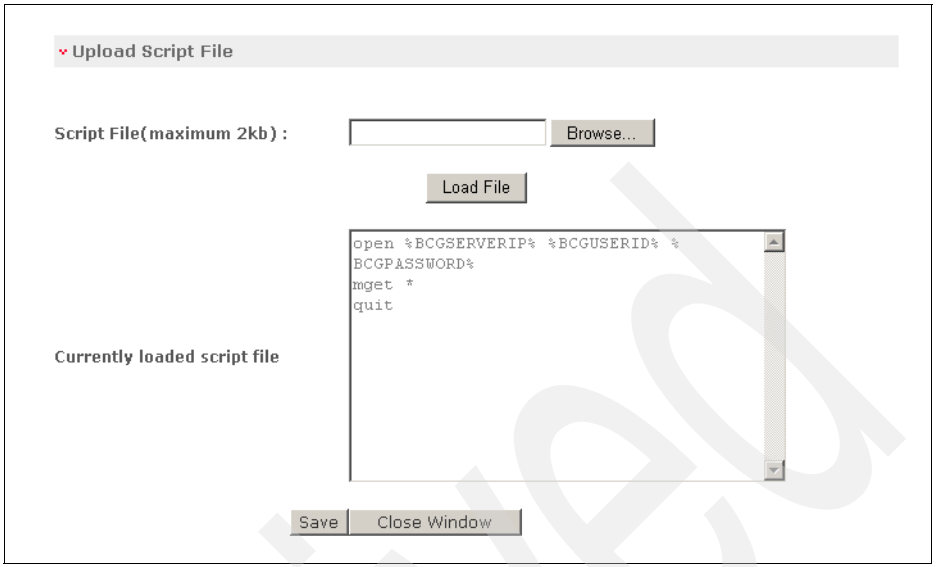


Figure 13-14 Uploading the script file

9. Lock User attribute locks the user account. This prevents other users from using the same account. Leave the default value selected for this, as shown in Figure 13-15 on page 437.
10. In addition to these attributes, there is a section for User defined attributes, where users can define and use their own attributes. These attributes are used in the script file, as substitution values for arguments between the %% tags .
11. Click **Save**.

Archived

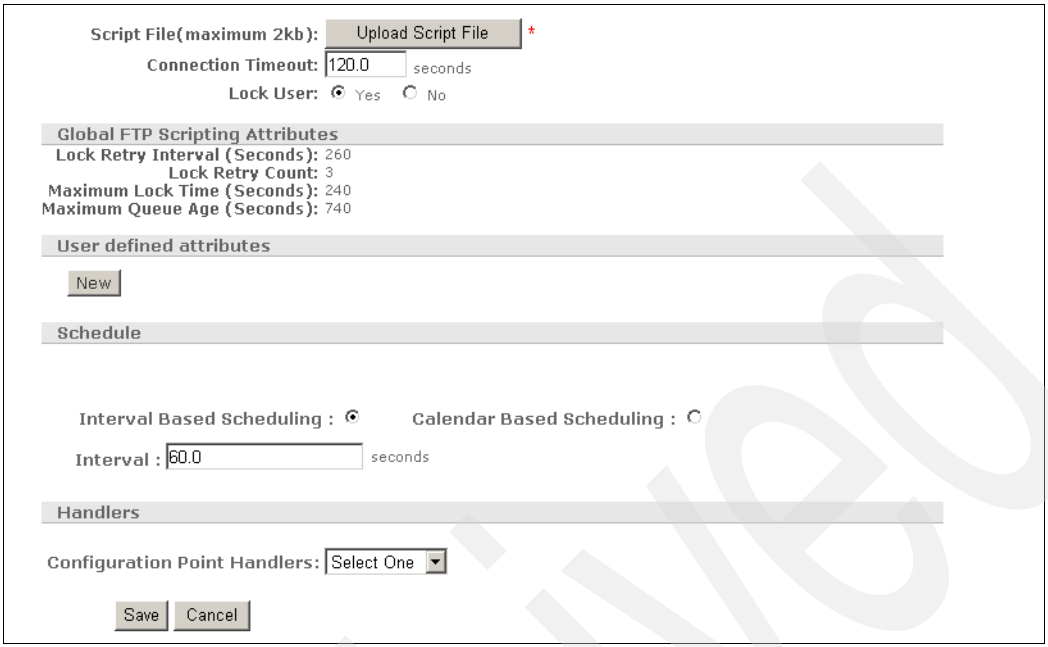


Figure 13-15 Configuring FTP Scripting Target (2 of 2)

12. Click **List** to see the FTPScriptingTarget listed under the list of targets configured for Company E, as shown in Figure 13-16.

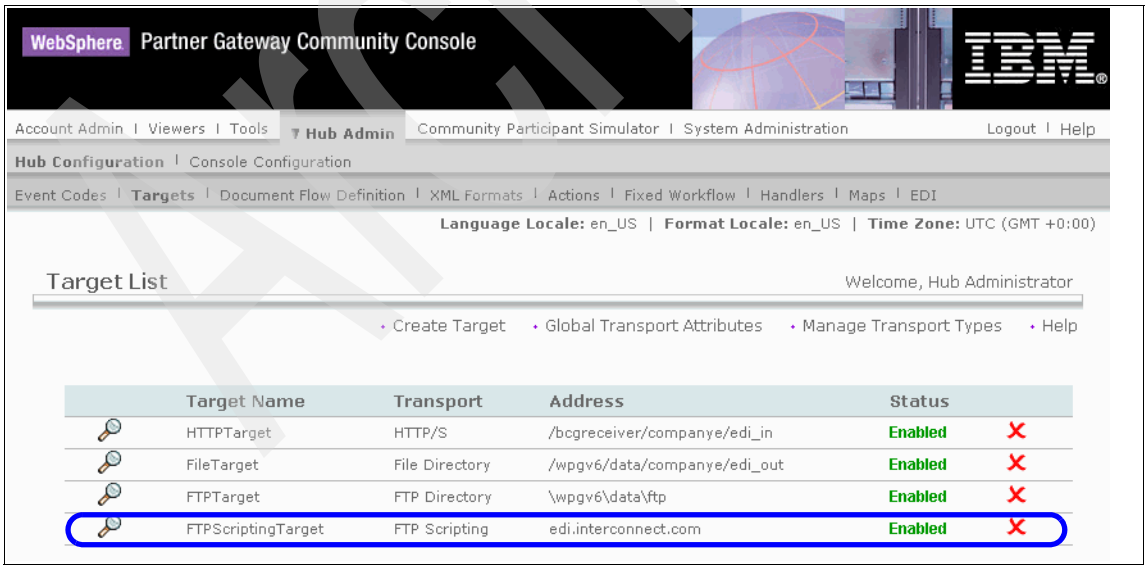


Figure 13-16 List of targets configured for Company E

Archived

13.3.2 Updating the WebSphere Partner Gateway configuration

As part of the outbound document flow, we configured the B2B capabilities of Company E and Company V, so that you can send and receive EDI-X12 files. Now we shall configure the participant connection between Company V and Company E.

1. Log on to WebSphere Partner Gateway server on Company E as hubadmin and select **Hub Admin** → **Participant Connections.**
2. Select **Company V** as the source and **Company E** as the target. Click **Search**.
3. The possible participant connections are shown in Figure 13-17.
   a. Click **Activate** for the connection between None/EDI-X12 and None/EDI-X12.

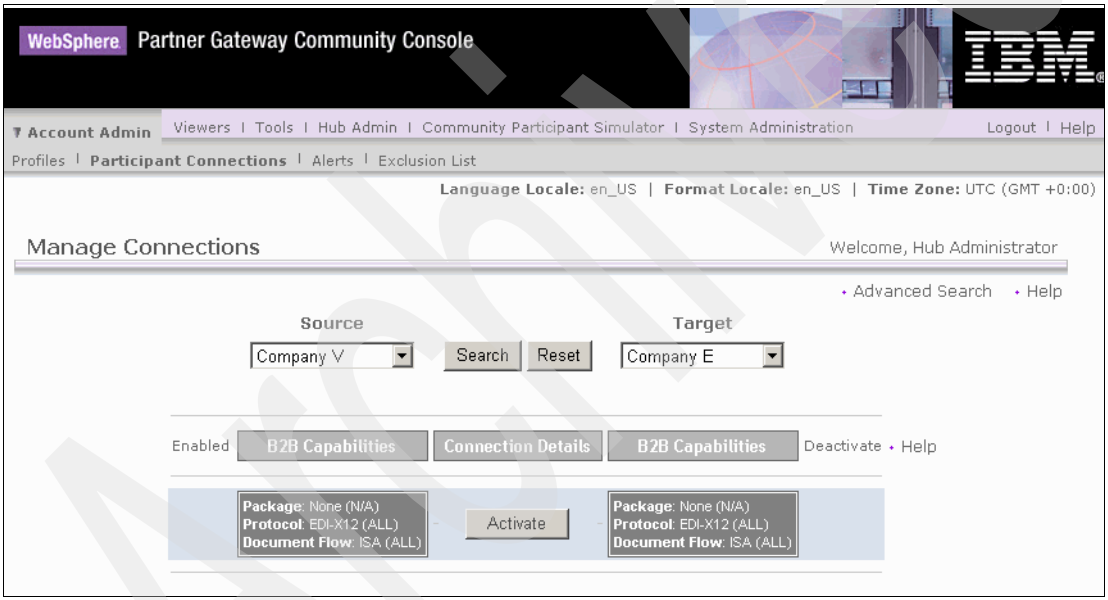


Figure 13-17 List of possible participant connections

The list of activated connections is as shown in Figure 13-18 on page 439.

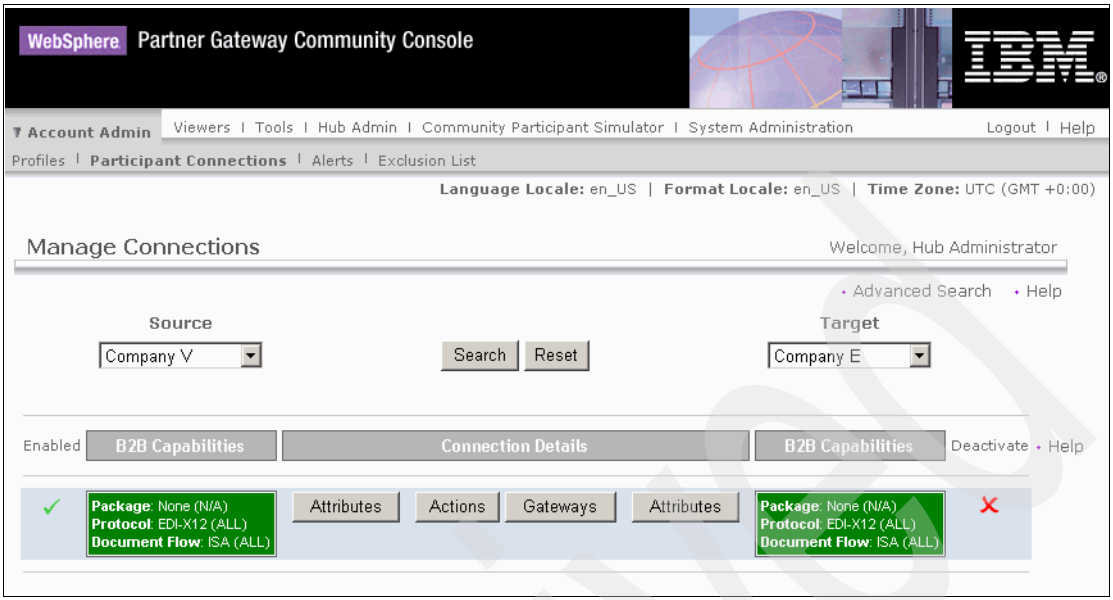


Figure 13-18 List of activated connections between Company V and Company E

In the Gateways tab for Company E , select the **File Directory Gateway**, which is pointed to \wpgv6\data\companye\edi_in directory on the physical file system.

13.3.3 Validating communication

When you have configured the FTP Scripting receiver, the script file is executed at scheduled intervals of time. As we have seen from Example 13-2 on page 435, the script file for the receiver executes the **`mget`** command to pull the files from Company V's mailbox on the EDI services company. This is repeated at scheduled intervals. Company V's mailbox is continuously polled for any documents that are for Company E. When there is an EDI-X12 file in the mailbox, the file is received into a shared file system. The Document Manager later picks it up and the participant connection from Company V to Company E is used to send the document to Company E's gateway. Thus the received file is sent to Company E, which can be found in the folder \wpgv6\data\companye\edi_in, as we had configured a File Directory Gateway for Company E. After the file is picked up, it is removed from the mailbox.

13.4 Securing VAN connectivity

From our prior discussion on connectivity to VAN, we have come to know that the basic protocol used is FTP. With FTP, security has always been a concern. The Internet Engineering Task Force (IETF) has accepted a standard, RFC2228, to and a number of security extensions to FTP. For more information about this standard, visit this Web site:

http://www.ietf.org/rfc/rfc2228.txt?number=2228

This standard allows for a Secure Socket Layer (SSL) like extension of FTP. We can add an SSL handshake to the FTP communication between FTP Scripting functions in WebSphere Partner Gateway and the FTP EDI services gateway. This SSL handshake allows for authentication of server or both client and server, and encryption of the data so that only the intended receiver can decrypt it.

In this redbook, we assume that the FTP gateway on the EDI services company is set up for secure exchange, and that we are provided with a Root certificate of the signer of the EDI services public certificate. This root certificate serves as the trust certificate for Company E, enabling Company E to trust any data coming from the EDI services company gateway, in effect from Company V.

13.4.1 Uploading certificates to Company E's WebSphere Partner Gateway

To enable server authentication, we must ensure that the certificate of Certifying Authority (CA), is available to WebSphere Partner Gateway. CA certificates or root certificates are uploaded to WebSphere Partner Gateway in the Operator profile.

When this file is received by Company E, the hubadmin user can upload it.

1. Select **Account Admin** → **Profiles** → **Community Participant**.
2. Click **Search** and click the **magnifying glass** icon next to the profile of the **Hub Operator**.
3. Select **Certificates**.
4. You should now see the list of certificates that are currently used by Company E as shown in Figure 13-19 on page 441. Click **Load Certificate**.

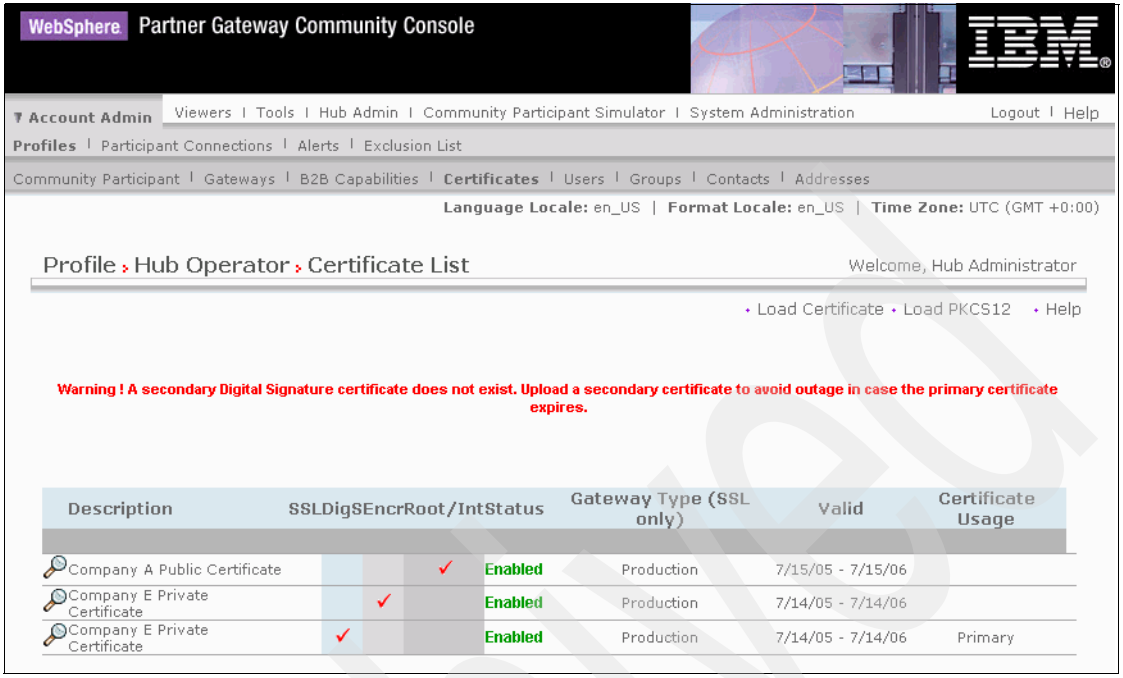


Figure 13-19 List of certificates under the Hub Operator profile

5. In the Create New Certificate window (Figure 13-20 on page 442), complete these tasks:

   a. For Certificate Type, select **Root Certificate**.
   b. Type a description for the new certificate.
   c. For Status, select **Enabled**.
   d. For Certificate, click **Browse** to locate the certificate on your hard disk.
   e. Click **Upload**.

Archived

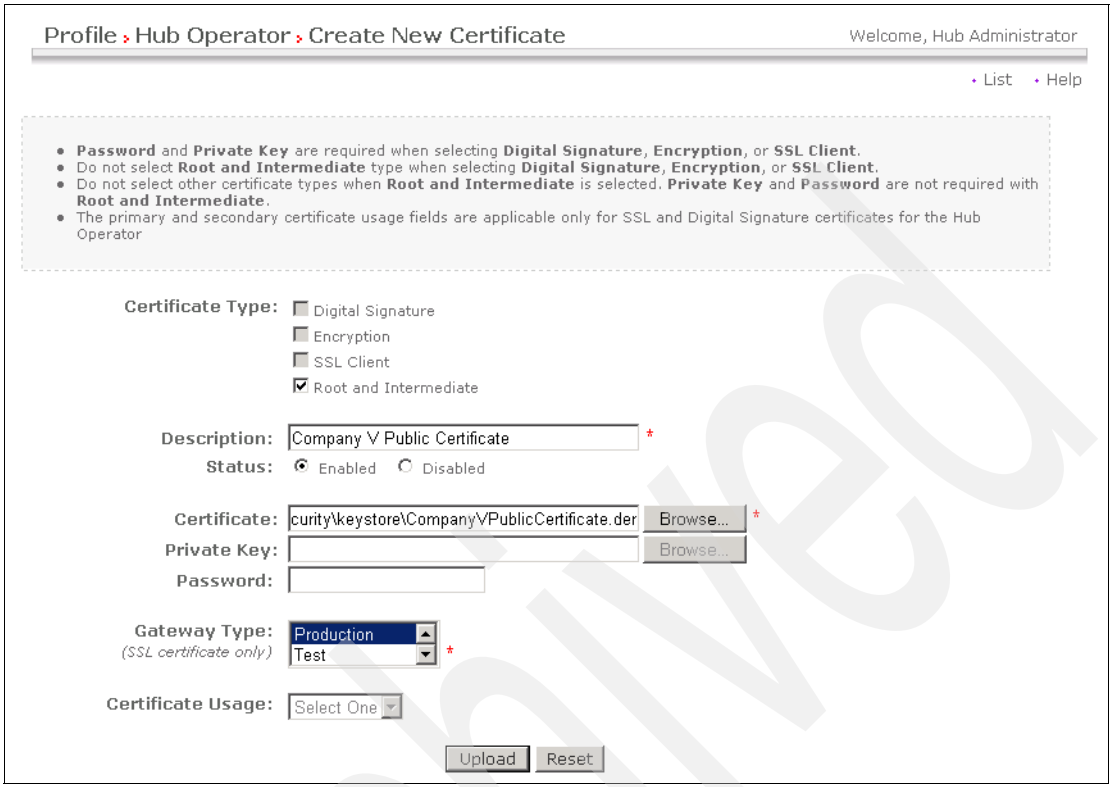


Figure 13-20 Uploading a new certificate

6. The WebSphere Partner Gateway server analyzes the certificate and presents the user with several pieces of information about the certificate. Review it carefully and click **Save**. Figure 13-21 on page 443 shows the new list of certificates in the profile of the Operator.

Archived

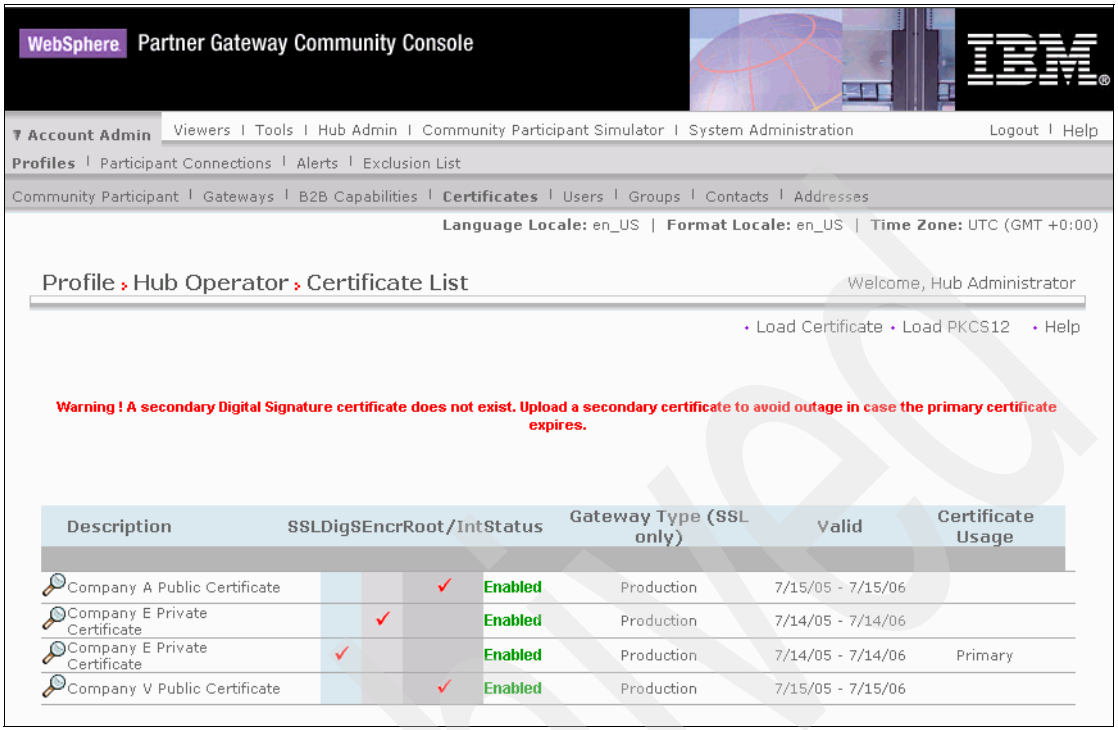


Figure 13-21 List of certificates under the Hub Operator profile

13.4.2 Updating the FTP Scripting gateway to use secure mode

The next change to the WebSphere Partner Gateway server is to update the existing FTP Scripting gateway, shown in Figure 13-22 on page 444.

Archived

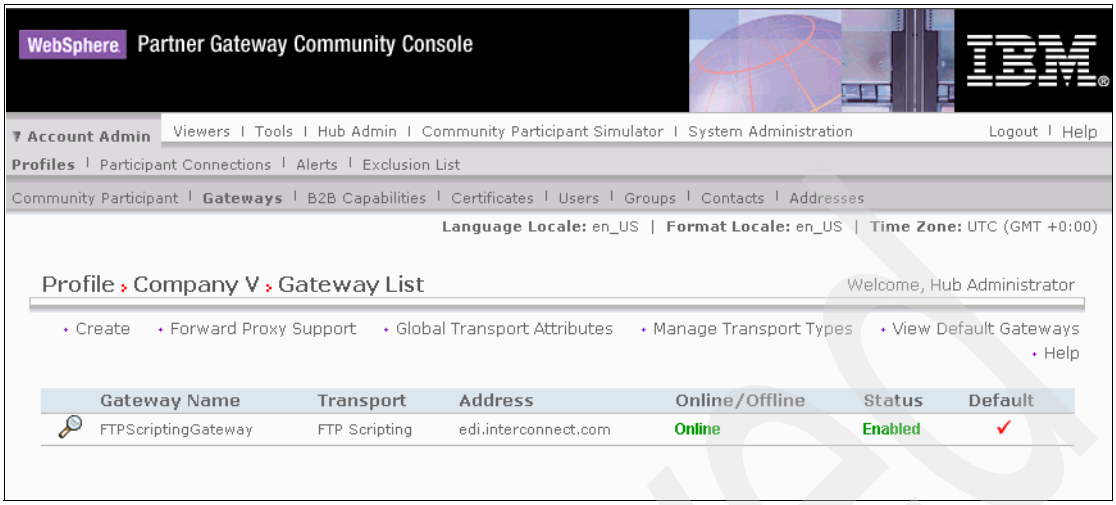


Figure 13-22 Existing FTP Scripting Gateway in the profile of Company V

Follow the steps below to update the gateway configuration:

1. Click on the magnifying glass icon next to the FTPScriptingGateway.
2. In the gateway details window shown in Figure 13-23 on page 445,change the radio button selection for FTPS Mode. Switch the FTPS Mode to **Yes**.

Archived

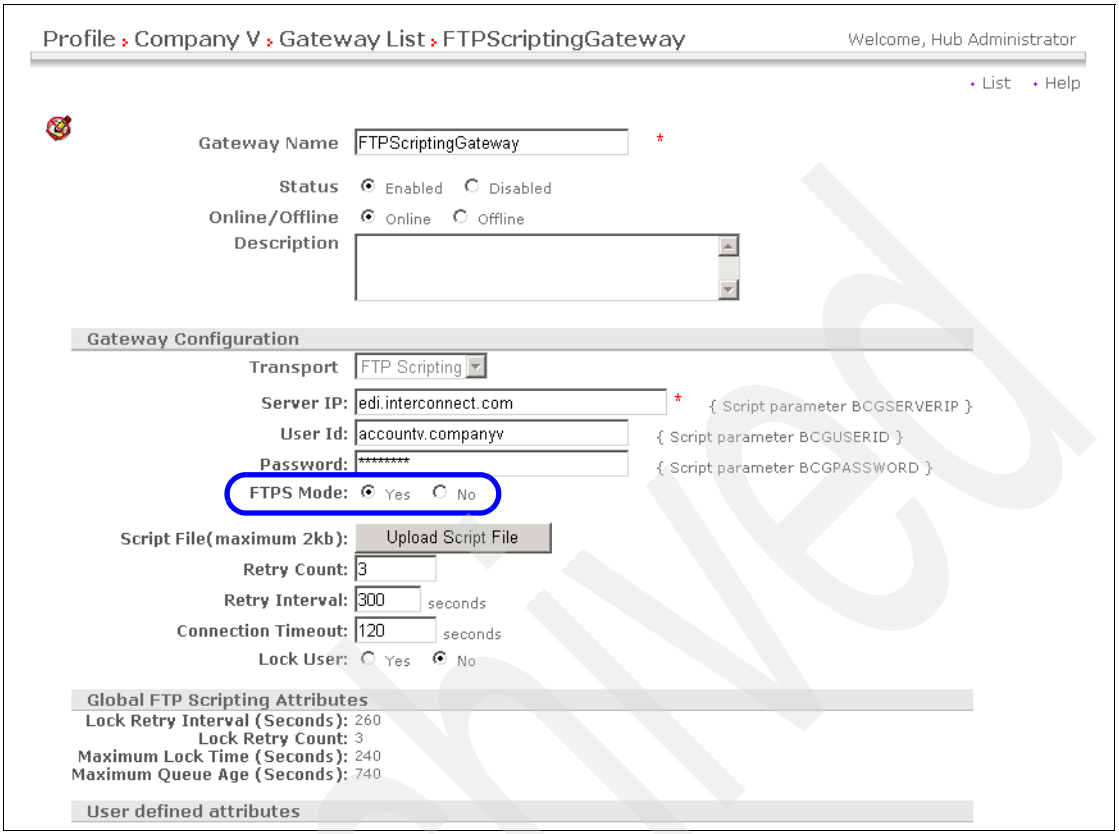


Figure 13-23 Updating the gateway to use secure mode

3. Click **Save**.

13.4.3 Validating outbound communication

To validate the outbound communication, you can use the same procedure as outlined in 13.2.6, “Validating communication” on page 433. Use the document viewer to inspect the results of the communication.

This time, when the WebSphere Partner Gateway Scripting function contacts the EDI services company’s FTP gateway, it sends an **`auth`** command. The FTP Gateway responds by sending its public certificate. This certificate is validated by WebSphere Partner Gateway, for which it needs the certificate of the CA that signed the certificate of EDI services. This certificate has been uploaded in section 13.4.1, “Uploading certificates to Company E’s WebSphere Partner Gateway” on page 440. As a result, WebSphere Partner Gateway is now able to trust the communication from the EDI services company. The FTP Scripting

Archived

function of WebSphere Partner Gateway uses the certificate to encrypt any further communication between the EDI services company and Company E.

The success of the outbound operation can be seen in the document viewer.

13.4.4 Using secure mode for inbound communication

To configure secure mode for inbound communication we need to do the following.

Updating the configuration of FTP Scripting target

Follow the steps below to update the configuration of the FTP Scripting target on Company E, so that secure communication mode is enabled.

1. While logged in as hubadmin on the WebSphere Partner Gateway server of Company E, go to **Hub Admin** → **Hub Configuration** → **Targets**.
2. Click the **magnifying glass** icon next to the **FTPScriptingTarget.**
3. Select **Yes** for FTPS Mode, as shown in Figure 13-24 on page 447.

Archived

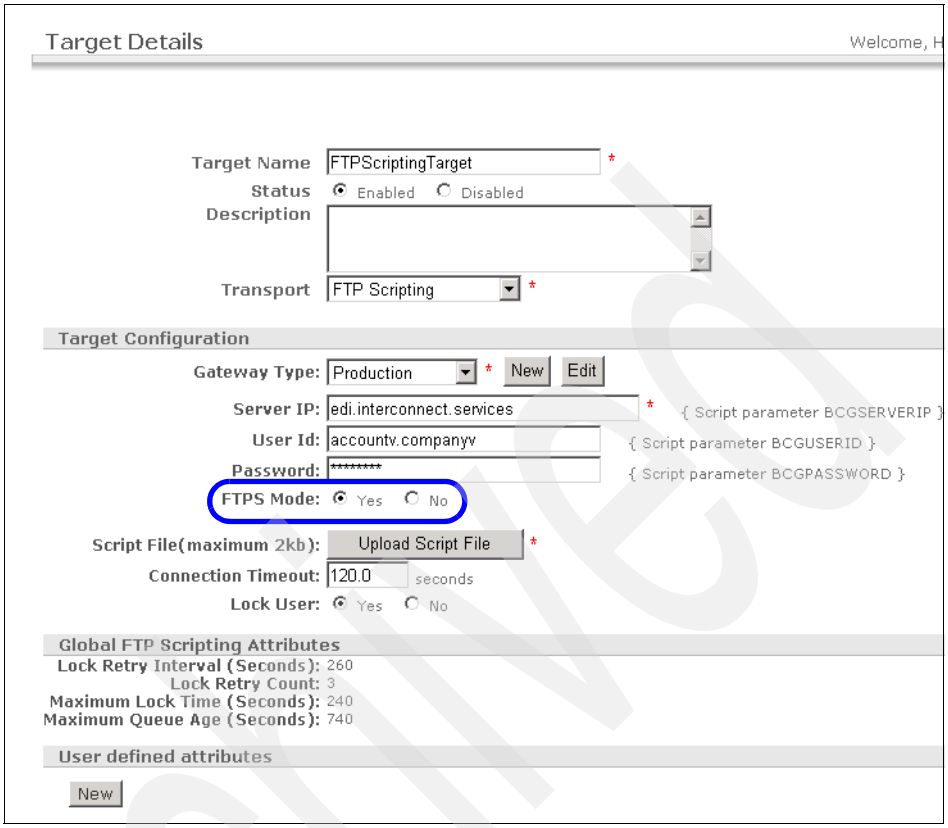


Figure 13-24 Updating the FTP Scripting Receiver configuration for secure mode.

4. Click **Save**.

13.4.5 Validating inbound communication

Inbound communication is similar to what was discussed in 13.3.3, “Validating communication” on page 439, except that this time the communication is over a secure medium. The SSL handshake and authentication happens as explained in 13.4.3, “Validating outbound communication” on page 445.

The success of the transfer can be seen using the Document Viewer at Company E.

Archived

13.5 Summary

At the end of this chapter we see how we can build an exchange for Company E with Company V, which communicates to the outside world only on its own private VAN. This is possible by using the FTP Scripting capabilities of WebSphere Partner Gateway.

Archived

Part 4

Native mapping support for non-EDI standards

Archived

© Copyright IBM Corp. 2005. All rights reserved.

Archived

14

Native mapping support in WebSphere Partner Gateway

Before you can use WebSphere Partner Gateway to translate data, and send or receive transactions, messages, and files, you must define certain information. This information describes how your system sends and receives data, the format of the data in your applications, mapping to a standard, to and from whom you receive data, and other pertinent information.

WebSphere Partner Gateway V6 introduced native mapping support. Mapping support was previously provided with WebSphere Data Interchange. Data Interchange Services client is an interface with which you can use a PC to create and maintain XML Schema document definitions, XML DTD document definitions, EDI Standards, ROD document definitions, and maps.

Beginning in Part 4, “Native mapping support for non-EDI standards” on page 449 we explain how to use Data Interchange Services client to develop data transformation and validation maps for WebSphere Partner Gateway to use when processing XML documents, and Record Oriented Data (ROD) documents.

In Part 5, “Native mapping support for EDI standards” on page 575 we extend this to EDI and the additional functions for WebSphere Partner Gateway to use when processing EDI documents.

© Copyright IBM Corp. 2005. All rights reserved.

14.1 Install the Data Interchange Services client

The Data Interchange Services client has an InstallShield wizard that guides you through the installation process. To use the InstallShield wizard, you must be logged in as an administrator.

Follow these steps to install the Data Interchange Services client:

1. Insert the Data Interchange Services CD into the CD-ROM drive.
   a. On the menu bar, click **Start** → **Run**.
   b. Find the directory that contains the **DISClient60_Ent.exe** executable.
   c. Run this executable to start the InstallShield wizard.
2. The Welcome window opens as the InstallShield wizard prepares to install the Data Interchange Services client. Click **Next**.

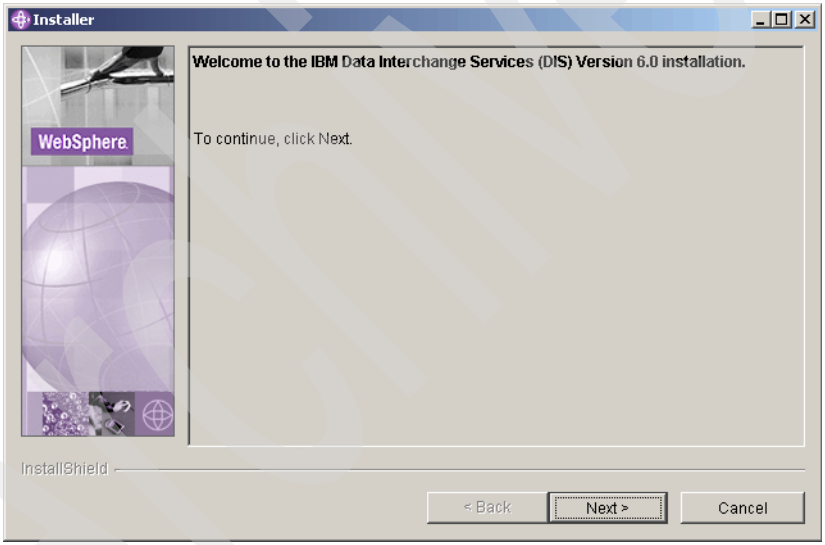


Figure 14-1 Welcome window

3. The license agreement opens. Read the information and license terms on the panel. Click the appropriate button to accept the terms of the license agreement and to indicate that you have read the notice and agree to its terms. Click **Next**.

Archived

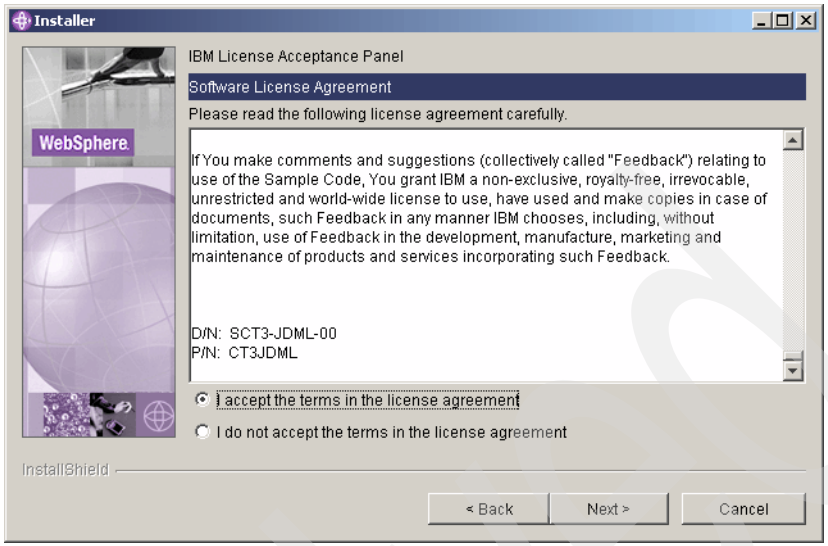


Figure 14-2 License agreement

4. Change the default installation directory (Figure 14-3) for the Data Interchange Services client to:

```
C:\IBM\DIS_Client\V6.0
```

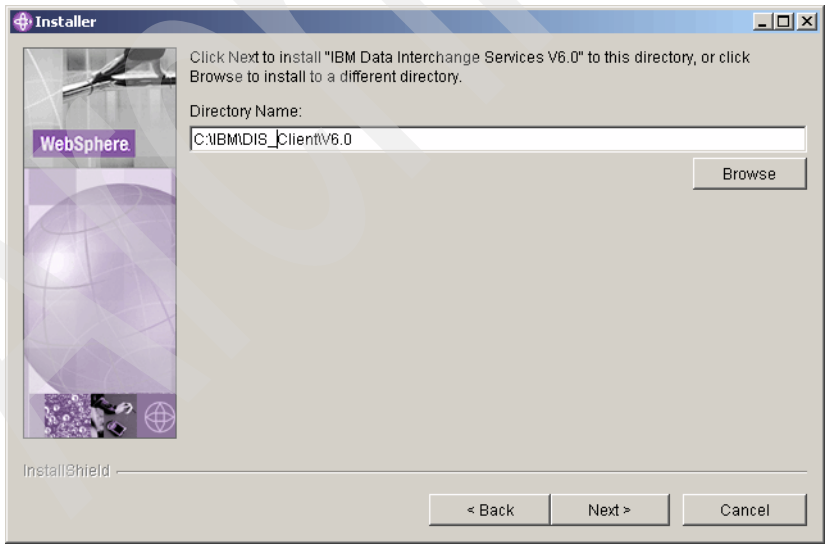


Figure 14-3 Select installation directory

5. The Installer window (Figure 14-4) displays a listing of the installation directory and the total size requirement. Click **Next**.

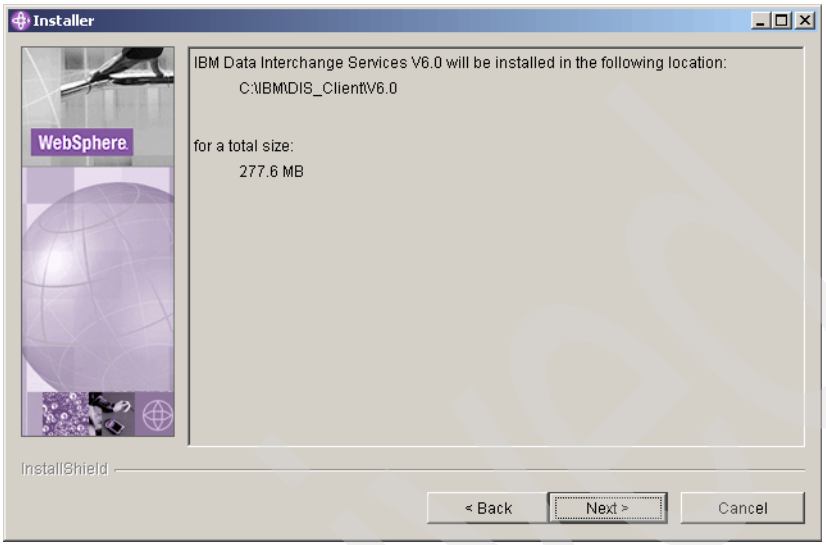


Figure 14-4 Selected directory

6. The InstallShield wizard begins copying program files (Figure 14-5). To stop this process at any point, click **Cancel**.

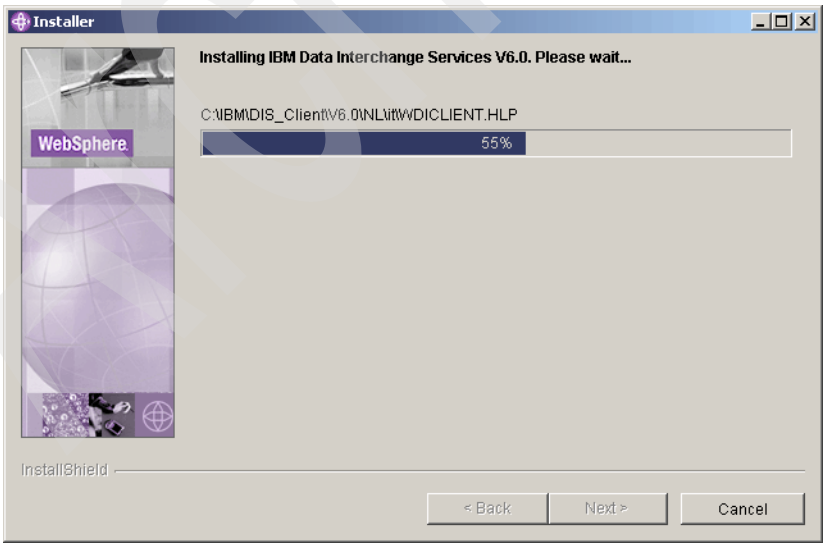


Figure 14-5 Progress window

7. The window displays the successful installation message (Figure 14-6). Click **Finish**.

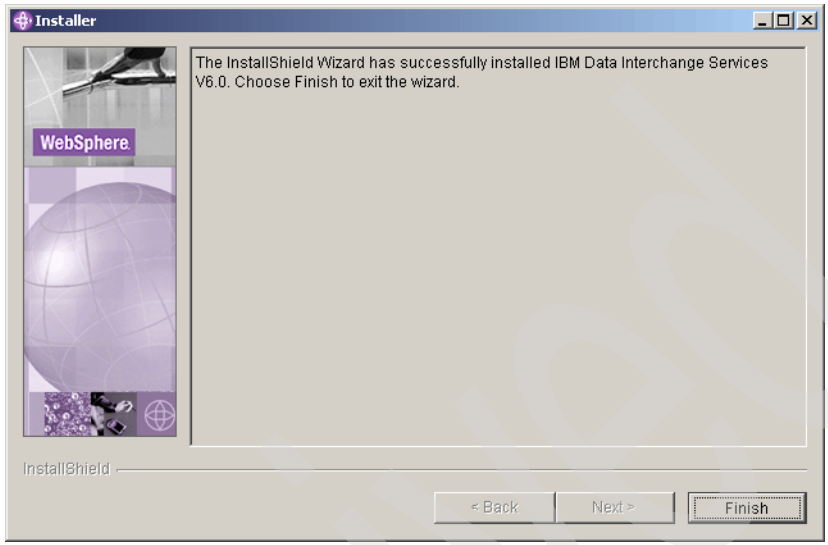


Figure 14-6 Successful installation

The installation of the Data Interchange Services client is complete.

Archived

14.2 The system view of Data Interchange Services

When installing and configuring a Data Interchange Services environment, you need to make a distinction between a development environment and a runtime environment. In a development environment users:

- Import EDI standards.
- Import XML DTDs or Schemas.
- Create record oriented document definitions (ROD).

> **Note:** For WebSphere Data Interchange users, the ROD format is same as Application Data format (ADF) or Data format (DF).

- Create maps.

In a runtime environment, the translation and routing of EDI, XML and ROD messages occur. The platform of the run time can be different from the platform of the development environment.

14.3 Development environment

The development environment can be configured in a number of ways. A typical development environment consists of the Data Interchange Services client on a number of Windows machines and either a DB2 or Oracle database accessed with Open Database Connectivity (ODBC). The database can be local or remote. This database could also be running on a different platform from the Data Interchange Services client.

Archived

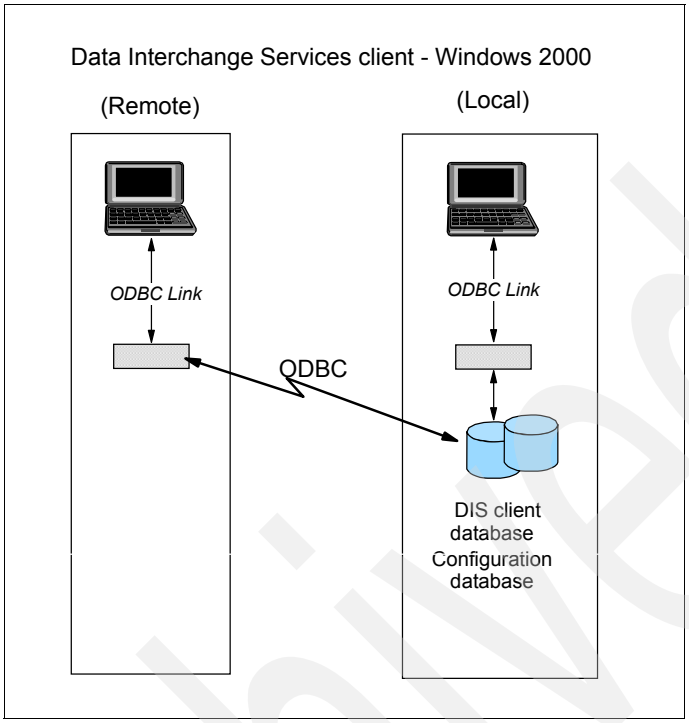


Figure 14-7 Client connectivity

An alternative development environment to working on a remote DB2 or Oracle database is to use the local development environment installed when performing a default installation of the Data Interchange Services client. This local development environment consists of a connection to a Microsoft Access database.

14.3.1 Configure a development environment using DB2

As discussed above, an alternative to using the local Microsoft Access database is to connect the Data Interchange Services client to an industrial strength relational database management system such as DB2 or Oracle. A script is provided to create the tables necessary to build such a development environment in either environment.

This script, called bcgDIS.sql, is installed during the DBLoader installation and can be found under the scripts directory of the DB2 loader directory

Follow the steps which detail how to build a development database for the Data Interchange Services client in DB2:

Archived

1. If you are installing on Windows, log on with your Db2 administrator user ID. Click **Start** → **Programs** → **IBM DB2** → **Command Line Tools** → **Command Window** to open a DB2 command window.
2. At the command prompt issue the command:

   ```
   DB2 CREATE DATABASE <DB_NAME> ON C: USING CODESET UTF-8 TERRITORY US
   ```

3. Next we establish a connection to the newly created database. Issue the command:

   ```
   CONNECT TO <DB_NAME>
   ```

4. After successfully connecting to the database we create a buffer pool. Issue the command:

   ```
   CREATE BUFFERPOOL BUFF32K IMMEDIATE ALL DBPARTITIONNUMS SIZE 500
   NUMBLOCKPAGES 0 PAGESIZE 32 K
   ```

5. Create a tablespace:

   ```
   CREATE TABLESPACE TS_PROFILE IN DATABASE PARTITION GROUP IBMDEFAULTGROUP
   PAGESIZE 32K MANAGED BY SYSTEM USING ('C:\IBM\bcgdbloader\tables\profile')
   EXTENTSIZE 32 PREFETCHSIZE 32 BUFFERPOOL BUFF32K OVERHEAD 24.10
   TRANSFERRATE 0.90 DROPPED TABLE RECOVERY On
   ```

6. Finally, having created the database, we can run the script that will create the tables used by the Data Interchange Services client. Navigate to the scripts directory located in the <DBLOADER_INSTALL_DIR> and issue the command:

   ```
   db2 -td! -f bcgDIS.sql -z DIS.log
   ```

Archived

14.4 Component view of the development environment

With the Data Interchange Services client successfully installed, we can launch the application from our Windows 2000 menu bar.

To open the Data Interchange Services client, follow these steps:

1. Click the **Start** menu.
2. Select **Programs** → **IBM Data Interchange Services 6.0** → **DIS Client 6.0**

   The Data Interchange Services client opens. For users of the WebSphere Data Interchange Client, the Data Interchange Services Client will look familiar. The Data Interchange Services client is essentially a cut-down version of the WebSphere Data Interchange client, including only the functional areas that are specific to WebSphere Partner Gateway. The functional areas of relevance are:

   – XML functional area
   – EDI functional area
   – Record Oriented Data functional area
   – Mapping functional area.

   Each of these functional areas can be accessed by clicking the relevant icon from the Data Interchange Services client navigation bar. Figure 14-8 shows a view of the Data Interchange Services client navigation bar for a development environment.

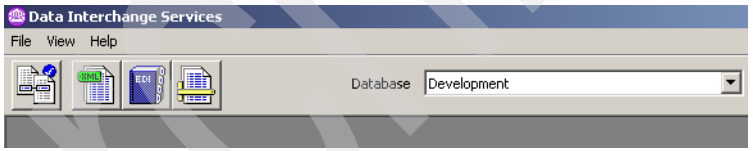


Figure 14-8 Data Interchange Services client navigation bar

If the Data Interchange Services client were connected to a runtime environment, the navigation bar as shown in Figure 14-8 would appear with a slight difference: the icon representing the Record Oriented Data functional area would be inactive. The navigation bar shown when the Data Interchange Services client is connected to a runtime environment as shown in Figure 14-9 on page 460.

Archived

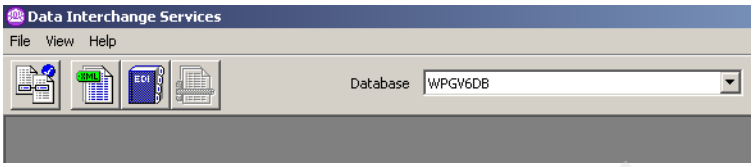


Figure 14-9 Connected to runtime

Connection to a runtime environment highlights yet another difference between the WebSphere Data Interchange Client and the Data Interchange Services client. The list windows shown under each of the functional areas listed previously differ based on connection type.

14.5 Connect the Data Interchange Services client to a development database

After performing a default installation, the Data Interchange Services client opens with a connection to the development database already established. This development database is, in reality, a Microsoft Access database that is installed during the Data Interchange Services installation. This Microsoft Access database is located on the same machine as the Data Interchange Services client and is used to store all configuration information required to work with the Data Interchange Services client.

As discussed above, an alternative development environment using either DB2 or Oracle is also an option for the Data Interchange Services client. To connect to such a database, the particular database should be registered as a data source on the Windows machine on which the Data Interchange Services client is running.

From the Data Interchange Services client, follow these steps:

1. Select **View** → **Databases** from the menu bar to display a view of all database connections that are available to the Data Interchange Services client.
2. To create a new database connection select **File** → **New**.
3. In the Database window that opens, enter the following details:
   a. Enter a **Database Name.**
   b. Select the **Data Source Name** from the list of available data sources on the Windows machine.
   c. Enter the **Database Qualifier** for the Data Source Name.

d. Set the **Database Type** as **Data Interchange Services**.

4. Click **OK**.

The newly added database should now be available from the **Database** menu displayed in the Data Interchange Services client. To work with the newly added database, simply select the database from the drop-down menu and select any of the icons from the Data Interchange Services client toolbar. You are asked to enter a **User ID** and **Password** to authenticate access to the database. After you have entered a valid user ID and password, click **OK** to access the configuration data stored in the database.

14.6 Functional areas

The functional areas of importance to the Data Interchange Services client development environment are discussed in detail in the following sections.

14.6.1 XML functional area

The XML functional area provides access to XML dictionaries, schema document definitions, DTD document definitions, and Namespace objects. You can access the XML functional area by clicking the **XML button** on the Data Interchange Services client navigator bar.

Figure 14-10 XML functional area icon

Clicking the XML button displays a functional area window that contains one list window for each XML related object type. The XML functional area contains one list window for each of the following components:

- XML dictionaries
- Schema document definitions
- DTD document definitions
- Namespace objects

Archived

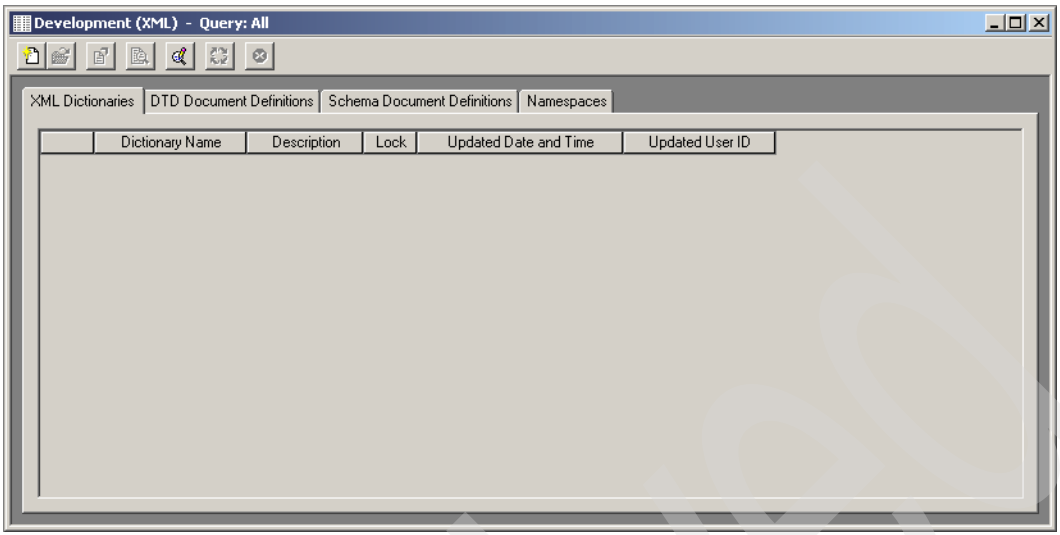


Figure 14-11 XML functional area

XML dictionaries

The XML dictionary is used to logically group a set of related XML schemas and DTDs. You should create an XML dictionary when you have a new group of associated schemas or DTDs that you wish to load into the Data Interchange Services client.

Schema document definitions and DTD document definitions

Unlike EDI standards where there are a small number of dominant EDI standard formats that define the document structure, there are numerous different XML schemas and DTDs. Also, because XML is *eXtensible*, users are free to create their own schemas and DTDs if they choose. Instead of restricting users to a fixed subset of schemas and DTDs, Data Interchange Services allows you to import the schemas and DTDs you use. You can obtain your schemas and DTDs from various sources, such as industry groups, standards bodies, vendors, trading partners, or you can create them yourself. You can map the documents described by those schemas and DTDs after the schemas and DTDs are imported into Data Interchange Services.

Note: An XML dictionary must be created before a schema or DTD can be imported into the Data Interchange Services client.

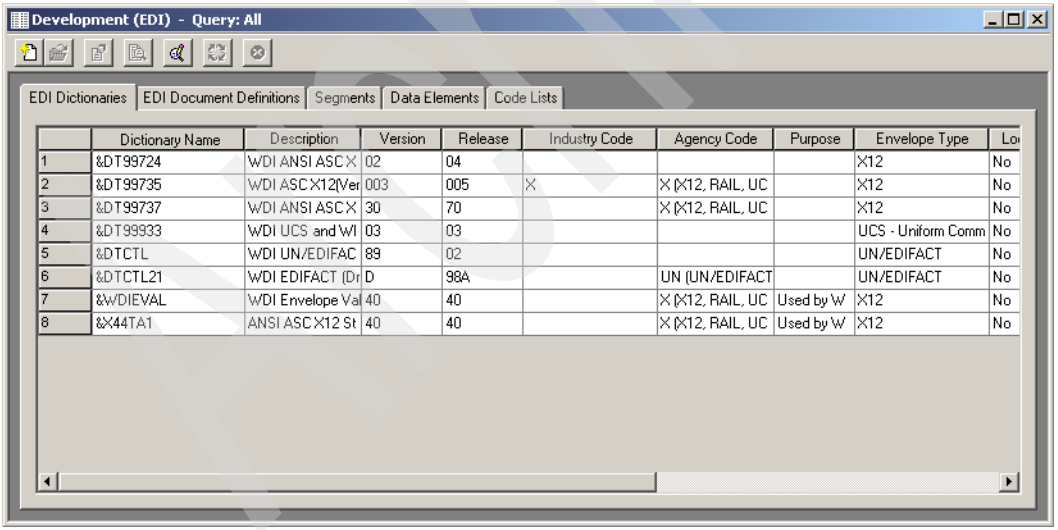
Namespace objects

The Namespaces list window is used to display a list of existing Namespace objects already defined in the Data Interchange Services client. An XML namespace provides a simple method for qualifying element and attribute names used in XML documents by associating them with namespaces identified by URI references. The namespace objects displayed in the Namespaces list window are typically those that are already associated with the schema definitions defined in the Schema Document Definitions list window.

14.6.2 EDI functional area

The EDI functional area is used to view and maintain EDI Standards. You can access the EDI functional area by clicking **EDI** on the Data Interchange Services client navigator bar. This displays a functional area window that contains several list windows. Each of these list windows displays a component that is part of the EDI functional area. The EDI functional area contains one list window for each of the following components:

- EDI dictionary
- EDI document definitions
- EDI segments
- EDI data elements
- Code lists



	Dictionary Name	Description	Version	Release	Industry Code	Agency Code	Purpose	Envelope Type	Lo
1	&DT99724	WDI ANSI ASC X	02	04				X12	No
2	&DT99735	WDI ASC X12(Ver	003	005	X	X (X12, RAIL, UC		X12	No
3	&DT99737	WDI ANSI ASC X	30	70		X (X12, RAIL, UC		X12	No
4	&DT99933	WDI UCS and WI	03	03				UCS - Uniform Comm	No
5	&DTCTL	WDI UN/EDIFAC	89	02				UN/EDIFACT	No
6	&DTCTL21	WDI EDIFACT (Dr	D	98A		UN (UN/EDIFACT		UN/EDIFACT	No
7	&WDIEVAL	WDI Envelope Val	40	40		X (X12, RAIL, UC	Used by W	X12	No
8	&X44TA1	ANSI ASC X12 St	40	40		X (X12, RAIL, UC	Used by W	X12	No

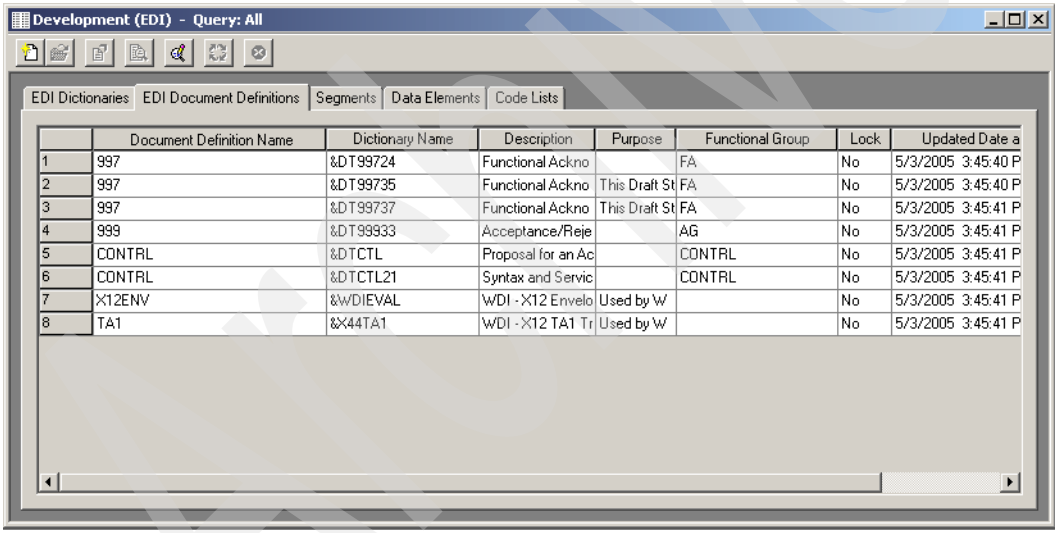
Figure 14-12 EDI functional area

EDI dictionary

You set up and maintain EDI dictionaries through the EDI dictionaries list window. An *EDI dictionary* is a means of logically grouping a set of related EDI Documents. The relationship between these EDI documents is typically the EDI standard and revision to that particular standard, to which the documents conform. Figure 14-12 on page 463 shows a view of the EDI dictionaries available in the Data Interchange Services client after a default installation.

EDI document definitions

The EDI Document Definitions list window displays a list of EDI documents already defined to the Data Interchange Services client. These EDI documents typically represent business documents such as invoices or purchase orders and are defined by standards bodies such as ANSI X12 or EDIFACT. Depending on the standard used, these EDI documents are referred to as *transactions* (X12) or *messages* (EDIFACT). Figure 14-13 shows a view of the EDI document definitions available in the Data Interchange Services client after performing a default installation.



	Document Definition Name	Dictionary Name	Description	Purpose	Functional Group	Lock	Updated Date a
1	997	&DT99724	Functional Ackno		FA	No	5/3/2005 3:45:40 P
2	997	&DT99735	Functional Ackno	This Draft St	FA	No	5/3/2005 3:45:40 P
3	997	&DT99737	Functional Ackno	This Draft St	FA	No	5/3/2005 3:45:41 P
4	999	&DT99933	Acceptance/Reje		AG	No	5/3/2005 3:45:41 P
5	CONTRL	&DTCTL	Proposal for an Ac		CONTRL	No	5/3/2005 3:45:41 P
6	CONTRL	&DTCTL21	Syntax and Servic		CONTRL	No	5/3/2005 3:45:41 P
7	X12ENV	&WDIEVAL	WDI - X12 Envelo	Used by W		No	5/3/2005 3:45:41 P
8	TA1	&X44TA1	WDI - X12 TA1 Tr	Used by W		No	5/3/2005 3:45:41 P

Figure 14-13 EDI dictionaries

EDI segments

An EDI document definition set is composed of EDI segments. In essence, each line of a business document corresponds to an *EDI segment*. EDI elements begin with a segment identifier assigned by the EDI Standard. They are mandatory, conditional, optional, or floating (floating EDI segments appear anywhere in the EDI Document Definition). All EDI segments except for floating

Segments appear in a fixed sequence for a given EDI document definition. The EDI segments that are defined in each of the EDI document definitions are displayed in the Segments list window.

EDI data elements

An EDI segment is composed of *EDI data elements*, which represent the individual units of data found in business documents, such as quantity ordered or unit price. EDI data elements appear in a sequence specified by the EDI Standard and are separated by a delimiting character, such as an asterisk. They have a minimum and maximum length, and are mandatory, conditional, or optional. The individual data elements that are used to define each EDI segment displayed in the Segments list window can be viewed in the Data Element list window.

Code lists

A *code list* is essentially a list of acceptable values. Code lists are provided by EDI Standards and indicate which values are valid for a specific EDI data element. A view of the code lists already defined to the Data Interchange Services client is shown in the Code Lists window.

14.6.3 Record Oriented Data (ROD) functional area

The *Record Oriented Data functional area* is used to view and maintain ROD document definitions, which describe the layout of proprietary documents. Often, an organization might refer to ROD files as *flat files* or *raw data*. You can access the Record Oriented Data functional area by clicking the **ROD icon** (Figure 14-14) on the Data Interchange Services client navigator bar.

Figure 14-14 ROD functional area icon

Clicking the ROD icon displays a functional area window that contains several list windows. Each of these list windows displays a component that is a part of the Record Oriented Data functional area:

- ROD dictionaries
- Record ID information objects
- ROD Document Definitions
- Loops
- Records
- Structures
- Fields

Figure 14-15 shows a view of the Record Oriented Data functional area.

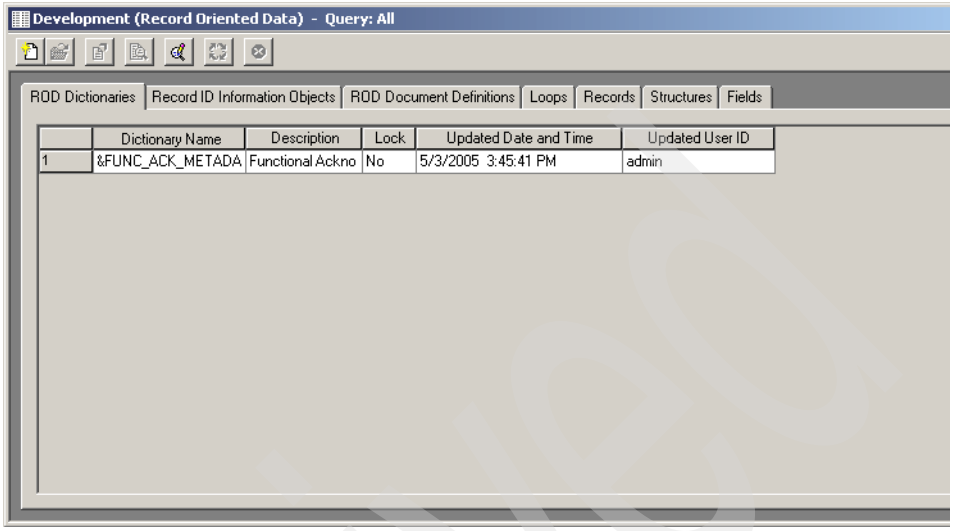


Figure 14-15 Record oriented data functional area

ROD dictionaries

You set up and maintain ROD dictionaries through the ROD dictionaries list window. An *ROD dictionary* is a logical means of grouping a related set of Record Oriented Data documents.

Record ID information objects

An *ROD document definition* is comprised of a number of record definitions. Each record definition is identified by a record id information object. The record ID information object is a definition of the type and position of the record ID within a record.

ROD document definitions

The term *ROD document definition* refers to the way a business application structures data (a document). These ROD documents typically represent business documents such as invoices or purchase orders and are defined by the organization or application that generates the document. The Data Interchange Services client needs a description of the document for each proprietary document that you integrate with Data Interchange Services. An application's document layout must be described to Data Interchange Services so that it can translate data between an application’s document in one format and a document in another document format.

Records

One or more records are used to define a ROD document definition. As the ROD document definition is used to represent business documents, *records* in the ROD document definition are used to represent each line of the business document. Records can be built from structures or distinct fields. Records can also be grouped and defined as loops.

Loops

Loops define a set of related records that are grouped and defined as repeating.

Structures

A *ROD structure* is a group of related ROD fields, which is probably unique to your company. When multiple ROD fields always appear together, you can designate the group as a ROD structure and give it a ROD structure name.

Fields

Fields are fundamental pieces of data, such as prices or item numbers or first names. Fields in an ROD document can be thought of as EDI data elements or XML elements.

14.6.4 Mapping functional area

The Mapping functional area is used to view and maintain maps and related items. You access the Mapping functional area by clicking the **Mapping** icon (Figure 14-16) on the Data Interchange Services client navigator bar.

Figure 14-16 Mapping functional area icon

Clicking the Mapping icon displays a functional area window that contains several list windows.

- Data transformation maps
- Validation maps
- Functional acknowledgement map
- Control strings
- Global variables
- Translation tables
- Code lists
- User exits

Archived

Figure 14-17 shows a view of the Mapping functional area.

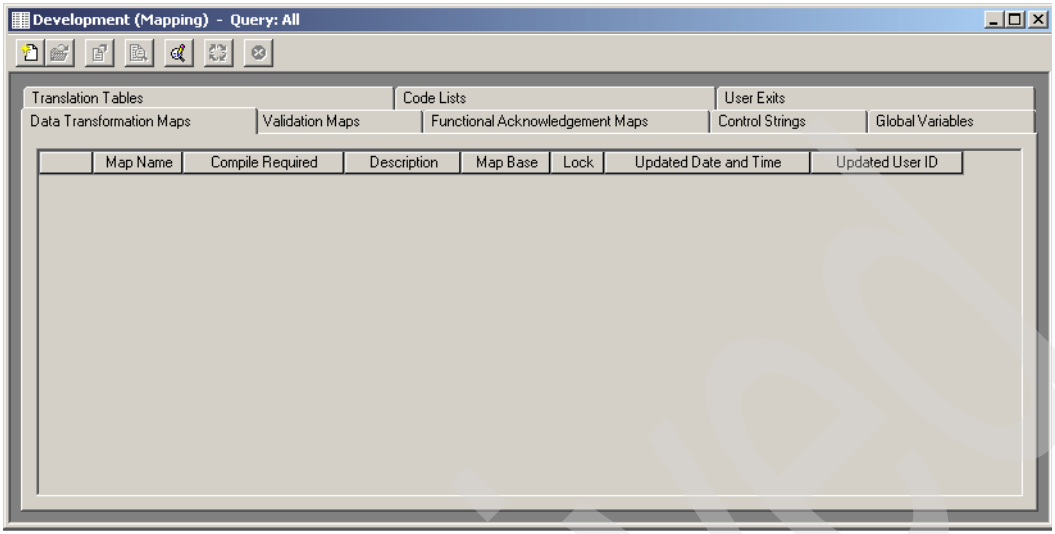


Figure 14-17 Mapping functional area

Data transformation maps

The primary function of Data Interchange Services is to start with a document in one format and convert it to another format. This cannot be done without instructions explaining how to create the second document from the first document. Data Interchange Services uses a map to provide instructions on how to convert a document in one format to another format. Before Data Interchange Services can translate a document from one form to another, a map must be created. Data transformation maps are the primary type of map used by Data Interchange Services to convert a document from one format to another. They provide the information needed by Data Interchange Services to gather data from a source document and create a target document using the source data. These maps provide a powerful method of mapping data from one document format to another.

Validation maps

Documents that are translated with a data transformation map are automatically validated against a document's definition. EDI documents are validated to ensure they comply with the corresponding EDI Standard. Validation maps provide the instructions needed to perform additional validation beyond what is specified in the EDI Standard. For example, the EDI standard might define a data element as having an integer value. Our organization has further defined this field has having an integer value between 0 and 9. A validation map could be used to enforce and validate this tighter constraint imposed by our organization.

Functional acknowledgement map

Functional acknowledgement maps provide the instructions to Data Interchange Services on how to produce a functional acknowledgement. Data Interchange Services can automatically generate EDI Standard functional acknowledgements for EDI documents received from a trading partner. To produce a functional acknowledgement, a functional acknowledgement map must be associated with the source document that is being translated in WebSphere Partner Gateway. The source document must be an EDI document.

Control strings

To make translation of a document more efficient, Data Interchange Services uses control strings to drive translation. There are several types of control strings used within Data Interchange Services. The visible control strings are the map control strings. Instead of using a map directly, Data Interchange Services uses a control string representing the map. Maps are never directly used to translate a document. Map control strings are created by compiling maps. Map control strings are required for all map types: data transformation maps, validation maps, and functional acknowledgement maps.

Global variables

A *global variable* defines a variable that can be used across translations. These variables are used much like variables in most programming languages. They can hold and manipulate data in data transformation maps, validation maps, and functional acknowledgement maps. While a document is being translated, data can be put into a global variable. After the translation of the document ends, the data remains in the global variable. The data in the variable might be available in the next translation, depending on the scope of the variable, regardless of the map that performs the translation. During subsequent translations within the scope of the variable, the data in the global variable can be obtained, manipulated, and changed.

Translation tables

A *translation table* is used to translate a value into a corresponding value. Translation tables are used in data transformation maps, validation maps, and functional acknowledgement maps. Translation tables contain a list of unique values. These are called source values. Each unique source value has a corresponding target value assigned to it. The target values do not have to be unique.

Code lists

A *code list* is a list of acceptable values. Code lists are provided by EDI Standards. They indicate which values are valid for a specific EDI Data Element. A code list can also be created to specify any list of valid values. All code lists

can be used in maps to validate any source simple element or variable. When Data Interchange Services validates or translates a document and the use of a code list is specified to validate a piece of data, the specified code list is searched to see if it contains the value.

User exits

A *user exit* profile defines a field exit routine to Data Interchange Services. It provides Data Interchange Services with the information it needs to call a field exit routine. A field exit routine can be called from data transformation maps, validation maps and functional acknowledgement maps. Field exit routines are invoked during the translation of a document when the Exit() function is encountered in a mapping command.

As we move through the next few chapters, we show examples using the various functional areas of the DIS client as we build, test, and deploy our maps.

14.7 Runtime environment

Data Interchange Services runtime environment requires an installation of WebSphere Partner Gateway to perform validation and transformation as defined in the Data Interchange Services client. Specifically, it is the Document Manager component of WebSphere Partner Gateway that performs translation and validation.

Data Interchange Services client uses three different databases, two of which can have multiple occurrences.

The three databases are:

- Data Interchange Services client database

  A Data Interchange Services client database contains build time data and runtime data. This database is used *only* by Data Interchange Services client. It is used to develop and maintain maps, document definitions, and runtime objects such as control strings, code lists, translation tables, User Exit profiles, and global variables.

  Runtime data is moved from this database to a corresponding Document Manager database using export and import functions.

- Configuration database

  The Configuration database is used *only* by Data Interchange Services client. It contains information used to access the other databases. The information includes queries, the message log, preference information, and information about the other databases defined to Data Interchange Services client. It can

reside on a local system or a remote system. The Configuration database can be a single-user database or it can be a shared database.

- Document Manager database

  Data Interchange Services uses WebSphere Partner Gateway's Document Manager database (BCGAPPS is the default database) when it is translating documents. The Document Manager database contains *only runtime data*. Maps and document definitions are not kept in this database.

  Runtime data is moved to this database from a Data Interchange Services client database using export and import functions. Data Interchange Services client can view and manipulate the data contained in this database.

  The translator component of WebSphere Partner Gateway typically accesses only the Document Manager database.

The import to the Document Manager database is achieved using the provided tools which will be outlined in the chapters detailing our mapping.

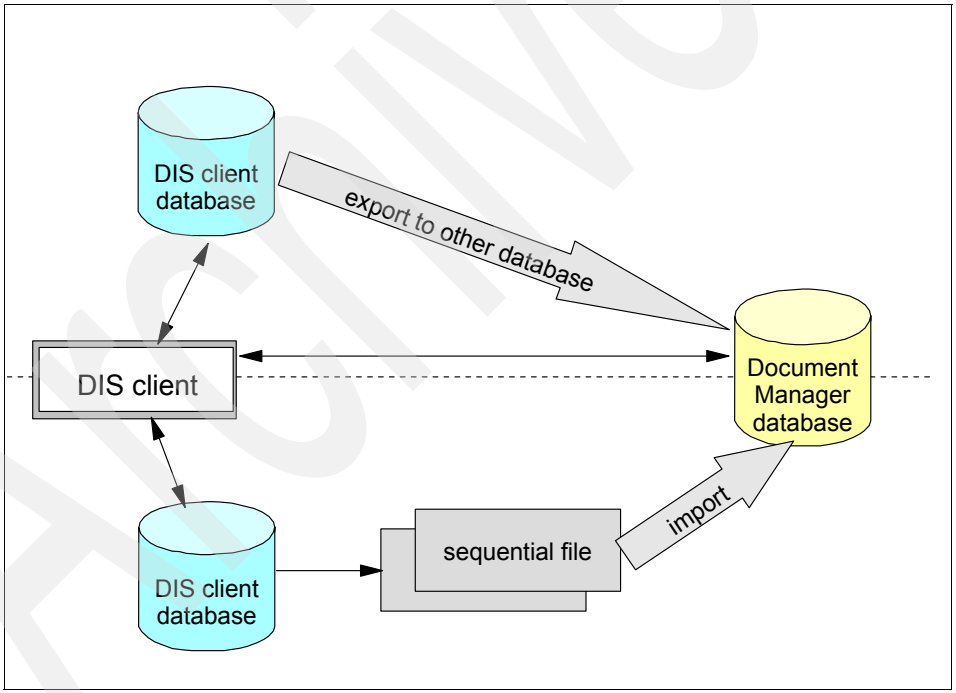


Figure 14-18 From development to production (runtime)

Archived

Archived

15

Mapping

By using the Data Interchange Services client, you can create or import document definitions for the source and target documents, and then create a map which relates the elements in the source document to elements in the target document. Data Interchange Services client permits any combination of source and target document types in a map. For example, you can create a map which transforms an EDI document into an XML document, or an ROD document into an EDI document. Similarly, EDI-to-EDI and XML-to-XML maps are possible, along with the other combinations.

A Data Interchange Services client data transformation map (DT map) relates a source document to a target document. In the Data Interchange Services client, there are three types of documents: EDI, XML, and ROD. An EDI document can be a transaction such as a purchase order, or an invoice in a format defined by a standards body such as X12 or EDIFACT. An XML document can be any type of information encoded using XML tags according to an XML schema or DTD. A Record Oriented Data (ROD) document can be one of many types of data processing application formats.

© Copyright IBM Corp. 2005. All rights reserved.

15.1 Scenario overview

The scenario in this chapter sees WebSphere Partner Gateway translate a proprietary Record Oriented Data (ROD) document to an XML message as defined by an XML Schema. The documents involved in translation are first defined in the Data Interchange Services client. When the Data Interchange Services client has a definition of each document, a data transformation map can be built and exported to WebSphere Partner Gateway. The data transformation map will be built in the default development environment of the Data Interchange Services client.

Figure 15-1 shows that ROD documents are generated by a back-end application and are placed in a directory for use by WebSphere Partner Gateway. WebSphere Partner Gateway retrieves the ROD document and translates the message into an XML format as defined in the data transformation map.

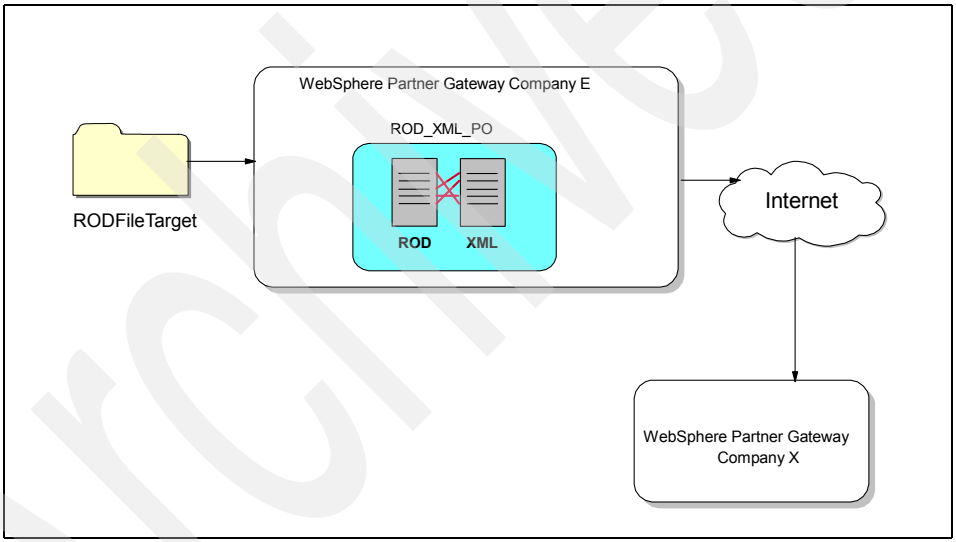


Figure 15-1 Transformation between ROD and XML

Archived

15.2 Create a ROD document definition

In this section we detail the steps involved in creating a new ROD document definition in the Data Interchange Services client. A ROD document definition typically represents a business document or transaction, for example a purchase order or invoice. The way in which this document definition is defined tends to be proprietary and specific to the particular application that generates the ROD document. Before attempting to create a new ROD document in the Data Interchange Services client the precise details of the ROD definition should be known. These details should be made available from the application that generates the data.

In our example, the ROD document we define in the Data Interchange Services client is a purchase order document. A definition of the records and fields that define this document is shown in Table 15-1.

Table 15-1 ROD document definition

Record Name	Field Name	Field Length
HEADER		
	REC_ID	3
	DATE	10
	ORDER_NUMBER	10
	SUPPLIER_NUMBER	10
	CUSTOMER_NUMBER	10
BODY		
	REC_ID	3
	LINE_NUMBER	10
	QUANTITY	10
	PRICE	10
	DESCRIPTON	10
	PART_NUMBER	10
	QTY_COMMITTED	10
FOOTER		
	REC_ID	3

Archived

	TOTAL_QUANTITY	10
	TOTAL_PRICE	10

The steps involved in creating a new Record Oriented Data document include:

1. Create a new ROD dictionary
2. Create a new Record ID object
3. Create a new Record Oriented Document definition.

15.2.1 Create a new ROD dictionary

Follow these steps to create an ROD dictionary definition in the Data Interchange Services client using the detail in Table 15-1 on page 475:

1. Open the Record Oriented Data functional area by clicking the **ROD** icon.
2. Select the **ROD Dictionaries** tab to display a list of existing Record Oriented Data dictionaries.
3. Select **File** → **New** from the Data Interchange Services client.
4. Enter a unique dictionary name. In the example provided with this book, we named the this dictionary **CUSTOM_ROD**.
5. Save the new dictionary by selecting **File** → **Save**.

15.2.2 Create a new Record ID

Now that we have created a new ROD dictionary, we must create a new Record ID object that will be used to identify records in our ROD document definition.

To create a new Record ID object, follow these steps:

1. Select the **Record ID Information Objects** tab from the Data Interchange Services client Record Oriented Data functional area.
2. Select **File** → **New**.
3. Give the Record ID Information object a unique name. In our example, we called the record ID to be used with our ROD document definition `RECORD_ID`.
4. Ensure that **Raw Data** is checked.
5. Enter a **Position** value of `1`. This is character position within the record that signified the start of a record id value.
6. Enter a length value of `3`. This identifies our Record ID object as being three characters in length.
7. Enter the data type as **AN = Alphanumeric.**

8. Save the new Record ID Information Object. Click **File** → **Save,** then **File** → **Close**.

15.2.3 Create a new Record Oriented Document definition

Having created our ROD Dictionary and Record ID Information Object we can begin to create our new ROD document definition. We can see from the definition in Table 15-1 on page 475 that our document definition is comprised of three records. Each of these records is, in turn, comprised of a number of fields and each of these fields has a character type and data length. We therefore begin to create our new document definition by creating each of the required fields with their respective data types.

Create a new field definition

To create a new field definition, follow these steps:

1. Select the **Fields** tab from the Record Oriented Data functional area.
2. Select **File** → **New** from the Data Interchange Services menu bar.
3. Enter the following required data:
   - Field Name
   - Dictionary Name
   - Data Type
   - Field Length
4. Repeat the steps above for each of the field definitions in the table Table 15-1 on page 475. For example, to create a `REC_ID` field definition:
   a. Open the Record Oriented Data functional area from the Data Interchange Services client.
   b. Click the **Fields** tab.
   c. Select **File** → **New** from the menu bar.
   d. Enter the Field Name as `REC_ID`**.**
   e. Select the dictionary name **CUSTOM_ROD** from the **Dictionary Name** drop-down menu.
   f. Select the Data Type as **AN = alphanumeric.**
   g. Enter the **Field Length** as 3. Your new field definition should appear as shown in Figure 15-2 on page 478.

Archived

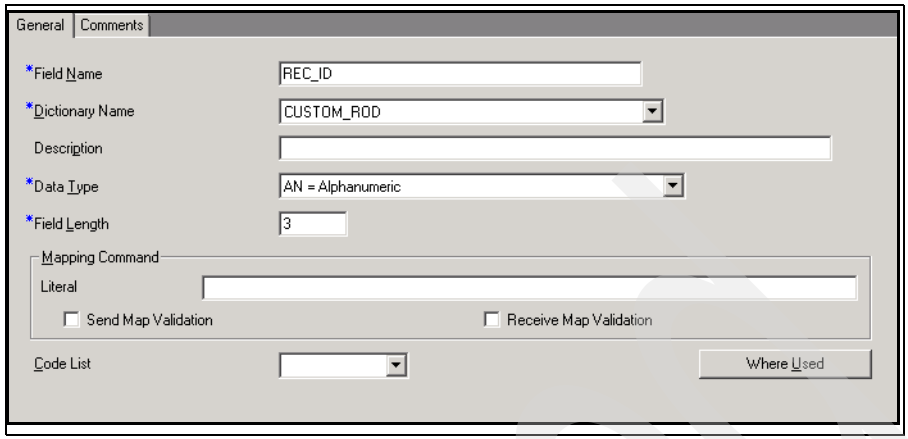


Figure 15-2 New field definition

5. Finally save and close the new field. Select **File** → **Save** then **File** → **Close** from the Data Interchange Services client menu bar.

After creating all of the field definitions listed in table Table 15-1 on page 475, our next step is to create the record definitions.

Create a new Record definition

To create a new record, follow these steps:

1. Select the **Records** tab from the Record Oriented Data functional area.
2. Select **File** → **New** from the menu bar.
3. Provide information for each of the mandatory fields:
   a. Record Name
   b. Dictionary Name
   c. Record ID Information
4. Optionally enter information about the **Record Identifier**.
5. Save and close the new ROD Record.

In step 4, we mention the Record Identifier. This record identifier is a means of uniquely identifying a record definition when a document is being processed by the Data Interchange Services component of WebSphere Partner Gateway. The record identifiers for each of the records listed in the Table 15-1 on page 475 are shown in Table 15-2 on page 479.

Table 15-2 Record identifiers

Record Name	**Record Identifier**
HEADER	HED
BODY	BOD
FOOTER	FOT

After the ROD Record definition is created, we can proceed to define each of the associated fields with the record.

We take the example of creating a ROD Record definition for the `HEADER` record as defined in Table 15-1 on page 475. To create the ROD Record definition, follow these steps:

1. Open the Record Oriented Data functional area.
2. Select the **Records** tab.
3. Chose **File** → **New** from the menu bar.
4. Enter the Record Name as `HEADER`.
5. Select the Dictionary Name **CUSTOM_ROD.**
6. Select the Record ID Information as **RECORD_ID.**
7. Enter the Record Identifier as `HED`. The record identifier value is taken from Table 15-2. Your new ROD Record definition should appear as shown in Figure 15-3.

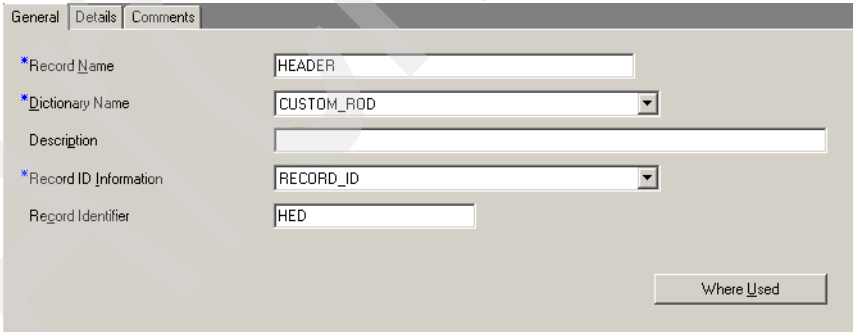


Figure 15-3 New ROD record definition

8. Save and close the new ROD record definition.

Our next step is to define the fields associated with the HEADER record.

1. From the Record Oriented Data functional area, select the **Records** tab and locate the entry with a Record Name HEADER.
2. Double click the **HEADER** table entry to display an empty **Details** tab as shown in Figure 15-4.

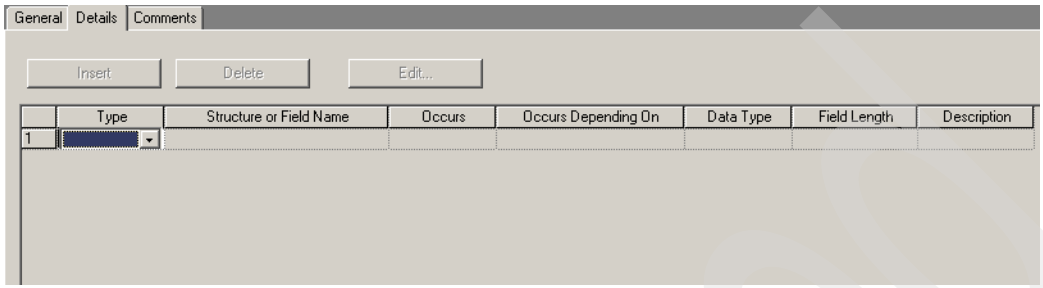


Figure 15-4 Empty details

3. Select **Field** from the **Type** drop-down menu.
4. Click the **Structure or Field Name** column to display a drop-down menu of field definitions available for selection.
5. Select the **REC_ID** field from the drop-down selection, defined as the first field in the HEADER record in Table 15-1 on page 475. The Occurs, Data Type and Field Length cells should be completed automatically with the data defined for the particular field. A new row will appear in the details table.
6. Repeat the steps above selecting the Type as **Field** and the Structure or Field Name as **DATE**. Perform this action for both the SUPPLIER_NUMBER and CUSTOMER_NUMBER fields.
7. When complete, the **Details** tab should appear as shown in Figure 15-5.

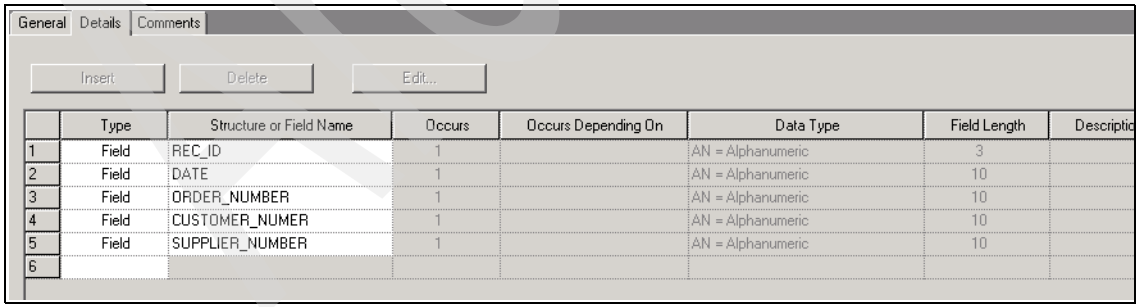


Figure 15-5 HEADER record definition

8. The **General** tab of the HEADER record should appear as shown in Figure 15-6 on page 481.

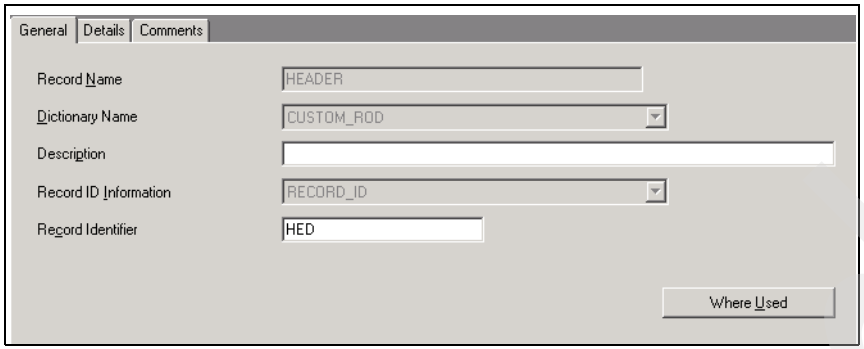


Figure 15-6 General tab of the HEADER record

9. Save and close the HEADER record definition.

The preceding steps should be completed for both the BODY and FOOTER records. After all records have been defined in the Data Interchange Services client, we can create the final object that will define our purchase order document, the ROD document definition.

To create a new ROD document definition, follow these steps:

1. Select the **ROD Document Definitions** tab from the ROD functional area.
2. Chose **File** → **New** from the Data Interchange Services client menu bar. The ROD Document Definition Editor opens on the **General** tab page as shown in Figure 15-7.

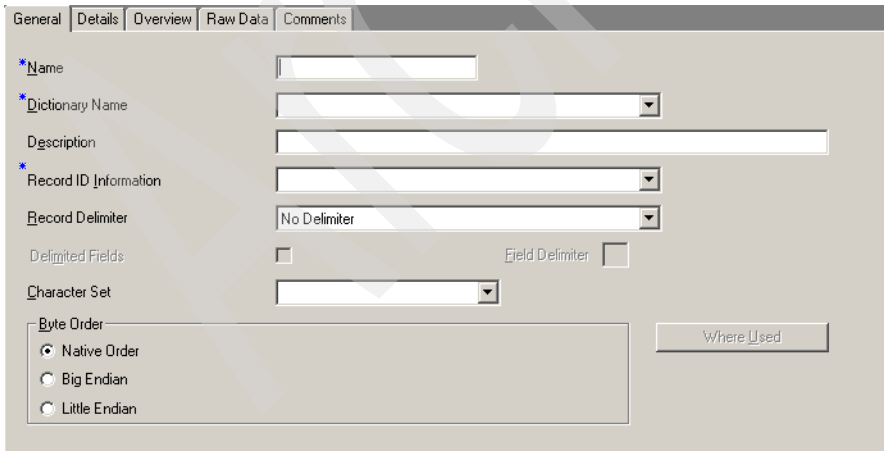


Figure 15-7 General tab page

In creating our new ROD document definition, we will pay particular attention to three of the tab pages displayed:

- General tab page
- Details tab page
- Raw Data tab page.

The **General** tab page holds basic information about the ROD document definition, such as the name of the ROD document definition, the name of the dictionary it is a part of, and other information identifying the type of ROD document definition.

Follow the steps below to complete this page for our new document definition:

1. Enter the Name as `PURCHASE_ORDER`.
2. Select the dictionary name **CUSTOM_ROD** from the Dictionary Name drop-down menu.
3. Select **RECORD_ID** from the Record ID Information.
4. Chose **Carriage Return/Line Feed** from the Record Delimiter menu. The **General** tab should appear as shown in Figure 15-8.

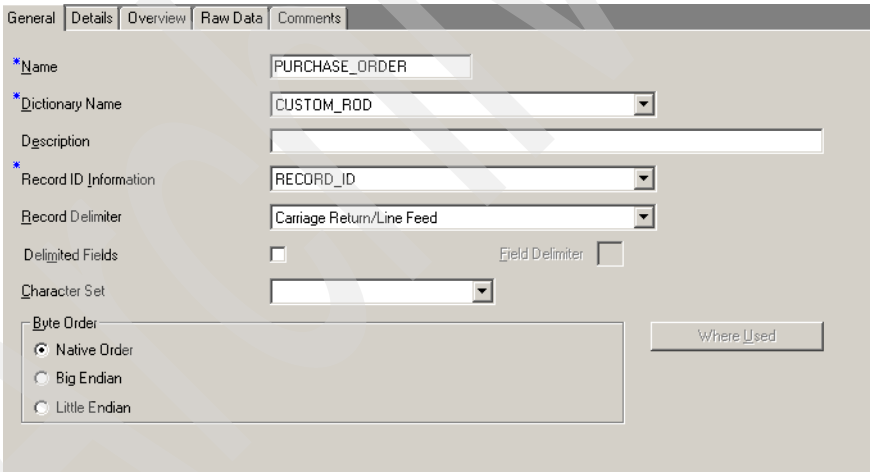


Figure 15-8 General tab of the PURCHASE_ORDER document definition

The **Details** tab page of the ROD Document Definition Editor identifies the first level of components that define the layout of the document represented by the ROD Document Definition Format. The first level of the document layout can contain Loops and Records. Loops, in turn, contain Records and other Loops. Records contain Structures and Fields. In our purchase order document, the first

level of components will be record definitions. These will be the record definitions created earlier.

To define these record definitions, follow these steps:

1. Set the Type as `Record` in the first row displayed.
2. Select **HEADER** as the Loop or Record Name.
3. The Area, Maximum Repeat and Record ID columns will be completed automatically.

Repeat this set of steps for both the `BODY` and `FOOTER` records. The `BODY` record should be entered on the second row of the display table. Enter the `FOOTER` record on the third row of the display table.

4. One final step is to amend the Maximum Repeat value automatically entered for the `BODY` record. We want the `BODY` record to repeat more than once in our document definition. To allow the `BODY` record repeat an unlimited number of times, select the check box in the **Unlimited Repeat** column for the `BODY` record entry as shown in Figure 15-9.

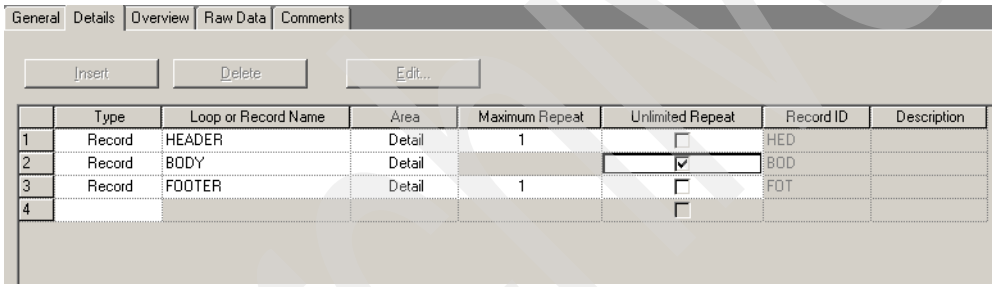


Figure 15-9 Details tab of the PURCHASE_ORDER document definition

The final tab we need to address before saving our new document definition is the **Raw Data** tab. This raw data tab page on the ROD Document Definition Editor allows you to indicate which fields in the document will contain important information that can be used in a translation by WebSphere Partner Gateway.

The important information we need to capture here is:

1. The beginning record
2. The field containing the Interchange Sender ID
3. The field containing the Interchange Receiver ID.

The steps required to complete the raw data tab for the PURCHASE_ORDER document definition are as follows:

1. Define the beginning record as **HEADER**.

2. Define the field containing the Interchange Sender ID as **CUSTOMER_NUMBER.**
3. Define the field containing the Interchange Receiver ID as **SUPPLIER_NUMBER.** The Details tab of the PURCHASE_ORDER document definition should appear as shown in Figure 15-10.

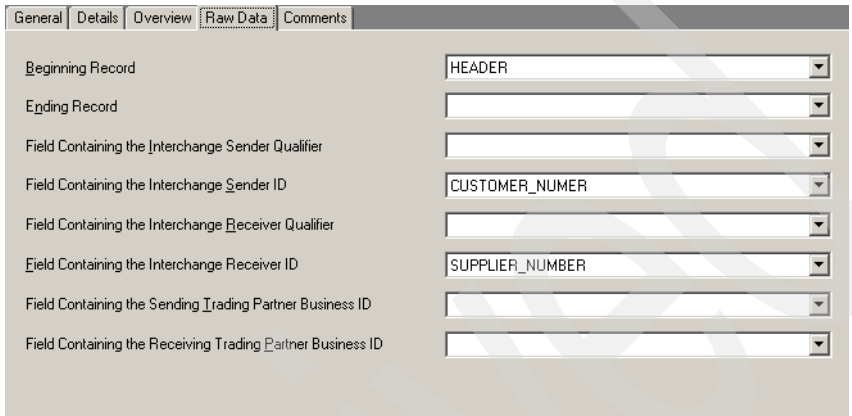


Figure 15-10 Details tab of the PURCHASE_ORDER document definition

After completing the Raw Data tab page, we save the new document definition. This ROD document definition is now available for use in a data transformation map. Before proceeding with the creation of our data transformation map, we must first import the target XML schema that will be used in our data transformation map.

15.3 Import XML Schema

The Data Interchange Services client provides the functionality to import both XML Schema and XML DTD. In our scenario, we are creating an ROD to XML data transformation map. The XML document type used in our map is best defined by an XML Schema. To work with this schema in a data transformation map we must first import the definition.

Before importing an XML Schema we must first create an XML Dictionary that is used to logically group a set of XML Schema or XML DTD.

To create a new XML Dictionary, follow these steps:

1. Open the XML functional area by clicking the **XML** icon in the Data Interchange Services client.
2. Select the **XML Dictionaries** tab.

3. Chose **File** → **New** from the menu bar to display the XML Dictionary Editor.
4. Provide a Dictionary Name as shown in Figure 15-11. This name should be unique among the set of XML Dictionaries already defined in the Data Interchange Services client. For the purpose of this example, we have named the XML Dictionary `CUSTOM_XML`.

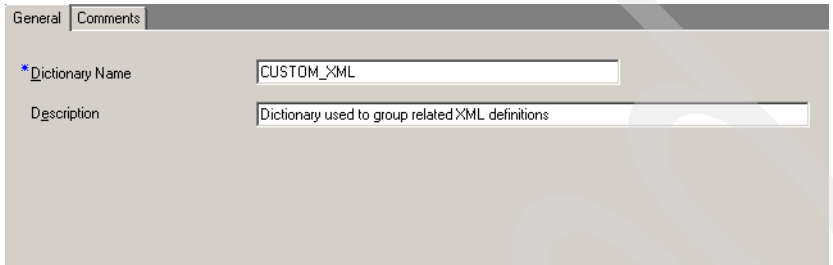


Figure 15-11 Create a new XML Dictionary

5. Save the new XML dictionary and close the XML Dictionary Editor.

To import our XML Schema definition into our newly created XML Dictionary, follow these steps:

1. Select **File** → **Open Import File** from the Data Interchange Services client, as shown in Figure 15-12. The Select Import File window opens.

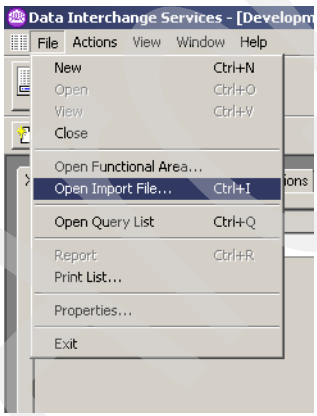


Figure 15-12 Open a new import file

2. Change the Files of Type option to **XML Schema File (*.xsd).**
3. Navigate to the directory that contains the XML Schema to be imported, select the file, then click **Open**. The Import XML Schema window opens and

displays a Document Definition Name and Dictionary Name as shown in Figure 15-13.

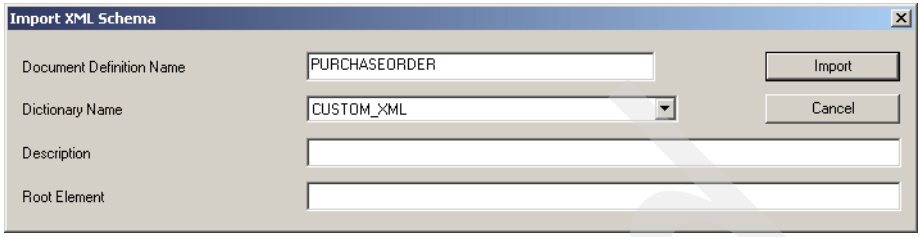


Figure 15-13 Import XML Schema

The **Document Definition Name** that is displayed is taken from the root tag as defined in the XML Schema. Replace this name with `PURCHASE_ORDER`.

4. The Dictionary Name displayed is the name of the only XML Dictionary defined in our database and should display as CUSTOM_XML. This can remain unchanged.
5. Enter the Root Element value of `purchaseOrder` as shown in Figure 15-14. The entry is case sensitive and defines the root element of XML files that conform to this schema definition.

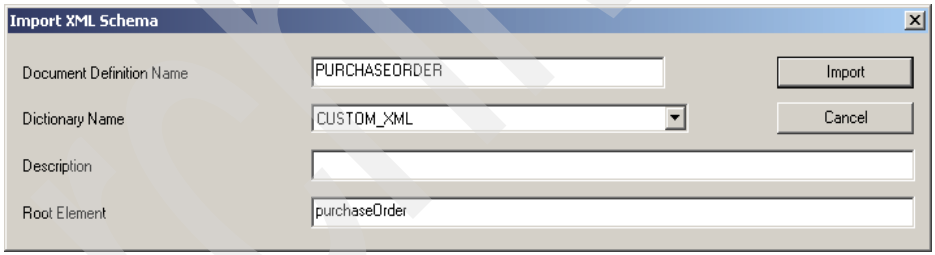


Figure 15-14 Enter a root element

6. Click **Import**.
7. The Data Interchange Services client informs us that a namespace entry will be created for the dictionary CUSTOM_XML (Figure 15-15). Click **OK**.

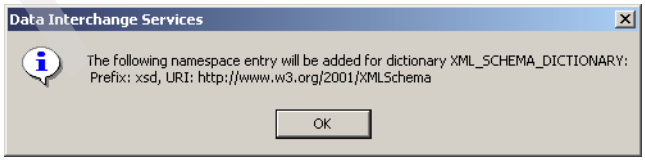


Figure 15-15 A new namespace entry will be created

8. Click **OK** when you are informed the file has been imported successfully (Figure 15-16).

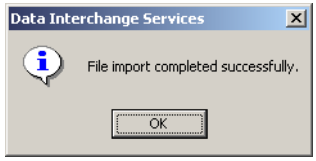


Figure 15-16 Successful import

9. We can view the new Schema Document Definition by selecting the **Schema Document Definitions** tab from the XML functional area. If no entry is displayed, refresh the view by choosing **View** → **Refresh** (Figure 15-17).

