IBM® WebSphere®

End-to-end Integration with IBM Sterling B2B Integration and Managed File Transfer Solutions

Implement an end-to-end integration with IBM Sterling and WebSphere Portfolios

Learn how to design a B2B solution for small and large partners

Experiment with real life scenarios

James Ballentine
Claudemir Braghirolli
Vasfi Gucer
Rahul Gupta
James B Herry
Richard Kinard
Gianluca Meloni
Bala Sivasubramanian
Eduardo Ribeiro de Souza
Frank Strecker
Gang Yin

Redbooks

ibm.com/redbooks

International Technical Support Organization

End-to-end Integration with IBM Sterling B2B Integration and Managed File Transfer Solutions

July 2012

SG24-7992-00

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

First Edition (July 2012)

This edition applies to the following products:
- IBM Sterling B2B Integrator Version 5.2.3
- IBM Sterling File Gateway Version 2.2.3
- IBM Sterling Connect:Direct Version 4.6
- IBM WebSphere Message Queue Version 7.0.1
- IBM WebSphere Message Broker Version 8.0
- IBM WebSphere Transformation Extender Design Studio Version 8.4
- IBM WebSphere Transformation Extender for Integration Servers Version 8.4
- IBM WebSphere DataPower B2B Appliance XB62 Version 4.0.2.1

Contents

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

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

CICS®
DataPower®
DB2®
IBM®
IMS™
Redbooks®
Redbooks (logo) ®
Smarter Commerce™
Tivoli®
WebSphere®
z/OS®

The following terms are trademarks of other companies:

Connect:Direct, Connect:Enterprise, Gentran:Server, Sterling Integrator, and Sterling Commerce are trademarks or registered trademarks of Sterling Commerce, Inc., an IBM Company.

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

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

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Preface

Across numerous vertical industries, enterprises are challenged to improve processing efficiency as transactions flow from their business communities to their internal systems and vice versa, simplify management and expansion of the external communities, accommodate customer and supplier preferences, govern the flow of information, enforce policy and standards, and protect sensitive information. Throughout this process, external partners must be on-boarded and off-boarded, information must flow across multiple communications infrastructures, and data must be mapped and transformed for consumption across multiple applications.

Some transactions require synchronous or real-time processing while others are of a more periodic nature. For some classes of customer or supplier, the enterprise might prefer a locally-managed, on-premise solution. For some types of communities (often small businesses), an as-a-Service solution might be the best option. Many large enterprises combine the on-premise and as-a-Service approach to serve different categories of business partners (customers or suppliers.

This IBM® Redbooks® publication focuses on solutions for end-to-end integration in complex value chains and presents several end-to-end common integration scenarios with IBM Sterling and IBM WebSphere® portfolios.

It is not the goal of this publication to provide deep technical detail for every product, or to provide the singular best solution for any specific scenario, but rather to provide insight on how integration using the products can be accomplished.

We first define a high level integration solution architecture and provide several examples of how this architecture can be applied to different industries.

We believe that this publication will be a reference for IT Specialists and IT Architects implementing an integration solution architecture involving IBM Sterling and IBM WebSphere portfolios.

The team who wrote this book

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

James Ballentine is an Advisory Software Engineer for the IBM Sterling B2B Integrator Support Team, part of B2B Commerce Solutions of the IBM Industry Solutions group. He has worked as a support professional for IBM Sterling B2B Integrator for the past 7 years, 5 of which were spent as the team lead for the Database support team. Prior to joining this team, he had worked for nearly 10 years as a database administrator, designer, and developer in a variety of positions. He has extensive experience with RDBMS technologies, as well as various programming languages and associated technologies.

Claudemir Braghirolli works at Customer Support, IBM Software Group, Commerce based in São Paulo, Brazil, his native town. He has 19 years of experience with the products of IBM Sterling Connect:Direct family and 7 years of experience with IBM Sterling B2B Integrator. His areas of expertise include implementation and support of IBM Sterling Connect:Direct family products, support of IBM Sterling B2B Integrator, and development of applications under IBM Sterling B2B Integrator.

Vasfi Gucer is a Project Leader at the IBM International Technical Support Organization. He has been with the ITSO since January 1999. He has more than 12 years of experience in the areas of systems management, networking hardware, and software on mainframe and distributed platforms. He writes extensively and teaches IBM classes worldwide on IBM products. Vasfi is also an IBM Certified Senior IT Specialist, PMP, and ITIL Expert.