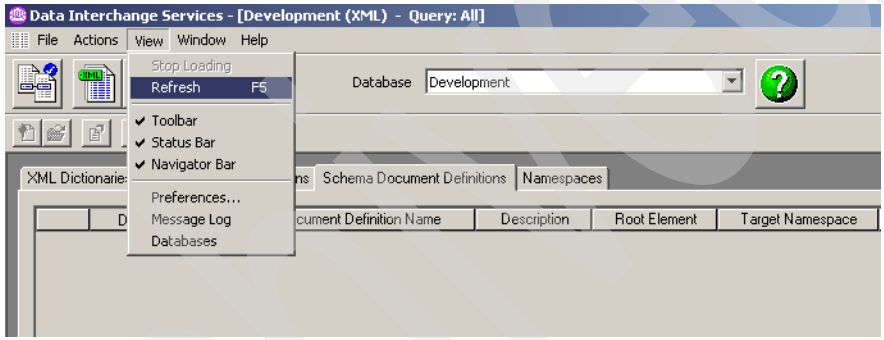


Figure 15-17 Refresh view

The contents of the XML schema we have just imported are shown in Example 15-1.

Example 15-1 Contents of XML schema

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<xsd:element name="purchaseOrder" type="order" />

   <xsd:complexType name="order">
      <xsd:sequence>
         <xsd:element name="orderHeader" type="header"/>
         <xsd:element name="lineDetail" type="detail" minOccurs="0" maxOccurs="unbounded" />
         <xsd:element name="orderTrailer" type="trailer" />
      </xsd:sequence>
   </xsd:complexType>
```

Archived

```
   <xsd:complexType name="header">
      <xsd:sequence>
         <xsd:element name="customerNumber" type="xsd:string"/>
         <xsd:element name="orderNumber" type="xsd:string"/>
         <xsd:element name="orderFlag" type="xsd:string"/>
         <xsd:element name="orderDate" type="xsd:string"/>
         <xsd:element name="contractNumber" type="xsd:string"/>
         <xsd:element name="supplierNumber" type="xsd:string"/>
         <xsd:element name="supplierName" type="xsd:string"/>
         <xsd:element name="supplierAddress1" type="xsd:string"/>
         <xsd:element name="supplierAddress2" type="xsd:string"/>
         <xsd:element name="supplierAddress3" type="xsd:string"/>
         <xsd:element name="buyerName" type="xsd:string"/>
         <xsd:element name="buyerSerial" type="xsd:string"/>
         <xsd:element name="currencyCode" type="xsd:string"/>
         <xsd:element name="paymentTerms" type="xsd:string"/>
         <xsd:element name="deliveryAddressName" type="xsd:string"/>
         <xsd:element name="deliveryAddress1" type="xsd:string"/>
         <xsd:element name="deliveryAddress2" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>

   <xsd:complexType name="detail">
      <xsd:sequence>
         <xsd:element name="lineNumber" type="xsd:string"/>
         <xsd:element name="partNumber" type="xsd:string"/>
         <xsd:element name="itemDesc" type="xsd:string"/>
         <xsd:element name="quantity" type="xsd:string"/>
         <xsd:element name="uom" type="xsd:string"/>
         <xsd:element name="unitPrice" type="xsd:string"/>
         <xsd:element name="dueDate" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>

   <xsd:complexType name="trailer">
      <xsd:sequence>
         <xsd:element name="totalDollars" type="xsd:string"/>
         <xsd:element name="totalQuantity" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>

</xsd:schema>
```

Archived

15.4 Create a data transformation map

To create a new Data Transformation Map, follow these steps:

1. Open the Mapping functional area of the Data Interchange Services client and select the **Data Transformation Maps** tab.
2. Chose **File** → **New** from the menu bar. The Create a Data Transformation Map - Map Name window opens.
3. Provide the Map Name as `ROD_XML_PO`.
4. The map name should be unique among the set of maps already defined in the Data Interchange Services client database. You can check existing map names by clicking **Show Existing Map Names**.
5. Enter a map name and description.
6. Enter a description for the new map and click **Next**. The Create a Data Transformation Map - Source or Target window opens.
7. Our map is going to be defined as a target-based map. A description of each map type is provided in the window. Select **Target** and click **Next** (Figure 15-18).

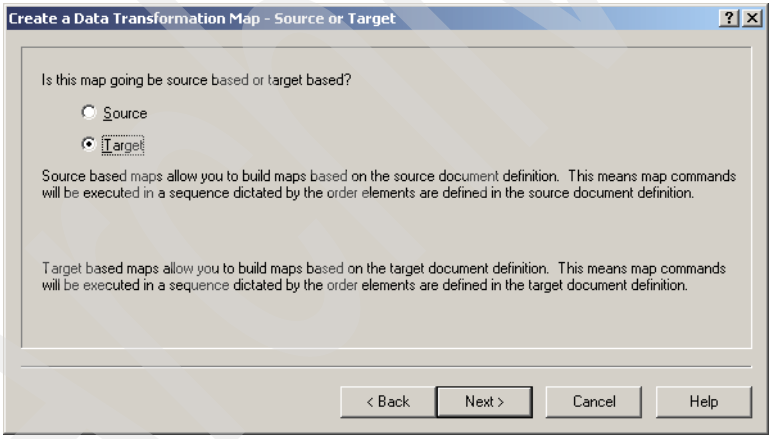


Figure 15-18 Specify source or target

8. The syntax type of our source document is ROD. Select **Record Oriented Data** (Figure 15-19).

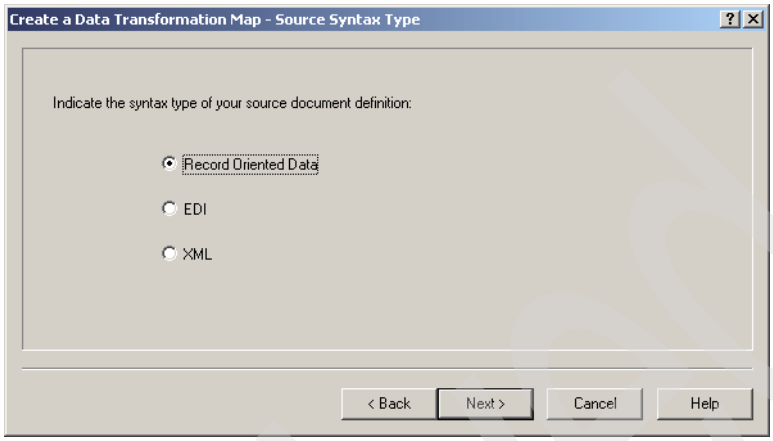


Figure 15-19 Indicate syntax type

9. Click **Next**. The Create a Data Transformation Map - Source Dictionary window opens.
10. Select **CUSTOM_ROD** and click **Next** (Figure 15-20).

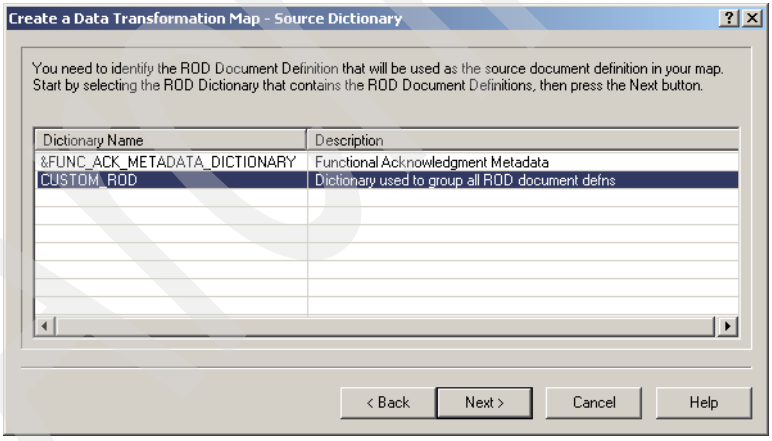


Figure 15-20 Select source dictionary

11. Select **PURCHASE _ORDER** from the Create a Data Transformation Map - Source ROD Document Definition window (Figure 15-21).

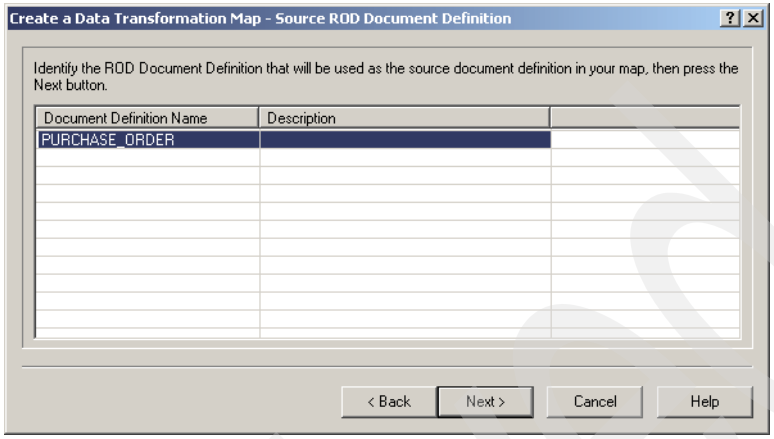


Figure 15-21 Select source ROD document definition

12. Click the **Next** button to display the Create a Data Transformation Map - Target Syntax Type window.
13. Select **XML** (Figure 15-22) and click **Next**.

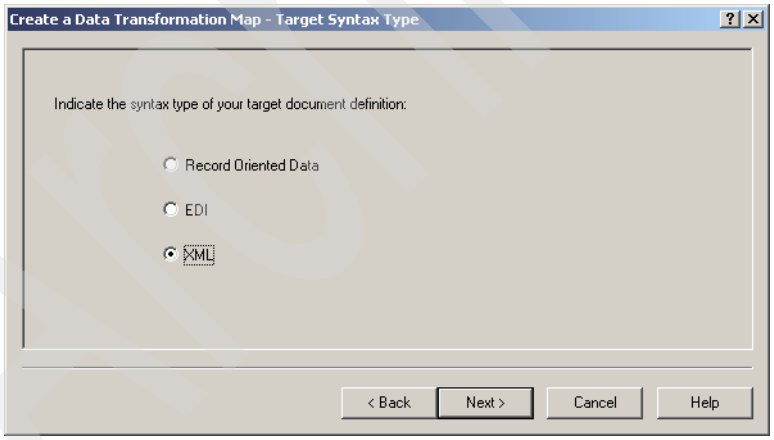


Figure 15-22 Select target syntax type

Archived

14. Select **CUSTOM_XML** and click **Next** (Figure 15-23).

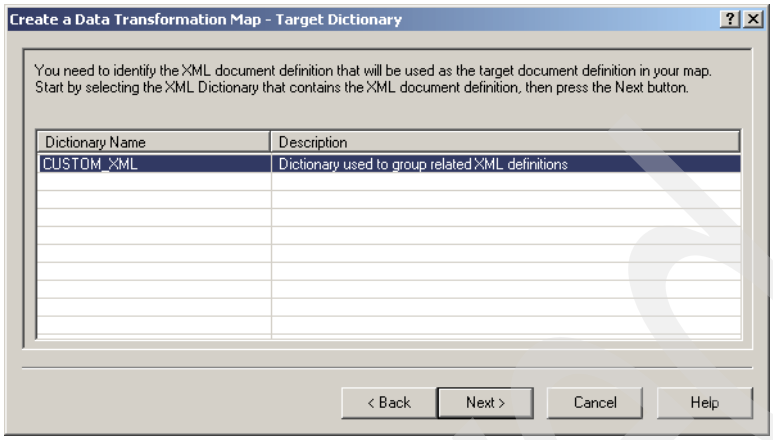


Figure 15-23 Select target dictionary

15. Select **PURCHASE_ORDER** and click **Next** (Figure 15-24).

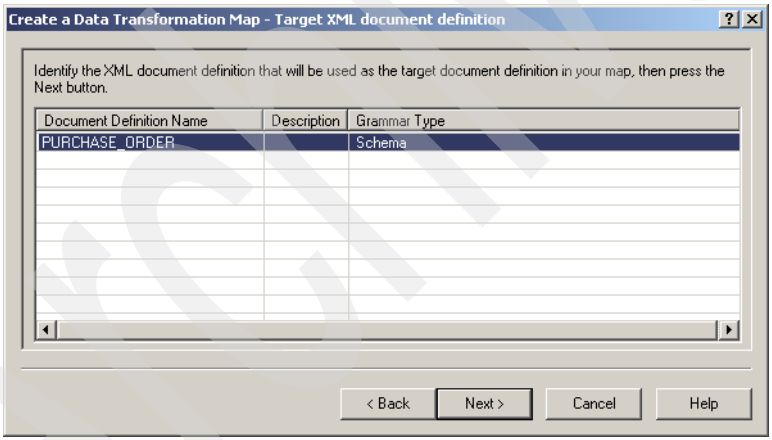


Figure 15-24 Select XML document definition

Archived

16. The Create a Data Transformation Map - Confirmation window opens. Review the information displayed in this window and, if it is correct, click **Finish** (Figure 15-25).

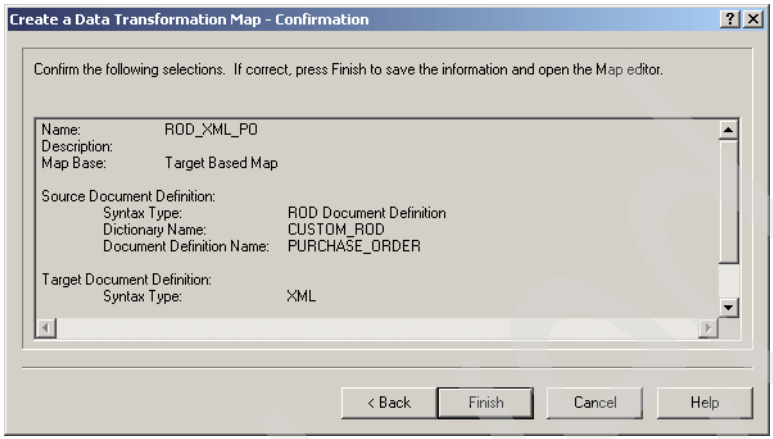


Figure 15-25 Confirm selections

17. The Data Transformation Map Editor opens, as shown in Figure 15-26.

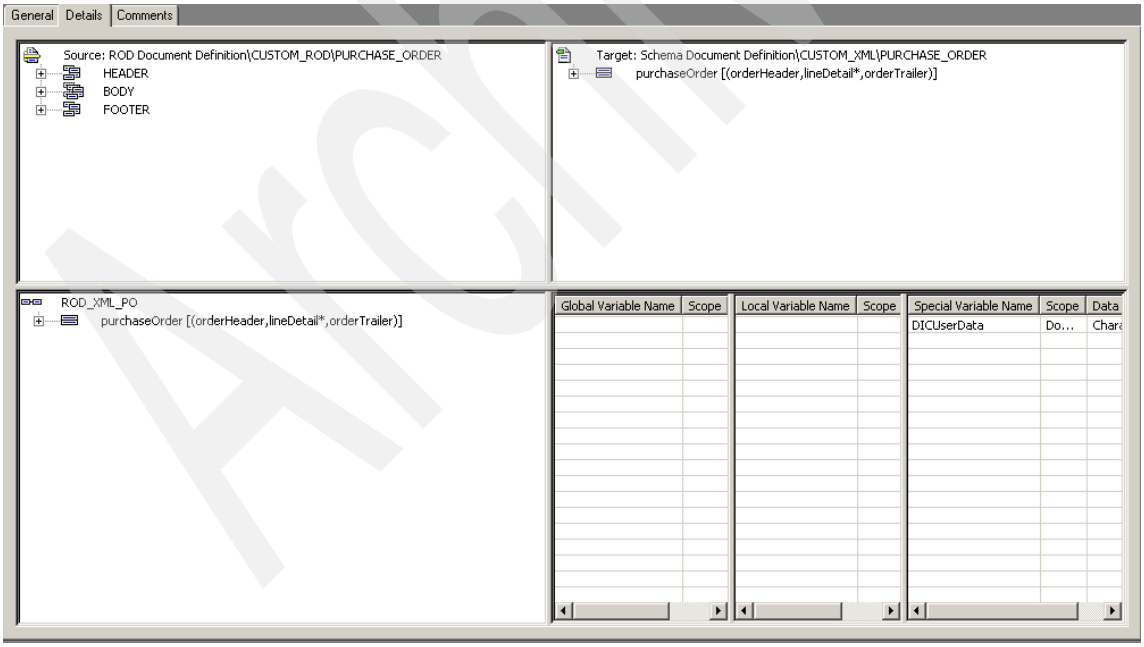


Figure 15-26 Map editor

Archived

The Details tab of the Data Transformation Map Editor is where mapping is performed. Mapping creates mapping commands that instruct Data Interchange Services how to convert a document in one format to a document in another format. Almost all work on a Data Transformation Map will occur on this tab page.

There are four panes contained in the Details tab page:

1. Source Document Definition window

   The Source Document Definition window is located in the upper left corner of the Details tab page. It displays the document definition for the layout of the source document. The *source document* is the document that will be translated into another format. Use the Source Document Definition window as a reference of the source document layout and to map elements in the source document definition to elements in the target document definition and to variables.

2. Target Document Definition window

   The Target Document Definition window is located in the upper right corner of the Details tab page. It displays the document definition that describes the layout of the target document. The target document is the document that will be produced from the translation of the source document. Use the Target Document Definition window as a reference of the target document layout and to map elements in the source document definition to elements in the target document definition and to map variables to elements in the target document definition.

3. Mapping Command window

   The Mapping Command window is located in the lower left corner of the Details tab page. If the map is source based, the Mapping Command window displays a representation of the source document definition. If the map is target based, the Mapping Command window displays a representation of the target document definition. The representation is similar to the base document definition, but it may not be exactly the same. Use this window to create and maintain mapping commands. During translation of a document Data Interchange Services will execute mapping commands in the order they are encountered in the Mapping Command window. In source based maps, mapping commands are executed only when the parent element of a mapping command is encountered while reading the source document. Elements in the Mapping Command window can be mapped to variables and to elements in the Source Document Definition window or Target Document Definition window, depending on whether the map is source or target based

4. Variables window

   The Variables window is located in the lower right corner of the Details tab page. It is divided into three window panes, one for each type of variable. Use the Variables window to manage Local Variables and to map with Global Variables, Local Variables, and Special Variables.

Each of the four windows contained on this tab page can be sized. To resize a window, click a window divider and drag it to the desired position. Note that changing the size of a window affects the size of other windows on the tab page.

Now that we have explained each of the four windows displayed on the Details tab page of the Data Transformation Map Editor we can build the mapping commands. These commands define the relationship between the source and target document definitions and ultimately shape the result of translation.

The most simple mapping command to define is the MapFrom() command. It is a simple drag and drop action.

1. Expand the tree structures in both the Source Document Definition window and the Target Document Definition window. Both the source elements and target elements should be visible.
2. Drag the source field ORDER_NUMBER and drop it onto its corresponding target element orderNumber. The results of this simple drag and drop action are shown in the Mapping Command window. Essentially, we have created a MapFrom() function (Figure 15-27 on page 496).

Archived

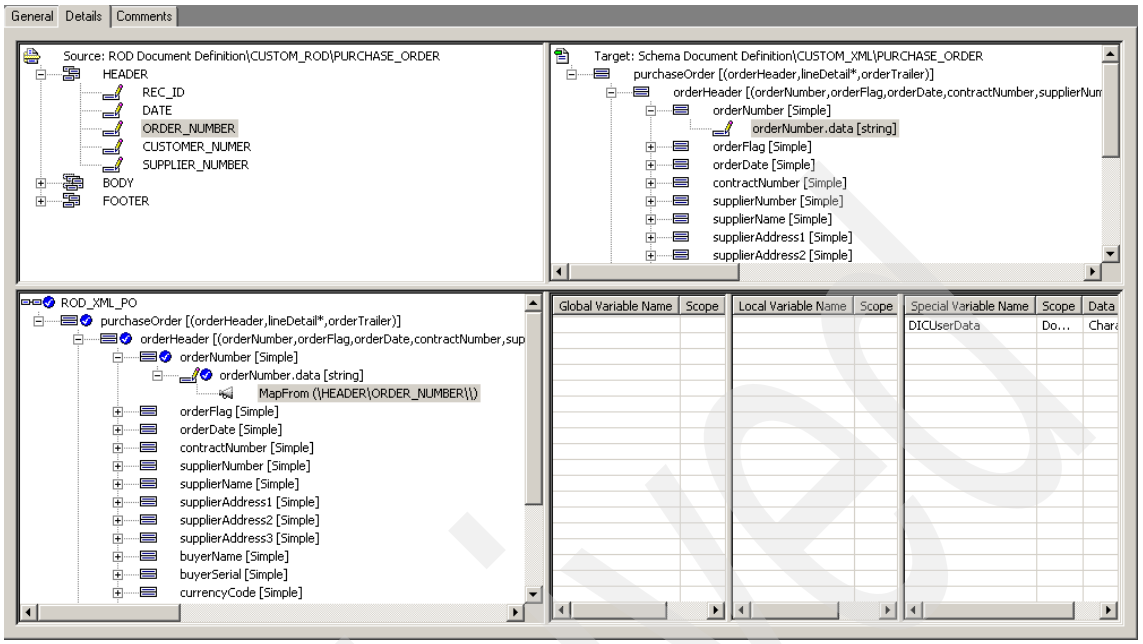


Figure 15-27 Mappings

3. In addition to moving element values from source elements to target elements, a target element can also be assigned a specific value that might not originate from the source document. As an example, we can populate the XML target element orderDate with the system date by invoking the built-in Date() function of Data Interchange Services.
4. Right-click the element **orderDate** in the Mapping Command window and select **Insert Within Command Assignment**.
5. The Mapping Command Editor window opens, displaying a template defining structure of a new mapping command Figure 15-28.

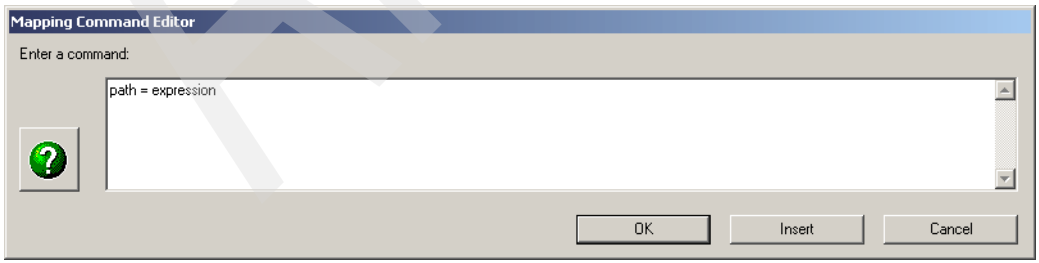


Figure 15-28 Mapping Command Editor

6. Drag the target element from the Target Document Definition window and drop it onto the Mapping Command Editor. You should drop the target element on the `path` value displayed in the editor. The Data Interchange Services client then generates the correct access path for you.
7. To add the Date() function to our mapping command, simply highlight the expression value shown in the Mapping Command Editor, right-click and select **Functions** → **Date** (Figure 15-29).

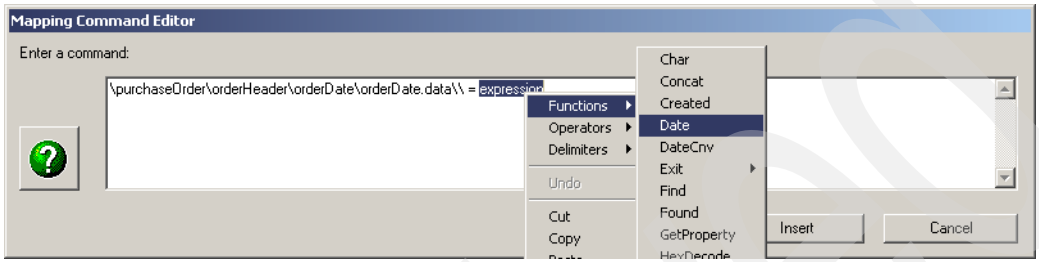


Figure 15-29 Invoke a built in function

8. The Mapping Command Editor should appear as shown in Figure 15-30.

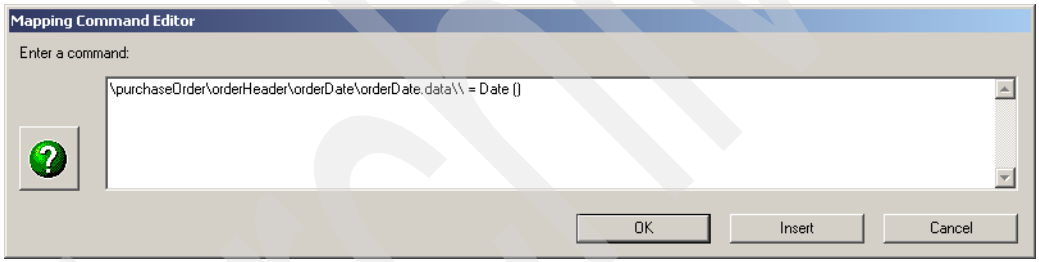


Figure 15-30 Mapping Command Editor

9. Click **OK** to return to the Details tab page of the Mapping Command window.

You will have noticed when right clicking the **expression** value from the Mapping Command Editor and selecting **Functions**, that a number of other predefined functions are available when mapping with the Data Interchange Services client. Although we do not discuss each of these functions in this book, the context-specific Help menu available in the Data Interchange Services client discusses each function in detail. To access the Help information, follow these steps:

1. Select **Help** from the Data Interchange Services client menu bar.
2. Chose **Search for Help On...** to display the Help Topics window.

3. Enter `mapping commands` as the search parameter (Figure 15-31).

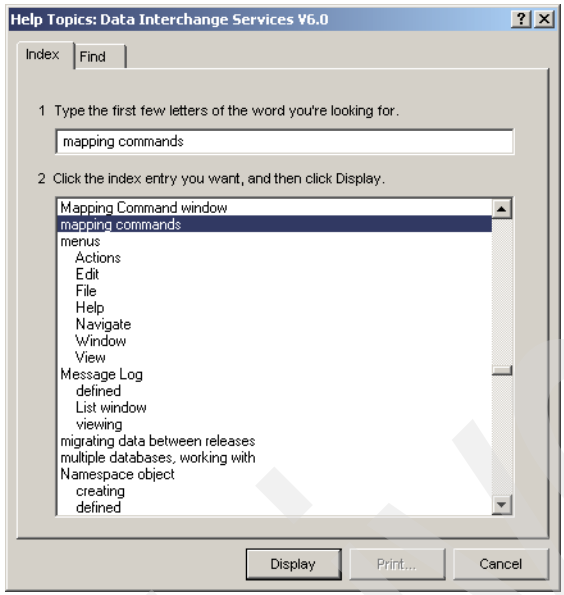


Figure 15-31 Help window

4. Double-click **mapping commands** in the lower window to display Help information about Mapping Commands and Functions as shown in Figure 15-32 on page 499.

Archived

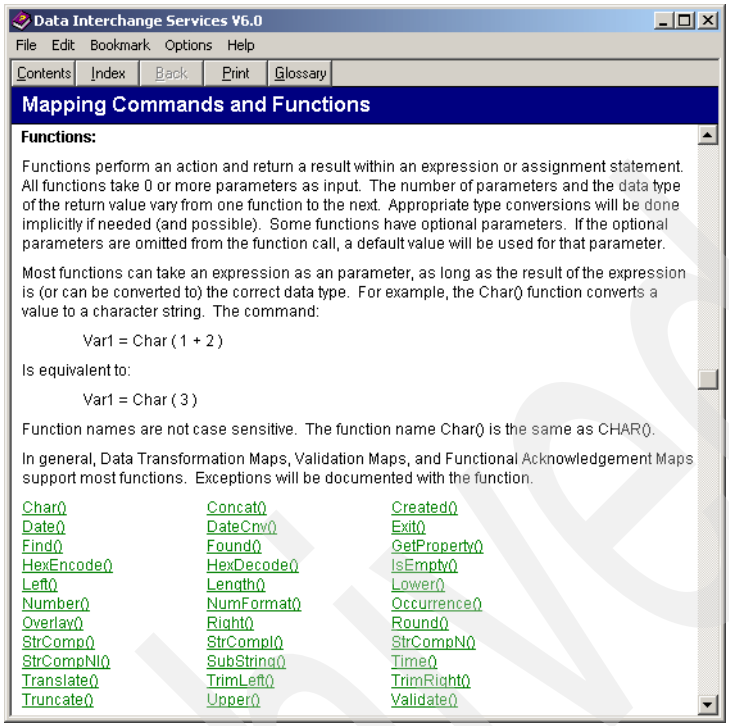


Figure 15-32 Mapping command help

Mapping the supplierNumber and supplierName elements is a bit more complex. When mapping the supplierNumber element we want to store the value mapped as a local variable so that it can be used again later in the map. Essentially, we want to use the value used in this local variable to determine the value assigned to the supplierName element in result of translation.

To achieve this scenario, we first create a local variable in the Data Interchange Services client. We call this new local variable `supplierNum`. To create a new local variable, follow these steps:

1. Right-click in the **Variables** window below the Local Variable Name column (Figure 15-33 on page 500).

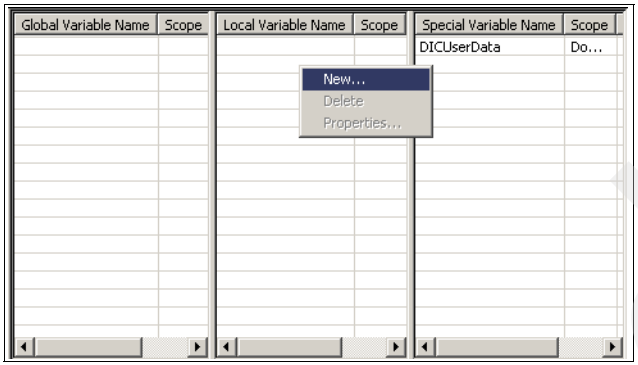


Figure 15-33 Variables window

1. The Create a New Local Variable window opens (Figure 15-34). Give the new local variable a name, for example `supplierNum`. Accept all other default values and click **OK**.

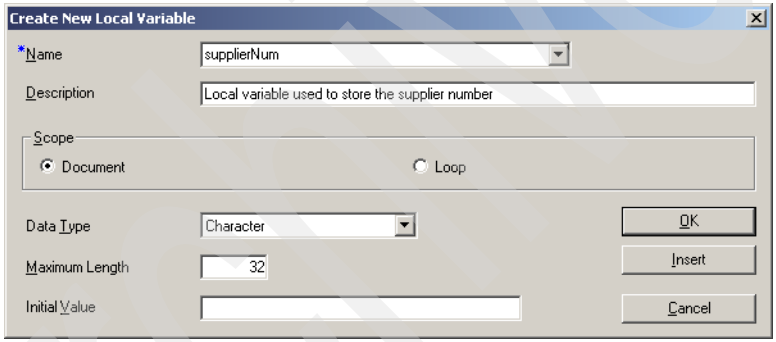


Figure 15-34 Create new local variable

Our next step is to assign a value from the source document definition to our new local variable. Because this step will take more than one mapping command to complete, we can create a new Command Group to logically group the set of commands that complete the assignment.

To create a new Command Group, follow these steps:

1. Right-click the **supplierNumber** element in the Mapping Command window (Figure 15-35) and select **Insert Within** → **Command Group... .**

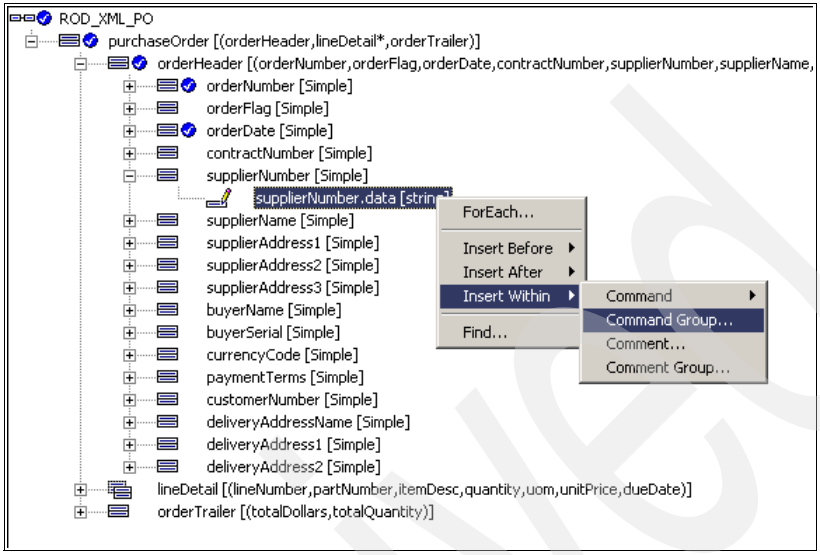


Figure 15-35 Create new command group

2. Enter a brief description of the purpose this command group serves in the Command Group Editor (Figure 15-36) for example, `Assign and trim supplier number using supplierNum variable`. Click **OK** to return to the Mapping Command window.

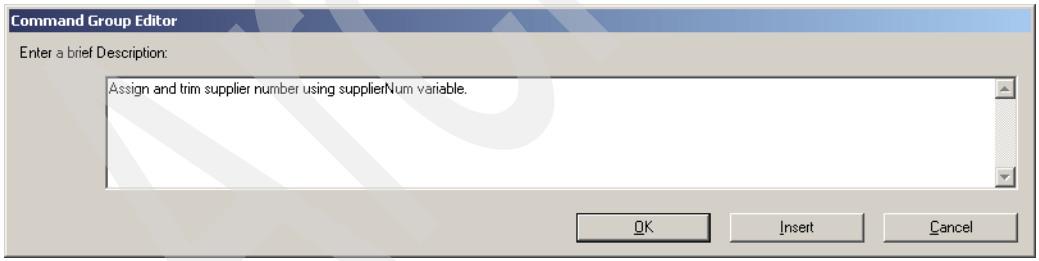


Figure 15-36 Command Group Editor

Now that we have created our command group, we can create the mapping commands that will achieve our goal of assigning a value to the supplierNum variable.

3. Right-click the newly created command group and select **Insert Within** → **Command** → **Assignment** (Figure 15-37 on page 502).

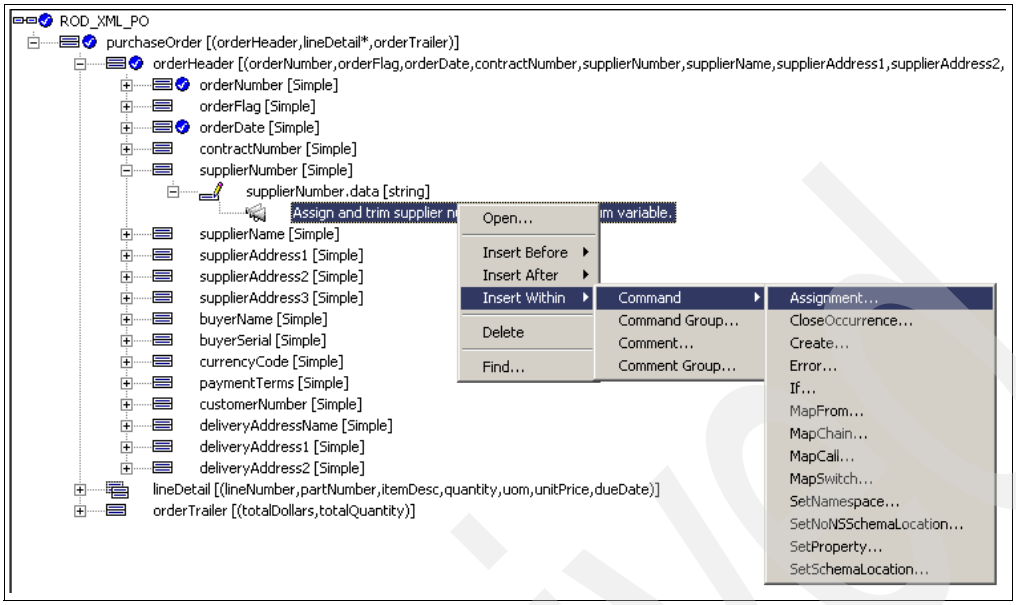


Figure 15-37 Assign variables

4. Assign a value to the supplierNum local variable by opening the Mapping Command Editor on the supplierNumber element in the Mapping Command window.
5. From the Variables Window, select the **supplierNum** local variable and drop it over the path value displayed in the Mapping Command Editor.
6. Select **SUPPLIER_NUMBER** from the Source Document Definition window and drag it over the expression value displayed in the Mapping Command Editor Figure 15-38). Click **OK** to complete the command.

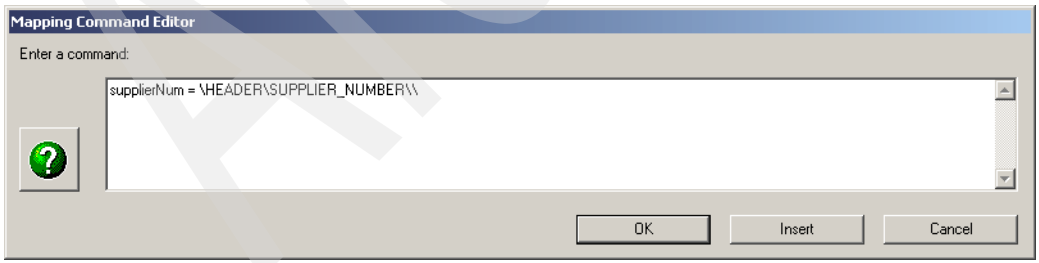


Figure 15-38 Mapping Command Editor

This mapping command alone is enough to assign a value from the source document to our supplierNum local variable. We face an issue with the size of

the SUPPLIER_NUMBER field in the source document however. Because our mapping command literally assigns the value of the SUPPLIER_NUMBER field from the source document to our local variable, the string assigned to supplierNum will be 10 characters in length. The supplier number data received is only 8 characters in length, so we need to invoke the TrimRight() function on our supplierNum local variable to remove any trailing white space. To invoke the TrimRight() function, follow these steps:

7. Right-click the assignment created in the previous step.
8. Chose **Insert After Command Assignment**.
9. Drag supplierNum from the local variables column of the Variables window over the path value in the Mapping Command Editor.
10. Select the expression value from the Mapping Command Editor, then right-click and select **Functions** → **TrimRight** → **TrimRight(charValue)**.
11. Drag supplierNum from the local variable column of the Variables window and drop it over the charValue string in the Mapping Command Editor. When complete, your mapping command should appear as shown in Figure 15-39.

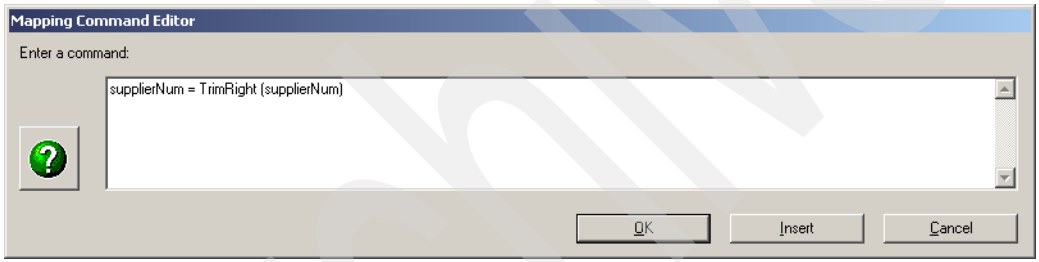


Figure 15-39 Mapping Command Editor

12. Click **OK** to complete the mapping command and return to the Mapping Command window.
13. Finally, we want to map the supplierNumber element itself in the target document definition. We can do this using the MapFrom() function.
14. Select the command group created earlier in the Mapping Command window.
15. Right-click and select **Insert After** → **Command** → **MapFrom...**

> **Important:** Ensure you insert this new mapping command after the execution of the command group. The supplierNum value we require is the value after the execution of the TrimRight() function.

Archived

16. Select **supplierNum** from the local variables column in the Variables window and drag it over the expression value in the Mapping Command Editor (Figure 15-40).

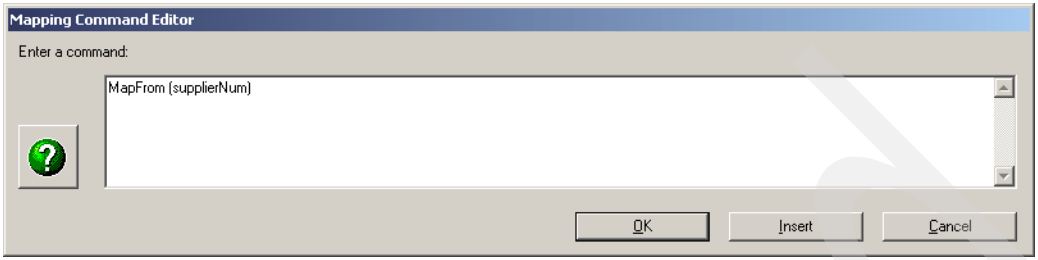


Figure 15-40 Mapping Command Editor

17. Click **OK** to complete the mapping command.

18. At this point, the Mapping Command window should appear as shown in Figure 15-41.

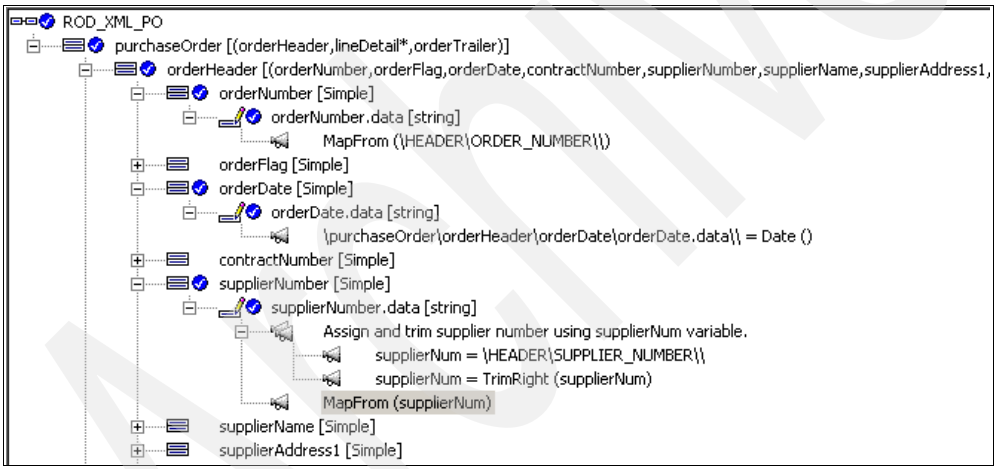


Figure 15-41 Mapping Command window

Now that we have stored the supplier number value in our supplierNum local variable, we can use that value to perform some conditional mapping. With conditional mapping, we can generate mapping commands that will only be executed after an expression has been evaluated. The expression being evaluated must result in a `true` value before the conditional commands are executed.

To create our conditional mapping commands, follow these steps:

1. Select the supplierName element from the Mapping Commands window.
2. Right-click supplierName and select **Insert Within** → **Command** → **If...**
3. We are using the **StrComp** function to test the value of our supplierNum local variable against two predefined strings; `companya` and `companyx`
4. Select the expression value in the Mapping Command Editor, right-click and chose **Functions** → **StrComp**. The Mapping command editor should appear as shown in Figure 15-42.

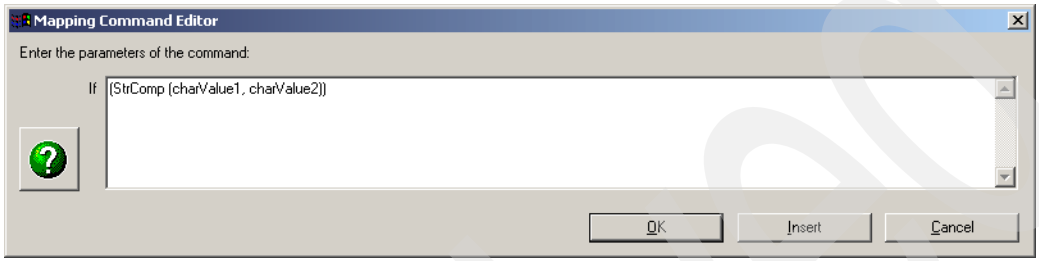


Figure 15-42 Mapping Command Editor with expression value

5. Select **supplierNum** from the local variables column of the Variables window and drag it over the charValue1 string in the Mapping Command Editor.
6. Manually overwrite the charValue2 string with `companya`. These are the two string values that will be compared by the StrComp() function. If these values are equal, the StrComp() function will return a value of 0. Thus to test our condition, we need to include a test against the return value of the StrComp() function.
7. Position the cursor after the StrComp() function but before the final right parenthesis in the Mapping Command Editor. Right-click and select **Operators EQ**. This inserts the equals operator into our mapping command. Finally enter **0** after the EQ value in the mapping command.

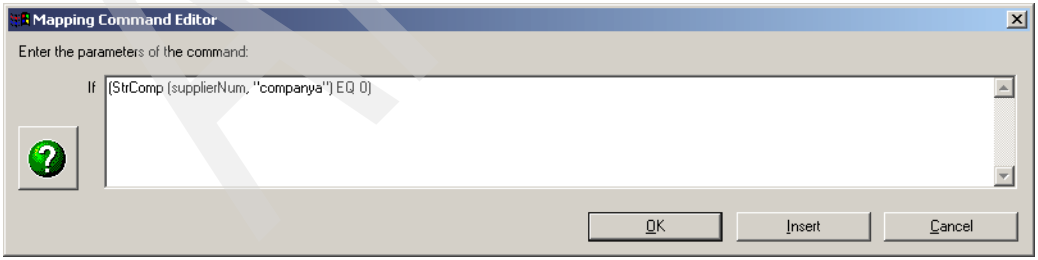


Figure 15-43 Complete mapping command

8. Click **OK** to complete the command.

9. Select the newly created If command from the Mapping Command window then right click and select **Insert Within Command MapFrom...** (Figure 15-44)**.**

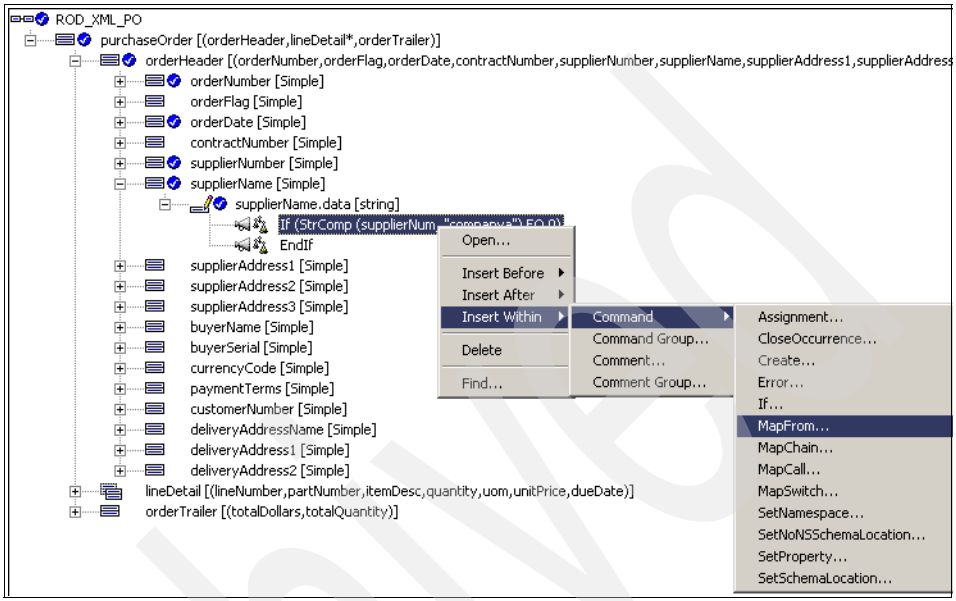


Figure 15-44 Map Command window

10. Overwrite the expression value in the Mapping Command Editor with the string `Company A` and click **OK** (Figure 15-45).

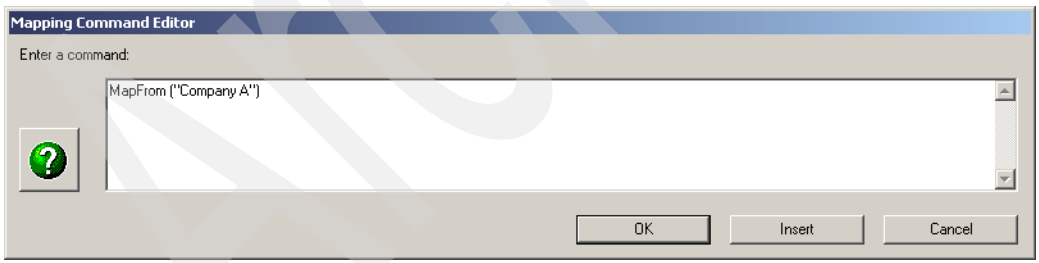


Figure 15-45 Overwrite expression

Archived

11. Select the **`If`** command from the Mapping Command window, right-click and select **Insert After Command ElseIf**... (Figure 15-46).

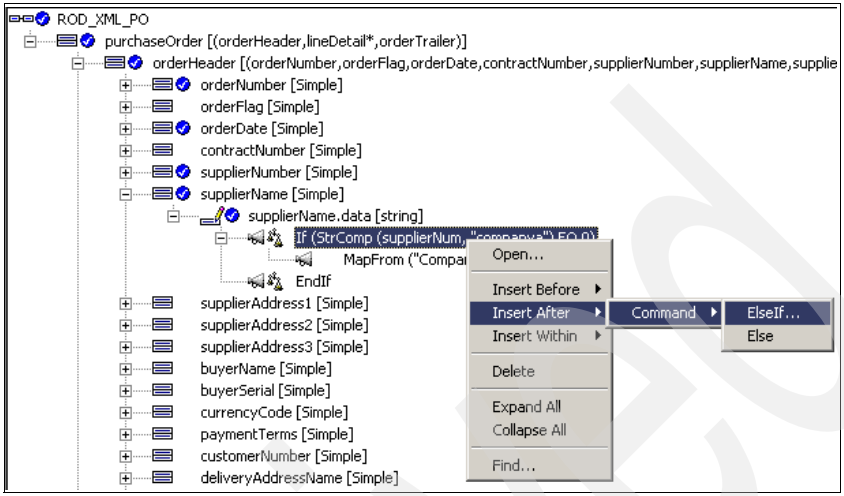


Figure 15-46 Else if command

12. Replace the expression value with another **`StrComp()`** function as shown in Figure 15-47 and click **OK**.

Figure 15-47 Edit expression

13. Finally, create a **`MapFrom()`** command within the **`ElseIf`** command just created. This MapFrom command should use the string `Company X` as shown in Figure 15-48 on page 508.

Archived

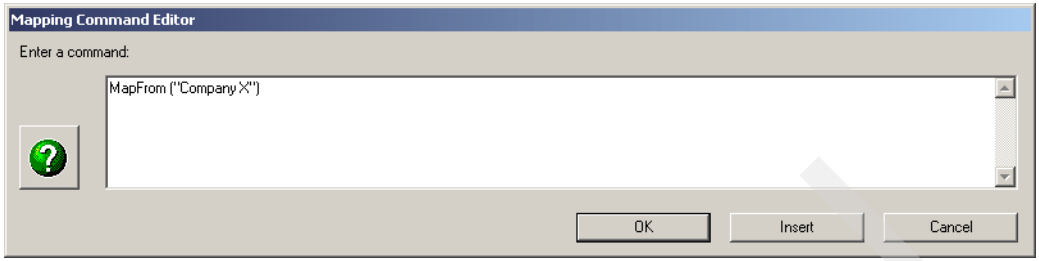


Figure 15-48 MapFrom command

14. At this point, the mapping commands for the orderHeader element should appear as shown in Figure 15-49.

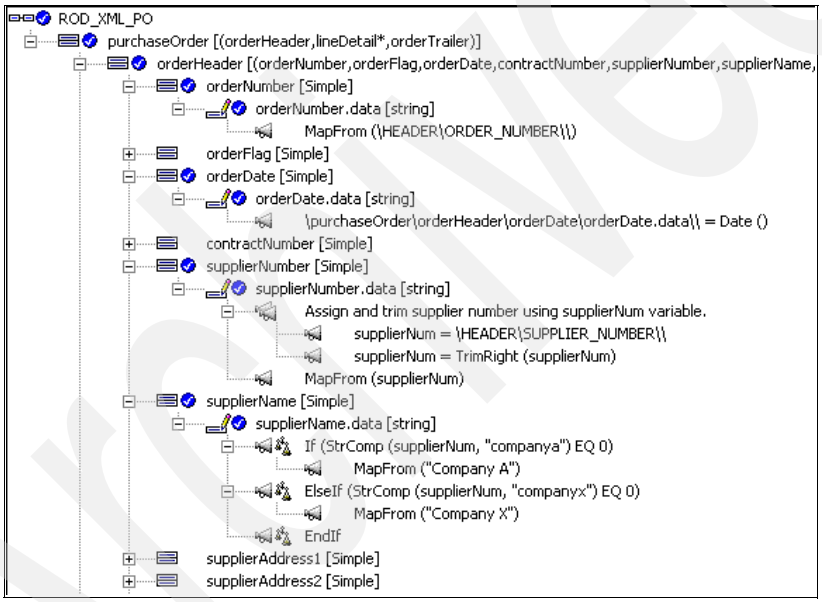


Figure 15-49 Mapping Command window for orderHeader

15. Our final step in mapping the orderHeader element is to map the customerNumber element. This is done using simple drag and drop. Simply drag the CUSTOMER_NUMBER field in the source document definition and drop it over the customerNumber field in the target document definition.

Now that we have completed the mapping commands for the orderHeader element, our next step is to populate the lineDetail element. Because this element is defined in our XML schema as having the ability to repeat, we want to create a new lineDetail element for each occurrence of the BODY record in our

Archived

source document definition. Recall from defining the ROD document definition in the Data Interchange Services client that the BODY record is also defined as having the ability to repeat.

The Data Interchange Services client provides a ForEach() function to allow us achieve this repetitive mapping. The transformation engine will repeat the same block of commands for each occurrence of a repeating element in the source document. To add this command, follow these steps:

1. Right-click the **lineDetail** element of the target document definition from the Mapping Command window. Select **ForEach...**
2. The Mapping Command Editor opens. Drag the source record, BODY, from the source document definition to the editor and drop it over the sourcePath value. Click **OK** to return to the Mapping Command window (Figure 15-50).

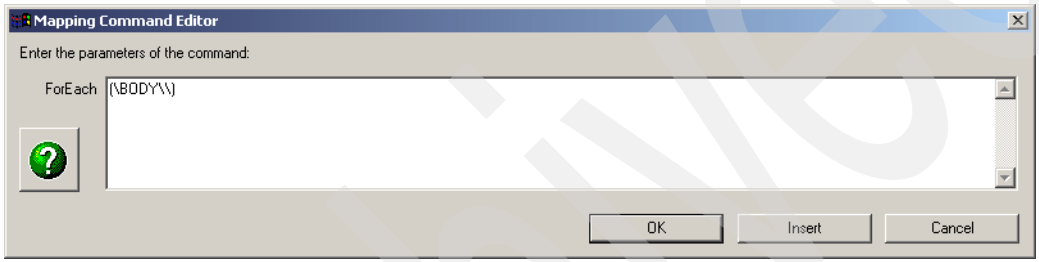


Figure 15-50 Mapping for each

3. We can now proceed with the mapping required for the elements contained within the lineDetail element. These elements can be mapped using a simple drag and drop action using the corresponding source elements. When complete, your Mapping Command window should appear as shown in Figure 15-51 on page 510.

Archived

Figure 15-51 Mapping wondiw for OrderTrailer

4. Finally, map the orderTrailer element as shown in Figure 15-52.

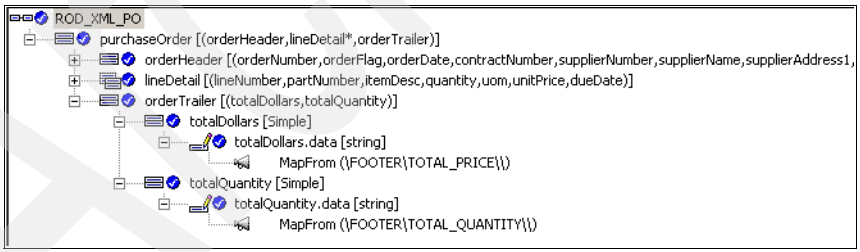


Figure 15-52 Order Trailer

5. The map is now complete and can be saved and compiled.
6. Select **Actions Compile** from the Data Interchange Services client. Click **OK** when asked do you want to save this changed object. The Execution Status window opens and details of the map compilation are displayed. The final entry in the Execution Status window should read `Compile completed successfully` (Figure 15-53 on page 511).

Archived

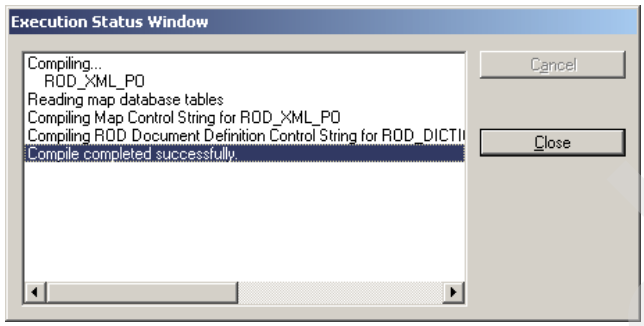


Figure 15-53 Compilation status

7. Click the **Close** button to return to the Data Transformation Map Editor.

15.5 Create a connection to the runtime database

Now that we have created and compiled our Data Transformation map in our development environment, we are in a position to export the map to our WebSphere Partner Gateway runtime database. In order to do so, we must first create a new database connection from the Data Interchange Services client. Before creating this connection however, we must first define an ODBC connection on the Microsoft Windows 2000 machine on which the Data Interchange Services client is running.

From the Data Interchange Services client menu bar select **View** → **Databases**. This displays a list window with details of all databases to which the Data Interchange Services client can connect.

1. From the menu bar select **File** → **New**. The Database window opens.
2. Enter the following details:
   - Give the database a name, for example `WPGV6DB`.
   - Select the Data Source Name.
   - Enter the Database Qualifier.
   - Set the Database Type as **Document Manager Database.**
3. Click **OK**. See Figure 15-54 on page 512.

Archived

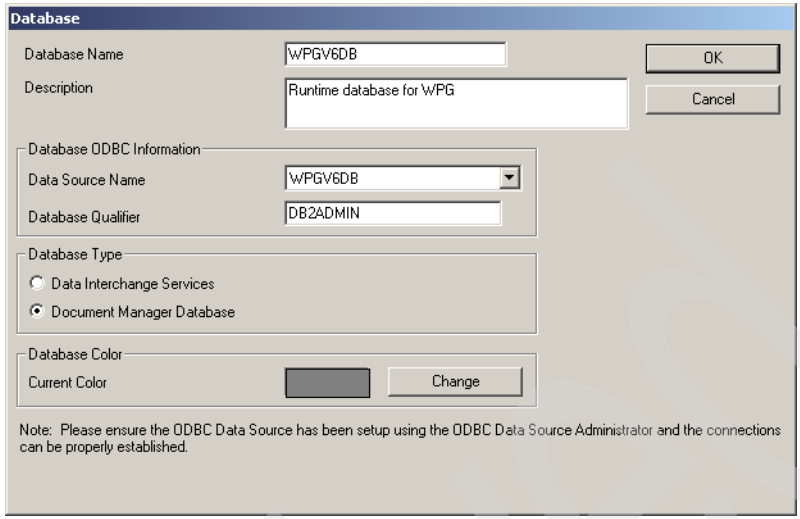


Figure 15-54 New database window

4. Open the Mapping functional area by clicking the **Mapping** icon from the Data Interchange Services client.
5. Select the **Control Strings** tab, then select the entry created for the **ROD_XML_PO** data transformation map created earlier.
6. Select **Actions** → **Export To Other Database...** from the Data Interchange Services client menu. The Select a Database window opens (Figure 15-55.

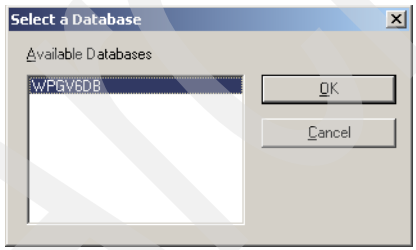


Figure 15-55 Select the database

7. Select **WPGV6DB** and click **OK**. The Connect to DB2 Database window opens (Figure 15-56 on page 513).

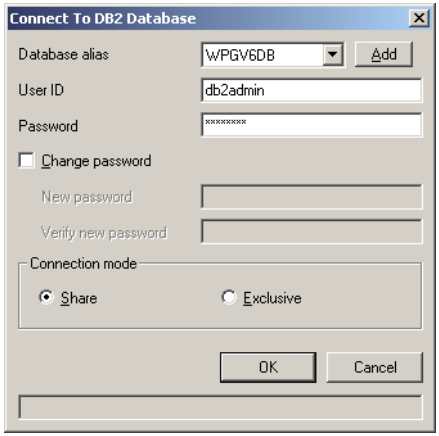


Figure 15-56 Enter user and password

8. Enter a valid username and password to connect to the WebSphere Partner Gateway runtime database. Click **OK**. The Export with Control String window opens.
9. Because our map employs an XML namespace that was created in the Data Interchange Services client when originally importing the XML schema, select **XML Namespace Objects**. Click **OK** (Figure 15-57).

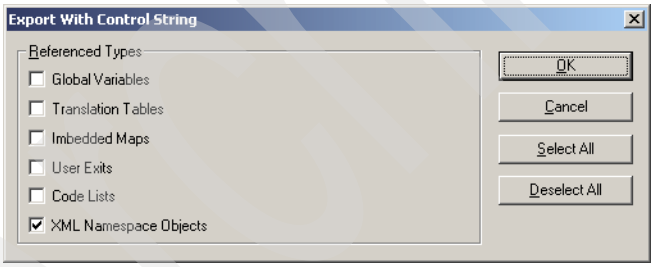


Figure 15-57 Enable namespaces

Archived

10. The Execution Status Window (Figure 15-58) opens and displays information about the export of our control string. The last line of this window should read `Export completed successfully`. Click **Close**.

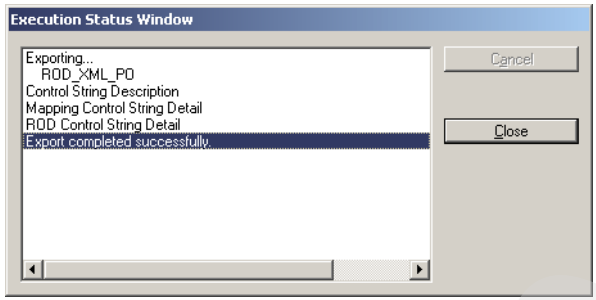


Figure 15-58 Export status

15.6 Configure WebSphere Partner Gateway at Company E

Now that the control string has been exported to the runtime database of WebSphere Partner Gateway in the previous section, we can take a look at the console of WebSphere Partner Gateway to verify if the transformation map has been uploaded successfully.

15.6.1 Verify that the map has been uploaded

Follow the steps:

1. While logged in as hubadmin to the WebSphere Partner Gateway server on Company E, go to **Hub Admin** → **Hub Configuration** → **Maps** → **Transformation Maps.**
2. Find the map ROD_XML_PO listed, as shown in Figure 15-59 on page 515.

Archived

Figure 15-59 Uploaded transformation map

3. Click the **magnifying glass** next to the uploaded map to see the document flow definition, as shown in Figure 15-60 on page 516.

Archived

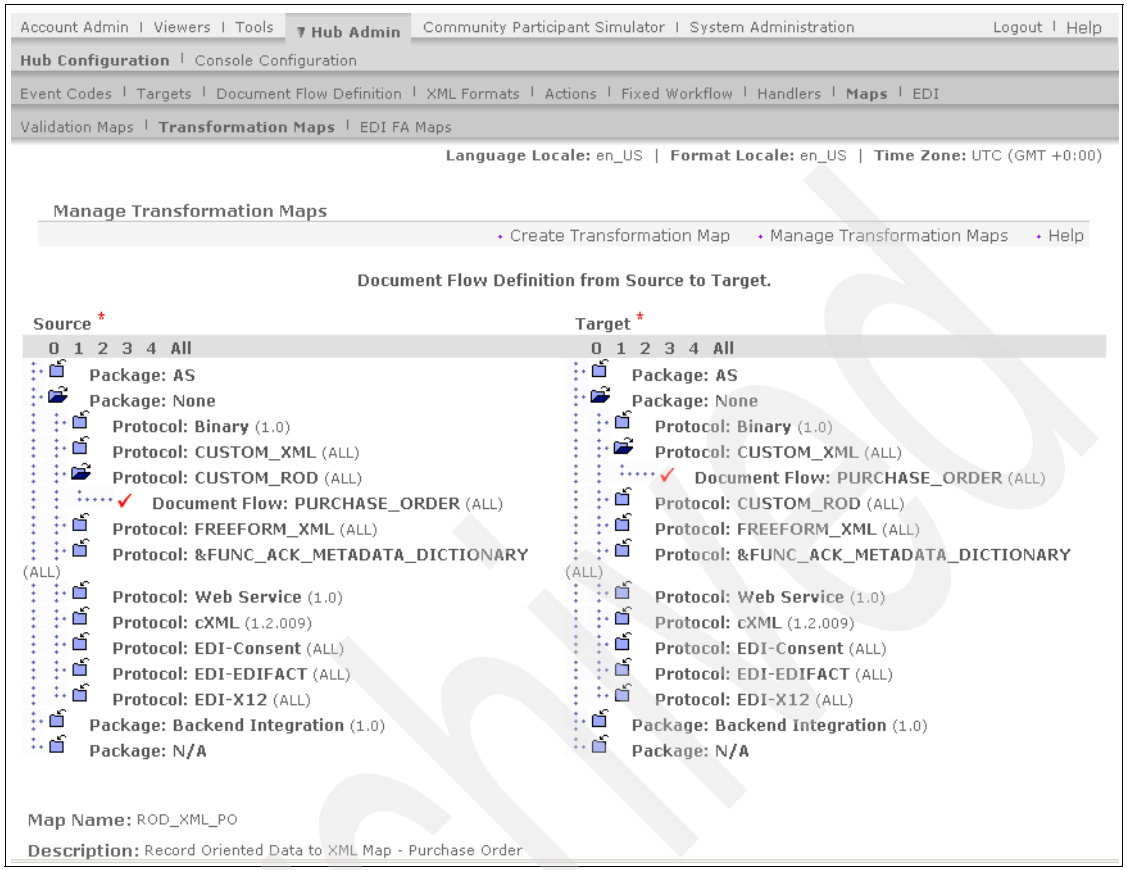


Figure 15-60 document flow definition

15.6.2 Create a target for ROD document

We cannot use the existing File Directory target on Company E. We would need to do some preprocessing on the ROD document at Company E before it is ready to be transformed and sent to Company X.

In this section, we configure a new File Directory target:

1. While still logged in as hubadmin, go to **Hub Admin** → **Hub Configuration** → **Targets.**
2. Click **Create**.
3. In the Target Details window enter the details as shown in Figure 15-61 on page 517.

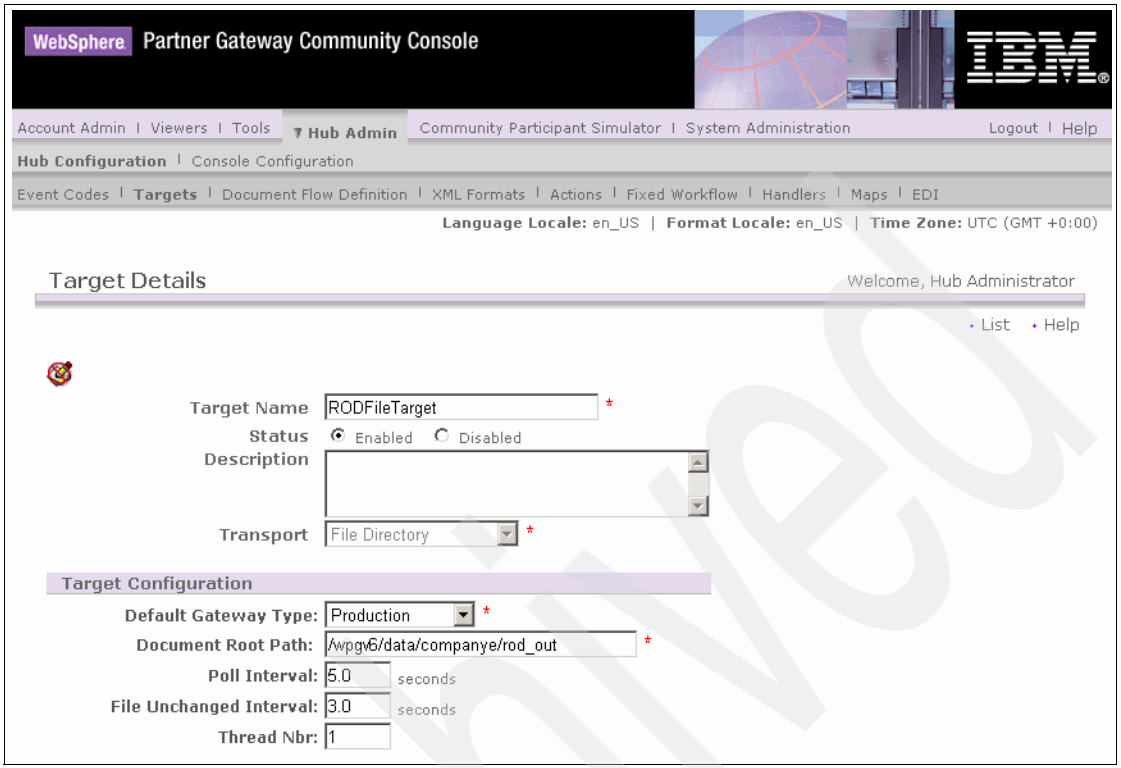


Figure 15-61 Creating a file directory target for ROD (1 of 2)

4. Further in the Target Details window, select the Configuration Point Handler as **preprocess**.
5. This brings up a two windows, from the available list of handlers on the left window select **com.ibm.bcg.edi.receiver.preprocesshandler.RODSplitterHandler** and click **Add**. This adds the handler to the right hand window, as shown in Figure 15-62 on page 518.

Archived

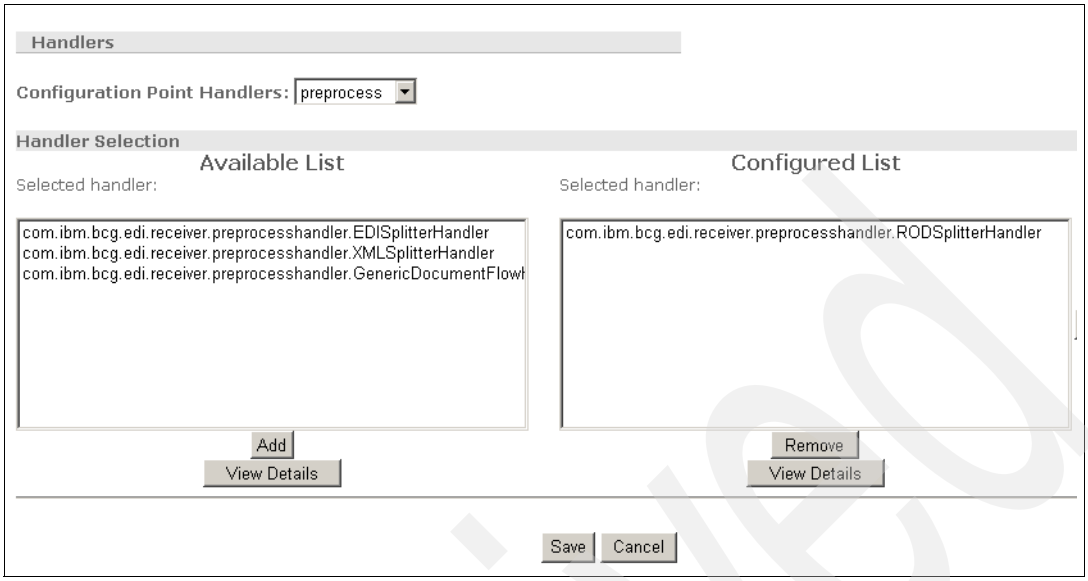


Figure 15-62 Creating a file directory target for ROD (2 of 2)

The RODSplitterHandler is added so that the ROD document is subject to some preprocessing such as appending the information about its METADICTIONARY and METADOCUMENT on to the ROD document received on this target, this information is essential for the further processing of the ROD document.RODSplitterHandler also splits multiple ROD messages present in one single document into separate ROD documents before sending them to the transformation engine or the process engine.

6. Click **Configure** next to the preprocess handler window to fill in the attributes required to configure the RODSplitterHandler. These values give the DICTIONARY and DOCUMENT information of the input ROD document to the handler, as shown in Figure 15-63 on page 519.

Archived

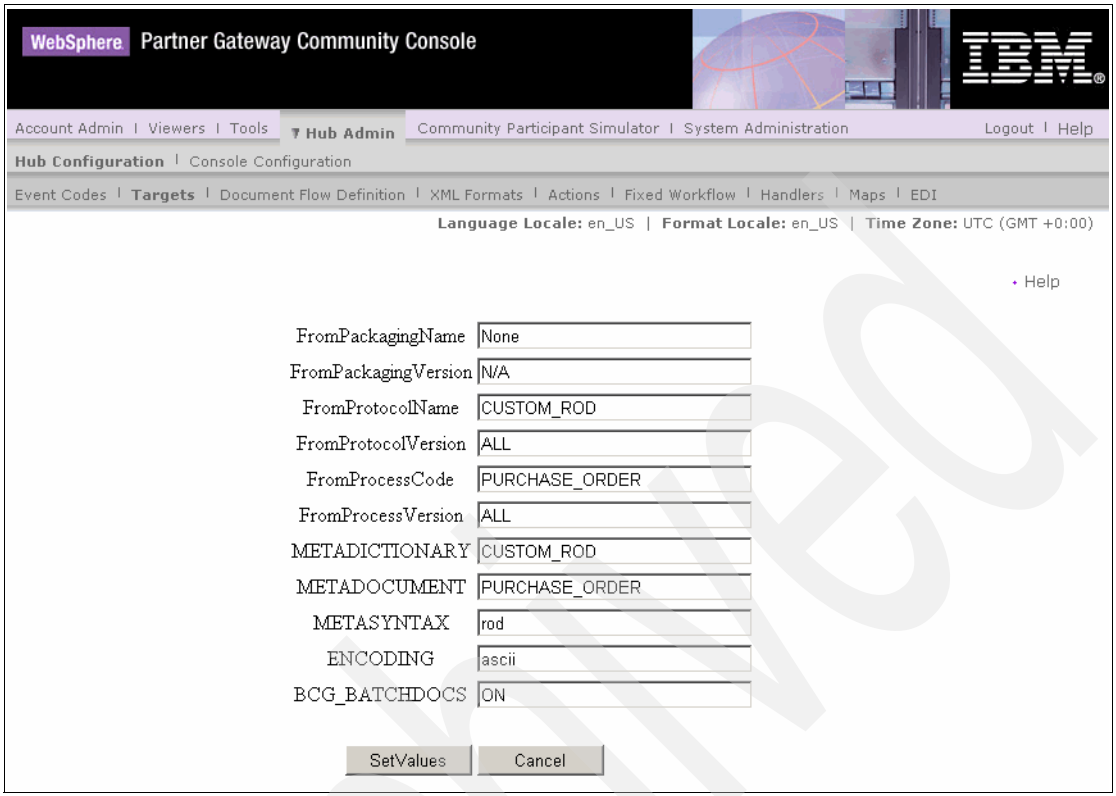


Figure 15-63 Configuring the RODSplitterHandler

7. Click **SetValues**.
8. Click **Save** in the Target Details page.
9. Click **List** to display the list of targets on Company E, which is as shown in Figure 15-64 on page 520.

Archived

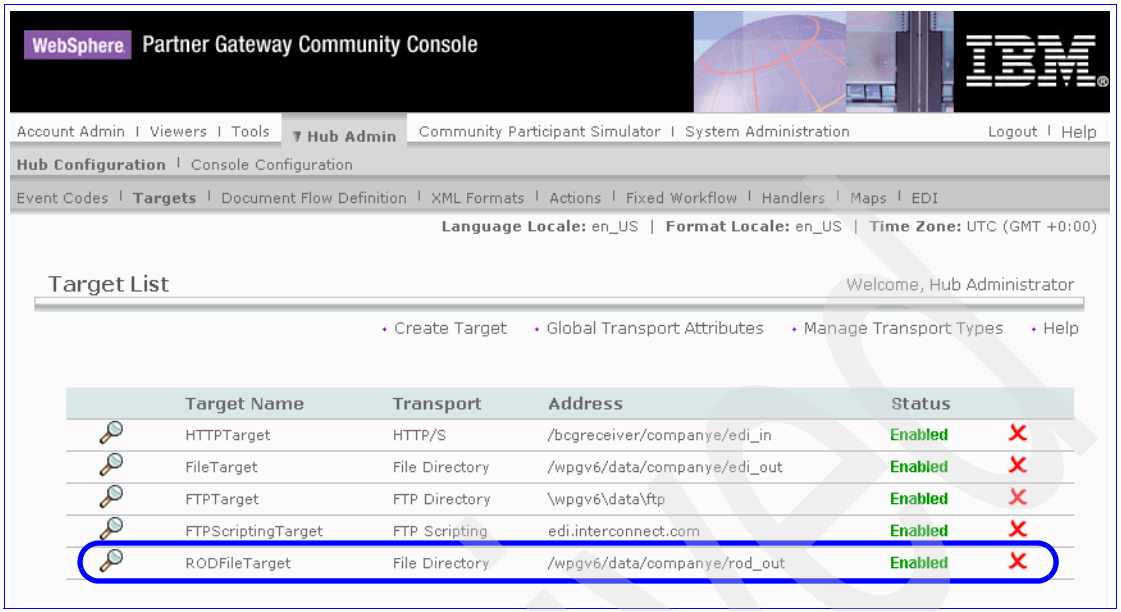


Figure 15-64 List of Targets

15.6.3 Update the configuration for Company E

To enable Company E to have the ROD document transformed to XML using the transformation map we uploaded (described in 15.5, “Create a connection to the runtime database” on page 511) and then sent to Company X, we need to update the profiles of the Company E and Company X.

Follow the steps to update the configuration for Company E:

1. While still logged as hubadmin go to **Hub Admin** → **Hub Configuration** → **Document Flow Definition.**
2. As shown in Figure 15-65 on page 521, verify if the required document flow definition for Company E is enabled.

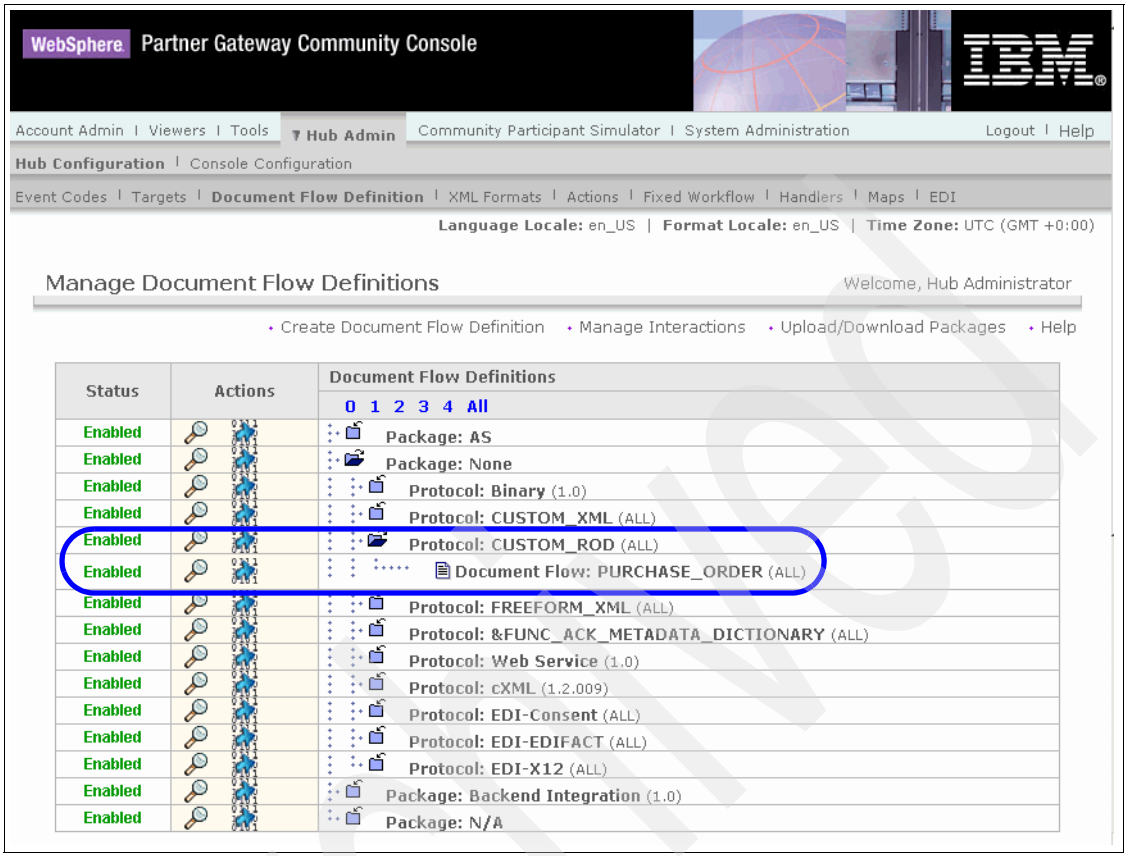


Figure 15-65 Verify that document flow definition for Company E is enabled

3. Select **Account Admin** → **Profiles** → **Community Participant.**
4. Click **Search**. From the list of participants displayed, click the **magnifying glass** next to **Company E** to open Company E's profile. Click **B2B capabilities**, opening the window shown in Figure 15-66 on page 522.
5. **Enable** None (Package) / CUSTOM_ROD(Protocol) / PURCHASE_ORDER (Document Flow), as the B2B capability for Company E.

Archived

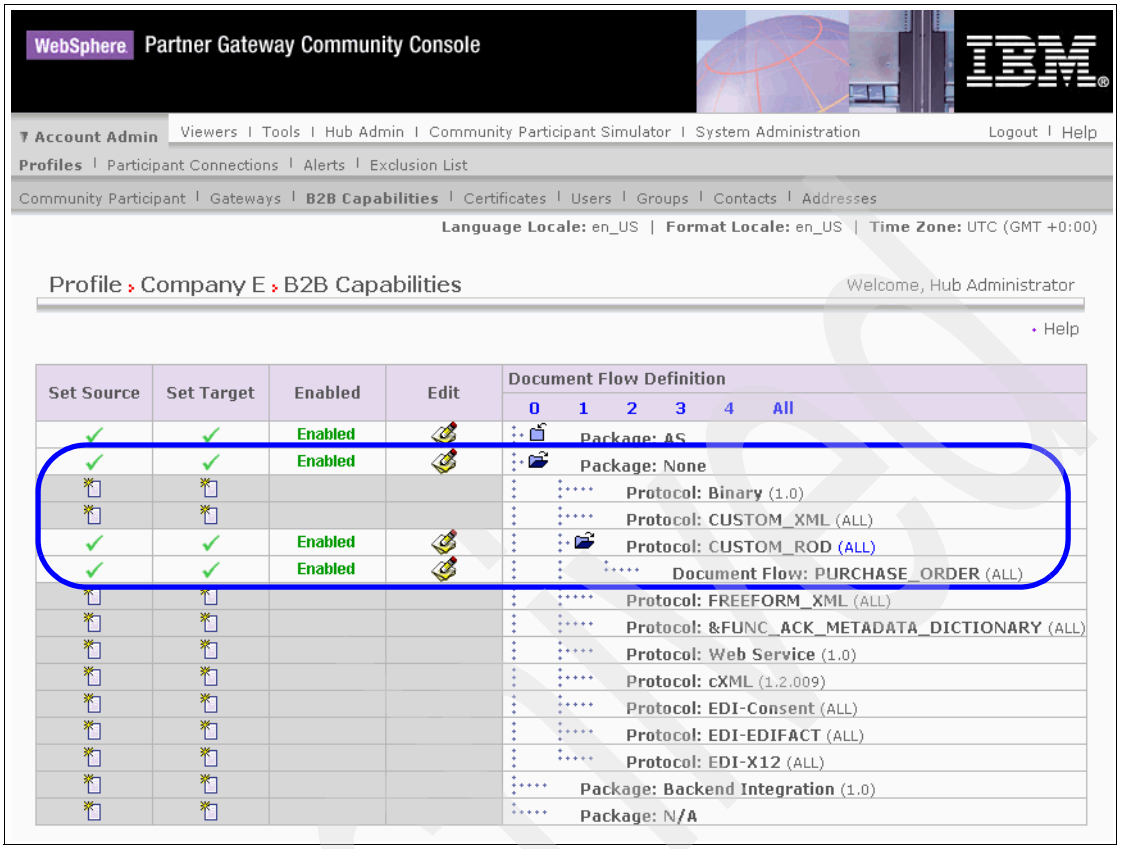


Figure 15-66 Company E's B2B capability

6. When the appropriate B2B capability is enabled, under the profile of Company E, click **Gateways**.

7. A list of Gateways for Company E is displayed as shown in Figure 15-67. We will use the existing FileSystemGateway for our discussion in this chapter.

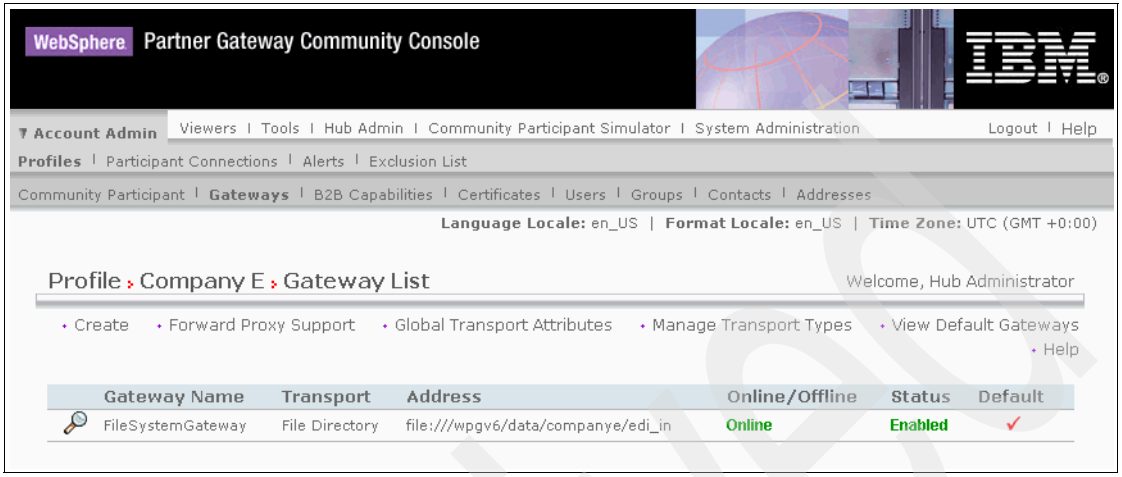


Figure 15-67 List of Gateways for Company E

15.6.4 Update the configuration for Company X

Follow the steps to update the configuration for Company X:

1. While still logged as hubadmin, select **Hub Admin** → **Hub Configuration** → **Document Flow Definition.**
2. Verify if the required document flow definition for Company X is enabled, as in Figure 15-68 on page 524.

Archived

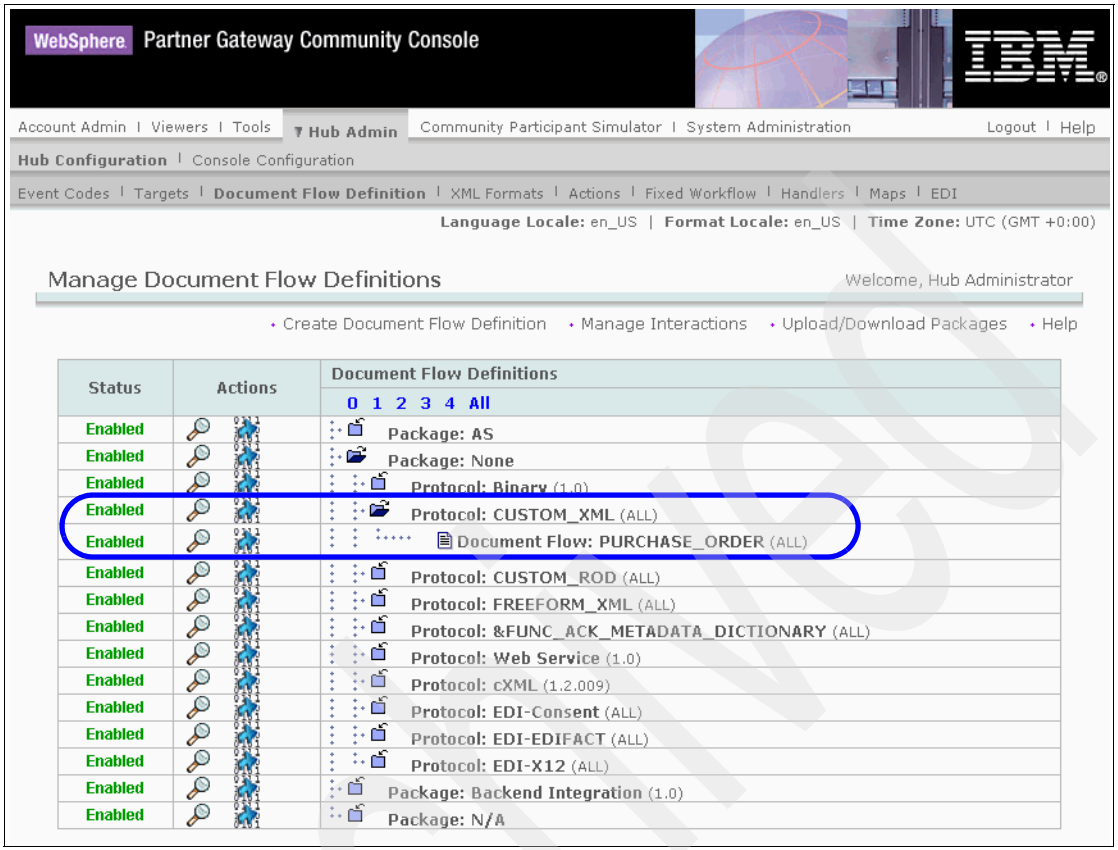


Figure 15-68 Verifying if the document flow definition for Company X is enabled

3. Select **Account Admin** → **Profiles** → **Community Participant**.
4. Click **Search**. From the list of participants displayed, click the **magnifying glass** next to **Company X,** opening Company X's profile. Click **B2B capabilities**, opening a window as shown in Figure 15-69 on page 525.
5. **Enable** None (Package) / CUSTOM_XML (Protocol) / PURCHASE_ORDER (Document Flow), as the B2B capability for Company X.

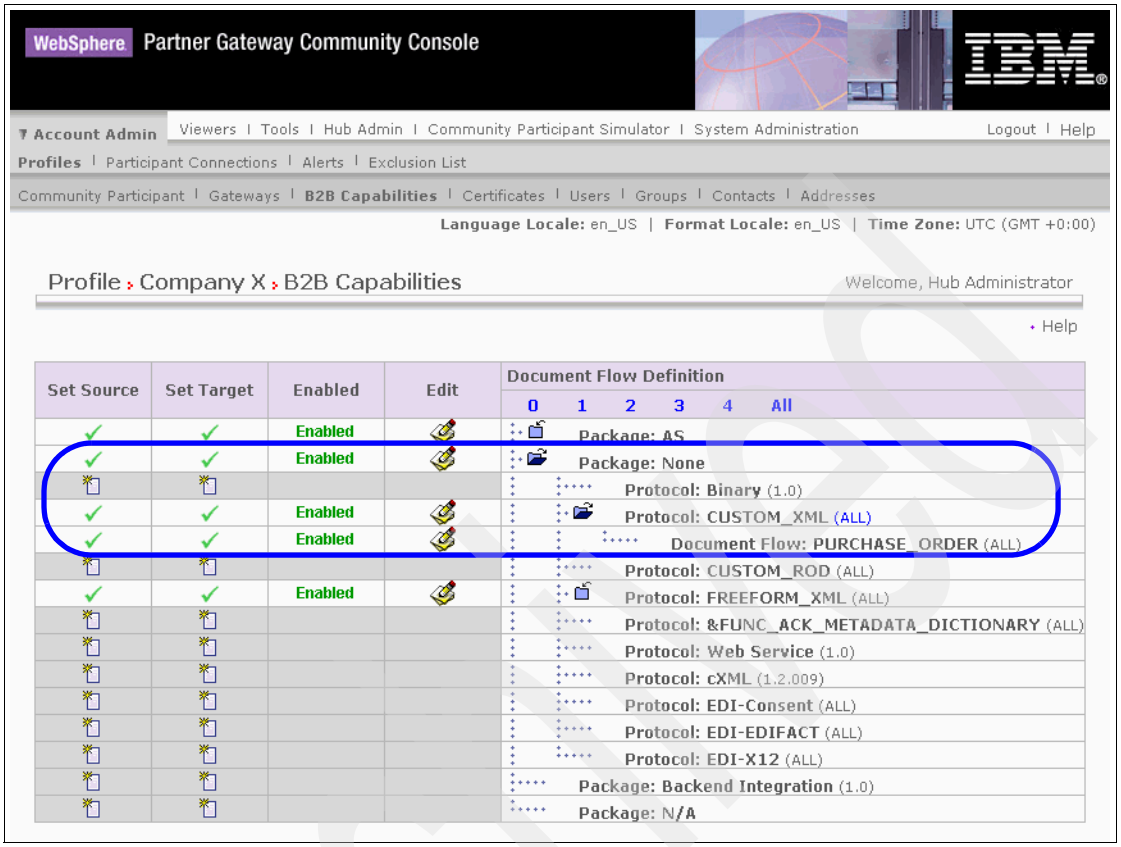


Figure 15-69 Company X's B2B capability

6. Click **Gateways**. This brings up a list of gateways for Company X, as shown in Figure 15-70 on page 526.

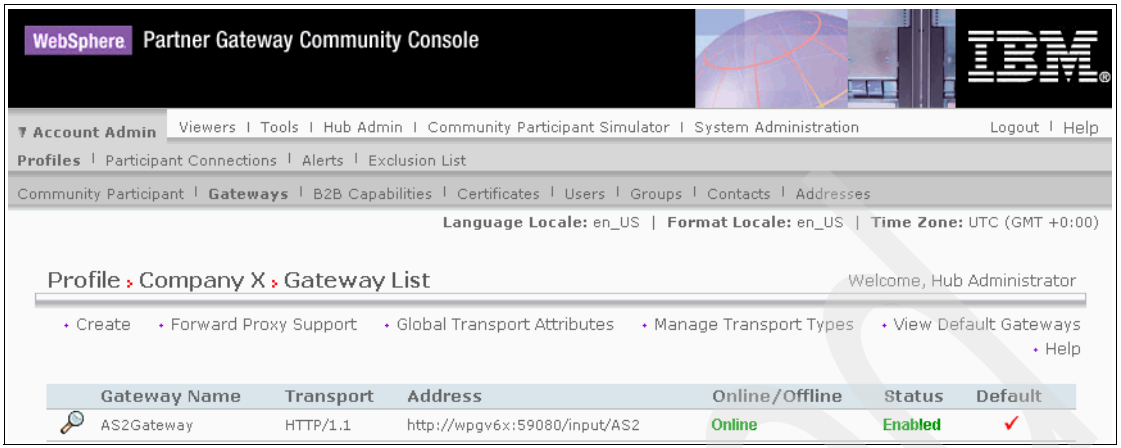


Figure 15-70 List of Gateways for Company X

7. In this step we will create a new Gateway to post the transformed XML document from Company E to Company X. To create a new Gateway under the profile of Company X, follow these steps:

   a. Click **Create**.

   b. The gateway details window opens, as shown in Figure 15-71 on page 527. Type `HttpGateway` for the Gateway Name.

   c. Select **HTTP/1.1** as the Transport.

   d. Leave the default choice for the Forward Proxy List.

   e. Type the following for Address, `http://wpgv6x:59080/input/HTTP`.

   f. Enter the User Name and password as it was configured in WebSphere Partner Gateway Express. Refer to 10.3.2, “Customizing the profile of Participant Company E” on page 281.

   g. Leave the default values in the other fields and click **Save**.

Archived

Profile › Company X › Gateway List › HttpGateway Welcome, Hub Administrator

• List • Help

Gateway Name: HttpGateway *

Status: Enabled Disabled

Online/Offline: Online Offline

Description:

Gateway Configuration

Transport: HTTP/1.1

Forward Proxy List: Use default forward proxy

Address: http://wpgv6x:59080/input/HTTP *

User Name: companye

Password: ********

Retry Count: 3

Retry Interval: 300 seconds

Number of Threads: 3

Validate Client IP: No Yes

Auto Queue: No Yes

Connection Timeout: 120 seconds

Handlers

Configuration Point Handlers: Select One

Save Cancel

Figure 15-71 Gateway Details

h. Click **List** to see the list of Gateways for Company X. It should be as shown in Figure 15-72 on page 528.

Archived

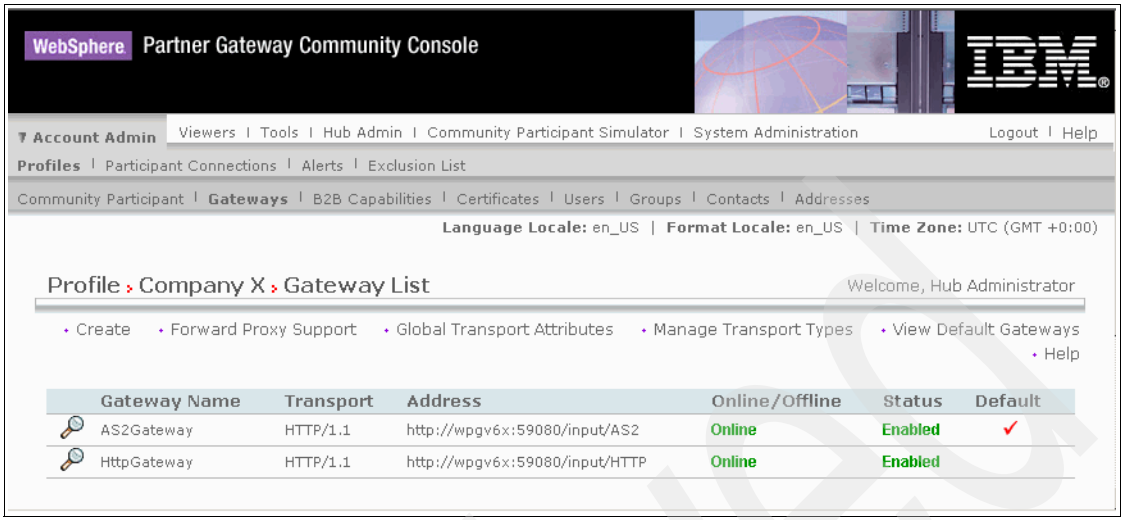


Figure 15-72 List of gateways for Company X

15.6.5 Create an interaction

Now that we have updated the profiles for Companies E and X, we need to create a new interaction between the appropriate source and target types, and select the appropriate transformation map and appropriate action.

Creating a new Action

To verify if the required Action for our example, an ROD translate, is present in the list of default Actions, follow these steps:

1. While still logged in as hubadmin, select **Hub Admin** → **Hub Configuration** → **Actions**.
2. This window displays the list of default Actions available, shown in Figure 15-73 on page 529.

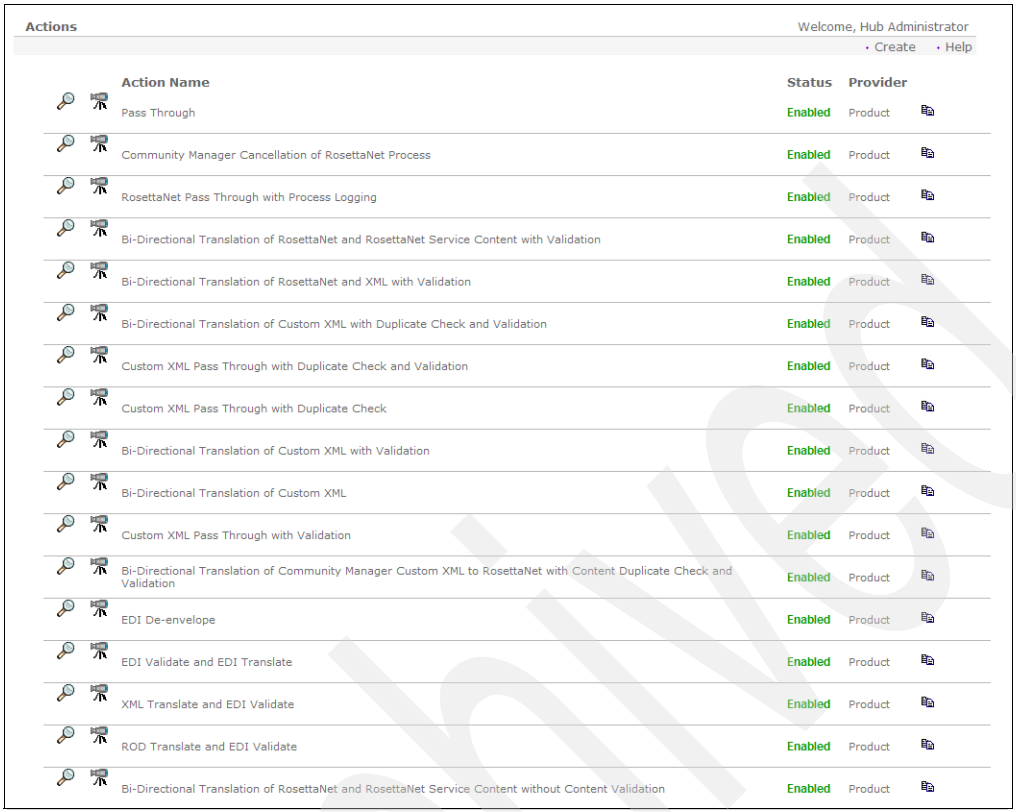


Figure 15-73 List of default actions

3. Because the action we want is not present in the list of defaults, click **Create**.
4. In the Create Actions window shown in Figure 15-74 on page 530, type the name of the Action as `ROD Translate`, for example.
5. Check the **Enabled** box.
6. Select the handler, **com.ibm.bcg.edi.business.process.RODTranslatorFactory** from the list of available handlers, on the left window and click **Add**, to add it to the Configured List window on the right hand side, as shown in Figure 15-74 on page 530.

Archived

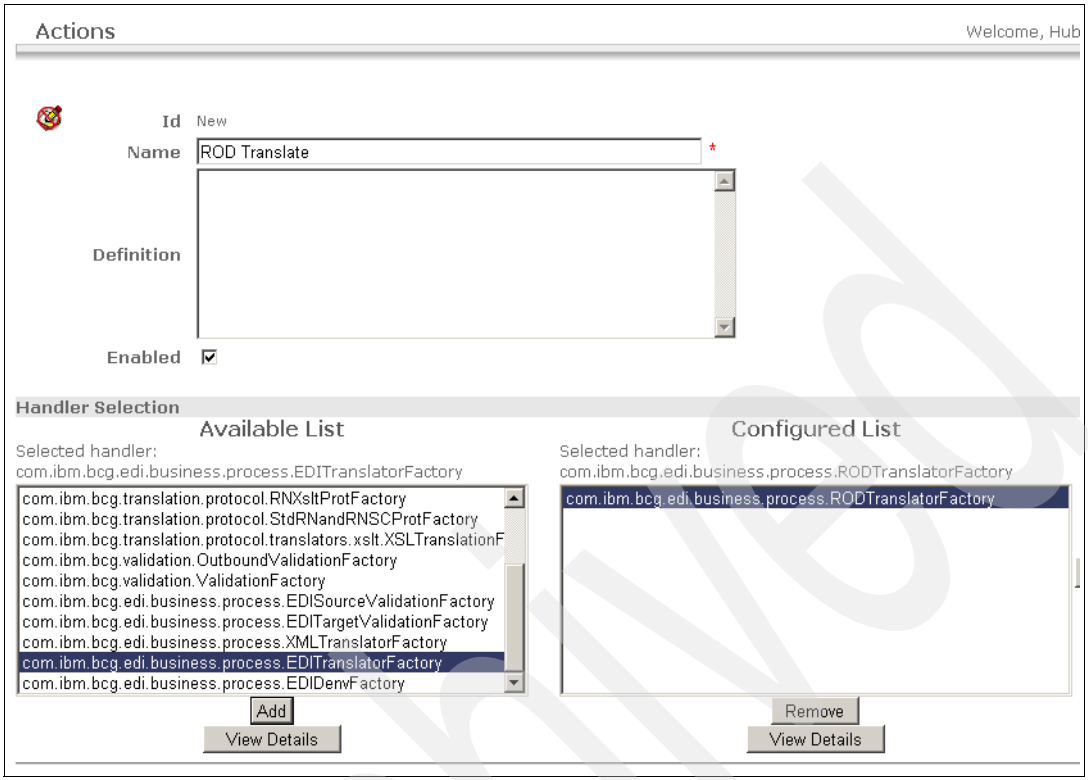


Figure 15-74 Action details window

7. Click **Save**.
8. Click **List** to view the new Action listed, as in Figure 15-75 on page 531.

Archived

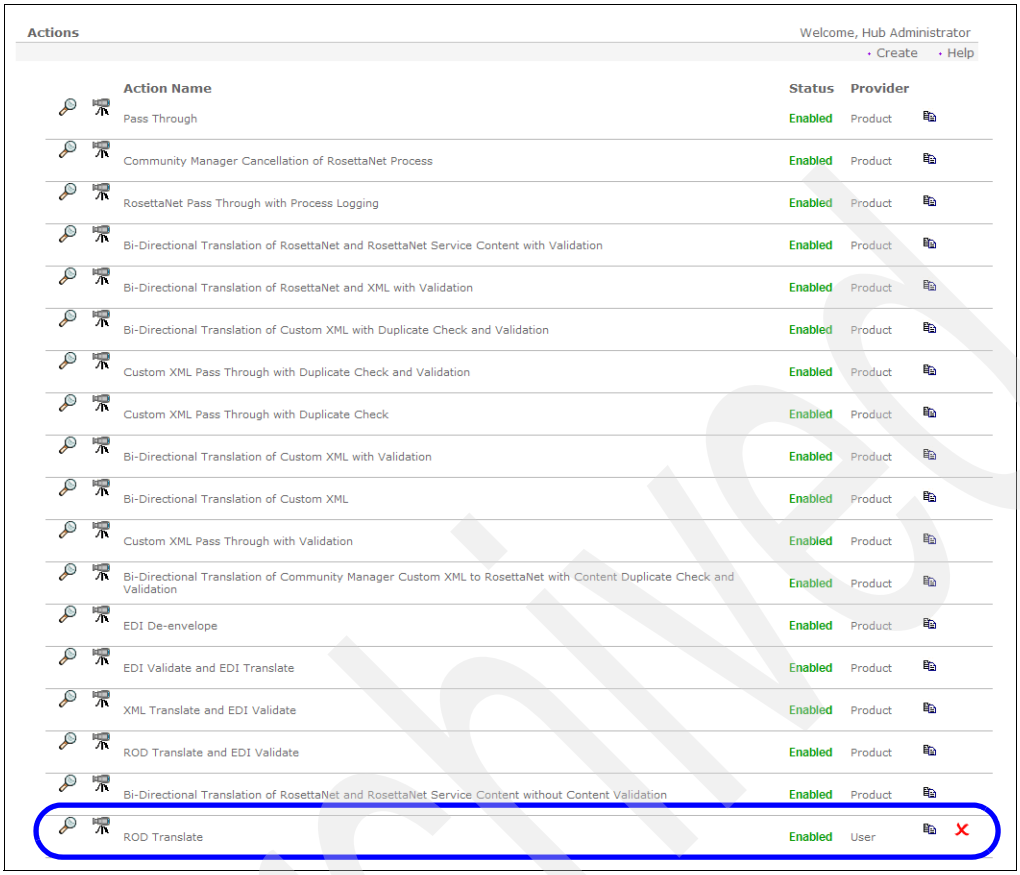


Figure 15-75 List of actions

Create an interaction

Follow these steps to create the interaction:

1. While still logged in as hubadmin, select **Hub Admin** → **Hub Configuration** → **Document Flow Definition.**
2. Click **Manage Interactions**.
3. In the Manage Interactions window, click **Create Interaction**.
4. Create an interaction between source as None/CUSTOM_ROD/PURCHASE_ORDER and target as None/CUSTOM_XML/PURCHASE_ORDER, as shown in Figure 15-76 on page 532.
5. Select the map as **ROD_XML_PO** and select the Action as **ROD Translate**.

Archived

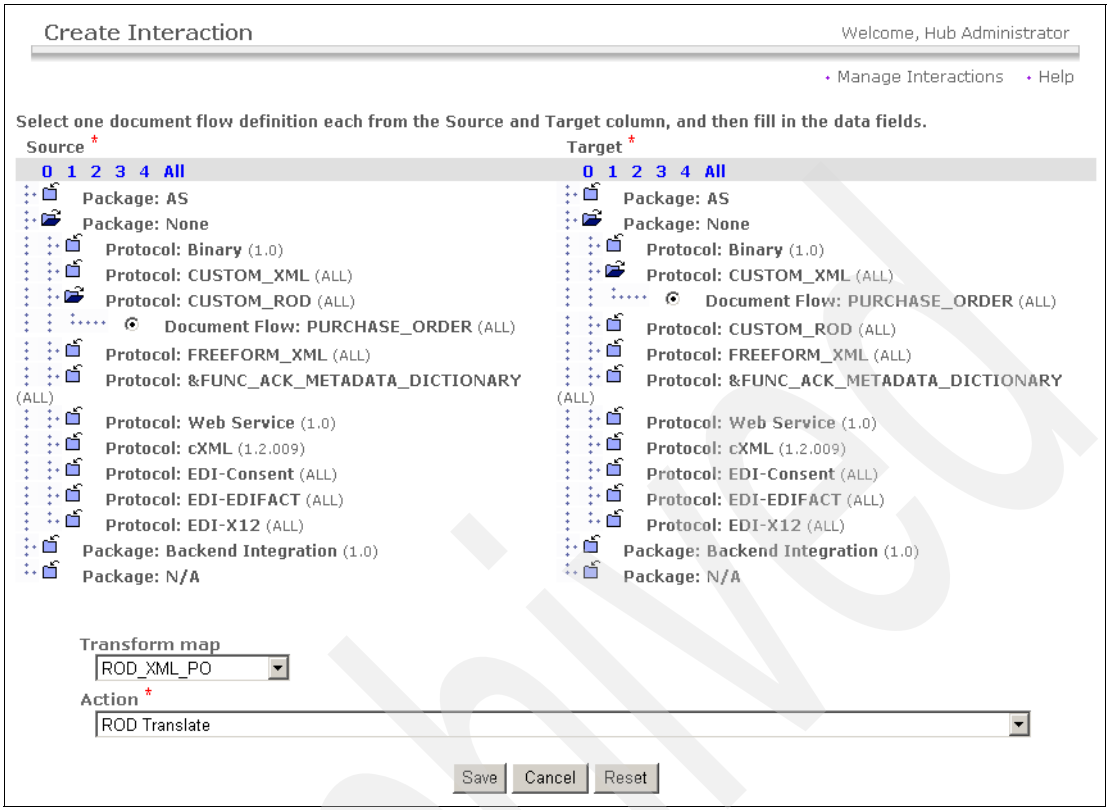


Figure 15-76 Creating an interaction

6. Click **Save**.
7. On the Manage Interactions window, click **Search**. A list of available interactions is returned, including the new interaction we just created. The new interaction is highlighted in Figure 15-77 on page 533.

Archived

Enabled	Edit	Source		Target
✓	🔍	**Package: AS** AS (N/A) **Protocol: Binary** Binary (1.0) **Document Flow: Binary** Binary (1.0)	**Transformation Map:**	**Package: Backend Integration** Backend Integration (1.0) **Protocol: Binary** Binary (1.0) **Document Flow: Binary** Binary (1.0)
✓	🔍	**Package: Backend Integration** Backend Integration (1.0) **Protocol: Binary** Binary (1.0) **Document Flow: Binary** Binary (1.0)	**Transformation Map:**	**Package: AS** AS (N/A) **Protocol: Binary** Binary (1.0) **Document Flow: Binary** Binary (1.0)
✓	🔍	**Package: AS** AS (N/A) **Protocol: Binary** Binary (1.0) **Document Flow: Binary** Binary (1.0)	**Transformation Map:**	**Package: None** None (N/A) **Protocol: Binary** Binary (1.0) **Document Flow: Binary** Binary (1.0)
✓	🔍	**Package: None** None (N/A) **Protocol: Binary** Binary (1.0) **Document Flow: Binary** Binary (1.0)	**Transformation Map:**	**Package: AS** AS (N/A) **Protocol: Binary** Binary (1.0) **Document Flow: Binary** Binary (1.0)
✓	🔍	**Package: None** None (N/A) **Protocol: CUSTOM_ROD** CUSTOM_ROD (ALL) **Document Flow: PURCHASE_ORDER** PURCHASE_ORDER (ALL)	**Transformation Map:** ROD_XML_PO	**Package: None** None (N/A) **Protocol: CUSTOM_XML** CUSTOM_XML (ALL) **Document Flow: PURCHASE_ORDER** PURCHASE_ORDER (ALL)
✓	🔍	**Package: AS** AS (N/A) **Protocol: FREEFORM_XML** FREEFORM_XML (ALL) **Document Flow: FORECAST** FORECAST (ALL)	**Transformation Map:**	**Package: None** None (N/A) **Protocol: FREEFORM_XML** FREEFORM_XML (ALL) **Document Flow: FORECAST** FORECAST (ALL)
✓	🔍	**Package: None** None (N/A) **Protocol: FREEFORM_XML** FREEFORM_XML (ALL) **Document Flow: FORECAST** FORECAST (ALL)	**Transformation Map:**	**Package: AS** AS (N/A) **Protocol: FREEFORM_XML** FREEFORM_XML (ALL) **Document Flow: FORECAST** FORECAST (ALL)
✓	🔍	**Package: AS** AS (N/A) **Protocol: EDI-X12** EDI-X12 (ALL) **Document Flow: ISA** ISA (ALL)	**Transformation Map:**	**Package: None** None (N/A) **Protocol: EDI-X12** EDI-X12 (ALL) **Document Flow: ISA** ISA (ALL)
✓	🔍	**Package: None** None (N/A) **Protocol: EDI-X12** EDI-X12 (ALL) **Document Flow: ISA** ISA (ALL)	**Transformation Map:**	**Package: AS** AS (N/A) **Protocol: EDI-X12** EDI-X12 (ALL) **Document Flow: ISA** ISA (ALL)

Figure 15-77 List of interactions

15.6.6 Create participant connections

After creating the necessary interaction, we now move on to create the required participant connection to enable the transformation and transfer of the ROD document.

Follow these steps to create the participant connections:

1. While logged in as hubadmin, select **Account Admin** → **participant connections.**
2. Select the Source as **Company E** and target as **Company X,** and click **Search**.
3. This lists the available participant connections existing between Company E and Company X, as shown in Figure 15-78 on page 534.

Figure 15-78 List of participant connections between Company E and Company X

4. Click **Activate**, against the participant connection highlighted in Figure 15-78. This results in Figure 15-79 on page 535.

Archived

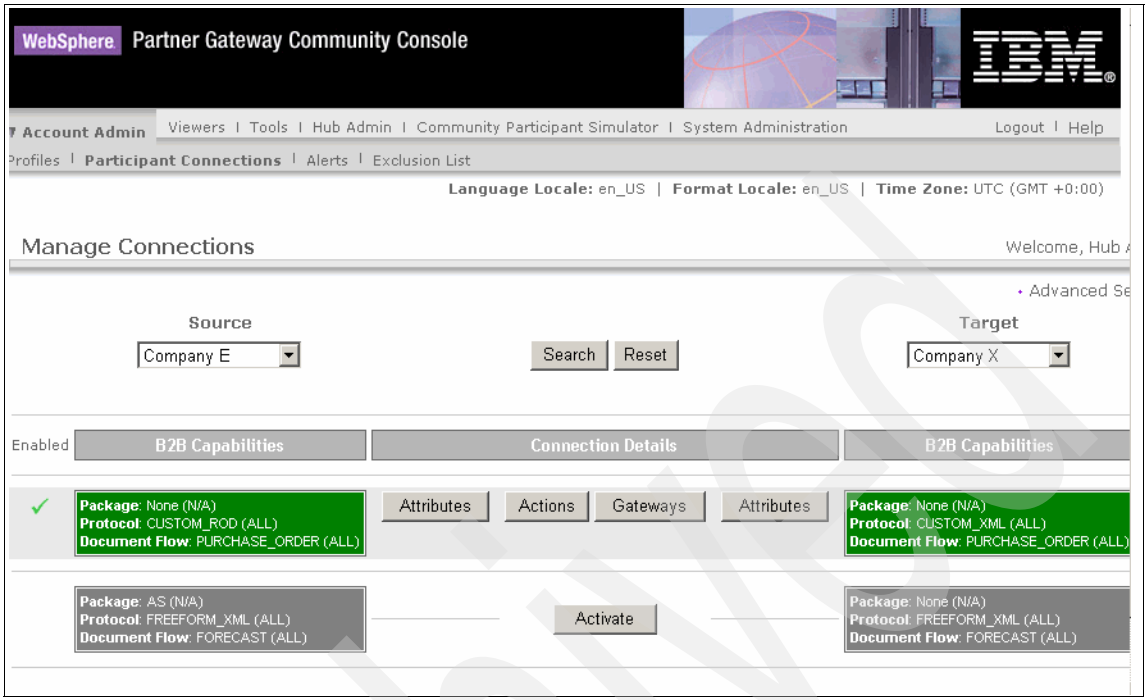


Figure 15-79 Active participant connections between Company E and Company X.

5. Click **Gateways**, which opens the window shown in Figure 15-80. Select **HttpGateway** as the target gateway. Click **Save**.

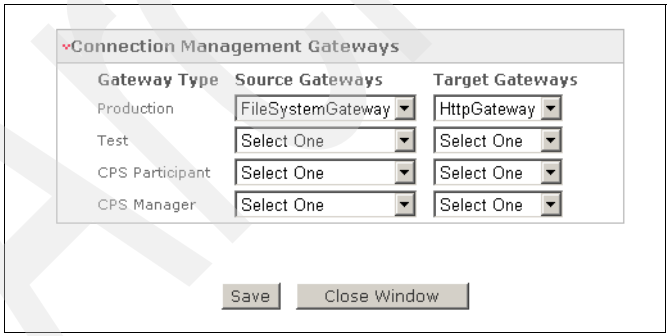


Figure 15-80 Selecting the gateway for the participant connection

Archived

15.7 Validating communication

After the configuration steps are complete, we can validate the communication. Drop the ROD file in /wpgv6/data/companye/rod_out folder. After the file is picked up from here, go to the Document Viewer to see the status of the document sent.

Figure 15-81 shows the Document Viewer results.

1. We see that the document was successfully sent to the target as there is a status of green check mark against the document.

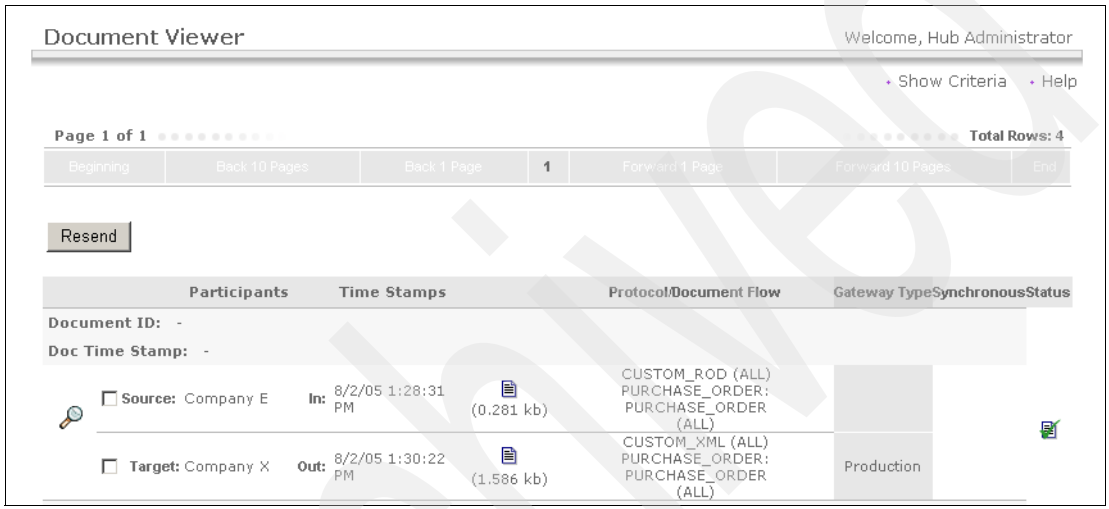


Figure 15-81 Document viewer showing successful transfer

2. Click the document icon against the source to see the source ROD document, shown in Figure 15-82 on page 537.

Archived

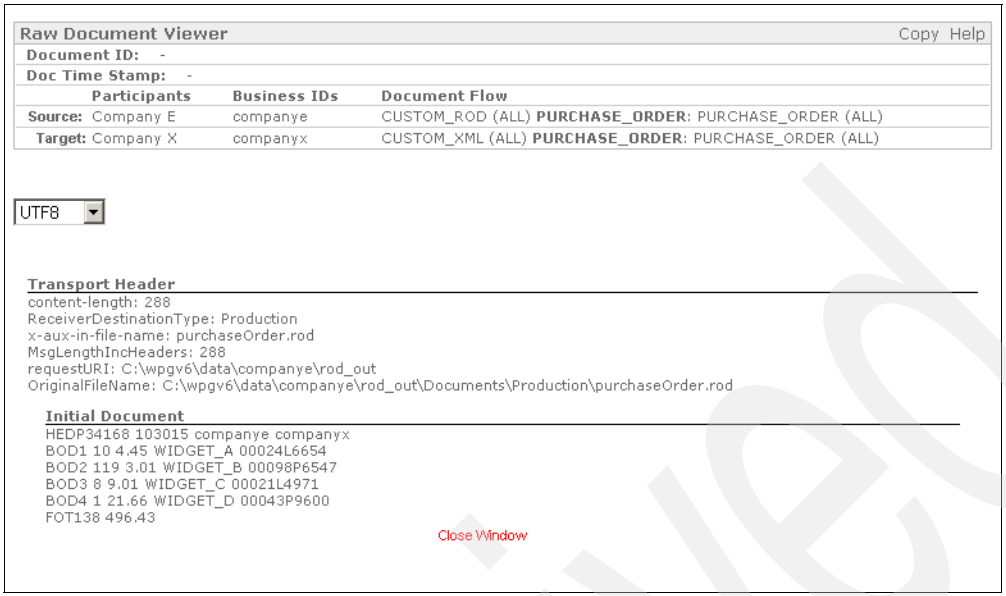


Figure 15-82 Source ROD document

3. To view the target document, click the document icon against the target. The transformed XML document is shown in Figure 15-83 on page 538.

Archived

```
Translated Document
<?xml version="1.0" encoding="UTF-8"?>
<purchaseOrder>
  <orderHeader>
    <orderNumber>103015</orderNumber>
    <orderFlag>UO</orderFlag>
    <orderDate>P34168</orderDate>
    <supplierNumber>companyx</supplierNumber>
    <supplierName>Company X</supplierName>
    <customerNumber>companye</customerNumber>
  </orderHeader>
  <lineDetail>
    <lineNumber>1</lineNumber>
    <partNumber>00024L6654</partNumber>
    <itemDesc>WIDGET_A</itemDesc>
    <quantity>10</quantity>
    <uom>EA</uom>
    <unitPrice>4.45</unitPrice>
  </lineDetail>
  <lineDetail>
    <lineNumber>2</lineNumber>
    <partNumber>00098P6547</partNumber>
    <itemDesc>WIDGET_B</itemDesc>
    <quantity>119</quantity>
    <uom>EA</uom>
    <unitPrice>3.01</unitPrice>
  </lineDetail>
  <lineDetail>
    <lineNumber>3</lineNumber>
    <partNumber>00021L4971</partNumber>
    <itemDesc>WIDGET_C</itemDesc>
    <quantity>8</quantity>
    <uom>EA</uom>
    <unitPrice>9.01</unitPrice>
  </lineDetail>
  <lineDetail>
    <lineNumber>4</lineNumber>
    <partNumber>00043P9600</partNumber>
    <itemDesc>WIDGET_D</itemDesc>
    <quantity>1</quantity>
    <uom>EA</uom>
    <unitPrice>21.66</unitPrice>
  </lineDetail>
  <orderTrailer>
    <totalQuantity>496.43</totalQuantity>
  </orderTrailer>
</purchaseOrder>
Close Window
```


Figure 15-83 Transformed output received at the target - XML document

The communication can be validated at Company X:

1. Log in as admin to WebSphere Partner Gateway Express on Company X.
2. Go to **HTTP** → **Received.**
3. Click **Search**.
4. The Received Documents window lists all the documents received and their status. Click the **magnifying glass** icon against the most recently received document (assuming the document we sent was the latest one to arrive at Company X). The Received Documents Detail window as shown in Figure 15-84 on page 539, shows that the document sent by Company E was successfully received.

Archived

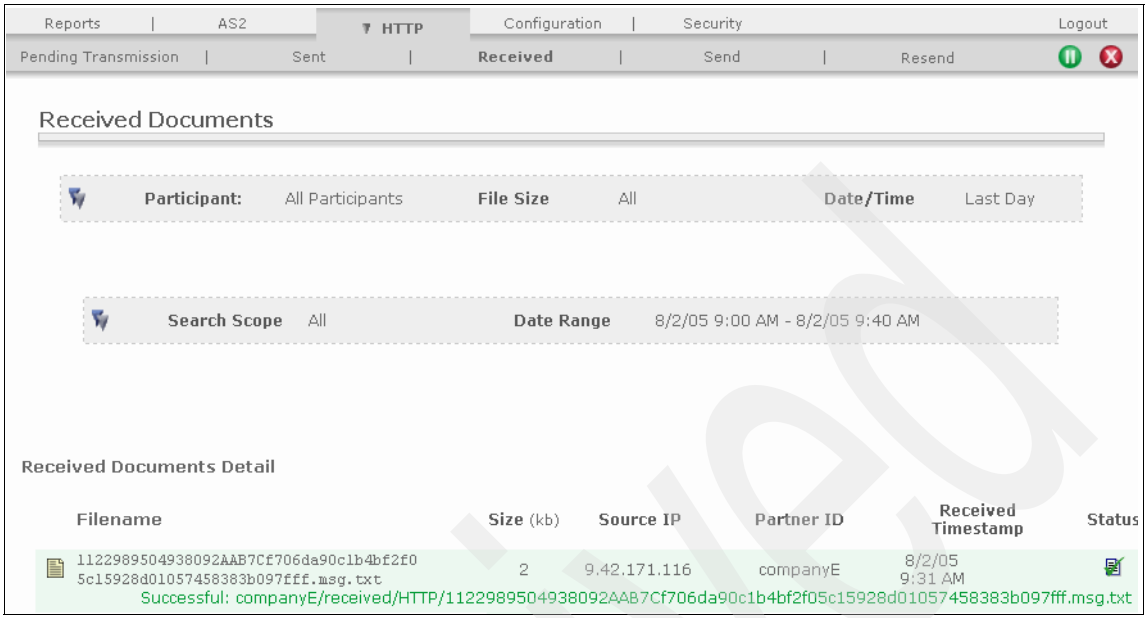


Figure 15-84 Received Document details on Company X

5. Click the document icon next to the received document, which opens the window shown in Figure 15-85. This shows that the XML document which is the same as seen in Figure 15-83 on page 538, is received at Company X.

Figure 15-85 XML document received by Company X

Archived

Archived

16

Advanced mapping

In this chapter, we extend our work with the Data Interchange Services client by looking into some of the available advanced mapping functions.

Archived

© Copyright IBM Corp. 2005. All rights reserved.

16.1 MapSwitch()

The MapSwitch command instructs the Data Interchange Services to process the inbound message with a data transformation map other than the current map if the value contained in a certain field, for example, is a certain value.

In our scenario, we will switch to a map called ROD_ROD_FLAT if the supplier number contained in the source message is `companyx`.

We start by creating a new Data Transformation map:

1. Open the Mapping functional area from the Data Interchange Services client and select **File** → **New** from the menu bar. The Create a Data Transformation Map window opens.
2. Provide a Map Name, for example `ROD_ROD_SWITCH` and a Description. Click **Next**.

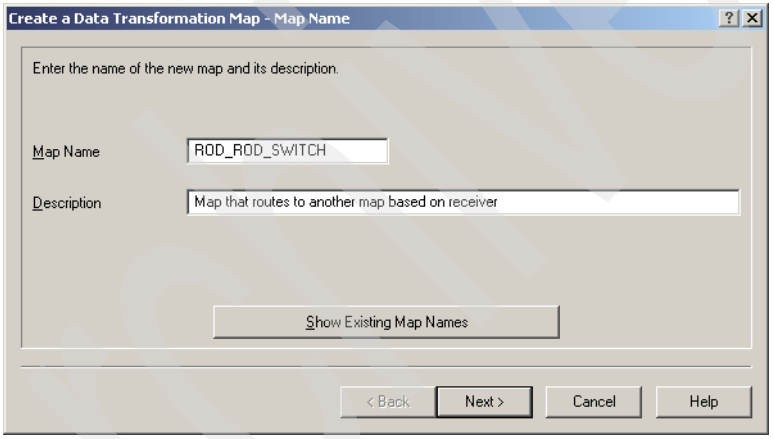


Figure 16-1 Create new map

3. Navigate through the Create a Data Transformation Map window providing the details that follow. The map should be defined as a target-based map with a record-oriented data source syntax type. The source dictionary is the CUSTOM_ROD and the source ROD document definition is PURCHASE_ORDER.
4. The target document definition should be the same, that is source dictionary CUSTOM_ROD and source ROD with document definition PURCHASE_ORDER.
5. After reviewing the details provided to create our new Data Transformation map, click **Finish**. Review the selections in the Confirmation window (Figure 16-2 on page 543).

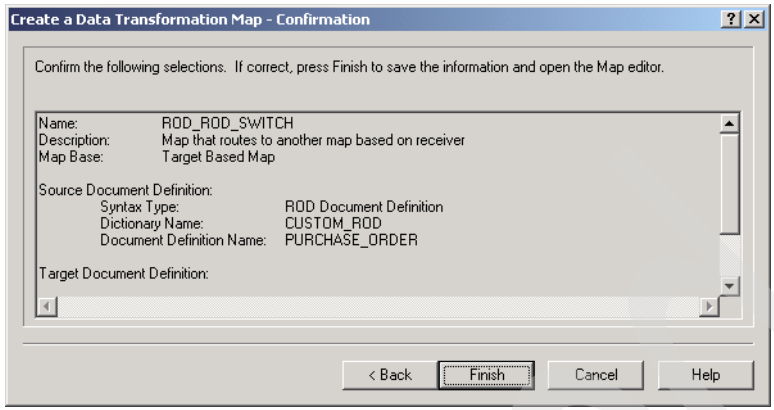


Figure 16-2 Confirm selections

6. Our new map opens on the Details tab page of the Data Transformation Map Editor. We can now create the logic that will allow our map to call another map based on the receiver ID defined in the source document.
7. Expand the HEADER record in the Mapping Command window. Right-click the **SUPPLIER_NUMBER** field and select **Insert Within** → **Command** → **If** (Figure 16-3). The Mapping Command Editor opens.

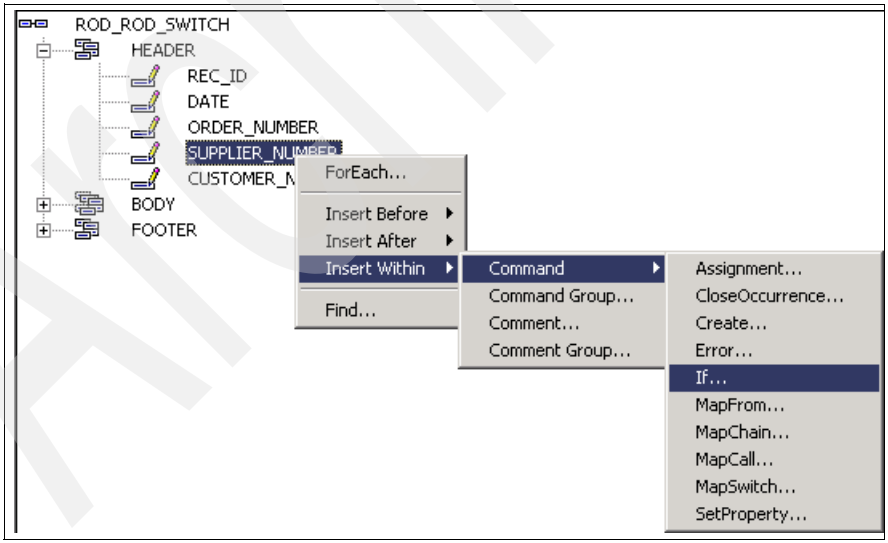


Figure 16-3 Insert If command

8. To test the SUPPLIER _NUMBER value, we use the StringComp() function discussed earlier. In this example however, we show how the TrimRight()

function can be nested within the StrComp() function. Select the **expression** value from the Mapping Command Editor, right-click and choose **Functions** → **StrComp**.

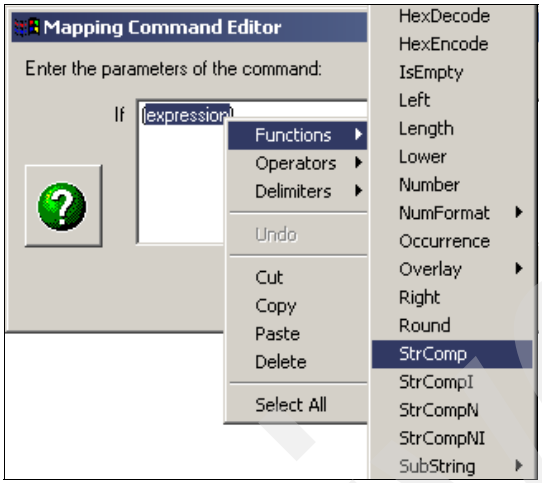


Figure 16-4 Create test expression

9. Select the **charValue1** string from within the StrComp() function then right click and choose **Functions** → **TrimRight** → **TrimRight(charValue)** as shown in Figure 16-5 on page 545. Expand the HEADER record in the Source Document Definition window and drag the SUPPLIER_NUMBER field over the charValue string used to define a parameter to the TrimRight() function.

10. The final step in completing this mapping command is to provide a value that will be compared against the SUPPLIER_NUMBER field. Overwrite the charValue2 string with `companyx`. The StrComp() function should return a value of 0 if the two values are equal. Enter `EQ 0` after the StrComp() function, but before the closing right parenthesis. Your mapping command should appear as shown in Figure 16-6 on page 545. Click **OK** to close the Mapping Command Editor and return to the Data Transformation Map Editor.

Archived

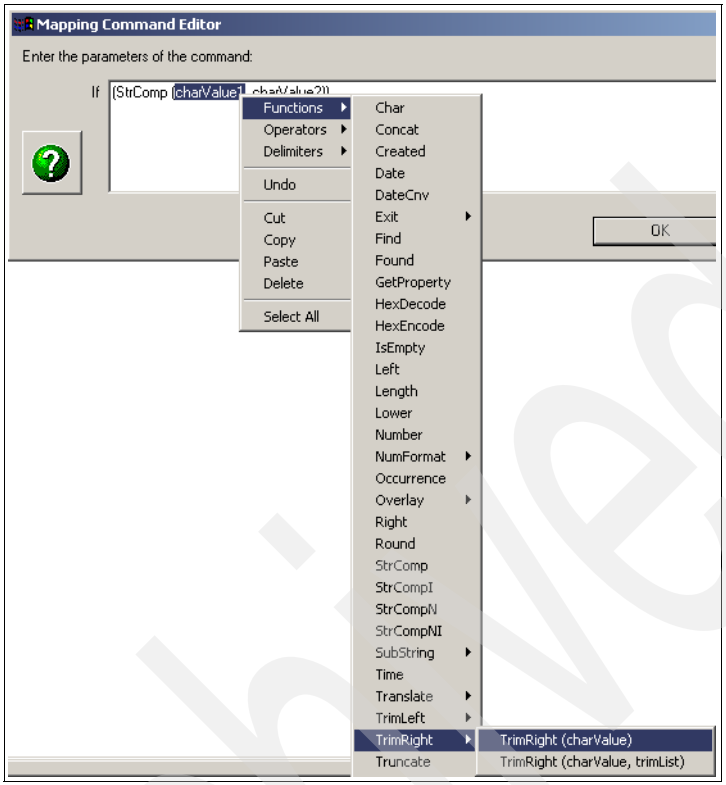


Figure 16-5 Create comparison value

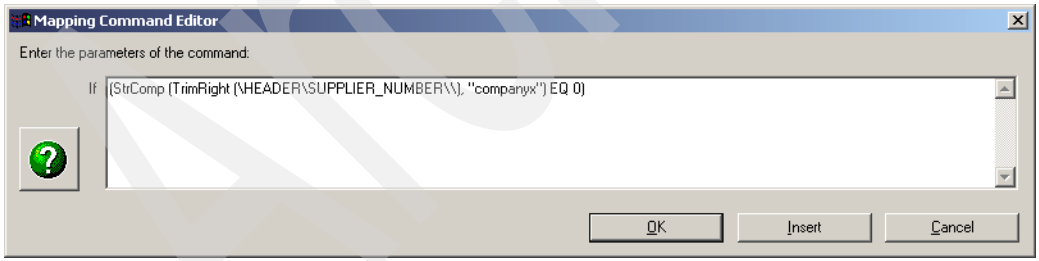


Figure 16-6 Mapping command

11. If the results of our string comparison function reveal that the value contained in the SUPPLIER_NUMBER field matches companyx, then we require that the source document be transformed by a particular map. This is a map other than the current ROD_ROD_SWITCH map. To route our source message to a different map, we use the MapSwitch command.

Archived

12. From the Mapping Command window, right-click the **If** mapping command just created. Select **Insert Within** → **Command** → **MapSwitch. . .** The Mapping Command Editor window opens and displays a template for the MapSwitch command.

13. Overwrite the mapName value that is passed as a parameter to the MapSwitch() function with the name of the map to be called, for example `ROD_ROD_FLAT` (Figure 16-7).

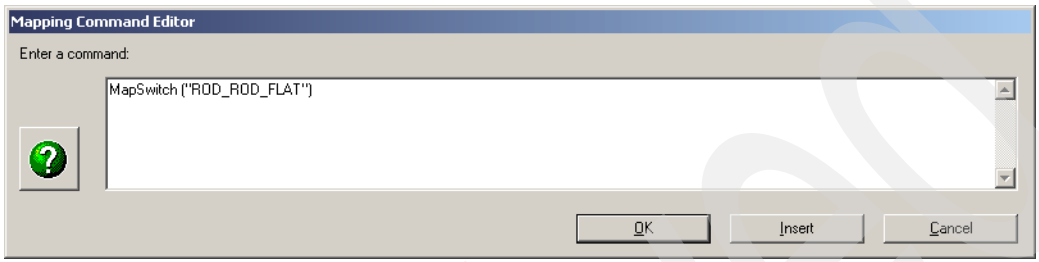


Figure 16-7 Enter map name for MapSwitch

14. This MapSwitch command instructs the Data Interchange Services to process the inbound message with a data transformation map called ROD_ROD_FLAT if the supplier number contained in the source message is companyx.

> **Important:** We overwrite the mapName value because the map we hope to switch to has not yet been defined in the Data Interchange Services client. If the map we want to switch to is already defined in the Data Interchange Services client, we could simply highlight the mapName value, right-click and select **Values** → **Map Name...** A list of maps that already exists in the database will display. Select the map name and click **OK** to insert a mapName value in our MapSwitch() function.

15. The mapping commands for our HEADER record should appear as shown in Figure 16-8 on page 547.

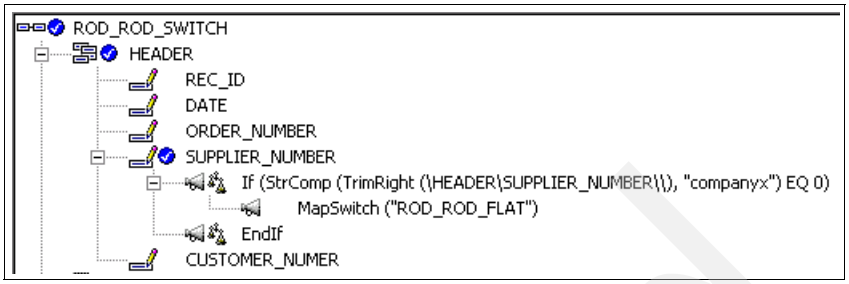


Figure 16-8 Mapping commands for HEADER

Next we build a mapping command that instructs our map to process the source message by another map in all other scenarios. That is, when the supplier number received in the source message is not companyx.

To create this command, follow these steps:

1. Right-click the **If** command created earlier and select **Insert After** → **Command** → **Else** (Figure 16-9).

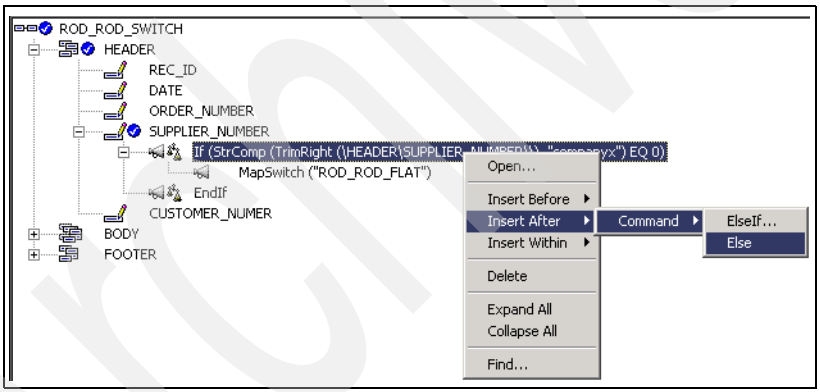


Figure 16-9 Add Else command

2. A new Else command is entered in the Mapping Command window. Right-click this Else mapping command and select **Insert Within** → **Command** → **MapSwitch. . .** (Figure 16-10 on page 548).

Archived

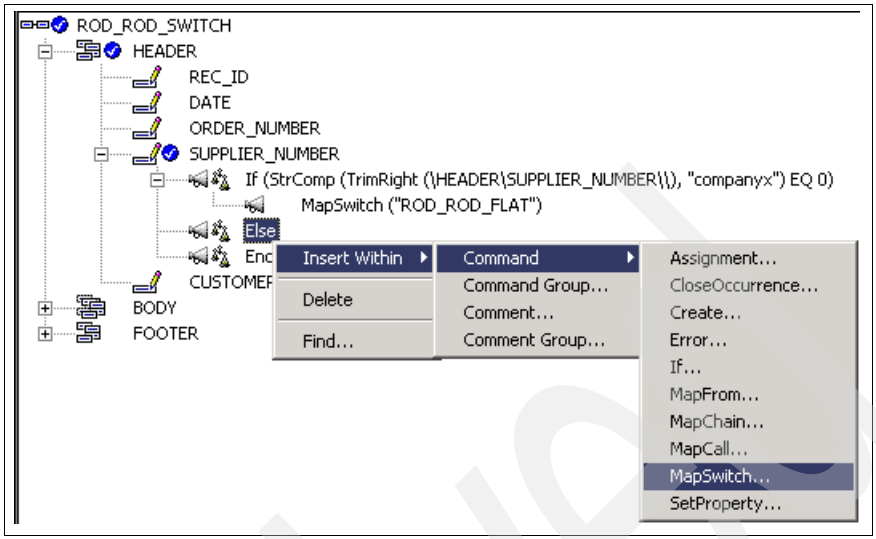


Figure 16-10 Insert MapSwitch

3. When the Mapping Command Editor opens, overwrite the mapName value with the name of the map to be called, for example `ROD_ROD_ORDER`. This map will be executed when our source document contains a supplier number other than companyx. Click **OK** to close the Mapping Command Editor. The Mapping Command window of our ROD_ROD_SWITCH map should now appear as shown in Figure 16-11.

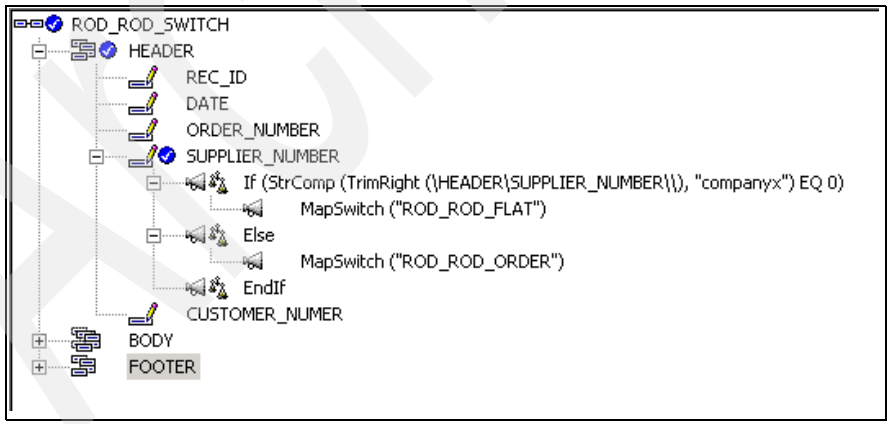


Figure 16-11 Mapping command window

4. Save and close the ROD_ROD_SWITCH map. Finally, compile the map.

Our next step is to create the two ROD to ROD maps that are defined in each of the MapSwitch commands used previously namely:

- ROD_ROD_ORDER
- ROD_ROD_FLAT

The ROD_ROD_ORDER data transformation map is, essentially, a one-to-one map. This allows a document of type PURCHASE_ORDER to pass through Data Interchange Services without actually changing format.

Create a new data transformation map in the Data Interchange Services client using the following details:

- Map Name: `ROD_ROD_ORDER`
- Target-based map
- Syntax type of source document definition: Record Oriented Data
- Source Dictionary: CUSTOM_ROD
- Document Definition Name: PURCHASE_ORDER
- Syntax type of the target document definition: Record Oriented Data
- Target Dictionary: CUSTOM_ROD

1. When we have defined the new map, the Details tab of the Data Transformation Map Editor opens.
2. For each of the fields in the HEADER record, drag the field from the source document definition and drop the field on its corresponding target field. When complete, the HEADER record in the Mapping Command window should appear as shown in Figure 16-12.

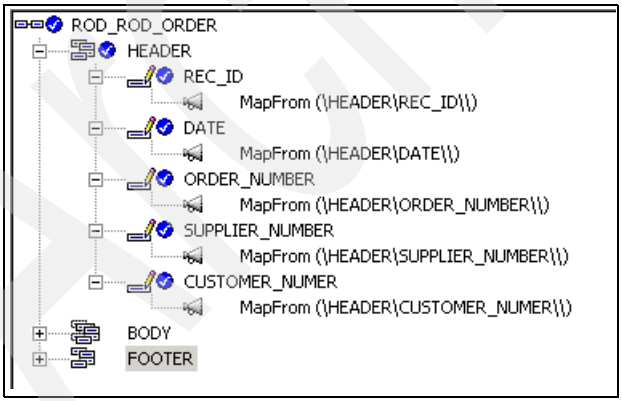


Figure 16-12 Completed HEADER mapping commands

3. Our next step is to ensure that all instances of the BODY record in the source message get mapped to a BODY record in the target document. To do this we use the **`For Each`** mapping command.

4. From the Source Document Definition window, select the **BODY** record and drag it onto the BODY record in the Target Document Definition. This should generate a new ForEach mapping command in the BODY record that is displayed in the Mapping Commands window.

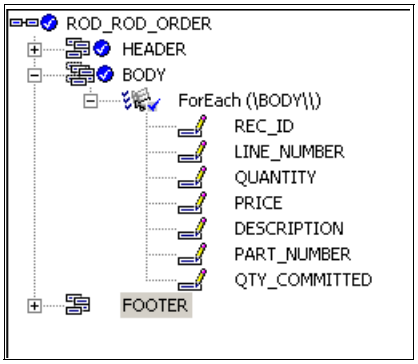


Figure 16-13 Single BODY mapping commands

5. After this **`ForEach`** command is generated, we can drag each field defined in the BODY record onto its corresponding target field. When complete, the mapping commands in the BODY record should appear as shown Figure 16-14.

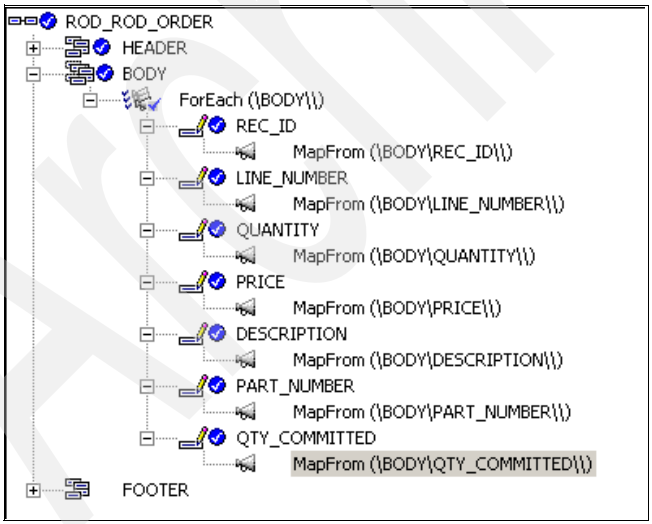


Figure 16-14 Mapping commands for BODY iteration

6. The FOOTER record is defined as occurring only once, so we need not worry about mapping multiple instances of the record. As before, drag and drop

Archived

each of the fields in the source record onto its corresponding field in the target record definition.

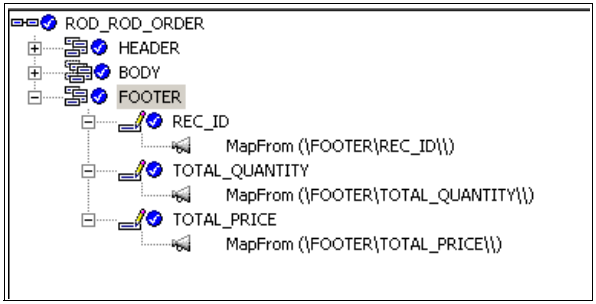


Figure 16-15 FOOTER mapping commands

7. We can now save and close our ROD_ROD_ORDER map. The map should also be compiled.

Our final map required the creation of a new ROD document. This new ROD document consists of the same fields and records as the PURCHASE_ORDER document defined in Chapter 15, “Mapping” on page 473. Because of this, we can simply copy the original PURCHASE_ORDER document to create our new ROD document definition. To copy the PURCHASE_ORDER document definition, open the Record Oriented Data functional area and open the ROD Document Definitions tab.

1. Select the **PURCHASE_ORDER** document definition from the list of displayed ROD Document Definitions. From the Data Interchange Services Client menu bar select **Actions** → **Copy. . .** .

   The Copy Object window opens (Figure 16-16).

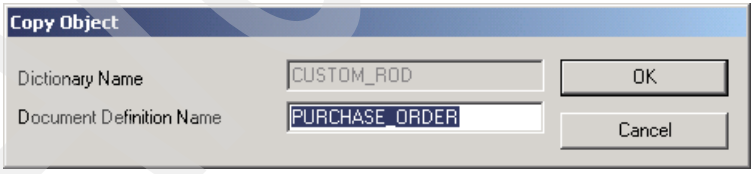


Figure 16-16 Copy object window

2. Change the Document Definition Name to `FLAT_ORDER` and click **OK**. A new row is added to the list of ROD document definitions displayed in the view. This new ROD document definition is identical to the original PURCHASE_ORDER apart from the name.

3. Open the FLAT_ORDER document definition from the list of ROD Document Definitions. Select the **General** tab and choose **No Delimiter** as the Record Delimiter. Save the ROD document definition.

   Now that we have created our new ROD document definition, we can create our ROD_ROD_FLAT data transformation map. The mapping commands in this map are identical to those used in the ROD_ROD_ORDER map. For this reason we can simply copy the ROD_ROD_ORDER map and alter the Target Document Definition used in the map.

4. To copy the ROD_ROD_ORDER map, open the Mapping functional area and select the **Data Transformation Maps** tab. Select the **ROD_ROD_ORDER** entry from the list of data transformation maps displayed then chose **Actions** → **Copy**. . . from the Data Interchange Services client menu bar. The Copy Data Transformation Map window opens.

Figure 16-17 Select map to copy

5. Change the Map Name to `ROD_ROD_FLAT` and click **OK**. A new row is created for the ROD_ROD_FLAT data transformation map. We now need to change the target document definition used in the map to FLAT_ORDER.

6. Open the Data Transformation Map Editor by double-clicking **ROD_ROD_FLAT** from the list of data transformation maps. Select the **General** tab and change the Document Definition Name of the Target Document Definition to `FLAT_ORDER` (Figure 16-18 on page 553).

Archived

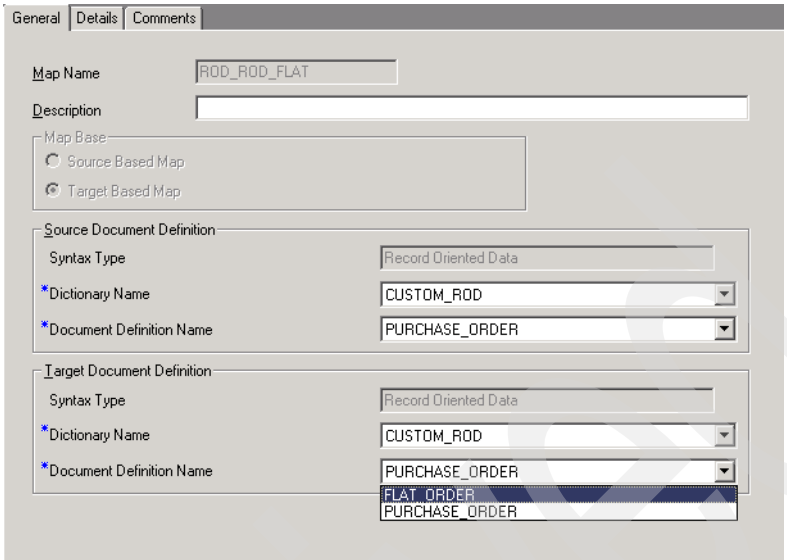


Figure 16-18 Enter detailed for copied map

7. The map is loaded again and the references in the mapping commands are updated. You can now open the **Details** tab and verify the Target Document Definition window displays FLAT_ORDER as the document definition, as shown in Figure 16-19.

Figure 16-19 Target Document Definition

8. Save and close the new ROD_ROD_FLAT data transformation map.
9. Compile the map.

16.1.1 Exporting the maps to a runtime database

We now have all the data transformation maps that are required to complete our scenario. Now that we have defined and compiled each of these maps in the Development environment of the Data Interchange Services client, we can export our maps to a runtime database. For each of the maps we just created, follow these steps to export the control string to the runtime database.

1. Open the Mapping functional area from the Data Interchange Services client and ensure the Compile Required column displays Yes for:
   - ROD_ROD_SWITCH
   - ROD_ROD_ORDER
   - ROD_ROD_FLAT
2. If any of the entries show as No in the Compile Required column, compile the map before proceeding.
3. Open the **Control Strings** tab from the Mapping functional area. You should see a row for each of the data transformation maps used in this scenario in the list of control strings. Starting with ROD_ROD_SWITCH, select the control string from the list shown and chose **Actions** → **Export to Other Database. . .** from the Data Interchange Services menu bar (Figure 16-20).

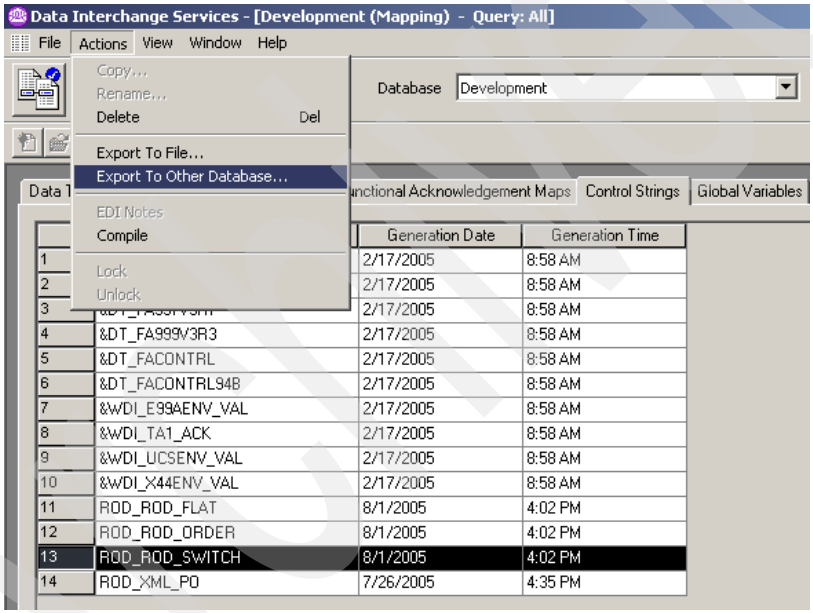


Figure 16-20 Select map to be exported

4. The Select a Database window opens. Select **WPGV6DB** from the list of available databases and click **OK** (Figure 16-21 on page 555).

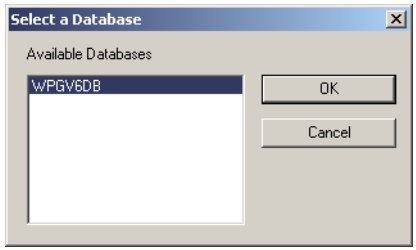


Figure 16-21 Select database for import

5. Enter a valid user ID and password to connect to the runtime database. The Export with Control String window opens. For each of the maps defined, there is no need to select any of the Referenced Types, so simply select **OK**. The Execution Status Window opens (Figure 16-22).

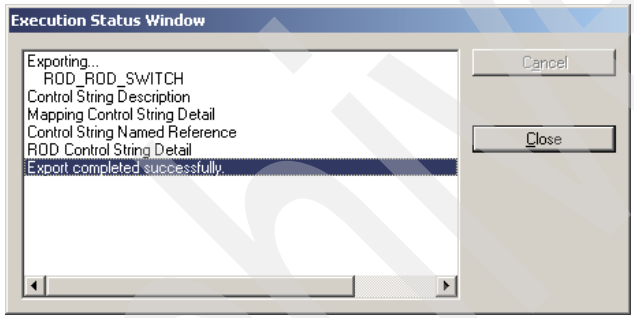


Figure 16-22 Export status

6. When this window displays `Export completed successfully`, click **Close** to return to the Control Strings tab of the Mapping functional area. Repeat these steps for both ROD_ROD_ORDER and ROD_ROD_FLAT.

16.2 Configure WebSphere Partner Gateway at Company E

Because we exported the control string to the runtime database of WebSphere Partner Gateway in the previous section, we can now look at the console of WebSphere Partner Gateway to verify that the transformation map has been uploaded successfully.

Archived

16.2.1 Verify that the map has been uploaded

Follow the steps to verify that the map has been uploaded:

1. While logged in as hubadmin to the WebSphere Partner Gateway server on Company E, go to **Hub Admin** → **Hub Configuration** → **Maps** → **Transformation Maps.**
2. We can see the maps listed as shown in Figure 16-23.

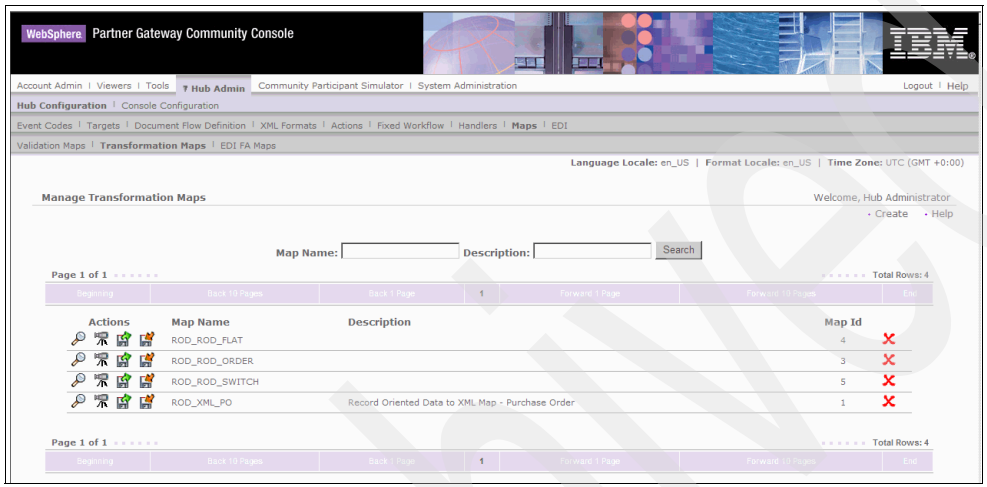


Figure 16-23 List of transformation maps uploaded

3. Click the **magnifying glass** icon next to the uploaded map, **ROD_ROD_SWITCH**, to see the document flow definition, as shown in Figure 16-24 on page 557.

Archived

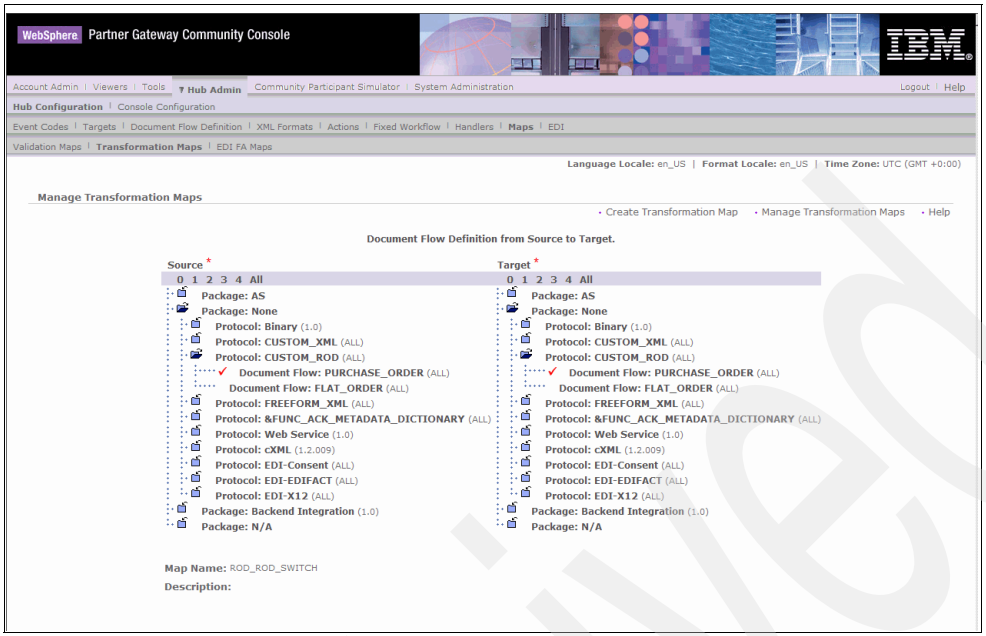


Figure 16-24 document flow definition defined by map ROD_ROD_SWITCH

4. Similarly, click the **magnifying glass** icons next to the maps **ROD_ROD_ORDER** and **ROD_ROD_FLAT**, to view the document flow definitions defined by them. These are shown in Figure 16-25 on page 558 and Figure 16-26 on page 559, respectively.

Archived

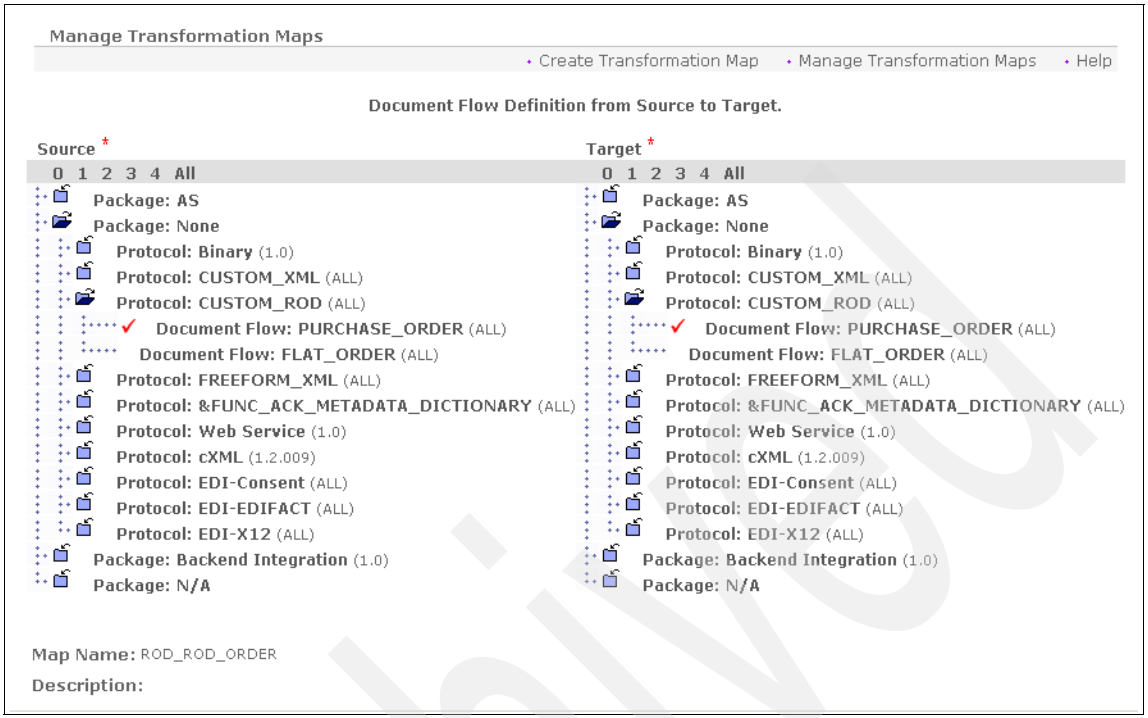


Figure 16-25 document flow definition defined by map ROD_ROD_ORDER

Archived

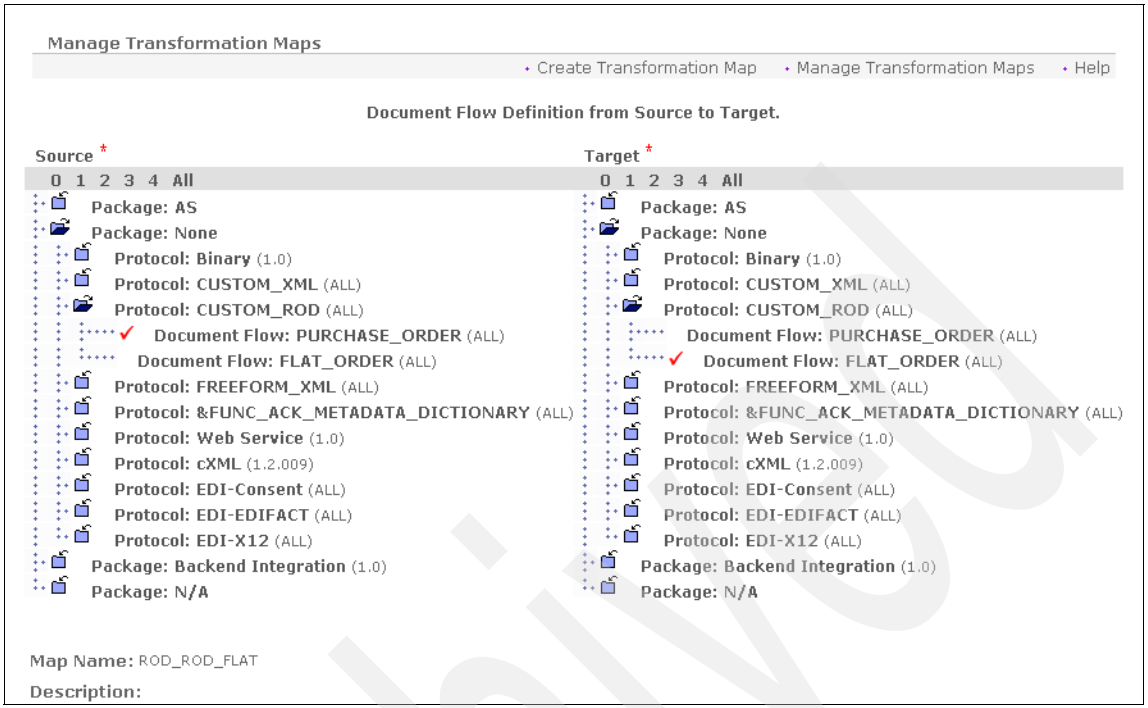


Figure 16-26 document flow definition defined by map ROD_ROD_FLAT

16.2.2 Update the configuration of Company E

To demonstrate a ROD to ROD transformation scenario, we can continue to use the file directory target, RODFileTarget created in 15.6.2, “Create a target for ROD document” on page 516.

Update the B2B capabilities of Company E, following these steps:

1. While still logged in as hubadmin select **Hub Admin** → **Hub Configuration** → **Document Flow Definition.**
2. As shown in Figure 16-27 on page 560, verify that the required document flow definition for Company E is enabled.

Archived

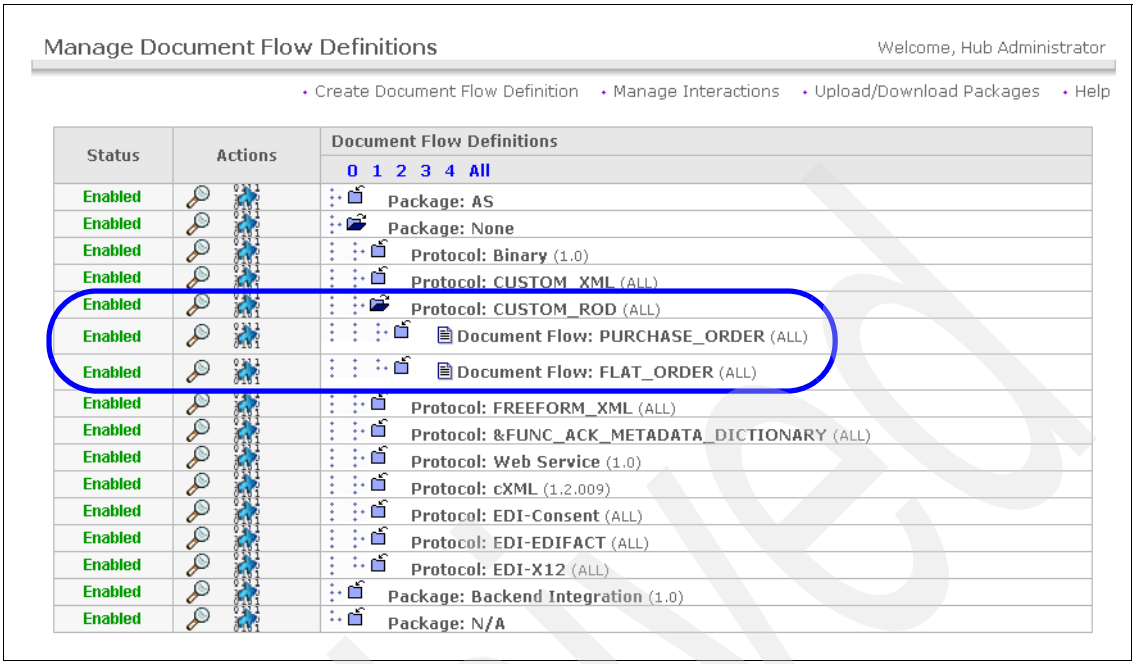


Figure 16-27 Verifying if the required document flow definition for Company E is enabled

3. Select **Account Admin** → **Profiles** → **Community Participant.**
4. Click **Search**. From the list of participants displayed, click the **magnifying glass** icon next to **Company E**. This opens Company E's profile. Click **B2B capabilities**, which opens a window as shown in Figure 16-28 on page 561.
5. **Enable** None(Package) / CUSTOM_ROD(Protocol) / PURCHASE_ORDER (Document Flow) , as the B2B capability for Company E.

Archived

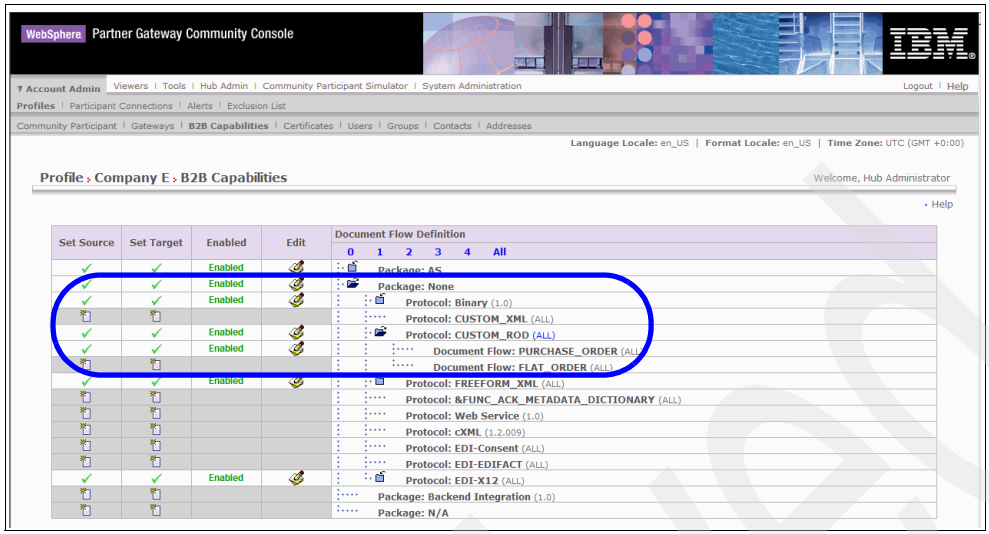


Figure 16-28 B2B capability of Company E

6. When the appropriate B2B capability is enabled, under the profile of Company E, click **Gateways**.
7. A list of Gateways for Company E is displayed. We use the existing FileSystemGateway for our discussion in this chapter.

16.2.3 Update the configuration for Company X

Follow these steps to update the configuration for Company X:

1. While still logged in as hubadmin, select **Hub Admin** → **Hub Configuration** → **Document Flow Definition.**
2. Because Company X has to have the B2B capability to receive None/CUSTOM_ROD/PURCHASE_ORDER, first, verify if the required B2B capability is enabled, which we have already done as shown in Figure 16-27 on page 560.
3. Select **Account Admin** → **Profiles** → **Community Participant**.
4. Click **Search**. From the list of participants displayed, click the **magnifying glass** icon for **Company X**. This opens Company X's profile. Click **B2B capabilities**, which opens a window as shown in Figure 16-29 on page 562.
5. **Enable** None(Package) / CUSTOM_ROD(Protocol) / PURCHASE_ORDER (Document Flow) , as the B2B capability for Company X.

Archived

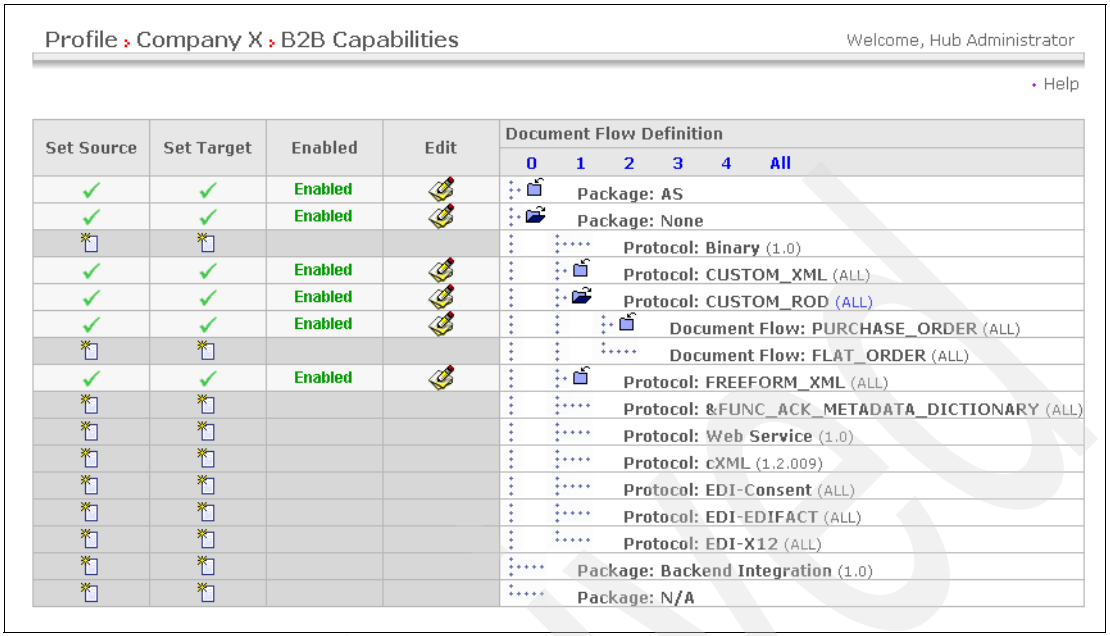


Figure 16-29 B2B capability of Company X

6. Click **Gateways**.

   This brings up a list of gateways for Company X, as shown in Figure 16-30.

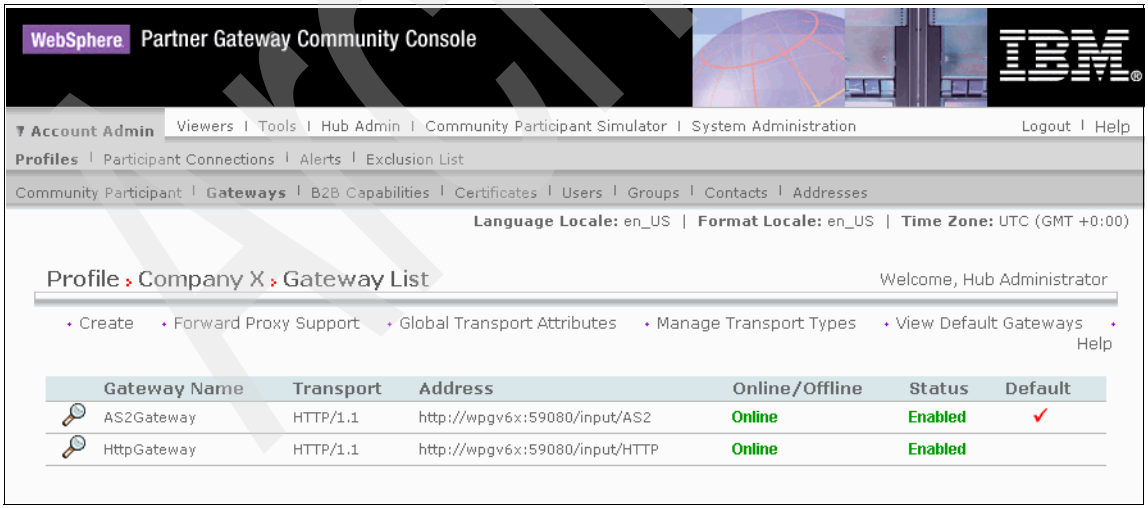


Figure 16-30 List of gateways for Company X

We will use the HttpGateway for this scenario.

16.2.4 Update the configuration for Company A

Follow these steps to update the configuration for Company A:

1. While still logged as hubadmin, select **Hub Admin** → **Hub Configuration** → **Document Flow Definition.**
2. Because Company X has to have the B2B capability to receive None/CUSTOM_ROD/PURCHASE_ORDER first, verify that the required B2B capability is enabled. We have already done this, as shown in Figure 16-27 on page 560.
3. Select **Account Admin** → **Profiles** → **Community Participant**.
4. Click **Search**. From the list of participants, click the **magnifying glass** icon next to **Company A**. This opens Company A's profile. Click **B2B capabilities**, which opens a window as shown in Figure 16-31.

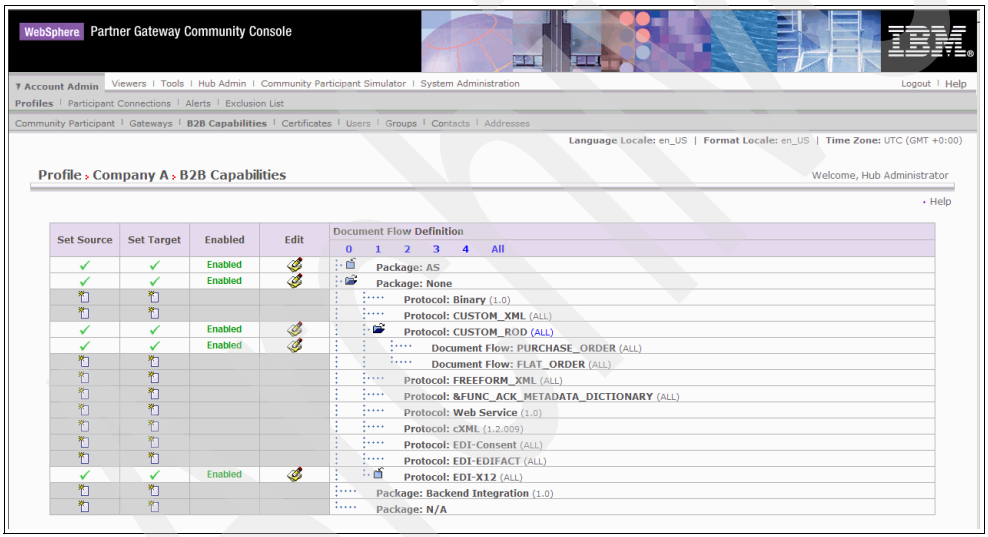


Figure 16-31 B2B capabilities of Company A

5. Click **Gateways**.

   This brings up a list of gateways for Company A, as shown in Figure 16-32 on page 564.

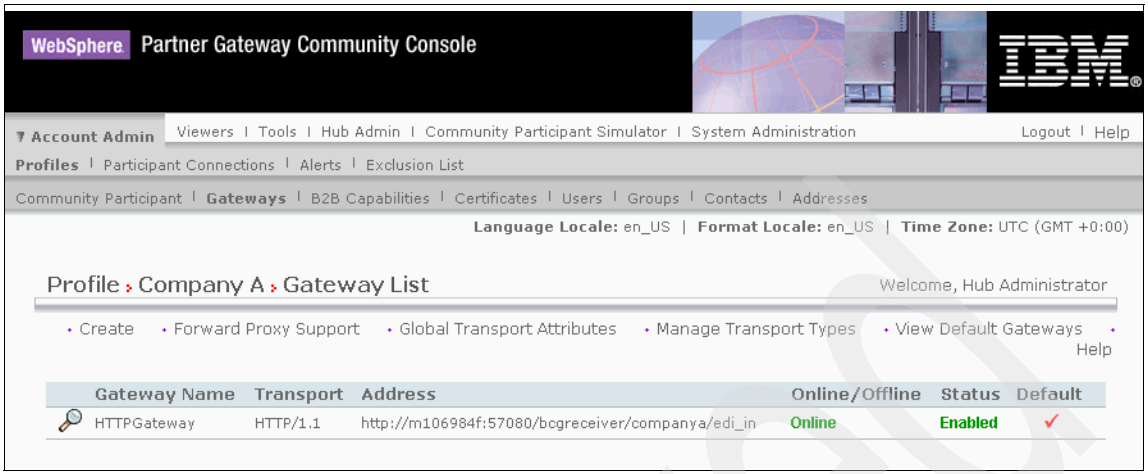


Figure 16-32 List of gateways for Company A

16.2.5 Create an interaction

After updating the profiles for Company E, Company X and Company A, we now need to create a new interaction between the appropriate source and target types, also selecting the appropriate transformation map and appropriate action.

Follow the steps to create the interaction:

1. While still logged in as hubadmin, select **Hub Admin** → **Hub Configuration** → **Document Flow Definition.**
2. Click **Manage Interactions**.
3. In the Manage Interactions window, click **Create Interaction**.
4. Create an interaction between source as None/CUSTOM_ROD/PURCHASE_ORDER and target as None/CUSTOM_ROD/PURCHASE_ORDER, as shown in Figure 16-33 on page 565.
5. Select the map as **ROD_ROD_SWITCH** and Action as **ROD Translate.**

Archived

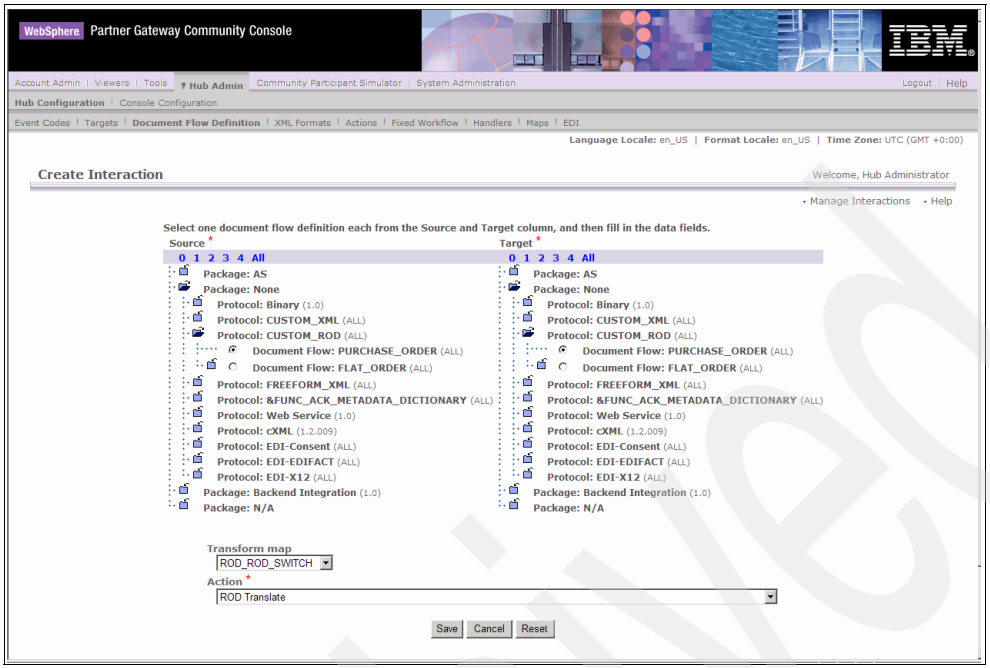


Figure 16-33 Creating an interaction

6. Click **Save**.
7. In the Manage Interactions window, click **Search**. This returns a list of interactions available, including the new interaction we have just created.

16.2.6 Create participant connections

In this section, we create the required participant connections to enable the transformation and transfer of the ROD document.

In this scenario, we are using the same ROD DICTIONARY and DOCUMENT definitions at both Company A and Comany X. Company E will transform an ROD document into the appropriate form, based on the recipient identifier mentioned in the ROD document. If the document has to be routed to Company A, the ROD_ROD_SWITCH map switches to execute the map ROD_ROD_ORDER map which gives a segmented output.

Similarly, if the recipient identifier is that of Company X, the ROD_ROD_SWITCH map switches to execute the map ROD_ROD_FLAT map which gives an unsegmented output. The switching of the map is internal and is the result of executing the ROD_ROD_SWITCH map. As a result, we configure

Archived

our participant connection using just the ROD_ROD_SWITCH map as the transformation map.

Create a participant connection between Companies E and X

Follow these steps to create the participant connections:

1. While logged in as hubadmin, select **Account Admin** → **participant connections.**
2. Select the source as **Company E** and target as **Company X,** and click **Search**.

   This lists the available participant connections existing between Company E and Company X, as shown in Figure 16-34.

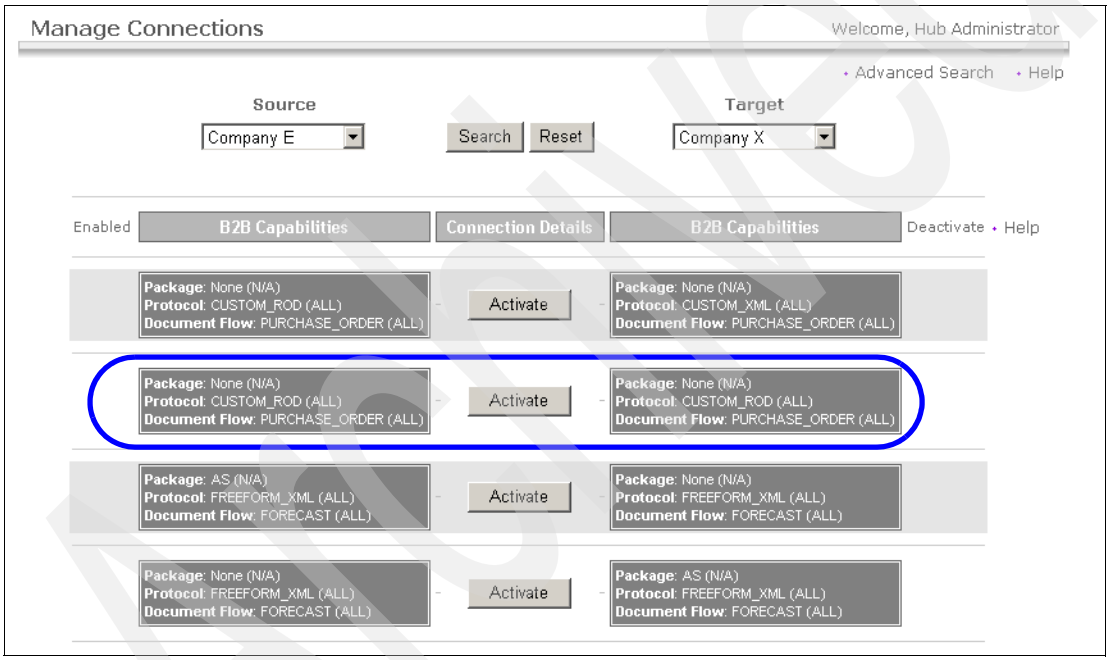


Figure 16-34 List of participant connections between Company E and Company X

3. Click **Activate** against the participant connection highlighted in Figure 16-34. This results in Figure 16-35 on page 567.

Archived

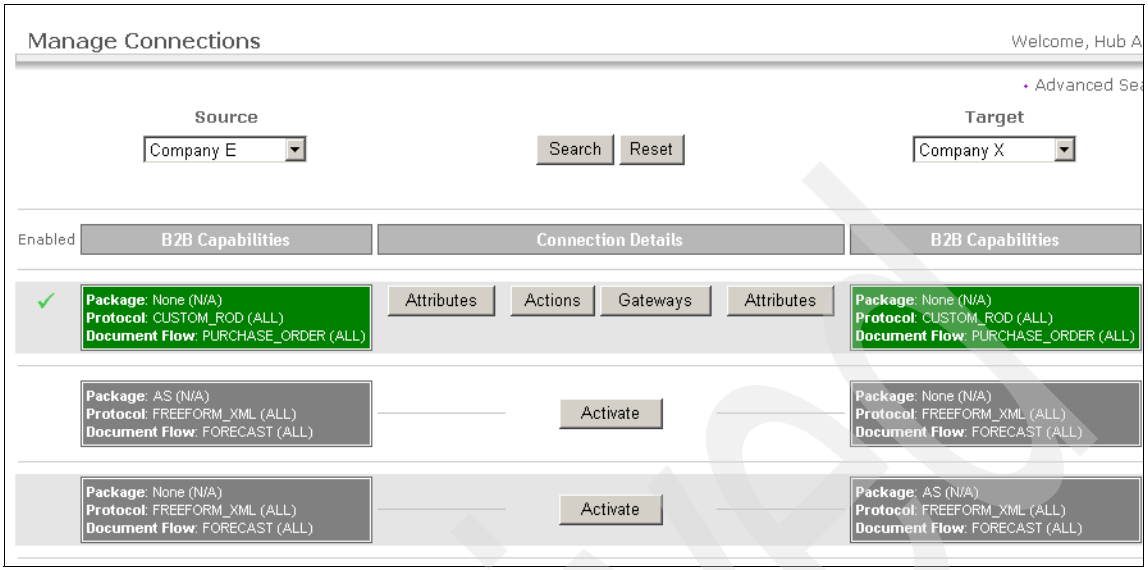


Figure 16-35 Active participant connection between Company E and Company X

4. Click **Gateways** to open a window.
5. Select **HttpGateway** as the target gateway, as shown in Figure 16-36. Click on **Save**.

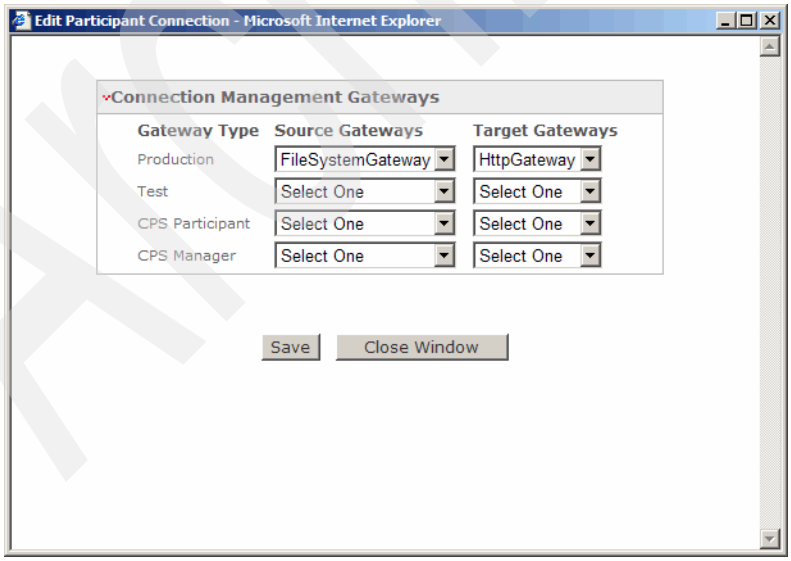


Figure 16-36 Selecting the gateway for participant connection

Archived

Create a participant connection between Company E and A

Follow these steps to create the participant connection:

1. While logged in as hubadmin, select **Account Admin** → **Participant Connections.**
2. Select the source as **Company E** and target as **Company A**, and click **Search.** This lists the available participant connections existing between Company E and Company A, as shown in Figure 16-37.

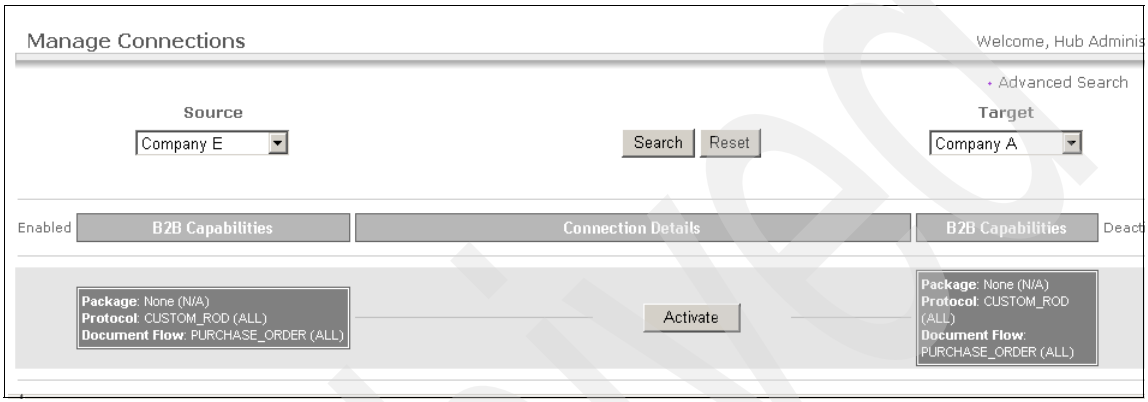


Figure 16-37 List of participant connections between Company E and Company A

3. Click **Activate**, resulting in Figure 16-38.

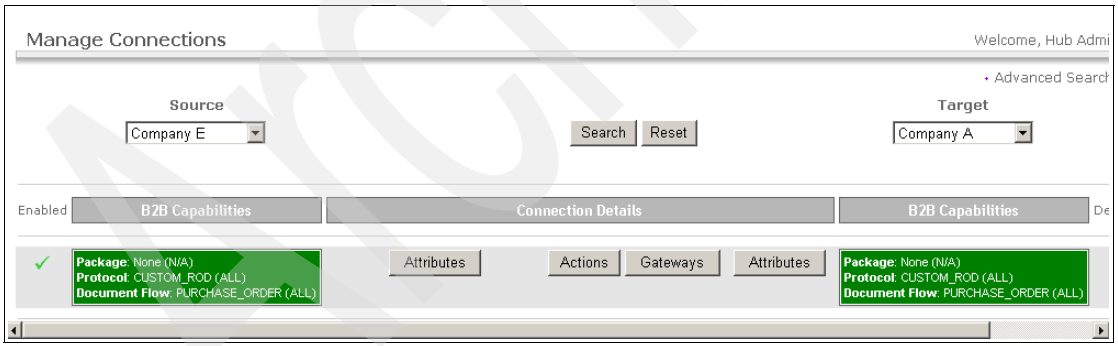


Figure 16-38 Active participant connection between Company E and Company A

4. Click **Gateways**. Use the default **HttpGateway** for Company A and the default **FileSystemGateway** for Company E.

Archived

16.3 Validate communication

After the configuration steps are complete, we can validate the communication. Drop the ROD file in /wpgv6/data/companye/rod_out folder. When this file is picked up from here, go to the Document Viewer to see the status of the document sent.

Communication between Company E and Company X

Figure 16-39 shows the Document Viewer results. We see that the document was successfully sent to the target as there is a status of green check mark against the document.

Figure 16-39 Document viewer showing successful transfer

Click the document icon against the source to see the source ROD document, shown in Figure 16-40 on page 570.

Archived

Raw Document Viewer Copy Help

Document ID: -

Doc Time Stamp: -

	Participants	Business IDs	Document Flow
Source:	Company E	companye	CUSTOM_ROD (ALL) **PURCHASE_ORDER**: PURCHASE_ORDER (ALL)
Target:	Company X	companyx	CUSTOM_ROD (ALL) **FLAT_ORDER**: FLAT_ORDER (ALL)

UTF8

Transport Header

```
content-length: 298
ReceiverDestinationType: Production
x-aux-in-file-name: purchaseOrder.rod
MsgLengthIncHeaders: 298
requestURI: D:\wpgv6\data\companye\rod_out
OriginalFileName: D:\wpgv6\data\companye\rod_out\Documents\Production\purchaseOrder.rod
```

Initial Document

```
HED27072005 P368876 companye companyx
BOD1 10 4.45 WIDGET_A 00024L6654
BOD2 119 3.01 WIDGET_B 00098P6547
BOD3 8 9.01 WIDGET_C 00021L4971
BOD4 1 21.66 WIDGET_D 00043P9600
FOT4 138 496.43
```

Close Window

Figure 16-40 Source ROD document

To view the target document, click the document icon against the target. The transformed ROD document is shown in Figure 16-41.

Raw Document Viewer Copy Help

Document ID: -

Doc Time Stamp: -

	Participants	Business IDs	Document Flow
Source:	Company E	companye	CUSTOM_ROD (ALL) **PURCHASE_ORDER**: PURCHASE_ORDER (ALL)
Target:	Company X	companyx	CUSTOM_ROD (ALL) **FLAT_ORDER**: FLAT_ORDER (ALL)

UTF8

Transport Header

```
POST /input/AS2 HTTP/1.1
Content-Type: application/octet-stream
Content-Length: 328
Connection: Keep-Alive
Keep-Alive: timeout=5, max=10000
Host: wpgv6x:59080
Content-Disposition: attachment; filename=purchaseOrder.rod
x-out-filename: purchaseOrder.rod
Authorization: Basic Y29tcGFueWU6cGFydG5lcl9l
```

Translated Document

```
HED27072005 P368876 companye companyx BOD1 10 4.45 WIDGET_A 00024L6654 BOD2 119 3.01 WIDGET_B 00098P6547
BOD3 8 9.01 WIDGET_C 00021L4971 BOD4 1 21.66 WIDGET_D 00043P9600 FOT0000000004138 496.43
```

Close Window

Figure 16-41 Target ROD document

Archived

Communication from Company E to Company A

Figure 16-42 shows the Document Viewer results. We see that the document was successfully sent to the Company A as there is a status of green check mark against the document.

Figure 16-42 Document viewer showing successful transfer

Click the document icon against the source to see the source ROD document, shown in Figure 16-43 on page 572.

Archived

Raw Document Viewer Copy Help

Document ID: -

Doc Time Stamp: -

	Participants	Business IDs	Document Flow
Source:	Company E	companye	CUSTOM_ROD (ALL) **PURCHASE_ORDER**: PURCHASE_ORDER (ALL)
Target:	Company A	companya	CUSTOM_ROD (ALL) **PURCHASE_ORDER**: PURCHASE_ORDER (ALL)

UTF8

Transport Header

```
content-length: 298
ReceiverDestinationType: Production
x-aux-in-file-name: purchaseOrder.rod
MsgLengthIncHeaders: 298
requestURI: D:\wpgv6\data\companye\rod_out
OriginalFileName: D:\wpgv6\data\companye\rod_out\Documents\Production\purchaseOrder.rod
```

Initial Document

```
HED27072005 P368876 companye companya
BOD1 10 4.45 WIDGET_A 00024L6654
BOD2 119 3.01 WIDGET_B 00098P6547
BOD3 8 9.01 WIDGET_C 00021L4971
BOD4 1 21.66 WIDGET_D 00043P9600
FOT4 138 496.43
```

Close Window

Figure 16-43 Source ROD document

To view the target document, click the document icon against the target. The transformed ROD document is as shown in Figure 16-44.

Raw Document Viewer Copy Help

Document ID: -

Doc Time Stamp: -

	Participants	Business IDs	Document Flow
Source:	Company E	companye	CUSTOM_ROD (ALL) **PURCHASE_ORDER**: PURCHASE_ORDER (ALL)
Target:	Company A	companya	CUSTOM_ROD (ALL) **PURCHASE_ORDER**: PURCHASE_ORDER (ALL)

UTF8

Transport Header

```
POST /bcgreceiver/companya/edi_in HTTP/1.1
Content-Type: application/octet-stream
Content-Length: 340
Connection: Keep-Alive
Keep-Alive: timeout=5, max=10000
Host: m106984f:57080
Content-Disposition: attachment; filename=purchaseOrder.rod
x-out-filename: purchaseOrder.rod
```

Translated Document

```
HED27072005 P368876 companye companya
BOD1 10 4.45 WIDGET_A 00024L6654
BOD2 119 3.01 WIDGET_B 00098P6547
BOD3 8 9.01 WIDGET_C 00021L4971
BOD4 1 21.66 WIDGET_D 00043P9600
FOT0000000004138 496.43
```

Close Window

Figure 16-44 Target ROD document

Archived

Archived

Archived

Part 5

Native mapping support for EDI standards

Archived

© Copyright IBM Corp. 2005. All rights reserved.

Archived

17

Introduction to EDI technology

This chapter provides a brief overview of the electronic data interchange (EDI) concept and technology. It also describes WebSphere Data Interchange and how the product fits into a typical EDI solution.

Archived

© Copyright IBM Corp. 2005. All rights reserved.

17.1 EDI terms and concepts

Electronic data interchange is a concept that has been in commercial use for more than 30 years. It is widely accepted by companies all over the world as the way to electronically exchange business data.

Over the years, we have seen a variety of interpretations of the term EDI. A common and basic definition of *EDI* is: *the transfer of business data between computer applications using a mutually agreed standard to describe the data contained in the message.*

Typically this means that business data is extracted from a company's internal application in an application-specific data format. This data extraction can be implemented in several ways. It can be a daily batch job reading information in a database and generating a file in an application-specific format. This user format data file is then translated into a standardized format such as EDI or XML and transmitted over a network to a trading partner.

An alternative technique is to generate WebSphere MQ messages from within the application. That WebSphere MQ message is again in an application-specific format.

For both techniques, the message ends in some way at a translator component, where the application-specific structure of the data is mapped to an EDI standard format, as shown in Figure 17-1.

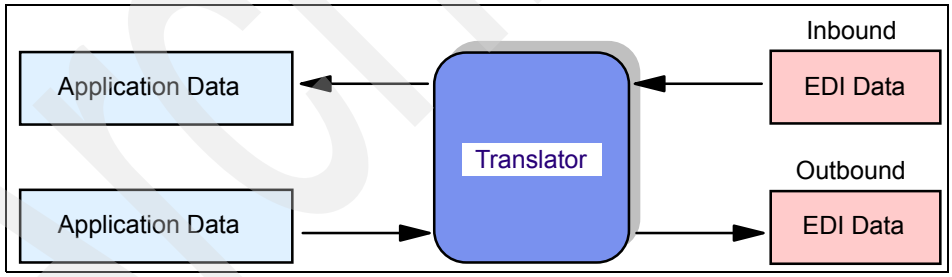


Figure 17-1 The role of the translator

Archived

The receiving trading partner then retranslates the received EDI or XML message back into an application-readable format that fits into their processes, as shown in Figure 17-2.

EDI-based information exchange is usually a two-way process. Thus, the translator component is also used to translate incoming EDI messages into an application-specific format.

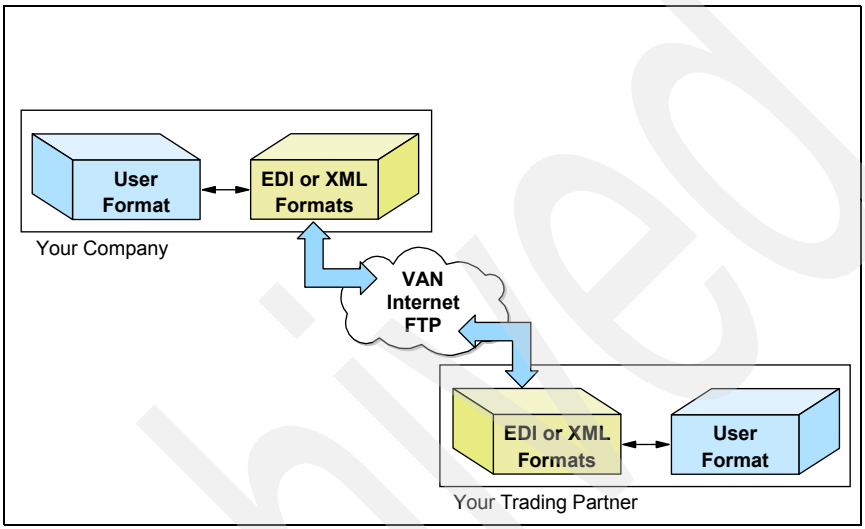


Figure 17-2 Basic EDI message flow showing translation and mapping

From a company perspective, the EDI concept means business integration and process automation. Business documents such as purchase orders, invoices, shipping notices, and price catalogs are exchanged between companies over a network in a structured and computer-processable format.

Figure 17-3 on page 580 shows a typical flow of actions and data between a buyer and a seller. Usually, a buyer asks for a quote. Then when a quote is received, a purchase order request can be sent by the buyer to the supplier. This information exchange is handled typically by a purchasing application at the buyer side and handled by a sales management application at the seller side.

When the goods are ready to be delivered, a shipping notice is sent by the seller to the buyer. This time, the information exchange is likely between different components of the IT infrastructure at both the seller and the buyer sides.

Thus, the use of EDI between two companies implies integration between the applications at each end. The purchase order generated by the purchase application needs to be known by the application used in the warehouse or by the accounting application.

Archived

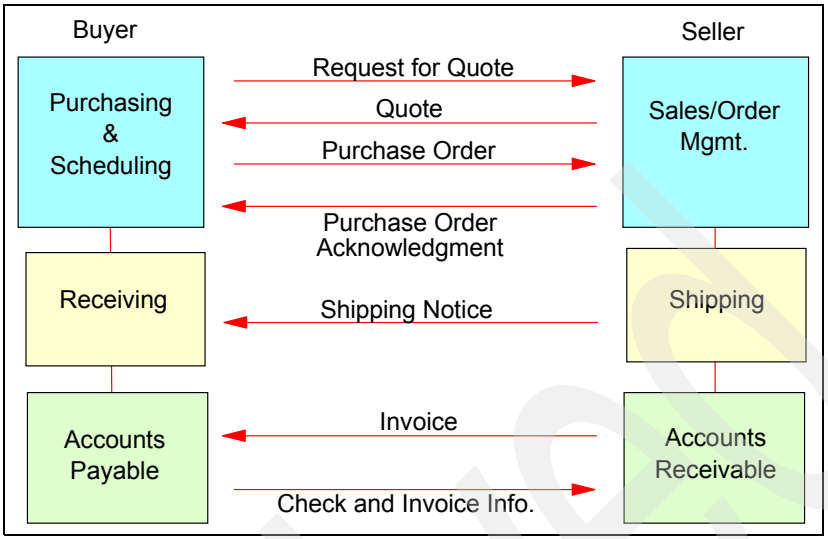


Figure 17-3 EDI and the business cycle

Since the early days of EDI, many new initiatives and techniques have been adopted by the market. Words that hardly existed at that time, such as the Web, XML, B2B, and business process management (BPM), are a natural part of today's business environment. The obvious question is, why is EDI still so important? Here are a few reasons:

- EDI is an important part of companies' B2B strategies.
- Of Fortune 500 companies, 95% use EDI.
- A total of 80% of business transactions are conducted with EDI value-added networks (VAN) today.
- EDI continues to deliver a significant return on investment (ROI).
- EDI continues to evolve in response to new enterprise and industry requirements, and competitive pressures (for example, HIPAA, AS1, AS2).

17.2 Benefits of EDI

The market is driving every business to act smarter and quicker and to be more visible. Much of this can be achieved using EDI. Even better, EDI can give companies a better knowledge of their markets, because it opens possibilities to collect and analyze information from the EDI transactions they are generating.

Among the most visible benefits of adopting EDI are:

- Reduction of data entry errors
- Reduced cycle time
- Minimizing paper use
- Improved relationships with your business partners
- More easily shared information in electronic form over the organization
- Improved inventory management

17.3 EDI components

The term EDI is a concept. It does not define any technique, nor point to any specified product or service. An EDI transmission can be divided into two logical parts: the message itself and the communication.

17.3.1 Message standards

Because the idea of EDI is to have a standardized message, several different standards have been developed and established over the years. The most commonly used message standards are:

- ANSI ASC X12: U.S. standard
- EDIFACT: United Nations recommended standard, used mainly in Europe
- UNTDI: U.K. retail standard
- ODETTE: European automotive industry
- Others such as HIPAA, VICS, VDA, and UCS

The standardized messages are built by components such as elements, segments, and transactions/messages. Between every object there is a separator.

The *elements* are the individually defined fields such as amount, name or quantity. Two or more elements can be grouped together, forming a *composite element*.

A *segment* is a set of elements or composite elements built to a logical entity such as name and address or pricing information.

An *envelope* contains overall information about the transaction or message, such as sender and receiver, type, and control values.

A set of segments put together in a specified order all wrapped in an envelope make a *transaction* or *message,* such as an invoice or a purchase order. The envelope contains information about sender and receiver, transaction or message type, and so on.

Archived

Figure 17-4 shows a graphic of the structure of an EDI message.

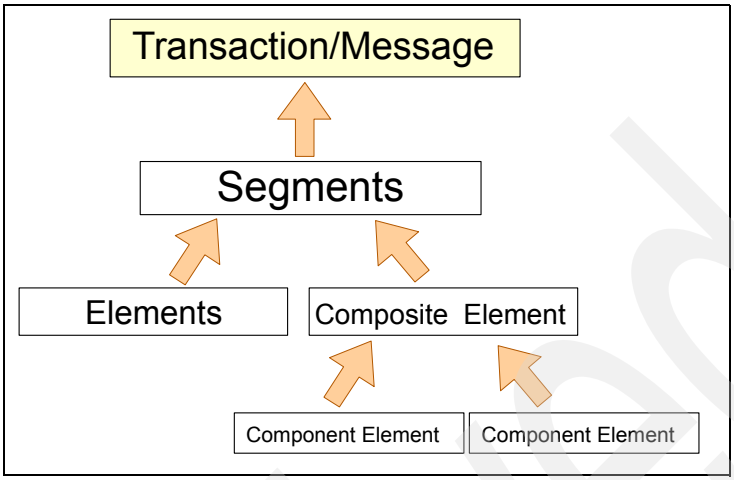


Figure 17-4 Components of an EDI message

Example 17-1 shows a sample X12 transaction. The first three lines are part of the envelope. The line starting with ST*810 is the start of the actual message. This time it is an 810 message, which is used to send invoices.

Example 17-1 X12 transaction (invoice)