Rahul Gupta is an Advisory IT Specialist with IBM Global Technology Services in India. He is a Certified SOA Architect with 6 years of professional experience in IBM messaging technologies. At his current assignment, he works as a middleware consultant for various clients in North America. His core experiences are in lab testing, performance tuning, and Level 3 development for both WebSphere Message Broker and WebSphere MQ products. Rahul has been a technical speaker for messaging-related topics at various WebSphere conferences and is a recognized inventor by the IBM innovation community.

James B Herry is a Senior Managing Consultant, currently with IBM Software Services - Industry Solutions. He joined IBM in 2006 after working for 16 years as an application development professional, specializing in middleware integration. James graduated from Elon College in 1990 with a Bachelor's degree in Computer Science. Since joining IBM, he has specialized in helping clients in the financial and health care industries integrate WebSphere Transformation Extender into their middleware solutions, both as a solution architect and as an integration consultant.

Richard Kinard is the Product Manager for WebSphere DataPower Appliances. He is a subject matter expert in business-to-business (B2B) technologies and has over 12 years of experience designing, developing, and implementing B2B solutions. He has worked on many initiatives with Internet standards organizations to promote B2B interoperability and was a Senior Product Manager of a successful B2B application prior to working for IBM.

Gianluca Meloni is a Technical Support Analyst for IBM Industry Solutions, B2B Commerce, based in Italy. He has 11 years of experience in the B2B integration field. He holds a degree in mathematics from the University of Cagliari. His areas of expertise include IBM Sterling B2B Integrator, IBM Sterling File Gateway, IBM Sterling Connect:Direct and IBM Sterling Gentran:Server for UNIX. Originally from Sardinia, he is now home-based in Sanremo with hub office in Milan.

Bala Sivasubramanian is an IBM Certified Senior IT Specialist for IBM US SWG - Industry Solution Service, which focuses on Telecom, Media, and Entertainment. Previously, he worked in the IBM US GBS Wireless Practice as Senior IT Specialist and Integration Architect for the Tier 1 Service Providers in North America and has wide, international experience in the real world Telecom projects that involve complex integration solutions. His expertise includes a broad focus on the telecommunications industry. He has a deep knowledge of IBM tooling / product suite / Industry frameworks, IBM Telecom SDP & SPDE Framework, Parlay & TMF Standards, and OSS/BSS vendor product integration with IBM Web Sphere Middleware Products. He is the co-author for IBM Redbooks publications on the Telecommunication Industry Solution and is co-author for the IBM Certifications on the IBM Middleware and Tooling Products. He is also an IBM Certified Senior IT Specialist - Application Integration and Middleware and has 8+ IBM Product Certifications. Additionally, he holds a degree in Computer Science Engineering.

Eduardo Ribeiro de Souza is a Senior Solution Engineer and an architect to integration solutions such as Enterprise Application Integration (EAI), Business to Business Integration (B2Bi), Business Process Management (BPM), Business Activity Monitoring (BAM), and composite applications. He has more than 15 years of experience in developing and integrating enterprise applications across a variety of industries. He holds a Master's degree in Production Engineering from Federal University of Santa Catarina, Brazil.

Frank Strecker is an Advisory IT Architect and Support Engineer in the European IBM Sterling B2B Integrator support team. Before joining IBM Industry Solutions support, he used to work as a WebSphere MQ and Message Broker Consultant in the German IBM Software Services for WebSphere team. After working hours, Frank teaches the Alexander Technique.

Gang Yin is a Product Developer in Sterling B2B China team, focusing on development and L3 support of Sterling B2B cloud services. His areas of expertise include WebSphere ESB, WebSphere Message Broker, and WebSphere DataPower appliances.

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

Dave Bennin, Shari Deiana, Emma Jacobs, Stephen Smith
International Technical Support Organization, Raleigh Center

Bobbi Burris, Rex Chan, Daniel Davis, Diane Defallo, Fritz Fahrenback, Charles Fenton, Bill Hines, Michael Hudson, Lorenz Jimenez, Scott Lewis, Dirk Maney, Jeff Matteo, George Middlebrook, Jeff Leach, Todd Margo, Mark Roper, Pamela Taylor
IBM USA

Ken Clark, David Hardcastle
IBM UK

Andrew Pang
IBM Canada

The team also would like to thank Bandana Batra for providing valuable input on the book content.

Now you can become a published author, too!

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

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

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

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

- Stay current on recent Redbooks publications with RSS Feeds:

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

Part 1

Introducing the proposed solution architecture

In this part of the book, we introduce our proposed solution architecture for implementing end-to-end integration in complex value chains. In Chapter 1, “Introduction” on page 3, we describe this architecture at a high level. In the rest of the chapters in this part, we delve into details of the components of this proposed solution architecture.