```
ISA*00*          *00*          *ZZ*CELORGC02      *ZZ*IBMIRLPROD     *021018
*0229*U*00401*000008899*0*P*~¬
GS*IN*CELITALY*IBMIRELAND*20021018*0229*8899*X*004010¬
ST*810*88990001¬
BIG*20021017*0002146553**P350342***DR¬
CUR*SE*USD¬
REF*D2*0080118614¬
N1*SE*Celestica Italia S.r.l.*92*103015¬
N2*Celestica Italia S.r.l.¬
N3*Via Lecco 61¬N4* Vimercate - MI - IT**20059¬
REF*GT*IT03029690967¬
N1*BY*IBM INTERNATIONAL HOLDINGS¬
N2*IBM INTERNATIONAL HOLDINGS¬
N3*MULHUDDARTH¬
N4*DUBLIN*DB*15 ¬
REF*GT*IE6602632V¬
IT1*000001*4*EA*1767.87*PE*BP*00004N3524*VP*4N3524¬
TXI*VA*0*0¬
PID*F****BK C_F_CARDINAL¬
REF*ZZ*7071.48¬TDS*707148*707148¬
TXI*VA*0¬
CTT*1¬
```

```
SE*22*88990001¬
GE*1*8899¬
IEA*1*000008899¬
```

Both the X12 and the EDIFACT transactions in Example 17-1 on page 582 and Example 17-2 are presented with one segment per row to be easier to view. Normally, a new segment follows directly after the previous segment to save space.

Example 17-2 EDIFACT message (Purchase Order)

```
UNB+UNOA:2+3568579005454:14+3015437860102:14+021003:0053+02018852760++ORDERS'
UNH+1+ORDERS:D:93A:UN:EAN007'
BGM+220::9+001779'
DTM+137:20021002:102'
DTM+2:20021005:102'
DTM+63:20021005:102'
NAD+BY+3568579005454::9'
NAD+SU+3015166100102::9'
NAD+DP+3568579005454::9'
CUX+2:EUR:9+3:EUR:4'
LIN+1++3560998032054:EN::9'
QTY+21:2'QTY+59:1'
PRI+AAA:798.33::NTP'
LIN+2++3560998032054:EN::9'
QTY+21:5'QTY+59:1'
PRI+AAA:34.6::NTP'
UNS+S'
UNT+17+1'
UNZ+1+02018852760'
```

From a message organization point of view, both look similar. Every segment starts with a three-letter word indicating the type of segment that follows. Each element within the segment is separated from the next one by a separator. Finally, the message structure uses a segment terminator. There are more rules than this. Some elements are optional, while other elements are conditional. Element A and B are labeled conditional when, for example, the appearance of element A implies the appearance of element B.

17.3.2 Communication

Transportation of the EDI file over a network can be done in many ways. You can use any network and any protocol as long as it fits the need. This section looks at three types of communication:

- VAN communication
- Internet (AS1, AS2, FTP...)

Archived

- WebSphere MQ

This section focuses more on the communication aspect between two trading partners. There is also a communication aspect within the IT setup of a trading partner. The data has to be sent from the internal applications to the EDI translator software. After translation, the data must be handed over to some communication software.

VAN communication

Using a VAN for the transmission of files is traditionally seen as the most secure way of communication (Figure 17-5). Apart from doing pure communication, a VAN also provides value adds such as:

- Built-in security features that help protect against unauthorized access to customer data
- Restart and recovery facilities that help to reduce or eliminate the impact of communications interruptions
- Archive capability for the online retention of data copies
- 24x7 availability
- Notification of message arrivals that meet predefined criteria, such as a message from a specific trading partner

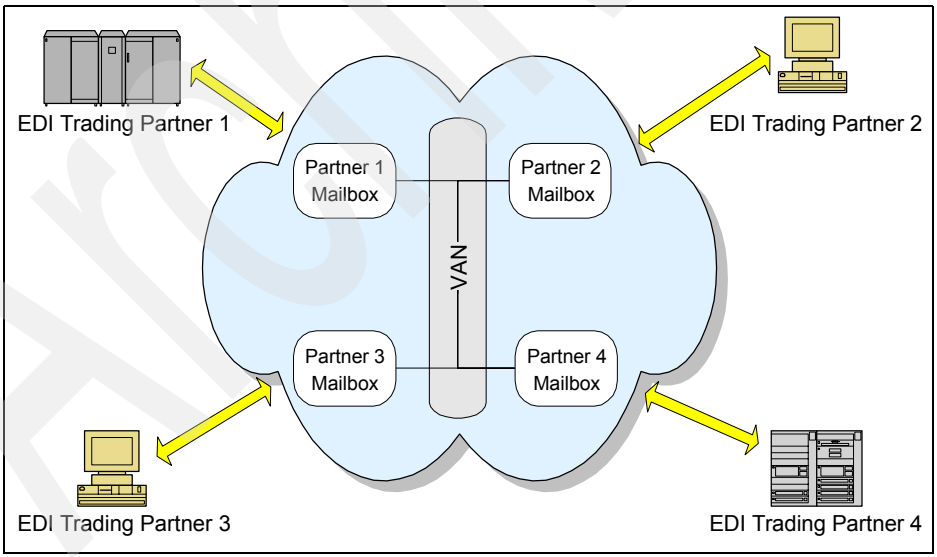


Figure 17-5 EDI VAN network

The VAN Gateway software drops off and picks up EDI documents in their mailbox. The VAN provides store-and-forward mailbox services. The physical

communication system between the VAN Gateway and the VAN network can vary from dial-up to FTP, or some proprietary communication technology.

IBM Information Exchange (IIN) is an example of such a value-added network.

EDI over the Internet

The initiative to move toward securely transmitted EDI messages over the Internet is known as EDIINT. Currently there are two main EDI INT initiatives, known as applicability statements AS1 and AS2. They describe how current Internet standards can be used to achieve VAN functionality.

- AS1 uses MIME and SMTP.
- AS2 uses MIME and HTTP for process-to-process real-time EDI.

The Internet solutions are often considered much cheaper than traditional VANs. However, Internet solutions often leave it to the user to add functionality to achieve adequate security, reliability, and other features that are included in a VAN.

IBM Business Exchange Services - Internet transfer is an example of Internet communication.

Message queuing

Message queueing (MQ) connects commercial systems in today's business. It provides assured, once-only delivery of data in any format.

IBM WebSphere MQ is an example.

17.4 The evolution of EDI

In today's economy, market dynamics have converged to form a business model that provides for the integration of different trading partners in a value chain. Depending heavily on Internet technologies, this model can enable highly coordinated trading communities, each with the ability to operate as a virtual enterprise.

In the virtual marketplace, business relationships are formed electronically. Buyers and sellers come together without the benefit of paper contracts, fee schedules, or sales people to close the deal. This Web economy requires an agile enterprise, one that can work more directly with suppliers and customers, as well as respond more rapidly and intelligently to change. The need for flexibility and lower costs, such as VAN charges, are driving the evolution of EDI.

Archived

Organizations are recognizing the value of many years of investments in EDI. Rather than replace the present solution, they plan to extend and evolve the EDI transactions. This existing EDI solution is considered as a part of a multi-modal B2B gateway or hub alongside XML, Web solutions, and portals. By integrating B2B and EDI technologies, event-driven or process-driven integration models can be supported using the existing EDI solution (Figure 17-6).

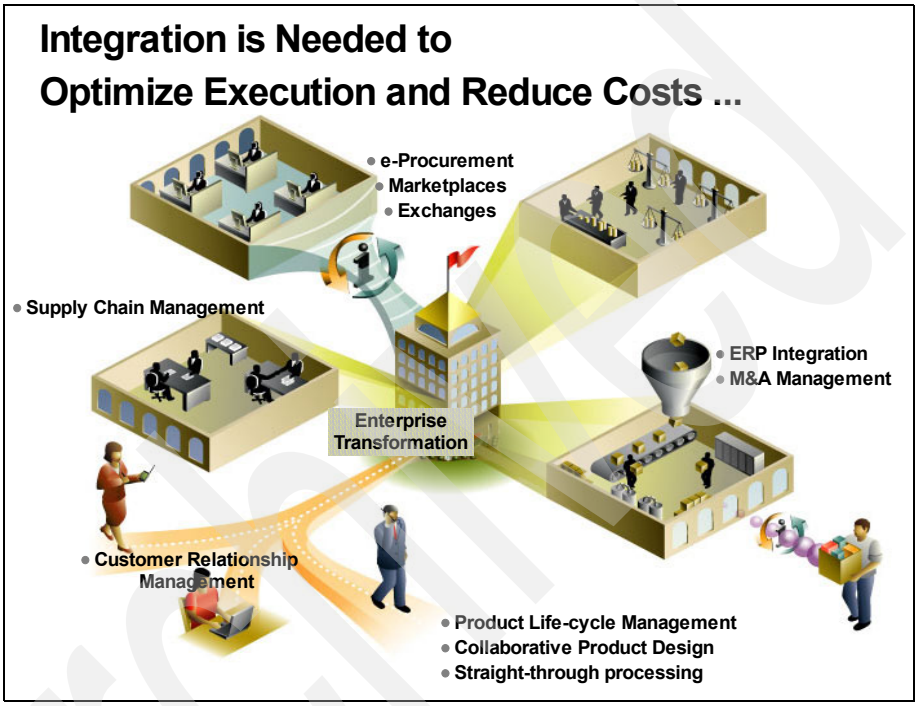


Figure 17-6 Business initiatives = business integration

The Internet is widely perceived as being much less expensive than a VAN, but this is not necessarily the case. VANs generally provide valuable services, such as Trading Partner Agreement (TPA) management, service-level administration, security, and store-and-forward capability. The Internet requires you to manage these elements yourself, which means the total costs are not always lower than a VAN.

EDI users cannot realize the full value of the Internet in e-commerce applications until the entire underlying business process is optimized. BPM is the automation, optimization, and management of end-to-end business process flows. And, in this case, it is accomplished by integrating front-end Web applications to back-end existing applications and to existing EDI trading partners.

Earlier phases of EDI achieved efficiency by automating manual processes. Now however, the focus is on business process integration and optimizing business operations. EDI steps are tied into the full value-chain processes with the ability to share information throughout.

The result of these trends is that traditional EDI customers are facing increasing challenges to remain competitive. To grow or even preserve your business, you need to integrate your existing EDI applications with core business processes by distributing transactions or information to and from various back-end applications. Situations in which this kind of integration can help are:

- Where you have typically spent tens or hundreds of thousands of dollars on your current EDI solutions and you want to leverage that investment.
- When internal departments lack timely information about EDI transactions and make costly mistakes or give poor service.
- Where competitiveness suffers from an inability to track or manage the distribution of EDI messages within the business.
- When manual processing of EDI messages is slow, error-prone, and consumes valuable resources.

17.4.1 Elements of an EDI solution

In addition to obvious components of an EDI solution, such as application programs and systems, VANs, and trading partners, a complete and flexible solution should include the following important elements.

Translators

A universal problem in the integration of applications is the conversion of shared data from one format to another. Common data fields, such as names, addresses, and numbers, often have different formats across disparate systems. The traditional approach to EDI implementation is to place the function that converts application data to the EDI standard directly into the business application. This approach is less effective because a separate program is required for each transaction as well as for each trading partner. In addition, it is difficult to keep up with new versions of standards because programs must be modified every time a trading partner adopts a newer standard or version of the standard.

This approach has changed with the introduction of third-party translation software, also known as *mappers*. The translator is responsible for mapping application data to the specific EDI format and vice versa. This translation software is implemented in either a centralized engine or in an adapter. It must handle primary EDI standards as well as different and evolving versions of each standard.

Batch enveloper/de-enveloper

Typically, because VAN charging is based on each sent transaction, enterprises have been driven to find ways to reduce the number of transactions and to compress more information into each. Consequently, EDI messages are sent in large batches, which can then be grouped from, or split out to, several divisions or areas of an enterprise.

Enveloping batch messages involves placing the EDI standard header and trailer around transactions in preparation for sending. When the envelope is complete, the package can then be sent to a trading partner through a VAN. Similarly, batch transactions must be deenveloped when they are received from the VAN.

Message router

When the EDI message is deenveloped, it can be divided into function groups. Each function group can relate to a different division or area of the business. A mechanism is needed to sort messages destined for different groups and deliver them to the appropriate target applications. This means there is a requirement to fan in and fan out messages. Message transformation might also be required to convert the message into the correct format for the end applications.

Trading Partner Agreements (TPA)

A *TPA* is an agreement related to the exchange of information in electronic transactions. The term includes a particular agreed-upon standard for business documents as well as communications and business protocols, the service-level agreement, and more. TPAs can also be extended to include business events. For example, if an event occurs in one organization that may affect processes in a second organization, the TPA can specify that the second organization be alerted to the event.

Archived

18

Mapping for EDI transformation and validation

In the previous part of the book we used the Data Interchange Services to map for non-EDI formats. In this chapter we continue our scenario for mapping to EDI standards for document exchange.

© Copyright IBM Corp. 2005. All rights reserved.

18.1 XML_EDI_ORDER

Before creating our new XML to EDI map, we must first define both the source and target document definitions in the Data Interchange Services client. The XML schema used to define the source document definition has already been imported in Chapter 15, “Mapping” on page 473. We will use the PURCHASE_ORDER schema located in the CUSTOM_XML dictionary again in this scenario. The EDI order used as the target document definition has yet to be created in the Data Interchange Services client. Luckily, the creation of this document type is not one that we complete ourselves. A large variety of EDI standards and subsets of those standards are already defined for the Data Interchange Services client and available for download from the WebSphere Partner Gateway Web site.

The EDI standard used in the data transformation map that we will create is version 4 revision 1 of the ANSI X12 purchase order transaction, known as the 850. To define this 850 transaction, download the entire X12 V4R1 standard from the WebSphere Partner Gateway Web site. The standard is packaged in a zip format. In order to import into the Data Interchange Services client we extract the x12v4r1-utf8.eif file from the compressed package. When we have the required *.eif file, we can import the required transaction.

To import an EDI transaction into the Data Interchange Services client:

1. Select **File Open Import File...** from the Data Interchange Services client. The Select Import File window opens.
2. Navigate to the directory in which the **x12v4r1-utf8.eif** file is stored. Select the file and click **Open**. The Import File Window opens as shown in Figure 18-1.

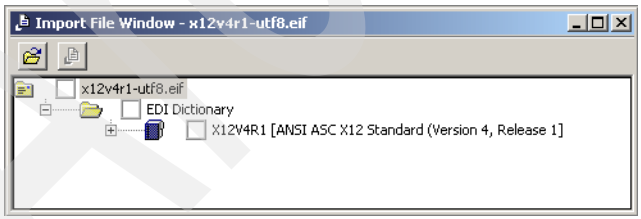


Figure 18-1 Import file window

3. Expand the X12V4R1 tree and scroll down until the 850 [Purchase Order] is visible in the window.
4. Select the **850** as shown in Figure 18-2 on page 591.

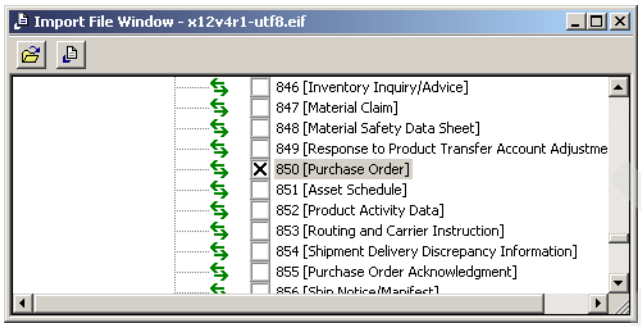


Figure 18-2 Select the 850

5. Click **Import Selected Documents**. The Select a Database window opens. Because we will create our data transformation map in the Data Interchange Services client development database, select **Development**, as shown in Figure 18-3. Click **OK**.

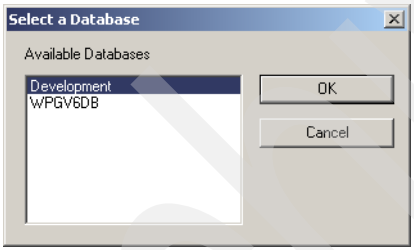


Figure 18-3 Select the Development database

6. The Execution Status Window opens and displays details of the import. When the window displays `Import completed successfully`, click **Close** (Figure 18-4).

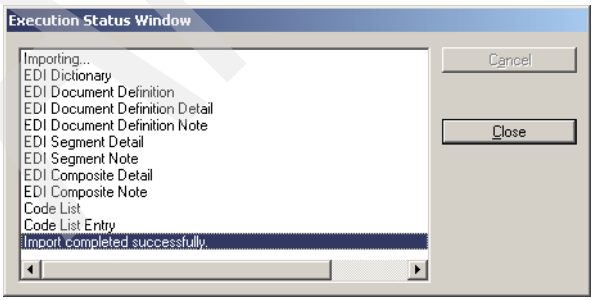


Figure 18-4 Execution status

Archived

Now that both our source XML schema and target EDI transactions are available in the Data Interchange Services client, we can go ahead and create a new data transformation map.

Open the Mapping functional area and follow these steps:

1. Select the Data Transformation Maps tab. Create a new map by selecting **File** → **New** from the Data Interchange Services client menu bar.

   The Create a Data Transformation Map window opens.

2. Create a new data transformation map using the following details:
   - Map Name: `XML_EDI_PO`
   - **Target-based map**
   - Source syntax type: **XML**
   - Source dictionary: **CUSTOM_XML**
   - Source XML document definition: **PURCHASE_ORDER**
   - Target Syntax Type: **EDI**
   - Dictionary name: **X12V4R1**
   - Target EDI document definition: **850**

3. Complete the wizard after reviewing the selections made by clicking **Finish**. The map loads and the Data Transformation Map Editor opens on the **Details** tab page.

18.2 Mapping commands

The mapping commands used to build our target EDI transaction are relatively simple and have already been seen in Chapter 15, “Mapping” on page 473 and Chapter 16, “Advanced mapping” on page 541. We can start by mapping the mandatory data elements in the BEG segment of the 850 transaction.

18.2.1 BEG Segment

Element 353 at position 1 of the BEG segment is controlled by an EDI code list. This code list restricts the set of valid values that can be entered in the data element. To determine which code list is used on this data element we should open the EDI functional area.

> **Note:** There is no need to close the Data Transformation Map Editor which should be open at this point. Simply open the EDI functional area which will in turn open over the current view.

To determine the code list for a given data element, follow these steps:

1. Open the Data Elements tab from the EDI functional area.
2. Look through the list of data elements displayed until you find element 353.

> **Tip:** The data elements view is sorted first by Dictionary Name then by Data Element name.

3. When you find element 353, double-click the entry to display the EDI Data Element Editor. The editor opens on the **General** tab.

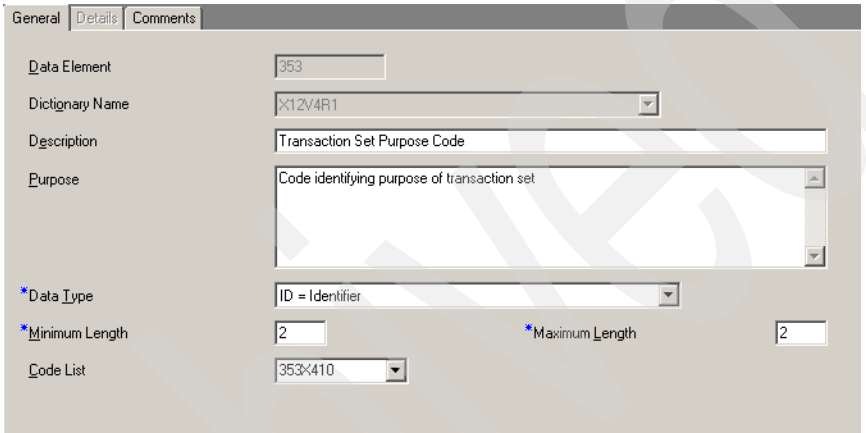


Figure 18-5 EDI data element editor

4. The code list responsible for maintaining a list of valid values for this data element is shown in Figure 18-5. We can see the code list name is 353X410, so our next step is to explore the list of values defined in the list.
5. Close the EDI data element editor to return to the EDI functional area.
6. Select the **Code Lists** tab and search for the entry with a **Code List Name** 353X410.
7. Double-click the entry for **353X410** to display the Code List Editor. A table is displayed with a list of valid values for data elements controlled by this code list. This table is shown in Figure 18-6 on page 594.

Archived

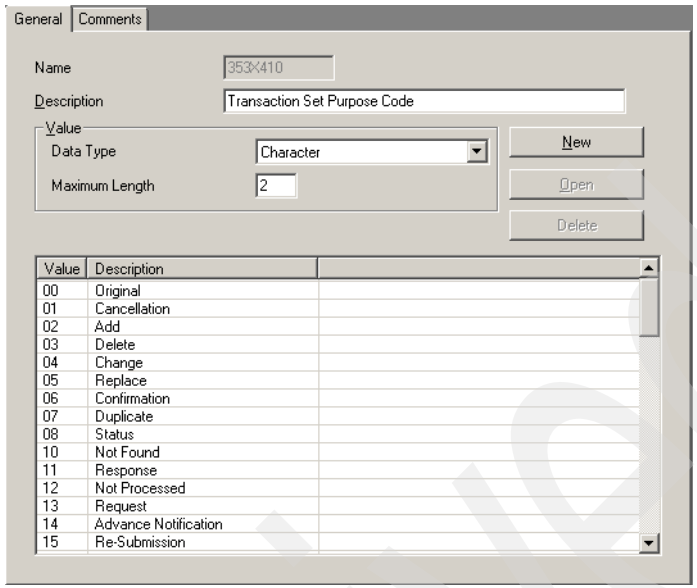


Figure 18-6 Code list editor for 353X410

8. Using the table displayed in Figure 18-6 we can determine a valid value to use in our mapping command for element 353 position 1 of the BEG segment. Because this is an original order, we will use code 00 as defined in the code list.
9. Close both the code list editor and the EDI functional area to return to the data transformation map editor. In the mapping command window, assign a value of 00 to the first element of the BEG segment as shown in Figure 18-7.

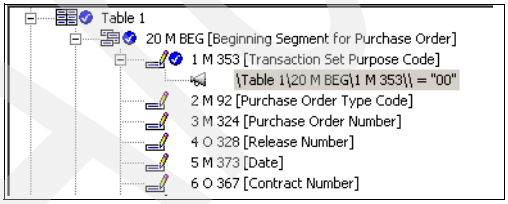


Figure 18-7 Assign value to segment

We can map the data element at position 2 of the BEG segment in much the same way. We first examine the code list responsible for maintaining the list of valid values that can be used in this data element. When we know these, we can assign an appropriate value using an assignment mapping command.

10. Open the EDI functional area on the Data Elements tab. Search for element 92 and determine the code list used by this data element. This code list is reference as 92X410.

11. Next, open the **Code Lists** tab from the EDI functional area and select a valid value to use in our XML to EDI data transformation map. In our example, we will used the value **SA** that defined our EDI transaction as a stand alone order.

12. After assigning a value of SA to data element 92, the mapping command window should appear as shown in Figure 18-8.

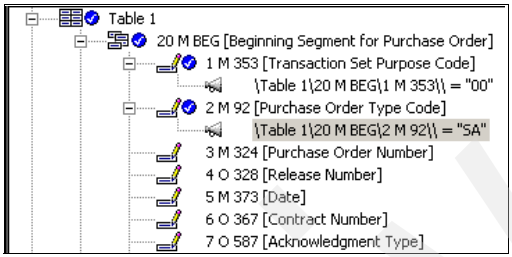


Figure 18-8 Assign value to data element

13. Data element 324 is the first element in the target document definition that is mapped from an XML element in the source document. Drag the orderNumber element from the source document definition and drop it on element 324 at position 3 of the BEG segment in the target document definition.

14. We complete the mapping of the BEG segment using one of the predefined functions available in the Data Interchange Services client. Map element 373 at position 5 of the BEG segment using the Date() function (Figure 18-9).

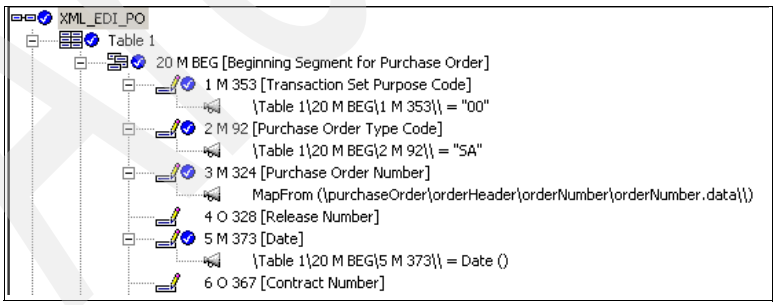


Figure 18-9 BEG segment mapping commands

Archived

18.2.2 N1 Loop

The next segment in which we define mapping commands for our target EDI transaction is the N1 segment at position 310 of Table 1.

Element 98 at position 1 of the N1 element is again validated by a code list. Looking at the EDI data element editor from the EDI functional area, the code list responsible for this data element is 98X410. We can therefore map element 98 using the value ST from 98X410 - this ST value flags the details in this segment as ship to details:

1. Use the assignment command in the mapping commands window to apply this value as shown in Figure 18-10.

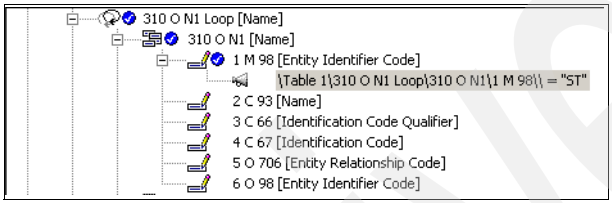


Figure 18-10 Using the assignment command

2. Map element 93 at position 2 of the N1 segment from the buyerName data element in the source document definition.
3. Assign a value of 12 to element 66 at position 3 of the N1 segment. The code list responsible for this data element, code list 66X410, defines 12 as a value flagging the next data element as a phone number.
4. We therefore enter a phone number in data element 67 at position 4 of the N1 segment. Assign a fictitious number of 555 555 1234 to this data element. This completes the mapping of the N1 segment at position 420. When complete the mapping command window should appear as shown in Figure 18-11.

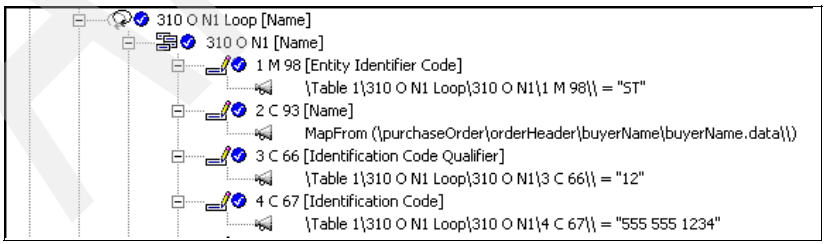


Figure 18-11 N1 mapping commands

5. In the N2, N3 and N4 segments at positions 320, 330 and 340 respectively we map only the mandatory data elements contained in each. Map each of these data elements as follows:
   - Map element 93 at position 1 of the N2 segment from the deliveryAddressName element in our source document definition.
   - Map element 166 at position 1 of the N3 segment from the deliveryAddress1 element in our source document definition.
   - Map element 19 at position 1 of the N4 segment from the deliveryAddress2 element in our source document definition.
6. When each of these mapping commands have been implemented, the mapping command window should appear as shown in Figure 18-12.

Figure 18-12 Mapping commands

18.2.3 PO1 Loop

Our final task is to map the PO1 Loop with item data from our source document. In doing so, we map data elements from both the PO1 segment and PID segment:

1. Expand Table 2 in the mapping commands window.
2. Next expand the PO1 Loop at position 10 and finally expand the PO1 segment itself (Figure 18-13 on page 598).

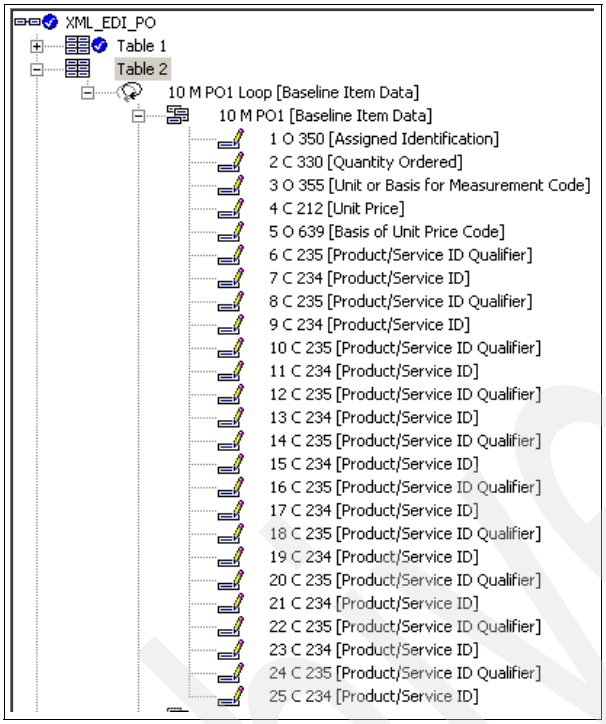


Figure 18-13 PO1 segment

Our first task is to create a new **For Each** command so that a new PO1 Loop is generated for each lineDetail element received in the source document definition.

To create this **For Each** command, follow these steps:

1. Drag the lineDetail element and drop it over the PO1 Loop at position 10 in table 2. When this action is complete the mapping command window should appear as shown in Figure 18-14.

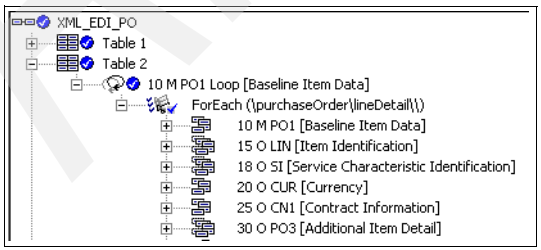


Figure 18-14 Completed mapping commands

2. Map element 330 at position 2 of the PO1 segment from the quantity element of the source document definition.
3. Map element 355 at position 3 of the PO1 segment from the uom element of the source document definition.
4. Map element 212 at position 4 of the PO1 segment from the unitPrice element of the source document definition.
5. Assign a value of BP to element 235 at position 6 of the PO1 segment. This BP value flags the value in the next data element as being the buyer's part number.
6. Map element 234 at position 7 of the PO1 segment from the partNumber element of the source document definition.
7. When complete the mapping command window should appear as shown in Figure 18-15.

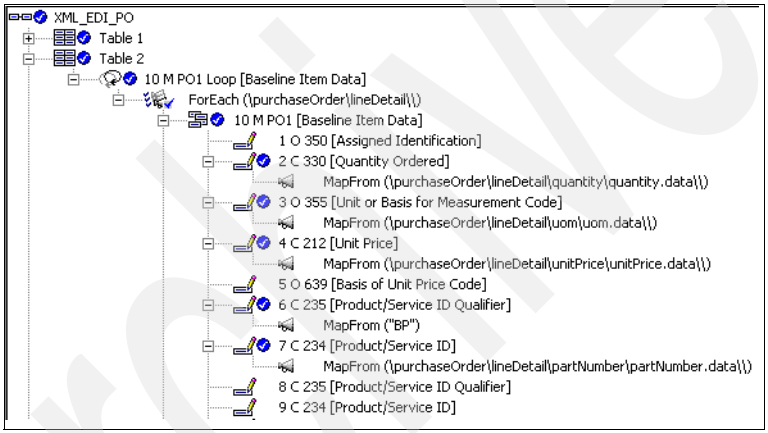


Figure 18-15 PO1 mapping commands

The final step in completing our XML to EDI map is to create mapping commands for the PID segment contained within the PO1 Loop. The PID segment itself is contained within a PID Loop at position 50 of table 2.

8. Expand the PID Loop in the mapping command window, then expand the PID segment itself. The mapping command window should appear as shown in Figure 18-16 on page 600.

Archived

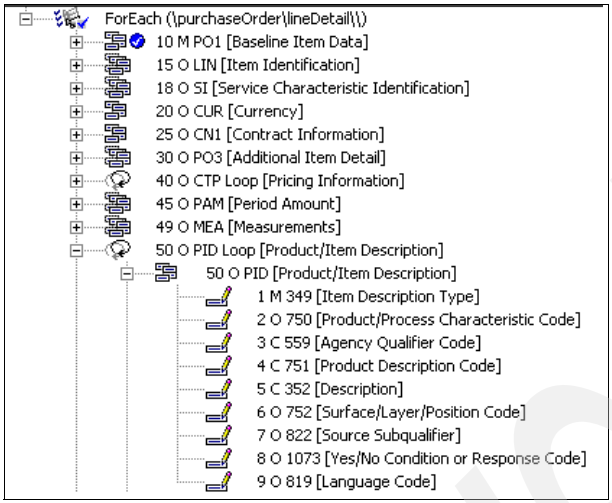


Figure 18-16 Mapping command window for PID segment

9. Assign a value of F to element 349 at position 1 of the PID segment. This F value defines the product or item description as defined in the PID segment as being freeform. This value is defined in code list 349X410.
10. Assign a value of AB to element 559 at position 3 of the PID segment. This AB value defines the product description code as being assigned by the buyer.
11. Map element 352 at position 5 of the PID segment from the itemDesc element from the source document definition.

Archived

18.3 EDI_XML_PO-ACK

As with the XML to EDI data transformation map created earlier, before creating our EDI to XML map we must first define the document types involved in the map to the Data Interchange Services client. This data transformation map will involve two documents representing a Purchase Order Acknowledgement transaction. The source document will be an ANSI X12 855 transaction and the target document will be an XML representation of a purchase order acknowledgement as defined by a Document Type Definition (DTD).

To create the 855 definition in the Data Interchange Services client we must use the Standard export / import file (.eif) file which we downloaded earlier.

To import the 855 transaction, follow these steps:

1. Select **File** → **Open Import File...** from the Data Interchange Services menu bar. The Select Import File window opens.
2. Navigate to the directory in which the **x12v4r1-utf8.eif** is stored, select the file and click **Open**. The Import File Window opens.
3. Expand **X12V4R1** then scroll down and select **855 [Purchase Order Acknowledgement]**.Click **Import Selected Documents**. The Select a Database window opens.
4. Highlight the Development database entry and click **OK**. The Execution Status Window opens. Click **Close** when informed that the import has completed successfully.

Now that we have imported the required EDI definition we can import our XML DTD that defines the XML purchase order acknowledgement into the Data Interchange Services client. The content of our XML DTD can be seen in Example 18-1.

Example 18-1 Order Acknowledgment DTD

```
<!ELEMENT orderAcknowledgement (ackHeader, ackBody*, ackFooter)>

<!ELEMENT ackHeader
      (customerNumber,
       orderNumber,
       date,
       supplierNumber)>

<!ELEMENT ackBody
      (lineNumber,
       quantityRequested,
       quantityCommited,
       price,
```

```
        description,
        partNumber)>

<!ELEMENT ackFooter
       (totalQuantity,
        totalPrice)>
```

To import our XML DTD into the Data Interchange Services Client:

1. Select **File** → **Open Import File...** from the Data Interchange Services client menu bar. The Select Import File window opens.
2. Change the Files of Type entry to XML DTD File (*.dtd) and navigate to the directory in which the XML DTD is stored.
3. Select **XML DTD** and click **Open**. The Select a Database window opens. Select the entry for our Development database and click **OK**. The Import XML DTD window opens (Figure 18-17).

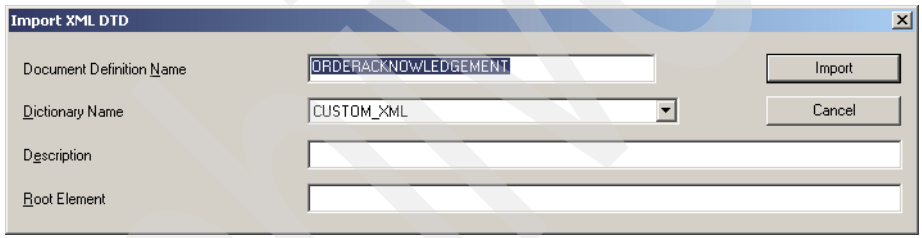


Figure 18-17 Select target DTD

4. The Document Definition Name defaults to the value of the DTD root tag, in this case that is ORDERACKNOWLEDGEMENT. Change the document definition name to `PURCHASE_ORDER_ACK`.
5. The dictionary name should default to `CUSTOM_XML`, if not set `CUSTOM_XML` as the dictionary name.
6. Enter `orderAcknowledgement` as the Root Element. The Import XML DTD window should appear as shown in Figure 18-18 on page 603. Click **Import**.

Archived

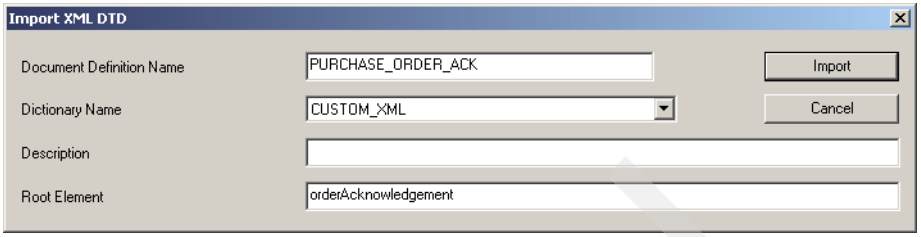


Figure 18-18 Select Root Element

7. The Import File window opens. When the import is complete, an information window opens. Click **OK**.
8. You can view the new XML DTD in the Data Interchange Services client by opening the XML functional area and selecting **DTD Document Definitions**.
9. Now that we have defined both the EDI 855 and XML DTD in the Data Interchange Services client, we can begin to create our new data transformation map.
10. To create a new data transformation map, open the Mapping functional area and select the **Data Transformation Maps** tab. Select **File** → **New** to create a new data transformation map. The Create a Data Transformation Map window opens.
11. Create the new data transformation map using the following details:
    - Map Name: `EDI_XML_PO-ACK`
    - **Target-based map**
    - Source syntax type: **EDI**
    - Source dictionary: **X12V4R1**
    - Source EDI document definition: **855**
    - Target syntax type: **XML**
    - Target dictionary: **CUSTOM_XML**
    - Target XML document definition: PURCHASE_ORDER_ACK
12. Confirm the values selected when creating the data transformation map and click **Finish**. The new map loads and opens on the Details tab of the Data Transformation Map Editor.

Our first step is to map the customerNumber and supplierNumber fields in the target document definition. We map these two elements first because the mapping commands used for each are very similar. The information we require to populate these fields will be retrieved from the ISA segment of the EDI X12 source message. The ISA segment is considered as the opening segment of an EDI X12 envelope and is a common segment to all EDI X12 transactions.

Archived

For this reason, the ISA segment does not appear as a segment available for mapping in the source document definition. In order to retrieve the detail from this segment we use the GetProperty() function of the Data Interchange Services client. The GetProperty() function allows us access to the information defined at the envelope level of an EDI X12 transaction.

In the Mapping Command window (Figure 18-19), perform these steps:

1. Expand the **ackHeader** tree of the XML DTD.
2. Expand the **customerNumber** element until the customerNumber.ANY element is visible.
3. Right-click the **customerNumber** element and choose **Insert Within** → **Command** → **MapFrom**... The Mapping Command Editor opens.

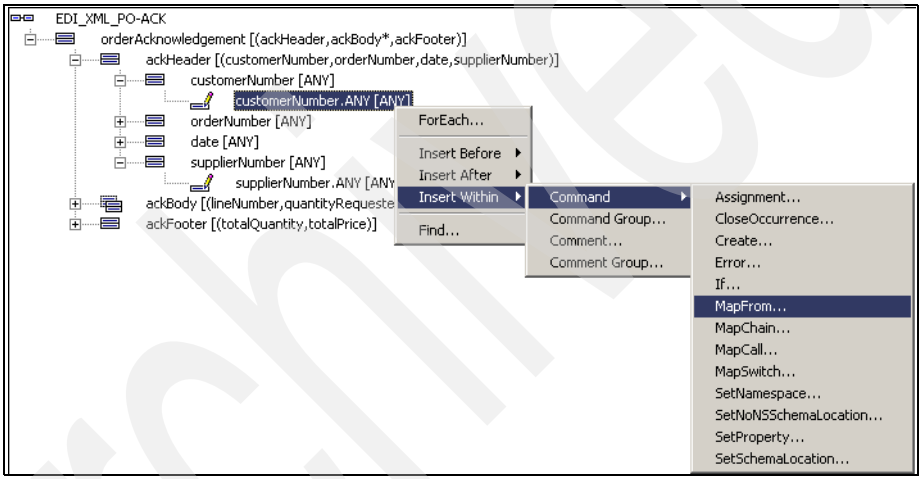


Figure 18-19 Mapping Editor

4. Highlight the expression value that is passed as a parameter to the MapFrom() function, right-click and select **Functions** → **GetProperty()** (Figure 18-20 on page 605).

Archived

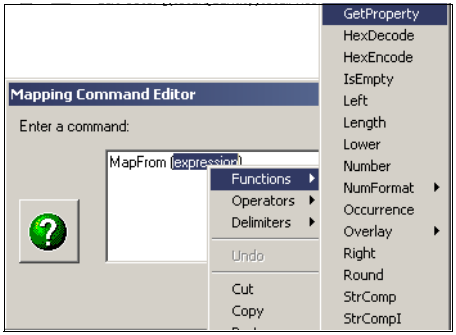


Figure 18-20 Using GetProperty function

5. Highlight the **propertyName** value, right-click, and select **Values** → **Generic Envelope Properties** → **"IchgRcvrId"**

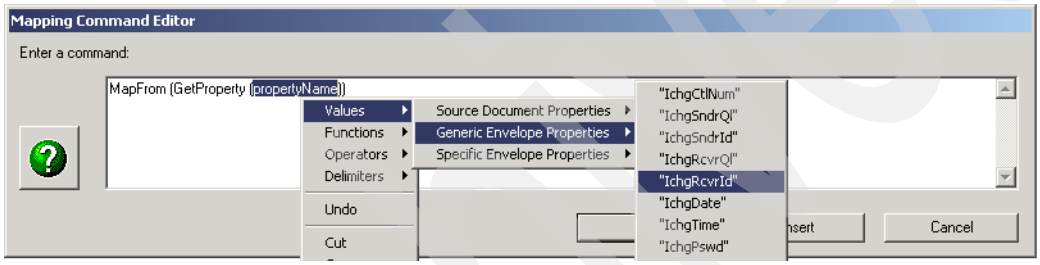


Figure 18-21 Mapping Command Editor

6. The Mapping Command Editor should appear as shown in Figure 18-22.

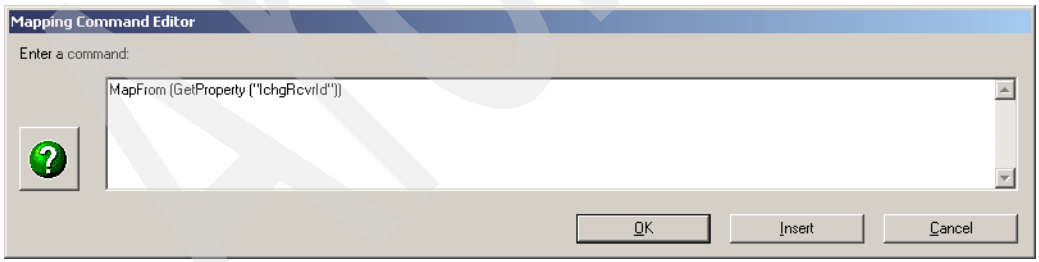


Figure 18-22 Mapping command

7. Click **OK** to complete the mapping command.

We map the supplierNumber element in much the same way as the customerNumber element. The only difference in the mapping command will be the parameter passed to the GetProperty() function. We map the

supplierNumber element from the interchange sender ID contained in the ISA envelope.

8. Simply select the **"IchngSndrId"** when entering a value for the GetProperty() parameter. The mapping command editor appears as shown in Figure 18-23.

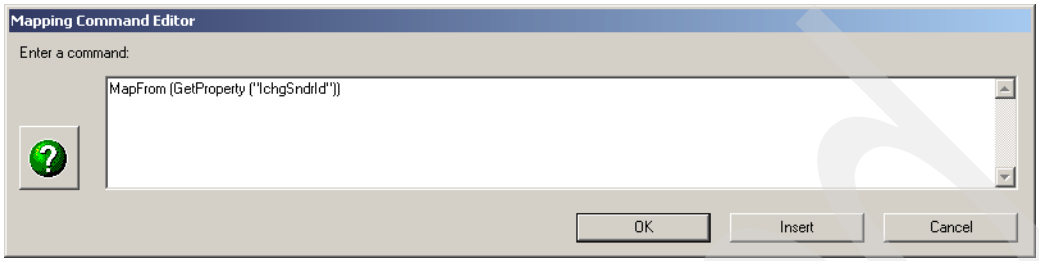


Figure 18-23 Select property

9. When the supplierNumber and customerNumber elements have been mapped, the mapping command window appears as shown in Figure 18-24.

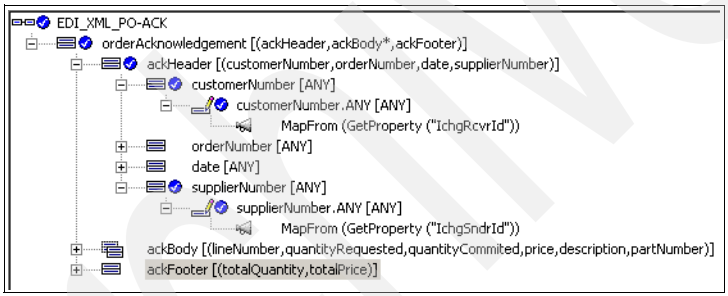


Figure 18-24 Completed mapping

10. Map the orderNumber element using the simple drag and drop technique. The source EDI data element used in the mapping command is element 324 at position 3 of the BAK segment.

11. Map the date element in the target document definition from the BAK segment. Drag data element 373 at position 4 of the BAK segment over the date element in the target document definition and drop it to create a MapFrom() command.

    The ackHeader element resembles Figure 18-25 on page 607.

Archived

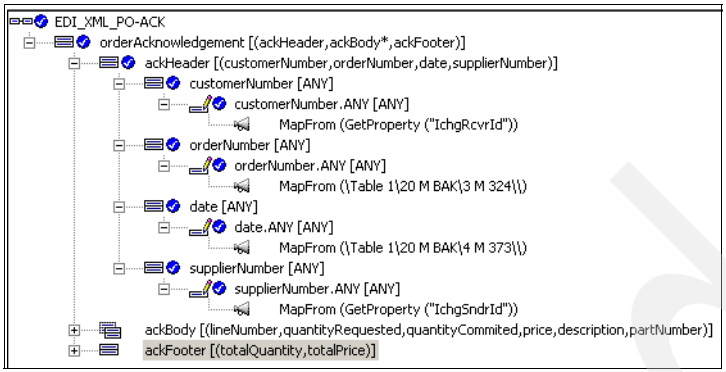


Figure 18-25 ackHeader element

Now that we have completed our mapping commands for the ackHeader element, we can progress to the mapping of the ackBody element.

The first mapping command that we include is a `For Each` command. We want to map the elements in the ackBody element for each occurrence of the PO1 segment in the source document definition.

To create this For Each mapping command, follow these steps:

1. Expand **Table 2** in the Source Document Definition window.
2. Select and drag the **PO1 Loop** at position 10 of Table 2 over the ackBody element in the mapping command window.
3. Drop the PO1 Loop to create a new `For Each` command. This will ensure that a new ackBody data element is created in the output of the data transformation map for each PO1 segment in the source document.

The lineNumber element in our target document definition has no corresponding data element in the source document. We can create line number data however, by using a local variable that will increment its value by one each time a new ackBody element is generated. To do this, we create a new local variable called lineCount.

To create this variable (Figure 18-26 on page 608),follow these steps:

1. Right-click anywhere within the Local Variable Name column of the Variables Window. The Create New Local Variable window opens.
2. Give the variable a name of `lineCount` and set the Data Type to **Integer**.
3. Accept all other default values and click **OK**.

Archived

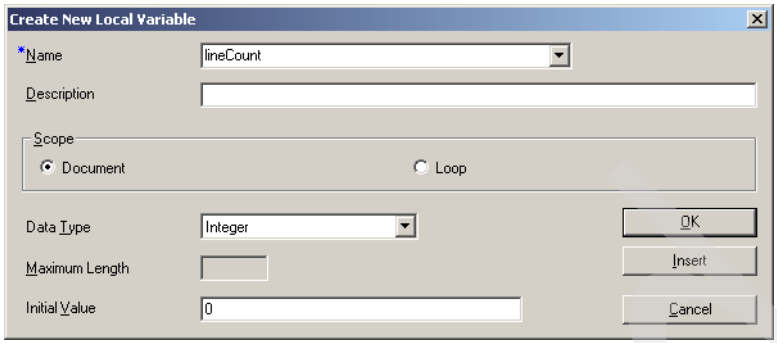


Figure 18-26 Create local variable

Next, we expand the **ackBody** element in the Mapping Commands window. We want to create the mapping commands that will both increment the lineCount variable and to map the lineCount value to our output data.

4. Expand and select **lineNumber**.
5. Right-click and choose **Insert Within Command Assignment**. The Mapping Command Editor window opens.
6. Drag the lineCount variable over the path value in the mapping command editor.
7. In the same way, drag the lineCount variable over the expression value and drop it there.
8. Finally enter `+ 1` at the end of the mapping command so that the mapping command editor appears as shown in Figure 18-27.
9. Click **OK** to complete the mapping command.

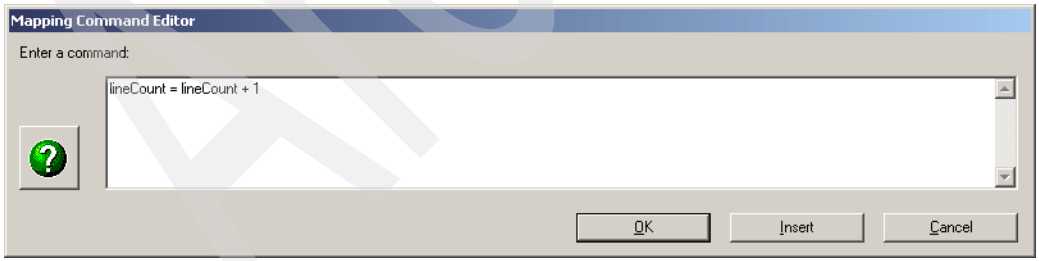


Figure 18-27 Mapping command to increment counter

Now that we have inserted a mapping command that will increment the lineCount variable by one each time, an ackBody element is generated in the output data, we want to map the value of this variable to the result of translation.

10. Simply right-click the mapping command created above in the Mapping Command window and select I**nsert After Command MapFrom**....

    The Mapping Command Editor opens.

11. Drag the **lineCount** variable from the Variables window onto the expression value in the mapping command editor and drop it.

12. Click **OK** to complete the mapping command.

13. The Mapping Command window appears as shown in Figure 18-28.

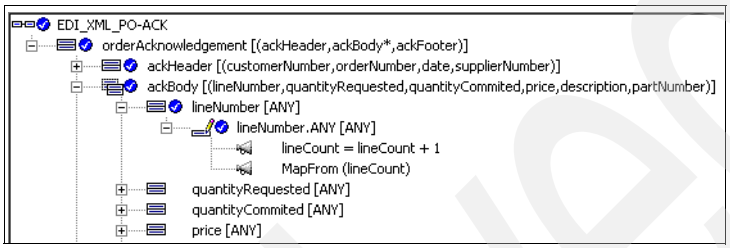


Figure 18-28 Completed mapping command

The remainder of the ackBody element can be mapped from source data elements in the source document definition.

14. Map the both the quantityCommitted and quantityRequested elements from data element 330 at position 2 of the PO1 segment.

15. Map the price element from data element 212 at position 4 of the PO1 segment.

16. Map the description element from data element 352 at position 5 of the PID segment. The PID segment itself is located at position 50 in table 2.

17. Map the partNumber element from data element 234 at position 7 of the PO1 segment.

18. The Mapping Command window resembles Figure 18-29 on page 610.

Archived

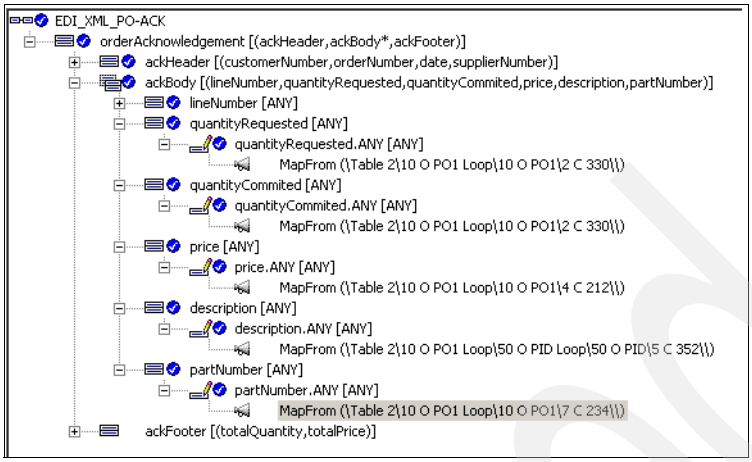


Figure 18-29 Completed mapping command

To map the totalQuantity element in the ackFooter element, we require a new local variable. There is no corresponding data element with information about total quantity in the source data document. To create this information, we use a local variable that will sum the quantity mapped to the quantityCommityted element.

Our first step is to create a new local variable:

1. In the Variables window, right-click anywhere in the area of the Local Variable Name column.
2. The Create New Local variable window opens (Figure 18-30).
3. Create a new local variable using the name `totalQuantity` with a data type of **Integer**.
4. Accept all other defaults and click **OK**.

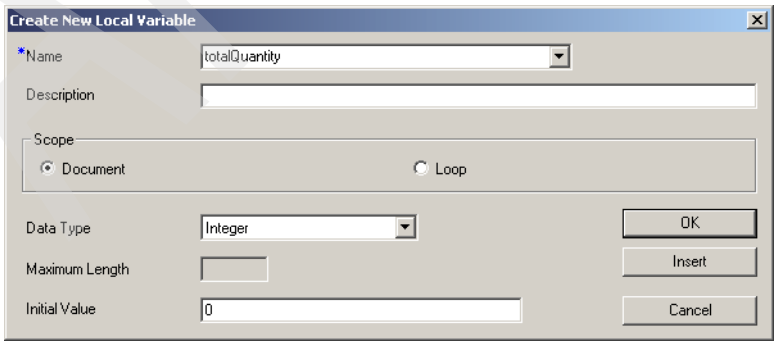


Figure 18-30 Create a local variable for quantity

5. Select the **MapFrom()** command located in the quantityCommitted element in the Mapping Commands window (Figure 18-31).
6. Right-click the command and select **Insert After** → **Command** → **Assignment**. The Mapping Command Editor window opens.

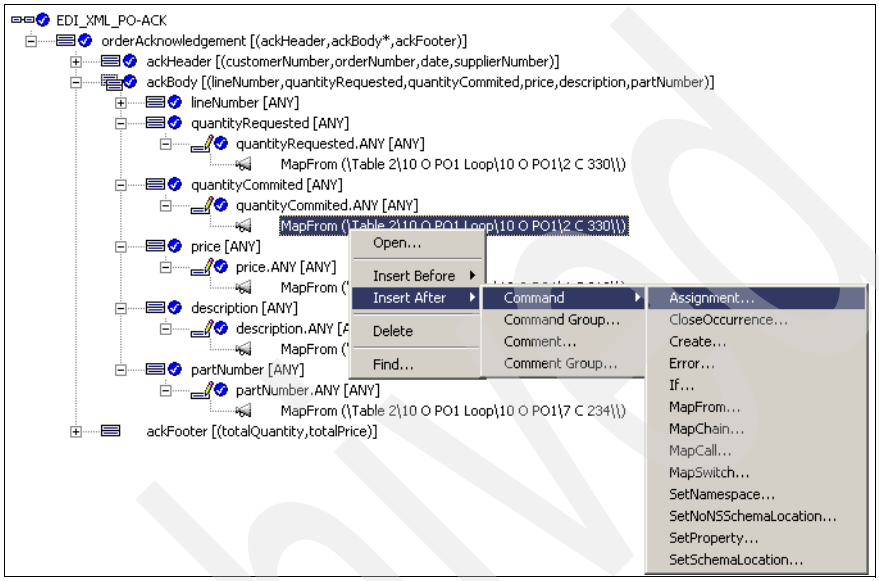


Figure 18-31 Assign a value

7. Drag the **totalQuantity** variable from the Variables window and drop it over the path value in the Mapping Command Editor.
8. Drag data element **330 at position 2** of the PO1 segment and drop it onto the expression value in the Mapping Command Editor.
9. Finally end the mapping command by appending `+ totalQuantity` to the end of the command. The Mapping Command Editor resembles Figure 18-32.
10. Click on the **OK** button to complete the mapping command.

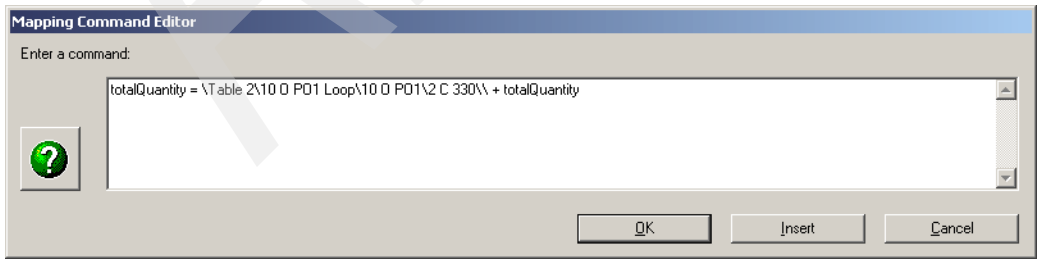


Figure 18-32 Mapping expression

Archived

11. The Mapping Command window should appear as shown in Figure 18-33.

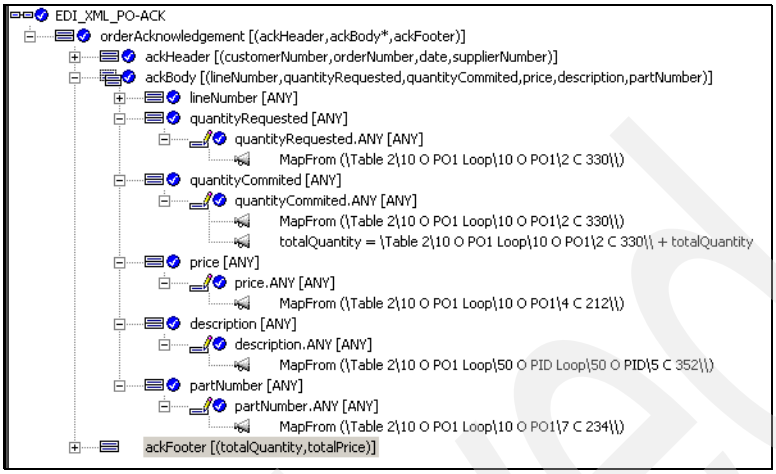


Figure 18-33 Completed command

12. To complete the EDI_XML_PO-ACK data transformation map we must map the ackFooter element. The only element we map here is the totalQuantity element. Simply create a **MapFrom** command within the totalQuantity element and drag the **totalQuantity** local variable into the Mapping Command Editor.

13. The MappingCommand window appears as shown in Figure 18-34.

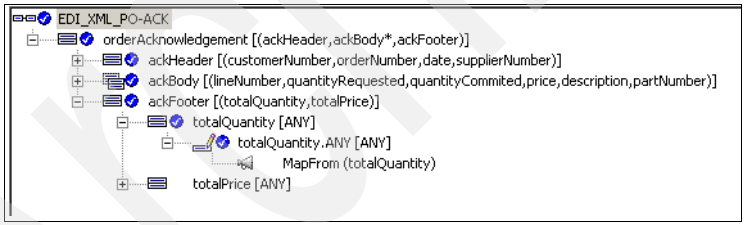


Figure 18-34 Completed map

14. This completes the mapping commands for our EDI to XML map. Save, close and compile the map.

18.4 353X410_VALIDATE

The concept behind code lists was discussed earlier in this chapter. In this scenario, Company E will only translate purchase order acknowledgements that are flagged with either an accept or reject tag. As you might expect, this accept

or reject tag is used in purchase order acknowledgements to inform customers whether an order (850) has been accepted or rejected.

This information is conveyed in the BAK segment. Element 353 at position 1 of the BAK segment is managed by code list 353X410 and is used as a transaction set purpose code. Although this code list defines numerous values, Company E will only process Purchase Order Acknowledgements whose BAK segment contains a 353 data element with a value of either 06 - Confirmation or 44 Reject.

To create this heightened level of validation, we must first create a new validation map in the Data Interchange Services client. To create a new validation map:

1. Open the mapping functional area from the Data Interchange Services client and select the **Validation Maps** tab. A set of default validation maps are displayed.
2. Create a new validation map by selecting **File** → **New** from the Data Interchange Services client menu bar. The Create a Validation Map window opens (Figure 18-35).
3. Provide a Map Name, for example `353X410_VALIDATE`. Click **Next**.

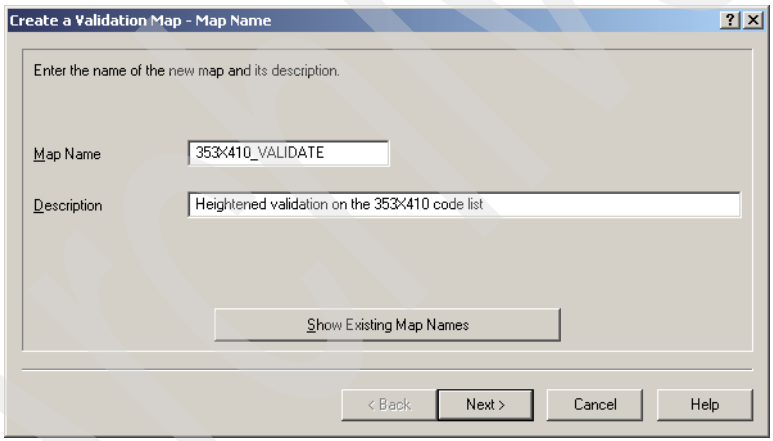


Figure 18-35 Create new map

4. In the Source Dictionary window (Figure 18-36 on page 614), select **X12V4R1** as the source dictionary for our new validation map. Click **Next**.

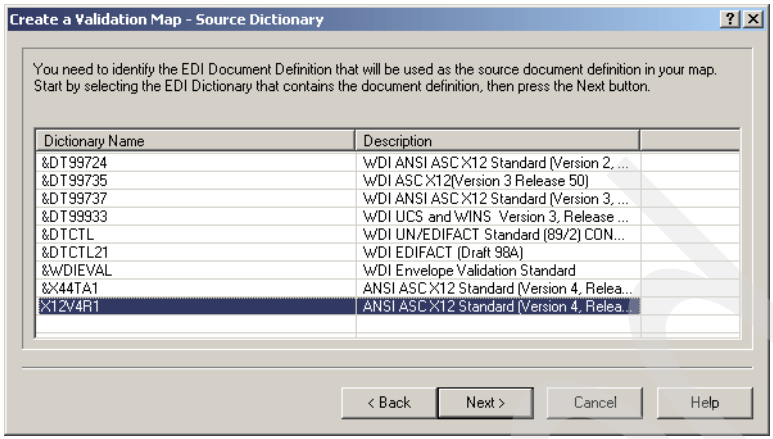


Figure 18-36 Source dictionary

5. Choose **855** (Figure 18-37) as the document definition name and click **Next**.

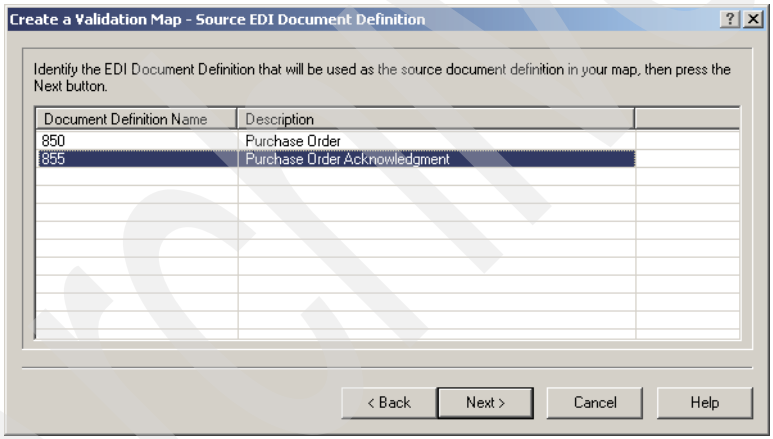


Figure 18-37 Select source document definition

6. Finally review details of the selections you made (Figure 18-38 on page 615)and click **Finish**.

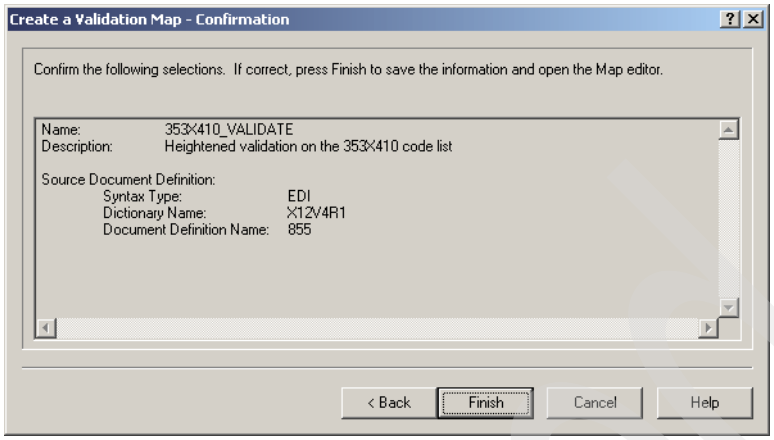


Figure 18-38 Verify selections

7. The map is loaded, and the Validation Map Editor opens on the Details tab (Figure 18-39).

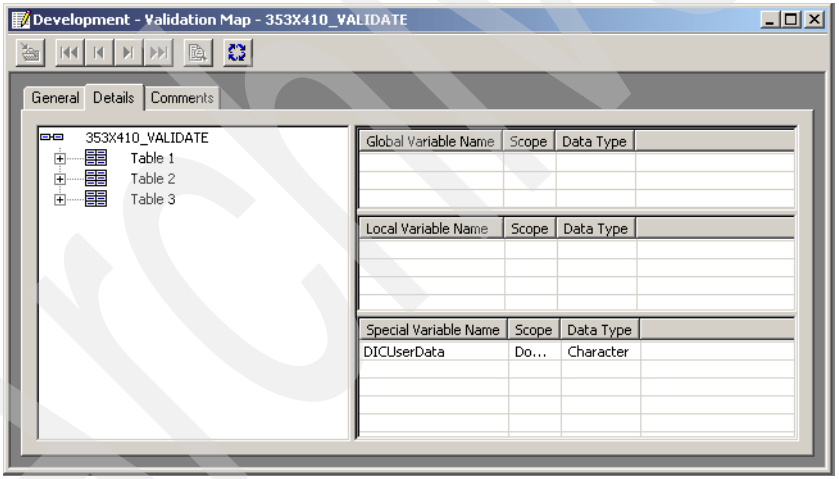


Figure 18-39 Validation map editor

When the validation map editor window opens we can create the mapping commands that will enforce the logic required by Company E. Remember, only a value of 44 or 06 will be received in data element 353 of the BAK segment. The commands used here are almost identical to those used when creating a regular data transformation map.

Archived

To create the required mapping commands, follow these steps (Figure 18-40):

1. Expand **Table 1** from the mapping command window.
2. Expand the **BAK segment** located at position 20 in Table 1.
3. Right-click **element 353 at position 1** of the BAK segment and chose Insert Within Command If...
4. The Mapping Command Editor opens in the same way as it would if this were a data transformation map.

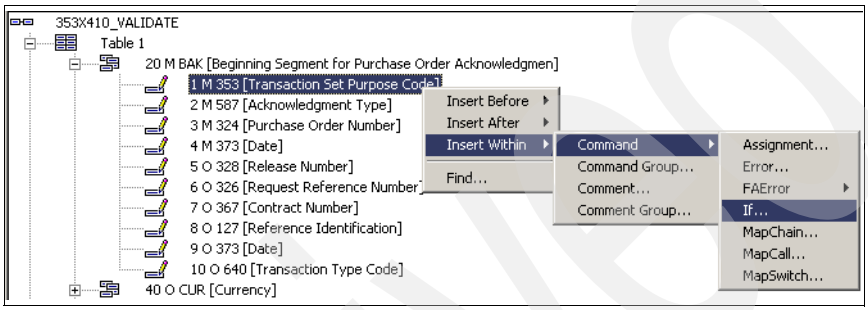


Figure 18-40 Add a new If condition

5. Select the expression value display between parentheses in the mapping command editor. Right-click and select **Functions** → **StrComp**. The string compare function used here works in precisely the same manner as it would in a data transformation map.
6. The mapping command editor displays a template for the string compare function as shown in Figure 18-41.

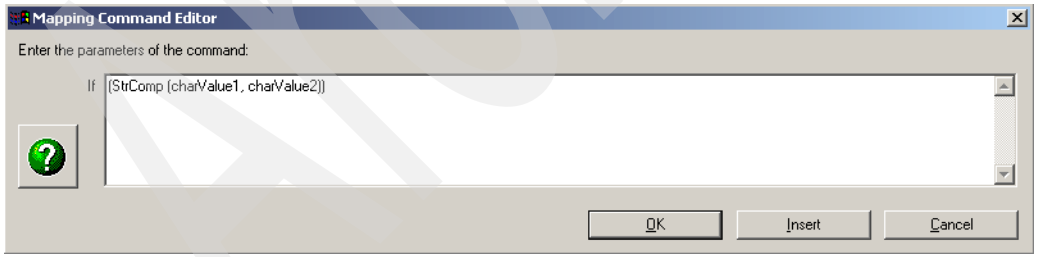


Figure 18-41 StrComp template

7. Replace the charValue1 string that is passed as a parameter to the StrComp() function with the value received in the source document BAK segment. To do this, drag **element 353** of the BAK segment from the mapping command window onto the charValue1 string in the mapping command window.

8. Replace charValue2 with the value with which we want to compare, 44 or 06. For this command replace charValue2 with `06`. Your mapping command should appear as shown in Figure 18-42.

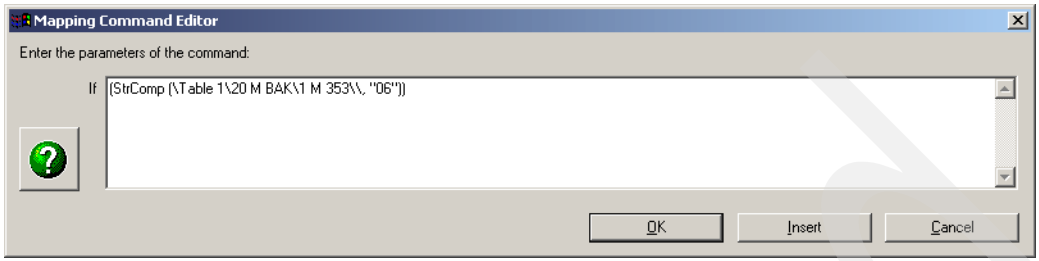


Figure 18-42 Enter compare value

We are using the StrComp function to test that the value received in element 353 of the BAK segment is not equal to either 06 or 44. To test that the value received is not equal to 06 for this mapping command we use the NE operator to complete the command.

9. Enter NE 0 between the last two parenthesis on the right hand side of the mapping command. Your mapping command should appear as shown in Figure 18-43.

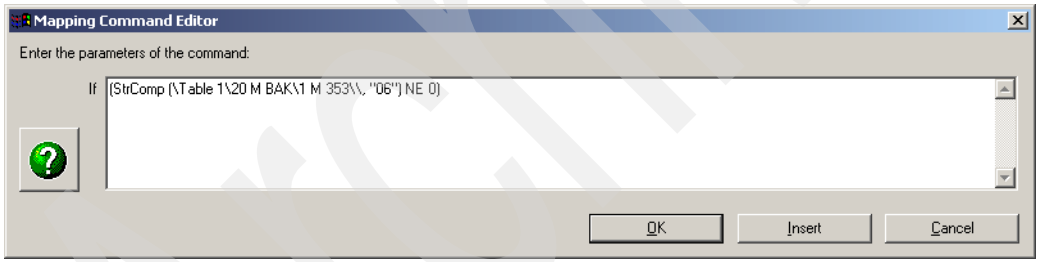


Figure 18-43 Enter condition

This command is sufficient to check that value received in element 353 is not equal to 06. We also want to ensure that any value received is not equal to 44.

10. To create this test, simply copy the existing mapping commands shown in the Mapping Command Editor and paste the commands after the last right parenthesis. Replace the 06 value in the newly pasted commands with `44`.

11. Your mapping command should appear as shown in Figure 18-44 on page 618.

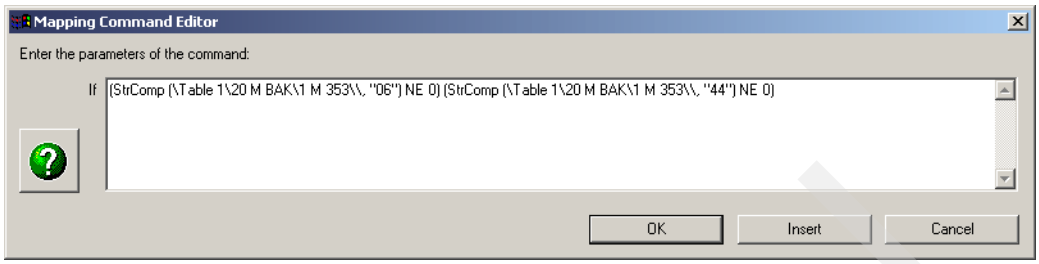


Figure 18-44 Completed mapping command

12. Finally we need to link these two mapping commands with a logical `and` operator. In the Mapping Command Editor, position the cursor between the two mapping commands displayed, right-click, and select **Operators** → **AND** as shown in Figure 18-45.

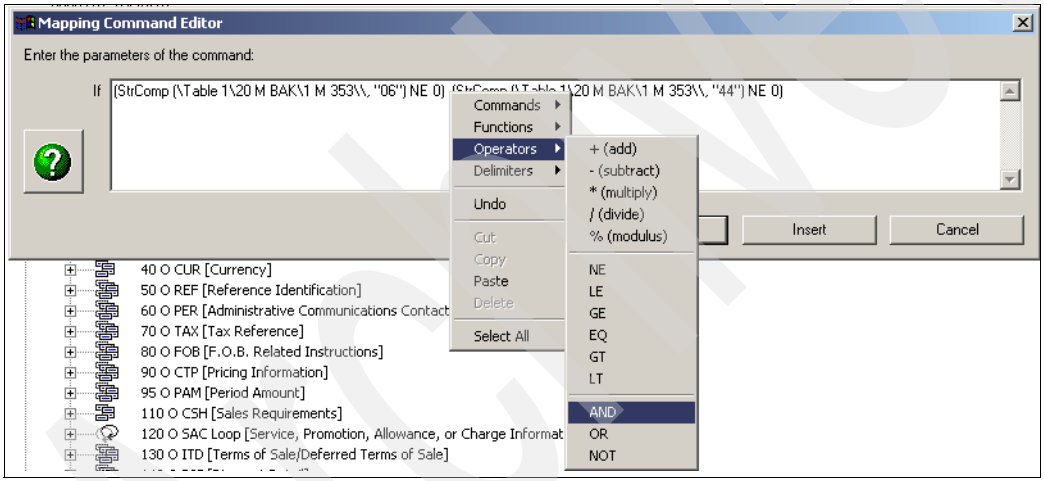


Figure 18-45 Link mapping commands

13. Complete the mapping command by adding an opening and closing parenthesis, then click **OK**.

14. When complete, your Mapping Command Editor should resemble the Mapping Command contents as shown in Example 18-2.

Example 18-2 Mapping command

```
((StrComp (\Table 1\20 M BAK\1 M 353\\, "06") NE 0) AND (StrComp (\Table 1\20 M BAK\1 M 353\\,
"44") NE 0))
```

Essentially, this command is telling the Data Interchange Services to execute the mapping commands contained within the If statement if the value received in

element 353 is not 06 or 44. At the moment, we have no mapping commands defined within the If statement. The mapping commands we create will throw an error and render the source input as invalid when they are executed.

15. To create these mapping commands, right-click the **`If`** command created earlier and select **Insert Within** → **Command** → **Error** as shown in Figure 18-46.

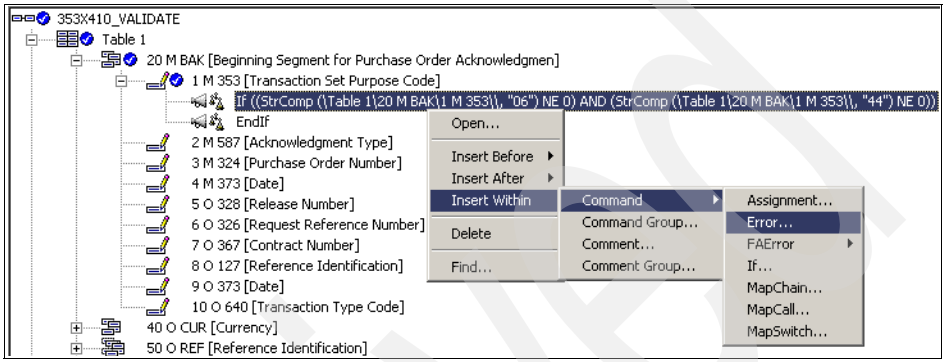


Figure 18-46 Insert error handling

16. The Mapping Command Editor opens with a template of the Error() function displayed as shown in Example 18-47.

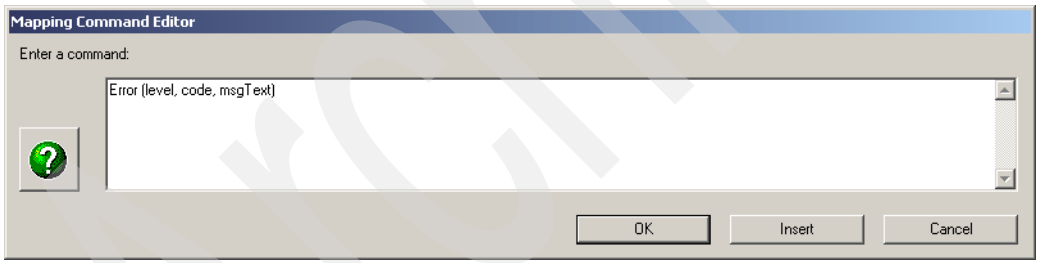


Figure 18-47 Error function

17. Select the level value that is being passed as a parameter to the Error() function and right click. Select **Values** → **1 (Simple Element Error)** as shown in Figure 18-48 on page 620.

Archived

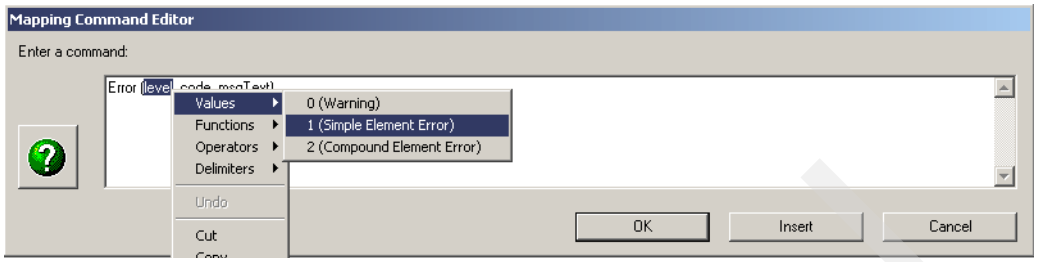


Figure 18-48 Select passed parameter

18. The `code` and `msgText` values are simply string values that are passed to the Error function and are later displayed in the log files if a validation error occurs. Overwrite `code` with any string value, for example `101`. Overwrite msgText with an error message, for example **"`Data element error`".**
19. Your mapping command should appear as shown in Figure 18-49. Click **OK** to complete the mapping command.

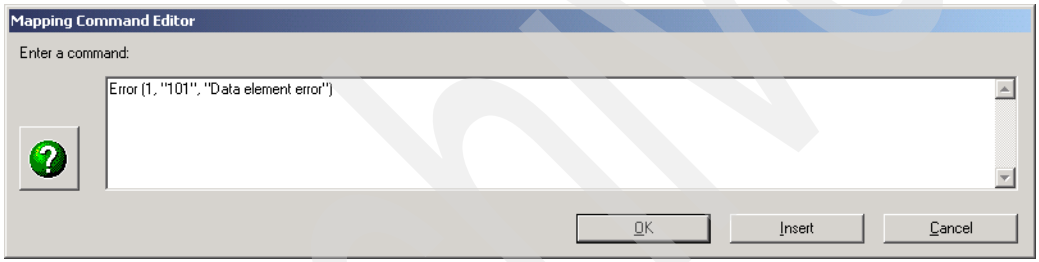


Figure 18-49 Mapping command

20. The mapping command window should appear as shown in Example 18-50.

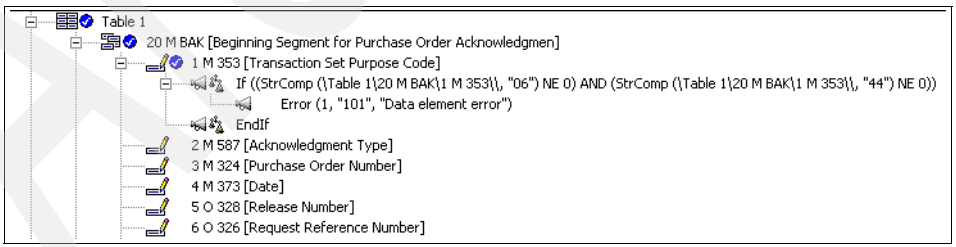


Figure 18-50 Mapping command window

21. This mapping command is all that is required to complete our validation map. Save, close and compile the map.

Archived

19

Enveloping and de-enveloping

Another important function we need to discuss is EDI enveloping and de-enveloping in WebSphere Partner Gateway. As we described in 17.4.1, “Elements of an EDI solution” on page 587, *enveloping* refers to grouping multiple EDI transactions into one single interchange.

Enveloping batch messages involves placing the EDI standard header and trailer around transactions in preparation for sending. When the envelope is complete, the package can then be sent to a trading partner through a VAN. Similarly, batch transactions must be de-enveloped when they are received from the VAN.

De-enveloping refers to unpackaging the single EDI interchange into multiple EDI transactions. For example, an EDI-X12 interchange might contain multiple transactions such as 850, 879, 890 and so on. When it is de-enveloped it is split into individual transactions and each transaction flows through the system individually.

Important: De-enveloping is different from splitting. *Splitting* applies when multiple interchanges are placed together in a single file (this is covered later in Chapter 21, “Batching and EDI splitting” on page 729). De-enveloping applies when multiple transactions are placed in a single interchange.

© Copyright IBM Corp. 2005. All rights reserved.

19.1 About EDI validation

Before we move into our scenario for EDI validation, it would be a good idea to explain what we mean when we discuss EDI validation.

As far as the EDI validation is concerned, we have two types of validation: data validation and segment validation (which is Service Segment validation). The next sections define what we mean by each of these two types of validation, what our scenarios cover, and what is available for each.

19.1.1 Data validation

We perform data validation using our validation maps. This will validate the real content of the EDI transaction. This will be performed using the action **EDI Validate** on an interaction. At the same time, this also does a cursory check on the ISA/GS segment fields. Because this is a cursory check, it only flags an error if the mandatory fields are not present. This however does not enforce a mandatory fill for the elements that are mandatory in the segment. As a result, the mandatory elements when present in the segment, with blank values, pass this level of service segment validation. For example, we can analyze the following EDI Interchange:

Example 19-1 Data validation example

```
ISA*00*          *00*          *  *111111111       *  *555555555
*241002*1635*U*00401*000000014*0*P*z!
GS*  *  *  *241002*1635*000000014* *004010!
ST*855*000000014!
BAK*06*AT*P365542*01101402!
PO1**132*EA*10.56*PE*BP*00031PO241!
PID*F****SCSI Card!
PO1**16*EA*1.99*PE*BP*00021PO241!
PID*F****IDE Card!
SE*7*000000014!
GE*1*000000014!
IEA*1*000000014!
```

In the EDI interchange in Example 19-1, the header information begins with `ISA*00*`. All the core data encapsulated by the interchange begins with `BAK*`. The header closes the message again, starting with `SE*7*`.

Data validation validates the data part of the EDI interchange. At the same time, data validation also checks for the minimal header information. For example in Example 19-1, if replace `ST` with `SY`, the EDI validator invoked by the action EDI Validate will flag an error. However at the same time, the EDI Validator will not look for the blank spaces for ISA05 and ISA07 (the Interchange Qualifiers). In

other words, we can say that the EDI Validator will allow the EDI document because it is *technically* correct in that all of the mandatory fields and segments are there, despite the fact that they might not contain any data required for normal business use.

The EDI Validator will not flag an error if blank values are encountered for mandatory elements in the service segments, such as the ISA05 and ISA07.

19.1.2 Service Segment validation

In Example 19-1 on page 622, the segments which begin and end the message are called Service Segments (the ISA/GS/ST segments). Some of the fields are left blank. If we want to force all the segments to be filled in and to verify their content, we can use the Service Segment validation by **s**witching on the attribute **Detailed Validation of Segment**. Detailed Validation of Segment provides in-depth validation of the Service Segments.

> **Important:** Detailed Validation of Segment will be used to validate only the ISA/GS/ST segments. The default setting for Detailed Validation of Segment is OFF.

With this knowledge, we can analyze Example 19-1 on page 622 again with Detailed Validation of Segment. In Example 19-1 on page 622, we can observe that many fields are blank in ISA/GS segments. Because of these blank fields, this document fails the detailed validation. With Detailed Validation of Segment turned on, WebSphere Partner Gateway will never accept the above document .

To correct that. we would have to modify the above document along the lines of Example 19-2, with the fields of the Service Segments filled with data.

Example 19-2 Completed Service Segments

```
ISA*01*ISA0200000*02*ISA0400000*ZZ*companya       *ZZ*companye
*050202*0518*U*00410*000000002*0*T* !
GS*PR*AAAAAAAAAAAAAAA*AAAAAAAAAAAAAAA*051002*1635*000000014*xx*004010!
ST*855*000000014!
BAK*06*AT*P365542*01101402!
PO1**132*EA*10.56*PE*BP*00031P0241!
PID*F****SCSI Card!
PO1**16*EA*1.99*PE*BP*00021P0241!
PID*F****IDE Card!
SE*7*000000014!
GE*1*000000014!
IEA*1*000000002!
```

As we see in Example 19-2 on page 623, we have all the fields populated. But all these changes are limited to ISA/GS segments, the data need not be modified because it is not the subject of the validation.

Important: To simplify our scenario examples, we have chosen *not* to turn on the Service Segment validation. Our validation is limited to validating our data using validation maps and the cursory Service Segment checking that this implies. We use technically correct EDI interchanges that contain the mandatory segments, but we do not fill them with the detailed business data that Service Segment validation would require.

19.2 Scenario overview

The participants involved in this scenario are Company E and Company A.

Our scenario discussing the enveloping and de-enveloping capabilities of WebSphere Partner Gateway can be described as follows. See Figure 19-1 on page 625.

For the enveloping scenario, an XML file is dropped into JMS Target of the WebSphere Partner Gateway Enterprise of Company E.

1. WebSphere Partner Gateway at Company E completes these tasks:
   a. Transforms the XML Purchase Order into an EDI-X12 850 Purchase Order transaction using the map XML_EDI_PO.
   b. Envelops the EDI transaction.
   c. Packages the EDI Purchase Order with AS packaging.
   d. Sends the EDI Purchase Order envelope to Company A using HTTP.
2. WebSphere Partner Gateway at Company A receives the EDI Purchase Order envelope and saves it to the file system gateway (.../companya/edi_in).

For the de-enveloping scenario, an EDI-X12 Purchase Order Acknowledgement envelope with AS packaging is posted from Company A to Company E.

1. WebSphere Partner Gateway at Company E completes these tasks:
   a. Receives the EDI envelope.
   b. De-envelopes the EDI Purchase order envelope into an EDI-X12 855 transaction.
   c. Transforms the EDI-855 into an XML purchase order acknowledgement.

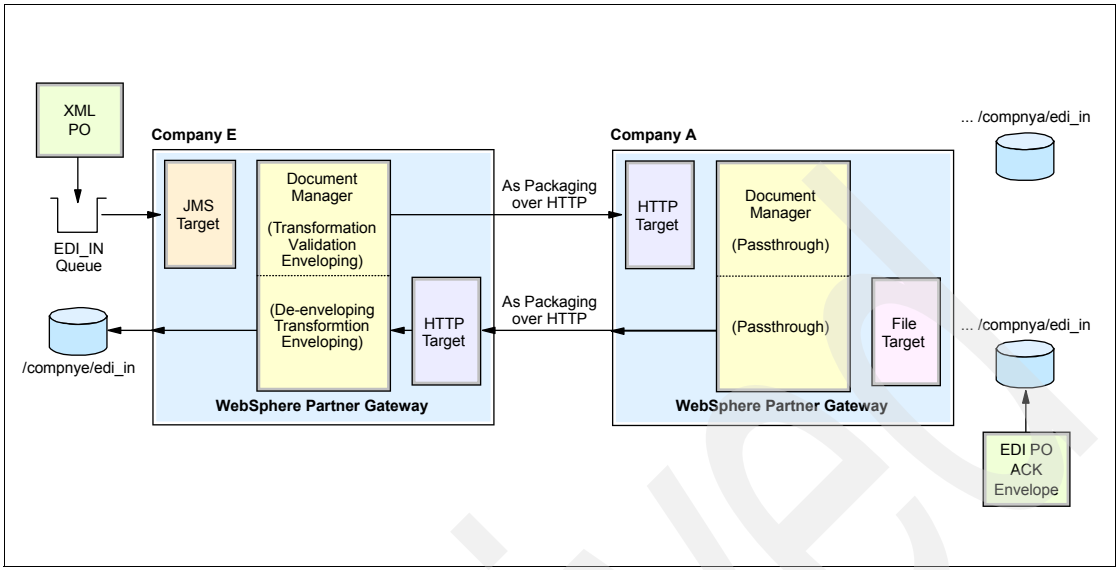


Figure 19-1 Business Scenario for enveloping and de-enveloping

19.3 Implement the enveloping scenario

To implement our enveloping scenario, we must first perform some additional configuration tasks on WebSphere Partner Gateway Enterprise on Company E (See the Figure 19-2 on page 626):

1. Create a JMS Target on Company E.
2. Upload the map which transforms the XML to EDI 850.
3. View the document flow definitions created by the map.
4. Create an XML format.
5. Upload the schema to validate the inbound XML and associate the schema with the document flow definition.
6. Create the custom action.
7. Create the necessary interactions.
8. Add the required attributes to the document flow definitions.
9. Update the Participant Profiles of Company E and Company A on the Company E Server.
10. Configure the envelope.
11. Activate the participant connections and set the required attributes.

Archived

No more additional configuration is required for WebSphere Partner Gateway Advanced which is on Company A.

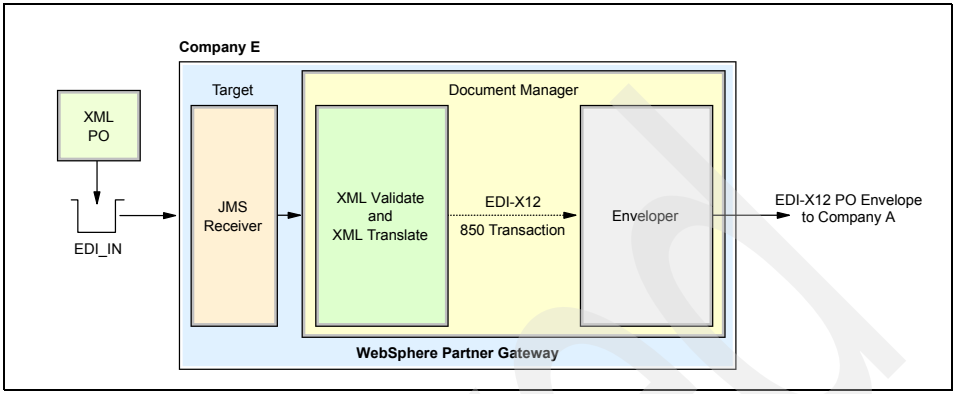


Figure 19-2 Enveloping scenario

To implement our de-enveloping scenario, no more additional configuration is required on WebSphere Partner Gateway Advanced on Company A. Here are the additional steps that we need to perform on Company E (Figure 19-3):

1. Upload the Map to perform EDI to XML conversion.
2. Upload the Validation Map
3. Create the necessary interactions for the document flow definition.
4. Update the Participant profiles of Company E and Company A.
5. Activate the participant connections and assigning the attributes.

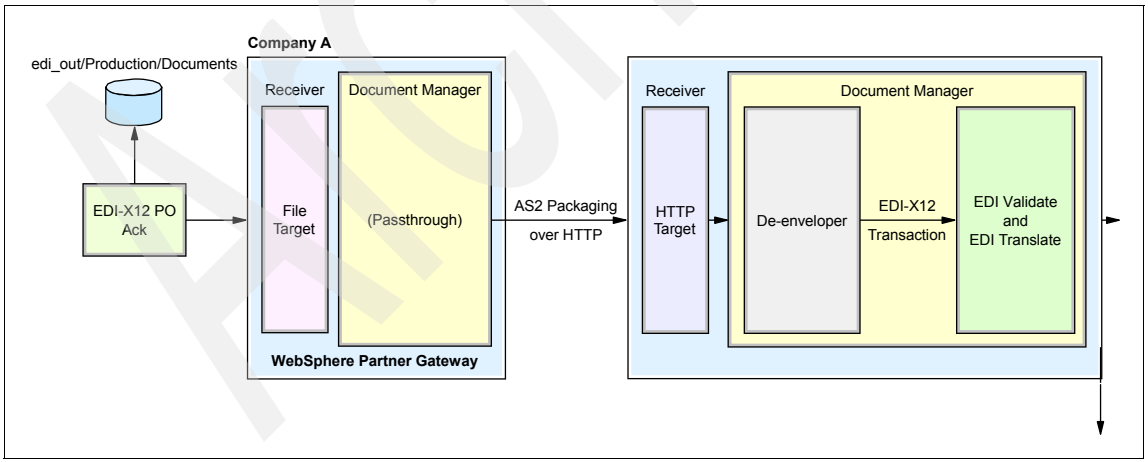


Figure 19-3 De-enveloping scenario

The following sections deal with each of the configuration steps in more detail.

19.4 Configure Company E for XML to EDI outbound

In this section, we perform the configuration for the outbound EDI, which encompasses the configuration of the enveloper.

Note: In this example, the EDI standard is: ANSI-X12, with the protocol being X12V4R1.

19.4.1 Enable JMS

WebSphere Partner Gateway uses the JMS API to interact with WebSphere MQ. The JMS API is part of the Java 2 Platform, Enterprise Edition (J2EE), specification. It is a layer that is developed on top of the standard WebSphere MQ API. You must set up the mapping between JMS objects and WebSphere MQ objects.

The mapping is retrieved by the JMS API using the Java Naming and Directory Interface™ (JNDI) API. JNDI providers can be anything, for example Lightweight Directory Access Protocol (LDAP) servers or naming services provided by other WebSphere Application Server instances in your network or a simple file. The actual mapping commands are independent of the JNDI provider.

WebSphere MQ provides the JMSAdmin tool to store the links between a JMS object and a WebSphere MQ object in a JNDI directory. Set up your choice of JNDI provider in the JMSAdmin.config configuration file, which you can find in the Java\bin directory in the WebSphere MQ installation folder. Perform the mapping on the machine where the JMS application is running, and not on the machine that runs the real JMS server (or queue manager).

On the WebSphere Partner Gateway Enterprise machine, open the file jmsadmin.config in the folder C:\WMQ\Java\bin in a text editor.

1. Uncomment the INITIAL_CONTEXT_FACTORY setting for the JNDI provider of your choice. Example 19-3 on page 628 is set for the file-based JNDI provider. Each JNDI provider needs to be addressed in a different way, which is expressed via the PROVIDER_URL parameter.
2. Update the file-based URL to point to a valid directory on the file system, for example C:\WMQ\Java\JNDI. Notice that the syntax of PROVIDER_URL uses the forward slash instead of the normal backward slash of Windows

Note: Make sure that you uncomment only one setting of each parameter. If you want to use multiple providers, you need to create multiple configuration files and pass the name of the configuration file as a parameter when running the tool JMSAdmin.

Example 19-3 JMSAdmin configuration file

```
#  The following line specifies which JNDI service provider is in use.
#  It currently indicates an LDAP service provider. If a different
#  service provider is used, this line should be commented out and the
#  appropriate one should be uncommented.
#
#INITIAL_CONTEXT_FACTORY=com.sun.jndi.ldap.LdapCtxFactory
INITIAL_CONTEXT_FACTORY=com.sun.jndi.fscontext.RefFSContextFactory
#INITIAL_CONTEXT_FACTORY=com.ibm.ejs.ns.jndi.CNInitialContextFactory
#INITIAL_CONTEXT_FACTORY=com.ibm.websphere.naming.WsnInitialContextFactory
#
#  The following line specifies the URL of the service provider's initial
#  context. It currently refers to an LDAP root context. Examples of a
#  file system URL and WebSphere's JNDI namespace are also shown, commented
#  out.
#
#PROVIDER_URL=ldap://polaris/o=ibm,c=us
PROVIDER_URL=file:/C:/WMQ/Java/JNDI
#PROVIDER_URL=iiop://localhost/
```

3. Use the JMSAdmin tool to create the mapping between JMS objects and MQ objects.

   Start JMSAdmin in a command window, with C:\WMQ\Java\bin as the current directory.

4. Create mapping between a queue connection factory object and a queue manager. We also need mapping between the JMS queue and the MQ queue.

   Example 19-3 lists the commands used in JMSAdmin to set up the mapping.

   a. Use the first command to create a context called WPG_JMS:

      ```
      def ctx(WPG_JMS)
      ```

      Tip: You can think of a *context* as a folder to store related objects. It is not required to create a context, however, it is always a good practice to group common objects together. Defining all JMS objects in the root context can make things confusing and difficult to manage.

   b. Switch to the newly created context:

      ```
      chg ctx(WPG_JMS)
      ```

   c. Create the JMS mappings within this context. The first object is called a *queue connection factory.* We set up the queue connection factory as an MQ client object. The queue connection factory uses the client transport to talk to the queue manager partner_e.bcg.queue.manager. We connect this queue manager using the MQ client channel object called

java.channel. The queue connection factory refers to the hostname and the port that is used by the MQ listener component (which we setup in 5.3.2, “Configuring WebSphere MQ 6.0” on page 65).

```
def QCF(WPG_QCF) tran(client) host(WPGV6DB) port(9999)
chan(java.channel) qmanager(partner_e.bcg.queue.manager)
```

d. The other object we need is a JMS queue that points to the WebSphere MQ queue that we will be using for to receive messages from back-end or other systems. The JMS queue JMS_EDI_IN refers to the local queue EDI_IN which is hosted by the partner_e.bcg.queue.manager.

```
def q(JMS_EDI_IN) qmanager(partner_e.bcg.queue.manager) queue(EDI_IN)
```

e. End the interactive command session in JMSAdmin, by entering `end`.

Example 19-4 Create JMS objects using JMSAdmin

```
5724-H72, 5655-L82, 5724-L26 (c) Copyright IBM Corp. 2002,2005. All Rights
Reser
ved.
Starting Websphere MQ classes for Java(tm) Message Service Administration

InitCtx> def ctx(WPG_JMS)

InitCtx> chg ctx(WPG_JMS)

InitCtx/WPG_JMS> def QCF(WPG_QCF) tran(client) host(WPGV6DB) port(9999)
chan(java.channel) qmanager(partner_e.bcg.queue.manager)

InitCtx/WPG_JMS> def q(JMS_EDI_IN) qmanager(partner_e.bcg.queue.manager)
queue(EDI_IN)

InitCtx/WPG_JMS> end
```

Important: If we inspect the contents of the c:\WMQ\Java\JNDI directory we see the WPG_JMS folder. This maps to the context created in the JMSAdmin session. Within this folder, the .bindings file is stored. This file is a text file that contains the JMS objects we created, and their mappings to actual MQ objects. Do not edit this file with a regular text editor, use only the JMSAdmin tools to manipulate the contents of this file.

5. Finally, using the WebSphere MQ explorer or runmqsc, create a local queue EDI_IN for queue manager partner_e.bcg.queue.manager.

Archived

19.4.2 Create a JMS Target

A *target* is a listening or a polling component where documents may come into WebSphere Partner Gateway from partners or the company's own applications. In this part of our scenario, we define a JMS Target. We have a queue on which a back-end application could put data in the form of messages that require transformation and transmission to a trading partner.

1. Log on to the console of WebSphere Partner Gateway as hubadmin and select **Hub Admin** → **Hub Configuration** → **Targets**.
2. Select **Create Target**.
3. For Target Name, type a new name, such as `JMSTarget`.
4. For Transport, select **JMS**. More attributes open in the window.
5. For JMS Provider URL, specify the provider URL for JNDI. In our setup, this URL is:

   `file://C:/WMQ/Java/JNDI/WPG_JMS`
6. For JNDI Factory Name, specify the implementation class of the JNDI interface:

   `com.sun.jndi.fscontext.RefFSContextFactory`
7. For JMS Factory Name, specify the queue connection factory `WPG_QCF`.
8. For JMS Queue Name, specify the queue `WPG_EDI_IN`.

   (As shown in Figure 19-4 on page 631)

Archived

Target Name JMSTarget *
Status Enabled Disabled
Description
Transport JMS *

Target Configuration

Gateway Type: Production * New Edit

JMS Provider URL: file:///C:/jndi/WPG_JMS *
User Id:
Password:

JMS Queue Name: JMS_EDI_IN
JMS Factory Name: WPG_QCF
Provider URL Package:
JNDI Factory Name: n.sun.jndi.fscontect.RefFSContextFactory

Time Out: 1 seconds
Thread Nbr: 1

Handlers

Figure 19-4 New JMS Target

9. Click **Save** to store the new target.

19.4.3 Transformation map

For this part of our scenario, we use the maps that were created in Chapter 16, “Advanced mapping” on page 541.

When the transformation map is imported into WebSphere Partner Gateway, the document definitions created in Data Interchange Services are displayed as document flow definitions (package, protocol, and document flow) on the Transformation Map and Manage Document Flow Definitions page.

For example, if we are converting an XML document to an X12 transaction, we import the map that defines the XML and X12 transaction document definitions and the transformation that is to take place.

- There are two methods for receiving the map files from the Data Interchange Services. If the Data Interchange Services client has a direct connection to the WebSphere Partner Gateway database, we can export the file directly to the database, as was shown in our example.

Archived

- Another likely scenario is that you can receive the files in e-mail or as an FTP transfer. In this case we would use the bcgDISImport utility to import the control string maps. See Example 19-5.

  To import the map control string using the utility:

  a. Open a command prompt and navigate to the directory WPG HOME\bin

  b. Enter **bcgDISImport.bat** with the parameters:

     - database username
     - database password
     - map control string (where this is the complete path of the map control string file exported from Data Interchange Services client.

  c. Press Enter.

     The Map is uploaded to WebSphere Partner Gateway database.

Example 19-5 Using bcgDISImport.bat

```
C:\<WPG HOME>\bin>bcgDISImport.bat <DB_UserID> <DB_Password>
<Controlstring.eif>
```

Note: Transformation and validation maps exported from the Data Interchange Services client or imported with the **bcgDISImport** utility cannot be downloaded from the WebSphere Partner Gateway Community Console. You administer these maps by connecting to the WebSphere Partner Gateway database through the Data Interchange Services client.

Upload the XMI to EDI Map using bcgDISImport.bat

Complete these tasks to transform and validate the data.

1. Copy the map **XML_EDI_PO.eif** from file system to WebSphere Partner Gateway Home\bin folder
2. Use the **bcgDISImport.bat** utility to import the maps to WebSphere Partner Gateway database, as shown in Example 19-6.

Example 19-6 XML_EDI_PO control string import

```
D:\WPG6.0\bcghub\bin>bcgDISImport.bat   bcgcon bcgcon XML_EDI_PO.eif
ECHO is off.
JDBC_URL = jdbc:db2://wpgv6db.itso.ral.ibm.com:50000/bcgapps
JDBC_DRIVER_CLASS = com.ibm.db2.jcc.DB2Driver
JDBC_QUAL = DB2ADMIN
USERID = bcgcon
IMPORT_FILE = XML_EDI_PO.eif
JAVA_HOME = D:\was\java
ECHO is off.
Connection to database made. Starting import process.
```

```
Control string description records...1
Control string records...............25
----------------------------------------------
Total number of records in import file:    26
Import complete.
```

When the transformation map control string has been uploaded, the document flow definitions created by the map can be viewed in the console:

1. Log-in to the Community Console.
2. Select **Hub Admin** → **Hub Configuration** → **Maps** → **Transformation Maps.**
3. We can see the XML_EDI_PO map under Manage Transformation Maps Tab (See Figure 19-5).

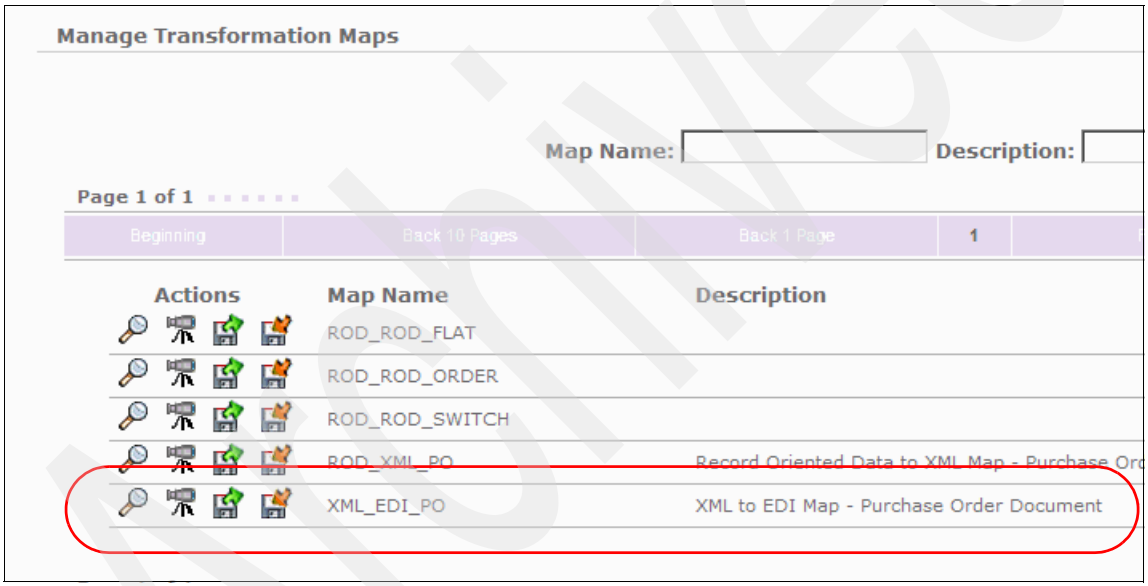


Figure 19-5 Transformation map XMl_EDI_PO

4. Select the icon next to the XML_EDI_PO map to view the document flow definitions created by the map.
5. We now see the Document Flow definitions created by the uploaded transformation map under Manage Transformation Maps (See Figure 19-6 on page 634).

   We see that the source document flow is None/CUSTOM_XML/PURCHASE_ORDER

   The target document flow is NA/X12V4R1/850.

Archived

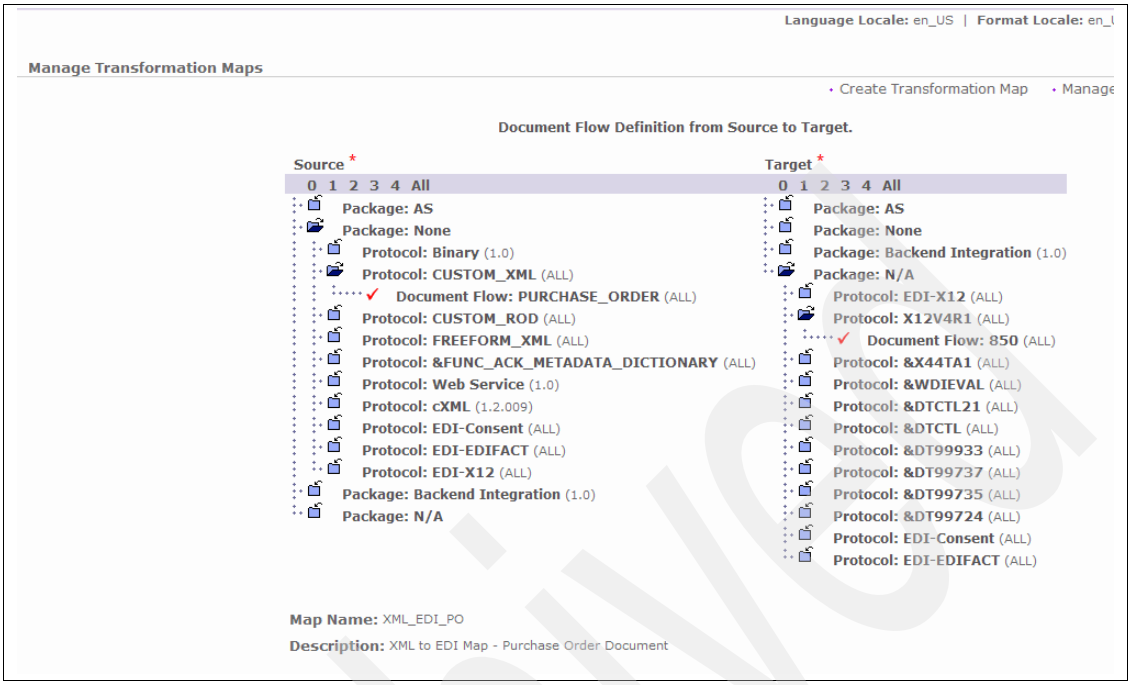


Figure 19-6 Document flow definitions created by the Map XML_EDI_PO

Because the map import has created Source and Target document flow definitions, we can move on to the next task of creating the XML formats and associating them with the XML document flow definition created by the map import.

19.4.4 Creating a new XML format

As discussed in 10.4.3, “Creating a new XML format” on page 290 a new XML format has to be created to let WebSphere Partner Gateway know about the structure of the incoming XML document.

Before proceeding to create the XML format, it would be better to examine the inbound XML document so that the necessary attributes required for the creation of an XML format can be noted down. See the Example 19-7.

Example 19-7 Sample XML document

```
<?xml version="1.0" encoding="UTF-8"?>
<purchaseOrder>
<orderHeader>
<orderNumber>P345322</orderNumber>
<orderFlag>0</orderFlag>
```

```
<orderDate>20030410</orderDate>
<contractNumber>00049IRL0092</contractNumber>
<supplierNumber>companya</supplierNumber>
<supplierName>Company A</supplierName>
<supplierAddress1>43 High Street</supplierAddress1>
<supplierAddress2>Dublin</supplierAddress2>
<supplierAddress3>Ireland</supplierAddress3>
<buyerName>Jane Doe</buyerName>
<buyerSerial>i83731</buyerSerial>
<currencyCode>IRL</currencyCode>
<paymentTerms>45 days</paymentTerms>
<customerNumber>companye</customerNumber>
<deliveryAddressName>Company E</deliveryAddressName>
<deliveryAddress1>North Carolina</deliveryAddress1>
<deliveryAddress2>USA</deliveryAddress2>
</orderHeader>
<lineDetail>
<lineNumber>1</lineNumber>
<partNumber>00021P0241</partNumber>
<itemDesc>SCSI Card</itemDesc>
<quantity>128</quantity>
<uom>EA</uom>
<unitPrice>98.62</unitPrice>
<dueDate>31122005</dueDate>
/lineDetail>
<lineDetail>
<lineNumber>2</lineNumber>
<partNumber>00031P0241</partNumber>
<itemDesc>IDE Card</itemDesc>
<quantity>100</quantity>
<uom>EA</uom>
<unitPrice>69.62</unitPrice>
<dueDate>20030630</dueDate>
</lineDetail>
<lineDetail>
<lineNumber>3</lineNumber>
<partNumber>00087P6821</partNumber>
<itemDesc>PCI Card</itemDesc>
<quantity>128</quantity>
<uom>EA</uom>
<unitPrice>1098.62</unitPrice>
<dueDate>20030630</dueDate>
</lineDetail>
<lineDetail>
<lineNumber>4</lineNumber>
<partNumber>00021P0241</partNumber>
<itemDesc>Deck of Cards</itemDesc>
<quantity>128</quantity>
<uom>EA</uom>
```


Archived

```
<unitPrice>98.62</unitPrice>
<dueDate>20030630</dueDate>
</lineDetail>
<orderTrailer>
<totalDollars>12623.36</totalDollars>
<totalQuantity>528</totalQuantity>
</orderTrailer>
</purchaseOrder>
```

1. In the console menu of WebSphere Partner Gateway, while logged on as hubadmin, select **Hub Admin** → **Hub Configuration** → **XML Formats**.
2. Select **Create XML Format** to go the page View XML format.
3. Fill in the following details in the View XML format page (See Figure 19-7 on page 637).
   a. Because we need to associate this XML format to the XML protocol of the incoming XML document, select **CUSTOM_XML** for Routing Format.
   b. Select the File Type as **XML.**
   c. For Identifier type,Select **Root Tag** and give the value as `purchaseOrder`, which matches with our example XML document shown in Example 19-7 on page 634.
   d. In the Schema Attribute section, for the Schema attributes Source and Target Business ID, select the **Type** as **Element Path** and give the values for these attributes as relative path of the sample XML document.

      For example, give the relative path for the Source BusinessID as:

      `/purchaseOrder/orderHeader/customerNumber`

      Similarly, fill in the value for Target Business ID as:

      `/purchaseOrder/orderHeader/supplierNumber`

      This is similar to what we did in 10.4.3, “Creating a new XML format” on page 290.
   e. Select the **Type** of the Schema Attributes for Source Document flow and Source Document flow Version as **Constant**.

      The Source Document Flow is `PURCHASE_ORDER` and Source Document Flow version as `ALL`.
4. Select **Save.**

Archived

View XML Format

Routing Format: CUSTOM_XML ALL *
File Type: XML *
Identifier Type: Root Tag * purchaseOrder *

Schema Attributes

Name	Type	Value
Source BusinessId	Element Path	/purchaseOrder/orderHeader/customerNumber
Target BusinessId	Element Path	/purchaseOrder/orderHeader/supplierNumber
Source Document Flow	Constant	PURCHASE_ORDER
Source Document Flow Version	Constant	ALL
Document Identifier	Element Path	
Document Timestamp	Element Path	
Duplicate Check Key 1	Element Path	
Duplicate Check Key 2	Element Path	
Duplicate Check Key 3	Element Path	
Duplicate Check Key 4	Element Path	
Duplicate Check Key 5	Element Path	

Save Cancel

Legend
* Required fields
Click to edit record
Click to delete record

Figure 19-7 Creating the XML format with the root tag purchaseOrder

19.4.5 Upload the XML Validation Map (XSD)

As the XML document is going to be translated into an EDI document, it needs to be validated first. WebSphere Partner Gateway validates the incoming XML documents against a DTD or a schema (XSD). For this particular example, we are using the schema to validate against the incoming XML document.

1. Log in to WebSphere Partner Gateway console as hubadmin and Select **Hub Admin** → **Hub Configuration** → **Maps** → **Validation Maps**.
2. Select **Create** to upload the new validation map.

Archived

Note: All the Schemas and DTDs that are uploaded to WebSphere Partner Gateway Database are treated as Validation Maps and can be found at Hub **Admin** → **Hub Configuration** → **Maps** → **Validation Maps**. A Validation map can be uploaded into WebSphere Partner Gateway either by uploading through the console or by using the DIS Client. When WebSphere Partner Gateway is installed, three Validation maps are uploaded by default. See Figure 19-8.

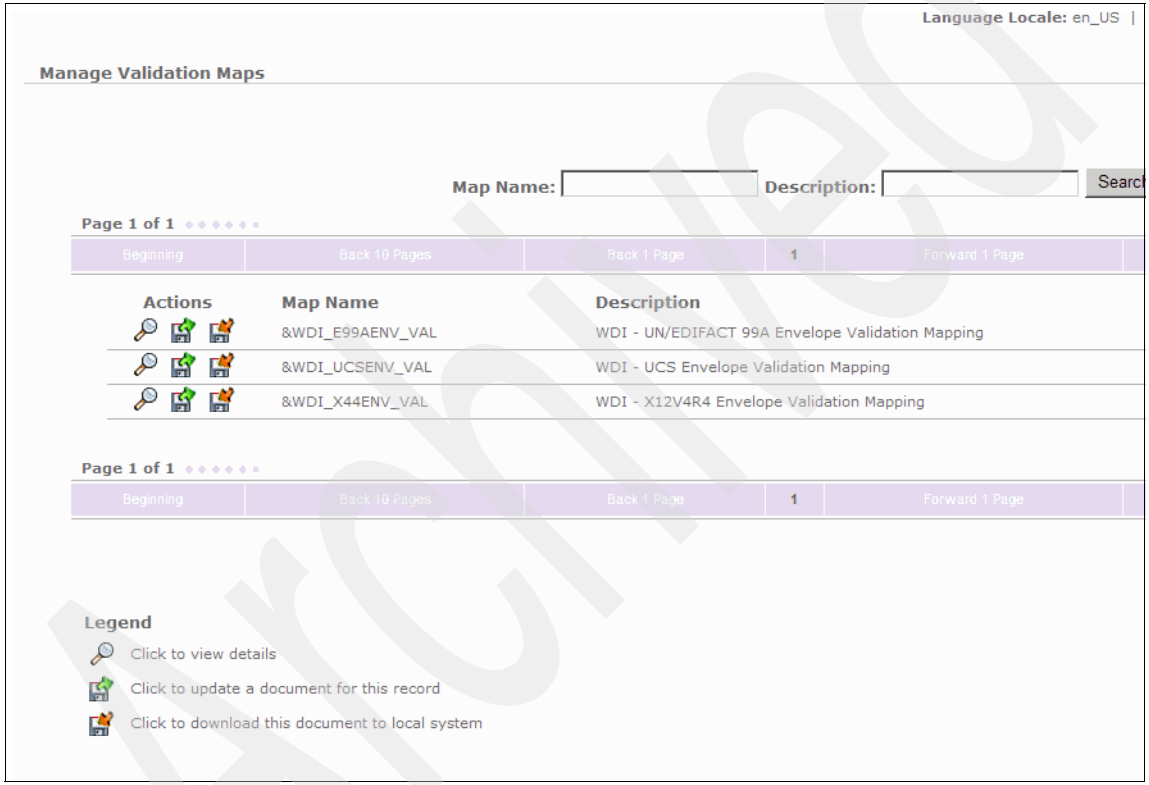


Figure 19-8 Manage validation Maps screen with the List of Default Maps

3. In the Validation Map Create Screen, select **Browse** to browse through Windows file system to choose the **purchaseOrder.xsd** file (See Figure 19-9 on page 639) and select **Save.**

Archived

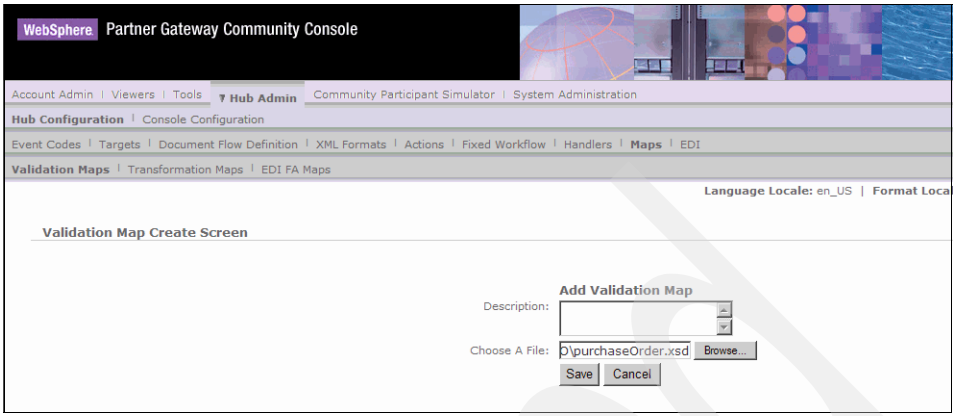


Figure 19-9 Uploading the Validation map purchaseOrder.xsd

4. We now see the uploaded schema (XSD) in the Manage Validations maps page. As we want to bind this validation map to the XML document flow definition that was created by the map, it has to be associated to the XML document flow definition.

   Select the icon next to the **purchaseOrder.xsd** to view the Manage Validations Page.

5. Under the System Levels, elect the **Package**, **Protocol** and **Document flow** that the validation map is to be associated with (See Figure 19-10 on page 640).

   In this case select **None/CUSTOM_XML/PURCHASE_ORDER**.

6. Select **Save.**

Archived

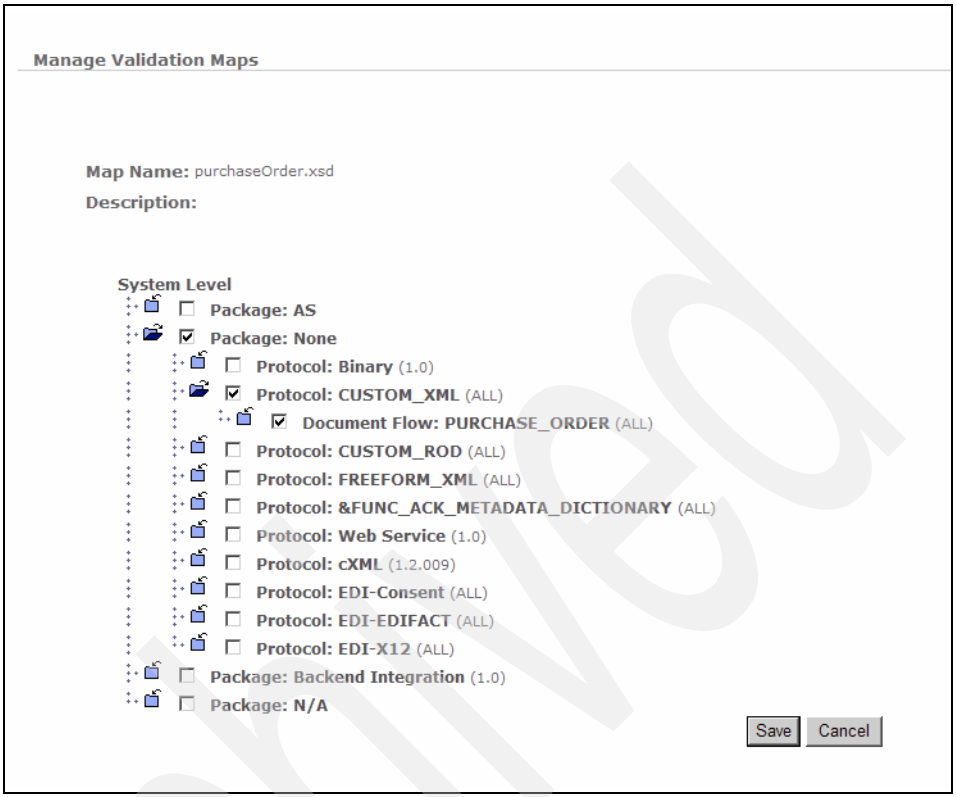


Figure 19-10 Associating the Validation map purchaseOrder.xsd with the Document flow definitions

Note: Multiple validation maps can be associated with the same document flow definition.

19.4.6 Add attributes to the document flow definition

For our enveloping scenario, WebSphere Partner Gateway expects two attributes to be in the document flow definition. These attributes need to be added to the document flow definition explicitly. The attributes that are going to be added are Validation Map and Interchange Identifier.

While logged in as hubadmin, follow these steps:

1. Select **Hub Admin** → **Hub Configuration** → **Document flow Definition**.
2. Select the **None** Package and select the **blue arrow** icon next to the Protocol **CUSTOM_XML.** See Figure 19-11 on page 641.

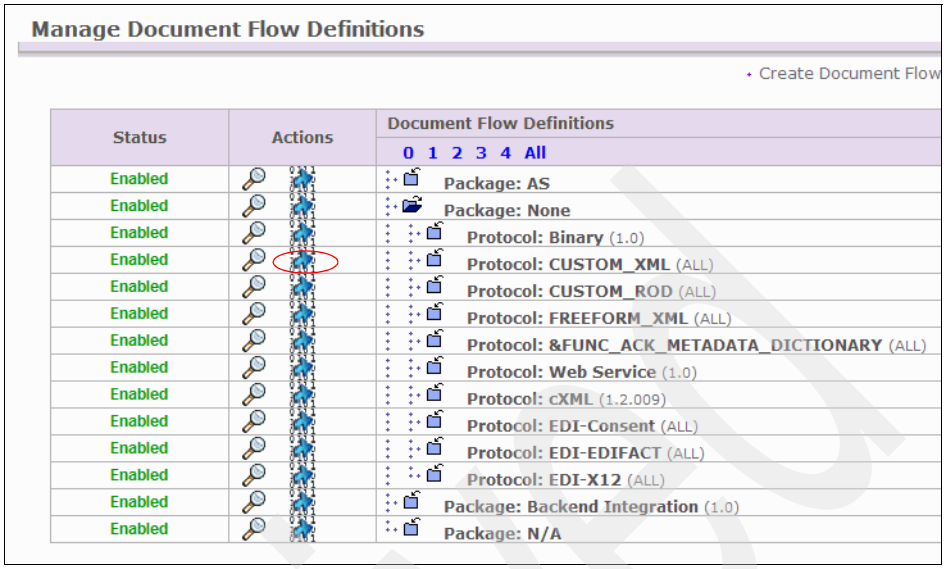


Figure 19-11 Select actions

3. In the Update Document flow definitions page, select **Add attributes** under Document flow Context Attributes. See Figure 19-12 on page 642.

Archived

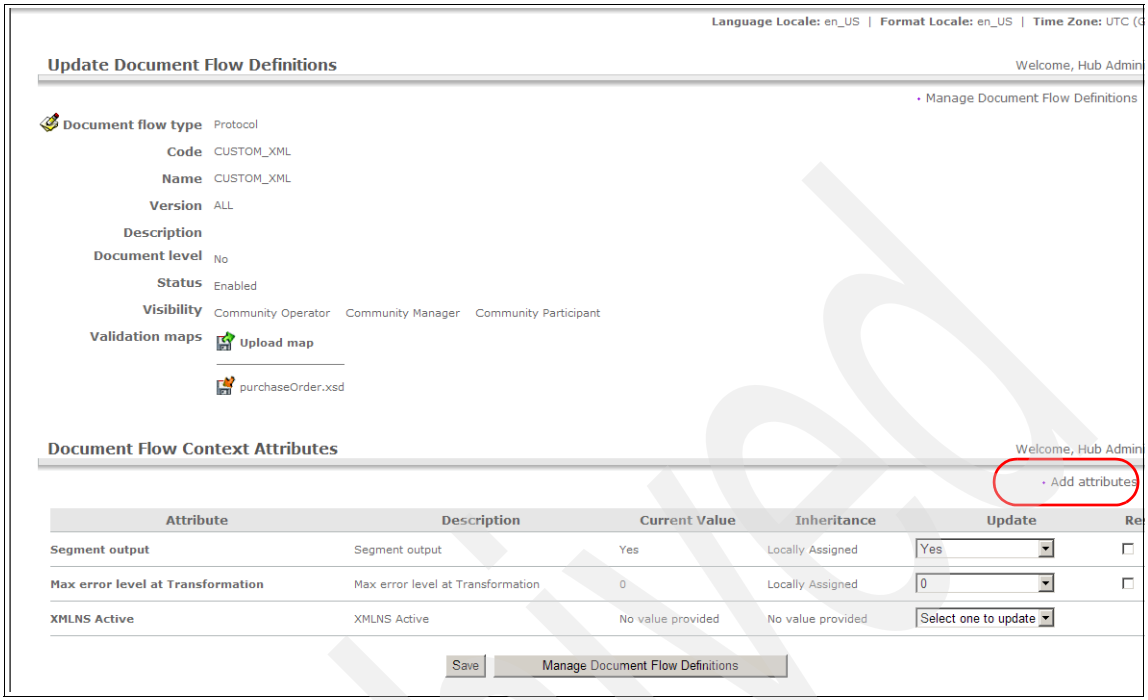


Figure 19-12 Adding new attributes to the Protocol CUSTOM_XML

4. In the Attributes list, select the check boxes next **Validation Map** and **Interchange Identifier** (see Figure 19-13 and Figure 19-14 on page 643). Select **Save** to return to the Update Document Flow definitions page.

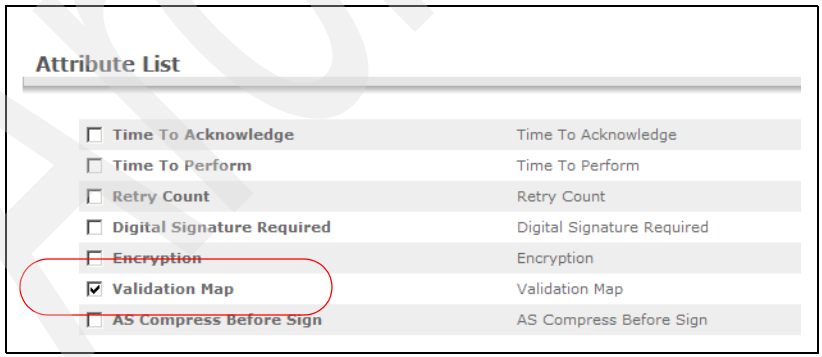


Figure 19-13 Adding the attribute Validation Map

Archived

✓	Segment output	Segment output
☐	Allow documents with duplicate document ids	Allow documents with duplicate document ids
✓	Max error level at Transformation	Max error level at Transformation
☐	FAMap	FAMap
☐	Envelope Profile	Envelope Profile
✓	XMLNS Active	XMLNS Active
☐	Max validation error level	Max validation error level
☐	Validation level	Validation level
☐	Char set validation table	Char set validation table
☐	Alphanumeric validation table	Alphanumeric validation table
☐	Generate group level info only in functional Ack	Generate group level info only in functional Ack
☐	Century control year	Century control year
☐	Detailed validation of segment	Detailed validation of segment
☐	Allow a TA1 request	Allow a TA1 request
☐	Discard Envelope if any errors	Discard Envelope if any errors
☐	Connection Profile Qualifier1	Connection Profile Qualifier1
☐	Interchange qualifier	Interchange qualifier
☑	Interchange Identifier	Interchange Identifier
☐	Interchange usage indicator	Interchange usage indicator
☐	Group application sender identifier	Group application sender identifier
☐	Group application receiver identifier	Group application receiver identifier
☐	Interchange reverse routing	Interchange reverse routing

Figure 19-14 Adding the attribute Interchange Identifier

This finishes the configuration required to add the attributes to the Protocol CUSTOM_XML.

19.4.7 Create the custom action

Because the inbound XML has to be validated before it is transformed, we need to create a custom action in WebSphere Partner Gateway console.

> **Note:** When we install the WebSphere Partner Gateway, we will be able to observe only a finite set of Actions, 17, in the Console. The introduction of User exits gives tremendous flexibility to the user so that you create any kind of an action with the available Handlers.

While logged in as hubadmin, follow these steps:

1. Select **Hub Admin** → **Hub Configuration** → **Actions** and select **Create** (Figure 19-15 on page 644).

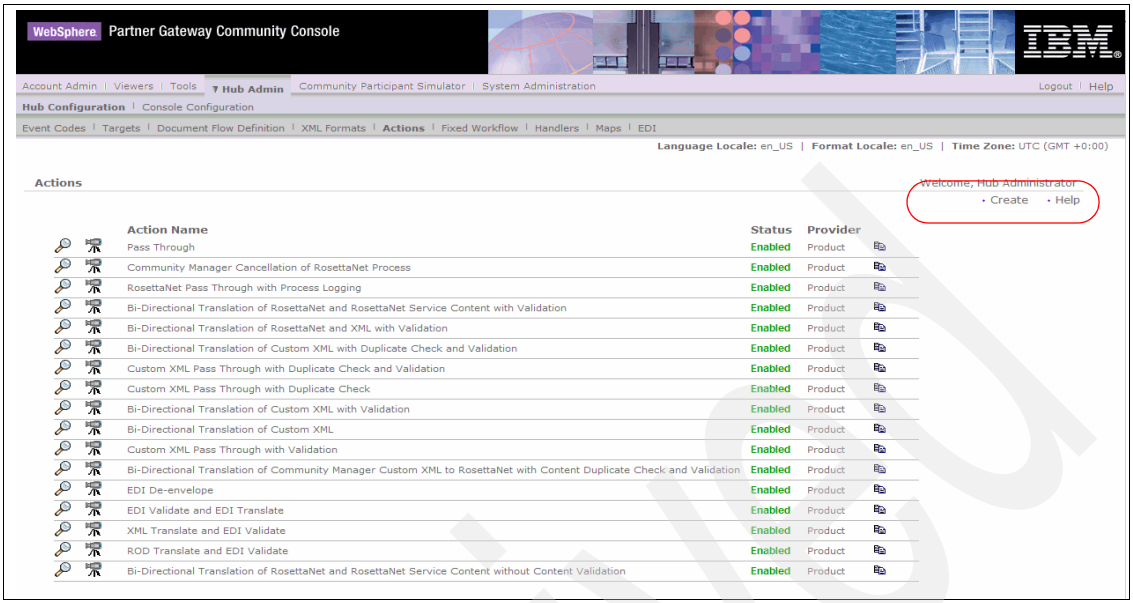


Figure 19-15 Creating a new action in WebSphere Partner Gateway Console

2. Give the Name of the Actions as `XML Validate, XML Translate and EDI Validate.`
3. Select **Enabled**.
4. Select the following handlers one by one from the **Available List** and add them to the **Configured List** by clicking **ADD** (as shown in Figure 19-16 on page 645)

   a. **com.ibm.bcg.validation.ValidationFactory**
   b. **com.ibm.bcg.edi.business.process.XMLTranslatorFactory**
   c. **com.ibm.bcg.edi.business.process.EDITargetValidationFactory**

> **Important:** Make sure that the order of the Handlers is restored correctly as shown in the Figure 19-16 on page 645. When the Action is invoked, it will try to execute the handlers in the order of the Configured List. For this reason, the order of the handlers should be maintained properly.

5. Select **Save to** save the Action and return to the Actions Page.

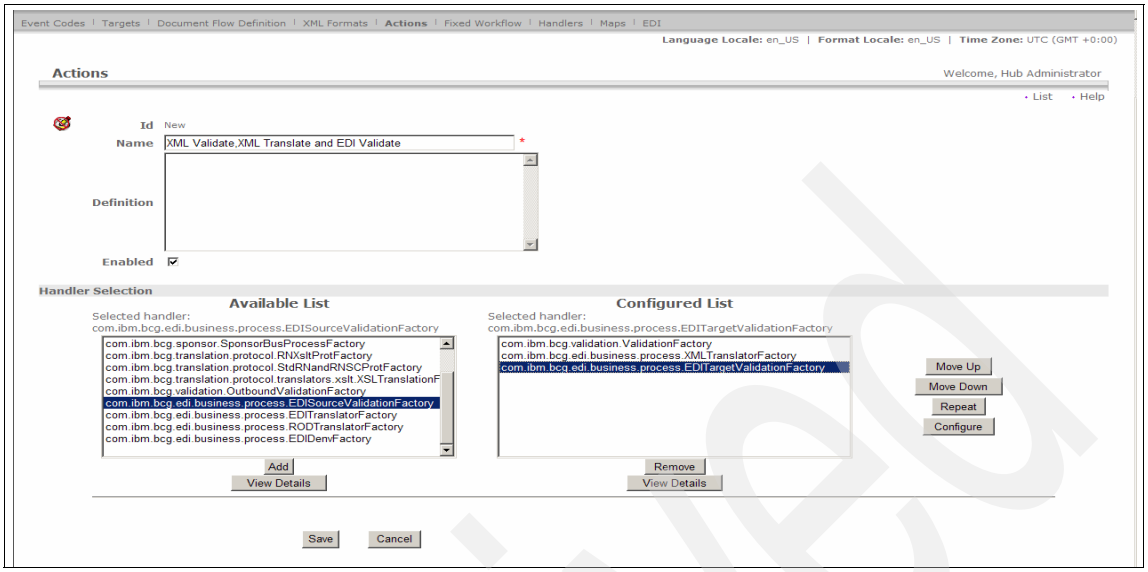


Figure 19-16 Creating the new action XML Validate,XMI Translate and EDI Validate

19.4.8 Create interactions

Interactions are the global set of channels that are possible with WebSphere Partner Gateway. WebSphere Partner Gateway understands the Document Flow from source to target using interactions.

For our scenario, we need to create two interactions:

1. We need the basic interaction which transforms the XML purchase order to EDI-X12 850 transaction.
2. The second interaction is the one which enables the EDI transaction to be enveloped.

Following is the procedure to create these interactions:

1. While logged on as hubadmin, select **Hub Admin** → **Hub Configuration** → **Document Flow Definition**.
2. Select **Manage Interactions**.
3. Select **Create Interaction**.
4. We need an interaction from None/CUSTOM_XML/PURCHASE_ORDER to N/A/X12V4R1/850 in the Package/Protocol/Document Flow order.

Archived

Note: The N/A Packaging is a new enhancement to WebSphere Partner Gateway. All of the raw EDI transactions which are yet to be enveloped or just de-enveloped will be placed under the N/A Packaging.

5. Select **XML_EDI_PO** from the Transform map drop-down box.
6. Select **XMLValidate,XML Translate and EDI Validate** from the Action drop-down box (Figure 19-17). Select **Save** to return to the Manage Interactions page. This finishes the creation of the first required interaction.
7. Select **Save.**

Note: The Action we have selected for this interaction is the one which we have created in the earlier section.

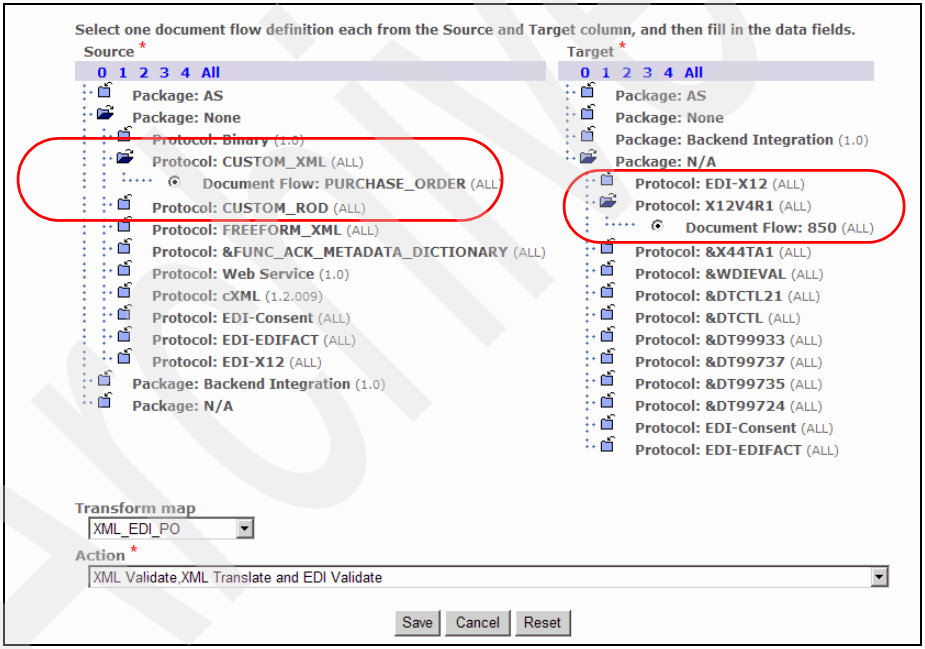


Figure 19-17 From None/CUSTOM_XML/PURCHASE_ORDER to N/A/X12V4R1/850

8. the second interaction we create is for the enveloping channel and it has to be created from N/A/EDI-X12/ISA to AS/EDI-X12/ISA.
9. No transformations are required for the enveloping channel, so directly select the Action as **Passthrough** (Figure 19-18 on page 647).
10. Select **Save.**

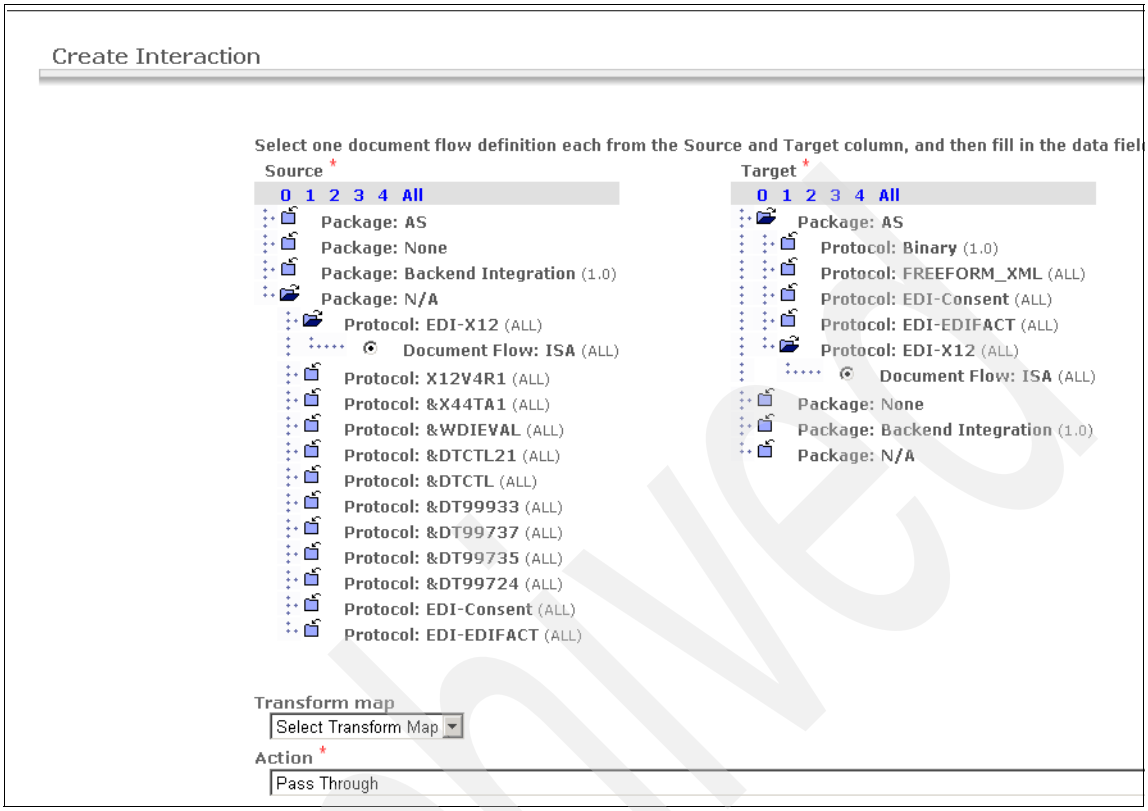


Figure 19-18 Creating the interaction from N/A/EDI-X12/ISA to None/EDI-X12/ISA

Note: Be aware of the following things about the interactions that we created:

- The Source document flow definitions for the interactions created are None/CUSTOM_XML/PURCHASE_ORDER,N/A/EDI-X12/ISA. Because Company E is the Source Participant in this example, its profile must be updated with these B2B capabilities.
- The Target document flow definitions for the interactions created are N/A/X12V4R1/850, AS/EDI-X12/ISA**.** Because Company A is the Target Participant in this example, its profile must be updated with the these B2B capabilities.

Archived

19.4.9 Update the Participant Profiles

As we did in the earlier chapters, we need to update the profiles of Company E and Company A to enable the XML to EDI document flow between these two partners.

Update the profile of the Company E on Company E Server

B2B capabilities for the Participant Company E must be updated on the WebSphere Partner Gateway Enterprise running on Company E:

1. While logged in as hubadmin, open the profile for Company E.
2. Select **B2B capabilities**.
3. In the B2B capabilities window, update the capabilities by **enabling** None,CUSTOM_XML,PURCHASE_ORDER,and N/A,EDI-X12,ISA in the order of Package,Protocol and document flow (Figure 19-19).

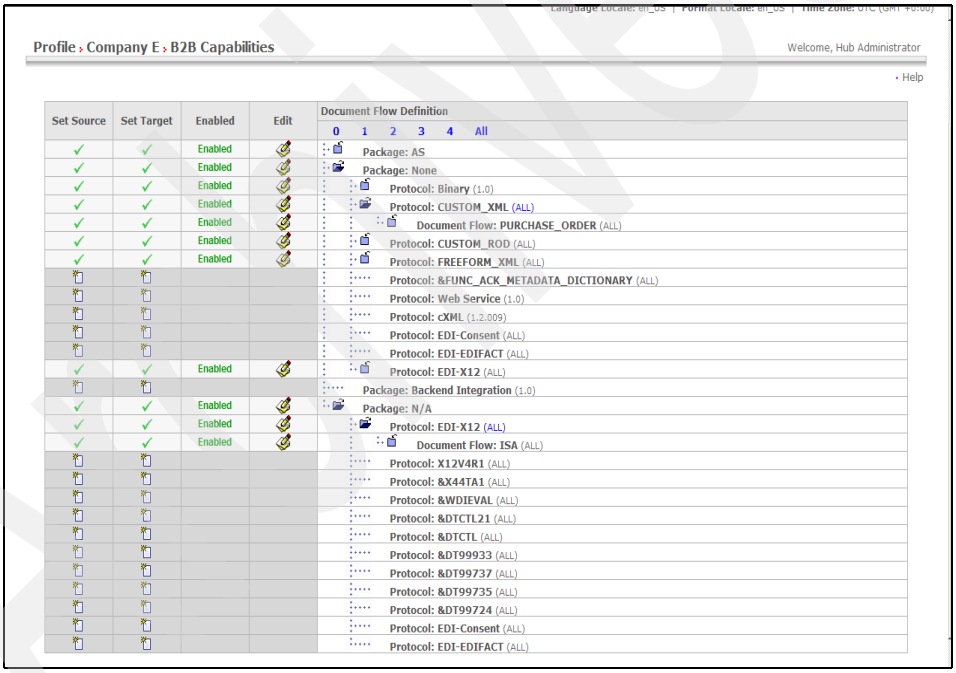


Figure 19-19 Updating the B2B capabilities for Company E

For the gateway configuration for the participant, there is no need to create another Gateway as default gateway for Company E is already present.

Update the profile of the Company A on Company E server

B2B capabilities of the Participant Company A must be updated with the new EDI capabilities on the WebSphere Partner Gateway Enterprise server running on Company E:

1. While logged in as hubadmin, open the profile for Company A.
2. Select **B2B capabilities**.
3. In the B2B capabilities window, update the capabilities by enabling N/A,X12V4R1,850 and AS,EDI-X12,ISA in the order of Package,Protocol and document flow (Figure 19-20).

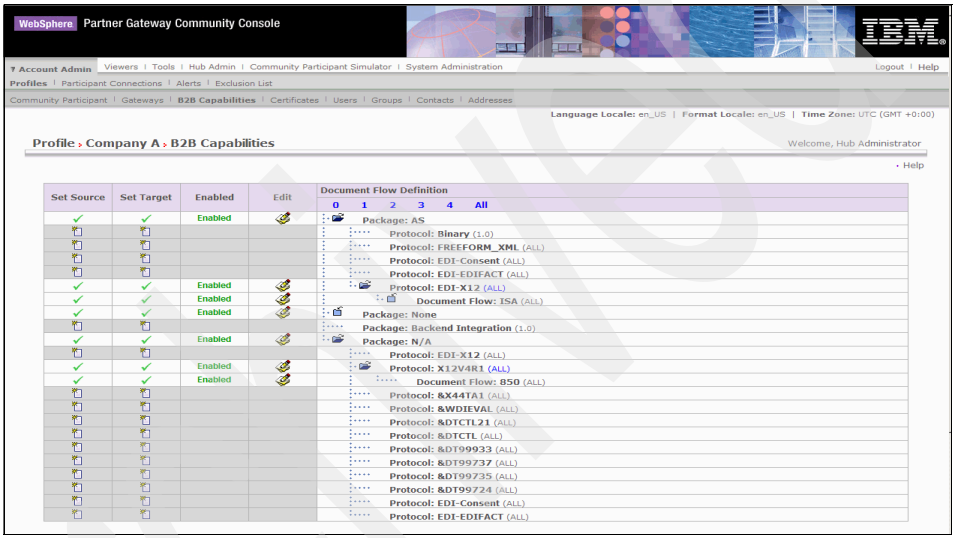


Figure 19-20 Updating the B2B capabilities for Company A

As far as the gateway configuration for Company A is concerned, there is no need to create another gateway as the default gateway is already present.

This completes the configuration for updating the profiles of the participants. Our next step is to create a new envelope profile

19.4.10 Configure the enveloper

Enveloper configuration in WebSphere Partner Gateway constitutes of two steps:

1. Creation of the envelope profile
2. Setting the enveloper scheduling interval

Archived

Create a new envelope profile

An *envelope profile* determines values that are placed in specific elements of the envelope. You assign the envelope profile to EDI transactions in the document flow definition in the Envelope Profile attribute.

WebSphere Partner Gateway provides a predefined envelope profile for each supported standard (X12, EDIFACT, or UCS). You can use these predefined envelope profiles directly, you can modify them, or you can copy them into new envelope profiles.

For our scenario, we will create a new envelope profile:

1. While Logged in as hubadmin, select **Hub Admin** → **Hub Configuration** → **EDI** → **Envelope Profile**
2. In the next page under the Envelope profiles list, we can see the Default Envelope profiles. Select **Create.**

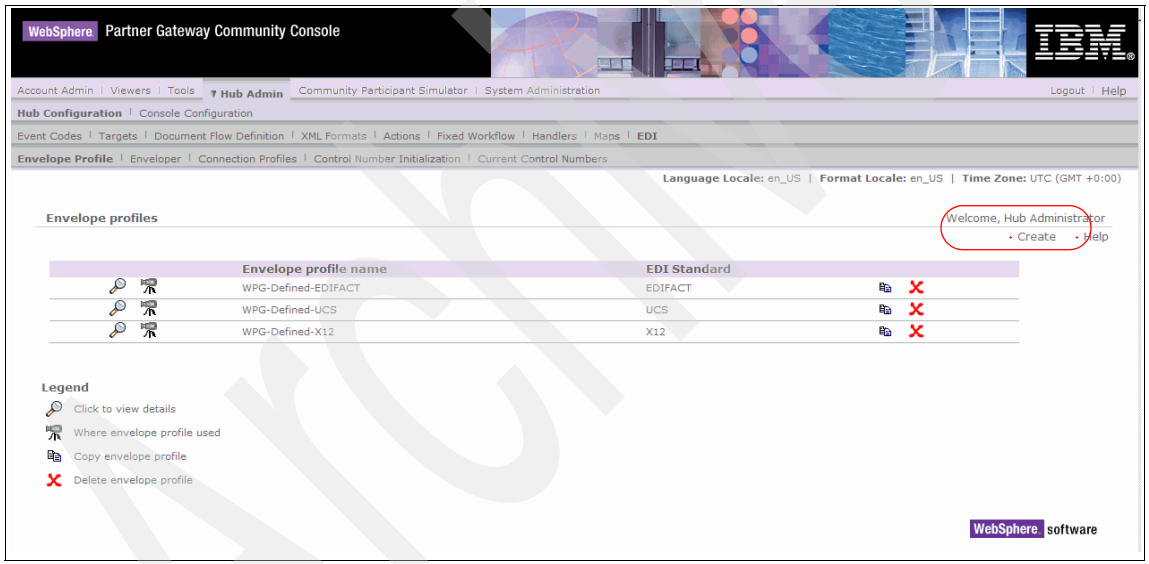


Figure 19-21 List of Default Envelope Profiles

3. In the next page, give the envelope profile a name (such as `Profile_E`) and select the EDI Standard as **X12**.
4. For the Envelope Profile Attributes, select **General** and use the values in Table 19-1 on page 651 to complete the form.
   a. The attributes INTCTLLEN,GRPCTLLEN,TRXCTLLEN denote the Control Number lengths for Interchange,Group and Transaction segments

Archived

respectively. The default values for these attributes in WebSphere Partner Gateway Enterprise is 9 (see Figure 19-22 on page 652).

b. Because we are dealing with EDI-X12 Protocol, the attribute ENVTYPE should be **X12**.

c. The attribute MAXSDOCS-Max Transaction Numbers per envelope can be given as 1000.

d. Select the CTLNUMFLAG as **N,** set by default in WebSphere Partner Gateway. See Figure 19-22 on page 652.

Note: For more information about these attributes, see the *Hub Administration Guide.*

Table 19-1 General Attributes for the envelope profile Profile_E

Attribute	Value
INTCTLLEN	9
GRPCTLLEN	9
TRXCTLLEN	9
ENVTYPE	X12
MAXDOCS	1000
CTLNUMFLAG	N

Archived

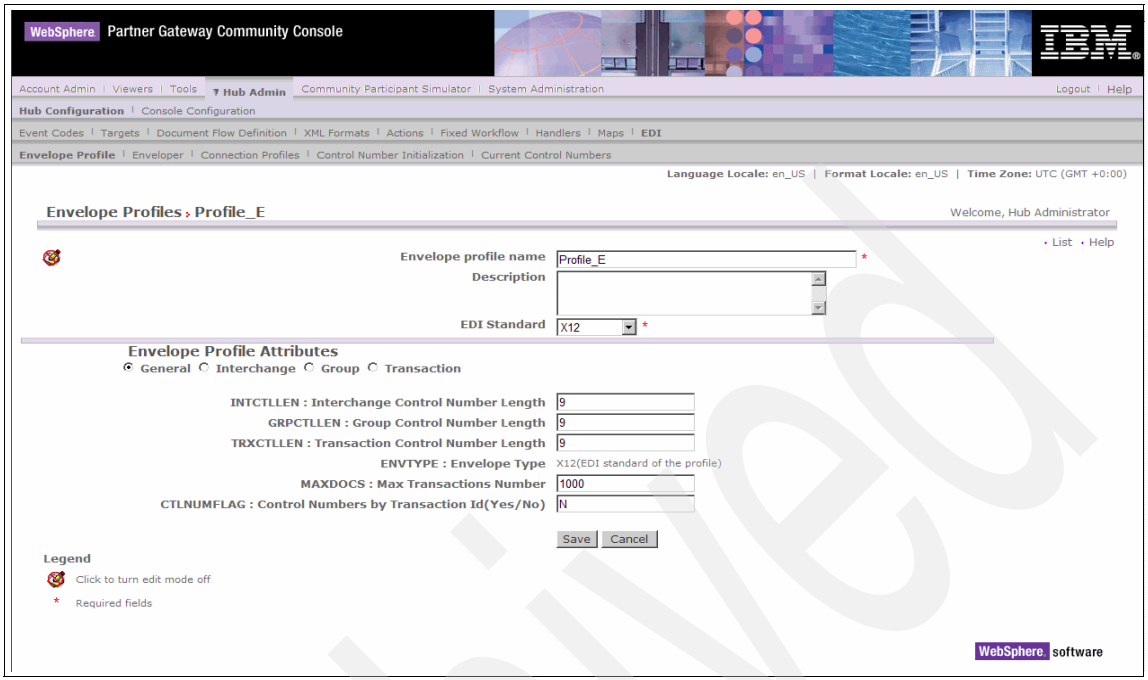


Figure 19-22 General Attributes for the Envelope Profile Profile_E

5. Select **Interchange** under the Envelope Profile Attributes and use values in Table 19-2 to complete the form.

Table 19-2 Interchange attribute values

Attribute	Value
ISA01	01
ISA02	ISA002
ISA03	02
ISA04	ISA004
ISA05	----
ISA06	----
ISA07	----
ISA08	----
ISA09	----

Archived

Attribute	Value
ISA10	----
ISA11	U
ISA12	02010
ISA13	----
ISA14	N
ISA15	T
ISA16	----

Note: All the attributes that are able to be filled in manually are optional. All the attributes that are mandatory such as Sender and Receiver IDs will be automatically filled in by Document Manager of WebSphere Partner Gateway in the run time. For the more detailed information about these attributes, refer to the product documentation.

Archived

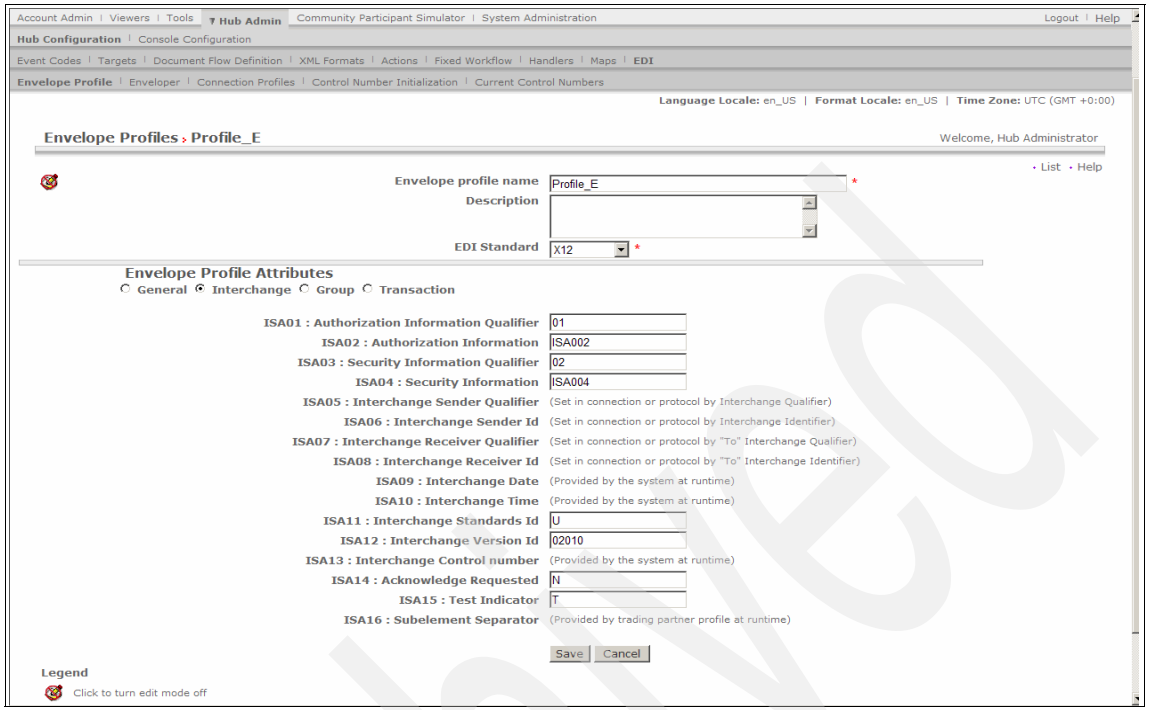


Figure 19-23 Interchange attributes for the Envelope Profile Profile_E

6. Select **Group** and fill in the form with the values shown in Table 19-3.
   a. Because the outgoing EDI transaction is a purchase order, select the GS01-functional Group ID as `PO` (see Figure 19-24 on page 655).
   b. Fill in the values AppSend, AppRecv for Application Sender ID, Application Receiver ID which are `GS02` and `GS03` respectively (Figure 19-24 on page 655).
   c. Group Version(GS08) can be filled in as `002040`.

Table 19-3 Group level attributes for the envelope profile Profile_E

Attribute	Value
GS01	PO
GS02	AppSend
GS03	APPRecv
GS04	----
GS05	----

Archived

Attribute	Value
GS06	----
GS07	----
GS08	002040

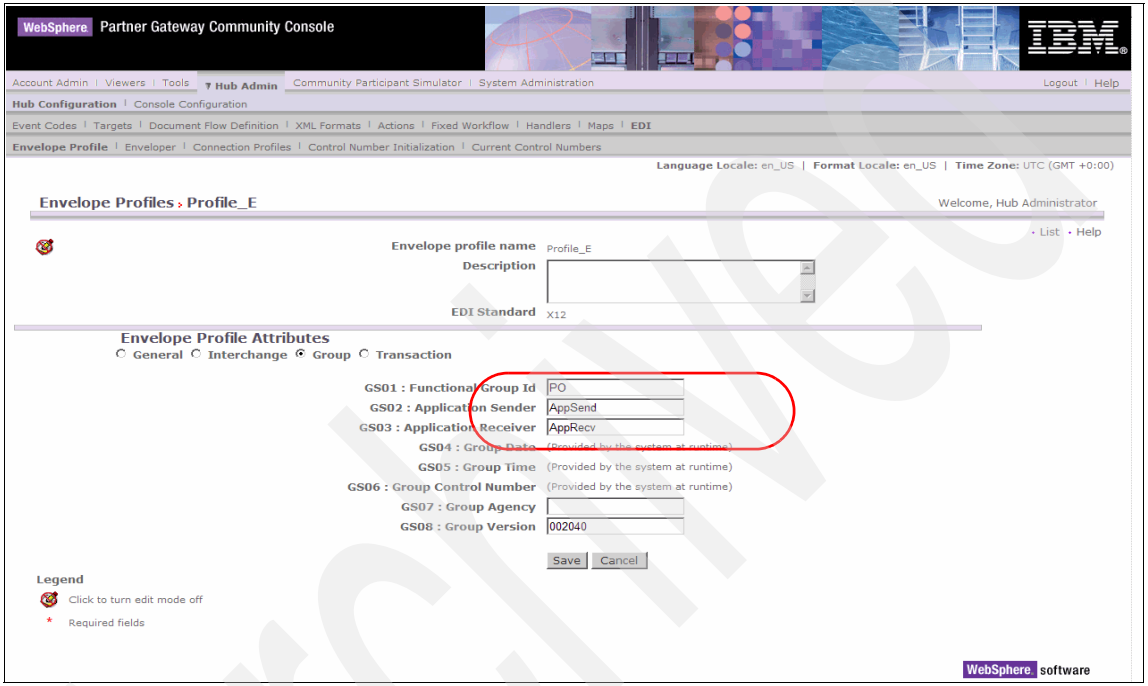


Figure 19-24 Group level attributes for the envelope profile Profile_E

7. Select **Save** to store the envelope profile configuration.
8. The envelope profile created can be seen under **Hub Admin** → **Hub Configuration** → **EDI** → **Envelope Profile** (Figure 19-25 on page 656).

Archived

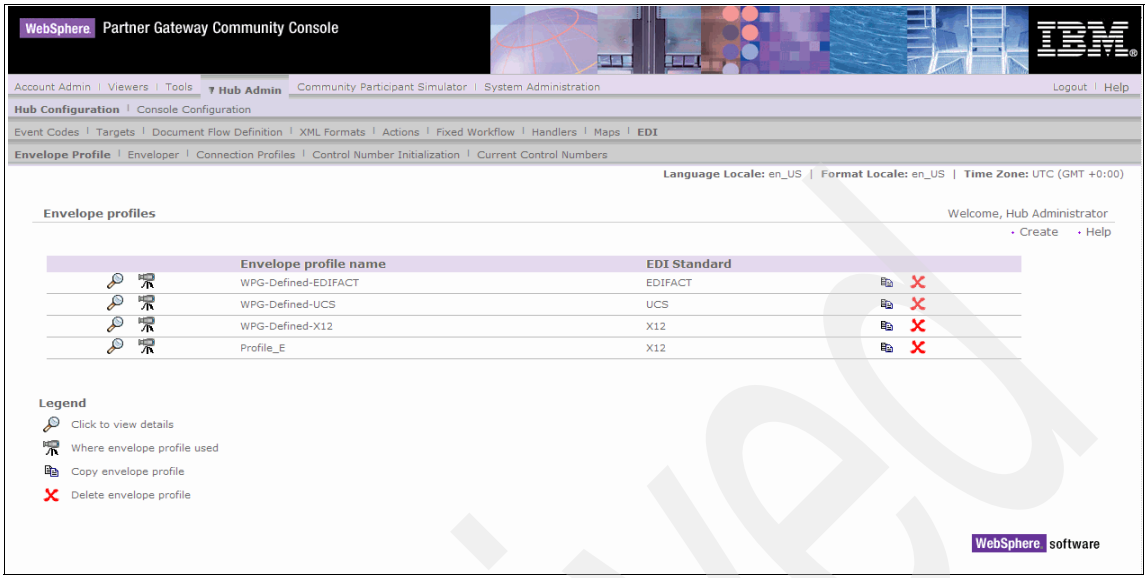


Figure 19-25 List of Envelope Profiles

Setting the enveloper scheduling interval

The enveloper scheduling interval is the period of time which governs the invocation of the enveloper. For example, if the interval is set as `60 seconds`, the enveloper is invoked every 60 seconds.

While Logged in as hubadmin, follow these steps:

1. Select **Hub Admin** → **Hub Configuration** → **EDI** → **Enveloper.**
2. Edit the enveloper configuration. Make sure that Interval based scheduling is selected and change the interval to `90 seconds` (Figure 19-26 on page 657).

Archived

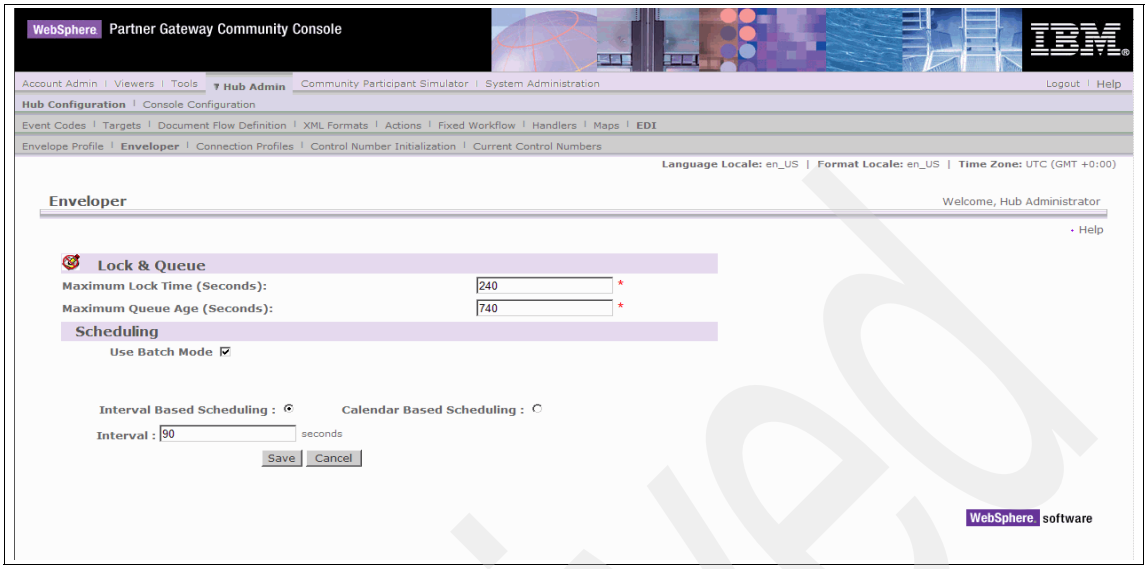


Figure 19-26 Setting the enveloper scheduling Interval

3. Select **Save** to save the configuration.

19.4.11 Activate the participant connections and attribute selection

The final step in the configuration is the creation of participant connections. While logged in as hubadmin, follow these steps:

1. Select **Account Admin** → **participant connections.**
2. Select **Company E** as Source Participant and **Company A** as the Target Participant. Select **Search**
3. Activate the following Connections by clicking **Activate**.
   – None/CUSTOM_XML/PURCHASR_ORDER to N/A/X12V4R1/850 (see Figure 19-27).

Figure 19-27 Enabling the participant connection from XML to EDI

   – N/A/EDI-X12/ISA to AS/EDI-X12/ISA (Figure 19-28 on page 658).

Archived

Figure 19-28 Enabling the participant connection responsible for enveloping

Some of the EDI attributes cannot be set in the envelope profile and can be set only at the protocol level in the document flow definitions or at the Connection level.In this example, we set these attributes at the Connection level.

1. Select **Attributes** on the Target side for the participant connection from None/CUSTOM_XML/PURCHASE_ORDER to N/A/X12V4R1/850 (Figure 19-29).

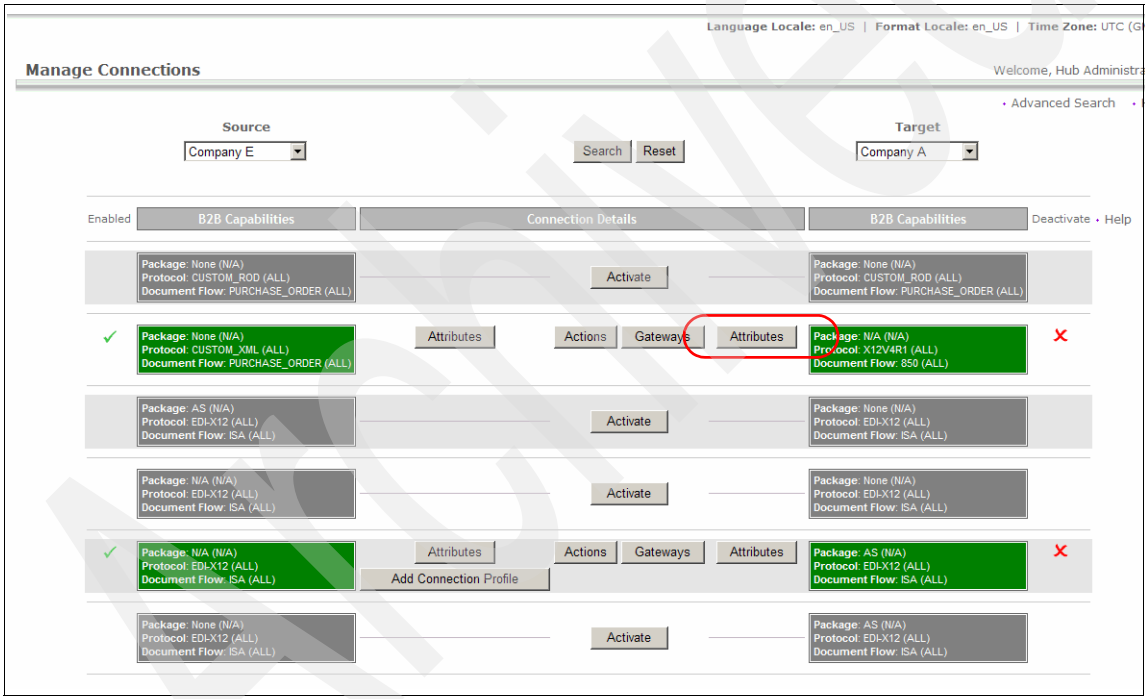


Figure 19-29 Selecting the Target side attributes

2. In the next page, we see the summary of the attributes that are activated. Select the **folder** icon next to the Protocol **X12V4R1** (Figure 19-30 on page 659).

Max validation error level	Max validation error level	0	Inherited from: **Scope:** Global **Type:** Max validation error level
Validation level	Validation level	0	Inherited from: **Scope:** Global **Type:** Validation level
Char set validation table	Char set validation table	CHARSET	Inherited from: **Scope:** Global **Type:** Char set validation table
Alphanumeric validation table	Alphanumeric validation table	ALPHANUM	Inherited from: **Scope:** Global **Type:** Alphanumeric validation table
Generate group level info only in functional Ack	Generate group level info only in functional Ack	No	Inherited from: **Scope:** Global **Type:** Generate group level info only in functional Ack
Century control year	Century control year	10	Inherited from: **Scope:** Global **Type:** Century control year
Detailed validation of segment	Detailed validation of segment	No	Inherited from: **Scope:** Global **Type:** Detailed validation of segment
Connection Profile Qualifier1	Connection Profile Qualifier1		No value provided
Interchange qualifier	Interchange qualifier		No value provided
Interchange Identifier	Interchange Identifier		No value provided
Interchange usage indicator	Interchange usage indicator		No value provided
Group application sender identifier	Group application sender identifier		No value provided
Group application receiver identifier	Group application receiver identifier		No value provided
Interchange reverse routing	Interchange reverse routing		No value provided
Interchange routing address	Interchange routing address		No value provided
Group application sender qualifier	Group application sender qualifier		No value provided
Group application receiver qualifier	Group application receiver qualifier		No value provided
Group application password	Group application password		No value provided
Validation Map	Validation Map		No value provided
XMLNS Active	XMLNS Active	No value provided	No value provided

Package: N/A (N/A)
Protocol: X12V4R1 (ALL)
Document Flow: 850 (ALL)

Legend
Expanded node. Click to collapse
Collapsed node. Click to expand

Figure 19-30 Selecting the Protocol-X12V4R1 in the Target attribute list

3. This leads us to the Attributes page for the protocol X12V4R1. As shown in Figure 19-31 on page 660:
   a. For the attribute Envelope Profile, Select **Profile_E** from the drop-down box.
   b. For the Attribute Interchange Identifier, type the value as `companya`.

Archived

Repeating data element character	Repeating data element character	provided	No value provided	
Decimal notation	Decimal notation	No value provided	No value provided	Select one to update
Segment output	Segment output	Yes	Inherited from: **Scope:** Global **Type:** Segment output	Select one to update
Allow documents with duplicate document ids	Allow documents with duplicate document ids	No	Inherited from: **Scope:** Global **Type:** Allow documents with duplicate document ids	Select one to update
Max error level at Transformation	Max error level at Transformation	0	Inherited from: **Scope:** Global **Type:** Max error level at Transformation	Select one to update
FAMap	FAMap	No value provided	No value provided	Select one to update
Envelope Profile	Envelope Profile	No value provided	No value provided	Profile_E Select one to update WPG-Defined-EDIFACT WPG-Defined-UCS WPG-Defined-X12 Profile_E
Max validation error level	Max validation error level	0	Inherited from: **Scope:** Global **Type:** Max validation error level	
Validation level	Validation level	0	Inherited from: **Scope:** Global **Type:** Validation level	
Char set validation table	Char set validation table	CHARSET	Inherited from: **Scope:** Global **Type:** Char set validation table	
Alphanumeric validation table	Alphanumeric validation table	ALPHANUM	Inherited from: **Scope:** Global **Type:** Alphanumeric validation table	
Generate group level info only in functional Ack	Generate group level info only in functional Ack	No	Inherited from: **Scope:** Global **Type:** Generate group level info only in functional Ack	Select one to update
Century control year	Century control year	10	Inherited from: **Scope:** Global **Type:** Century control year	
Detailed validation of segment	Detailed validation of segment	No	Inherited from: **Scope:** Global **Type:** Detailed validation of segment	Select one to update
Connection Profile Qualifier1	Connection Profile Qualifier1	No value provided	No value provided	
Interchange qualifier	Interchange qualifier	No value provided	No value provided	
Interchange Identifier	Interchange Identifier	No value provided	No value provided	companya
Interchange usage indicator	Interchange usage indicator	No value provided	No value provided	
Group application sender identifier	Group application sender identifier	No value provided	No value provided	
Group application receiver identifier	Group application receiver identifier	No value provided	No value provided	
Interchange reverse routing	Interchange reverse routing	No value provided	No value provided	
Interchange routing address	Interchange routing address	No value provided	No value provided	

Figure 19-31 Setting the Attributes on the Target side.

c. Select **Save** to save the attributes.

d. Select **Return** to the Manage Connections Page.

4. Select **Attributes** on the source side for the participant connection None/CUSTOM_XML/PURCHASE_ORDER to N/A/X12V4R1/850. See Figure 19-32 on page 661.

Archived

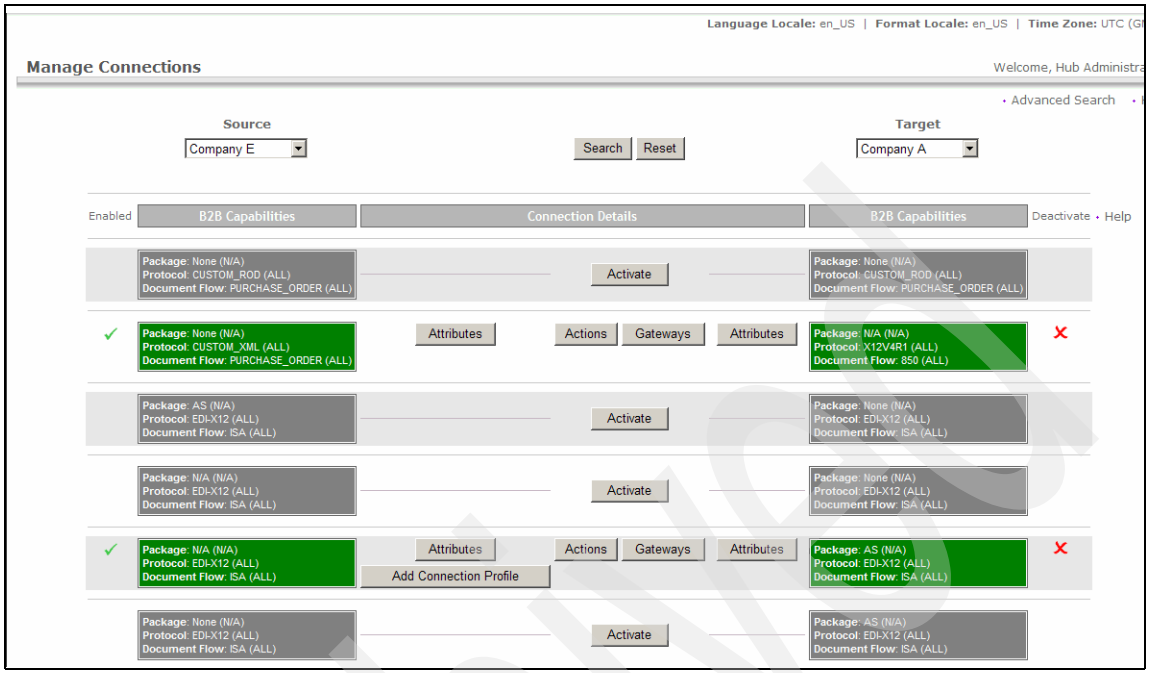


Figure 19-32 Selecting the source side Attributes

5. In the next screen, we see the summary of the attributes that are activated. Select the **folder** icon next to the Protocol **CUSTOM_XML**. See Figure 19-33 on page 662.

Archived

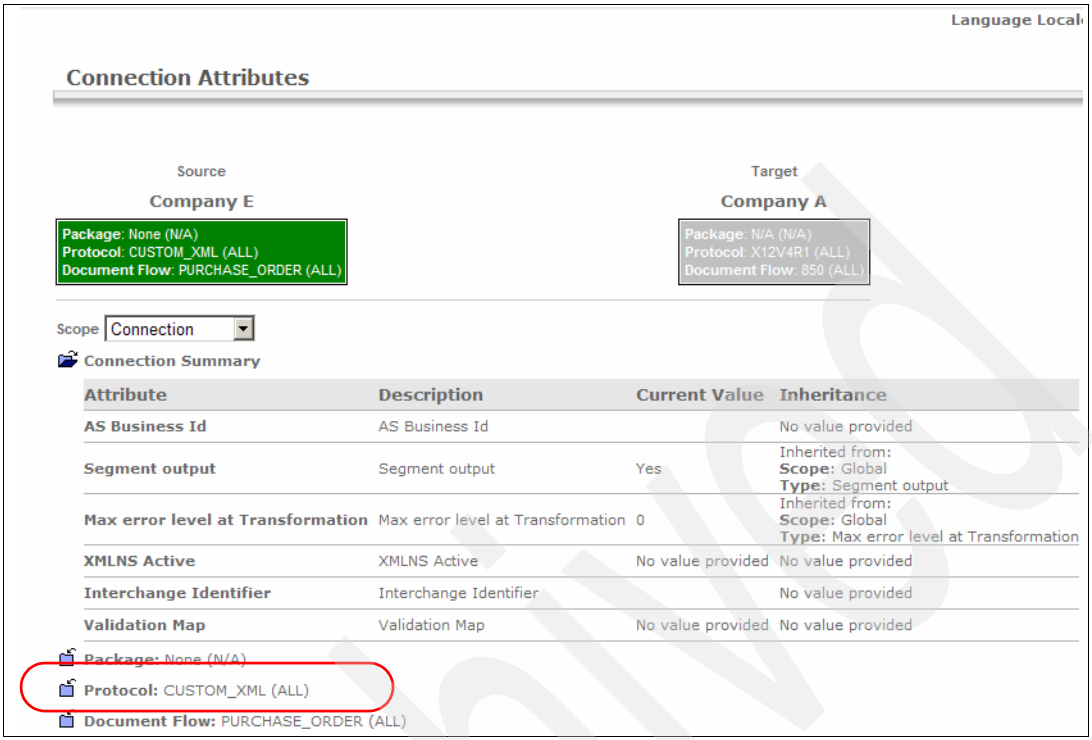


Figure 19-33 Selecting the attributes at the Protocol level for the XML protocol CUSTOM_XML

6. This leads us to the Attributes page for the protocol **CUSTOM_XML**. For the attribute Interchange Identifier, enter the value as `companye`, as shown in Figure 19-34 on page 663.

Archived

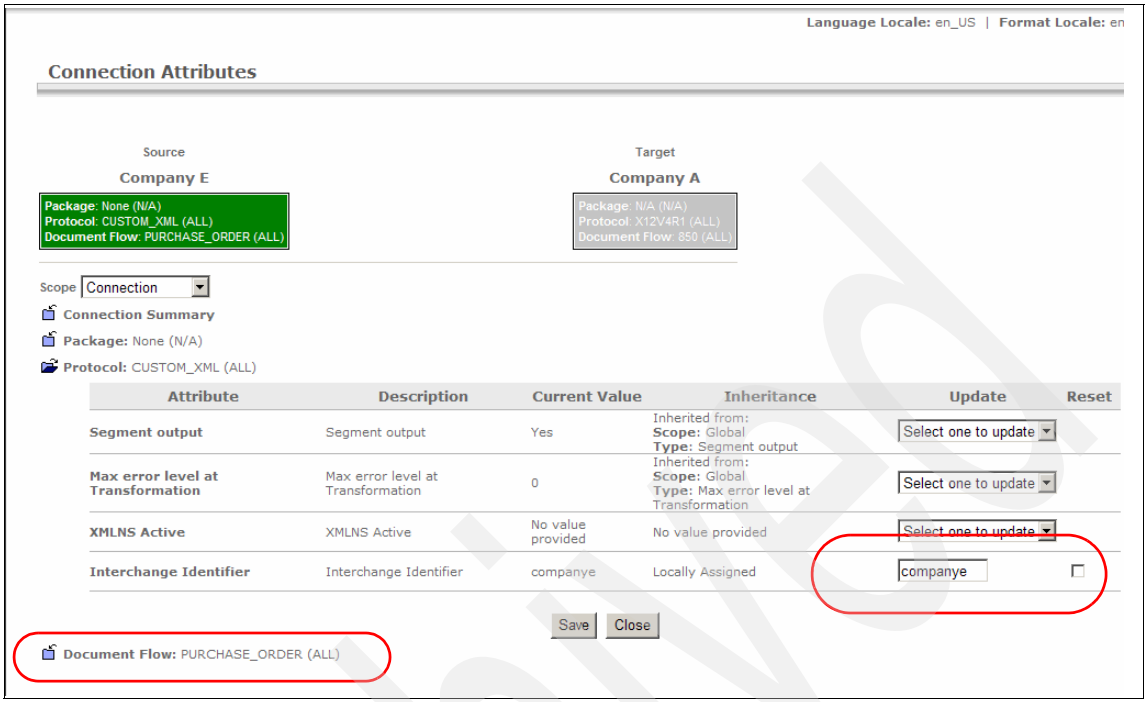


Figure 19-34 Setting the Attribute Interchange identifier at the Protocol level

7. Select **Save** to save the configuration. Select the **folder** icon next to Document flow **PURCHASE_ORDER** to set the attribute Validation Map (see Figure 19-34).
8. In the next screen, select the attribute Validation Map as **purchaseOrder.xsd.** (Figure 19-35 on page 664)
9. Select **Save** to save the configuration and select **Return** to return to the Manage participant connections page.

Archived

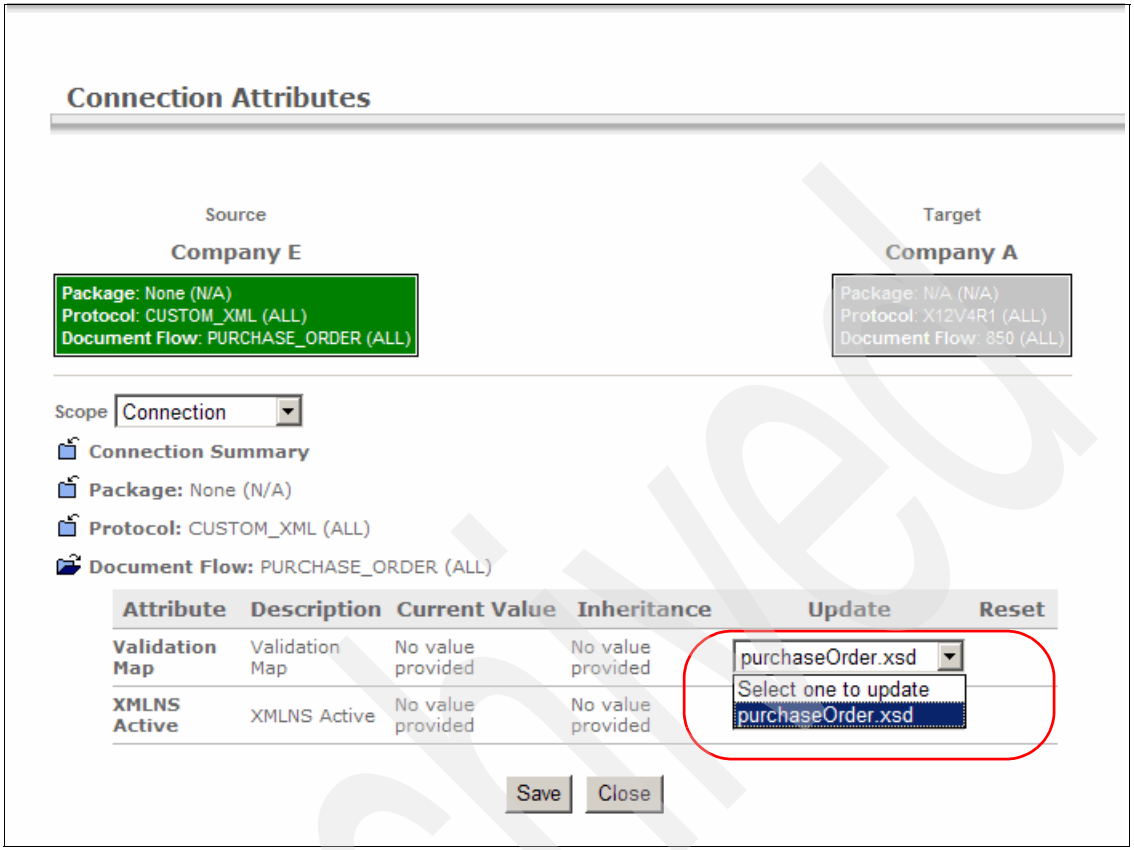


Figure 19-35 Setting the Attribute Validation Map at the Document flow level

10. Some of the attributes related to AS packaging need to be set on the N/A/EDI-X12/ISA to AS/EDI-X12/ISA participant connection.
11. Select the target side **Attributes** on this connection.
12. In the screen, we see the summary of the attributes for the document flow definition AS/EDI-X12/ISA.
13. Select the **folder** icon next to the Package **AS (s**ee Figure 19-36 on page 665**)**.

Archived

Time To Acknowledge	Time To Acknowledge	30	Inherited from: **Scope:** Global **Type:** Time To Acknowledge
Retry Count	Retry Count	3	Inherited from: **Scope:** Global **Type:** Retry Count
AS Compress Before Sign	AS Compress Before Sign	Yes	Inherited from: **Scope:** Global **Type:** AS Compress Before Sign
AS Compressed	AS Compressed	No	Inherited from: **Scope:** Global **Type:** AS Compressed
AS Encrypted	AS Encrypted	No	Inherited from: **Scope:** Global **Type:** AS Encrypted
AS MDN Http Url	AS MDN Http Url		No value provided
AS MDN Email Address	AS MDN Email Address		No value provided
AS MDN Asynchronous	AS MDN Asynchronous	Yes	Inherited from: **Scope:** Global **Type:** AS MDN Asynchronous
AS MDN Requested	AS MDN Requested	Yes	Inherited from: **Scope:** Global **Type:** AS MDN Requested
AS Message Digest Algorithm	AS Message Digest Algorithm	sha1	Inherited from: **Scope:** Global **Type:** AS Message Digest Algorithm
AS MDN Signed	AS MDN Signed	No	Inherited from: **Scope:** Global **Type:** AS MDN Signed
AS Signed	AS Signed	No	Inherited from: **Scope:** Global **Type:** AS Signed
AS Business Id	AS Business Id		No value provided
Allow documents with duplicate document ids	Allow documents with duplicate document ids	No	Inherited from: **Scope:** Global **Type:** Allow documents with duplicate document ids
Allow a TA1 request	Allow a TA1 request	Yes	Inherited from: **Scope:** Global **Type:** Allow a TA1 request
Discard Envelope if any errors	Discard Envelope if any errors	No	Inherited from: **Scope:** Global **Type:** Discard Envelope if any errors

Package: AS (N/A)
Protocol: EDI-X12 (ALL)
Document Flow: ISA (ALL)

Figure 19-36 Selecting the attributes for the Package AS

11. Set the attribute **AS MDN Requested** to **No** and select **Save.** See Figure 19-37 on page 666.

Archived

Package: AS (N/A)

Attribute	Description	Current Value	Inheritance	Update
Time To Acknowledge	Time To Acknowledge	30	Inherited from: Scope: Global Type: Time To Acknowledge	
Retry Count	Retry Count	3	Inherited from: Scope: Global Type: Retry Count	
AS Compress Before Sign	AS Compress Before Sign	Yes	Inherited from: Scope: Global Type: AS Compress Before Sign	Select one to update
AS Compressed	AS Compressed	No	Inherited from: Scope: Global Type: AS Compressed	Select one to update
AS Encrypted	AS Encrypted	No	Inherited from: Scope: Global Type: AS Encrypted	Select one to update
AS MDN Http Url	AS MDN Http Url	No value provided	No value provided	
AS MDN Email Address	AS MDN Email Address	No value provided	No value provided	
AS MDN Asynchronous	AS MDN Asynchronous	Yes	Inherited from: Scope: Global Type: AS MDN Asynchronous	Select one to update
AS MDN Requested	AS MDN Requested	Yes	Inherited from: Scope: Global Type: AS MDN Requested	No Select one to update Yes No
AS Message Digest Algorithm	AS Message Digest Algorithm	sha1	Inherited from: Scope: Global Type: AS Message Digest Algorithm	
AS MDN Signed	AS MDN Signed	No	Inherited from: Scope: Global Type: AS MDN Signed	Select one to update
AS Signed	AS Signed	No	Inherited from: Scope: Global Type: AS Signed	Select one to update
AS Business Id	AS Business Id	No value provided	No value provided	Select one to update

Save Close

Protocol: EDI-X12 (ALL)

Document Flow: ISA (ALL)

Figure 19-37 Select MDN requirements

This finishes the configuration of participant connections. We have now completed the additional configuration of WebSphere Partner Gateway Enterprise running on Company E server.

In the next section, we perform the configuration required for Company A to receive the EDI-X12 850 purchase order.

19.5 Configure Company A to receive EDI documents

No configuration updates are required for WebSphere Partner Gateway Advancedrunning on the Company A server because it is already configured to receive EDI-X12 interchanges. We do need to verify that this is the case:

1. Log in to WebSphere Partner Gateway Advanced on Company A as hubadmin.
2. Select **Account Admin** → **participant connections.**
3. In the Manage Connections screen, select the Source as **Company E** and the Target as **Company A**.
4. Select **Search**.
5. We see that the channel AS/EDI-X12/ISA to None/EDI-X12/ISA which is required to post the EDI-X12 envelope to Company E is already present.

Figure 19-38 From Company E to Company A's WebSphere Partner Gateway Advanced

In the next section, we validate our enveloping scenario.

19.6 Sending and validating the communication

Our document routing and validating of the communication really needs to be divided into two scenarios depending upon the enveloper scheduling interval.

In the case of XML to EDI scenarios, the enveloping scheduling interval is a critical determining factor as to how many EDI transactions will be enveloped into a single interchange.

For example, if multiple XML documents are posted to WebSphere Partner Gateway Receiver and if the enveloper scheduling is set to 600 seconds (10 minutes), there is good possibility that all the translated XML documents are going to be enveloped into a single EDI envelope.

However, if the enveloper scheduling interval is set as 30 seconds, we would probably be seeing multiple EDI envelopes in the document viewers.

So, we will use that as the basis for splitting our validation. We can call them the *short* interval and the *long* interval scenarios.

1. In the short interval validation:

Archived

a. A single XML document will be placed in the JMS Target of Company E.
b. It will be transformed.
c. It will be enveloped.
d. It will be routed to Company A.

For this validation, the enveloper scheduling interval will be set to 90 seconds as in 19.4.10, “Configure the enveloper” on page 649.

2. In the long interval validation:

   a. Three XML documents will be placed in the JMS Target
   b. They will be transformed.
   c. They will be enveloped.
   d. They will be routed to Company A.

   The enveloper scheduling interval will be set to 180 seconds, we expect to see all three documents enveloped into a single EDI interchange.

19.6.1 Short interval validation

We can now use our sample XML file to emulate sending the XML Purchase Order from a back-end system of Company E. First, we will do the short interval validation test:

1. Start RFHUTIL, which can be found in the Additional Materials (Figure 19-39 on page 669).
2. Enter the queue manager name and queue name that is being used for the JMSTarget (partner_e.bcg.queue.manager and EDI_IN).
3. Select **Read File**.
4. Browse the Additional Materials for the file POSample.xml; open this file.

Archived

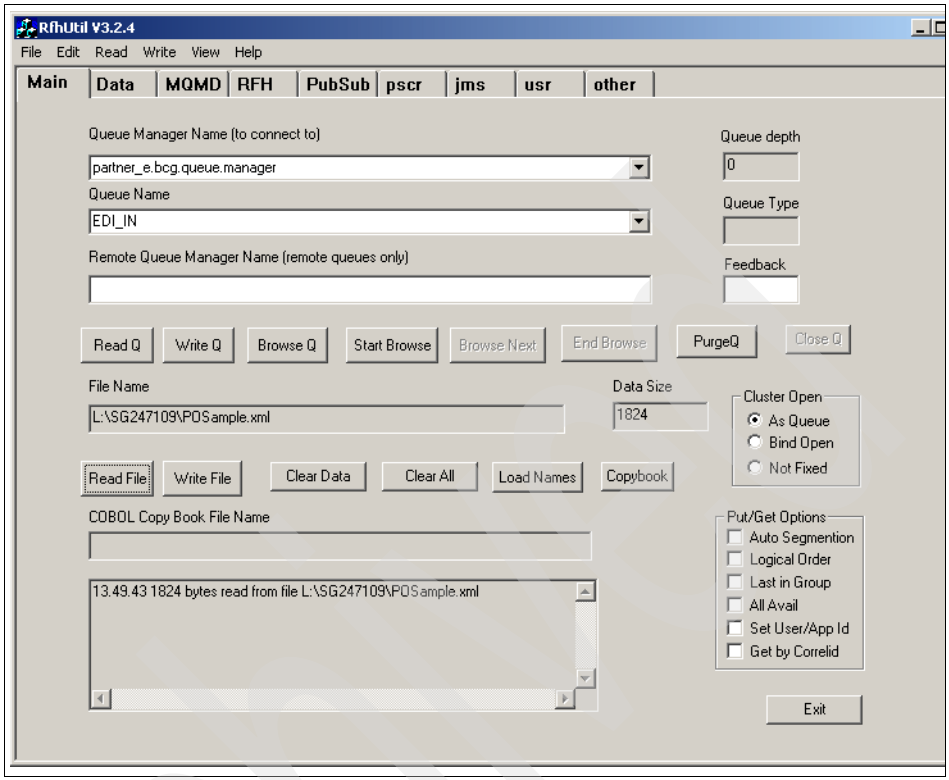


Figure 19-39 Select queue manager and queue

5. Select the **Data** tab and the radio button for **XML**. See Figure 19-40 on page 670.

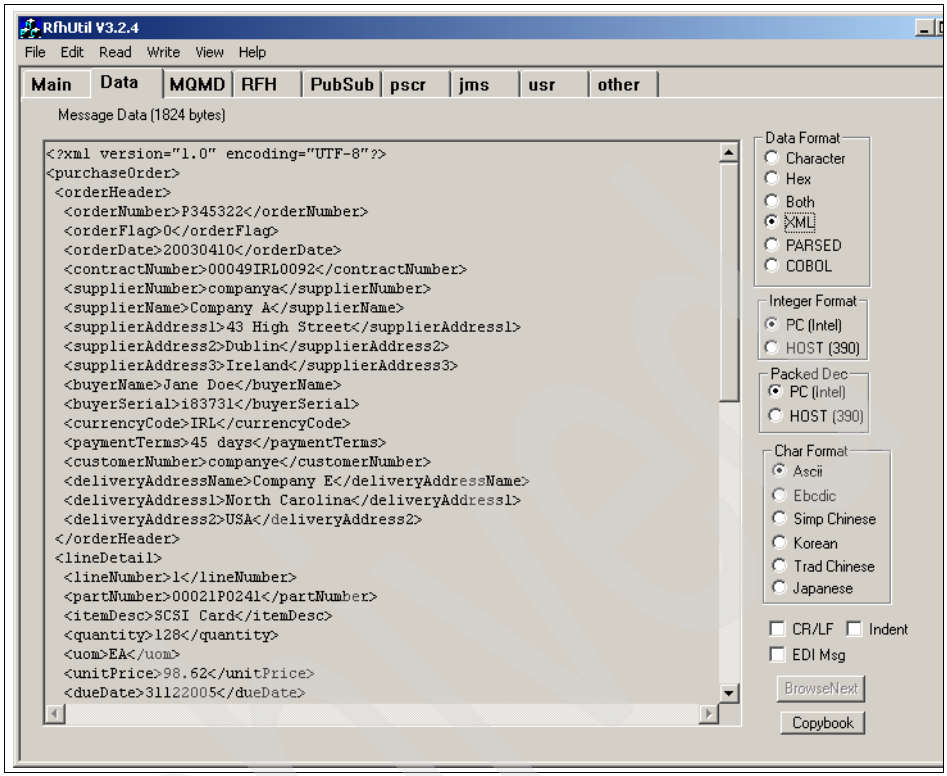


Figure 19-40 View data

6. Select the **MQMD** tab and set the MQ Message Format to **MQSTR.**

Note: Because we are not using Back-end Integration as the packaging type, we can simply send this message as Payload Only, that is the XML data itself (string data), the MQMD format of string). There is no requirement for us to provide JMS header information such as the details in the usr folder.

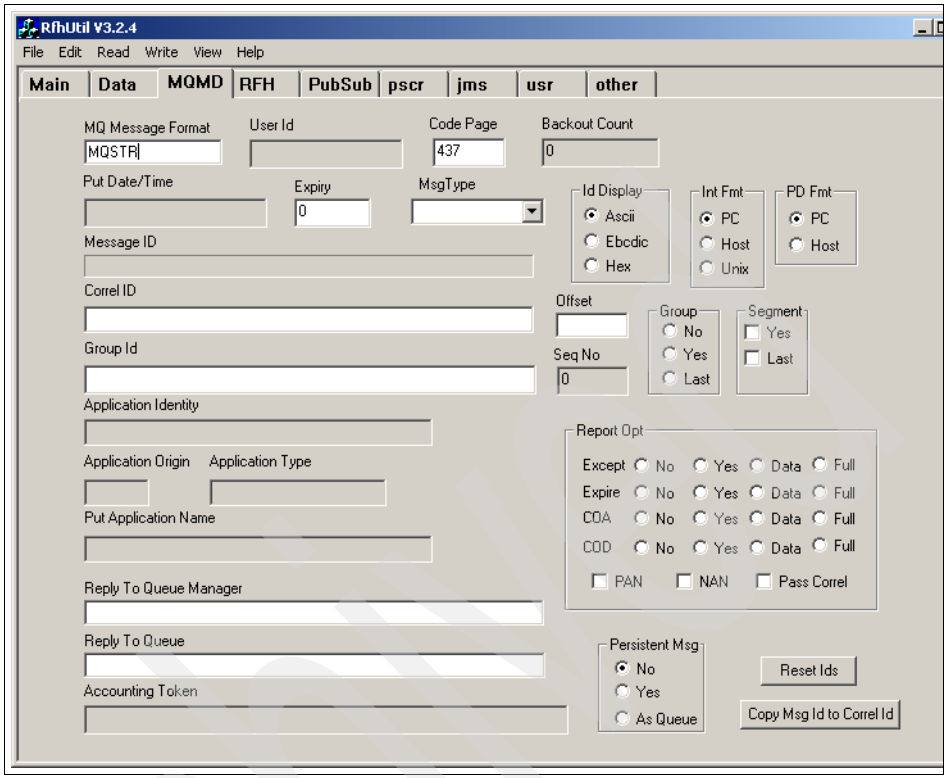


Figure 19-41 Set message descriptor

7. Select the **Main** (Figure 19-41) tab and select the **Write Q** button. This will write the XML data to the EDI_IN queue.
8. Go back to the console and select **Viewers** → **Document Viewers**. Search for the documents from Company E to Company A. We see entries similar to those shown in Figure 19-42.

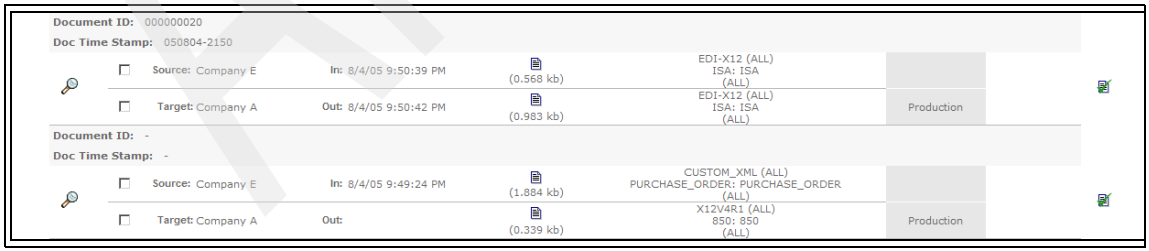


Figure 19-42 XML to EDI enveloped transaction

9. Using the Raw Document Viewer for the enveloped document, we see all the fields that are entered in the envelope profile (Figure 19-43).

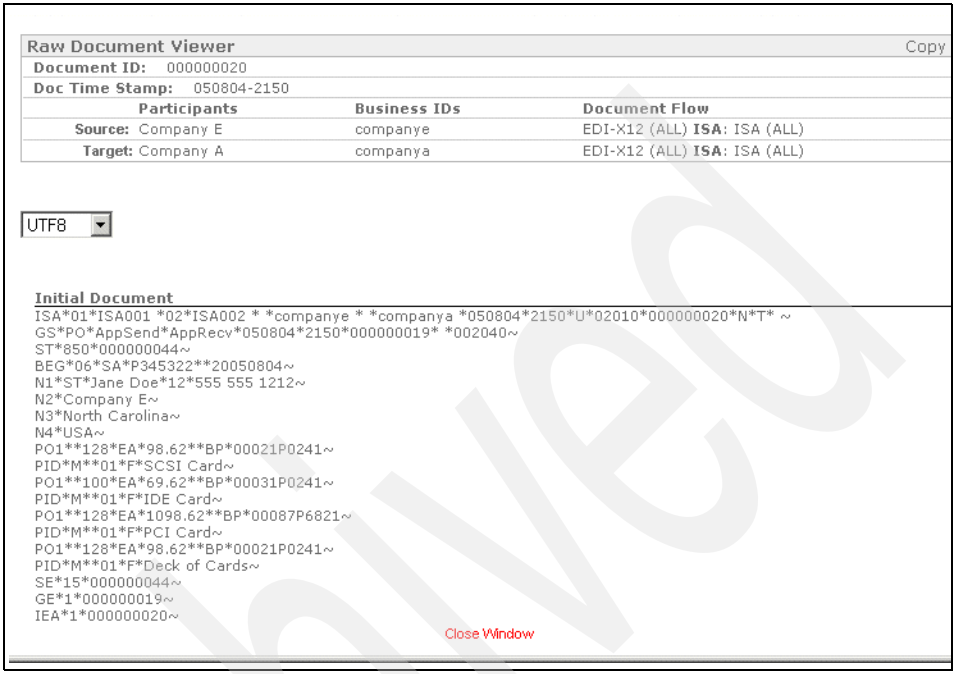


Figure 19-43 Raw document view of the enveloped document

19.6.2 Long interval validation

To validate long intervals, follow these steps:

1. Set the enveloper interval scheduling interval as `180 seconds` (Figure 19-44 on page 673).

Archived

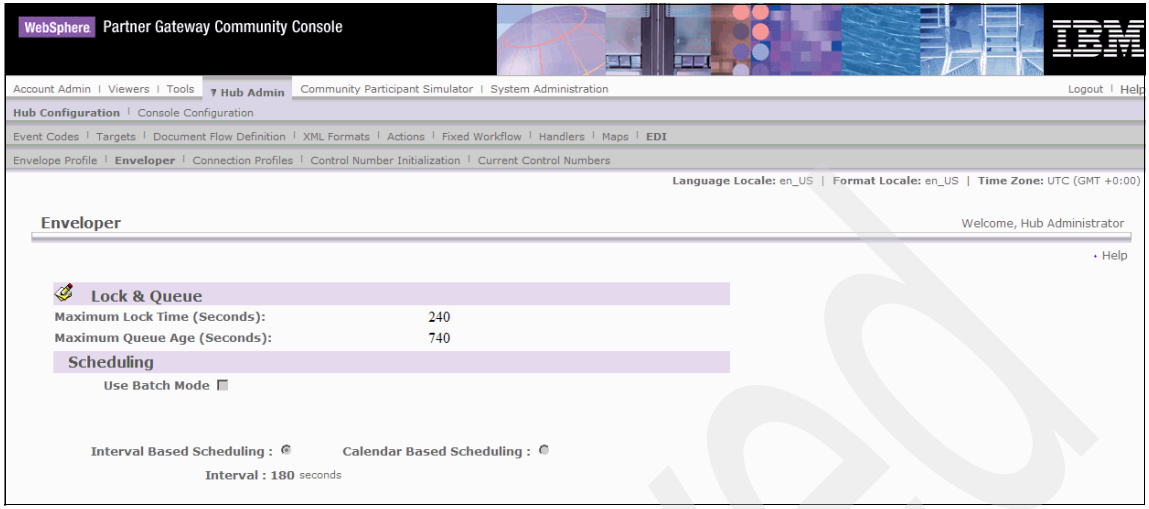


Figure 19-44 Setting the enveloper scheduling interval to 180 seconds

2. Using RFHUTIL, place three XML documents on the queue in relatively quick succession.
3. With the scheduling interval set to 180 seconds, all three translated XML EDI Transactions will enveloped into a single EDI Interchange. See Figure 19-45 on page 674.

Archived

Resend

	Participants		Time Stamps		Protocol/Document Flow	Gateway Type	Synchronous	Status
Document ID: 000000025								
Doc Time Stamp: 050805-1730								
Source:	Company E	In:	8/5/05 5:30:24 PM	(1.318 kb)	EDI-X12 (ALL) ISA: ISA (ALL)			
Target:	Company A	Out:	8/5/05 5:30:26 PM	(1.734 kb)	EDI-X12 (ALL) ISA: ISA (ALL)	Production		
Document ID: -								
Doc Time Stamp: -								
Source:	Company E	In:	8/5/05 5:27:35 PM	(1.884 kb)	CUSTOM_XML (ALL) PURCHASE_ORDER: PURCHASE_ORDER (ALL)			
Target:	Company A	Out:		(0.339 kb)	X12V4R1 (ALL) 850: 850 (ALL)	Production		
Document ID: -								
Doc Time Stamp: -								
Source:	Company E	In:	8/5/05 5:27:34 PM	(1.884 kb)	CUSTOM_XML (ALL) PURCHASE_ORDER: PURCHASE_ORDER (ALL)			
Target:	Company A	Out:		(0.339 kb)	X12V4R1 (ALL) 850: 850 (ALL)	Production		
Document ID: -								
Doc Time Stamp: -								
Source:	Company E	In:	8/5/05 5:27:33 PM	(1.884 kb)	CUSTOM_XML (ALL) PURCHASE_ORDER: PURCHASE_ORDER (ALL)			
Target:	Company A	Out:		(0.339 kb)	X12V4R1 (ALL) 850: 850 (ALL)	Production		
Document ID: 000000024								

Figure 19-45 Multiple XML documents enveloped

This finishes the enveloping scenario in which the inbound XML Purchase Order document has been transformed into an EDI-X12 Purchase Order envelope. In the next sections, the necessary configuration required to execute the de-enveloping scenario is explained in detail.

19.7 Implement the de-enveloping scenario

This section deals with the necessary configuration required for our de-enveloping part of the scenario.

19.7.1 Configure Company A to post an EDI-X12 interchange

No more configuration updates are required for WebSphere Partner Gateway Advanced on the Company A server because the necessary channel to post the document to WebSphere Partner Gateway Enterprise on Company E server has already been configured. We can verify this with these steps:

1. Log in to WebSphere Partner Gateway on Company A as hubadmin.
2. Select **Account Admin** → **Participant Connections.**

3. In the Manage Connections screen, select the Source as **Company A** and the Target as **Company E**. Select **Search**.
4. We see that the channel None/EDI-X12/ISA to AS/EDI-X12/ISA which is required to post the EDI-X12 Envelope to Company E is already present (Figure 19-46).

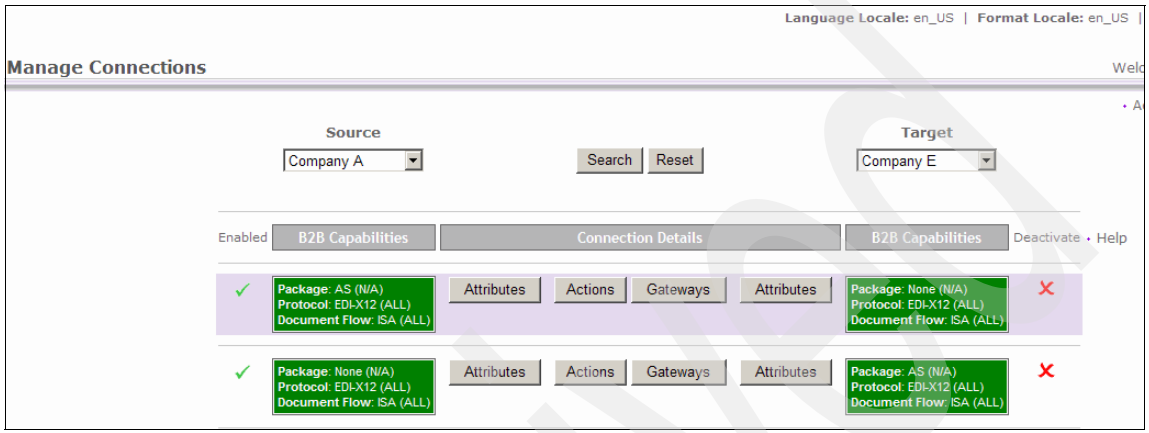


Figure 19-46 participant connections from Company A to Company E.

We now move on to the configuration updates required to WebSphere Partner Gateway on Company E.

19.7.2 Uploading the EDI to XML transformation map

To upload the map control string EDI_XML_PO-ack.eif which was placed in a file directory, the following steps need to be followed:

1. Copy the map XML_EDI_PO-ack.eif from the file system to WebSphere Partner Gateway Home\bin folder
2. Use the **bcgDISImport.bat** utility to import the maps to WebSphere Partner Gateway database as shown in Example 19-8.

Example 19-8 Upload the control string EDI_XML_PO-ACK