1

Introduction

In today's business and technology environment, it is likely that you deploy products from many different technology portfolios to meet the needs of your organization. It is also likely that you solve a multitude of business problems by making use of a multitude of technology offerings. You have probably thought about how you might best use the capabilities of the various products you have in place to more effectively integrate your business, while still leveraging the investment in the effort and technology you have already made. This IBM® Redbooks® publication addresses this concern, as it relates to products in the IBM Sterling and products in the IBM WebSphere® portfolios.

This publication provides a guide to end-to-end integration for organizations with these needs:

- Already have solutions deployed using both sets of products and want to more securely and seamlessly integrate them together.
- Want to extend the capabilities of an existing solution that makes use of one of the aforementioned product portfolios by integrating it with the other.
- Have inherited technology solutions through a merger or an acquisition and want to integrate the inherited capabilities with the existing functionality.

The products used for the scenarios described in this publication are not all-inclusive, but they provide you with ideas for how you can build basic integration between the products. The scenarios are based on real-world situations and give examples of what is involved with integrating the products in a live deployment. As is the case with building and deploying any solution, it is important to understand the requirements of the business and the capabilities of the available technology in order to provide the most cost effective and robust solution required to meet the goals of the business.

It is not the goal of this publication to provide deep technical detail for every product or to provide the singular best solution for any specific scenario, but rather to provide insight on how integration using the products can be accomplished.

This chapter introduces the content of this publication, the products that we use to demonstrate the capabilities of the software suites, the real world scenarios that we chose to demonstrate implementation of integration, and some alternative choices that you might want to make in your organization.

This chapter includes the following sections:

1.1 Scope

Because of the numerous possible points of integration between the products in the IBM Sterling and the IBM WebSphere portfolios, the scope of this publication excludes demonstrating integration between every possible product combination. Instead we provide examples, from which you can extend the concepts of integration to other product combinations.

We assume that you are knowledgeable regarding the products discussed in this publication. Therefore, installation, base configuration, and detailed instruction on the basic use of the products discussed in the scenarios are not provided. However, we do provide references to enable you to gain additional knowledge and skill for each of the products we discuss.

1.2 How to use this book

This publication is divided into two parts:

- Part 1, "Introducing the proposed solution architecture" on page 1 describes the conceptual architecture and how to implement each of the high level components of the architecture using products from IBM Sterling and IBM WebSphere portfolios. Each component is discussed as to its purpose within the functional area of the architecture in which it resides.

  In addition to the products actually used to implement the various scenarios appearing in this book, products that might be candidates to serve the same purpose are referenced. The products used in the scenarios discussed in this publication to demonstrate the integration capabilities are introduced in 1.8, "Lab environment" on page 10.
- Part 2, "Scenarios based on the proposed solution architecture" on page 101 describes four scenarios that demonstrate ways to integrate IBM Sterling and the IBM WebSphere portfolio products. Each scenario implementation is discussed. Each implementation is executed using a different combination of products and/or product capabilities from the available products in the IBM Sterling and WebSphere portfolios.

1.3 Products used

- IBM Sterling B2B Integrator V5.2.3:

  A transaction engine and toolkit that allows the user to define, create, implement, and manage process flows. These flows, in turn, allow for the processing, routing, translation, and storage of high volumes of inbound and outbound messages or files, and can also interact and integrate with both internal systems and external business partners. For more information, including the details of the supported protocols, see "Sterling B2B Integrator" on page 21. You can also see the IBM Sterling B2B Integrator 5.2 Information Center at:

  http://publib.boulder.ibm.com/infocenter/sb2bi/v5r2/index.jsp
- IBM Sterling File Gateway V2.2.3:

  An application that integrates with, and extends, the ability of IBM Sterling B2B Integrator to accept, manage, route, and process files from external partners, using multiple different protocols and formats. It can simplify the management of external partners and communities and can also assist in seamless integration of internal and external systems. For more information, see "IBM Sterling File Gateway" on page 58. Another source of information is IBM Sterling File Gateway 2.0 Information Center which can be found at:

  http://publib.boulder.ibm.com/infocenter/sfg/v2r0/index.jsp

- IBM Sterling Connect:Direct® V4.6:

  A peer-to-peer file-based integration middleware which provides secure delivery of high-volume data within and between enterprises. It is optimized for high performance file transfer of any type of data (text, EDI, binary, digital content, image) across multiple platforms, several file systems, and many different media. A large number of industries throughout the world use it to move large volumes of data and for connecting to remote offices. For more information, see “IBM Sterling Connect:Direct” on page 62 or *IBM Sterling Connect:Direct Product Overview* manual at:

  http://www.sterlingcommerce.com/documentation/home/MFT/ConnectDirect/Common/CDOverview.pdf

- IBM Sterling Secure Proxy V3.4:

  An application proxy between IBM Connect:Direct nodes or between a client application and an IBM Sterling B2B Integrator server. It provides a high level of data protection between external connections and your internal network. It defines an inbound node definition for each trading partner connection from outside the company and an outbound node definition for every company server to which SSP will connect. It supports the following protocols: C:D, PEsit, FTP, SFTP (SSH-FTP), FTPS (FTP over SSL/TLS), HTTP, HTTPS. For more information, see “Sterling Secure Proxy” on page 24, or see the following link:

  http://publib.boulder.ibm.com/infocenter/ssp/v3r4/index.jsp

- IBM Sterling Control Center V5.3:

  Provides centralized management, monitoring, and notification for IBM Sterling products. For more information, see “IBM Sterling Control Center” on page 67. You can also see the following link:

  http://publib.boulder.ibm.com/infocenter/sb2bi/v5r2/nav/6_5

- IBM WebSphere MQ V7.0.1:

  Messaging backbone that provides powerful and reliable connectivity and assured delivery of messages throughout the enterprise. For more information, see “WebSphere MQ” on page 30, or see:

  http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp

- IBM WebSphere Message Broker V8.0:

  High volume integration hub that delivers advanced enterprise service bus capabilities on the WebSphere MQ backbone. For more information, see “WebSphere Message Broker” on page 74, or see:

  http://publib.boulder.ibm.com/infocenter/wmbhelp/v8r0m0/index.jsp

- IBM WebSphere Transformation Extender Design Studio V8.4:

  Used to develop the maps used to translate and transform data to the desired formats. For more information, see “WebSphere Transformation Extender Design Studio” on page 81, or see:

  http://publib.boulder.ibm.com/infocenter/wtxdoc/v8r4m0/index.jsp

- IBM WebSphere Transformation Extender for Integration Servers V8.4:

  Allows execution of maps on a variety of hardware platforms by many software products. For more information, see “WebSphere Transformation Extender for Integration Servers” on page 87, or see:

  http://publib.boulder.ibm.com/infocenter/wtxdoc/v8r4m0/index.jsp

- IBM WebSphere Transformation Extender Industry Packs:

  Prebuilt industry specific artifacts providing out of the box functionality and used to accelerate map development. The packs used in the book scenarios are:

  – EDI pack V2.8.0
  – Healthcare pack V4.4.0

  For more information, see “WebSphere Transformation Extender Industry Packs” on page 87, or see:

  http://publib.boulder.ibm.com/infocenter/wtxdoc/v8r4m0/index.jsp

- IBM WebSphere DataPower® B2B Appliance XB62 v4.0.2.1:

  The XB62 is a B2B enabled security gateway that makes the services of one organization available to others in a controlled and secured manner providing capabilities such as connection security, B2B Messaging, non-repudiation and partner profile management. Additionally, it utilizes the integration services needed to make it easy to integrate to internal systems or directly to applications. These services built on-top of the DataPower appliance form factor adds integrated network functions decreasing the complexity and deployment footprint of the solution. For more information, see “WebSphere DataPower XB62 appliance” on page 25, or see:

  http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.wdatapower/wdatapower/1.0/xb62.html

- IBM Sterling File Transfer Service:

  Allows customers to manage a single, secure, and reliable connection to reach their business partners without the capital expense associated with on-premise software or the operational impact on IT staff. For more information, see “IBM Sterling File Transfer Service” on page 65, or see:

  http://publib.boulder.ibm.com/infocenter/sb2bsvcs/v1r0/nav/13

- IBM Sterling B2B Collaboration Network:

  Acts as the communications link between customers and their business partners, and offers a growing list of on-demand application services. For more information, see “IBM Sterling B2B Collaboration Network” on page 66, or see:

  http://publib.boulder.ibm.com/infocenter/sb2bsvcs/v1r0/nav/0

- IBM Sterling Web Forms:

  A customizable website operated by IBM that helps customers to integrate smaller trading partners who do not have EDI capabilities. For more information, see “IBM Sterling Web Forms” on page 67, or see:

  http://publib.boulder.ibm.com/infocenter/sb2bsvcs/v1r0/nav/9

1.4 Intended audience

This publication is intended for anyone who wants to gain insight into integrating solutions that use IBM Sterling and the IBM WebSphere portfolios.