```
D:\WPG6.0\bcghub\bin>bcgDISImport.bat   bcgcon bcgcon EDI_XML_PO-ACK.eif
ECHO is off.
JDBC_URL = jdbc:db2://wpgv6db.itso.ral.ibm.com:50000/bcgapps
JDBC_DRIVER_CLASS = com.ibm.db2.jcc.DB2Driver
JDBC_QUAL = DB2ADMIN
USERID = bcgcon
IMPORT_FILE = EDI_XML_PO-ACK.eif
JAVA_HOME = D:\was\java
ECHO is off.
```

```
Connection to database made. Starting import process.
Control string description records...1
Control string records...............23
---------------------------------------------
Total number of records in import file:    24
Import complete.
```

After the Transformation Map has been uploaded, the document flow definitions created by the map can be viewed in the console:

1. Log in to the WebSphere Partner Gateway console on Server E.
2. Select **Hub Admin** → **Hub Configuration** → **Maps** → **Transformation Maps.**
3. We see the EDI_XML_PO-ACK under Manage Transformation Maps Tab (Figure 19-47).

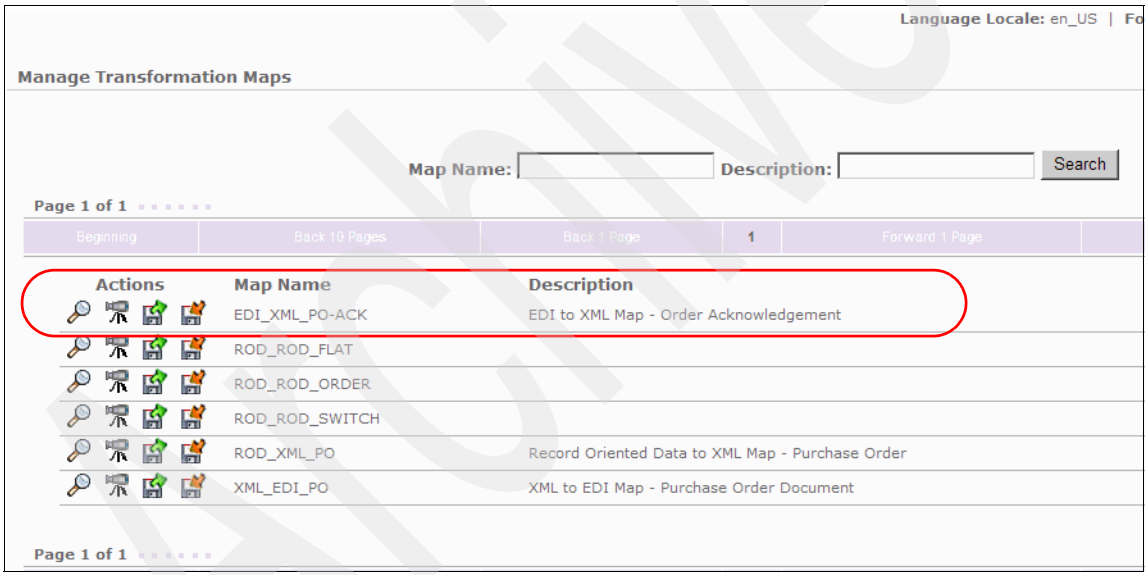


Figure 19-47 Map EDI_XMI_PO-ACK in WebSphere Partner Gateway Console

4. Select the icon next to the map EDI_XML_PO-ACK to view the document flow definitions created by the map import (Figure 19-48 on page 677).
   a. N/A/X12V4R1/855 is the document flow definition that is created on the source side.
   b. None/CUSTOM_XML/PURCHASE_ORDER_ACK is the document flow definition created for the target side.

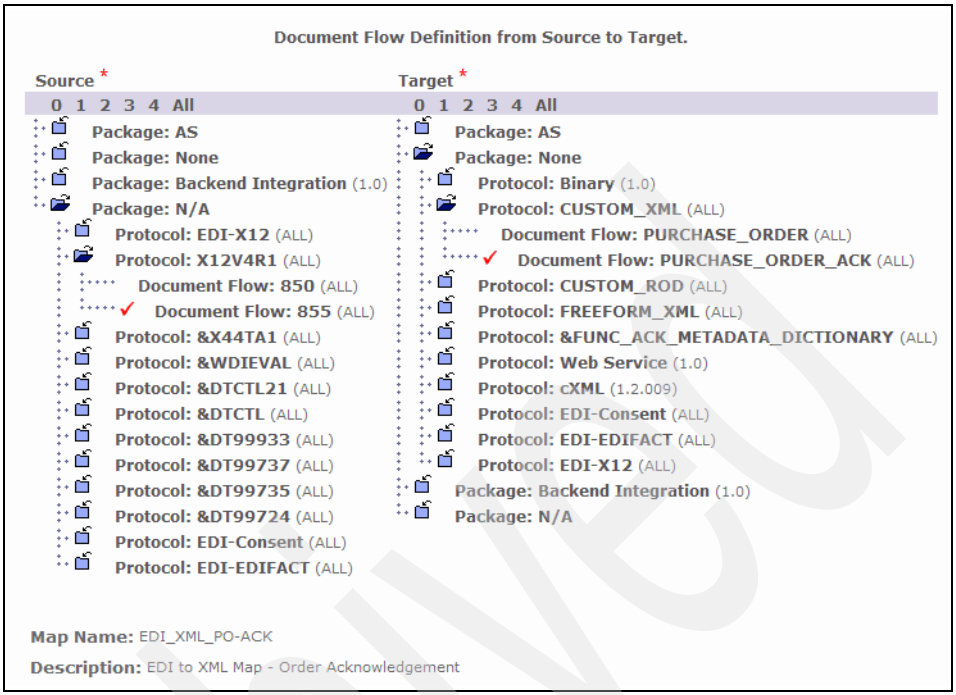


Figure 19-48 Document flow definitions created by the Map EDI_XML_PO-ACK

19.7.3 Upload the EDI validation map

As we saw in Chapter 18, “Mapping for EDI transformation and validation” on page 589, the EDI validation maps are a special class of maps, which do the validation against a specific fields of an EDI document. For example, these fields can be the segments of the EDI-X12 Transaction.

The Validation Maps will be uploaded to WebSphere Partner Gateway through the Data Interchange Services Client Utility.

1. In the Data Interchange Services Client Console, click **FIle** → **Open Import File.**
2. Upload the 355X410_VALIDATE map (as shown in Figure 19-49 on page 678).

   We see the map under the Validation Maps tab.

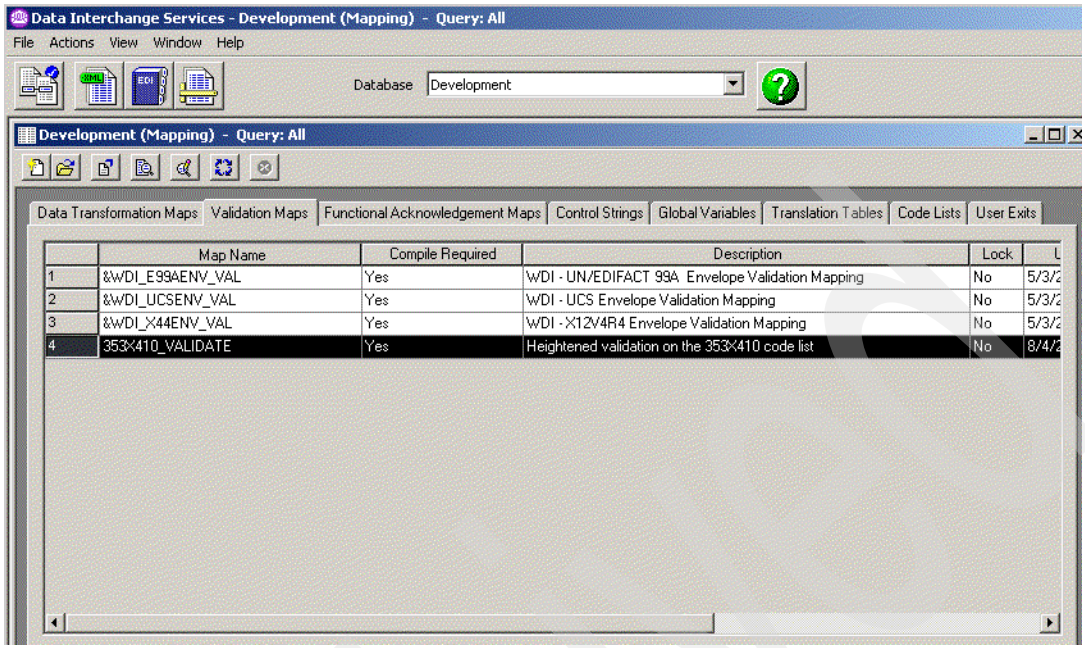


Figure 19-49 Validation map uploaded to Data Interchange Services Client

3. Click **Actions** → **Compile** to compile the map. The Compiled Map can be seen under the Control Strings Tab.
4. Click **Actions** → **Export to Other Database** to upload it to **WPGV6DB**.

   This finishes the task of uploading the EDI Validation Map to WebSphere Partner Gateway
5. You can see the uploaded map under **Hub Admin** → **Hub Configuration** → **Maps** → **Validation Maps** (Figure 19-50 on page 679).

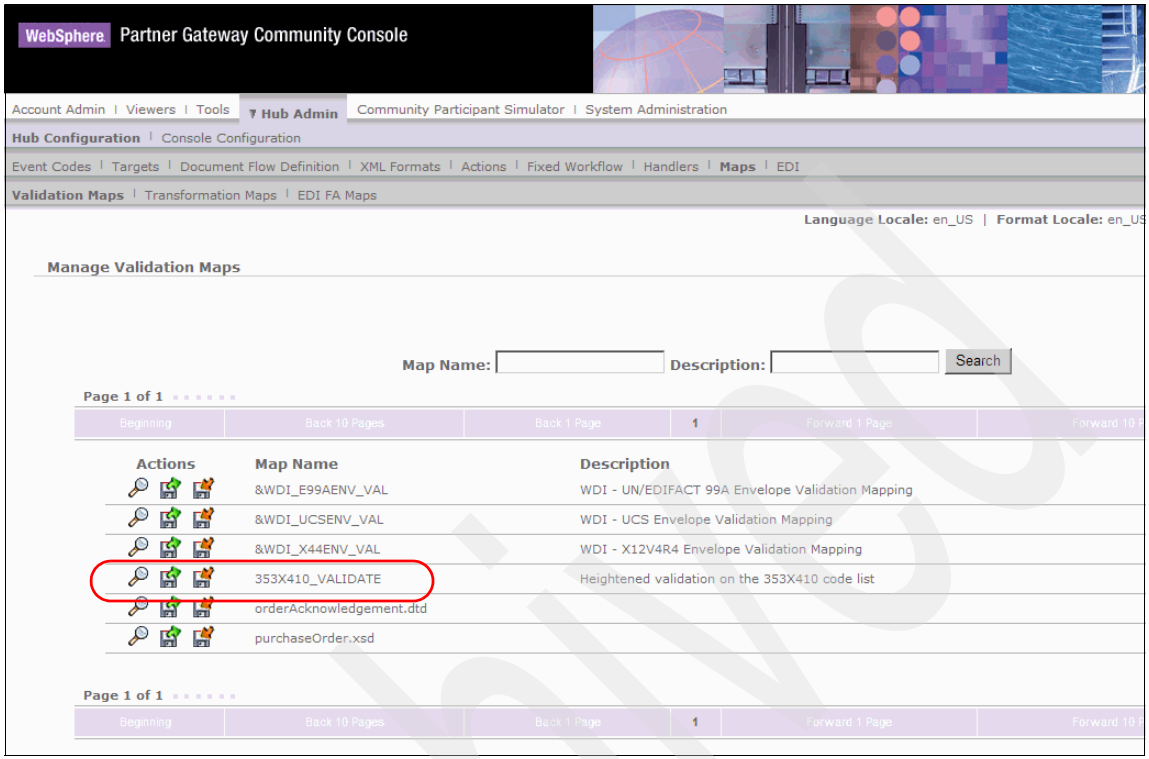


Figure 19-50 Validation map

6. Click the view details icon of the map to see the association between the map and the document flow definition N/A/EDI-X12/ISA (Figure 19-51 on page 680).

Archived

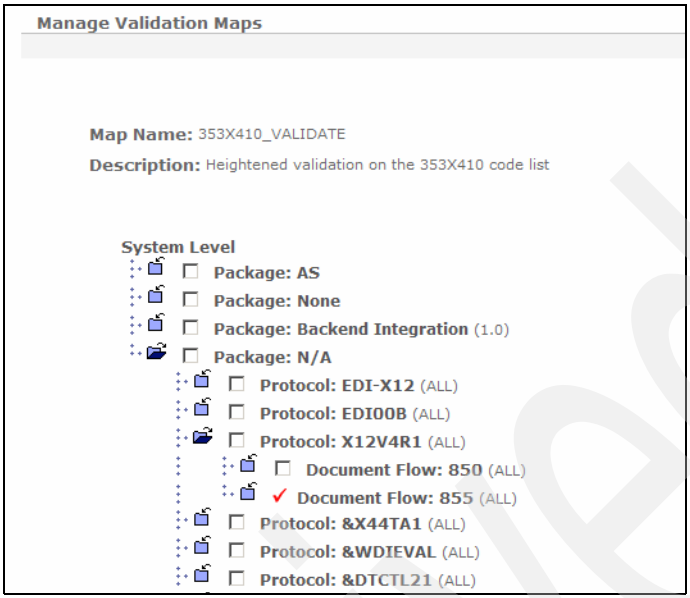


Figure 19-51 Manage validation maps

19.7.4 Create the interactions for the document flow definition

To implement the de-enveloping scenario, two interactions need to be created in WebSphere Partner Gateway.

1. We need an interaction to de-envelope the incoming EDI-X12 envelope. At this step, we will have the de-enveloped EDI-X12 855 transactions.
2. We need to transform the EDI-X12 855 transaction into and equivalent XML Purchase Order Acknowledgement.

While logged on as hubadmin, follow these steps:

1. Select **Hub Admin** → **Hub Configuration** → **Document Flow Definition.**
2. Select **Manage Interactions**.
3. Select **Create Interaction**.
4. Create the interaction from AS/EDI-X12/ISA to N/A/EDI-X12/ISA.
5. Select the action as **EDI De-envelope** from the Action drop-down box (Figure 19-52 on page 681).

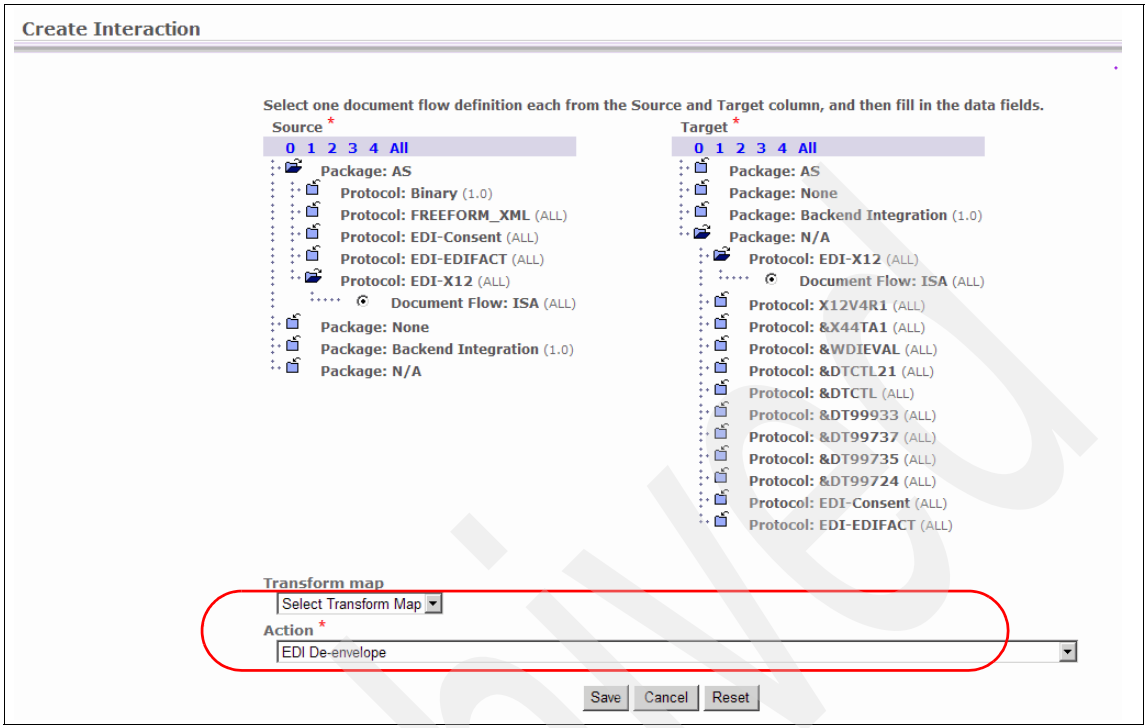


Figure 19-52 Creating the interaction from AS to N/A Packaging for de-enveloping

6. Select **Save.**
7. Create a second interaction for N/A/X12V2R1/855 to None/CUSTOM_XML/PURCHASE_ORDER_ACK.
8. Select **EDI_XMI_PO-ACK** from the Transform Map drop-down box.
9. Select the Action as **EDI Validate, EDI Translate** from the **Action** drop-down box. See Figure 19-53 on page 682.

Archived

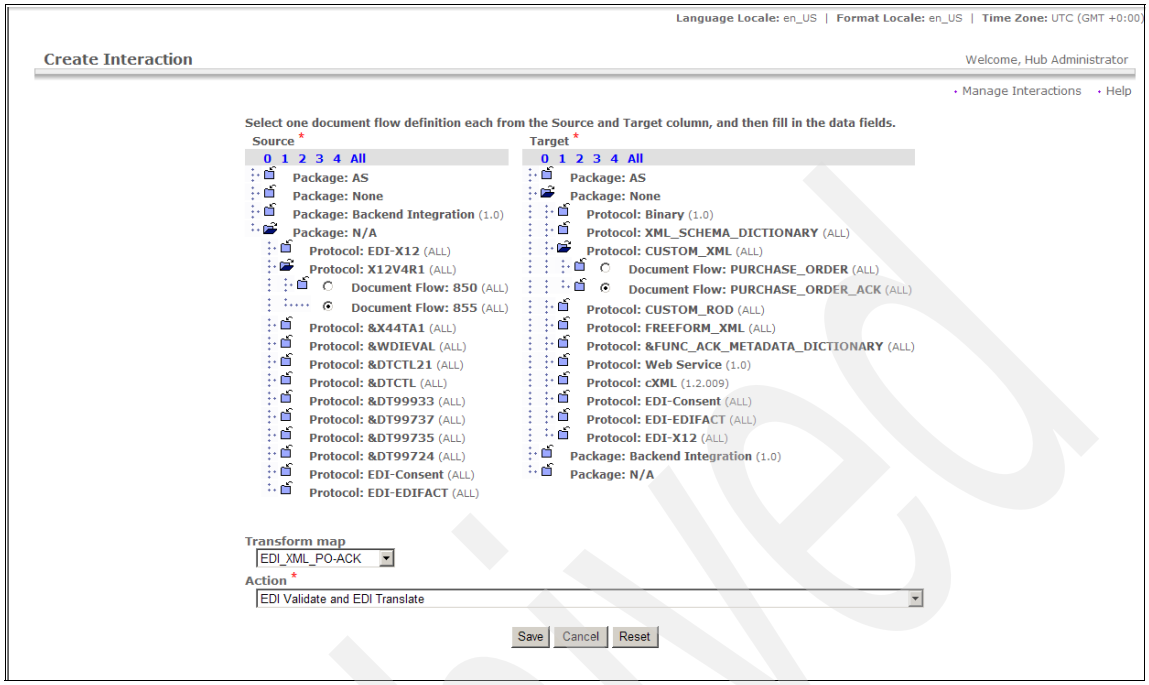


Figure 19-53 Creating the interaction for EDI to XML conversion

Note: The following should be noted from the interactions that are created above:

- The source document flow definitions for the interactions created are AS/EDI-X12/ISA,N/A/X12V4R1/855. Because Company A is the Source Participant in this example, its profile must be updated with these B2B capabilities.
- The target document flow definitions for the interactions created are N/A/EDI-X12/ISA,None/CUSTOM_XML/PURCHASE_ORDER_ACK. Because Company E is the Target Participant in this example, its profile must be updated with the these B2B capabilities.

19.7.5 Update the participant profiles

We need to update the profiles of Company A and Company E to enable the EDI to XML document flow between these two partners.

Update the profile of Company A on Company E server

B2B capabilities for the participant Company A must be must be updated on the WebSphere Partner Gateway Enterprise running on Company E. While logged in as hubadmin,open the profile for Company E, follow these steps:

1. Select **B2B capabilities.**
2. In the B2B capabilities window, update the capabilities by enabling AS/EDI-X12/ISA and N/A/X12V4R1/855 in the order of Package,Protocol and document flow (Figure 19-54).

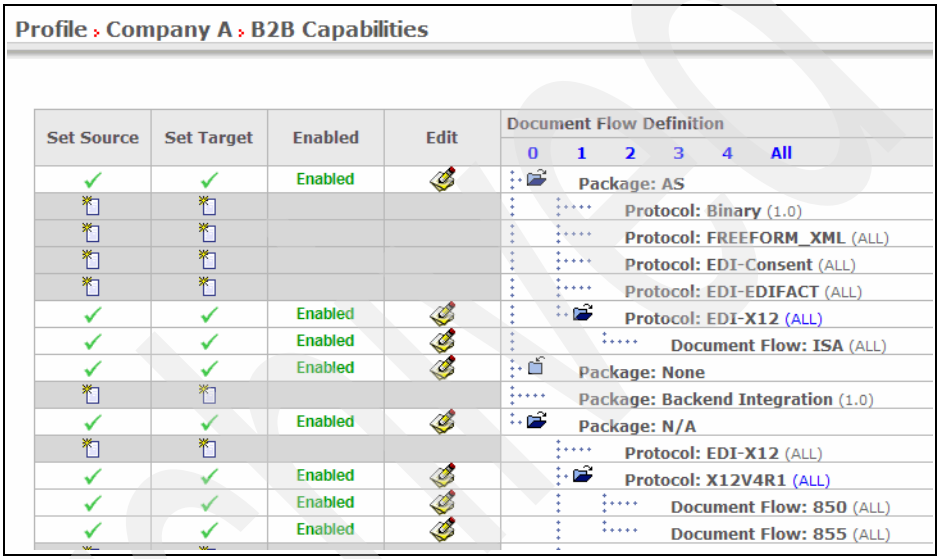


Figure 19-54 B2B capabilities for Company A to implement the de-enveloping scenario

No configuration updates are required as far as the gateway for Company A is concerned because the default gateway is already present.

Update the profile of Company E on Company E server

B2B capabilities for the participant Company E must be must be updated on the WebSphere Partner Gateway Enterprise running on Company E.

While logged in as hubadmin:

3. Open the profile for Company E.
4. Select **B2B capabilities**.
5. In the B2B capabilities window, update the capabilities by enabling N/A/EDI-X12/ISA and None/CUSTOM_XML/PURCHASE_ORDER_ACK in

Archived

the order of Package,Protocol and document flow (Figure Figure 19-55 on page 684).

Profile › Company E › B2B Capabilities

Set Source	Set Target	Enabled	Edit	Document Flow Definition (0 1 2 3 4 All)
✓	✓	Enabled		Package: AS
✓	✓	Enabled		Package: None
✓	✓	Enabled		Protocol: Binary (1.0)
✓	✓	Enabled		Protocol: CUSTOM_XML (ALL)
✓	✓	Enabled		Document Flow: PURCHASE_ORDER (ALL)
✓	✓	Enabled		Document Flow: PURCHASE_ORDER_ACK (ALL)
✓	✓	Enabled		Protocol: CUSTOM_ROD (ALL)
✓	✓	Enabled		Protocol: FREEFORM_XML (ALL)
				Protocol: &FUNC_ACK_METADATA_DICTIONARY (ALL)
				Protocol: Web Service (1.0)
				Protocol: cXML (1.2.009)
				Protocol: EDI-Consent (ALL)
				Protocol: EDI-EDIFACT (ALL)
✓	✓	Enabled		Protocol: EDI-X12 (ALL)
				Package: Backend Integration (1.0)
✓	✓	Enabled		Package: N/A
✓	✓	Enabled		Protocol: EDI-X12 (ALL)
✓	✓	Enabled		Document Flow: ISA (ALL)
✓	✓	Enabled		Protocol: X12V4R1 (ALL)

Figure 19-55 B2B capabilities for Company E for the de-enveloping scenario

As far as the gateway configuration for Company E is concerned, there is no need to create another gateway because the default gateway is already present.

Activate the participant connections and attribute selection

The final step in the configuration is the creation of participant connections.

While logged in as hubadmin, follow these steps:

1. Select **Account Admin** → **participant connections.**
2. Select **Company A** as the Source Participant and **Company E** as the Target Participant.
3. Select **Search.**
4. **Activate** the following Connections:

   AS/EDI-X12/ISA to N/A/EDI-X12/ISA

   N/A/X12V4R1/855 to None/CUSTOM_XML/PURCHASE_ORDER_ACK.

When the participant connection is activated, some of the EDI related attributes need to be set for the participant connections.

For our de-enveloping scenario, we need to set three attributes:

- Allow documents with duplicate document ids
- Validation map
- Validation level.

> **Note:** The Attribute **Allow documents with duplicate document ids** enables the user to send the EDI Interchanges with the same control numbers more than once. If this attribute is set to **No**, EDI documents with the same Control Numbers will be not be processed by WebSphere Partner Gateway and they will always be marked as failed. The default value for this attribute is **No**.

1. Select **Attributes** on the source side of the participant connection for AS/EDI-X12/ISA to N/A/EDI-X12/ISA.
2. Select the **folder** icon next to the Protocol EDI-X12. See Figure 19-56.

Time To Acknowledge	Time To Acknowledge	30	Inherited from: **Scope:** Global **Type:** Time To Acknowledge
Retry Count	Retry Count	3	Inherited from: **Scope:** Global **Type:** Retry Count
AS Compress Before Sign	AS Compress Before Sign	Yes	Inherited from: **Scope:** Global **Type:** AS Compress Before Sign
AS Compressed	AS Compressed	No	Inherited from: **Scope:** Global **Type:** AS Compressed
AS Encrypted	AS Encrypted	No	Inherited from: **Scope:** Global **Type:** AS Encrypted
AS MDN Http Url	AS MDN Http Url		No value provided
AS MDN Email Address	AS MDN Email Address		No value provided
AS MDN Asynchronous	AS MDN Asynchronous	Yes	Inherited from: **Scope:** Global **Type:** AS MDN Asynchronous
AS MDN Requested	AS MDN Requested	Yes	Inherited from: **Scope:** Global **Type:** AS MDN Requested
AS Message Digest Algorithm	AS Message Digest Algorithm	sha1	Inherited from: **Scope:** Global **Type:** AS Message Digest Algorithm
AS MDN Signed	AS MDN Signed	No	Inherited from: **Scope:** Global **Type:** AS MDN Signed
AS Signed	AS Signed	No	Inherited from: **Scope:** Global **Type:** AS Signed
AS Business Id	AS Business Id		No value provided
Allow documents with duplicate document ids	Allow documents with duplicate document ids	No	Inherited from: **Scope:** Global **Type:** Allow documents with duplicate document ids
Allow a TA1 request	Allow a TA1 request	Yes	Inherited from: **Scope:** Global **Type:** Allow a TA1 request
Discard Envelope if any errors	Discard Envelope if any errors	No	Inherited from: **Scope:** Global **Type:** Discard Envelope if any errors

Package: AS (N/A)

Protocol: EDI-X12 (ALL)

Document Flow: ISA (ALL)

Figure 19-56 Selecting the attributes for the Protocol EDI-X12 Under AS Packaging

Archived

3. Select **Yes** from the drop-down box for the attribute **Allow documents with duplicate document ids**. See Figure 19-57 on page 686.
4. Select **Save** to store the configuration.
5. Select **Return** to return to the Manage Connections page.

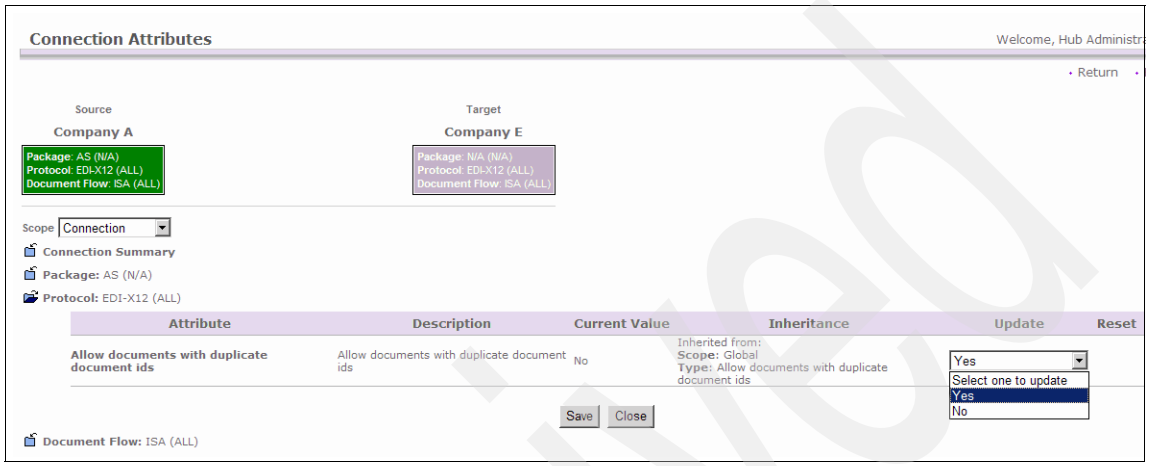


Figure 19-57 Setting the attribute Allow documents with duplicate document ids to Yes

We now set the attribute for validation map:

6. Select **Attributes** on the source side of the participant connection N/A/X12V4R1/855 to **None**/CUSTOM_XML/PURCHASE_ORDER_ACK.
7. Select the **folder** icon next to the protocol X12V4R1, as shown in Figure 19-58 on page 687.

Archived

Max validation error level	Max validation error level	0	Inherited from: **Scope:** Global **Type:** Max validation error level
Validation level	Validation level	0	Inherited from: **Scope:** Global **Type:** Validation level
Char set validation table	Char set validation table	CHARSET	Inherited from: **Scope:** Global **Type:** Char set validation table
Alphanumeric validation table	Alphanumeric validation table	ALPHANUM	Inherited from: **Scope:** Global **Type:** Alphanumeric validation table
Generate group level info only in functional Ack	Generate group level info only in functional Ack	No	Inherited from: **Scope:** Global **Type:** Generate group level info only in functional Ack
Century control year	Century control year	10	Inherited from: **Scope:** Global **Type:** Century control year
Detailed validation of segment	Detailed validation of segment	No	Inherited from: **Scope:** Global **Type:** Detailed validation of segment
Connection Profile Qualifier1	Connection Profile Qualifier1		No value provided
Interchange qualifier	Interchange qualifier		No value provided
Interchange Identifier	Interchange Identifier		No value provided
Interchange usage indicator	Interchange usage indicator		No value provided
Group application sender identifier	Group application sender identifier		No value provided
Group application receiver identifier	Group application receiver identifier		No value provided
Interchange reverse routing	Interchange reverse routing		No value provided
Interchange routing address	Interchange routing address		No value provided
Group application sender qualifier	Group application sender qualifier		No value provided
Group application receiver qualifier	Group application receiver qualifier		No value provided
Group application password	Group application password		No value provided
Validation Map	Validation Map	No value provided	No value provided
XMLNS Active	XMLNS Active	No value provided	No value provided

Package: N/A (N/A)
Protocol: X12V4R1 (ALL)
Document Flow: 855 (ALL)

Figure 19-58 Selecting the attributes for the document flow PURCHASE_ORDER_ACK

8. Select **Validation Level** as **2** from the drop-down box (Figure 19-59).

			Type: Segment output	
Allow documents with duplicate document ids	Allow documents with duplicate document ids	No	Inherited from: **Scope:** Global **Type:** Allow documents with duplicate document ids	Select one to update
Max error level at Transformation	Max error level at Transformation	0	Inherited from: **Scope:** Global **Type:** Max error level at Transformation	Select one to update
FAMap	FAMap	No value provided	No value provided	Select one to update
Envelope Profile	Envelope Profile	No value provided	No value provided	Select one to update
Max validation error level	Max validation error level	0	Inherited from: **Scope:** Global **Type:** Max validation error level	Select one to update
Validation level	Validation level	0	Inherited from: **Scope:** Global **Type:** Validation level	2
Char set validation table	Char set validation table	CHARSET	Inherited from: **Scope:** Global **Type:** Char set validation table	
Alphanumeric validation table	Alphanumeric validation table	ALPHANUM	Inherited from: **Scope:** Global **Type:** Alphanumeric validation table	
Generate group level info only in functional Ack	Generate group level info only in functional Ack	No	Inherited from: **Scope:** Global **Type:** Generate group level info only in functional Ack	Select one to update
Century control year	Century control year	10	Inherited from: **Scope:** Global **Type:** Century control year	
Detailed validation of segment	Detailed validation of segment	No	Inherited from: **Scope:** Global **Type:** Detailed validation of segment	Select one to update
Connection Profile Qualifier1	Connection Profile Qualifier1	No value provided	No value provided	
Interchange qualifier	Interchange qualifier	No value provided	No value provided	
Interchange Identifier	Interchange Identifier	No value provided	No value provided	
Interchange usage indicator	Interchange usage indicator	No value provided	No value provided	
Group application sender identifier	Group application sender identifier	No value provided	No value provided	
Group application receiver identifier	Group application receiver identifier	No value provided	No value provided	
Interchange reverse routing	Interchange reverse routing	No value provided	No value provided	

Select one to update
0
1
2

Figure 19-59 Setting the attribute Validation map

Archived

9. In the same page, click the **folder** icon next to **Document Flow:855**.
10. Select **355X410_VALIDATE** from the Validation Map Screen. Click **Save.**
11. Select **Return** to return back to the Manage Connections Page.

The necessary configuration required to implement the de-enveloping scenario is complete. In the next section, we validate this configuration by routing the EDI Purchase Order Acknowledgement envelope through WebSphere Partner Gateway.

19.8 Receiving and validating the communication

Before validating the communication, the role of the EDI validation map has to be discussed. The validation map **355X410_VALIDATE** that was uploaded as a part of this example, is a simple map that validates the edi data element 353 at position 1 of BAK segment(BAK01) of the EDI-855 Purchase Order (Example 19-9).

Example 19-9 EDI Purchase Order

```
ISA*00*          *00*          * *companya * *companye
*241002*1635*U*00401*000000014*0*P*z!
GS*  *  *  *241002*1635*000000014* *004010!
ST*855*000000014!
BAK*06*AT*P365542*01101402!
PO1**132*EA*10.56*PE*BP*00031P0241!
PID*F****SCSI Card!
PO1**16*EA*1.99*PE*BP*00021P0241!
PID*F****IDE Card!
SE*7*000000014!
GE*1*000000014!
IEA*1*000000014!
```

Based on the value of this data element, we can divide the de-enveloping scenario into two parts.

1. Firstly, a positive validation in which the edi data element 353 at position 1 of BAK segment will be set to 06
2. The second validation is a negative scenario in which the edi data element 353 at position 1 of BAK segment is set to some value other than 06.

19.8.1 Positive validation

In this scenario, the following EDI PO ACK in Example 19-10 on page 689 is routed from Company A to Company E.

Example 19-10 EDI Purchase order ack PO

```
ISA*00*          *00*          * *companya * *companye
*241002*1635*U*00401*000000014*0*P*z!
GS*  *  *  *241002*1635*000000014* *004010!
ST*855*000000014!
BAK*06*AT*P365542*01101402!
PO1**132*EA*10.56*PE*BP*00031P0241!
PID*F****SCSI Card!
PO1**16*EA*1.99*PE*BP*00021P0241!
PID*F****IDE Card!
SE*7*000000014!
GE*1*000000014!
IEA*1*000000014!
```

We see the edi data element in this scenario is 06:

1. Log in to the WebSphere Partner Gateway Advanced on Company A and place the data shown in Example 19-10 in the file directory Receiver (`/home/bcguser/wpgv6/data/companya/edi_out`).
2. Click **Viewers** → **Document Viewers .**
3. Search for the document exchange from Company A to Company E. We see the successful document transfer from Company A to Company E.

   See Figure 19-60.

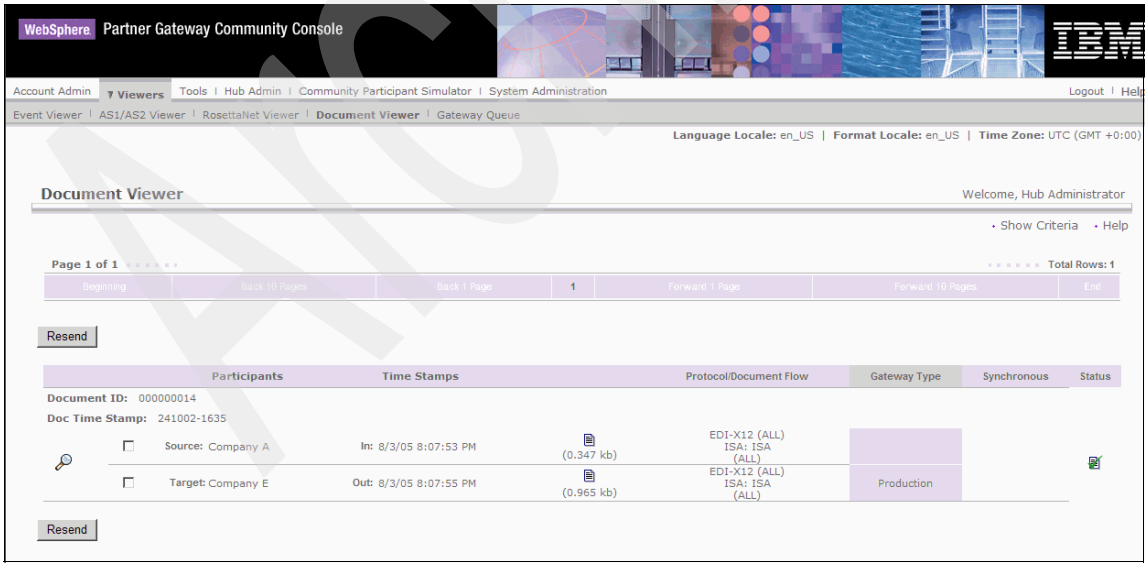


Figure 19-60 Successful document transfer from Company A to Company E

Archived

4. Log in to WebSphere Partner Gateway Enterprise on Company E and Click **Viewers → Document Viewers.**
5. Search for the documents exchanged between Company A and Company E. We see the following documents in the viewers (Figure 19-61).

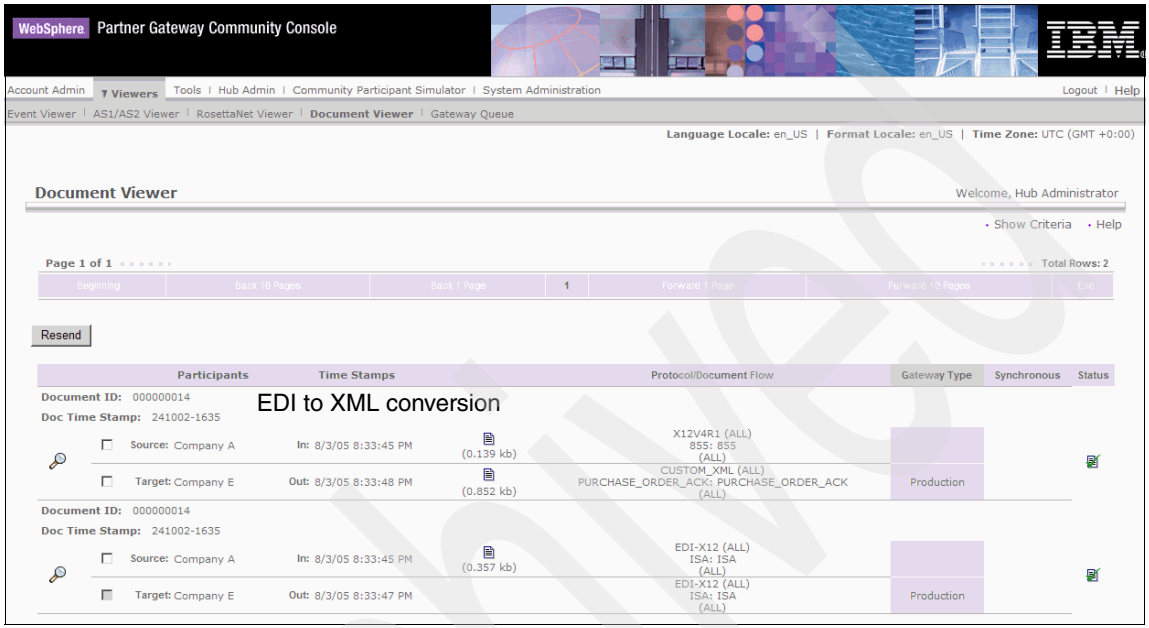


Figure 19-61 EDI to XML conversion documents

6. For the same scenario, when the input document consists of more than one transaction and is posted from Company A to Company E, we see the following output in the Company E document viewer (Figure 19-62 on page 691).

Archived

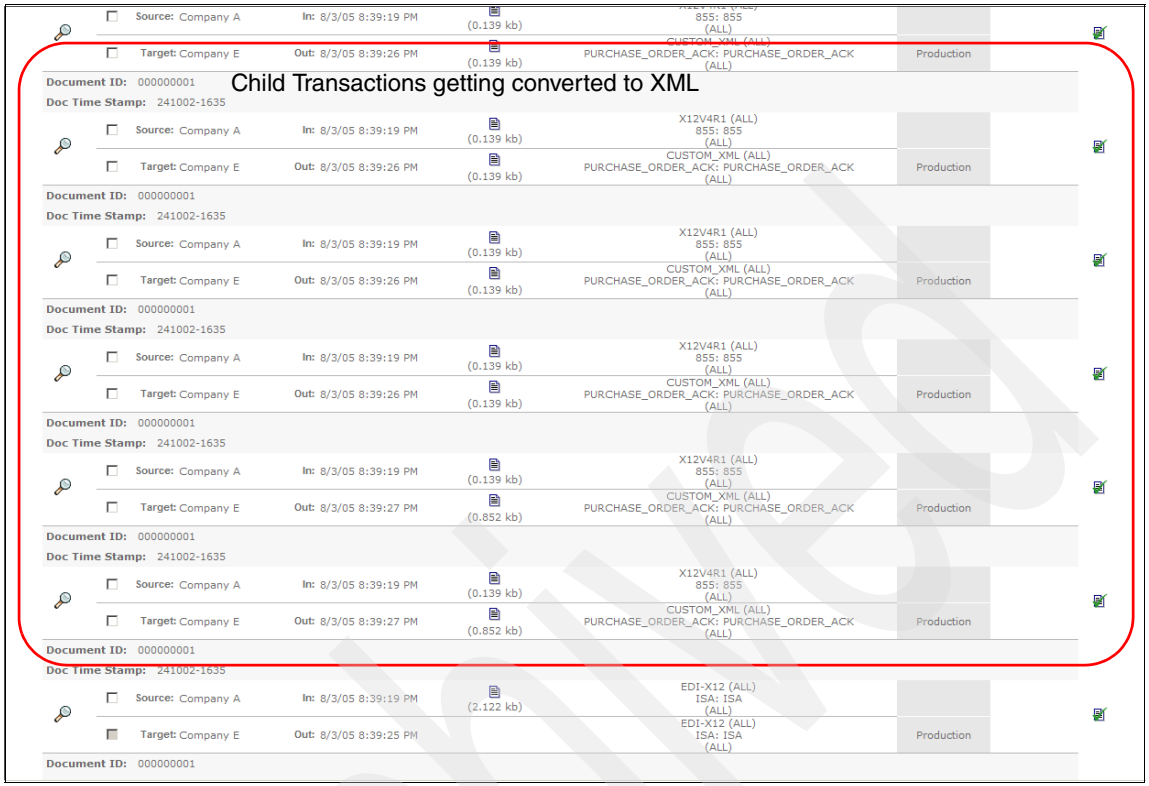


Figure 19-62 EDI-X12 PO Ack with multiple transactions routed

7. If we click the **view details** icon of the Parent document EDI-X12 Purchase Order Acknowledgement, and click **Source side child Transactions**, we will able to observe all the child EDI Transactions for the EDI-Envelope (see Figure 19-63 on page 692).

Archived

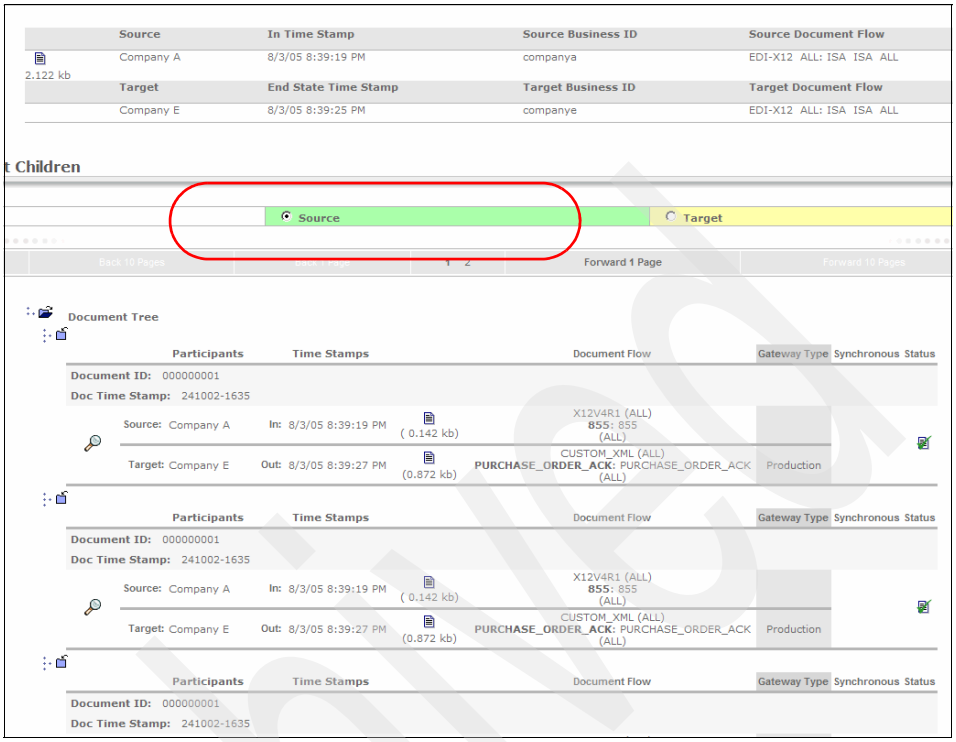


Figure 19-63 Child EDI transactions

With this we have finished the positive scenario in which the EDI 855 will be routed successfully when the data element 353 was set as 06. Next, we look at the second scenario in which Data Element 1 was not set to 06.

19.8.2 Negative validation

In this scenario, the following EDI-X12 PO ACK will be routed. We can observe that the edi data element 353 at position 1 of BAK segment (BAK01) is set to a value other than 06 (Figure 19-11). We have set it to a value of 09.

Example 19-11 EDI Purchase Order

```
ISA*00*          *00*          * *companya * *companye
*241002*1635*U*00401*000000014*0*P*z!
GS*  *  *  *241002*1635*000000014* *004010!
ST*855*000000014!
BAK*09*AT*P365542*01101402!
PO1**132*EA*10.56*PE*BP*00031P0241!
PID*F****SCSI Card!
PO1**16*EA*1.99*PE*BP*00021P0241!
```

```
PID*F****IDE Card!
SE*7*000000014!
GE*1*000000014!
IEA*1*000000014!
```

1. As we did in the earlier sections, this document will be placed in the file directory receiver of Company A.
2. After it is received at Company E, the document fails validation and the same can be observed in the Console of WebSphere Partner Gateway Enterprise on Company E (by clicking **Viewers** → **Document Viewers**) . We see the failed documents in Figure 19-64.

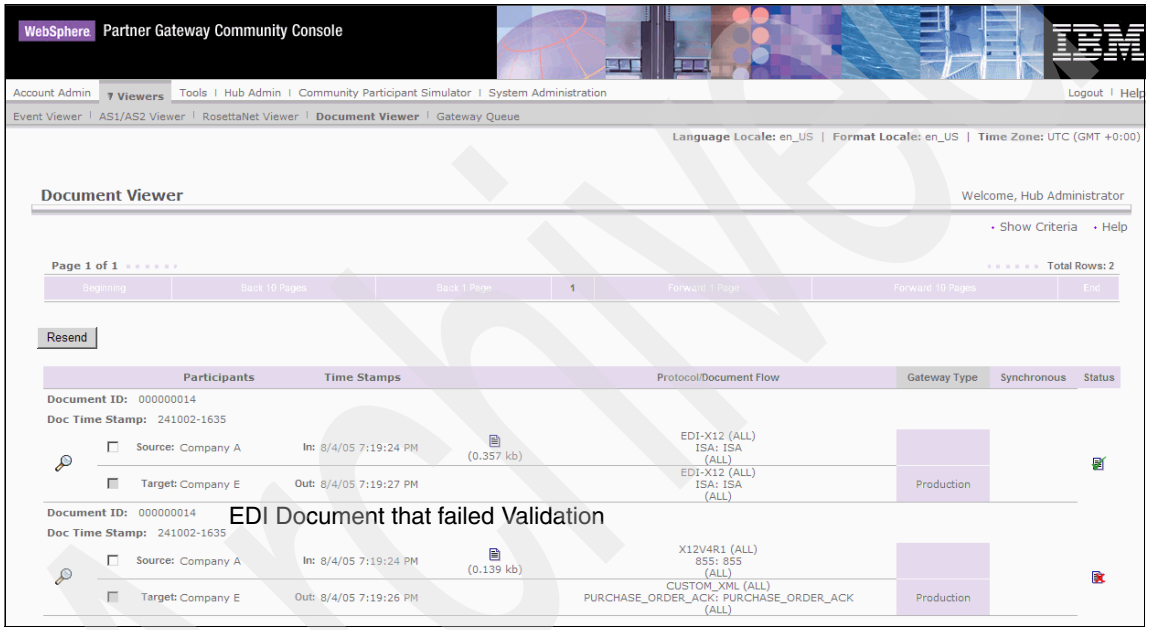


Figure 19-64 Documents failed Validation on WebSphere Partner Gateway

3. If we click the **View details** icon of the document in the viewer, we see the following error events in the Console in Figure 19-65 on page 694:

Archived

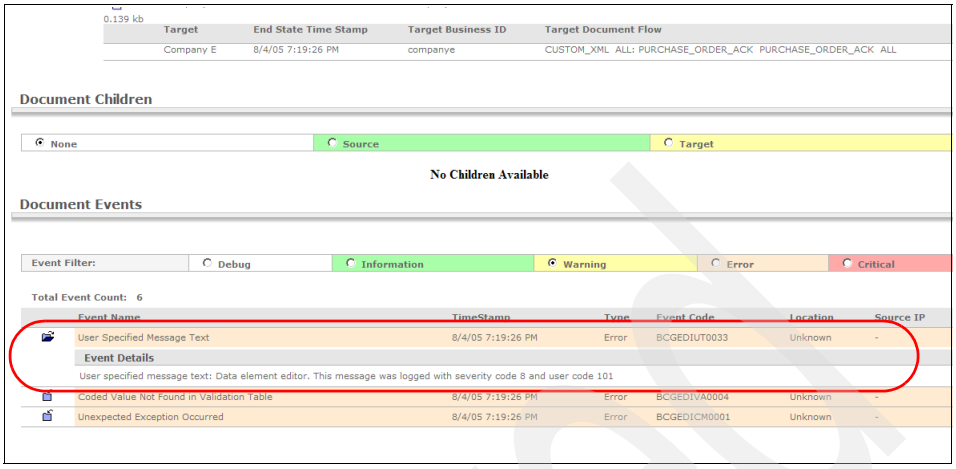


Figure 19-65 Failure events in the document Viewers

Archived

20

Functional acknowledgements

A reporting mechanism plays a vital role in any business document transfer. Functional Acknowledgements (FAs) are a special class of EDI transactions that forms the basis for the EDI business transaction reporting mechanism. *Functional Acknowledgements* are business receipts similar to RosettaNet signals or AS2 MDNs. An FA generally acknowledges the receipt of well-formed EDI. Even if the received EDI interchange is not well-formed, the FA will be generated to make sure that the sending trading partner is kept informed. FAs themselves are transactions and so will come from the partner in an EDI interchange, possibly along with other business transactions. A Functional acknowledgement can be requested at various levels such as interchange level, group level, transaction l and so on. For example, a group-level FA acknowledges the successful or unsuccessful receipt of an EDI group.

© Copyright IBM Corp. 2005. All rights reserved.

20.1 Structure of an FA

For this chapter, we are going to deal with the generation of two FAs, TA1 and 997 from the EDI-X12 transaction family. We can see the structure of each of these EDI transactions by taking a look at some sample transactions.

20.1.1 Analyzing a TA1 Transaction

The TA1 is the interchange level acknowledgment. It reports whether a single interchange (ISA - IEA envelope) was accepted or rejected. The TA1 is a single segment. It must be enveloped in an interchange containing one or more other transaction sets. An interchange may contain one or more TA1s. The TA1s follow the ISA interchange header, and precede the first GS functional group header.

We can interpret the TA1 Transaction (Example 20-1) as follows:

Example 20-1 Sample TA1 transaction

```
TA1*123456789*050731*1227*A*000
```

Note: The example above is a raw EDI transaction which is not enveloped within the ISA and IEA segments.

Points that can be noted from the TA1 transaction are:

- Assuming and asterisk (*) for the element separator and exclamation point (!) for the segment terminator, the TA1 shows that the interchange with control number 123456789 sent 07/31/2005 at 1227 was accepted.
- Element TA101 is the interchange control number, 123456789 which the TA1 is acknowledging.
- Element TA102 is the interchange date, 050731, interpreted as 7/31/2005.
- Element TA103 is the interchange time, 1227, the time zone is unspecified.

Note: Elements TA101 - TA103 that are explained above are taken from the ISA of "subject" interchange, that is, the interchange being acknowledged by the TA1.

- Element TA104 is the acknowledgment code, (A) for accepted.
- Element TA105 is the note code (reason code), (000) meaning no errors.

Similarly, the sample TA1 in Example 20-2 indicates interchange 666777888 was rejected due to an invalid control structure. An invalid control structure might include things such as a GS that is missing a GE.

Example 20-2 Sample TA1 transaction

```
TA1* 666777888*050228*1745*R*022!
```

The sender can request a TA1 from the receiver by setting element 14 of the ISA Interchange Control Header to 1 (Example 20-3).

Example 20-3 ISA segment which requests TA1

```
ISA*00*          *00*          *123456789*      *987654321*
*930419*0021*U*00200*123456789*1*T*:!
```

20.1.2 Analyzing a 997 Transaction

The 997 is a group level acknowledgement. It reports whether a single EDI group has been accepted or rejected. Interpreting a 997 is somewhat more complex. We look at the following example EDI-X12 997 transaction (Example 20-4).

Example 20-4 Sample EDI-X12 997 transaction

```
AK1*PO*1234~
AK9*A*1*1*1~
```

At the functional group level, results are conveyed by the AK1 response header and the AK9 response trailer. So, the above 997 transaction would indicate that group 1234 was accepted with no errors.

From the above 997 Transaction, the following points can be noted:

- Element AK101 is the functional group code. PO is the code for a group containing 850 (the Purchase Order transaction sets).
- Element AK102 is the functional group control number for the subject group, 1234.
- Element AK901 is the acknowledgment code, again (A) for accepted.

Archived

Note: The control number of the subject functional group is reported, but the control number of the subject interchange is not. Also, neither the date and time of the subject interchange nor the subject group are reported. As with the TA1, it is assumed that the sender and receiver of the subject interchange are reversed, and also the sender and receiver of the functional group are reversed.

As a second example, see Example 20-5.

Example 20-5 Sample EDI-X12 997 transaction

```
AK1*PO*1235~
AK9*R*1*1*0*5~
```

The 997 transaction in Example 20-5 indicates that group 1235 was rejected as a result of a mismatch in the number of included transaction sets. We can see the following points:

- Element AK102 is again the group control number, 1235.
- Element AK901 is now (R) for rejected.
- Element AK902 is number of transaction sets included, which is 1 in this case.
- Element AK903 is number of received transaction sets, which is 1 in this case.
- Element AK904 is the number of accepted transaction sets, zero.
- Element AK905, the error code, is (5), indicating Number of Included Transaction Sets Does Not Match Actual Count.

As our final example, Example 20-6 shows a sample EDI-X12 997 transaction reporting on a group containing three transaction sets, with last set rejected due to an error:

Example 20-6 Sample EDI-X12 997 transaction

```
AK1*PO*1401~
AK2*270*32765!
AK5*A!
AK2*270*32766!
AK5*A!
AK2*270*32767!
AK3*NM1*3**3!
AK5*R!
AK9*P*3*3*2~
```

From the EDI-X12 997 Transaction in Example 20-6 on page 698, we can make the following observations:

- Element AK102 is the group control number, 1401.
- Element AK901 is (P) for Partially Accepted, At Least One Transaction Set Was Rejected.
- Element AK904 reflects that only two transaction sets were accepted.
- The first two AK2 loops reflect that transaction sets 32765 and 32766 (AK202) were accepted (AK501 = A).
- The third AK2 loop indicates that transaction set 32767 (AK202) was rejected (AK501 = R).
- The AK3 segment in the third AK2 loop indicates that a mandatory NM1 segment (AK301) was missing (AK304 = 3 = Mandatory segment missing).

Note: The request for a 997 is not automatic. It has to be performed according to the business-level agreement between the trading partners who will be exchanging EDI Documents. For example, a business-level agreement is possible for two trading partners that for every EDI-X12 group that was received, the receiver has to send the FA 997.

20.2 Implementing our FA scenario

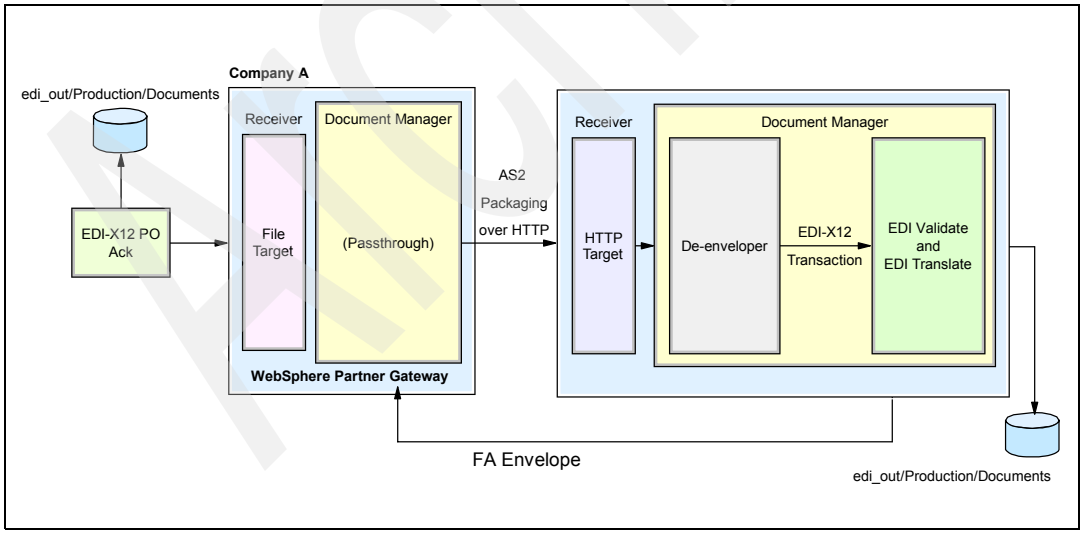


Figure 20-1 FA scenario

This scenario is an extension to the de-enveloping scenario that was described in 19.7, “Implement the de-enveloping scenario” on page 674. The scenario can be explained as follows.

After the EDI-X12 Purchase Order envelope is received at WebSphere Partner Gateway Enterprise on Company E from Company A, WebSphere Partner Gateway, validates the incoming EDI envelope and generates the TA1 (which can be requested by setting ISA14 to 1 in the interchange) and 997 FA transactions. Because the individual transactions cannot flow through the system, both the EDI FA transactions need to be enveloped and will be sent back to the Company A over AS Packaging.

Note: An FA will always be directed in the reverse direction to the original EDI Document flow. In our example, as the original EDI document is routed from Company A to Company E, so the FA envelope is routed from Company E to Company A.

To implement this scenario, the following configuration updates are needed on WebSphere Partner Gateway Enterprise on Company E:

1. Associate the FA Map to the EDI document flow definition.
2. Set the required attributes on the participant connections from Company A to Company E for FA generation.
3. Create the interactions for FA channels.
4. Update the participant profiles for Company E and Company A.
5. Create the envelope profile required for FAs.
6. Activate the channels and set the attributes for FA generation.

20.3 Configure Company E to generate the FAs

In this section, the configuration steps that need to be followed on WebSphere Partner Gateway Enterprise on Company E in order to implement the FA scenario are discussed in detail.

20.3.1 Associating the FA map with the EDI document flow definition

An *FA map* is a logical map that governs the structure of the outbound FA transaction. WebSphere Partner Gateway generates a default 997 FA which can be modified according to specific customer requirements using FA maps. When WebSphere Partner Gateway is installed, some default FA maps are also installed with the product.

We will be using one such map, `&DT99724`, to demonstrate in this example. To make use of this FA map, the FA map needs to be associated with the document flow definition to which the FA is going to acknowledge.

> **Note:** As far as the EDI FA TA1 in WebSphere Partner Gateway is concerned, there is no need to associate any FA map to the document flow. If the ISA 14 in the inbound EDI envelope for which the TA1 is acknowledging is set as 1, TA1 will be triggered automatically. But for rest of the FA maps, the explicit association of FA Maps to the document flow definitions is required.

While logged in as hubadmin, follow these steps:

1. Click **Hub Admin** → **Hub Configuration** → **Maps** → **EDI FA Maps**. We can see the list of default EDI FA Maps under this list (Figure 20-2).

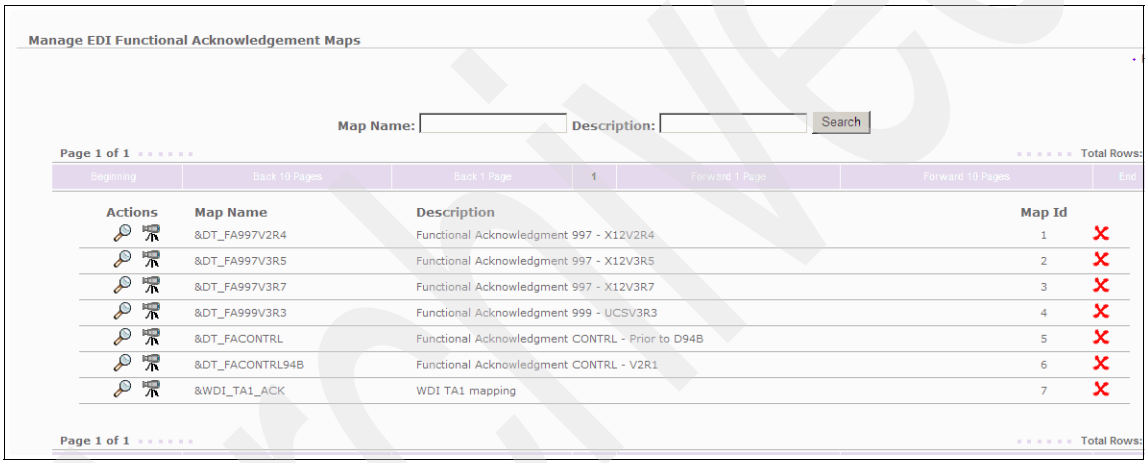


Figure 20-2 List of the EDI FA maps in Community Console

2. Click the icon next to the FA map **&DT_997V2R4** to view the FA Map Associations Page.
3. Under the System Level, select the **Package**, **Protocol** and **Document flow** for which the which the FA map has to be associated (seeFigure 20-3 on page 702). In this case, select **N/A/X12V4R1/855**.
4. Click **Save.**

Archived

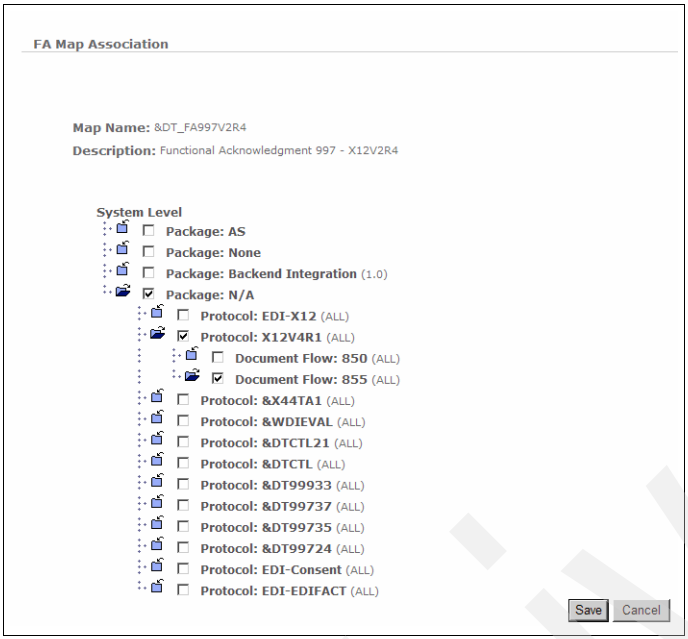


Figure 20-3 Associating the FA map &DT_997V2R4

This finishes the task of and associating the EDI FA Map with a document flow definition. We need to set the attributes on the participant connections from Company A to Company E so that the FA generation is triggered.

20.3.2 Set the attributes to trigger the FA generation

Two attributes, FA Map and Allow TA1 Request, have to be set on the participant connections from Company A to Company E so that the FA generation is activated.

To set the attribute FA Map, follow these steps while logged in as hubadmin:

1. Select **Account Admin** → **Participant Connections.**
2. In the Manage Connections Page, select Company A as the source participant and Company E as the target participant and click **Search**. In the page that appears, we see the de-enveloping channel and the transformation channel from Company A to Company E (Figure 20-4 on page 703).

Archived

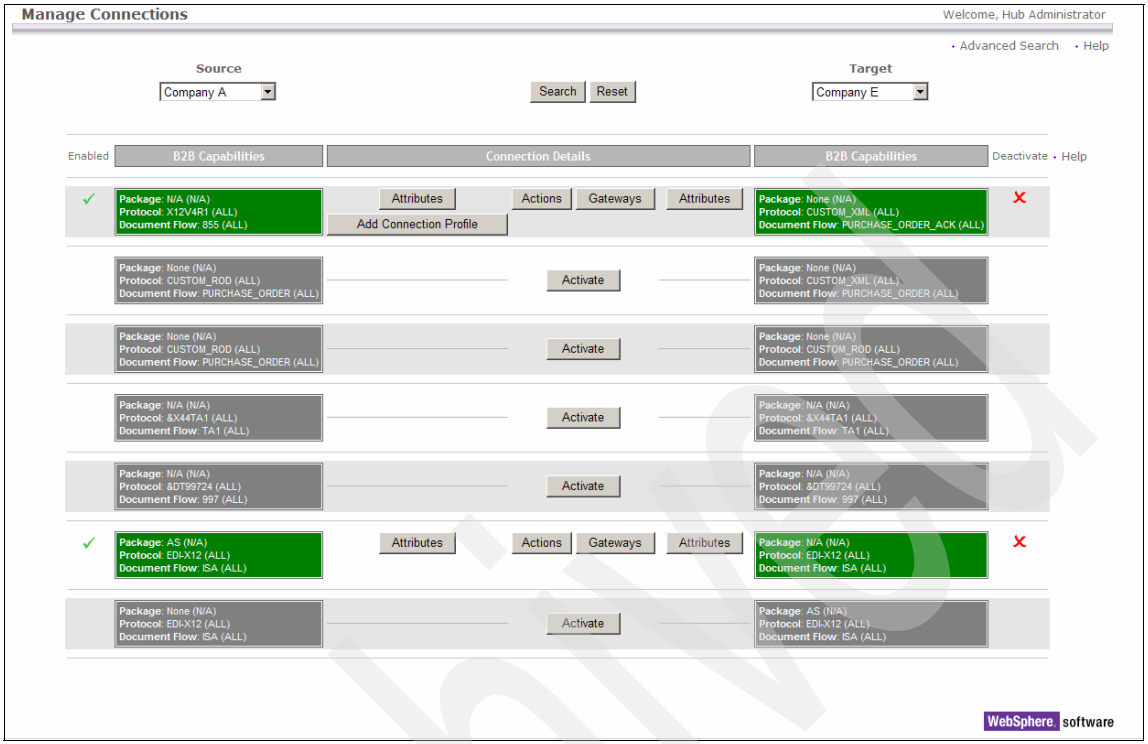


Figure 20-4 The Participant connections (from Company A to Company E) that triggers the EDI FAs

3. Select the **Source** side attributes for the participant connection N/A/X12V4R1/855 to None/CUSTOM_XML .
4. In the next screen, click the **folder** icon next to the protocol **X12V4R1**(Figure 20-5 on page 704).

Archived

			Type: Segment output
Allow documents with duplicate document ids	Allow documents with duplicate document ids	No	Inherited from: **Scope:** Global **Type:** Allow documents with duplicate document ids
Max error level at Transformation	Max error level at Transformation	0	Inherited from: **Scope:** Global **Type:** Max error level at Transformation
FAMap	FAMap	No value provided	No value provided
Envelope Profile	Envelope Profile	No value provided	No value provided
Max validation error level	Max validation error level	0	Inherited from: **Scope:** Global **Type:** Max validation error level
Validation level	Validation level	0	Inherited from: **Scope:** Global **Type:** Validation level
Char set validation table	Char set validation table	CHARSET	Inherited from: **Scope:** Global **Type:** Char set validation table
Alphanumeric validation table	Alphanumeric validation table	ALPHANUM	Inherited from: **Scope:** Global **Type:** Alphanumeric validation table
Generate group level info only in functional Ack	Generate group level info only in functional Ack	No	Inherited from: **Scope:** Global **Type:** Generate group level info only in functional Ack
Century control year	Century control year	10	Inherited from: **Scope:** Global **Type:** Century control year
Detailed validation of segment	Detailed validation of segment	No	Inherited from: **Scope:** Global **Type:** Detailed validation of segment
Connection Profile Qualifier1	Connection Profile Qualifier1		No value provided
Interchange qualifier	Interchange qualifier		No value provided
Interchange Identifier	Interchange Identifier		No value provided
Interchange usage indicator	Interchange usage indicator		No value provided
Group application sender identifier	Group application sender identifier		No value provided
Group application receiver identifier	Group application receiver identifier		No value provided
Interchange reverse routing	Interchange reverse routing		No value provided
Interchange routing address	Interchange routing address		No value provided
Group application sender qualifier	Group application sender qualifier		No value provided
Group application receiver qualifier	Group application receiver qualifier		No value provided
Group application password	Group application password		No value provided
Validation Map	Validation Map		No value provided
XMLNS Active	XMLNS Active	No value provided	No value provided

Package: N/A (N/A)
Protocol: X12V4R1 (ALL)
Document Flow: 855 (ALL)

Figure 20-5 Selecting the attributes for the protocol X12V4R1

5. In the next screen for the attribute FA Map, select the value **&DT99724** (Figure 20-6 on page 705).
6. Click **Save** and then **Return** to return to the Manage Connections screen.

Archived

Protocol: X12V4R1 (ALL)

Attribute	Description	Current Value	Inheritance	Update	Reset
Segment delimiter	Segment delimiter	No value provided	No value provided		
Data element delimiter	Data element delimiter	No value provided	No value provided		
Subelement delimiter	Subelement delimiter	No value provided	No value provided		
Release character	Release character	No value provided	No value provided		
Repeating data element character	Repeating data element character	No value provided	No value provided		
Decimal notation	Decimal notation	No value provided	No value provided	Select one to update	
Segment output	Segment output	Yes	Inherited from: **Scope:** Global **Type:** Segment output	Select one to update	
Allow documents with duplicate document ids	Allow documents with duplicate document ids	No	Inherited from: **Scope:** Global **Type:** Allow documents with duplicate document ids	Select one to update	
Max error level at Transformation	Max error level at Transformation	0	Inherited from: **Scope:** Global **Type:** Max error level at Transformation	Select one to update	
FAMap	FAMap	No value provided	No value provided	&DT_FA997V2R4	
Envelope Profile	Envelope Profile	No value provided	No value provided	Select one to update &DT_FA997V2R4	
Max validation error level	Max validation error level	0	Inherited from: **Scope:** Global **Type:** Max validation error level	Select one to update	
Validation level	Validation level	0	Inherited from: **Scope:** Global **Type:** Validation level	Select one to update	
Char set validation table	Char set validation table	CHARSET	Inherited from: **Scope:** Global **Type:** Char set validation table		
Alphanumeric validation table	Alphanumeric validation table	ALPHANUM	Inherited from: **Scope:** Global **Type:** Alphanumeric validation table		
Generate group level info only in functional Ack	Generate group level info only in functional Ack	No	Inherited from: **Scope:** Global **Type:** Generate group level info only in functional Ack	Select one to update	
Century control year	Century control year	10	Inherited from: **Scope:** Global **Type:** Century control year		
Detailed validation of segment	Detailed validation of segment	No	Inherited from: **Scope:** Global **Type:** Detailed validation of segment	Select one to update	
Connection Profile Qualifier1	Connection Profile Qualifier1	No value provided	No value provided		
Interchange qualifier	Interchange qualifier	No value provided	No value provided		

Figure 20-6 Selecting the attribute FA Map as &DT99724

7. To set the attribute Allow TA1 Request, select the participant connection from AS/EDI-X12/ISA to N/A/EDI-X12/ISA and click **Attributes** on the source side.
8. In the next screen, click the **folder** icon next to the document flow **ISA** (Figure 20-7 on page 706).

Archived

Time To Acknowledge	Time To Acknowledge	30	Inherited from: **Scope:** Global **Type:** Time To Acknowledge
Retry Count	Retry Count	3	Inherited from: **Scope:** Global **Type:** Retry Count
AS Compress Before Sign	AS Compress Before Sign	Yes	Inherited from: **Scope:** Global **Type:** AS Compress Before Sign
AS Compressed	AS Compressed	No	Inherited from: **Scope:** Global **Type:** AS Compressed
AS Encrypted	AS Encrypted	No	Inherited from: **Scope:** Global **Type:** AS Encrypted
AS MDN Http Url	AS MDN Http Url		No value provided
AS MDN Email Address	AS MDN Email Address		No value provided
AS MDN Asynchronous	AS MDN Asynchronous	Yes	Inherited from: **Scope:** Global **Type:** AS MDN Asynchronous
AS MDN Requested	AS MDN Requested	Yes	Inherited from: **Scope:** Global **Type:** AS MDN Requested
AS Message Digest Algorithm	AS Message Digest Algorithm	sha1	Inherited from: **Scope:** Global **Type:** AS Message Digest Algorithm
AS MDN Signed	AS MDN Signed	No	Inherited from: **Scope:** Global **Type:** AS MDN Signed
AS Signed	AS Signed	No	Inherited from: **Scope:** Global **Type:** AS Signed
AS Business Id	AS Business Id		No value provided
Allow documents with duplicate document ids	Allow documents with duplicate document ids	Yes	Inherited from: **Scope:** Connection **Type:** Allow documents with duplicate document ids
Allow a TA1 request	Allow a TA1 request	Yes	Inherited from: **Scope:** Global **Type:** Allow a TA1 request
Discard Envelope if any errors	Discard Envelope if any errors	No	Inherited from: **Scope:** Global **Type:** Discard Envelope if any errors

Package: AS (N/A)

Protocol: EDI-X12 (ALL)

Document Flow: ISA (ALL)

Legend

Figure 20-7 Selecting the attributes for the document flow ISA

9. In the next screen, make sure that the attribute Allow TA1 Request is Set to **Yes** (Figure 20-8 on page 707).

Archived

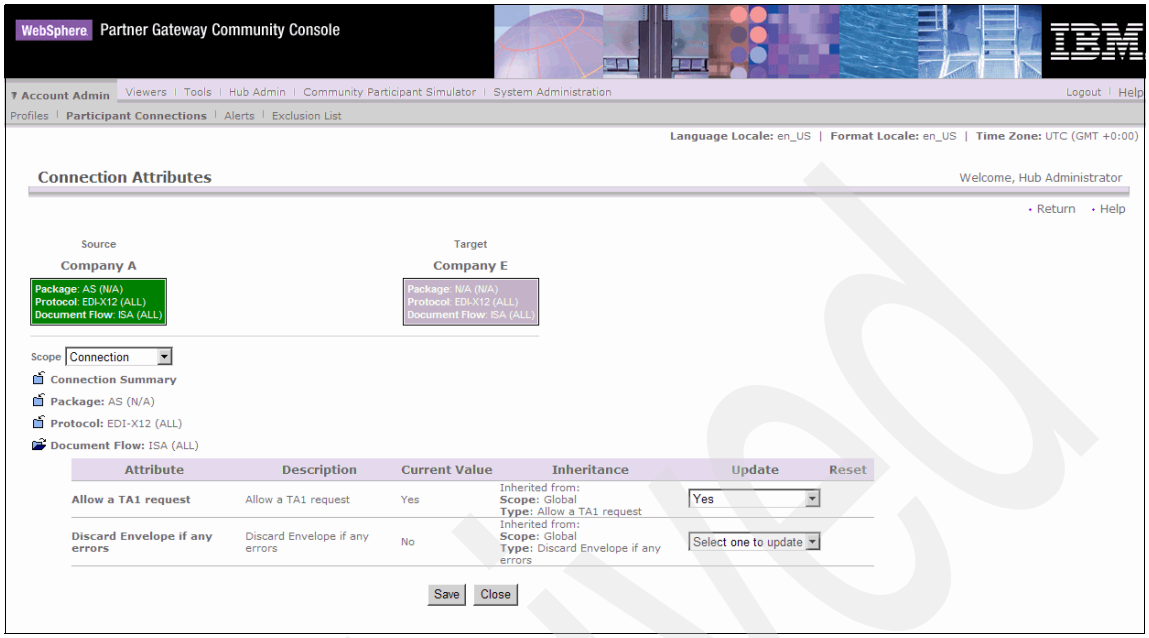


Figure 20-8 Setting the attribute Allow TA1 Request to Yes.

This finishes the task of setting the necessary attributes that are required for FA generation. We now need to create the required interactions.

20.3.3 Create interactions

For our scenario, we need to create three interactions

- An interaction for the generated FA TA1.
- An interaction for the generated FA 997.
- An interaction which enables the EDI FA transaction to be enveloped.

Follow this procedure to create these, while logged on as hubadmin:

1. Select **Hub Admin** → **Hub Configuration** → **Document Flow Definition.**
2. Click **Manage Interactions**.
3. Click **Create Interaction**.
4. We need an interaction from N/A/&X44TA1/TA1 to N/A/&X44TA1/TA1 in Package/Protocol/Document Flow order (Figure 20-9 on page 708).

Archived

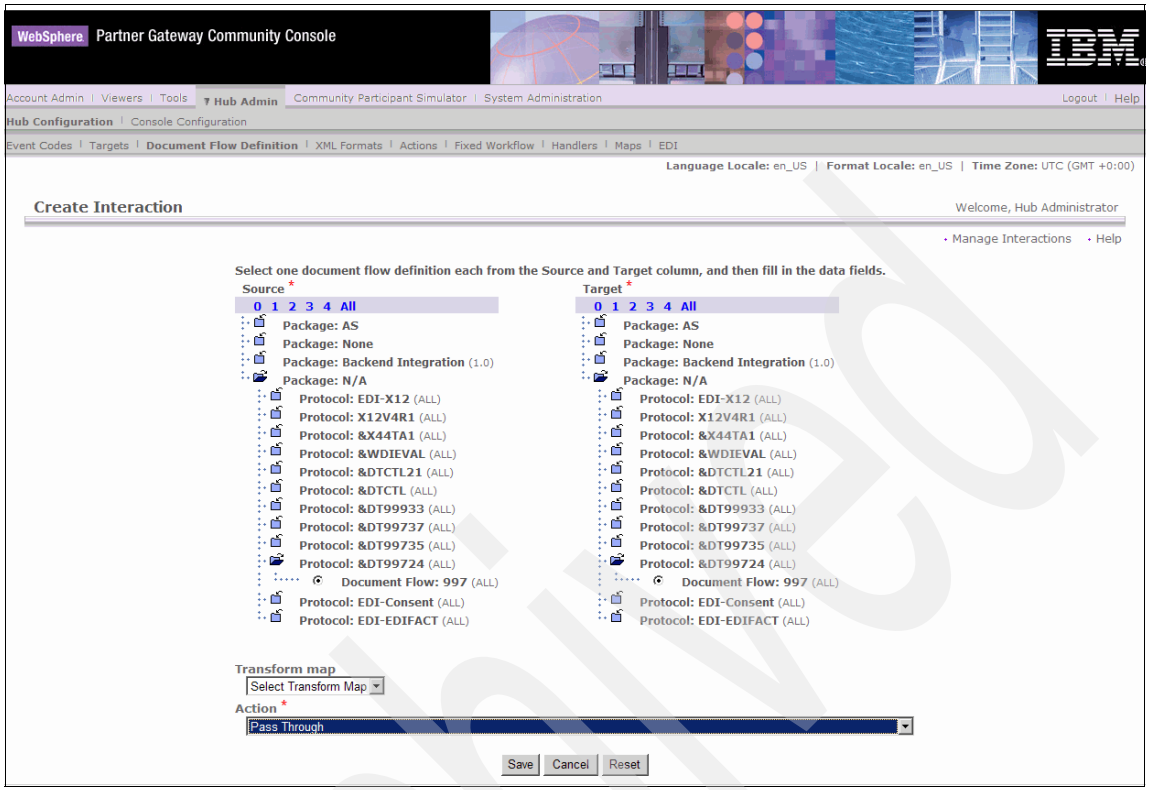


Figure 20-9 Creating the interaction from N/A/&DT99724/997 to N/A/&DT99724/997

5. Select **Pass through** from the **Action** drop-down box (Figure 20-9). Click **Save** to return to the Manage Interactions screen. This finishes the creation of the first interaction.

6. To create the second interaction, repeat the previous steps. The interaction that is going to be created is from N/A/&X44TA1/TA1 to N/A/&X44TA1/TA1. See Figure 20-10 on page 709.

Archived

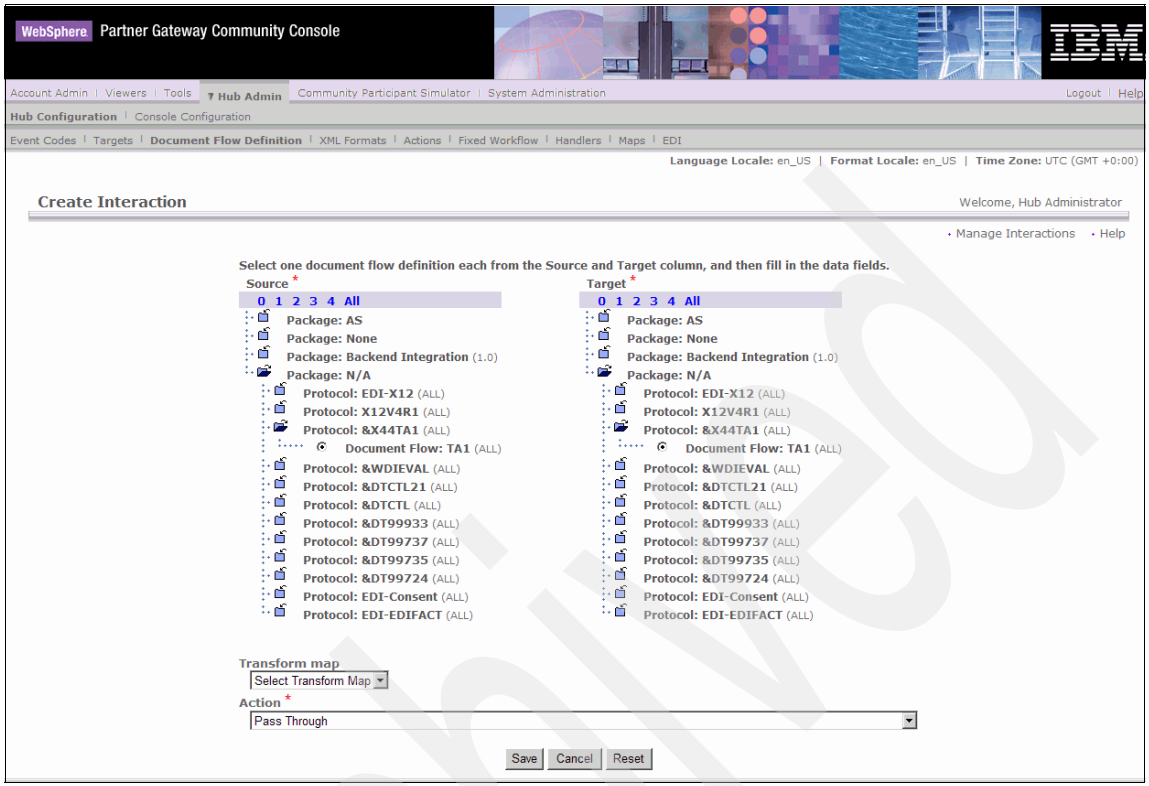


Figure 20-10 Creating the interaction from N/A/&X44TA1/TA1 to N/A/&X44TA1/TA1

7. The third required interaction is for the enveloping channel and it has to be created from N/A/EDI-X12/ISA to AS/EDI-X12/ISA. As we have already created this interaction while creating the channels for enveloping channel, (see Chapter 19, “Enveloping and de-enveloping” on page 621) there is no need to create this channel again.

Archived

Note: Be aware of the following items about the interactions that we just created:

The source document flow definitions for the interactions created are N/A/&X44TA1/TA1, N/A/&DT99724/997 and AS/EDI-X12/ISA. Because Company E is the source participant in this example, its profile must be updated with these B2B capabilities.

The target document flow definitions for the interactions created are N/A/&X44TA1/TA1, N/A/&DT99724/997 and AS/EDI-X12/ISA. Because Company A is the target participant in this example, its profile must be updated with the these B2B capabilities.

This ends the necessary configuration for interactions. In the next section, we update the partner profiles for Company E and A on Server E.

20.3.4 Updating the participant profiles

As in earlier chapters, we need to update the profiles of Company E and Company A to enable the FA document flow among these two partners.

Update the profile of Company E on the Company E server

The B2B capabilities for the participant Company E must be updated with the new EDI FA on the WebSphere Partner Gateway Enterprise running on Company E.

While logged in as hubadmin, follow these steps:

1. Open the profile for Company E.
2. Select B2B capabilities.
3. In the B2B capabilities window, update the capabilities by enabling N/A,&DT99724,997,N/A,&X44TA1,TA1 and N/A,EDI-X12,ISA in the order of Package, Protocol and document flow (as shown in Figure 20-11 on page 711). From our earlier configurations, the B2B capability N/A/EDI-X12/ISA has been enabled already.

Profile › Company E › B2B Capabilities

Set Source	Set Target	Enabled	Edit	Document Flow Definition 0 1 2 3 4 All
✓	✓	Enabled		Package: AS
✓	✓	Enabled		Package: None
				Package: Backend Integration (1.0)
✓	✓	Enabled		Package: N/A
✓	✓	Enabled		Protocol: EDI-X12 (ALL)
✓	✓	Enabled		Document Flow: ISA (ALL)
✓	✓	Enabled		Protocol: X12V4R1 (ALL)
				Document Flow: 850 (ALL)
				Document Flow: 855 (ALL)
✓	✓	Enabled		Protocol: &X44TA1 (ALL)
✓	✓	Enabled		Document Flow: TA1 (ALL)
				Protocol: &WDIEVAL (ALL)
				Protocol: &DTCTL21 (ALL)
				Protocol: &DTCTL (ALL)
				Protocol: &DT99933 (ALL)
				Protocol: &DT99737 (ALL)
				Protocol: &DT99735 (ALL)
✓	✓	Enabled		Protocol: &DT99724 (ALL)
✓	✓	Enabled		Document Flow: 997 (ALL)
				Protocol: EDI-Consent (ALL)
				Protocol: EDI-EDIFACT (ALL)

Figure 20-11 Updating the B2B capabilities for the Participant Company E

For the gateway configuration for Company E, there is no need to create another gateway. We can use the default gateway for Company E .

Update the profile of Company A on the Company E server

The B2B capabilities of the participant Company A must be updated with the new EDI FA capabilities on the WebSphere Partner Gateway Enterprise server running on Company E.

While logged in as hubadmin, follow these steps:

1. Open the profile for Company A,
2. Select **B2B capabilities**.
3. In the B2B capabilities window, update the capabilities by enabling N/A,&DT99724,997,N/A,&X44TA1,TA1 and AS,EDI-X12,ISA in the order of Package,Protocol and document flow (as shown in Figure 20-12 on page 712). Because of our earlier configurations, the B2B capability AS/EDI-X12/ISA has already been enabled.

Profile › Company A › B2B Capabilities

Set Source	Set Target	Enabled	Edit	Document Flow Definition (0 1 2 3 4 All)
✓	✓	Enabled		Package: AS
				Protocol: Binary (1.0)
				Protocol: FREEFORM_XML (ALL)
				Protocol: EDI-Consent (ALL)
				Protocol: EDI-EDIFACT (ALL)
✓	✓	Enabled		Protocol: EDI-X12 (ALL)
✓	✓	Enabled		Document Flow: ISA (ALL)
✓	✓	Enabled		Package: None
				Package: Backend Integration (1.0)
✓	✓	Enabled		Package: N/A
				Protocol: EDI-X12 (ALL)
✓	✓	Enabled		Protocol: X12V4R1 (ALL)
✓	✓	Enabled		Protocol: &X44TA1 (ALL)
✓	✓	Enabled		Document Flow: TA1 (ALL)
				Protocol: &WDIEVAL (ALL)
				Protocol: &DTCTL21 (ALL)
				Protocol: &DTCTL (ALL)
				Protocol: &DT99933 (ALL)
				Protocol: &DT99737 (ALL)
				Protocol: &DT99735 (ALL)
✓	✓	Enabled		Protocol: &DT99724 (ALL)
✓	✓	Enabled		Document Flow: 997 (ALL)
				Protocol: EDI-Consent (ALL)
				Protocol: EDI-EDIFACT (ALL)

Figure 20-12 Updating the B2B capabilities for the Participant Company A

As far as the Gateway Configuration for Company A is concerned, there is no need to create another gateway because the default gateway is already present.

This completes the configuration for updating the profiles of the participant. We need to activate the participant connections and set the EDI attributes on the participant connections.

20.3.5 Activate the participant connections and attribute selection

The final step in the configuration for FAs is the creation of participant connections. While logged in as hubadmin, follow these steps:

1. Click **Account Admin** → **Participant Connections**.
2. Select **Company E** as source participant and **Company A** as the target participant. Click **Search.**

3. Three participant connections are required. If not already active, activate by clicking **Activate** .

   a. Participant connections from N/A/&DT99724/997 to N/A/&DT99724/997 (responsible for the generation of FA 997). See Figure 20-13.

Figure 20-13 Enabling the participant connection for the generation of FA997

   b. Participant connection from N/A/&X44TA1/TA1 to N/A/&X44TA1/TA1(responsible for the generation of FA TA1). See Figure 20-14.

Figure 20-14 Enabling the participant connection for the generation of TA1

   c. Participant connection from N/A/EDI-X12/ISA to AS/EDI-X12/ISA (responsible for the enveloping of FA envelope)

   There is no need to enable this participant connection, because it is already present.

As explained in the Chapter 19, “Enveloping and de-enveloping” on page 621, some of the EDI attributes cannot be set in the envelope profile and can be set only at the protocol level in the document flow definitions, or at the connection level. In this example, we set these attributes at the connection level:

1. Click **Attributes** on the Target side for the participant connection from N/A/&X44TA1/TA1 to N/A/&X44TA1/TA1.
2. In the screen, we see the summary of the attributes that are activated. Click the **folder** icon next to the Protocol **&X44TA1** (Figure 20-15 on page 714).

Archived

Envelope Profile	Envelope Profile	No value provided	No value provided
Max validation error level	Max validation error level	0	Inherited from: **Scope:** Global **Type:** Max validation error level
Validation level	Validation level	0	Inherited from: **Scope:** Global **Type:** Validation level
Char set validation table	Char set validation table	CHARSET	Inherited from: **Scope:** Global **Type:** Char set validation table
Alphanumeric validation table	Alphanumeric validation table	ALPHANUM	Inherited from: **Scope:** Global **Type:** Alphanumeric validation table
Generate group level info only in functional Ack	Generate group level info only in functional Ack	No	Inherited from: **Scope:** Global **Type:** Generate group level info only in functional Ack
Century control year	Century control year	10	Inherited from: **Scope:** Global **Type:** Century control year
Detailed validation of segment	Detailed validation of segment	No	Inherited from: **Scope:** Global **Type:** Detailed validation of segment
Connection Profile Qualifier1	Connection Profile Qualifier1		No value provided
Interchange qualifier	Interchange qualifier		No value provided
Interchange Identifier	Interchange Identifier		No value provided
Interchange usage indicator	Interchange usage indicator		No value provided
Group application sender identifier	Group application sender identifier		No value provided
Group application receiver identifier	Group application receiver identifier		No value provided
Interchange reverse routing	Interchange reverse routing		No value provided
Interchange routing address	Interchange routing address		No value provided
Group application sender qualifier	Group application sender qualifier		No value provided
Group application receiver qualifier	Group application receiver qualifier		No value provided
Group application password	Group application password		No value provided
Validation Map	Validation Map		No value provided

Package: N/A (N/A)
Protocol: &X44TA1 (ALL)
Document Flow: TA1 (ALL)

Figure 20-15 Selecting the protocol &X44TA1 in the Target attribute List

3. This leads us to the Attributes page for the protocol &X44TA1.

   For the attribute Envelope Profile, select **WPG-Defined-X12** from the drop-down box (see Figure 20-16 on page 715).

> **Note:** A default set of envelope profiles (one for each EDI protocol) is provided. These profiles are added by default when the product is installed. You can either use one of these profiles or you can create your own profile, if required. Because the creation of an envelope profile is already demonstrated in Chapter 19, “Enveloping and de-enveloping” on page 621, we use the default EDI envelope profile.
>
> As explained in the earlier chapters, none of the envelope profile attributes are mandatory. The mandatory elements are always picked up from the connection level attributes. So, for our scenario, there is no need to fill in any values for the envelope profile **WPG-Defined-X12.**

4. For the attribute Interchange Identifier, enter the value as `companya` (Figure 20-16 on page 715).

		provided		
Segment output	Segment output	Yes	Inherited from: **Scope:** Global **Type:** Segment output	Select one to update
Allow documents with duplicate document ids	Allow documents with duplicate document ids	No	Inherited from: **Scope:** Global **Type:** Allow documents with duplicate document ids	Select one to update
Max error level at Transformation	Max error level at Transformation	0	Inherited from: **Scope:** Global **Type:** Max error level at Transformation	Select one to update
FAMap	FAMap	No value provided	No value provided	Select one to update
Envelope Profile	Envelope Profile	No value provided	No value provided	WPG-Defined-X12
Max validation error level	Max validation error level	0	Inherited from: **Scope:** Global **Type:** Max validation error level	Select one to update WPG-Defined-EDIFACT WPG-Defined-UCS WPG-Defined-X12 Profile_E
Validation level	Validation level	0	Inherited from: **Scope:** Global **Type:** Validation level	
Char set validation table	Char set validation table	CHARSET	Inherited from: **Scope:** Global **Type:** Char set validation table	
Alphanumeric validation table	Alphanumeric validation table	ALPHANUM	Inherited from: **Scope:** Global **Type:** Alphanumeric validation table	
Generate group level info only in functional Ack	Generate group level info only in functional Ack	No	Inherited from: **Scope:** Global **Type:** Generate group level info only in functional Ack	Select one to update
Century control year	Century control year	10	Inherited from: **Scope:** Global **Type:** Century control year	
Detailed validation of segment	Detailed validation of segment	No	Inherited from: **Scope:** Global **Type:** Detailed validation of segment	Select one to update
Connection Profile Qualifier1	Connection Profile Qualifier1	No value provided	No value provided	
Interchange qualifier	Interchange qualifier	No value provided	No value provided	
Interchange Identifier	Interchange Identifier	No value provided	No value provided	companya
Interchange usage indicator	Interchange usage indicator	No value provided	No value provided	
Group application sender identifier	Group application sender identifier	No value provided	No value provided	
Group application receiver identifier	Group application receiver identifier	No value provided	No value provided	
Interchange reverse routing	Interchange reverse routing	No value provided	No value provided	
Interchange routing address	Interchange routing address	No value provided	No value provided	
Group application sender qualifier	Group application sender qualifier	No value provided	No value provided	

Figure 20-16 Setting the attributes for the protocol &X44TA1

5. Click **Save** to save the attributes. Click **Return** to return to the Manage Connections Page.
6. Click the source side **Attributes** for the participant connection N/A/&X44TA1/TA1, and in the screen click the **folder icon** next to the Protocol &X44TA1 and set the attribute Interchange Identifier as **companye.**
7. Repeat the same steps for the connection N/A/&DT99724/997 to N/A/&DT99724/997 for the protocol &DT99724 (Figure 20-17 on page 716).

Archived

Allow documents with duplicate document ids	Allow documents with duplicate document ids	No	**Scope:** Global **Type:** Allow documents with duplicate document ids	Select one to update
Max error level at Transformation	Max error level at Transformation	0	Inherited from: **Scope:** Global **Type:** Max error level at Transformation	Select one to update
FAMap	FAMap	No value provided	No value provided	Select one to update
Envelope Profile	Envelope Profile	No value provided	No value provided	WPG-Defined-X12
Max validation error level	Max validation error level	0	Inherited from: **Scope:** Global **Type:** Max validation error level	Select one to update
Validation level	Validation level	0	Inherited from: **Scope:** Global **Type:** Validation level	Select one to update
Char set validation table	Char set validation table	CHARSET	Inherited from: **Scope:** Global **Type:** Char set validation table	
Alphanumeric validation table	Alphanumeric validation table	ALPHANUM	Inherited from: **Scope:** Global **Type:** Alphanumeric validation table	
Generate group level info only in functional Ack	Generate group level info only in functional Ack	No	Inherited from: **Scope:** Global **Type:** Generate group level info only in functional Ack	Select one to update
Century control year	Century control year	10	Inherited from: **Scope:** Global **Type:** Century control year	
Detailed validation of segment	Detailed validation of segment	No	Inherited from: **Scope:** Global **Type:** Detailed validation of segment	Select one to update
Connection Profile Qualifier1	Connection Profile Qualifier1	No value provided	No value provided	
Interchange qualifier	Interchange qualifier	No value provided	No value provided	
Interchange Identifier	Interchange Identifier	No value provided	No value provided	companya
Interchange usage indicator	Interchange usage indicator	No value provided	No value provided	
Group application sender identifier	Group application sender identifier	No value provided	No value provided	
Group application receiver identifier	Group application receiver identifier	No value provided	No value provided	
Interchange reverse routing	Interchange reverse routing	No value provided	No value provided	
Interchange routing address	Interchange routing address	No value provided	No value provided	
Group application sender qualifier	Group application sender qualifier	No value provided	No value provided	
Group application receiver qualifier	Group application receiver qualifier	No value provided	No value provided	
Group application password	Group application password	No value provided	No value provided	

Figure 20-17 Setting the EDI attributes on the Target side for the protocol &DT99724

Archived

Envelope Profile	Envelope Profile	No value provided	No value provided	Select one to update
Max validation error level	Max validation error level	0	Inherited from: **Scope:** Global **Type:** Max validation error level	Select one to update
Validation level	Validation level	0	Inherited from: **Scope:** Global **Type:** Validation level	Select one to update
Char set validation table	Char set validation table	CHARSET	Inherited from: **Scope:** Global **Type:** Char set validation table	
Alphanumeric validation table	Alphanumeric validation table	ALPHANUM	Inherited from: **Scope:** Global **Type:** Alphanumeric validation table	
Generate group level info only in functional Ack	Generate group level info only in functional Ack	No	Inherited from: **Scope:** Global **Type:** Generate group level info only in functional Ack	Select one to update
Century control year	Century control year	10	Inherited from: **Scope:** Global **Type:** Century control year	
Detailed validation of segment	Detailed validation of segment	No	Inherited from: **Scope:** Global **Type:** Detailed validation of segment	Select one to update
Connection Profile Qualifier1	Connection Profile Qualifier1	No value provided	No value provided	
Interchange qualifier	Interchange qualifier	No value provided	No value provided	
Interchange Identifier	Interchange Identifier	No value provided	No value provided	companye
Interchange usage indicator	Interchange usage indicator	No value provided	No value provided	
Group application sender identifier	Group application sender identifier	No value provided	No value provided	
Group application receiver identifier	Group application receiver identifier	No value provided	No value provided	
Interchange reverse routing	Interchange reverse routing	No value provided	No value provided	
Interchange routing address	Interchange routing address	No value provided	No value provided	
Group application sender qualifier	Group application sender qualifier	No value provided	No value provided	
Group application receiver qualifier	Group application receiver qualifier	No value provided	No value provided	
Group application password	Group application password	No value provided	No value provided	

Save Close

Document Flow: 997 (ALL)

Figure 20-18 Setting the EDI attributes on the Source side for the protocol &DT99724

8. On the participant connection N/A/EDI-X12/ISA to AS/EDI-X12/ISA, the AS MDN Requested has to be set to **No**.

This finishes the configuration required on WebSphere Partner Gateway Enterprise running at Company E for the generation of Functional Acknowledgements.

20.4 Configure Company A for receiving EDI

No additional configuration is required for WebSphere Partner Gateway Advanced.

Archived

20.5 Sending and validating the documents

The configuration required on both the channels has been finished. We now need to validate the communication by routing the EDI-X12 interchanges from Company A to Company E.

We will be routing three different EDI-X12 input interchanges so we can see the generation of FAs in three different scenarios.

- Scenario1

  Routing an EDI-X12 interchange from Company A to Company E with single group and single transaction which requests for TA1(the ISA 14 is set to 1)

- Scenario2

  Routing an EDI-X12 interchange from Company A to Company E that consists of four groups which request for TA1(the ISA14 is set to 1)

- Scenario3

  Routing a defective EDI-X12 interchange from Company A to Company E with a single group and multiple transactions which doesn't request for TA1(The ISA14 is set to 0)

In this section, we examine these scenarios in detail.

20.5.1 Scenario1

In this scenario, the following EDI interchange in Example 20-7 will be routed from Company A to Company E.

Example 20-7 EDI Interchange for Scenario1

```
ISA*00*          *00*          *  *companya       *  *companye
*241002*1635*U*00401*000000014*1*P*z!
GS*  *  *  *241002*1635*000000014* *004010!
ST*855*000000014!
BAK*06*AT*P365542*01101402!
PO1**132*EA*10.56*PE*BP*00031PO241!
PID*F****SCSI Card!
PO1**16*EA*1.99*PE*BP*00021PO241!
PID*F****IDE Card!
SE*7*000000014!
GE*1*000000014!
IEA*1*000000014!
```

From this interchange we can observe that the ISA14 segment is set to 1 which means that the request for TA1 is turned on.

Following are the steps to route this envelope from Company A to Company E and to route the FA envelope back to Company A.

On Company A

1. Place the above EDI envelope in the File Directory Receiver of Company A in the directory /home/bcguser/wpgv6/data/companya/edi_out.
2. While logged in as hubadmin, click **Viewers** → **Document Viewers** and Search for the Documents from the participants Company A to Company E. We see the EDI envelope being posted to Company E (Figure 20-19).

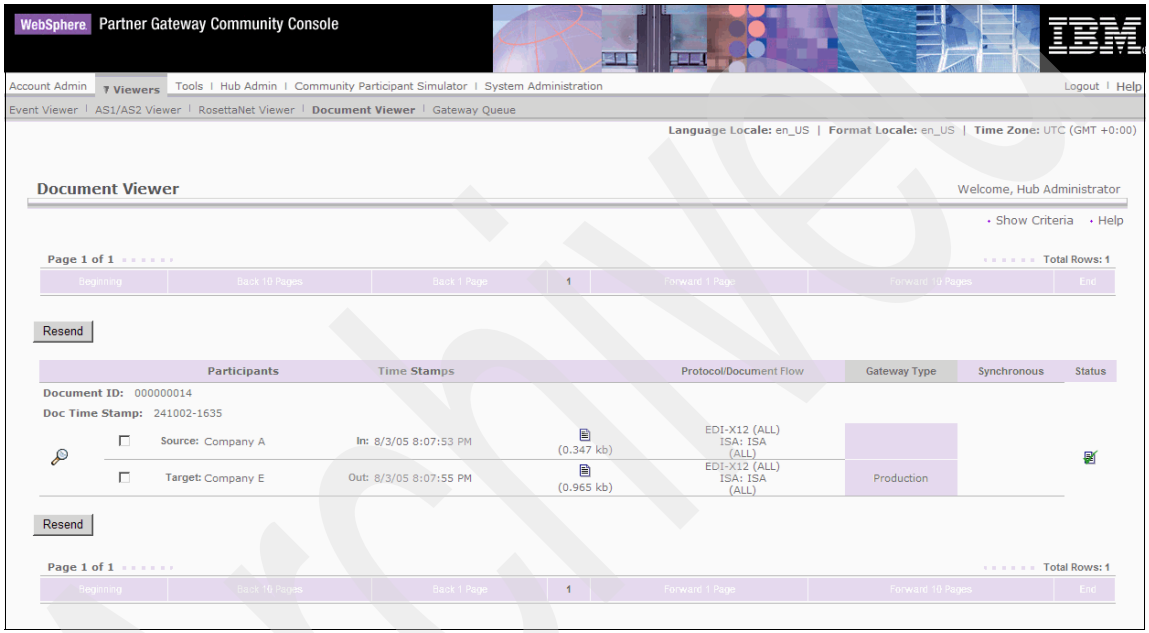


Figure 20-19 EDI envelope posted from Company A to Company E on Company A server

On Company E

1. While logged in as hubadmin, click **Viewers** → **Document Viewers** and search for the documents. We are able to see the documents in the Document Viewers (Figure 20-20 on page 720).

Archived

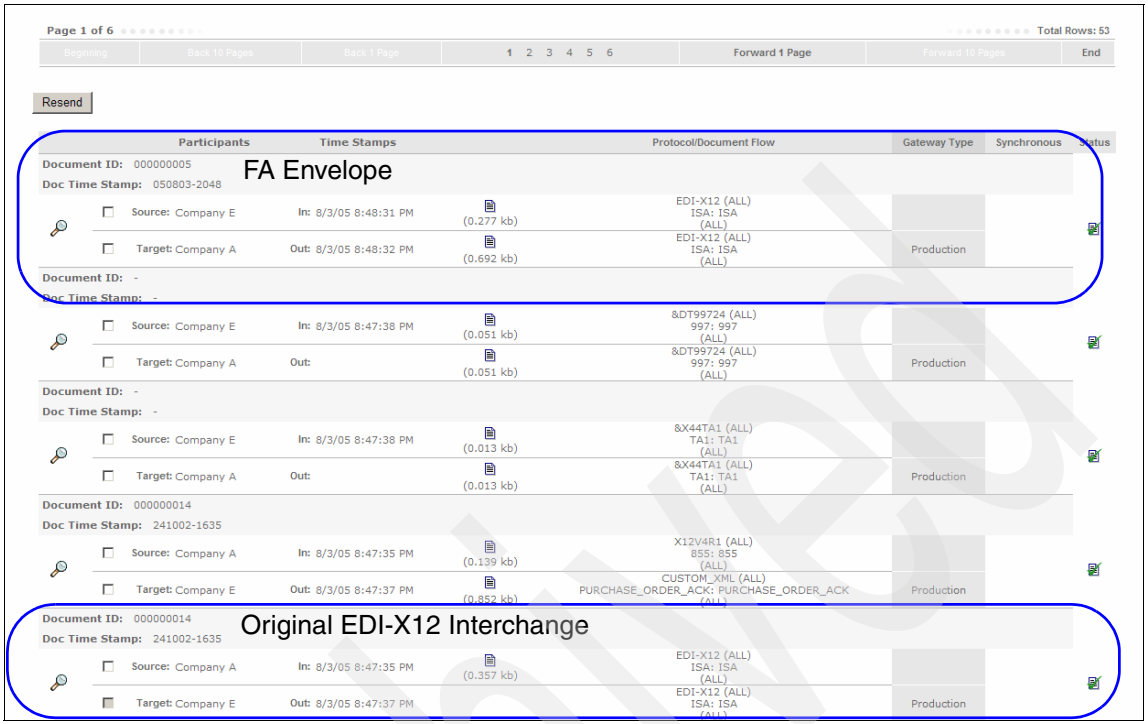


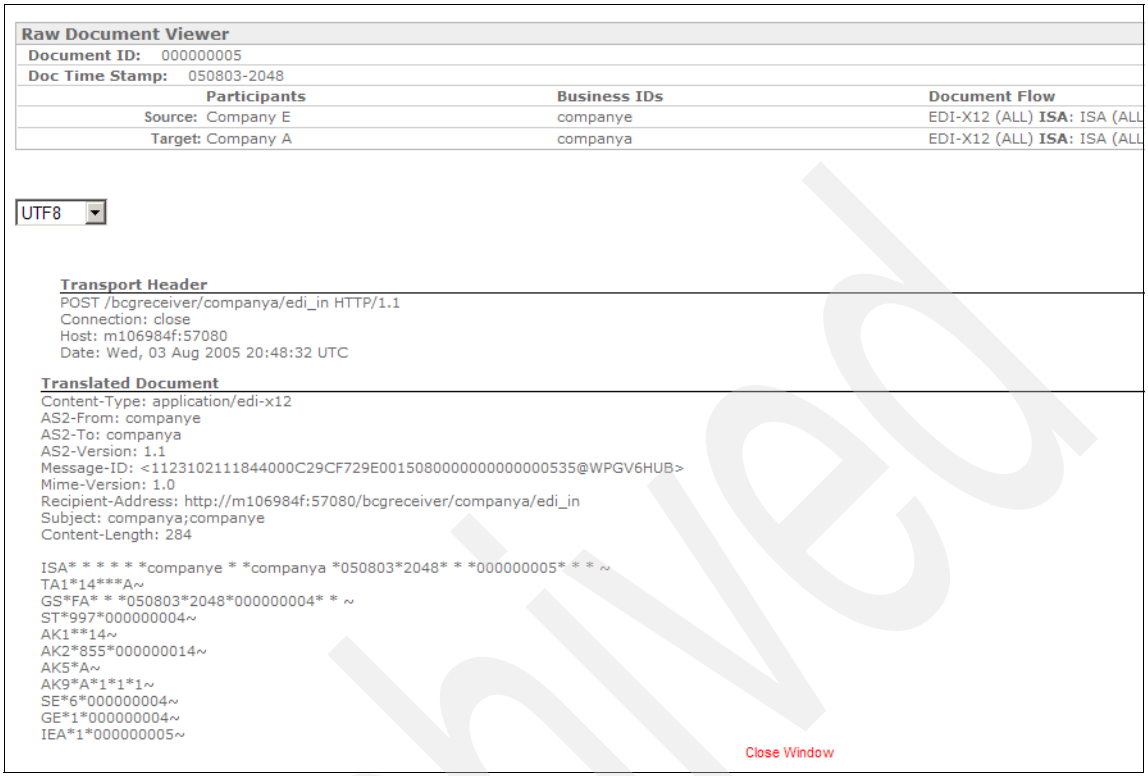
Figure 20-20 Original EDI envelope along with the FAs generated and FA envelope being observed in the Document Viewers

> **Note:** For easier understanding of the scenario, the original EDI document, the FA transactions and the FA envelope were shown in the same page. If you want to view only the generated FAs and FA envelope, click **Viewers** → **Document Viewers**. In the search criteria, select the source participant as **Company E** and target participant as **Company A**.

2. To view the generated FA envelope, in the Document Viewers, click the raw document (Figure 20-21).

Figure 20-21 Clicking the Raw document in the Document Viewers

3. We see the following output from the raw document viewer (Figure 20-22).



Raw Document Viewer

Document ID: 000000005

Doc Time Stamp: 050803-2048

	Participants	Business IDs	Document Flow
Source:	Company E	companye	EDI-X12 (ALL) **ISA**: ISA (ALL
Target:	Company A	companya	EDI-X12 (ALL) **ISA**: ISA (ALL

UTF8

Transport Header

```
POST /bcgreceiver/companya/edi_in HTTP/1.1
Connection: close
Host: m106984f:57080
Date: Wed, 03 Aug 2005 20:48:32 UTC
```

Translated Document

```
Content-Type: application/edi-x12
AS2-From: companye
AS2-To: companya
AS2-Version: 1.1
Message-ID: <1123102111844000C29CF729E0015080000000000000535@WPGV6HUB>
Mime-Version: 1.0
Recipient-Address: http://m106984f:57080/bcgreceiver/companya/edi_in
Subject: companya;companye
Content-Length: 284

ISA* * * * * *companye * *companya *050803*2048* * *000000005* * * ~
TA1*14***A~
GS*FA* * *050803*2048*000000004* * ~
ST*997*000000004~
AK1**14~
AK2*855*000000014~
AK5*A~
AK9*A*1*1*1~
SE*6*000000004~
GE*1*000000004~
IEA*1*000000005~
```

Close Window

Figure 20-22 EDI FA envelope that got generated by WebSphere Partner Gateway being observed in the raw Document viewers.

4. From Figure 20-22, note the following:
   a. The TA1 transaction is always placed in between ISA and GS segments.
   b. TA1 is showing the status A which means the envelope was accepted.
   c. The AK1 and AK9 segments which report for the whole group and AK2 and AK5 segments which report the reception of individual transactions.

This completes scenario1. In Scenario2, an EDI envelope with multiple groups will be posted.

20.5.2 Scenario2

In this scenario, the following EDI interchange (Example 20-8) with multiple groups will be posted to WebSphere Partner Gateway Enterprise from Company A.

Archived

Example 20-8 EDI Interchange for Scenario2

```
ISA*00*          *00*          *  *companya       *  *companye
*241002*1635*U*00401*000000001*1*P*z!
GS*  *  *  *241002*1635*000000001* *004010!
ST*855*000000014!
BAK*06*AT*P365542*01101402!
PO1**132*EA*10.56*PE*BP*00031P0241!
PID*F****SCSI Card!
PO1**16*EA*1.99*PE*BP*00021P0241!
PID*F****IDE Card!
SE*7*000000014!
GE*1*000000001!
GS*  *  *  *241002*1635*000000002* *004010!
ST*855*000000014!
BAK*06*AT*P365542*01101402!
PO1**132*EA*10.56*PE*BP*00031P0241!
PID*F****SCSI Card!
PO1**16*EA*1.99*PE*BP*00021P0241!
PID*F****IDE Card!
SE*7*000000014!
GE*1*000000002!
GS*  *  *  *241002*1635*000000003* *004010!
ST*855*000000014!
BAK*06*AT*P365542*01101402!
PO1**132*EA*10.56*PE*BP*00031P0241!
PID*F****SCSI Card!
PO1**16*EA*1.99*PE*BP*00021P0241!
PID*F****IDE Card!
SE*7*000000014!
GE*1*000000003!
GS*  *  *  *241002*1635*000000004* *004010!
ST*855*000000014!
BAK*06*AT*P365542*01101402!
PO1**132*EA*10.56*PE*BP*00031P0241!
PID*F****SCSI Card!
PO1**16*EA*1.99*PE*BP*00021P0241!
PID*F****IDE Card!
SE*7*000000014!
GE*1*000000004!
IEA*4*000000001!
```

1. The above EDI interchange in Example 20-8 on page 722 will be posted to Company E by placing it in the file directory target of Company A.
2. We can observe the following entries in the document viewers after the document is received at Company E (Figure 20-23).

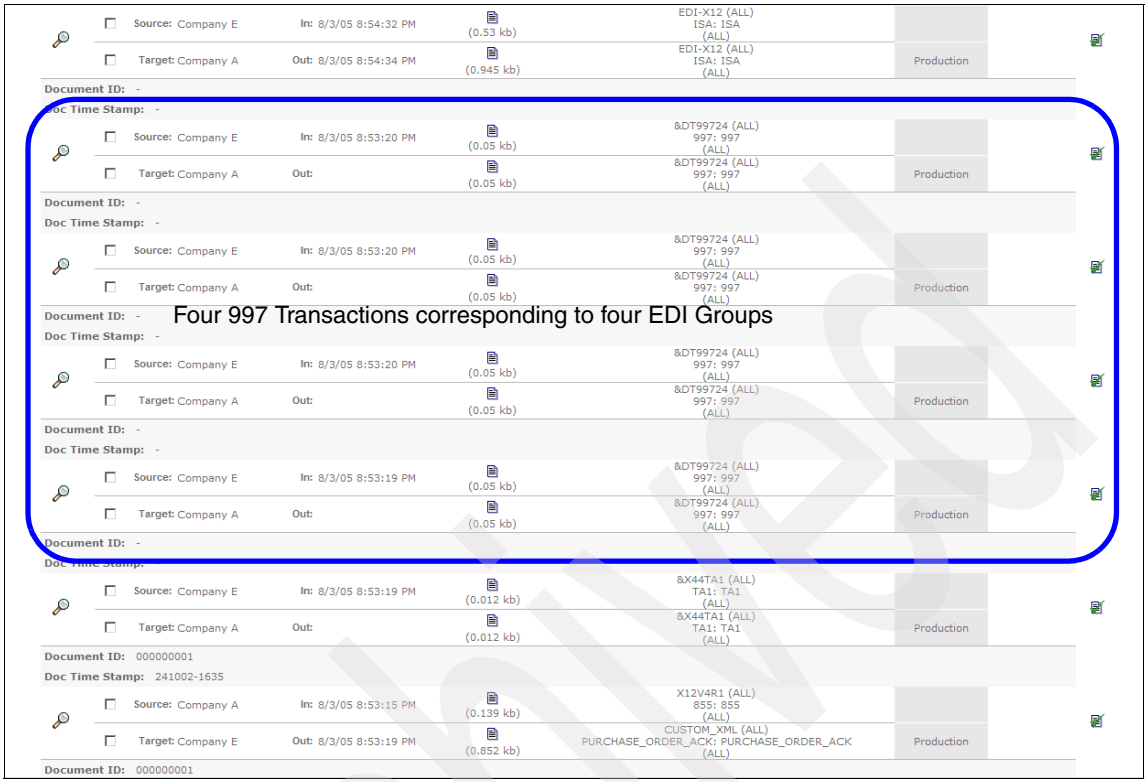
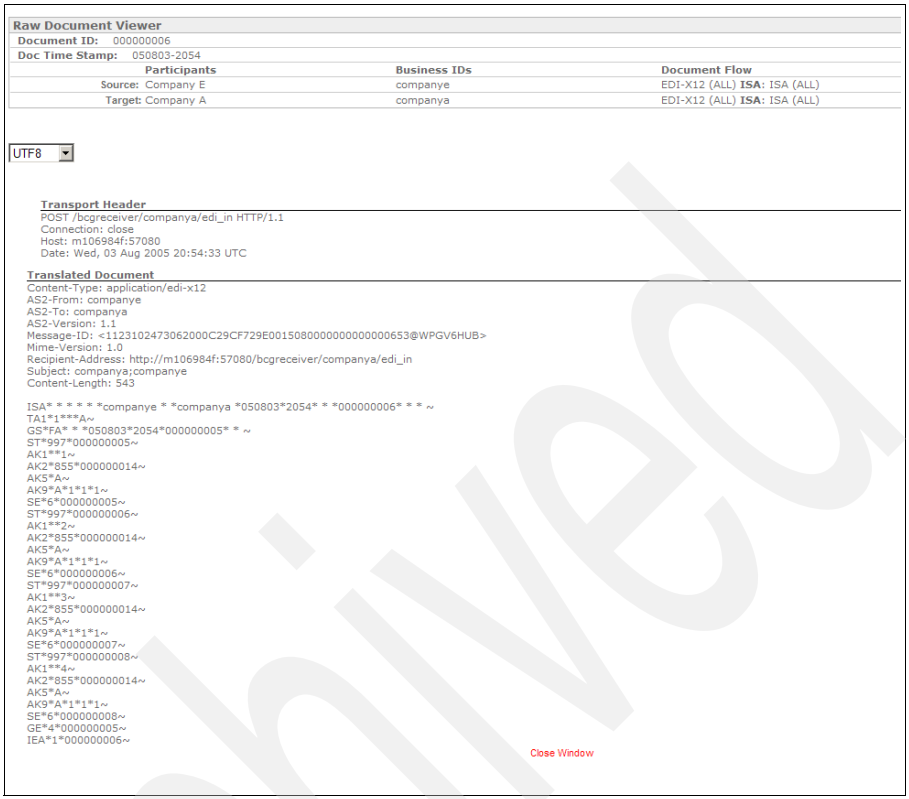


Figure 20-23 Four Functional Acknowledgements((97 Transactions)corresponding to four EDI groups

3. Looking at Figure 20-23, we can learn the following:
   a. As the original EDI input document has four groups and 997 being a group level EDI FA, we see four 997s in the document viewers.
   b. A TA1 is generated as the input document's ISA14 is set to 1.
4. In the raw document viewer for the FA envelope, we see Figure 20-24 on page 724.

Archived



Raw Document Viewer

Document ID: 000000006

Doc Time Stamp: 050803-2054

	Participants	Business IDs	Document Flow
Source:	Company E	companye	EDI-X12 (ALL) ISA: ISA (ALL)
Target:	Company A	companya	EDI-X12 (ALL) ISA: ISA (ALL)

UTF8

Transport Header

```
POST /bcgreceiver/companya/edi_in HTTP/1.1
Connection: close
Host: m106984f:57080
Date: Wed, 03 Aug 2005 20:54:33 UTC
```

Translated Document

```
Content-Type: application/edi-x12
AS2-From: companye
AS2-To: companya
AS2-Version: 1.1
Message-ID: <1123102473062000C29CF729E001508000000000000653@WPGV6HUB>
Mime-Version: 1.0
Recipient-Address: http://m106984f:57080/bcgreceiver/companya/edi_in
Subject: companya;companye
Content-Length: 543

ISA* * * * * *companye * *companya *050803*2054* * *000000006* * * ~
TA1*1***A~
GS*FA* * *050803*2054*000000005* * ~
ST*997*000000005~
AK1**1~
AK2*855*000000014~
AK5*A~
AK9*A*1*1~
SE*6*000000005~
ST*997*000000006~
AK1**2~
AK2*855*000000014~
AK5*A~
AK9*A*1*1~
SE*6*000000006~
ST*997*000000007~
AK1**3~
AK2*855*000000014~
AK5*A~
AK9*A*1*1~
SE*6*000000007~
ST*997*000000008~
AK1**4~
AK2*855*000000014~
AK5*A~
AK9*A*1*1~
SE*6*000000008~
GE*4*000000005~
IEA*1*000000006~
```

Close Window

Figure 20-24 Raw document of FA Envelope

5. We see that the envelope in Figure 20-24 is reporting all the transactions inside each group.

This finishes the analysis for Scenario2.

20.5.3 Scenario3

In this scenario, a malformed EDI envelope is routed to Company E and we observe the FA reporting mechanism under these circumstances.

> **Note:** For this scenario, TA1 is not requested.

The following document in Example 20-9 on page 708 will be posted from Company A to Company E.

Archived

Example 20-9 EDI Interchange for Scenario3

```
ISA*00*          *00*          *  *companya       *  *companye
*241002*1635*U*00401*000000001*0*P*z!
GS*  *  *  *241002*1635*000000001* *004010!
ST*855*000000001!
BAK*06*AT*P365542*01101402!
PO1**132*EA*10.56*PE*BP*00031P0241!
PID*F****SCSI Card!
PO1**16*EA*1.99*PE*BP*00021P0241!
PID*F****IDE Card!
SE*7*000000001!
ST*855*000000002!
BAK*06*AT*P365542*01101402!
PO1**132*EA*10.56*PE*BP*00031P0241!
PID*F****SCSI Card!
PO1**16*EA*1.99*PE*BP*00021P0241!
PID*F****IDE Card!
SE*7*000000002!
ST*855*000000003!
BAK*06*AT*P365542*01101402!
PO1**132*EA*10.56*PE*BP*00031P0241!
PID*F****SCSI Card!
PO1**16*EA*1.99*PE*BP*00021P0241!
PID*F****IDE Card!
SE*7*000000003!
ST*855*000000004!
BAK*06*AT*P365542*01101402!
PO1**132*EA*10.56*PE*BP*00031P0241!
PID*F****SCSI Card!
PO1**16*EA*1.99*PE*BP*00021P0241!
PID*F****IDE Card!
SE*7*000000077!
ST*855*000000005!
BAK*06*AT*P365542*01101402!
PO1**132*EA*10.56*PE*BP*00031P0241!
PID*F****SCSI Card!
PO1**16*EA*1.99*PE*BP*00021P0241!
PID*F****IDE Card!
SE*7*000000088!
GE*5*000000001!
IEA*1*000000001!
```

From the document in Example 20-9,we see that SE segment for the 4th and 5th transaction are erroneous.

1. Post this document from Company A to Company E, as in the previous two scenarios.

Archived

2. Observe the document Viewers at Company E (Figure 20-25).

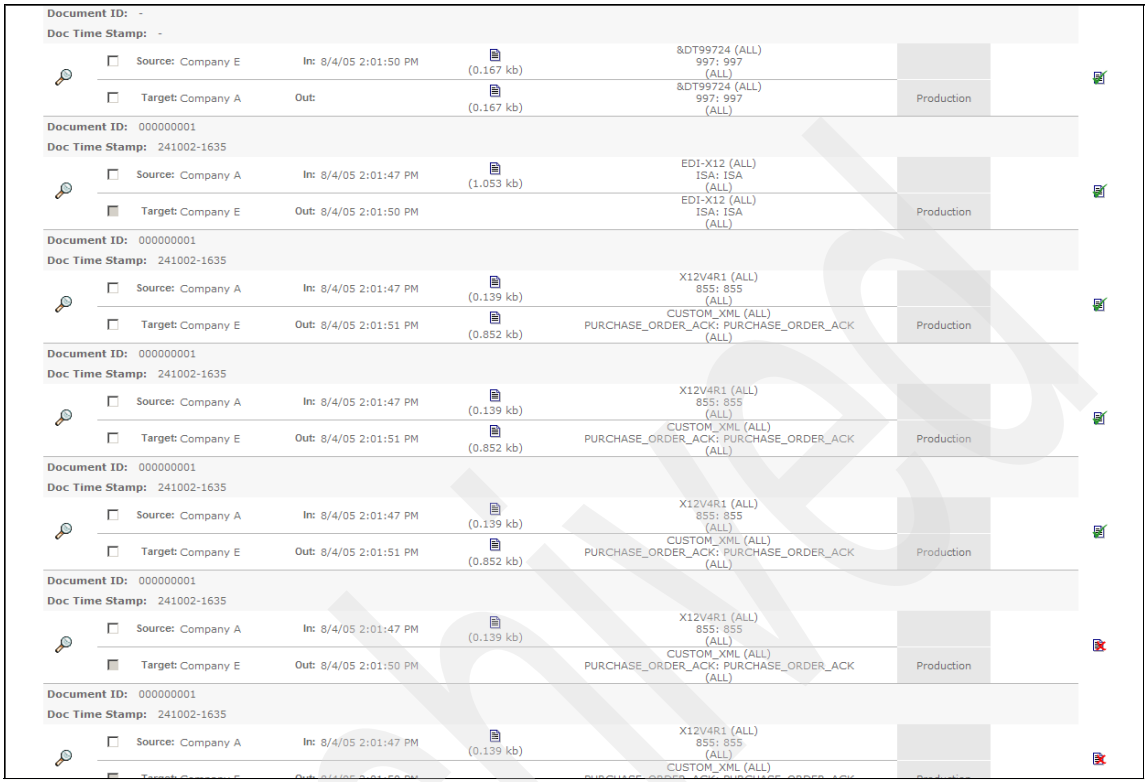
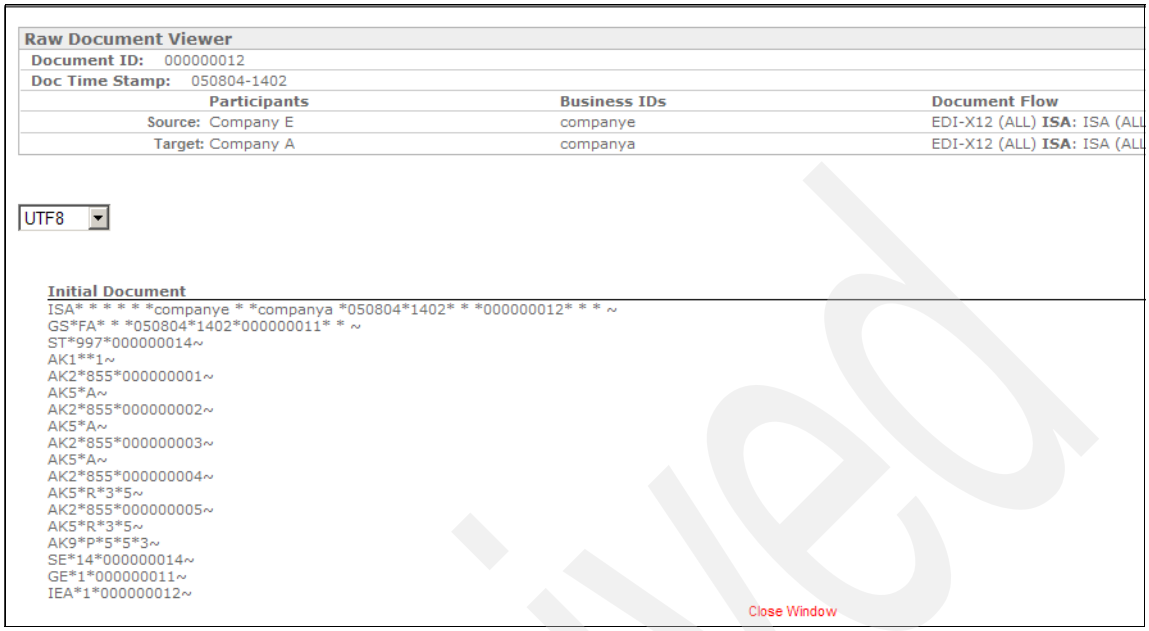


Figure 20-25 Document Viewers for Scenario3

3. From Figure 20-25, we see that the erroneous transactions are failed at WebSphere Partner Gateway Enterprise on Company E.
4. We now look at the raw document viewer for the FA envelope that was generated.

Archived



Raw Document Viewer

Document ID: 000000012

Doc Time Stamp: 050804-1402

	Participants	Business IDs	Document Flow
Source:	Company E	companye	EDI-X12 (ALL) **ISA:** ISA (ALL
Target:	Company A	companya	EDI-X12 (ALL) **ISA:** ISA (ALL

UTF8

Initial Document

```
ISA* * * * * *companye * *companya *050804*1402* * *000000012* * * ~
GS*FA* * *050804*1402*000000011* * ~
ST*997*000000014~
AK1**1~
AK2*855*000000001~
AK5*A~
AK2*855*000000002~
AK5*A~
AK2*855*000000003~
AK5*A~
AK2*855*000000004~
AK5*R*3*5~
AK2*855*000000005~
AK5*R*3*5~
AK9*P*5*5*3~
SE*14*000000014~
GE*1*000000011~
IEA*1*000000012~
```

Close Window

Figure 20-26 Raw document of EDI FA 997 Transaction for Scenario3

a. The AK901 segment shows an entry P which shows that the document is Partially accepted.

b. The AK904 segment shows an entry 3 which means that out of five transactions only three were accepted.

c. The AK5 segments for transactions 4 and 5 which shows that they are erroneous.

Archived

Archived

21

Batching and EDI splitting

In this chapter we discuss some additional features of WebSphere Partner Gateway. The splitting and batching of documents are used for flows involving transformation and EDI. We also use some of the skills covered previously in this redbook to create a new example flow, EDIFACT to X12, and provide output tailored to specific Participant preferences.

Archived

© Copyright IBM Corp. 2005. All rights reserved.

21.1 Batching and splitting basics

In Chapter 19, “Enveloping and de-enveloping” on page 621, we discussed the enveloping and de-enveloping of EDI documents. The batching feature of WebSphere Partner Gateway is closely linked to this enveloper function and is used to configure the manner in which the enveloper groups the EDI transactions or messages together into envelopes. Batching can preserve the batch in which documents are received or group them together based on the enveloper schedule function.

The EDI splitter is basically a way to separate large files containing multiple EDI envelopes that have been concatenated together. The Document Manager component can then process them successfully.

21.2 EDI Batching scenario

In this section, we review EDI batching.

21.2.1 Batching Scenario Overview

The batching function is actually a parameter in the EDI enveloper that was briefly discussed in Chapter 19, “Enveloping and de-enveloping” on page 621. Any inbound flow to WebSphere Partner Gateway involving transformation that results in an EDI document outbound will be enveloped and therefore affected by the batching setting. This could be ROD to EDI, XML to EDI, or EDI to EDI transformations. We have covered some ROD and XML transformation in Chapter 15, “Mapping” on page 473 and Chapter 19, “Enveloping and de-enveloping” on page 621 respectively. In this chapter we will define a new flow, an EDIFACT to EDI-X12 transformation, and use this example to illustrate the batching and splitting functions.

To WebSphere Partner Gateway, an EDI envelope is considered a **batch** of EDI transactions or messages. The goal of the batching function in this example will be to:

- Accept multiple inbound EDIFACT envelopes
- De-envelope the messages
- Transform each into an EDI-X12 transaction
- Envelope the resultant transactions into an ISA envelope
- Deliver to the Target Participant based on the batching parameter and the enveloper schedule.

Figure 21-1 on page 731 is a high-level diagram of what you configure in this chapter.

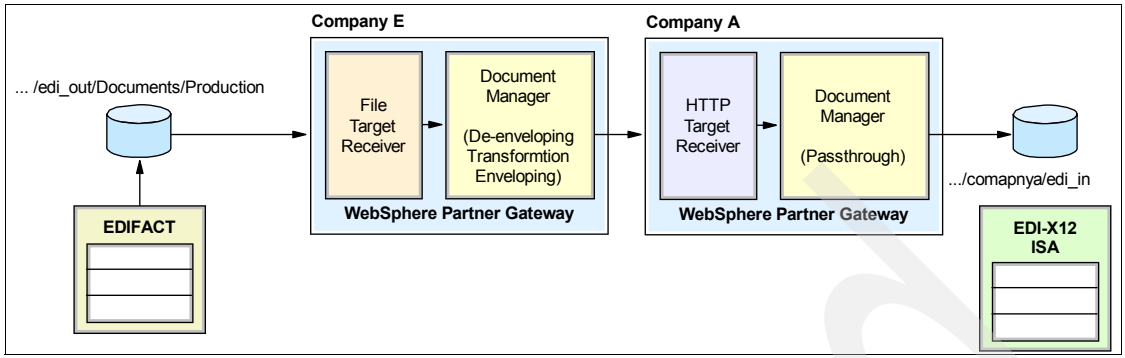


Figure 21-1 Batching scenario overview

Company E has been used to demonstrate several different protocols and packaging. For this example we add a new protocol, EDIFACT, to this varied list of supported protocols. Company E will pick up the EDIFACT envelopes locally and transform them into EDI-X12 envelopes for transmission. In this example, we are more concerned about tailoring the files as they exit Company E than the recipient of the resulting envelopes. Company A will fill the receiving participant role. Company A, prefers their documents as EDI-X12 standard packaged in AS. Therefore, Company E will have to transform the EDIFACT documents into EDI-X12 using the mapping skills covered in Chapter 15, “Mapping” on page 473, then send them out in AS packaging.

In terms of the B2B exchange configured so far in this redbook, you can think of Company E extending its operations into Europe, where EDIFACT is prevalent. Company E might have new back-end applications that take orders using EDIFACT standard but their partner, Company A, still require EDI-X12 for B2B communication.

Some of the configuration objects required for this example have been created and discussed in previous chapters. For this EDIFACT to EDI-X12 outbound flow, we discuss the steps required to define a new outbound communication between these WebSphere Partner Gateway participants.

21.2.2 Configuration for Company E

The hub administrator of Company E has several steps he needs to accomplish for this new flow. In Figure 21-1, you can see the basic pieces we need to complete this example. All of the steps have been covered in examples throughout the redbook, so it should seem familiar. For completeness, we have reviewed and explained each step in the process here:

1. A target is needed to pick up the files locally from Company E. The **`FileTarget`** created in Chapter 7, “Creating a basic B2B exchange” on page 137 is perfect for this task. The edi_out local directory is not EDI standard-specific and can pick up EDIFACT documents just as well as the EDI-X12 documents from previous examples.
2. Create and upload a new or existing map, EDIFACT to EDI-X12, into the DIS client.
3. From the DIS client, upload the map into the Company E WebSphere Partner Gateway. The new B2B capabilities will be uploaded along with the map as explained in Chapter 15, “Mapping” on page 473.
4. Update the B2B capabilities of the Company E Community Manager to reflect the source side of the three interactions that will be defined.
5. Update the B2B capabilities of the Company A Community Participant to reflect the target side of the three interactions that will be defined.
6. A gateway for Company A, the destination of the flow, is also required. This is **`HTTPGateway`** that already exists Company A’s profile. It was used in Chapter 7, “Creating a basic B2B exchange” on page 137 and Chapter 19, “Enveloping and de-enveloping” on page 621.
7. Define the interactions between Company E and Company A. We need these three:
   - None/EDIFACT/UNB to NA/EDIFACT/UNB, for De-enveloping the EDIFACT message.
   - NA/EDI00B/ORDERS to NA/X12V4R1/850, for transforming the EDIFACT EDI00B messages into EDI-X12 850 transactions.
   - NA/EDIX12/ISA to AS/EDIX12/ISA, for enveloping the resulting transactions back into EDI-X12 envelope

     This is where batching takes place.
8. Activate the new interactions.
9. Finally, configure and review the activated participant connections.

Reviewing the target

The EDIFACT order messages that Company E wants to transform and send to Company A need to be picked up from locally from Company E. In Chapter 7, “Creating a basic B2B exchange” on page 137 we created the **`FileTarget`** on Company E (Figure 21-2 on page 733). This target can be used for this example as well because it does not distinguish between the EDI standards. Company E can use this target to drop all of their EDI documents for outbound transmission, regardless of standard used.

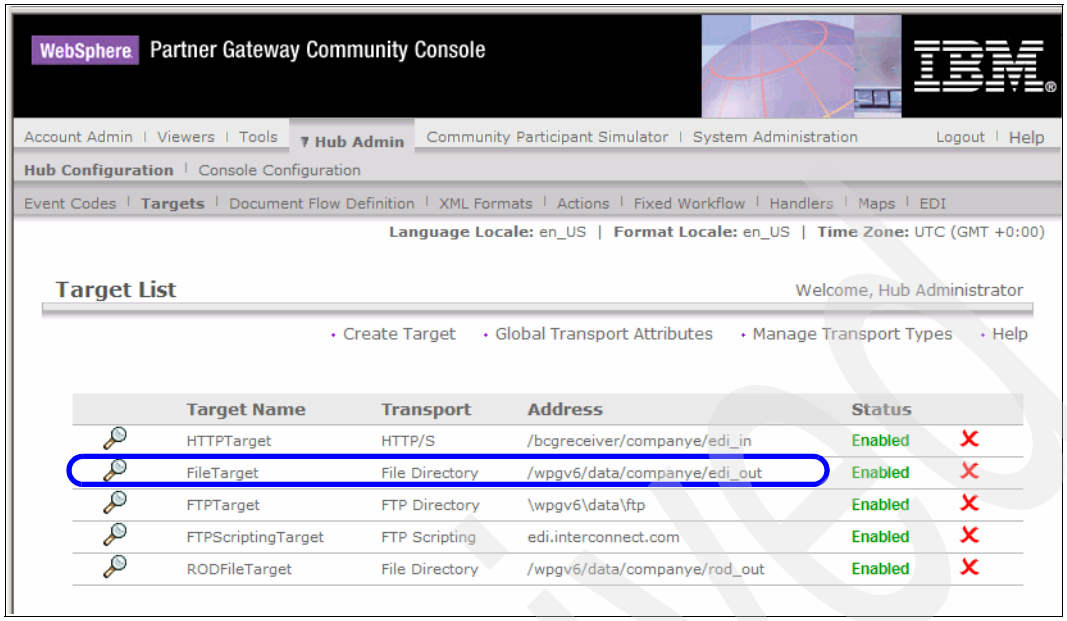


Figure 21-2 File Directory Target on Company E used to pickup EDIFACT documents

The EDIFACT to EDI-X12 map

The map needed here greatly depends on the specific requirements of both Company E and Company A, and the types of documents they are exchanging. For example purposes, we use a relatively simple map that maps all of the required EDIFACT standards of an EDI00B ORDERS message into an EDI-X12 X12V4R1 850 transaction. Chapter 15, “Mapping” on page 473 contains additional information about creating maps. A brief summary of the steps for this example are as follows:

1. Open the DIS client program. Open the Development database using **File** → **Open Functional Area**
2. Select **Development** and **Mapping**, then click **OK**.
3. Import the map from the EDIFACT_X12_PO.eif file, which is located in the Additional Materials. The map should be loaded into the DIS client Development database. To upload the map, follow these steps:
   a. Select **File** → **Open Import File**, then browse to the location of the **EDIFACT_X12_PO.eif** file. Click **Open**.
   b. In the Import File Window (Figure 21-3 on page 734), select the map.

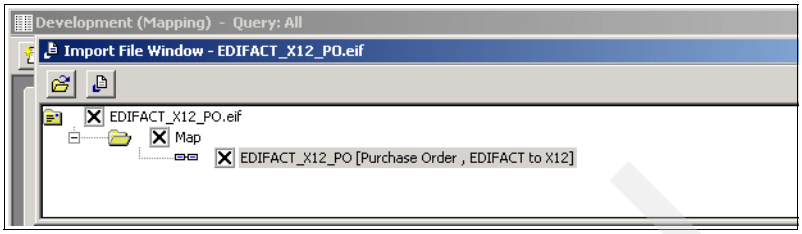


Figure 21-3 Loading the sample map into the DIS client database

c. Click **Import Selected Documents**, and choose the **Development** database.

d. When the import is complete, click **Close** on the Execution Status window.

4. After the map is loaded, it needs to be compiled. From the Development database view, **Data Transformation Map** tab, highlight **EDIFACT_X12_PO** and select **Compile** (Figure 21-4).

> **Note:** You might need to refresh the view to see the map in the Data Transformation Maps tab of the DIS client.

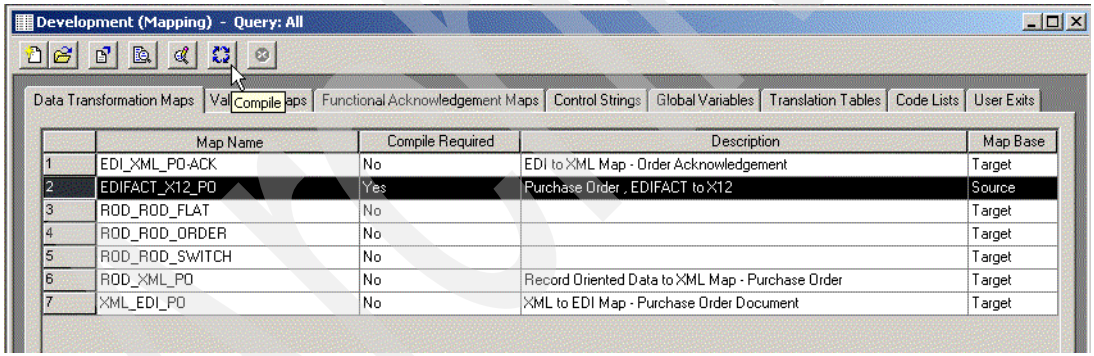


Figure 21-4 Compiling the sample map

After selecting **Compile**, look for the Execution Status window shown in Figure 21-5 on page 735.

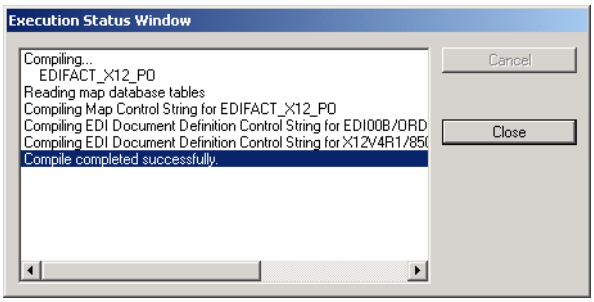


Figure 21-5 Successful compilation message

You now have a compiled EDIFACT to EDIX12 map that will transform an EDIFACT document. Now you need to transfer that map to the WebSphere Partner Gateway database.

Loading the Map into WebSphere Partner Gateway

When the map is loaded and compiled, you need to load the map into the database of WebSphere Partner Gateway. This will create the new document flow definitions that are needed by both Company E and Company A.

Here are the steps to load the map into the database using the ODBC connection created in Chapter 15, “Mapping” on page 473:

1. With the DIS client open, click the **Control Strings** tab.
2. Highlight the **EDIFACT_X12_PO control string ()** and select **Actions** → **Export To Other Database....** See Figure 21-6 on page 736.

Archived

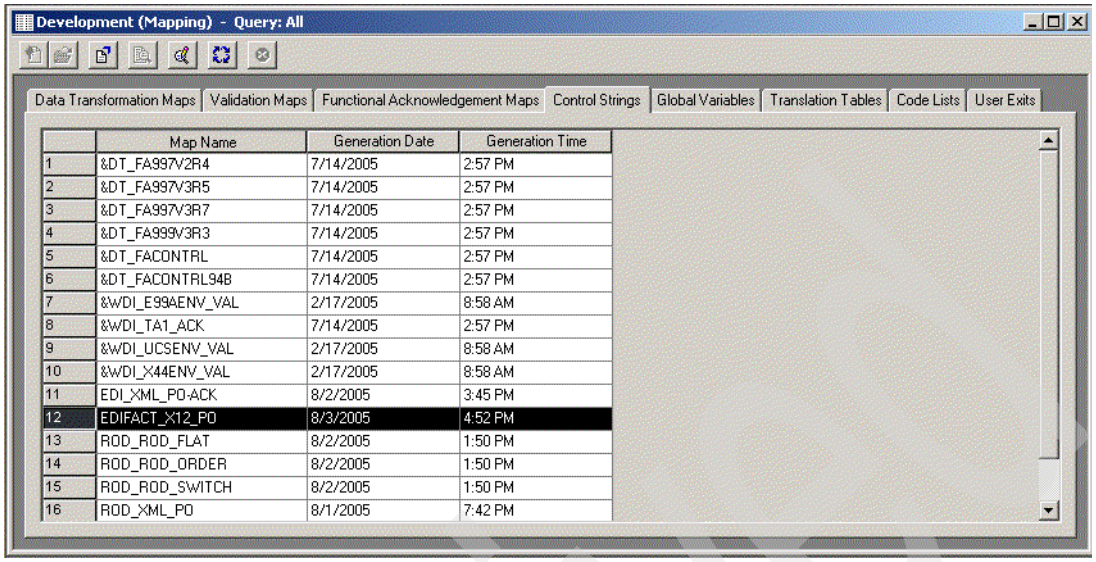


Figure 21-6 Selecting the Control String for export

3. In the Select a Database window, select the WebSphere Partner Gateway database, for example **WPGV6DB** (Figure 21-7).

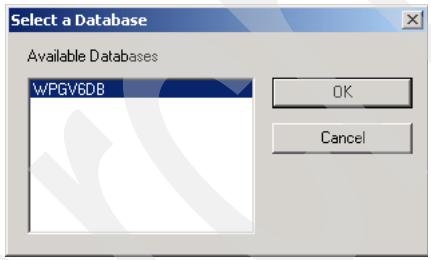


Figure 21-7 Selecting the WebSphere Partner Gateway Database

4. Provide a user ID and password with the authority to update the WPGV6DB in the Connect to DB2 Database window. Db2admin or bcgcon users should have sufficient access.
5. In the Export with Control String window, check **Code Lists** under Referenced Types (Figure 21-8 on page 737), then click **OK**.

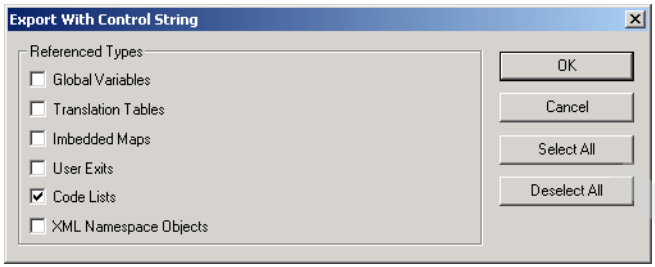


Figure 21-8 Referenced type selection

6. Close the Execution Status window after the export completes successfully.
7. (Optional) To verify the map was loaded into WebSphere Partner Gateway, open the Community Console of Company E and select **Hub Admin** → **Hub Configuration** → **Maps** → **Transformation Maps**. You should see the EDIFACT_X12_PO listed, as in (Figure 21-9).

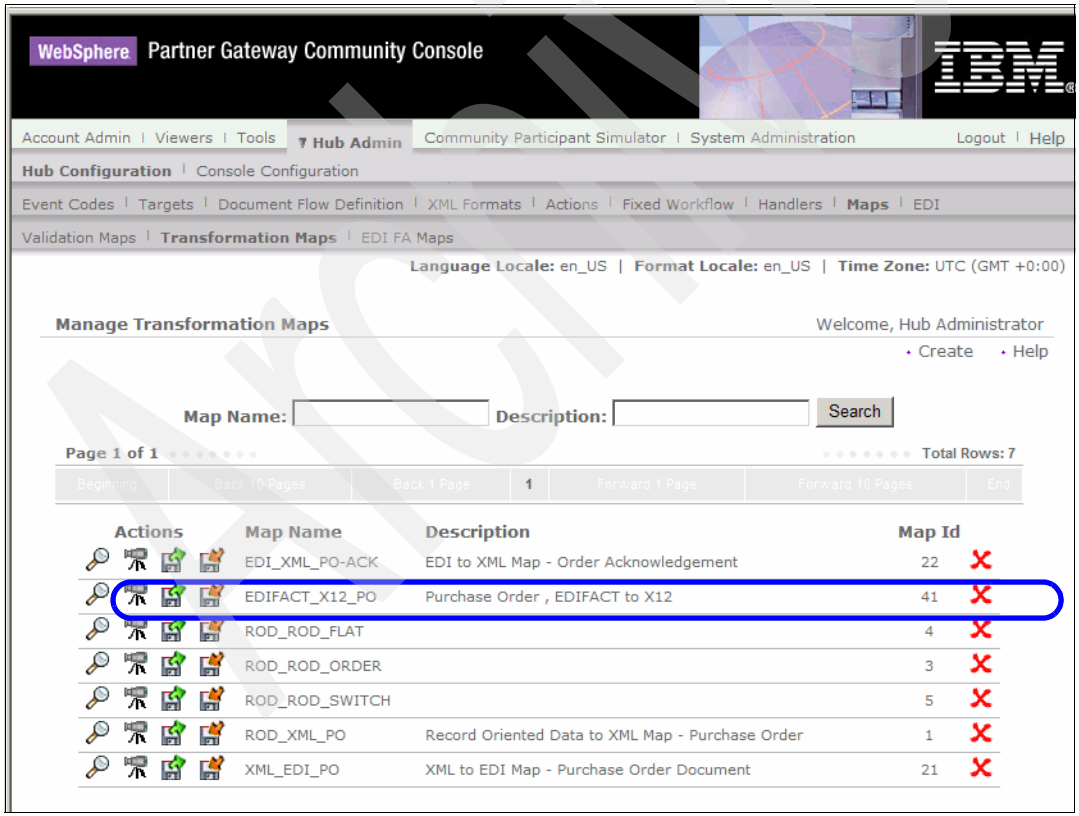


Figure 21-9 Transformation map loaded into WebSphere Partner Gateway

Archived

By loading the map into WebSphere Partner Gateway, some new document flow definitions, and specifically NA/EDI00B/ORDERS, have been added. To continue with the configuration of this example, enable the correct B2B capabilities of both Company E and Company A profiles.

Updating the Company E profile B2B capabilities

The Company E profile Community Manager will need to enable the new B2B capabilities. As Community Manager, you must enable the B2B capabilities at Company E for all three interactions that are needed. The Company E Community Manger must have the source side of these interactions enabled so that the participant connection can be activated. The three interactions are:

- None/EDIFACT/UNB to NA/EDIFACT/UNB
- NA/EDI00B/ORDERS to NA/X12V4R1/850
- NA/EDIX12/ISA to AS/EDIX12/ISA

To do this, follow these steps:

1. As the hub administrator (hubadmin) user on the Community Console of Company E, select **Account Admin** → **Profiles** → **Community Participant**.
2. Click **Search**.
3. From the list of configured participants, click the **magnifying glass** icon next to **Company E**.
4. Within the Company E profile, click **B2B capabilities**.
5. On this B2B capabilities screen, there are three package/protocol/document flow sets that need to be enabled to match the interactions we will create later. The steps to do this are:

   Note: Some of these capabilities will already be enabled for both Set Source and Set Target from earlier examples. If they are already enabled, you do not need to take additional action.

   a. Enable Set Source and Set Target for **Package: None** and then expand **Package: None** using the folder icon**.**
   b. Enable Set Source and Set Target for **Protocol: EDI-EDIFACT**, and then expand **Protocol: EDI-EDIFACT** using the folder icon.
   c. Enable Set Source and Set Target for **Document Flow: UNB** (Figure 21-10 on page 739).

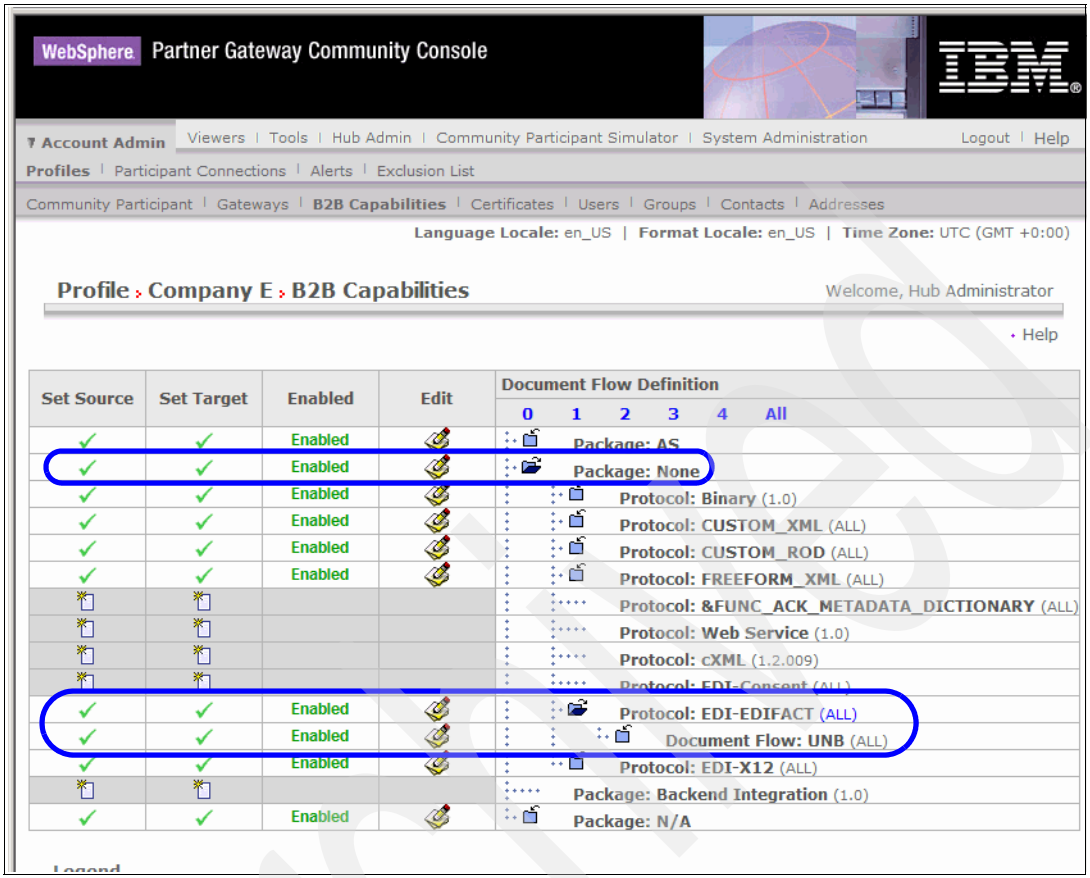


Figure 21-10 None/EDI-EDIFACT/UNB B2B capabilities activation

d. Enable the Set Source and Set Target for **Package: NA**, then expand **Package:NA** using the folder icon.

e. Enable Set Source and Set Target for **Protocol: EDI00B**, then expand **Protocol: EDI00B** using the folder icon.

f. Enable Set Source and Set Target for **Document Flow: ORDERS** (Figure 21-11 on page 740).

Archived

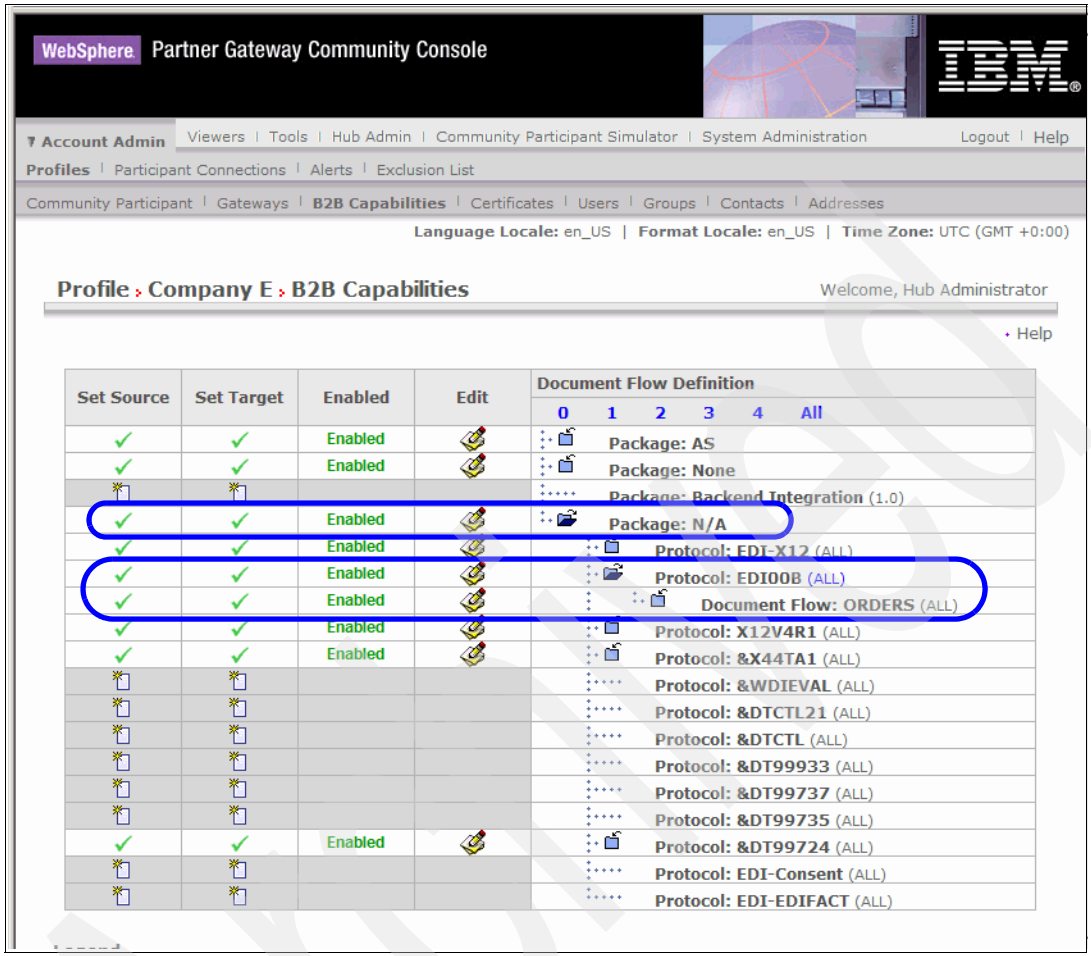


Figure 21-11 NA/EDI00B/ORDERS B2B capabilities activation

g. Under Package: NA, which has already been expanded, enable Set Source and Set Target for **Protocol: EDI-X12**.

h. Enable Set Source and Set Target for **Document Flow: ISA**.

The B2B capabilities screen of Company E profile should now look similar to Figure 21-12 on page 741. The highlighted sections are the B2B capabilities required for this example.

Profile › Company E › B2B Capabilities

Welcome, Hub Administrator

Help

Set Source	Set Target	Enabled	Edit	Document Flow Definition (0 1 2 3 4 All)
✓	✓	Enabled		Package: AS
✓	✓	Enabled		Package: None
✓	✓	Enabled		Protocol: Binary (1.0)
✓	✓	Enabled		Protocol: CUSTOM_XML (ALL)
✓	✓	Enabled		Protocol: CUSTOM_ROD (ALL)
✓	✓	Enabled		Protocol: FREEFORM_XML (ALL)
				Protocol: &FUNC_ACK_METADATA_DICTIONARY (ALL)
				Protocol: Web Service (1.0)
				Protocol: cXML (1.2.009)
				Protocol: EDI-Consent (ALL)
✓	✓	Enabled		Protocol: EDI-EDIFACT (ALL)
✓	✓	Enabled		Document Flow: UNB (ALL)
✓	✓	Enabled		Protocol: EDI-X12 (ALL)
				Package: Backend Integration (1.0)
✓	✓	Enabled		Package: N/A
✓	✓	Enabled		Protocol: EDI-X12 (ALL)
✓	✓	Enabled		Document Flow: ISA (ALL)
✓	✓	Enabled		Protocol: EDI00B (ALL)
✓	✓	Enabled		Document Flow: ORDERS (ALL)
✓	✓	Enabled		Protocol: X12V4R1 (ALL)
✓	✓	Enabled		Protocol: &X44TA1 (ALL)
				Protocol: &WDIEVAL (ALL)
				Protocol: &DTCTL21 (ALL)
				Protocol: &DTCTL (ALL)
				Protocol: &DT99933 (ALL)
				Protocol: &DT99737 (ALL)
				Protocol: &DT99735 (ALL)
✓	✓	Enabled		Protocol: &DT99724 (ALL)
				Protocol: EDI-Consent (ALL)
				Protocol: EDI-EDIFACT (ALL)

Figure 21-12 B2B capabilities enabled for the Company E profile

Updating the Company A profile B2B capabilities

In the EDIFACT to EDI-X12 outbound flow, Company E is the source and Company A is the target. To make the interactions available to the participant connections, the Company A profile (Community Participant) will also need some new B2B capabilities enabled. Company A profile must have the target side of these interactions enabled. The three interactions are:

- None/EDIFACT/UNB to **NA/EDIFACT/UNB**
- NA/97B/ORDERS to **NA/X12V4R1/850**
- NA/EDIX12/ISA to **AS/EDIX12/ISA**

The steps to update the Company A profile are as follows:

1. As the hub administrator (hubadmin) user on the Community Console of Company E, select **Account Admin** → **Profiles** → **Community Participant**.
2. Click **Search**.
3. From the list of configured participants, click the **magnifying glass** icon next to **Company A**.
4. Within the Company A profile, click **B2B capabilities**.
5. On this B2B capabilities screen, there are three package/protocol/document flow sets that need to be enabled to match the interactions we will create later. The steps to do this are:

   Note: As before, with the Company E profile, some of these B2B capabilities will already be enabled for both Set Source and Set Target. If they are already enabled, no additional action is needed.

   a. Enable Set Source and Set Target for **Package: N/A**, then expand **Package: NA** using the folder icon.
   b. Enable Set Source and Set Target for **Protocol: EDI-EDIFACT**, and then expand the **Protocol: EDI-EDIFACT** using the folder icon.
   c. Enable Set Source and Set Target for **Document Flow: UNB**
   d. Under Package:NA, enable Set Source and Set Target for **Protocol: X12V4R1**, then expand **Protocol: X12V4R1** using the folder icon.
   e. Enable Set Source and Set Target for **Document Flow: 850**.
   f. Enable Set Source and Set Target for **Package: AS**, then expand **Package: AS** using the folder icon.
   g. Enable Set Source and Set Target for **Protocol: EDI-X12** under Package: AS.
   h. Finally enable Set Source and Set Target for **Document Flow: ISA**.

   The B2B capabilities screen of Company A profile should now look similar to (Figure 21-13 on page 743). The highlighted sections are the B2B capabilities required for this example.

Archived

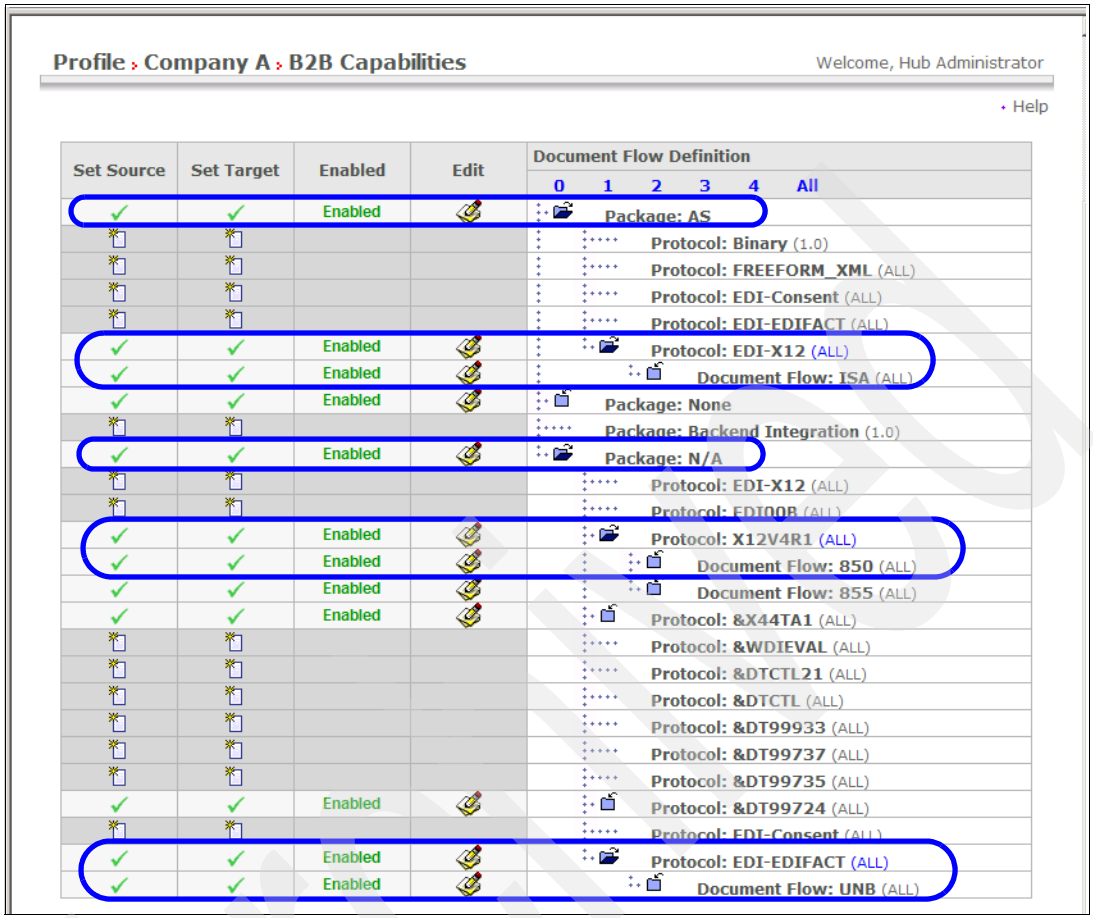


Figure 21-13 B2B capabilities enabled for the Company A profile

Reviewing the Company A profile gateway

A gateway is the exit point out of WebSphere Partner Gateway. Company E will need a gateway defined in the profile of Company A to which to deliver the transformed envelopes. In Chapter 7, “Creating a basic B2B exchange” on page 137, and again in Chapter 19, “Enveloping and de-enveloping” on page 621, we delivered outbound AS packaged EDI-X12 envelopes to **`HTTPGateway`**. This gateway already exists under the profile of Company A (Figure 21-14 on page 744) and we use it again in this example.

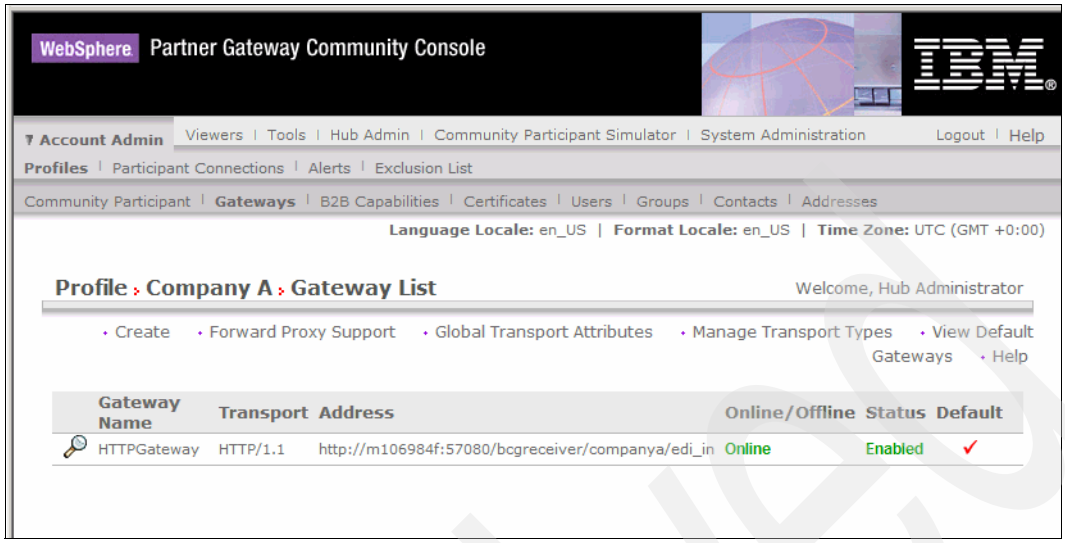


Figure 21-14 Company A profile defined gateway

Creating the new interactions

The next step for Company E hub administrator is to create the three interactions between Company E and Company A. You have already enabled the B2B capabilities of both participants to support their respective sides of these interactions, but the system needs the relationship between them defined. To do this, follow these steps:

1. As the hubadmin user on the Company E Community Console, select **Hub Admin** → **Hub Configuration** → **Document Flow Definition**
2. Click **Manage Interactions**.
3. As previously stated, there are three interactions needed. To create them, here are the steps
   a. For the None/EDI-EDIFACT/UNB to NA/EDI-EDIFACT/UNB interaction, follow these steps:
      i. Click **Create Interaction**
      ii. On the Source side, expand **Package:None**, **Protocol: EDI-EDIFACT**, and select **Document Flow: UNB**
      iii. On the Target side, expand, **Package:N/A**, **Protocol:EDI-EDIFACT**, and select **Document Flow:UNB**
      iv. For action, select **EDI De-envelope** (Figure 21-15 on page 745).
      v. Click **Save.**

Archived

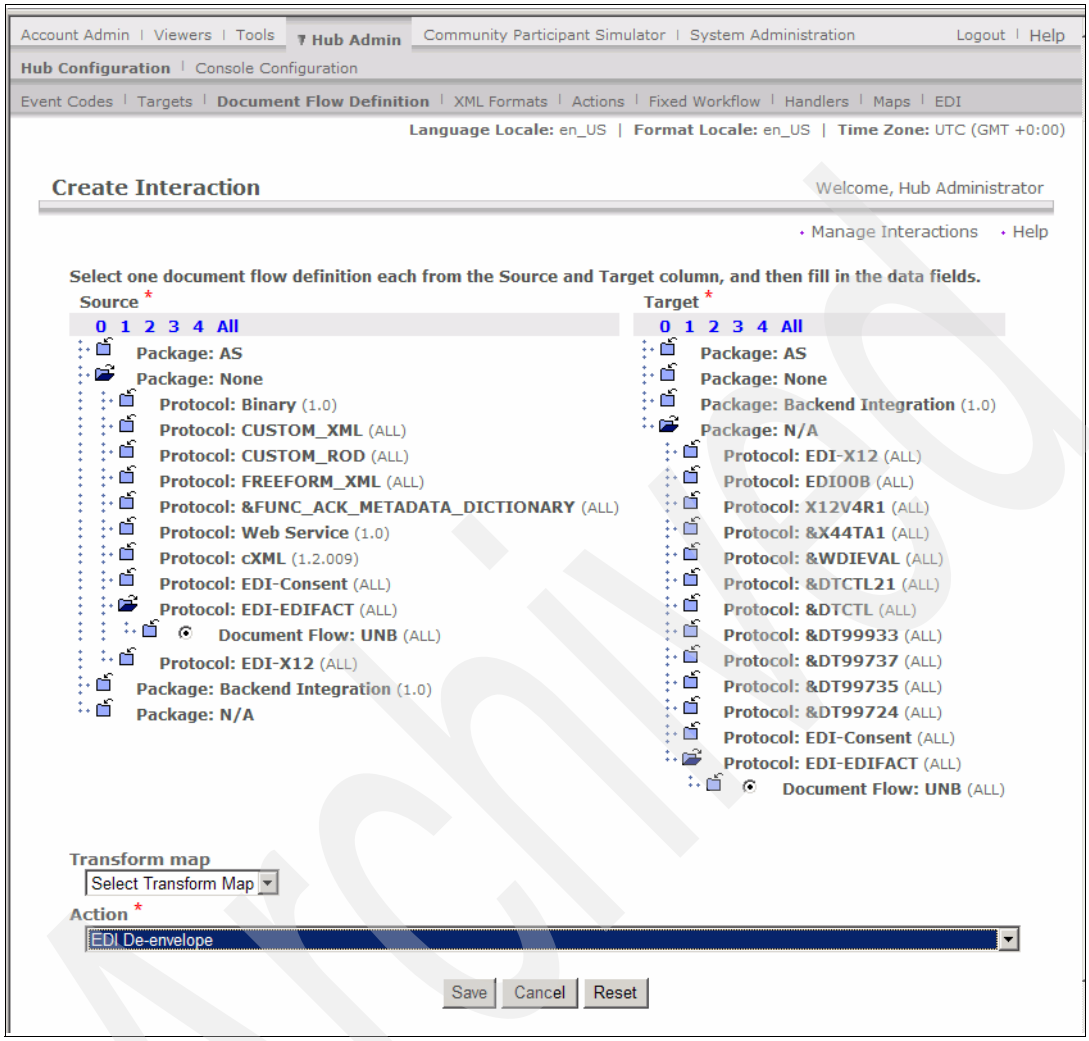


Figure 21-15 None/EDI-EDIFACT/UNB to NA/EDI-EDIFACT/UNB interaction

b. For the NA/EDI00B/ORDERS to NA/X12V4R1/850 interaction, follow these steps:

   i. Click **Create Interaction.**

   ii. On the Source side, expand **Package: N/A**, **Protocol: EDI00B**, and then select **Document Flow: ORDERS.**

   iii. On the Target side, expand **Package:N/A**, **Protocol: X12V4R1** and then **Document Flow: 850.**

Archived

iv. For Transformation map, select **EDIFACT_X12_PO** from the drop-down menu.

v. For Action, select **EDI Validate and EDI Translate** (Figure 21-16). The EDI validation that will be done here is the standard EDIFACT UNB validation.

vi. Click **Save**.

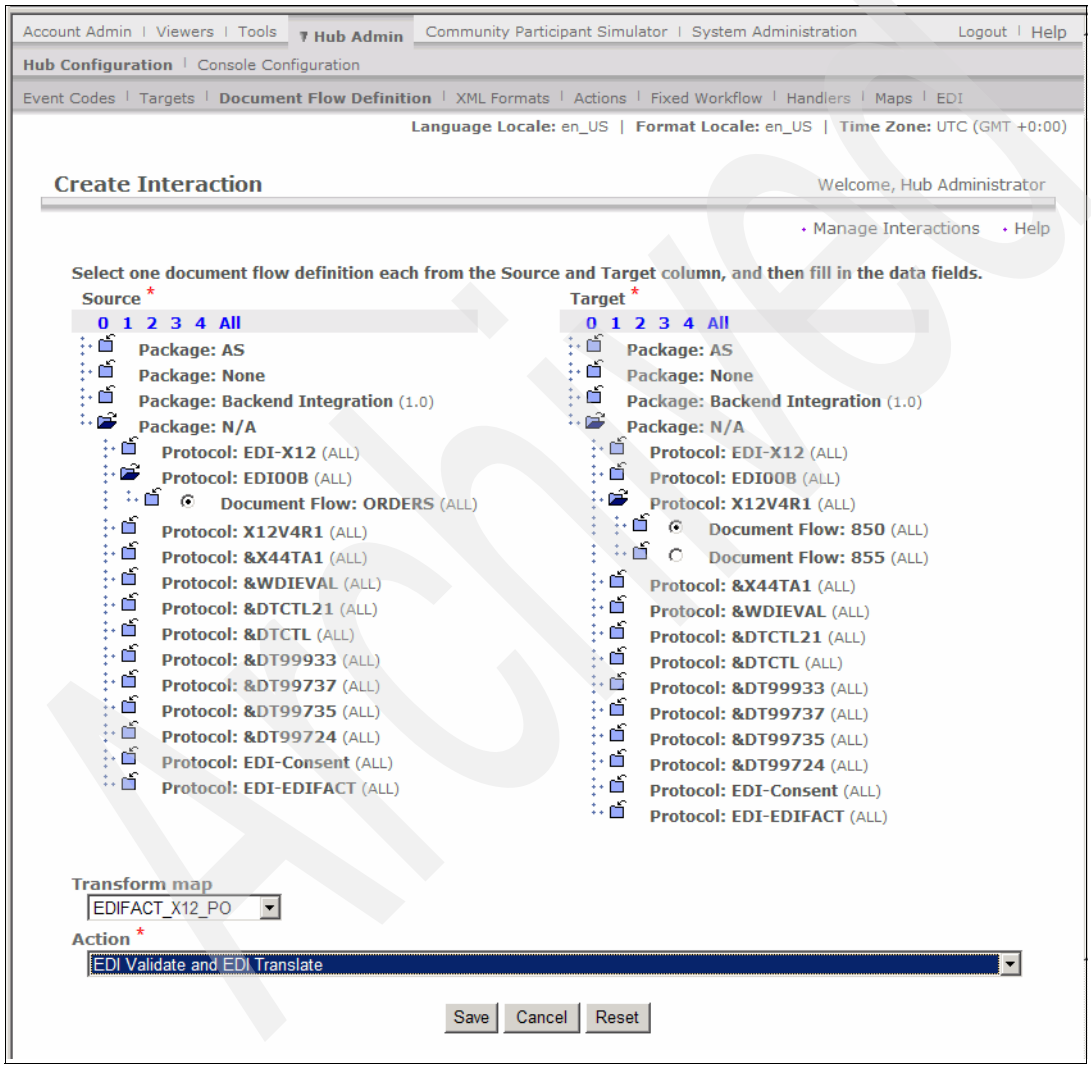


Figure 21-16 NA/EDI00B/ORDERS to NA/X12V4R1/850 interaction

c. For the third and final interaction, NA/EDI-X12/ISA to AS/EDI-X12/ISA, follow these steps:

   i. Click **Create Interaction**.

   ii. On the Source side, expand **Package: N/A**, **Protocol: EDI-X12**, and select **Document Flow: ISA**.

   iii. On the Target side, expand **Package: AS**, **Protocol: EDI-X12**, and select **Document Flow: ISA**.

   iv. For the Action, select **Pass-Through** (Figure 21-17 on page 748). The enveloping that occurs at this point happens automatically for EDI transactions coming out of WebSphere Partner Gateway. There is no EDI-envelope action to be set.

   v. Click **Save**.

Archived

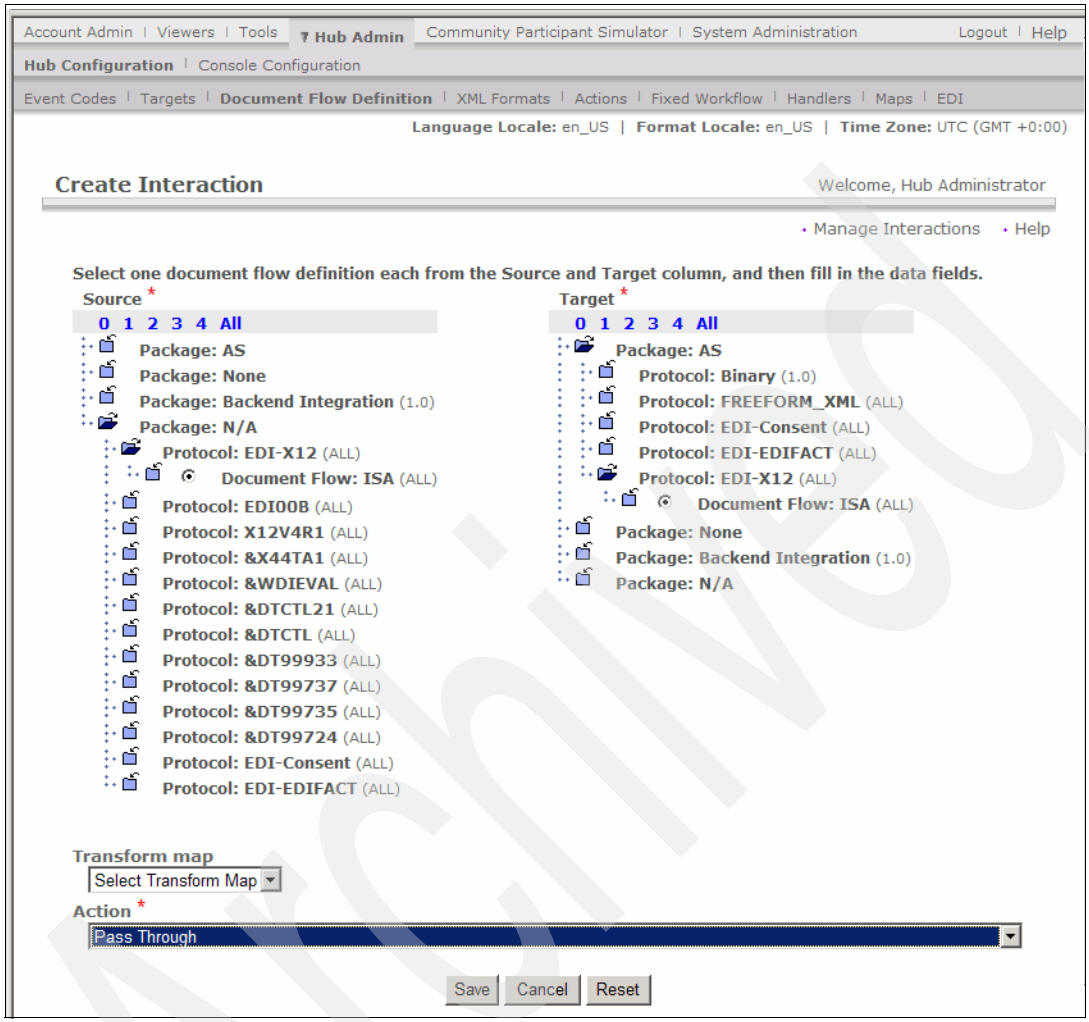


Figure 21-17 NA/EDI-X12/ISA to AS/EDI-X12/ISA interaction

> **Note:** After the **Save** action, you might receive an error (Figure 21-18 on page 749). This is because the same interaction was created for NA/EDI-X12/ISA to AS/EDI-X12/ISA in Chapter 19, “Enveloping and de-enveloping” on page 621 and is used again in Chapter 20, “Functional acknowledgements” on page 695. If you have not done these examples, no error is thrown.

Archived

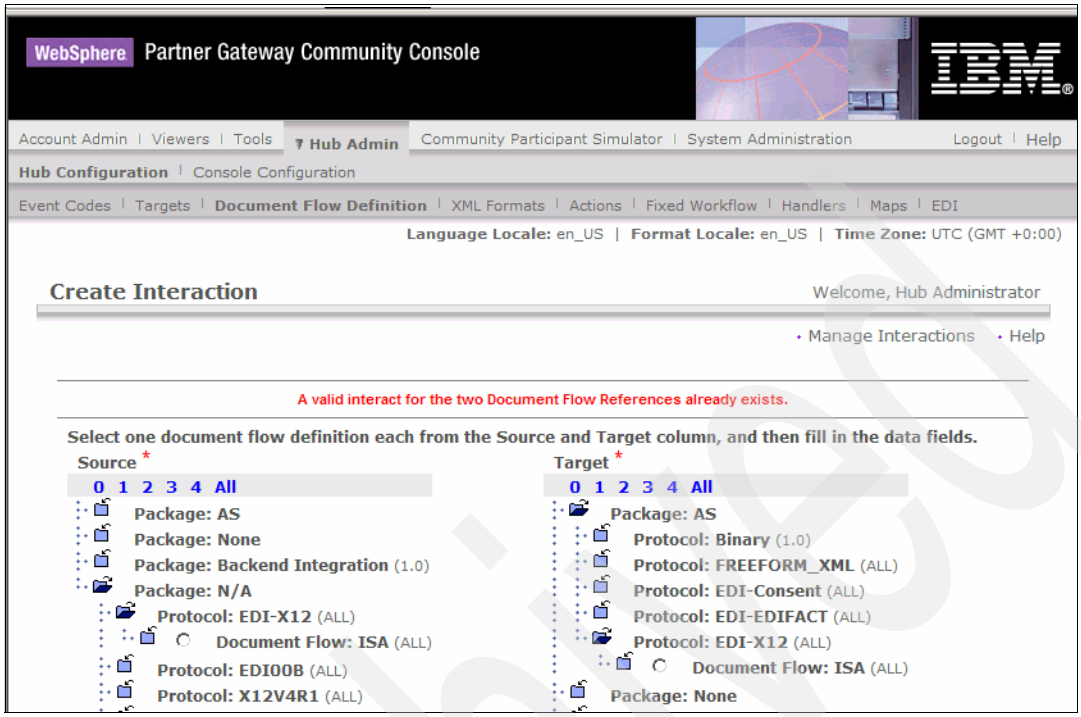


Figure 21-18 Possible interaction creating error

Activating the interactions

The interactions have now been defined on the WebSphere Partner Gateway system of Company E. However, they have not been activated. Follow these, steps:

1. As the hubadmin user on the Company E Community Console, click **Account Admin → Participant Connections.**
2. Select **Company E** as the Source and **Company A** as the Target, then click **Search**.
3. At this point in the redbook, there are several possible participant connections available. Right now, we are concerned with the three we just created.
   - None/EDI-EDIFACT/UNB to N/A/EDI-EDIFACT/UNB
   - N/A/EDI00B/ORDERS to N/A/X12V4R1/850
   - N/A/EDI-X12/ISA to AS/EDI-X12/ISA

   Click **Activate** for these three interactions.

Archived

Note: n most cases the N/A/EDI-X12/ISA to AS/EDI-X12/ISA will still be activated from previous examples in this redbook

The Activated interactions are highlighted in Figure 21-19.

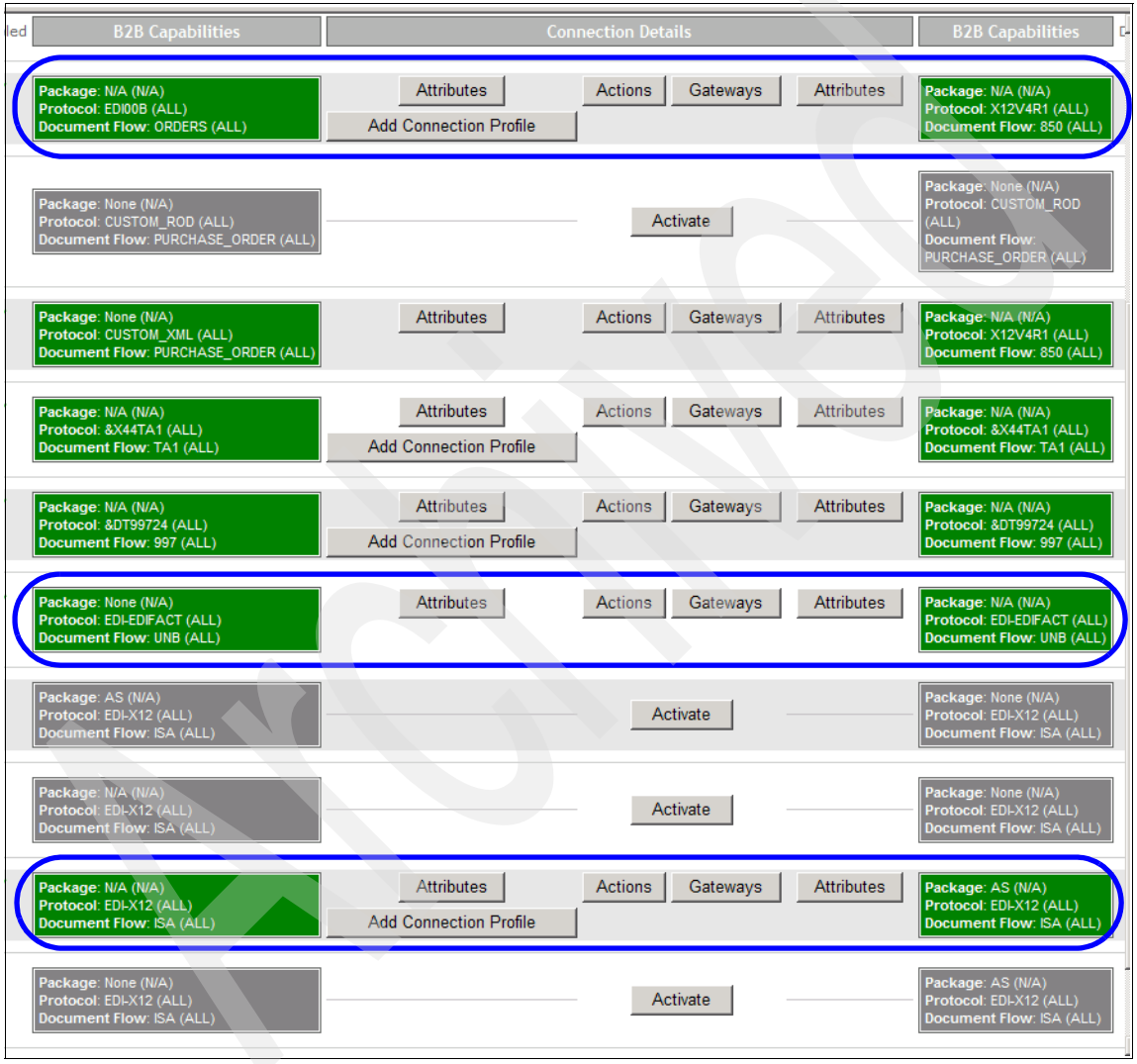


Figure 21-19 Activated participant connections

Configuring the activated participant connections

At this point, you have created all the pieces needed for the example outbound flow from Company E to Company A. However, it is a relatively complex configuration that should be tuned and reviewed to make sure the output is going to meet the expectations.

There a multitude of configurable parameters for EDI communication. Some of the parameters are mandatory, such as choosing an envelope profile, while others are optional. We should review the details of each participant connection, to confirm the current settings and add any that are needed:

1. For the None/EDI-EDIFACT/UNB to N/A/EDI-EDIFACT/UNB channel:
   a. Click **Attributes**, highlighted in Figure 21-20.

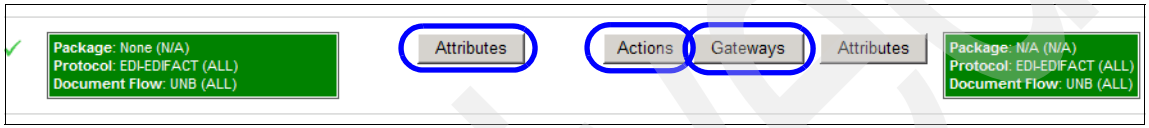


Figure 21-20 Configuring first interaction

   b. In the Connection Attributes window, we want to set Allow documents with duplicate document ids to **Yes**. This will make it possible to send the same sample document into the system without being rejected. To illustrate batching, we will send sample documents through several times to observe the output. You actually set the parameter by clicking the **Protocol: EDI-EDIFACT** folder and choosing **Yes** in the Update field (Figure 21-21). **Save** the changes.

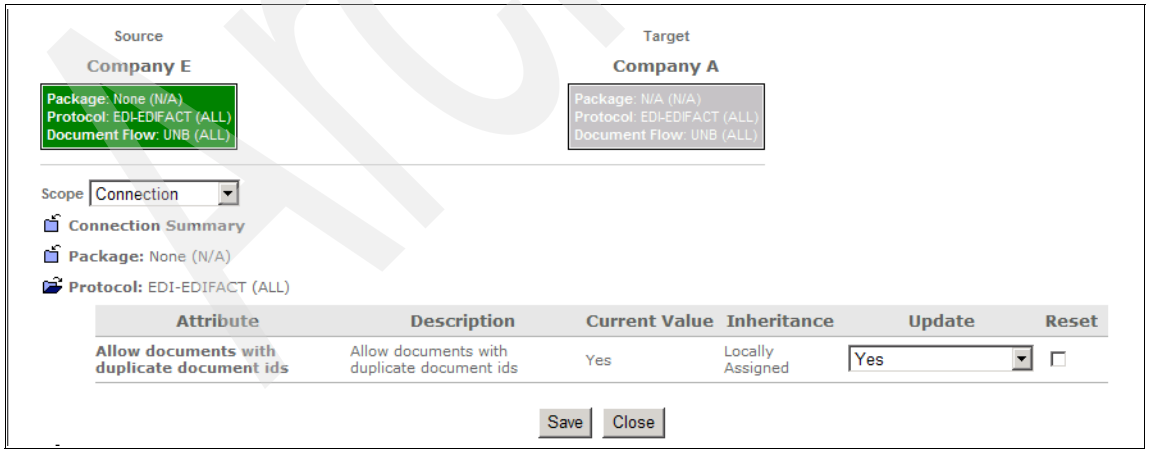


Figure 21-21 Updating connection attributes

Archived

c. Click **Return** to return back to the Manage Connections screen. Again, click the **Actions** button highlighted in Figure 21-20 on page 751. Here we just want to verify the Action is **EDI De-envelope**. Close the window.

d. Now click the **Gateways** button highlighted in Figure 21-20 on page 751. Again, we are just verifying the source and target gateways are correct. **FileSystemGateway** should be the source, and **HTTPGateway** should be the destination. These gateways should be the same for all three interactions. Even though the actual flow has some internal N/A package steps, this definition should always define the original source and final target.

2. The next channel is the N/A/EDI00B/ORDERS to N/A/X12V4R1/850 connection:

   a. First click the Source side **Attributes** button highlighted in Figure 21-22.

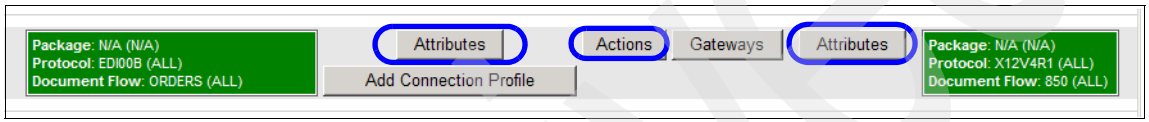


Figure 21-22 Configuring the second interaction

   b. Here you need to define the Interchange Identifier as `companye`. You can do this by clicking the folder icon next to **Protocol: EDI00B**, then adding `companye` into the field shown in Figure 21-23. This parameter allows the enveloper to determine the source business identifier for use in creating the envelope headers. **Save** the changes.

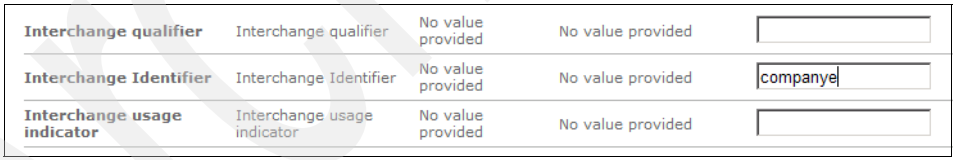


Figure 21-23 Updating the source side connection attributes

   c. Click the **Actions** button highlighted in Figure 21-22. Verify that the Transformation Map listed is **EDIFACT_X12_PO**. and that the Action is **EDI Validate and EDI Translate**. This connection defines translating the EDIFACT messages into EDIX12 transactions.

   d. Click the target side **Attributes** button highlighted in Figure 21-22. This time we are configuring two parameters. Click the folder icon next to **Protocol: X12V4R1** to update them.

      i. The first is the target side **I**nterchange Identifier and setting it to **companya**.

ii. The second is the **Envelope Profile**, which can be set to **WPG-Defined-X12**. This is the default X12 profile defined by WebSphere Partner Gateway. The EDIFACT messages are transformed into X12 transactions, so we want the enveloper to send them out in an X12 envelope.

Figure 21-24 shows the parameters with the correct information.

Envelope Profile	Envelope Profile	No value provided	No value provided	WPG-Defined-X12
Max validation error level	Max validation error level	0	Inherited from: **Scope:** Global **Type:** Max validation error level	Select one to update
Validation level	Validation level	0	Inherited from: **Scope:** Global **Type:** Validation level	Select one to update
Char set validation table	Char set validation table	CHARSET	Inherited from: **Scope:** Global **Type:** Char set validation table	
Alphanumeric validation table	Alphanumeric validation table	ALPHANUM	Inherited from: **Scope:** Global **Type:** Alphanumeric validation table	
Generate group level info only in functional Ack	Generate group level info only in functional Ack	No	Inherited from: **Scope:** Global **Type:** Generate group level info only in functional Ack	Select one to update
Century control year	Century control year	10	Inherited from: **Scope:** Global **Type:** Century control year	
Detailed validation of segment	Detailed validation of segment	No	Inherited from: **Scope:** Global **Type:** Detailed validation of segment	Select one to update
Connection Profile Qualifier1	Connection Profile Qualifier1	No value provided	No value provided	
Interchange qualifier	Interchange qualifier	No value provided	No value provided	
Interchange Identifier	Interchange Identifier	No value provided	No value provided	companya

Figure 21-24 Updating the target side connections attributes.

iii. Click **Save**.

3. The third interaction is the N/A/EDI-X12/ISA to AS/EDI-X12/ISA connection (Figure 21-25). There is no additional configuration needed here. However, you might want to verify that the action is **Pass-Through**, and that the AS packaging attribute has **AS MDN requested** set to No. These should be set correctly from earlier examples in this redbook, but you can use the highlighted buttons in Figure 21-25 to confirm it.

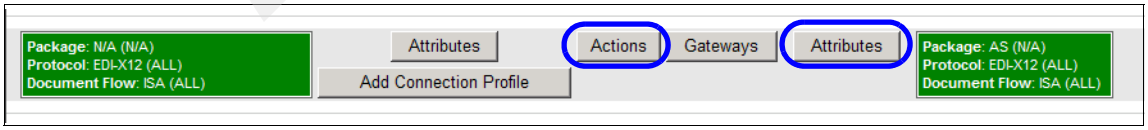


Figure 21-25 Configuring the third interaction

This completes the activation and configuration of the three interactions that define our EDIFACT to EDIX12 example from the Company E perspective.

21.2.3 Configuration for Company A

In the examples in Chapter 7, “Creating a basic B2B exchange” on page 137 and Chapter 19, “Enveloping and de-enveloping” on page 621, Company A received AS packaged EDI-X12 documents. Company A received the documents on their **HTTPGateway**, which then passes to them to the Document Manager component of Company A’s WebSphere Partner Gateway implementation. The action defined for the interaction is Pass-through. The Document Manager then delivers the files to the local **FileSystemGateway**.

Remember that Company E is the Participant that is adding a new back-end application that uses EDIFACT documents. By using EDIFACT to EDIX12 transformation, Company E handles the additional burden, and Company A can still receive the documents exactly as before. All configuration for Company A was described in Chapter 7, “Creating a basic B2B exchange” on page 137. No additional configuration is needed.

21.2.4 Validating the flow

Before we discuss the batching and splitting features, validate that the flow configured so far in this chapter is working successfully. To do that, you will need a new EDIFACT document. Figure 21-1 is an example EDIFACT document, called `edifact.inp` created for use in this scenario. You can see the business identifiers are set for Company E (companye) as the source and Company A (companya) as the target.

Example 21-1 edifact.inp (single message in envelope)

```
UNB+0001:+companye::+companya::+050802:1053+00001++++++++++++'
UNH+000001+ORDERS:EDI:00B:EN:+++++++'
BGM+105:::+WIDGETORDER:01++US'
DTM+137:050802:101'
NAD+COD+STANZA::+STANZA CO:RALEIGH:::+++++27709+NC'
LIN+WIDGET+++++'
PIA+5+BLUE WIDGET:PO::+RED WIDGET:PO::+YELLOW WIDGET:PO::++'
UNS+S'
MOA+146:2456.75::9:'
UNT+9+000001'
UNZ+1+00001'
```

To validate our transformation is working, put the sample EDIFACT document into the Company E local directory,

`/wpgv6/data/companye/edi_out/Documents/Production`. This will begin the outbound flow.

Use the Document Viewer to view the results. The output is displayed in Figure 21-26. The middle document is the De-enveloping of the EDIFACT envelope. The bottom document is the transformation of the single EDIFACT message into a single EDI-X12 850 transaction, and finally the top document is the packaging of the EDIX12 transaction into an ISA envelope with AS headers. The top document is the EDI-X12 envelope that gets delivered to the HTTPGateway of Company A.

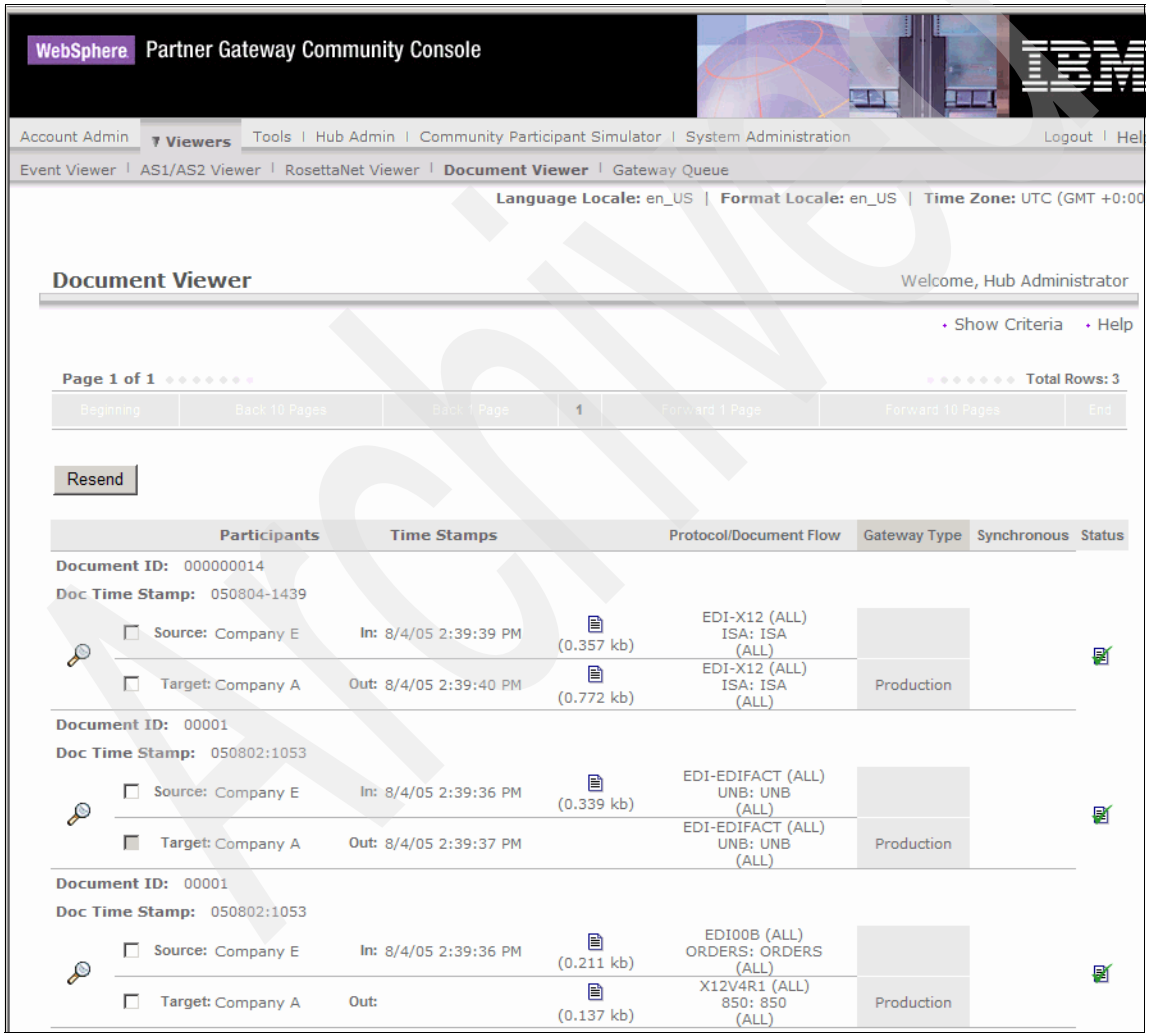


Figure 21-26 Document Viewer of Sample EDIFACT document traversing defined flow

21.2.5 Using batching to alter output

Now that you have configured and validated a successful EDIFACT to EDIX12 transformation flow, we can discuss the impact of batching on this scenario. You can determine the current batching setting by logging on as the hubadmin user on the Company E WebSphere Partner Gateway implementation and selecting **Hub Admin** → **Hub Configuration** → **EDI** → **Enveloper** (Figure 21-27).

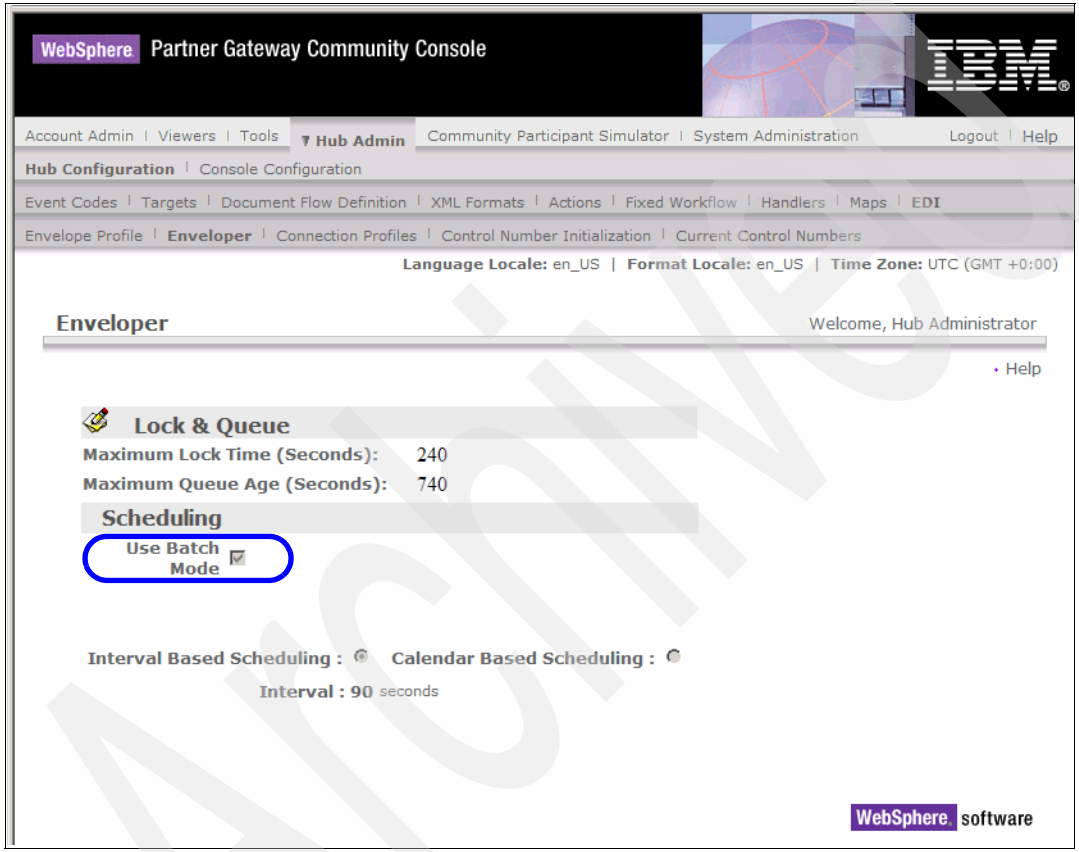


Figure 21-27 Batch setting on the EDI enveloper screen

Enveloping with Batch mode on

As you can see in Figure 21-27, the batch setting is on, and the Scheduling is set to Interval Based Scheduling on 90 second intervals. The interval scheduling means that every 90 seconds WebSphere Partner Gateway will take all the interactions currently awaiting output transport and group them in EDI envelopes before sending. The *Use Batch Mode* setting means that if the documents originally came into the system together in a single batch, it will attempt to keep those particular documents together, regardless of how long they take to

process, even if this means exceeding the set interval. Remember that WebSphere Partner Gateway considers an EDI envelope as a single batch.

To demonstrate this, we need a sample EDIFACT document that contains more than a single message. Figure 21-2 shows a sample EDIFACT document that contains three messages within the envelope named `edifact3.inp`.

Example 21-2 edifact3.inp (three messages, single envelope)