For example:

- You might be interested in building a new solution to meet the needs of your organization and want to make use of the most powerful features of both the IBM Sterling and the IBM WebSphere portfolios.

- You might be involved in an acquisition or merger situation, or looking to integrate the capabilities across solutions that already exist.
- You might want to eliminate redundancy of solutions across your company by integrating the best of the solutions that have already been built within different departments within your company, and these solutions make use of both technology portfolios.

If any of these situations describes your expectations, or if you are just interested in the capabilities of one or the other of these technology suites, and already use the other, you can benefit from this publication.

1.5 Conceptual architecture

Figure 1-1 shows the conceptual architecture (also referred as the solution architecture throughout the book) used to define each of the scenarios in this publication.

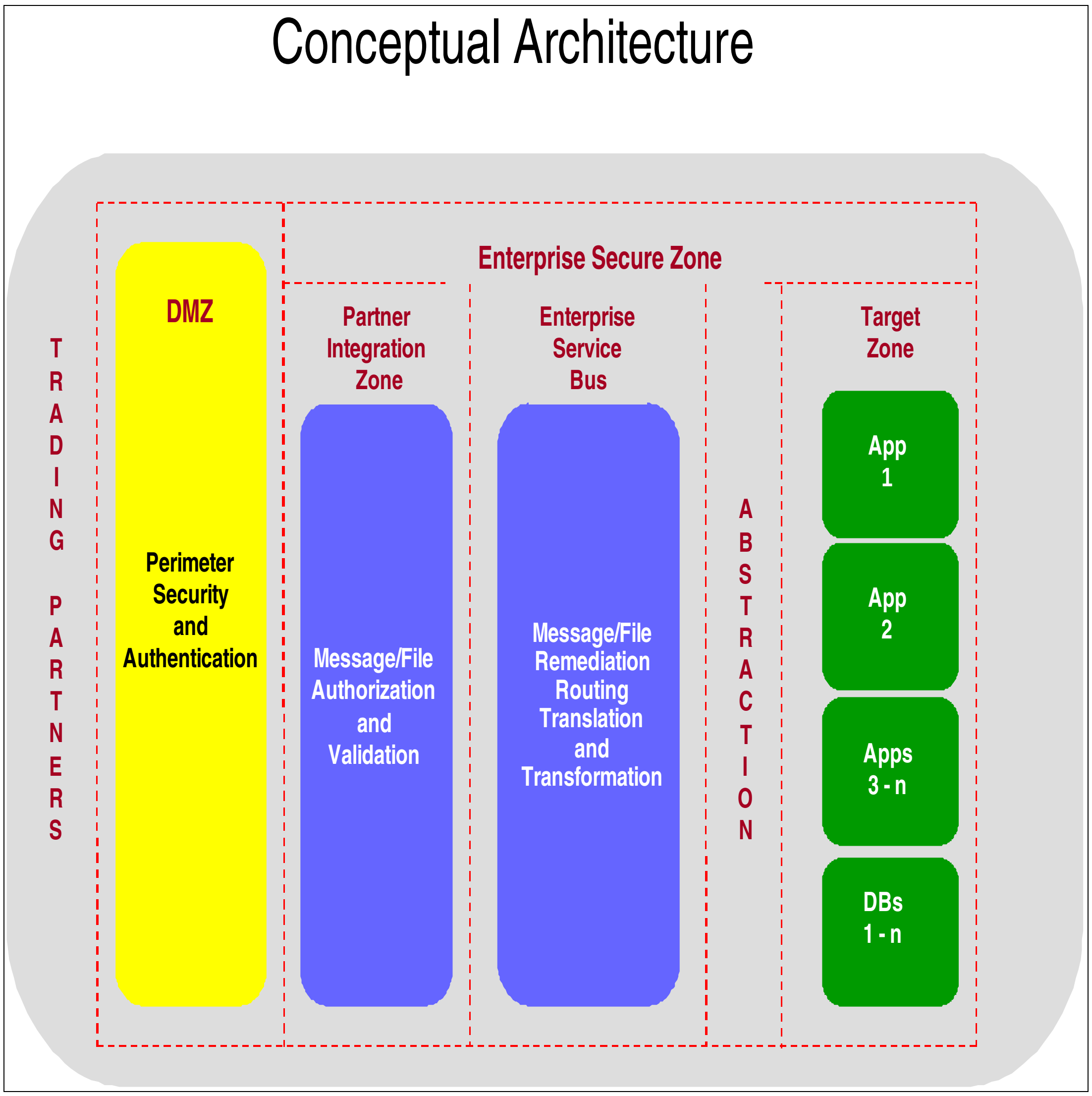


Figure 1-1 Conceptual architecture

The chosen architecture enforces message security into the organization through the DMZ. Then, inside the organization, trading partner management occurs in the Partner Integration Zone. In this zone, the solution validates the file or message type against the trading partner to determine if the trading partner is authorized to send this type of information, performs validation of format, and responds with the proper acknowledgement.

After the payload is determined to be an authorized file or message, it is handed off to the Enterprise Service Bus to be transformed and routed to the targeted destination. Details of what occurs in each functional area of the conceptual architecture are discussed in Part 1, “Introducing the proposed solution architecture” on page 1 of this book.

DataPower XB62: This architecture is slightly different when using the DataPower XB62 Appliance. In this scenario, the DataPower XB62 Appliance sits in the DMZ and moves B2B Security out to the edge of the network and rejects unwanted connections and partners before they get into the secured network. So when using DataPower XB62, partner management happens in the DMZ, eliminating the need for the Partner Integration Zone.

1.6 Business benefits

The solution architecture shown in Figure 1-1 streamlines end-to-end business processes through the ability to automate the complete process. “Real world” business processes are typically “end-to-end,” involving both internal employees and systems and external people and systems. The ability to streamline end-to-end business processes reduces operational costs through automation, improves process quality (again through automation and repeatable process models), and offers the possibility of strategic business transformation. In a sense, the solution architecture is a foundation enabler for IBM Smarter Commerce™.

The solution architecture helps organizations gain a reputation of being “easy to do business with” through a flexible business integration architecture. The IBM solution architecture enables external entities to interact with a “hub” organization in virtually any style of integration, whether file-based, messaged-based, or service-based. The solution architecture supports direct partner interactions through software or mediated interactions through IBM B2B Services. This enables a hub organization to optimize its value chain and be more agile. It can respond more quickly to changing business conditions and is more resilient to supply chain disruptions.