```
UNB+0001:+companye::+companya::+050802:1053+00001++++++++++++'
UNH+000001+ORDERS:EDI:00B:EN:+++++++'
BGM+105:::+WIDGETORDER:01++US'
DTM+137:050802:101'
NAD+COD+STANZA::+STANZA CO:RALEIGH:::+++++27709+NC'
LIN+WIDGET+++++'
PIA+5+BLUE WIDGET:PO::+RED WIDGET:PO::+YELLOW WIDGET:PO::++'
UNS+S'
MOA+146:2456.75::9:'
UNT+9+000001'
UNH+000002+ORDERS:EDI:00B:EN:+++++++'
BGM+105:::+WIDGETORDER:01++US'
DTM+137:050802:101'
NAD+COD+STELLA::+STELLA CO OP:DURHAM:::+++++27719+NC'
LIN+WIDGET+++++'
PIA+5+PINK WIDGET:PO::+WHITE WIDGET:PO::+BLACK WIDGET:PO::++'
UNS+S'
MOA+146:3456.25::9:'
UNT+9+000002'
UNH+000003+ORDERS:EDI:00B:EN:+++++++'
BGM+105:::+WIDGETORDER:01++US'
DTM+137:050802:101'
NAD+COD+WESTSIDE::+WESTSIDE SOCIETY:DURHAM:::+++++27719+NC'
LIN+WIDGET+++++'
PIA+5+TWISTED WIDGET:PO::+CURLED WIDGET:PO::+STRAIGHT WIDGET:PO::++'
UNS+S'
MOA+146:5000.00::9:'
UNT+9+000003'
```

If you take this sample document and place it in the FileTarget of Company E for processing, it should de-envelope into three messages instead of one.

To illustrate batching, try this:

1. Place the `edifact3.inp` into the .../edi_out/Documents/Production directory of Company E.
2. Refresh the directory. After the file has been picked up, copy the sample file there again.

3. Again refresh the directory. After the file has been picked up, place the file one more time, for a total of three times.

   The flow with batching enabled is represented by Figure 21-28.

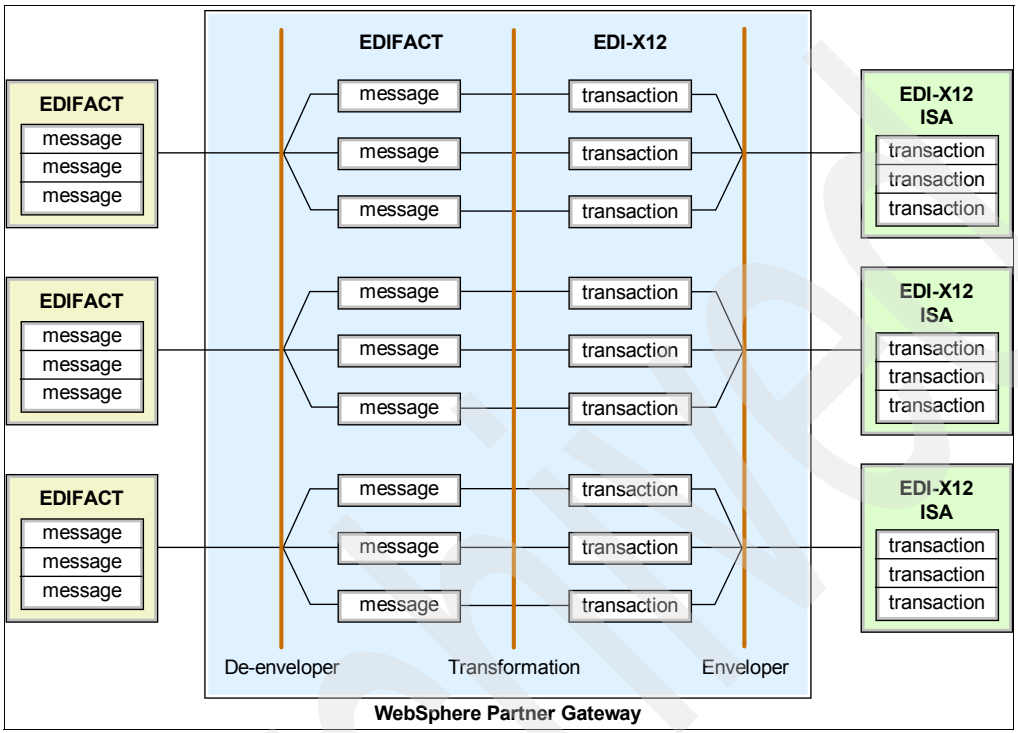


Figure 21-28 High -level view of Batching ON

> **Note:** EDI to EDI transformations are a special case within WebSphere Partner Gateway. Not only will the system try to preserve the enveloping structure of an inbound batch, the system will also attempt to preserve the order of the transactions in the received batch.

To verify that WebSphere Partner Gateway correctly batched the documents, you can use the Document Viewer. However, you might need to tune the search parameters because the document viewer displays all the data within an hour block of time. The best way to accomplish this, would be to set the Start Time to a time after the sample `edifact.inp` was sent, but before the three `edifact3.inp` documents were sent. This is shown in Figure 21-29 on page 759.

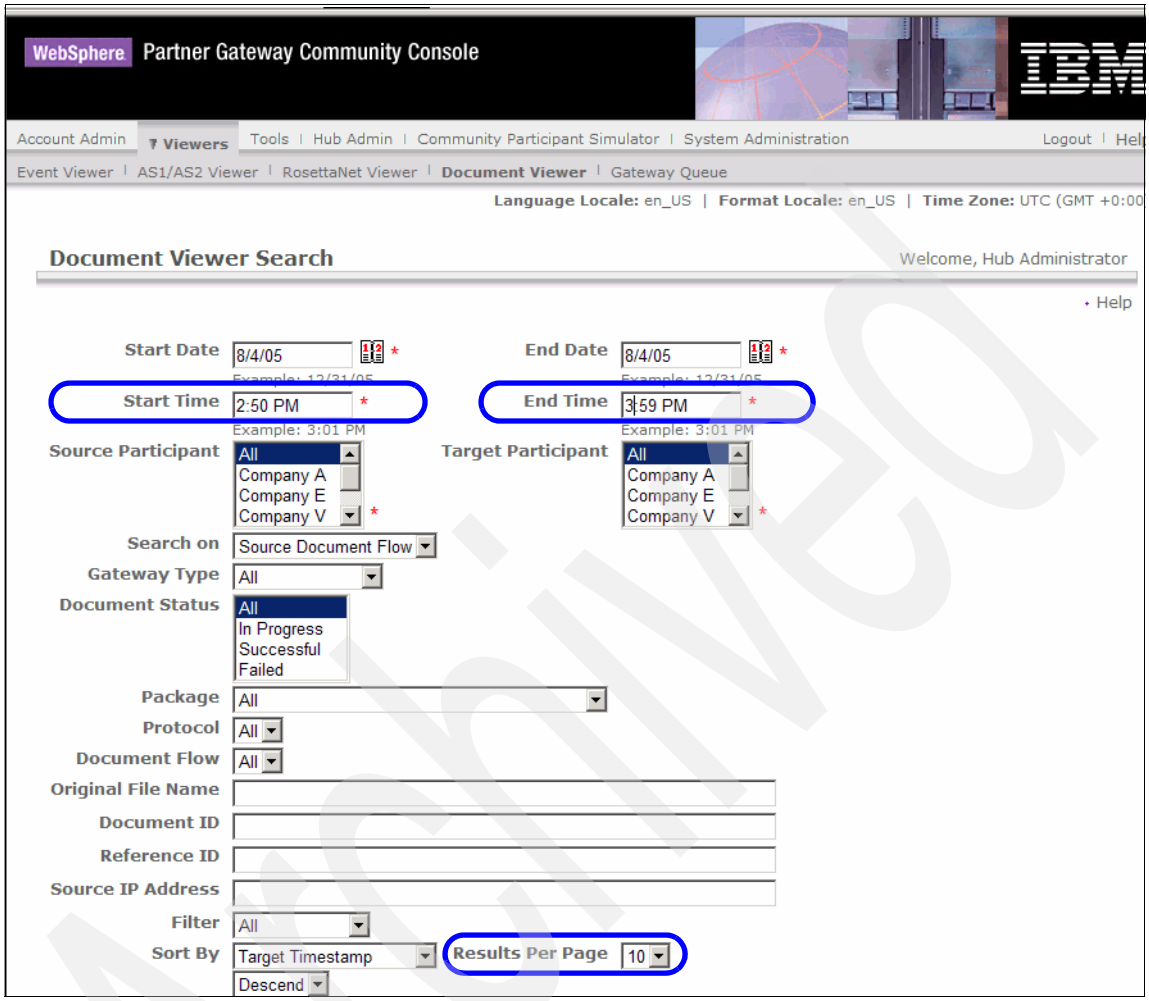


Figure 21-29 Document Viewer Search screen

To demonstrate batching, you sent three files, each with three transactions. The expected view based on the high-level picture in Figure 21-28 on page 758 should have 15 items in the Document Viewer. Three inbound EDIFACT envelopes, nine message to interaction transformations, and three outbound EDI-X12 envelopes to Company A. There are three outbound envelopes because batching attempts to preserve the batch structure of the inbound documents. Figure 21-30 on page 760 shows the results. The Total Rows of 15 is highlighted, as are the three outbound envelopes.

Archived

Tip: On the Document Viewer Search page, you can update the Results Per Page setting to **20,30** or even **50** documents so that more documents are visible on the Document Viewer screen at one time. This parameter is highlighted in Figure 21-29 on page 759.

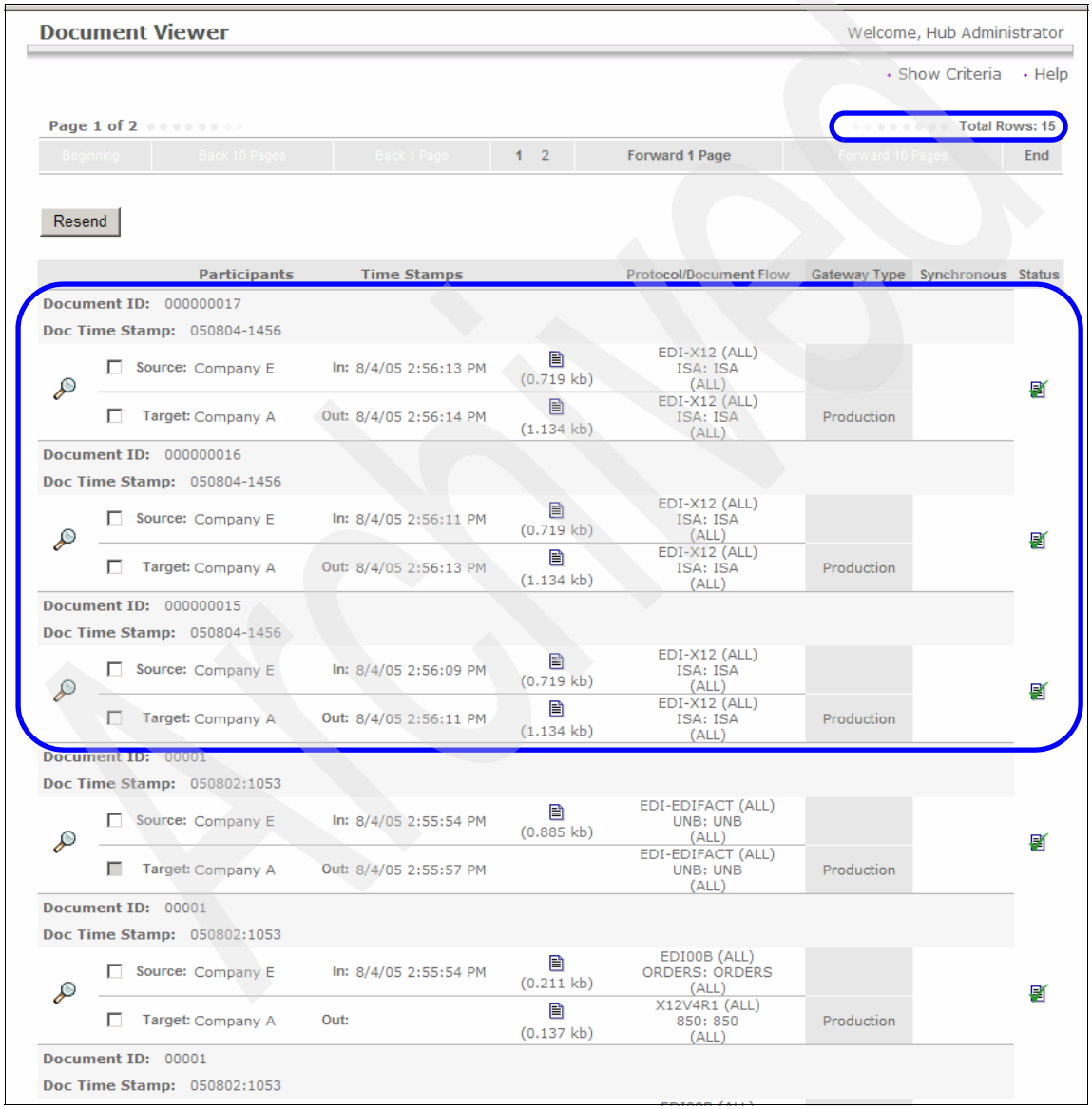


Figure 21-30 Document Viewer after edifact3.inp placed three times

If you view the details of the top envelope (click the **magnifying glass**), and use the raw document viewer to view the final document that was sent to Company A (the document icon next to **Company A**). Highlighted in Figure 21-31 are the three transactions in one of the outbound envelopes. This means the incoming EDIFACT envelope was preserved as a batch.

Figure 21-31 Raw document view of one of the batched envelopes

Enveloping with batch mode off

At this point, you have seen how WebSphere Partner Gateway packages the EDI envelopes when batch mode is enabled. When it is disabled, you are telling the

system to ignore the how the inbound transactions were received and rely solely on the scheduling of the enveloper.

First, turn off the Use Batch Mode check box. To do this, follow these steps:

1. As hubadmin user, go to **Hub Admin** → **Hub Configuration** → **EDI** → **Enveloper**.
2. Click **Edit** (pencil and paper icon) for the screen.
3. Uncheck **Use Batch Mode**.
4. Click **Save**, to save the changes.

To demonstrate the flow with batching now off, follow the same steps as before, with these differences:

1. Place the new three-message EDIFACT document into the `.../edi_out/Documents/Production` directory of Company E.
2. Refresh the directory. After the file has been picked up, copy the sample file there again.
3. Again refresh the directory. After it has been picked up, place the file one more time, for a total of three times.

A high-level representation of the result is shown in Figure 21-32 on page 763.

Archived

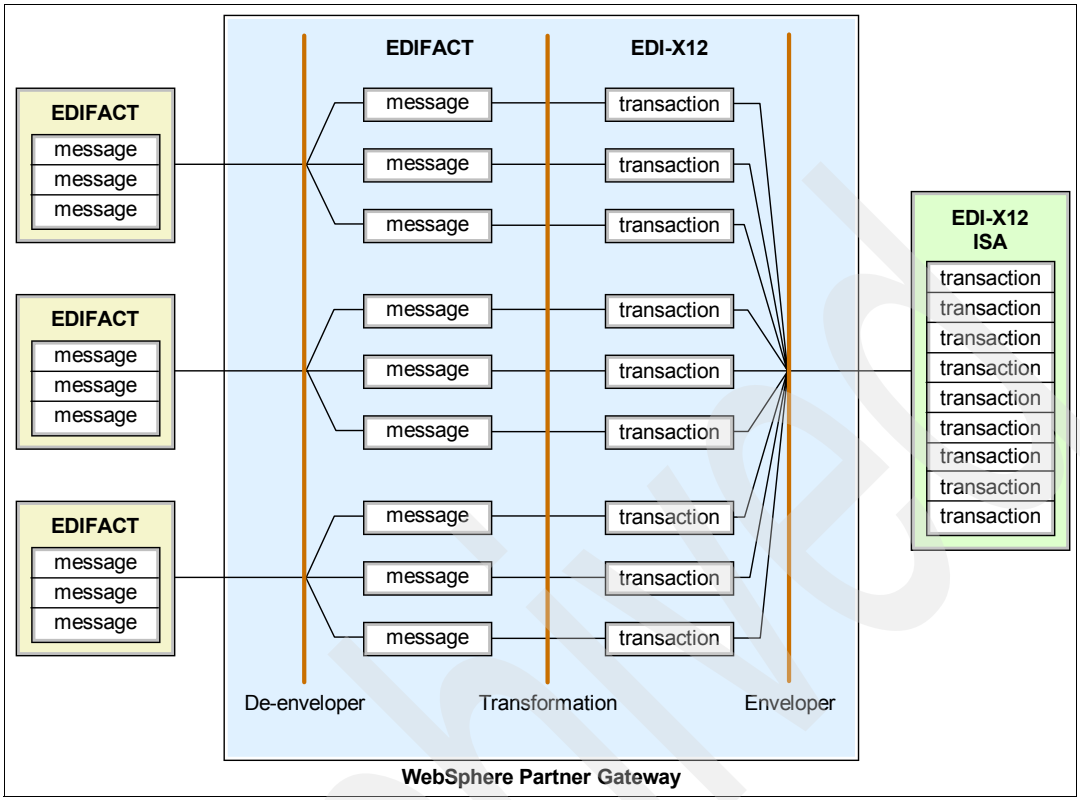


Figure 21-32 High-level view of Batching OFF

To verify the transaction, we will again use the Document Viewer. Remember to use the Start and End Time settings to separate the view of only this latest three test documents.

This time, we expect to see 13 items in the view. Three EDIFACT envelopes in, nine EDIFACT to EDIX12 transformations, but only *one* outbound envelope to Company A. With batching turned off, WebSphere Partner Gateway relies on the Scheduling which is currently set at 90 seconds. So as long as all the documents process within 90 seconds, the nine transactions waiting for transport to Company A will get enveloped into the same envelope this time. You could conceivably see two outbound envelopes if it takes longer than 90 seconds to process all nine interactions. However, the sample document is simple enough that it should not.

The results of the test, with batching turned off are shown in Figure 21-33 on page 764. One outbound envelope, much larger than previously is highlighted, along with the 13 total rows that were expected.

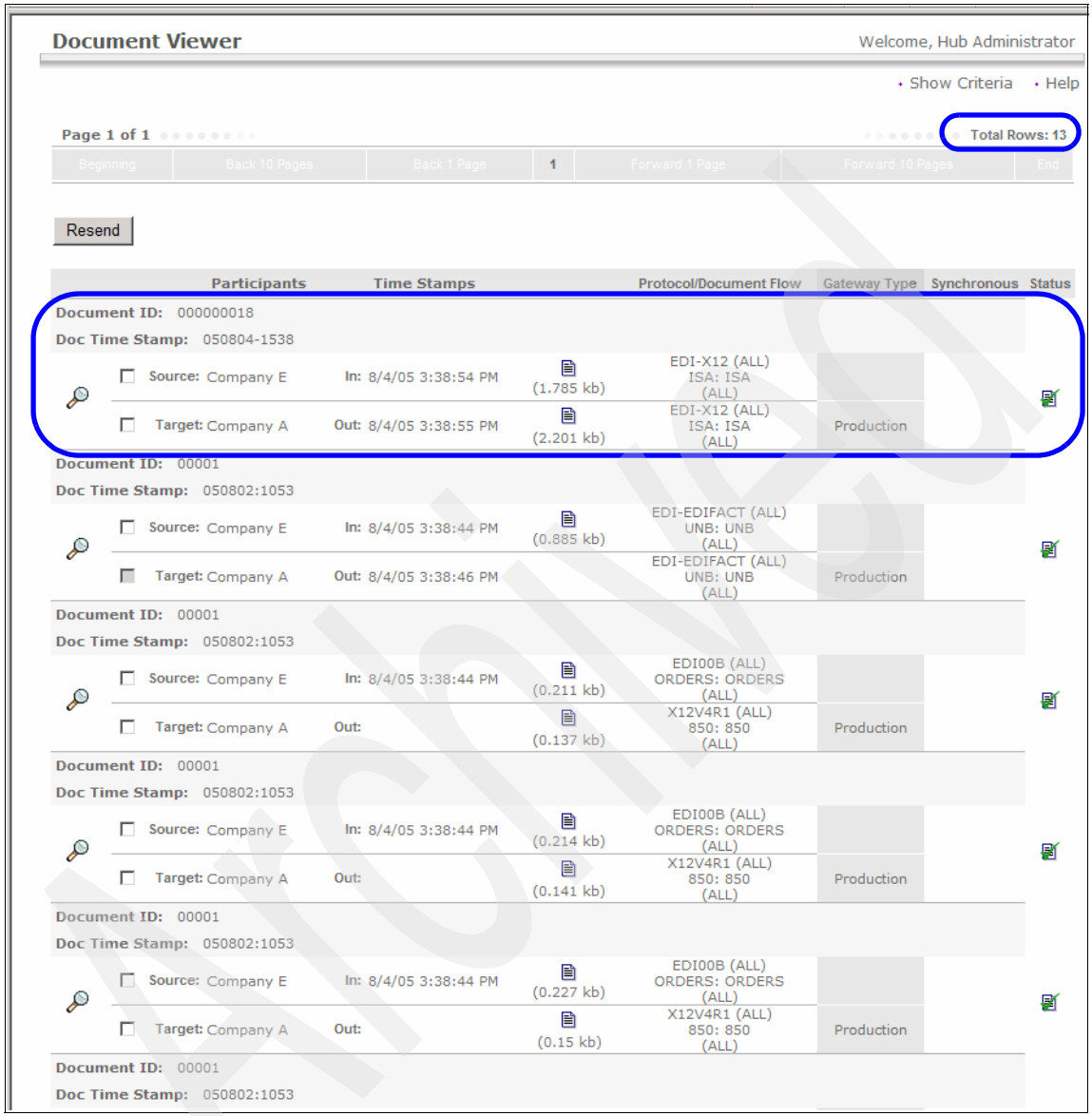


Figure 21-33 Batching off, document viewer results

Figure 21-34 on page 765 shows the raw document view of the envelope that was sent to Company A. Nine transactions are highlighted.

Raw Document Viewer - Microsoft Internet Explorer

```
AS2-From: companye
AS2-To: companya
AS2-Version: 1.1
Message-ID: <1123169935328000C29CF729E0015080000000000001553@WPGV6HUB>
Mime-Version: 1.0
Recipient-Address: http://m106984f:57080/bcgreceiver/companya/edi_in
Subject: companya;companye
Content-Length: 1828

ISA+ + + + + +companye + +companya +050804+1538+ + +000000018+ + + '
GS+PO+ + +companye+companya+20050804+1538+000000017+ +004010'
ST+850+000000026'
BEG+105+++01+050802++US'
N1+++COD+STANZA'
N3+STANZA CO'
PO1+WIDGET+++++PO+BLUE WIDGET+PO+RED WIDGET+PO+YELLOW WIDGET'
AMT+146+2456.75'
SE+7+000000026'
ST+850+000000027'
BEG+105+++01+050802++US'
N1+++COD+STELLA'
N3+STELLA CO OP'
PO1+WIDGET+++++PO+PINK WIDGET+PO+WHITE WIDGET+PO+BLACK WIDGET'
AMT+146+3456.25'
SE+7+000000027'
ST+850+000000028'
BEG+105+++01+050802++US'
N1+++COD+WESTSIDE'
N3+WESTSIDE SOCIETY'
PO1+WIDGET+++++PO+TWISTED WIDGET+PO+CURLED WIDGET+PO+STRAIGHT WIDGET'
AMT+146+5000'
SE+7+000000028'
ST+850+000000029'
BEG+105+++01+050802++US'
N1+++COD+STANZA'
N3+STANZA CO'
PO1+WIDGET+++++PO+BLUE WIDGET+PO+RED WIDGET+PO+YELLOW WIDGET'
AMT+146+2456.75'
SE+7+000000029'
ST+850+000000030'
BEG+105+++01+050802++US'
N1+++COD+STELLA'
N3+STELLA CO OP'
PO1+WIDGET+++++PO+PINK WIDGET+PO+WHITE WIDGET+PO+BLACK WIDGET'
AMT+146+3456.25'
SE+7+000000030'
ST+850+000000031'
BEG+105+++01+050802++US'
N1+++COD+WESTSIDE'
N3+WESTSIDE SOCIETY'
PO1+WIDGET+++++PO+TWISTED WIDGET+PO+CURLED WIDGET+PO+STRAIGHT WIDGET'
AMT+146+5000'
SE+7+000000031'
ST+850+000000032'
BEG+105+++01+050802++US'
N1+++COD+STANZA'
N3+STANZA CO'
PO1+WIDGET+++++PO+BLUE WIDGET+PO+RED WIDGET+PO+YELLOW WIDGET'
AMT+146+2456.75'
SE+7+000000032'
ST+850+000000033'
BEG+105+++01+050802++US'
N1+++COD+STELLA'
N3+STELLA CO OP'
PO1+WIDGET+++++PO+PINK WIDGET+PO+WHITE WIDGET+PO+BLACK WIDGET'
AMT+146+3456.25'
SE+7+000000033'
ST+850+000000034'
BEG+105+++01+050802++US'
N1+++COD+WESTSIDE'
N3+WESTSIDE SOCIETY'
PO1+WIDGET+++++PO+TWISTED WIDGET+PO+CURLED WIDGET+PO+STRAIGHT WIDGET'
AMT+146+5000'
SE+7+000000034'
```

Figure 21-34 Raw document view of nine transactions in the single X12 envelope

Archived

21.2.6 Batching review

You have now seen the WebSphere Partner Gateway use batching to modify the way it groups EDI interactions or messages into envelopes. This feature essentially allows you to tailor EDI output into groups based on your preference. With batching turned on, you can have WebSphere Partner Gateway attempt to preserve the envelope structure in which it was received. This is useful when the tracking of individual batches from certain partners is required. With batching turned off, the documents are enveloped within a more rigid time-based schedule, which can be useful when you want to minimize the use of the outbound gateway. For example, you want to wait one hour to gather many transactions for a trading partner, and then send a single envelope.

It is important to note that WebSphere Partner Gateway *attempts* to batch these interactions together when enveloping. It is not guaranteed to do so. The batching function uses some breakpoint rules to implement this batching feature.

For instance, a primary breakpoint would be a batch that has interactions destined for multiple partners. The system obviously cannot send a single envelope outbound in this case, it would need at least one envelope for each configured Participant depending on the number and structure of transactions. However in this case, it will continue to try to preserve the structure as much as.

An example of a secondary breakpoint would be an envelope profile. If the interactions between the participants have differing envelope profiles defined for certain type of transactions, then a inbound batch that has transactions destined for multiple profiles, might force the Document Manager to envelope those transactions separately.

21.3 Batching with other transformations

The example we discussed in this chapter illustrated batching for an EDI to EDI transformation flow. We used this example because it highlighted some of the batching rules unique to EDI batches, specifically the preservation of the original EDI envelopes. However, there are other transformations such as XML to EDI and ROD to EDI that also involve batching, because they need to envelope the outbound transactions when the result of the transformation is EDI.

XML and ROD documents do not have an envelope structure to preserve. So batching is a little more straightforward in their case. For example, a single ROD document coming into the WebSphere Partner Gateway and being transformed into EDI gets enveloped based on the enveloper schedule. However, the ROD Splitter has the ability to split several ROD documents that have been concatenated together. If a file consisting of several ROD documents together is

sent into the WebSphere Partner Gateway, it is considered a batch by the server. After transformation into EDI, the enveloper will attempt to keep these ROD documents to a single envelope if possible.

XML to EDI would function exactly as a ROD to EDI function. The XMLSplitterHandler would have to be configured as the preprocess Handler on the target that accepted the inbound XML documents, but it essentially functions the same as the ROD splitter. The XMLSplitterHandler gives that target the ability to split concatenated XML files into individual XML files, which would then be processed by the WebSphere Partner Gateway server as a batch. Configuration of a target preprocess handler on a target is covered in 21.4, “EDI Splitter” on page 767.

21.4 EDI Splitter

A *splitter* is a preprocess handler configured on the target that receives inbound documents. A ROD Splitter is mandatory, because WebSphere Partner Gateway determines the document flow from the ROD Splitter’s definition, (explained in Chapter 15, “Mapping” on page 473). XML and EDI Splitters are optional. Their function permits WebSphere Partner Gateway to split files that have several XML documents or EDI envelopes concatenated together into individual objects for processing by the server.

Figure 21-35 on page 768 displays the function of an EDI Splitter.

Archived

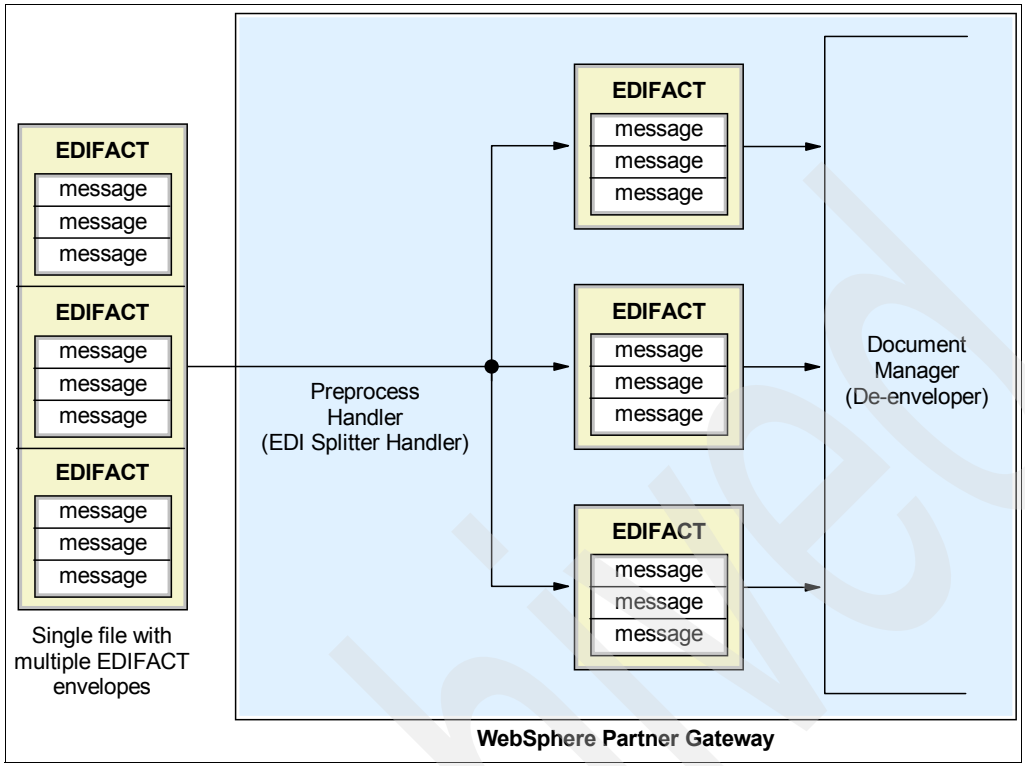


Figure 21-35 Function of an EDI Splitter

You can extend the configuration of the example in this chapter to accept these multiple EDI envelope files on the FileTarget of Company E by following these steps:

1. As the hubadmin user, select **Hub Admin** → **Hub Configuration** → **Targets**.
2. Click the **magnifying glass** next to **FileTarget**.
3. On the Target Details screen click **edit**.
4. In the Handlers section, for Configuration Point Handlers, use the drop down menu to select **preprocess**. Some additional boxes should appear (Figure 21-36 on page 769).

Archived

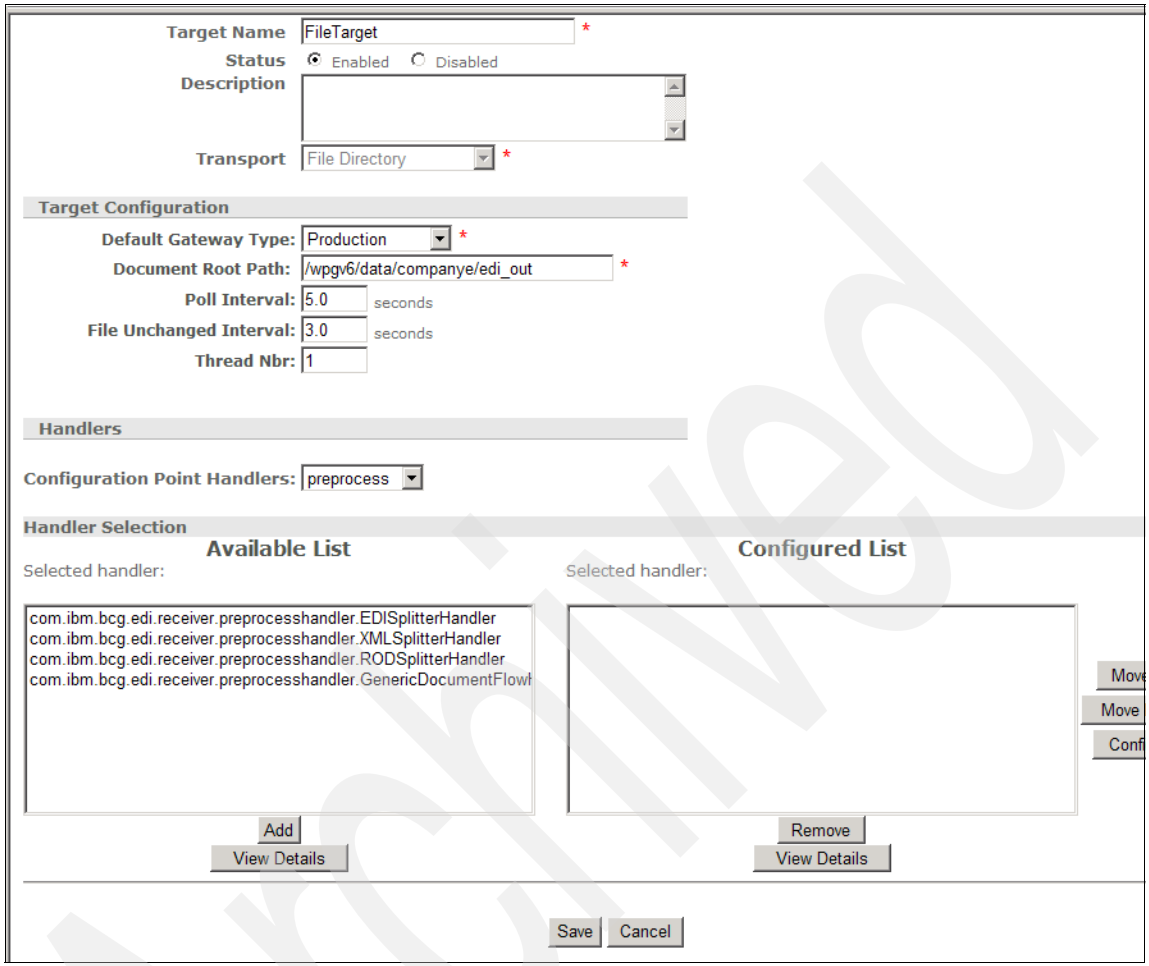


Figure 21-36 Editing the FileTarget to add a preprocess handler

5. Select **com.ibm.bcg.edi.receiver.preprocesshandler.EDISplitterHandler** from the Available List and click **Add**. Look for the EDISpliiterHandler in the Configured List box.
6. With the EDISplitterHandler in the Configured List box, click **Save.**

   The EDI Splitter is now configured on the same target into which you have been putting the EDIFACT files.

The EDI splitter is optional; you can send individual EDI envelopes into a target without configuring an EDI splitter. The splitter is only required if you want to string together multiple EDI envelopes into a single large file. In a real-world scenario this would be largely a function of your back-end system that supplies

Archived

the EDI files to start. Some applications provide EDI envelopes as output, while others can be configured to append these envelopes together.

In terms of the example in this chapter, you can now send multiple EDI envelopes in a single file into this File Directory target, and the EDI Splitter will split them into individual envelopes. For instance, you can append the `edifact3.inp` document it to itself twice, for a total of three envelopes in a single input file. If you drop the new input file onto the FileTarget of Company E, the EDI Splitter will split those three envelopes into individual envelopes and process them just as it did when we sent these envelopes one at a time. If batching is on, it will preserve the envelopes. If batching is off, it will rely on the secular.

The splitting and batching functions mainly come into play when there are large numbers of EDI envelopes, and therefore transactions, passing the WebSphere Partner Gateway server. The next chapter, (Chapter 22, “EDI message tracking” on page 771), discusses some other server features that allow better tracking of EDI exchanges.

Archived

22

EDI message tracking

In this chapter, we review the tools WebSphere Partner Gateway provides to track EDI communications.

Archived

© Copyright IBM Corp. 2005. All rights reserved.

22.1 EDI message tracking overview

In Chapter 11, “Managing the B2B exchange” on page 307, we discussed several tools used to manage the B2B exchange covered in this redbook. For example, the AS Viewer displays only AS packaged messages and groups the documents with their MDN replies. The Document Viewer shows all documents that flow through the WebSphere Partner Gateway Server.

Adding an EDI processing engine to WebSphere Partner Gateway makes the server very versatile, but also adds an extremely large amount of data to process. EDI envelopes are broken into individual messages or transactions for processing. Therefore, the Document Viewer shows not only the inbound envelopes, but all the transactions inside the envelopes that undergo transformation, and then the envelopes that are packaged for sending outbound. EDI envelopes can contain up to 9999 transactions per envelope, and the connections can be configured to use functional acknowledgements as well. This can add up quickly for an Enterprise level implementation.

The EDI tracking functionality of WebSphere Partner Gateway is designed to address some of the issues that arise when dealing with large amounts of EDI messages flowing through the Document Manager.

The features that are available in WebSphere Partner Gateway to help manage your EDI flows are:

- Drill up and Drill down capability of the Document Viewer

  These abilities link transactions to envelopes, and from the detail view in the Document Viewer.

- EDI FA Overdue Report

  This is a WebSphere Partner Gateway generated report that lists all the FA’s that are overdue on the system. The time limit is based on a configurable FA time limit attribute

- EDI Rejected Report

  Another report generated by the server that lists all FAs that were returned to system, and that reported a failed document.

22.2 Drill up and down capability

This section discusses how to implement these capabilities.

22.2.1 Drill down capability

Drill down is the ability to view an EDI envelope in the document viewer, and from within that view, view all the associated transactions or messages in that envelope. You can link to those individual transactions and see their processing result directly from the view of the parent envelope.

This is useful for determining the contents of a specific EDI envelope. For example, a WebSphere Partner Gateway server receives an envelope that has three EDIFACT messages in it. These three messages are to be transformed into EDI-X12 messages (edifact3.inp). By using the drill down capability, it is possible to see the result of those three transformations directly from within the detailed view of the parent envelope.

To demonstrate the drill down capability, you can use the Company E Community Console and the document flow we created previously.

1. Place the `edifact3.inp` from Chapter 21, “Batching and EDI splitting” on page 729 into the FileTarget directory of Company E.

   ```
   wpgv6/data/companye/edi_out/Documents/Production
   ```

2. Log on to the Community Console as the hubadmin user and select **Viewers** → **Document Viewer**.
3. Select the criteria on the Document Viewer Search page that will return the document flows of the file you just sent. You should expect five total rows displayed in the document viewer for this document.
4. Figure 22-1 on page 774 shows the resulting display in the Document Viewer after enveloping. Click the **magnifying glass** icon next to the **EDI-EDIFACT to EDI-EDIFACT** de-enveloping flow (highlighted in Figure 22-1 on page 774). It represents the original edifact3.inp file received by WebSphere Partner Gateway.

Archived

Resend

Participants	Time Stamps		Protocol/Document Flow	Gateway Type	Synchronous	Status
Document ID: 000000026						
Doc Time Stamp: 050805-1900						
Source: Company E	**In:** 8/5/05 7:00:27 PM	(0.719 kb)	EDI-X12 (ALL) ISA: ISA (ALL)			
Target: Company A	**Out:** 8/5/05 7:00:29 PM	(1.134 kb)	EDI-X12 (ALL) ISA: ISA (ALL)	Production		
Document ID: 00001						
Doc Time Stamp: 050802:1053						
Source: Company E	**In:** 8/5/05 6:58:18 PM	(0.885 kb)	EDI-EDIFACT (ALL) UNB: UNB (ALL)			
Target: Company A	**Out:** 8/5/05 6:58:25 PM		EDI-EDIFACT (ALL) UNB: UNB (ALL)	Production		
Document ID: 00001						
Doc Time Stamp: 050802:1053						
Source: Company E	**In:** 8/5/05 6:58:18 PM	(0.197 kb)	EDI00B (ALL) ORDERS: ORDERS (ALL)			
Target: Company A	**Out:**	(0.137 kb)	X12V4R1 (ALL) 850: 850 (ALL)	Production		
Document ID: 00001						
Doc Time Stamp: 050802:1053						
Source: Company E	**In:** 8/5/05 6:58:18 PM	(0.213 kb)	EDI00B (ALL) ORDERS: ORDERS (ALL)			
Target: Company A	**Out:**	(0.15 kb)	X12V4R1 (ALL) 850: 850 (ALL)	Production		
Document ID: 00001						
Doc Time Stamp: 050802:1053						
Source: Company E	**In:** 8/5/05 6:58:18 PM	(0.2 kb)	EDI00B (ALL) ORDERS: ORDERS (ALL)			
Target: Company A	**Out:**	(0.141 kb)	X12V4R1 (ALL) 850: 850 (ALL)	Production		

Resend

Figure 22-1 edifact3.inp in document viewer

5. The detailed view of this envelope is shown in Figure 22-2 on page 775. The Document Children section of this view (highlighted in Figure 22-2 on page 775), is where the drill down ability is located. Document Children is currently set to None, so no documents are displayed. Click **Source** to see the children messages in this envelope.

Archived

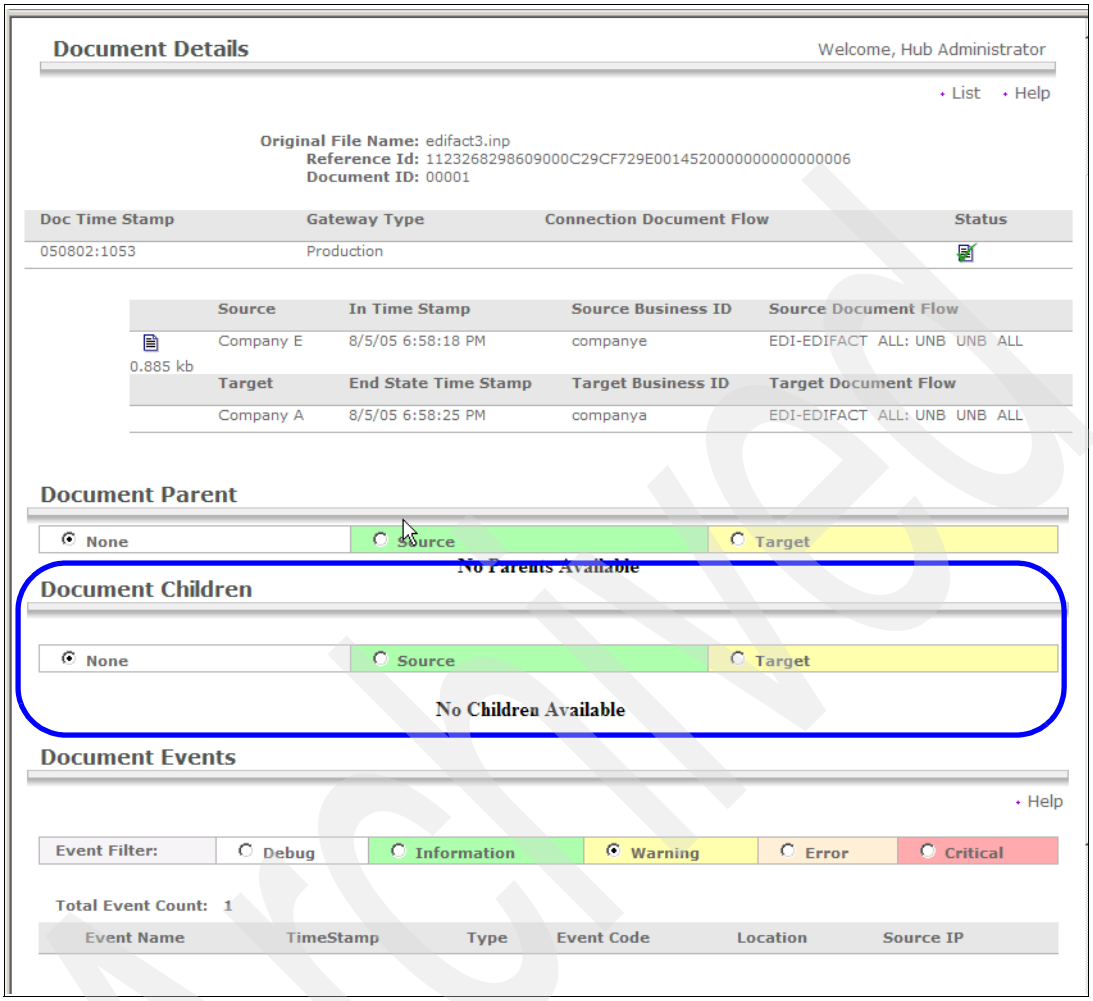


Figure 22-2 Detailed view of inbound envelope

Archived

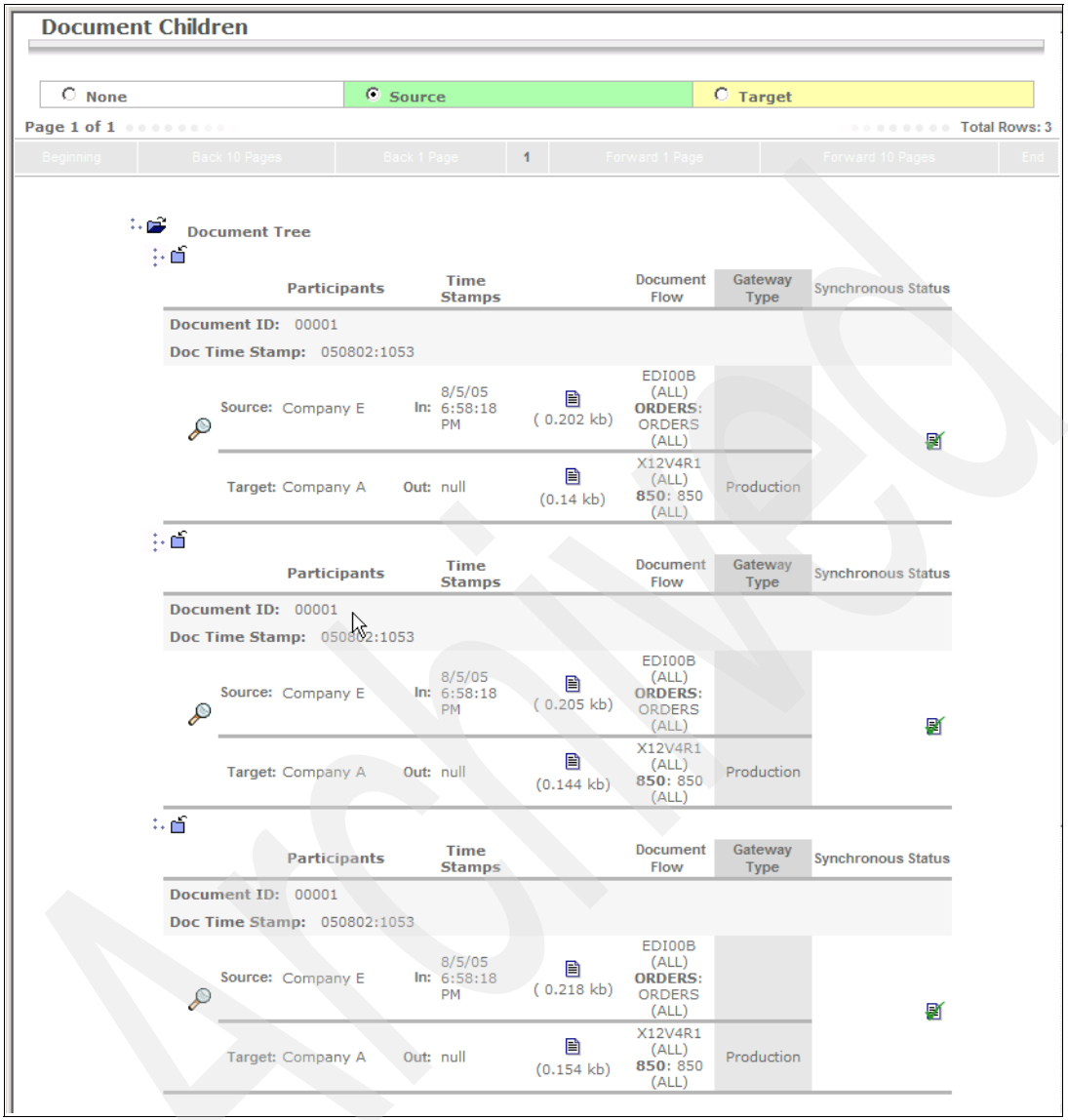


Figure 22-3 Document Children displayed

The view of the document children is shown in Figure 22-3. From this view you can see the individual transactions that were in this source envelope. In addition to seeing them, you can link directly to the detailed view of these transactions from the magnifying glass icons. The status is shown on the right side of the view. The status also gives you a quick look at the result of the child transaction.

If the transaction had failed, you could link directly to it, and view the details of why it failed.

Drill down can be used on any EDI envelope, not just the ones inbound to the server. For example, you can use this capability to view an envelope that was sent out of the system and link to the transactions that made up the envelope and their originators.

Instead of using the original EDIFACT envelope to view the children, use the Document Viewer to examine the outbound envelope that is sent to Company A. To do this, follow these steps:

1. Go back to the Document Viewer screen that showed the five total documents from the previous example (Figure 22-4 on page 778). This time, click the **magnifying glass** next to the envelope that was sent out of the system (Figure 22-4 on page 778).

Archived

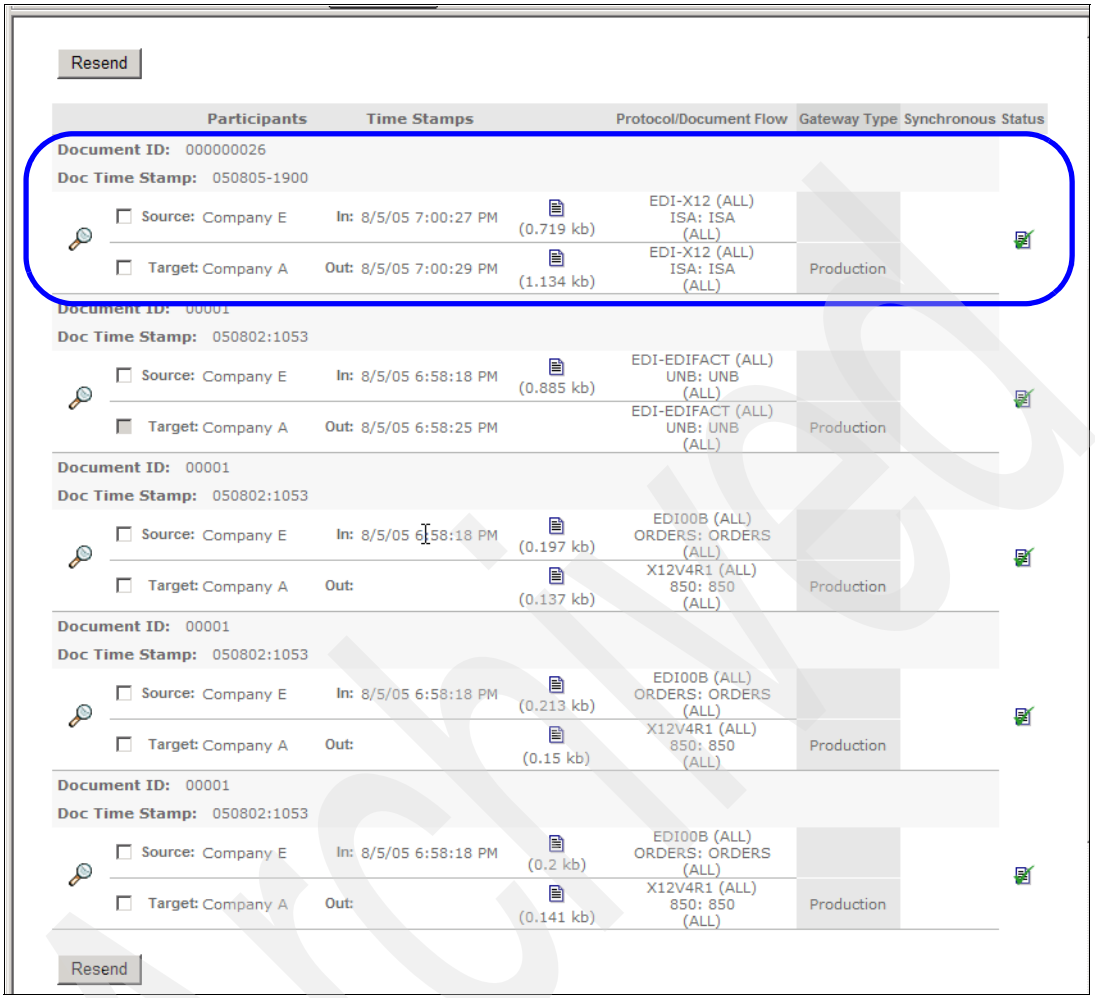


Figure 22-4 Selecting the outbound envelope.

2. The detailed view of this envelope is shown in (Figure 22-5 on page 779). The transactions that are inside this envelope are not source children, because they did not come from this envelope. They are target children since they were enveloped into this document for outbound transmission. Click **Target** (Figure 22-5 on page 779).

Archived

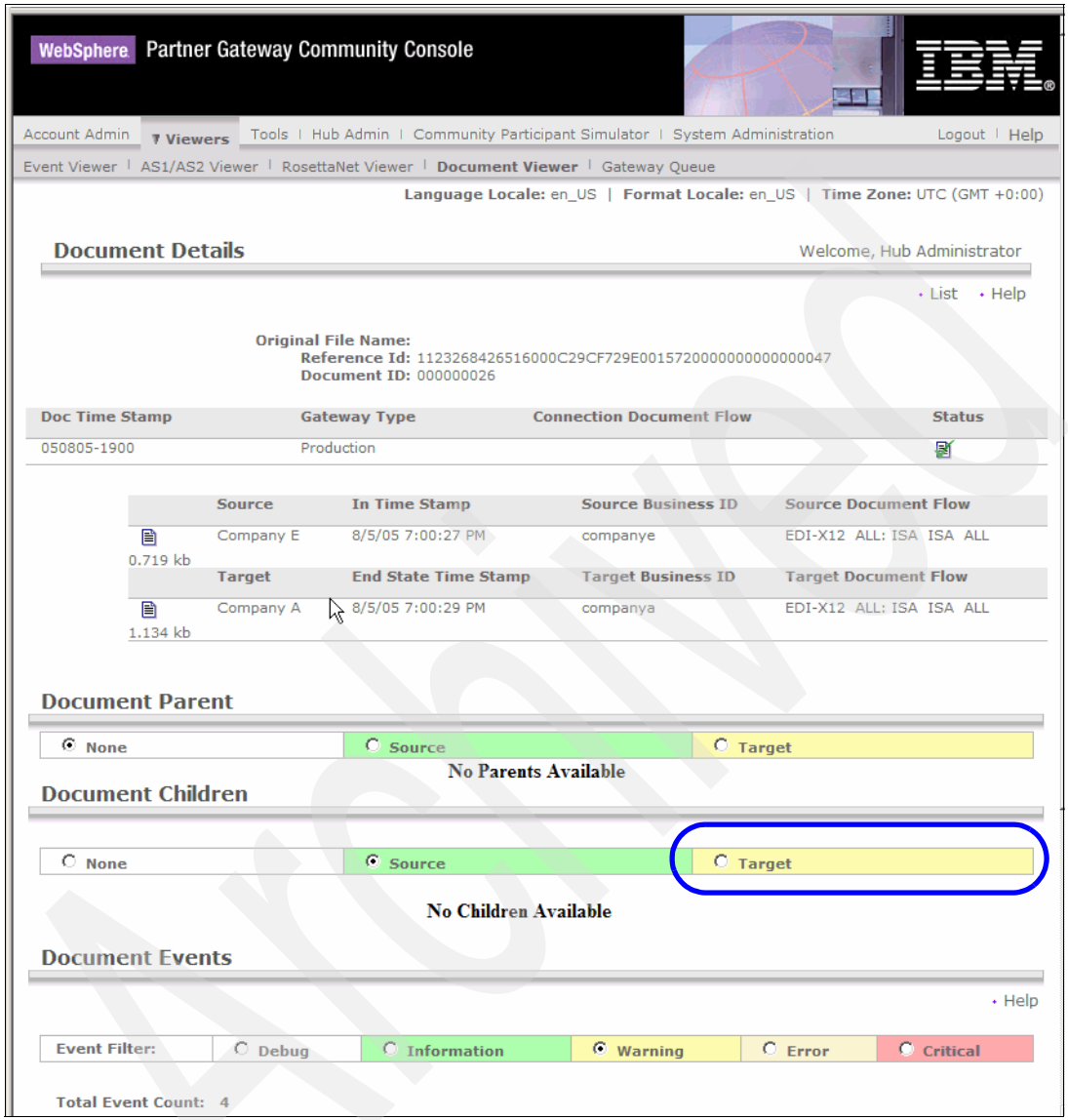


Figure 22-5 Selecting target children view

The resulting view is shown in Figure 22-6 on page 780. Again, you can see that children of an envelope (messages or transactions) can be directly linked from this envelope view.

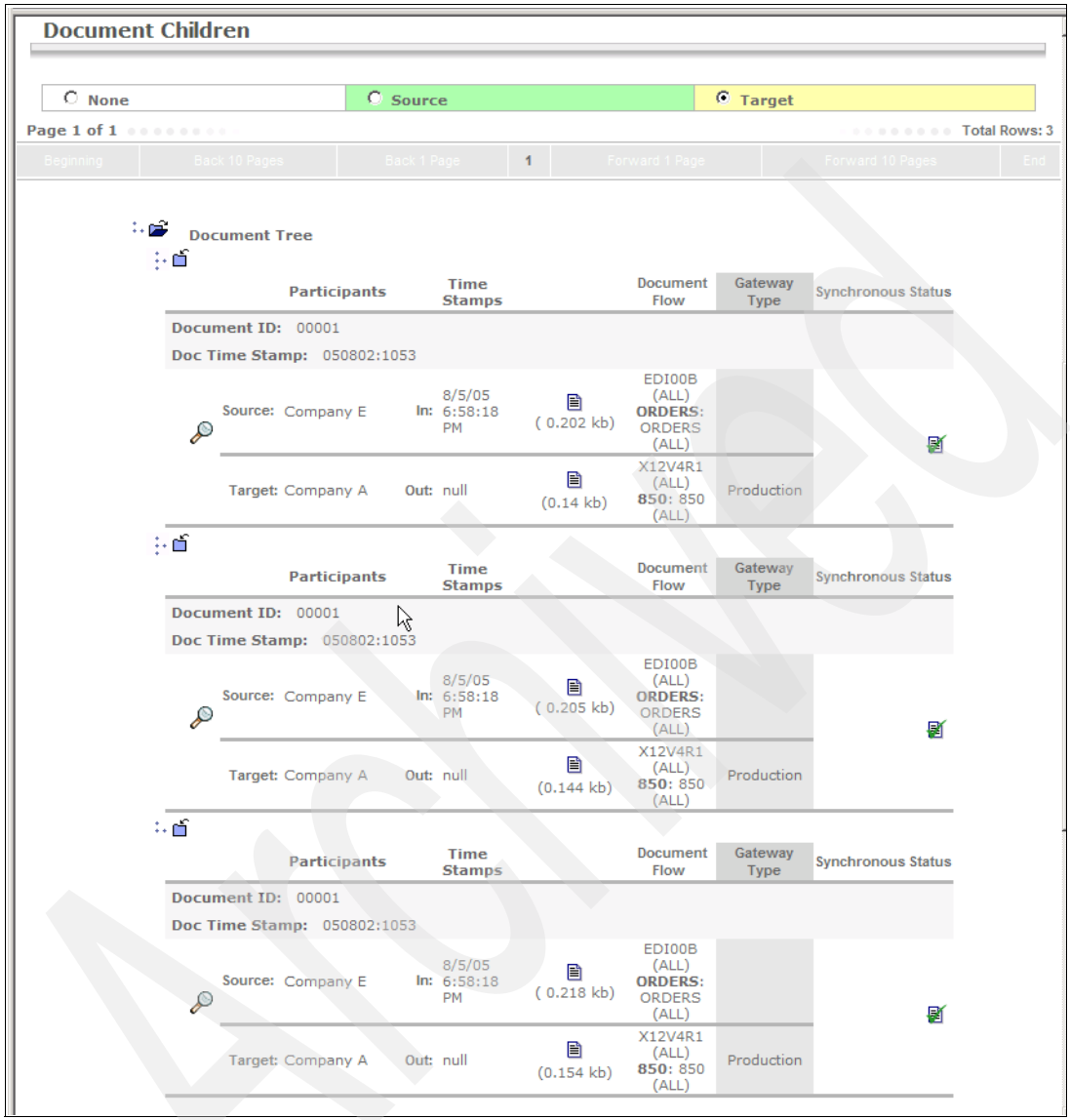


Figure 22-6 Target children view

22.2.2 Drill up capability

As you would expect, the *drill up* capability is the exact opposite of drill down. From the detailed view of an individual transaction you can link to its associated

parent envelope. This is especially useful when trying to determine in which envelope a specific transaction was received or sent.

This can also be very useful in an error scenario. For example an EDI document is received that has several transactions. One of the transactions fails to transform due to a formatting error. By using the drill up capability of the Document Viewer, you can link from that failed transaction directly to the envelope to determine the Participant at fault.

To demonstrate drilling up, we can again use the edifact3.inp document we used in the last section. However, this time we want to introduce an error into one of the transactions.

The original edifact3.inp document is shown in Example 22-1.

Example 22-1 edifact3.inp file

```
UNB+0001:+companye::+companya::+050802:1053+00001++++++++++++'
UNH+000001+ORDERS:EDI:00B:EN:+++++++'
BGM+105:::+WIDGETORDER:01++US'
DTM+137:050802:101'
NAD+COD+STANZA::+STANZA CO:RALEIGH:::+++++27709+NC'
LIN+WIDGET+++++'
PIA+5+BLUE WIDGET:PO::+RED WIDGET:PO::+YELLOW WIDGET:PO::++'
UNS+S'
MOA+146:2456.75::9:'
UNT+9+000001'
UNH+000002+ORDERS:EDI:00B:EN:+++++++'
BGM+105:::+WIDGETORDER:01++US'
DTM+137:050802:101'
NAD+COD+STELLA::+STELLA CO OP:DURHAM:::+++++27719+NC'
LIN+WIDGET+++++'
PIA+5+PINK WIDGET:PO::+WHITE WIDGET:PO::+BLACK WIDGET:PO::++'
UNS+S'
MOA+146:3456.25::9:'
UNT+9+000002'
UNH+000003+ORDERS:EDI:00B:EN:+++++++'
BGM+105:::+WIDGETORDER:01++US'
DTM+137:050802:101'
NAD+COD+WESTSIDE::+WESTSIDE SOCIETY:DURHAM:::+++++27719+NC'
LIN+WIDGET+++++'
PIA+5+TWISTED WIDGET:PO::+CURLED WIDGET:PO::+STRAIGHT WIDGET:PO::++'
UNS+S'
MOA+146:5000.00::9:'
UNT+9+000003'
```

Archived

To create an error, create a mismatch in the message control number for the second message in the group, as shown in Example 22-2.

Example 22-2 edifact3err.inp file (edifact3.inp with error introduced)

```
UNB+0001:+companye::+companya::+050802:1053+00001++++++++++++'
UNH+000001+ORDERS:EDI:00B:EN:+++++++'
BGM+105:::+WIDGETORDER:01++US'
DTM+137:050802:101'
NAD+COD+STANZA::+STANZA CO:RALEIGH:::+++++27709+NC'
LIN+WIDGET+++++'
PIA+5+BLUE WIDGET:PO::+RED WIDGET:PO::+YELLOW WIDGET:PO::++'
UNS+S'
MOA+146:2456.75::9:'
UNT+9+000001'
UNH+000002+ORDERS:EDI:00B:EN:+++++++'
BGM+105:::+WIDGETORDER:01++US'
DTM+137:050802:101'
NAD+COD+STELLA::+STELLA CO OP:DURHAM:::+++++27719+NC'
LIN+WIDGET+++++'
PIA+5+PINK WIDGET:PO::+WHITE WIDGET:PO::+BLACK WIDGET:PO::++'
UNS+S'
MOA+146:3456.25::9:'
UNT+9+000099'
UNH+000003+ORDERS:EDI:00B:EN:+++++++'
BGM+105:::+WIDGETORDER:01++US'
DTM+137:050802:101'
NAD+COD+WESTSIDE::+WESTSIDE SOCIETY:DURHAM:::+++++27719+NC'
LIN+WIDGET++++'
PIA+5+TWISTED WIDGET:PO::+CURLED WIDGET:PO::+STRAIGHT WIDGET:PO::++'
UNS+S'
```

Then save the new file as `edifact3err.inp`. This file is also provided in the Additional Materials.

Now that you have a new file, place it on the FileTarget for Company E. In the same directory as before:

```
/wpgv6/data/companye/edi_out/Documents/Production
```

Use the document viewer to view the results of this new file. Again, five documents are expected. They are shown in Figure 22-7 on page 783. The intentional error has worked, and one of the transaction transformations has failed.

Archived

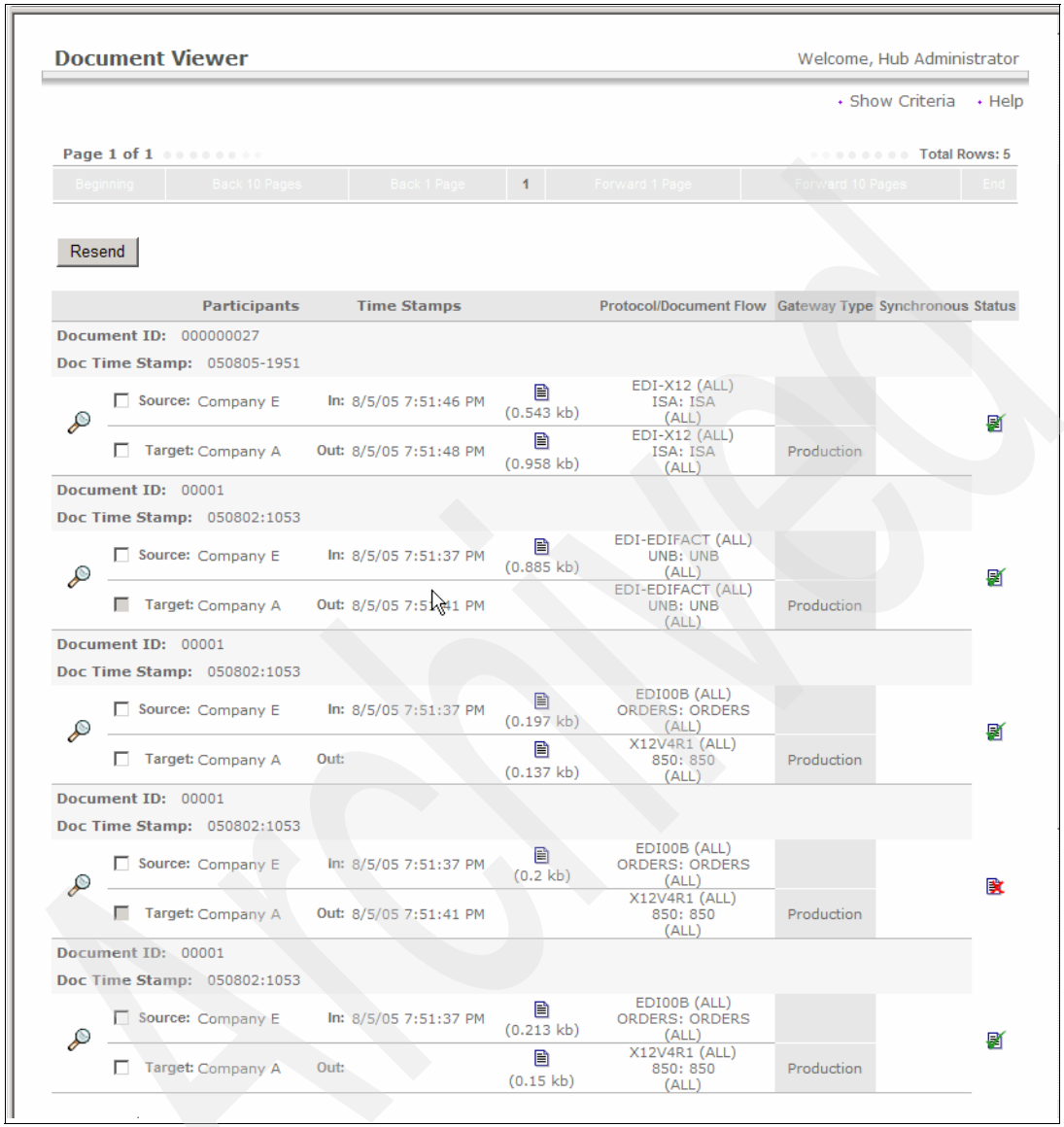


Figure 22-7 Document Viewer view of edifact3err.inp

To use the drill up capability, follow these steps:

1. Click the **magnifying glass** for the failed transaction. The detailed view is shown in Figure 22-8 on page 784.

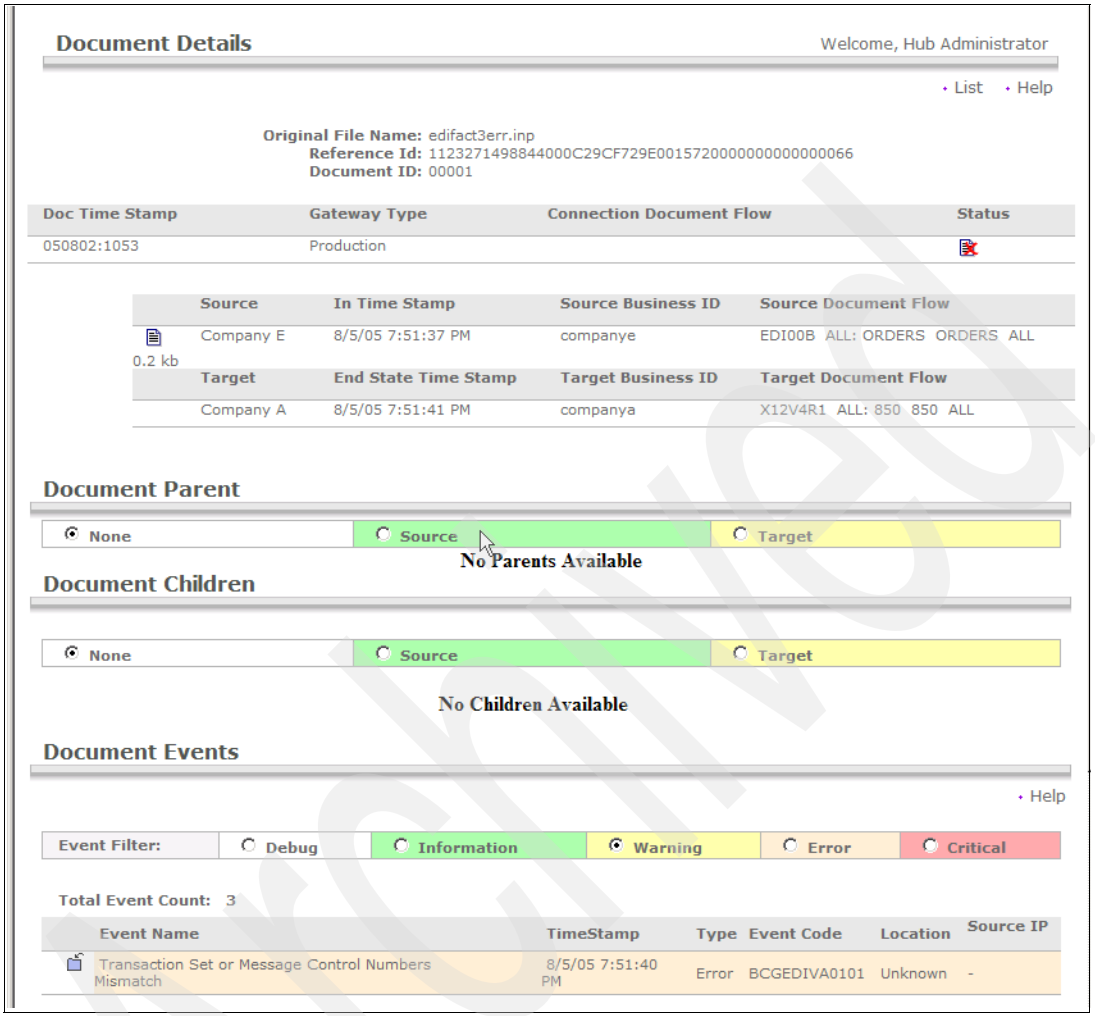


Figure 22-8 Detailed view of Failed transaction

2. This is the view of an EDIFACT message, so it is a child document. Click the **Source** under Document Parent to list the parent envelope that this failed message was received from. This is shown in Figure 22-9 on page 785.

Archived

Figure 22-9 Document Parent listed

To illustrate drill up and drill down capability, we have used relatively simple examples. A single EDIFACT envelope that contained only three messages was used. When the EDI documents that are traversing the WebSphere Partner Gateway system contain thousands of transactions or messages, you can see how the ability to link to its parent envelope or children transactions is a crucial function.

Archived

22.3 EDI Reports

EDI Reports is a function available on the Community Console under the Tools section. The use of these reports are similar to that of the document viewer. You provide certain criteria to the search page of the report, such as time, source and Target Participant, protocol, and click **Search**. WebSphere Partner Gateway searches all messages on the system within your search criteria and returns all documents that meet your criteria *and* the criteria of the report itself. There are two reports currently available:

- EDI FA Overdue Report
- EDI Rejected Report

Both of these EDI reports also give you the ability to export the results as a text file.

22.3.1 EDI FA Overdue Report

The EDI FA overdue report is applicable only to EDI communication channels that use functional acknowledgements. The report is tied to the FA time limit attribute in the EDI connection properties. This attribute would be used when two participants agree, most likely in the form of an service level agreement (SLA), on the amount of time the FAs should take to process. The report lists all document transactions that expect an FA, but which did not receive one in the agreed upon time frame.

The EDI FA overdue report search screen is shown in Figure 22-10 on page 787.

Archived

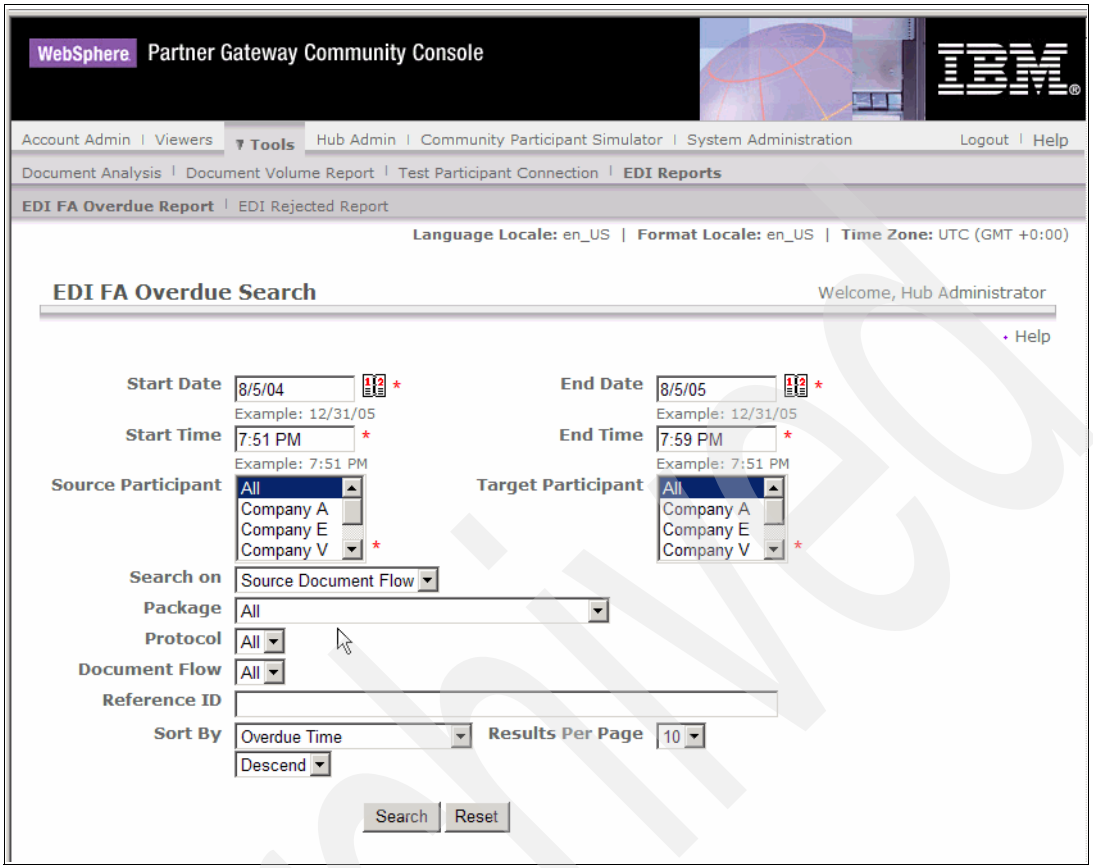


Figure 22-10 EDI FA Overdue Search

22.3.2 EDI Rejected Report

The EDI Rejected Report functions in a very similar manner to the EDI FA Overdue report. The EDI Rejected Search page allows you to select several parameters such as time frame and participants involved (Figure 22-11 on page 788). All FAs that fall within those parameters *and* that report a rejected EDI document are displayed.

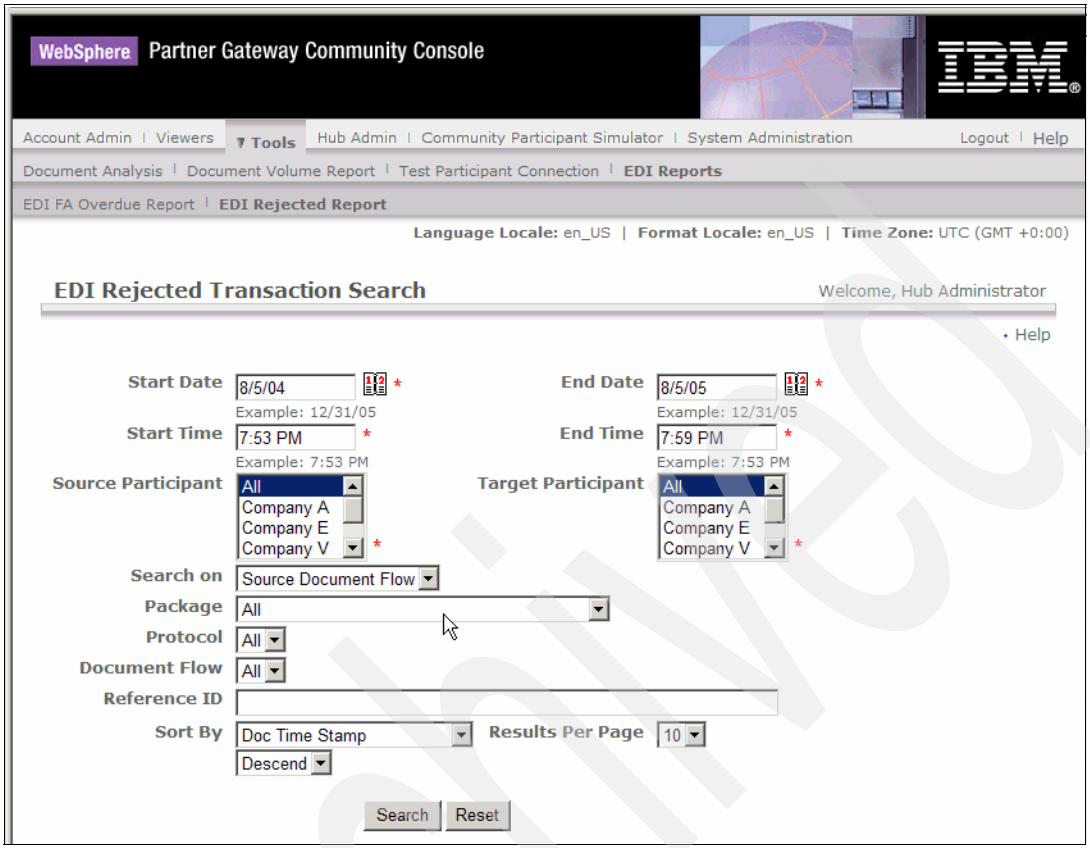


Figure 22-11 EDI Rejected Transaction Search

The data displayed by either of these EDI reports can be found in the Document Viewer as well. The reports themselves just make the gathering of this data easier by selecting very specific criteria, such as overdue FAs or rejected FAs. How often you use these tools will be highly dependent on your environment, and whether you intend to use EDI communication with functional acknowledgements.

Archived

Part 6

Appendixes

Archived

© Copyright IBM Corp. 2005. All rights reserved.

Archived

A

Additional material

This redbook refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

ftp://www.redbooks.ibm.com/redbooks/SG247109

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the redbook form number, SG247109.

Using the Web material

The additional Web material that accompanies this redbook includes the following files:

File name	*Description*
SG247109.zip	Sample data files and DIS maps

© Copyright IBM Corp. 2005. All rights reserved.

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space: 1 MB minimum
Operating System: Windows 2000
Processor: 1.7 GHz or higher
Memory: 2GB

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.

Archived

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 793. Note that some of the documents referenced here may be available in softcopy only.

- *B2B Solutions using WebSphere BI Connect Version 4.2.2,* SG24-6355-01

Online resources

These Web sites and URLs are also relevant as further information sources:

- WebSphere Business Integration Information Center

  http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp

- WebSphere Partner Gateway Information Center

  http://www-306.ibm.com/software/integration/wspartnergateway/library/infocenter/index.html

- WebSphere Partner Gateway Support

  http://www-306.ibm.com/software/integration/wspartnergateway/support/

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Archived

© Copyright IBM Corp. 2005. All rights reserved.

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Archived

Index

A
Access Control List 26
  See ACL
accounts, add the user 111
ACL
  what is 26
action
  create 528
  create custom action 643
activate
  attribute selection 657, 684, 712
  B2B capabilities 350, 354
  interactions 749
  participant connections 182–183, 188, 193, 246, 253, 300–301, 355, 366, 397, 404, 438, 534, 566, 568, 625–626, 657, 684, 702, 712, 749
  participant connections and attribute selection 657, 684, 712, 751
  user alerts 272
activated participant connections
  configure the 751
add
  attributes to document flow definition 640
  user and group 86
add attributes to document flow definition 640
AIX
  create user 362
  integrate with the FTP server 361
analyze a transaction
  997 697
  TA1 696
ANSI X12
  what is 25
Application Data format (ADF) 456
application integration architecture, point-to-point 11
AS2 identifier in WebSphere Partner Gateway Express 273
associate FA map with EDI document flow definition 700
authentication, FTPS client 383
automatic sending 305

B
B2B capabilities 171
  enabling 273
  providing 171
  update 738
  update for EDI splitting 738, 741
  update for FTP integration 386
B2B data communications, evolution of 9
B2B data structures, evolution of 7
B2B integration, types of 13
B2B, business-to-business 5
B2Bi, business-to-business integration 6
B2C, business-to-consumer 5
B2M2B, business-to-marketplace-to-business 7
batch enveloper and de-enveloper 588
batching
  using to alter output 756
  what is 766
bcgDIS.sql 457
bcgDISImport.bat 632
BEG Segment 592
business-to-business integration, B2Bi 6
business-to-business, B2B 5
  features and standards 40
  requirements 22
business-to-consumer, B2C 5
business-to-marketplace-to-business, B2M2B 7

C
certificate 28
  upload 377
  upload to partner server 220
  uploading to WebSphere Partner Gateway Express 440
certificates
  and digital signatures 28
client authentication, FTPS 383
command
  bcgDISImport.bat 632
  db2level 62
  dspmqver 61
command groups
  assignment 501

© Copyright IBM Corp. 2005. All rights reserved.

conditional mapping commands 503, 505
functions 503
repetitive mapping 505, 509
common data 83
common storage 36
communication 583
validate 433, 439
validate inbound 368, 447
validate outbound 357, 445
VAN 584
Community Console 35
what is 35
Community Manager
create 155
initial logon 165
what is 39
community participant 39
create 161, 284, 343, 423
WebSphere Partner Gateway Express 272
enabling B2B capabilities
in WebSphere Partner Gateway Express 273
components
configuration 37
run-time 33
composite element 581
conditional mapping commands 505
Configuration Point Handler 517
configure
enveloper 649
FTP Scripting target 434, 446
FTPS gateway 372
WebSphere MQ 65
Control strings 469
control strings 469
create
action 528
Community Manager 155
community participant 161, 284, 423
WebSphere Partner Gateway Express 272
custom action 643
document flow definition 285
envelope profile 650
field definition 477
file system gateway 166, 192
FTP Scripting gateway 389, 392, 443
FTP scripting gateway 392, 414
FTP Scripting target 405, 434
FTPS gateway 427
gateway 166
HTTP gateway 297
interaction 149, 293, 424, 528, 531, 564, 645, 707
JMS target 630
participant connection 355, 431, 533, 565–566, 568
queue manager 65
record definition 478
record ID 476
Record Oriented Document definition
See ROD definition
ROD definition 477
ROD dictionary 476
target 144
target for ROD document 516
target FTPscript 405
XML format 290, 634
custom action
create 643
custom transports
packages and protocols 42
customize
My Profile 280
cXML 8

D

data elements
EDI 465
Data format (DF) 456
Data Interchange Services client
bcgDIS.sql 457
compile map 510
components 459
configure 457
control string 469
create database 458
data transformation map
create 489
edit 493
data transformation maps 468
development database 460
development environment 456
DTD document definition 462
EDI code list 465, 469
EDI data element 465
EDI dictionary 464
EDI document definition 464

Archived

EDI functional area 463
EDI segment 464
export control string 512
export namespace objects 513
field
create 477
field exit 470
functional acknowledgement map 469
functional areas 461
global variable 469
install 452
loop 467
create 483
map editor 493
command groups 500–501
help 497
mapping command 494
source document definition 494
target document definition 494
variable 495
Mapping functional area 467
MapSwitch 542
namespace object 463
record 467
record ID 466, 478
create 476
Record Oriented Data functional area 465
ROD definition
create 477
ROD dictionary 466
create 476
ROD document definition 466
ROD field 467
ROD functional area 465
ROD structure 467
schema document definition 462
create 486
view 487
translation table 469
user exit 470
validation map 468
what is 473
XML dictionary 462
create 484
XML functional area 461
Data Interchange Services client databases
Configuration database 470
Data Interchange Services client database 470
Document Manager database 471
data structures
evolution of the B2B 7
industry-oriented XML 8
international XML 9
data structures for EDI 7–8
data transformation map 468, 473
data validation 622
database authentication 63
database owner 79
database performance considerations 82
database schema
install 79
DB2
verify level
Windows 62
default password
WebSphere Partner Gateway 104
default user
WebSphere Partner gateway 104
define
group
Windows 64
user
Windows 63
digital signature
verify 257
digital signatures 28
digital signatures and certificates 28
DIS client
See Data Interchange Services client
document definitions
DTD 462
EDI 464
ROD 466
Schema 462
document exchange 15
document flow definition
add attributes 640
associate FA map 700
associating FA map with 700
create 285
create interactions for 680
Document Manager 35
document type definition 29
drill down capability 773
drill up capability 780
DT map
See data transformation map
DTD

Archived

See document type definition
DTD document definition 462

E

e-business 4
ebXML 9
e-commerce 4
EDI 4
  and e-commerce 4
  ANSI X12 8
    sample message 582
  AS1 25
  AS2 25
  BEG Segment 592
  benefits 580
  communication 583
  components 581
  cXML 8
  document definitions 464
  document exchange 15
  document structure 581
  ebXML 9
  EDIFACT 8
    sample message 583
  evolution 585
  first era 7
  fourth era 9
  GENCOD 8
  message format standards 25, 581
  message standards 581
  messaging 15
  network bandwidths 15
  over the Internet 25, 585
  over value-added network EDI/VAN 25
  second era 8
  solution components
    batching 588
    trading partner agreements 588
    translator 587
  solution elements 587
  terms, concepts 578
  third era 8
  TRADACOMS 8
  UN/CEFACT 9
  VAN 9, 584
  VDA SEDAS 8
  what is 25
  xCBL 8
  XML 8
  XML based message formats 15
EDI data elements 465
EDI dictionary 464
  create 464
EDI document definition 464
EDI electronic data interchange 577
EDI FA Overdue Report 786
EDI functional area 463
EDI INT 585
EDI message tracking
  drill down 773
  drill up 780
EDI over the Internet 25, 585
  See EDI-INT
EDI over value-added network 25
  See EDI/VAN
EDI over value-added network, EDI/VAN 25
EDI Rejected Report 787
EDI segments 464
EDI solution message router 588
EDI solution, Elements of 587
EDI splitting
  update B2B capabilities 738, 741
  validate 754
  validate the flow 754
EDI standards 8
EDI to XML transformation map
  upload 675
EDI to XML transformation map, Uploading the 675
EDI validation map
  upload 677
EDI validation map, Upload the 677
EDI, data structures for 7–8
EDI, electronic data interchange 577
EDI/VAN EDI over value-added network 25
EDIFACT 8
  what is 25
EDIFACT to EDI-X12 map 733
EDIFACT to EDI-X12 map, The 733
EDI-INT
  AS1 25
  AS2 25
  what is 25
EDI-X12 interchange, configure Company A to post an 674
EDI-X12 map, The EDIFACT to 733
Electronic data interchange 25
electronic data interchange

Archived

See EDI
electronic data interchange (EDI) 25, 577
element 581
element, composite 581
Elements of an EDI solution 587
elements, EDI data 465
e-Marketplace 7
enable JMS 627
encryption 27
  Public Key Infrastructure 27
  secret key cryptography 27
encryption is enabled, validate that 241
encryption, problems with 323
encryption, symmetric key 27
encryption, validation and transformation 42
envelope 581
envelope profile
  create 650
envelope profile, create 650
enveloper scheduling interval, setting 656
enveloper, configure 649
enveloping
  batch mode off 761
  batch mode on 756
enveloping and de-enveloping
  add attributes to document flow definition 640
  attribute selection
    activate 657, 684
  create interactions for document flow definition 680
  update
    profile 648, 683
  upload validation map 637
enveloping with batch mode off 761
enveloping with batch mode on 756
evolution of B2B data communications 9
evolution of the B2B data structures 7
eXtensible Markup Language (XML) 29

F

field definition
  create 477
field definition,create 477
First era
  National and industry-oriented EDI data structures 7
  Point-to-point direct connections 9
format standards, message 25
format, create XML 290, 634
Fourth era
  International XML data structures 9
FTP 26
  create the directory structure 341
  what is 26
FTP integration
  update B2B capabilities 386
FTP script
  assign to gateway 407
FTP Scripting
  overview
    inbound 404
    outbound 385
FTP scripting
  create a script for use by the target 405
FTP Scripting gateway 414
  create 389, 443
  update to use secure mode 443
FTP Scripting gateway, create 392
FTP Scripting inbound, overview 404
FTP Scripting outbound, overview 385
FTP Scripting target
  configure 434
  create 405, 434
  FTPS 415
  update 446
  validate 412
FTPS
  client authentication 383
  create gateway 414
  FTP Scripting gateway 414
  FTP Scripting target 415
FTPS gateway
  configure 372
  create 392
  features 389
  securing 418
  what is 384
FTPS inbound 384
FTPS outbound 371
  validate 382
FTPS,client authentication 383
functional acknowledgement map 469
functional acknowledgements
  attribute selection
    activate 712
  Overdue Report 786
  set attributes to trigger FA generation 702

Archived

update
profile 710
functional area
EDI 463
Mapping 467
Record Oriented Data (ROD) 465
XML 461

G

gateway 38
Account Admin 106
and Community Manager 158
and Community Participant 175
and Document Manager 140
assign FTP script 407
configuration component 37
configure an FTPS 372
connection parameters for AS2 194
create 166
create FTP Scripting 392
default gateway 169
file system 143, 192
forward proxy 175
from target to 39
FTP 352
FTP Scripting 389
FTPS 372, 392, 414, 427
features 389
what is 384
FTPS for FTP Scripting 414
HTTP 38, 297
HTTP POST 196
interaction between target and 38
JMS 38
list 206
lists 169
queue 318
queue viewer 318
queued documents 319
securing the connection 418
select gateway for connection 183
status 206
target URI 141, 298
test connection 196
what is 38
GENCOD 8
generate private/public key pair 226
generate public/private key pair 212
global variables 469
group
define in Windows 64

H

hashing 27
message digest algorithm MD5 28
secure hash algorithm SHA 27
Healthcare Information Portability and Accountability Act
See HIPAA
help system, Starting the 135
HIPAA 25
HTTP 26
what is 26
hubadmin 104
hubadmin profile, update 343
hubadmin, password of 104
hubadmin, update profile of 422

I

inbound communication
using secure mode for 446
validate 368, 447
with FTP Scripting target, validate 412
inbound communication with FTP Scripting target, 412
inbound, FTPS 384
industry-oriented XML data structures 8
initial logon by Community Manager 165
install
WebSphere Partner Gateway Advanced 122
WebSphere Partner Gateway Enterprise 87
WebSphere Partner Gateway Express 263
WebSphere Partner GatewayEnterprise 87
install database schema
WebSphere Partner Gateway Advanced 114
WebSphere Partner Gateway Enterprise 79
Integrate with the AIX FTP server 361
integrating FTP servers
B2B capabilities 386
integration architecture, point-to-point application 11
interaction 179, 622
activate 749
and EDI envelopes 756
and envelopes in EDI batching 763
create 149, 153, 293–294, 347–348, 564,

Archived

645–646, 680–681, 707, 709, 744–745, 747
Detailed Validation of Segment action 623
EDI Validate action 622, 646
for document flow definition 680
manage 149, 348, 645
message transformation and EDI batching 759
Pass through action 708, 754
RosettaNet 308
search for 154
what is 37–38
XML Translate action 646
XML Validate action 646
interaction, create 293, 347, 387, 424, 528, 531, 564
Interactions
what are 38
interval validation
long 672
short 668

J

JMS
enable 627
verify installation
Windows 61
JMS target
create 630

K

key cryptography, secret 27

L

load map into WebSphere Partner Gateway 735
log files
WebSphere Partner Gateway 103
logging 79
long interval validation 672
Loop 467
N1 596
PO1 597
PO1 Loop 597

M

manual sending 303
map 514, 556, 735
Data transformation 468
EDIFACT to EDI-X12 733
Functional acknowledgement 469
Transformation 631
Upload 677
Uploading 675
Validation 468
map editor
command groups 501
mapper 587
mapping
communication 569, 571
create participant connection 566, 568
functional area 467
source document 473
target document 473
MDN HTTP URL not defined 321
menu options in WebSphere Partner Gateway 106
message digest algorithm (MD5) 28
message format standards 25
message queuing 23, 585
message router 588
message router, EDI solution 588
message standard 581
message standards 581
messaging and queuing 23
messaging protocol options 41
metadata 29
MIME 28
MIME and S/MIME 28
My Profile
customize 280

N

N1 Loop 596
Namespace objects 463
Negative validation 692
NLS support 107

O

OASIS 9
objects, Namespace 463
objects, Record ID information 466
ODETTE 25
Operator profile 39
Organization for Data Exchange through Teletransmission in Europe
See ODETTE
Organization for the Advancement of Structured Information Standards

Archived

See OASIS
outbound
FTPS 371
validate FTPS 382
outbound communication
validate 357, 445
outbound flow 140
outbound FTP Scripting overview 385
outbound Implementation Steps 339
Overdue Report, EDI FA 786
overview
FTP Scripting inbound 404
FTP Scripting outbound 385
overview,
Batching 730

P

participant connection
configure 751
create 355, 431, 533, 565
update 240, 375, 397, 410
participant connections and attribute selection
activate 657, 684, 712
Participant Profile
update 648, 682, 710
password of hubadmin 104
password policy 107
performance considerations, database 82
PO1 Loop 597
point-to-point application integration architecture 11
policy, password 107
Positive validation 688
post EDI-X12 interchange 674
preprocess
using a configuration point handler 517
using the RODSplitter 517
problems with digital signatures 325
problems with encryption 323
Process integration 17
profile
built-in 39
Operator 39
Profile management 39
properties of a JMS target 630
protocol
transport 26
providing
B2B capabilities 171
public key cryptography 27
Public Key Infrastructure (PKI) 27

Q

queue 26
Queue manager and publish/subscribe broker 35
queuing 24

R

Receiver 34
Record definition
create 478
record definition
create 478
Data Interchange Services client
record definition
create 478
Record ID
create 476
record ID
create 476
Record ID information objects 466
Record Oriented Data (ROD)
functional area 465
Record Oriented Data (ROD) functional area 465
Record Oriented Document definition
create 477
See ROD definition
Records 467
Redbooks Web site 793
Contact us xvii
Rejected Report, EDI 787
Report, EDI FA Overdue 786
Report, EDI Rejected 787
requirements
for B2B 22
review, Batching 766
ROD
document definitions 466
See Record Oriented Data
ROD dictionary 466
create 476
ROD document
create target for 516
ROD document definition 466
ROD functional area
Record Oriented Data 465

Archived

RODdefinition
create 477
RODSpliiterHandler
what is 518
RODSplitterHandler 517
configure 518
RosettaNet 8–9
router, Message 588
router, message 588
run-time components 33
run-time parameters of WebSphere Partner Gateway 98

S

S/MIME 28
S/MIME, MIME and 28
scheduling interval, set for enveloper 656
schema 29
Schema document definition 462
script file, writing 390
script for use by the target
create 405
Second era
International EDI data structures 8
Value-added networks 9
secret key cryptography 27
secret key cryptography, 27
secure hash algorithm (SHA) 27
secure mode 443
secure mode for inbound communication, using 446
secure mode, update FTP Scripting gateway to use 443
Secure Sockets Layer (SSL) 28
Secure Sockets Layer SSL 28
security
ACL 26
certificates 28
digital signatures 28
encryption 27
hashing 27
security options in WebSphere Partner Gateway 41
segment 581
send XML documents 302–303
sending
Automatic 305
Manual 303
Service Segment validation 623
set attributes to trigger the FA generation 702
set the enveloper scheduling interval 656
Short interval validation 668
signature verification 243
SMTP 26
IMAP 26
POP3 26
what is 26
SSL (Secure Sockets Layer) 28
standards, EDI 8
standards, Message 581
standards, message format 25
start
the help system 135
WebSphere Partner Gateway components
Windows 103
structures, Evolution of the B2B data 7
structures, industry-oriented XML data 8
structures, international XML data 9
support, NLS 107
symmetric key encryption 27

T

TA1 transaction, analyze a 696
tables, translation 469
target 37
configure 406
create 144, 345
create JMS 630
create script 405
properties for JMS 630
what is 34, 37
target for ROD document
create 516
Third era
Internet 10
National and industry-oriented XML data structures 8
TRADACOMS 8
Trading Partner Agreement (TPA) 588
transaction 581
analyze a 997 697
analyze a TA1 696
transformation map 631
Data 468
upload 675
Translation tables 469
translator 587

Archived

translators 587
Transport options 40
transport protocol 26
  FTP 26
  HTTP 26
  SMTP 26
Transport protocols 26
trigger FA generation 702
Types of B2B integration 13

U

UCS 25
UDDI Universal Description, Discovery, and Integration 29
UN/CEFACT 9
United Nations Trade Data Interchange (UNTDI) 25
United Nations Trade Data Interchange Standards
  See UNTDI
Universal Multi-Octet Coded Character Set
  See UCS
update
  B2B capabilities 386
  FTP Scripting gateway to use secure mode 443
  participant connection 375, 397, 410
  participant connections 240
  Participant Profiles 648, 682, 710
  profile 299, 350
  ubadmin profile 343
update configuration
  for maps 523, 561
update FTP Scripting gateway to use secure mode 443
update profile of hubadmin 422
upload
  certificate 377
  certificates in WebSphere Partner Gateway 440
  EDI to XML transformation map 675
  EDI validation map 677
  map using bcgDISImport.bat 632
  private key to own server 217, 231
  public certificate to partners server 220
  public key to the partners server 235
  XML to EDI Map using bcgDISImport.bat 632
  XML validation map 637
URL not defined, MDN HTTP 321
use bcgDISImport.bat to upload the XMI to EDI Map 632
use DB2 to configure a development environment 457
using batching to alter output 756
using secure mode for inbound communication 446

V

validate
  communication, VAN connectivity 433
  database schema installation 85
  digital signatures are enabled 257
  EDI splitting 754
  FTPconnection 399
  FTPS outbound 382
  inbound communication 368
  inbound communication with FTP scripting target 412
  outbound communication 357, 445
  that encryption is enabled 241
validate that digital signatures are enabled 257
validation map 468
  upload EDI validation map 677
  upload XML validation map 637
value-added network
  See VAN
VAN
  and the Internet 10
  communication 584
  EDI over 25
  evolution 9
  service provider 15
VAN communication 584
VAN connectivity
  validate
    inbound communication 447
VDA 8, 25
verify
  DB2
    Windows 62
  DB2 on AIX 110
  DB2 on Windows 62
  if map has been uploaded correctly 514, 556
  JMS installation
    Windows 61
  map has been loaded 556
  map has been uploaded 514, 556
  software prerequisites on Windows 60
  WebSphere MQ on AIX 111
  WebSphere MQ on Windows 61
verify if map has been uploaded 514, 556

Archived

VICS (Voluntary Inter-industry Communications Standards) 25
Voluntary Inter-industry Communications Standards (VICS) 9, 25

W

Web services 29
  UDDI 29
  WSDL 29
Web Services Description Language (WSDL) 29
WebSphere MQ
  configure on AIX 112
  configure on Windows 65
  queue manager and publish/subscribe broker 35
  verify on AIX 61, 111
WebSphere Partner Gateway
  architecture 32
  change password 105
  database owner 79
  features 32
  loading a map into 735
  menu options in 106
  run-time parameters of 98
  security options 41
  security options in 41
  update configuration of 366, 438
  uploading certificates in 440
WebSphere Partner Gateway Advanced
  add user accounts 111
  install database schema 114
WebSphere Partner Gateway Enterprise
  install database schema 79
WebSphere Partner Gateway Express
  add user accounts 63
  database authentication 63
  menu options 106
  run-time parameters 98
writing FTP script file 390
WSDL Web Services Description Language 29
WSDL, Web Services Description Language 29

X

xCBL 8
XML
  cross-industry standards 9
  cXML 8
  document definition 29
  ebXML 9
  for e-commerce 9
  growth of 8
  OASIS 9
  RosettaNet 8
  tag languages 29
  UN/CEFACT 9
  what is 29
  xCBL 8
XML data structures, industry-oriented 8
XML data structures, international 9
XML dictionaries 462
XML dictionary 462
XML format
  create 290, 634
XML functional area 461
XML validation map
  upload 637

Archived

Archived

Redbooks

B2B Solutions using WebSphere Partner Gateway V6.0

(1.5” spine)
1.5”<-> 1.998”
789 <->1051 pages

Archived

Archived

Archived

B2B Solutions using WebSphere Partner Gateway V6.0

Implement B2B solutions using WebSphere Partner Gateway

Explore the new and extended EDI features

Utilize the newly imbedded mapping capabilities

This IBM Redbook introduces you to business-to-business (B2B) solutions based on WebSphere Partner Gateway.

WebSphere Partner Gateway provides companies with the ability to deploy and manage an integrated B2B gateway solution, providing both traditional EDI translation and connectivity support, while also offering support for many of the newer standards such as AS1, AS2 and RosettaNet and, of course, support for legacy and proprietary data formats.

In this redbook we focus on many of the new EDI capabilities that have been introduced with this release, including:

- Enveloping and de-enveloping
- Functional acknowledgements
- FTP scripting
- Tracking of EDI interchanges (with reporting, and drill-up and drill-down functions)

We also introduce you to the imbedded transformation and mapping capabilities which are provided using the Data Interchange Services (DIS) client.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7109-00 ISBN 0738492914