The flexibility inherent in the solution architecture also facilitates business services innovation. It enables organizations to accelerate the deployment of new business services, exposed through MFT interfaces, traditional B2B document-oriented interfaces, or more progressive real-time Web Services. It can have widespread business benefits through:

- Faster time-to-value for new electronic business services.
- Expanding the accessible market for a service globally.
- Enabling an organization to target a different size customer segment, for example, ability (perhaps) to target the SMB world for the first time.

The solution architecture, because its a general framework that can be deployed as a corporate standard, enables reduced complexity and enhanced IT agility. It offers a defined framework for external entity interaction, enabling the strategic concept of “single view of partner” (external entity), thus reducing perceived external interface complexity for the hub organization and increasing the quality of external entity interactions.

Large enterprises typically have multiple systems (perceived as multiple “points of entry” by the external entity) used to interact with external entities. The solution architecture enables B2B architecture consolidation. It is a strategic IT cost containment initiative but also a strategic business quality improvement initiative.

The solution architecture helps organizations conform to modern information security mandates, which are increasingly stringent and vary geographically, by industry and even by process. The benefit here is to be able to satisfy the requirements of internal security audits as well as security expectations of external entities and governing bodies. The solution implements a robust “edge security” capability, which facilitates both secure and scalable B2B interactions, and which provides important value-add DMZ capabilities beyond traditional “proxy” servers.

Because the solution architecture offers a strategic approach to end-to-end integration, it thus provides a sound go-forward foundation for Smarter Commerce: the classic “invest strategically, implement tactically” kind of concept. The solution architecture is extensible. It can be widely applied and widely used to leverage existing infrastructure. The solution architecture facilitates both batch and real-time oriented B2B integration. As such, it enables the organization to optimize their approach to evolving traditional batch oriented processes to near real-time interactions. Thus, it facilitates the evolution toward the true digital business.

1.7 Overview of scenarios

The scenarios cover a subset of real world deployments that can be implemented using products from IBM Sterling and IBM WebSphere portfolios. The solutions to the scenarios are based on our experience deploying our products in the industries covered in the scenarios, and are based on the expertise gained using the products selected. The scenarios demonstrate how to integrate between the products we use to solve a specific business need. These configurations fulfill business requirements, IT constraints, and other external factors of each specific scenario.

Keep in mind that the configurations used in the scenarios are not your only option. We chose the implementation details we used to demonstrate how to integrate these specific products, yet still meet the requirements of the business. We also provide information about alternative configurations, that also meet the requirements of the various scenarios. The scenarios we chose to implement in this publication are based on use cases from the following industries:

- Health insurance
- Financial
- Supply Chain

We also chose to show scenarios making use of cloud services.

1.8 Lab environment

This section describes the lab environment used to implement the scenarios in Part 2 of this book. Host names as well as the operating system and release level or firmware release level are given for each host. The software stack installed on each host is also provided.

All Windows systems have the following configuration:

- Lenovo M57p (machine type 9196)
- Intel Core 2 Duo processor
- 8G RAM
- Gigabit ethernet
- Windows 2008 or Windows 7 64 bit

See Table 1-1 for details.

Table 1-1 Lab environment

Host name	**Operating system/ firmware level**	**Installed products/appliances**	**Scenario where used**
esb01.itso.ibm.com	Windows 7	IBM WebSphere MQ V7.0.1 IBM WebSphere Message Broker V8.0 IBM WebSphere Transformation Extender V8.4	Chapter 7, "Supply Chain scenario using AS2 and EDI" on page 261
esb02.itso.ibm.com	Windows 7	IBM WebSphere MQ V7.0.1 IBM WebSphere Message Broker V8.0 IBM WebSphere Transformation Extender V8.4	Chapter 5, "Health Insurance scenario" on page 103
cc01.itso.ibm.com	Windows 7	IBM WebSphere Transformation Extender V8.4 IBM Sterling Connect Direct V4.6 IBM Sterling Control Center V5.3	Chapter 6, "Financial Services scenario" on page 177
ssp01.itso.ibm.com	Windows 7	IBM Sterling Secure Proxy V3.4	Chapter 5, "Health Insurance scenario" on page 103
b2b01.itso.ibm.com	Windows 2008	IBM WebSphere MQ V7.0.1 IBM Sterling B2B Integrator V5.2.3 IBM Sterling File Gateway V2.2.3 IBM Transformation Extender V8.4 (connected to the Internet)	Chapter 5, "Health Insurance scenario" on page 103
b2b02.itso.ibm.com	Windows 2008	IBM WebSphere MQ V7.0.1 IBM WebSphere Message Broker V8.0 IBM WebSphere Transformation Extender V8.4 IBM Sterling Connect Direct V4.6 IBM Sterling B2B Integrator V5.2.3 IBM Sterling File Gateway V2.2.3	Chapter 6, "Financial Services scenario" on page 177
DataPower 9005/7199 Appliance	DataPower Firmware V4.0.2.1	IBM WebSphere DataPower B2B Appliance XB62	Chapter 7, "Supply Chain scenario using AS2 and EDI" on page 261

2

Enabling seamless and secure integration inside and outside of the enterprise

This chapter discusses integration inside and outside of the enterprise: the cooperation with external partners as customers and suppliers and the internal cooperation across the enterprise. Because internal and external integration require different approaches, we discuss both kinds of integration in distinct sections:

- External integration, which must be as secure as needed
- Internal integration, which must be as seamless as helpful

Starting from the conceptual architecture exposed in 1.5, "Conceptual architecture" on page 8, we discuss how the enterprise can establish safe connections to the outer world through a well equipped perimeter network. We do not discuss the construction of a DMZ in depth. But we do discuss the equipment needed there to establish safe partner operations through protocol termination and the implementation of the "Don't call us, we'll call you" principle between trusted zone and perimeter.

In this chapter, we show how gateways can help to isolate the internals from the partner communication. We show how good architecture and implementation guidelines can help to prohibit the proliferation of partner specific details into the entire enterprise. We also discuss the integration of the entire enterprise through hub or bus structures.

A preliminary section is included to help you consider general requirements and design forces in terms of IT integration and B2B. You can skip this section if you are familiar with the field, or you can read it to enjoy the high level discussion of principles based on a model of vintage integration by punchcards.

This chapter includes the following sections:

2.1 What are the business requirements?

What a business requires exactly in terms of internal and external integration is always based on the kind of business being done. Requirements depend (among others) on the following factors:

- Size of the enterprise
- Tools and skills available
- Tactical or strategic attitude

We consider some architectural aspects, such as extensibility, scalability, changeability, and resilience of designs, before we discuss the technical instrumentation of the design with particular products. We discuss these aspects and others, starting from a simple model, that helps to recapture basics and understand issues still relevant in today's integration concerns.

2.1.1 Basic concepts and design forces: A gentle introduction

Imagine a model of vintage integration, as shown in Figure 2-1, as it used to be common in the Seventies or Eighties: two cooperating enterprises share information about catalog data. Enterprise A, a wholesaler, sends updates to stock item catalog to Enterprise B (which is buying from A) overnight by exchanging batches of punchcards or tapes. Inside both of the enterprises, the operators simply used single sets of data in shared databases.

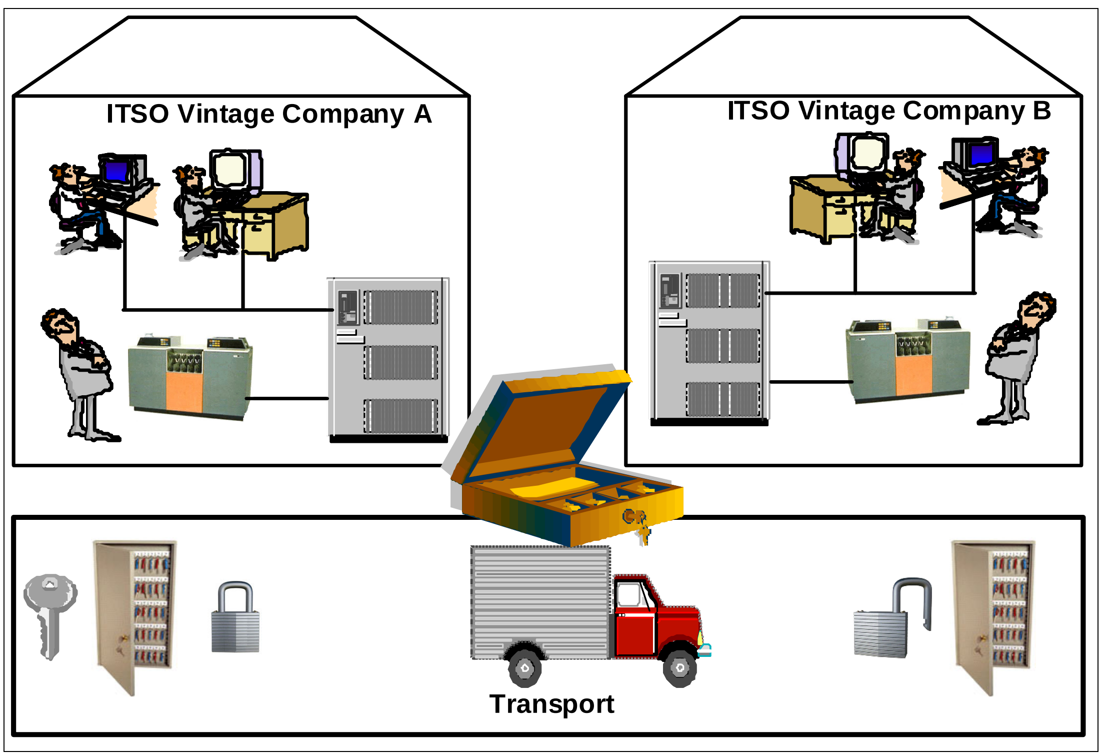


Figure 2-1 Introductory sample of vintage internal and external integration

The vintage integration model

The vintage model shows the systems connected by using the following process:

1. The sender produces punchcards by encoding the content of their system according to the code and protocol agreed to by the receiving end.

2. The sender packs and seals punchcards into a labeled and locked container and passes the container to the intermediary; for example, a courier.
3. The intermediary courier transports the container and delivers to the receiver as labelled.
4. The receiver unlocks and unseals container, and unpacks the cards from the container.
5. The receiver reads the cards by translating the cards into their systems.

While the basic concept is still the same, the media has changed. The content is now encoded, locked into containers for transport, transported, received, unlocked, decoded, and read.

Choosing encoding and protocol: Standards, skills, tools

In the early days, enterprises had to negotiate the formats and proprietary protocols they wanted to use by themselves. It proved to be costly, too expansive for single enterprises, and time consuming. As a result, standards emerged for both inside and outside communication. In 1975, the Transportation Data Committee in the American National Standards Institute (ANSI) published a first standard that led to ANSI ASC X12 EDI in 1979. See the following website for details:

http://www.x12.org/x12org/about/X12History.cfm

Since the EDI standards have been in place for more than thirty years now, we have a lot of standards today. Standards from all the four major eras of evolution in B2B communication now occur in the field. See Chapter 7 in the PDF file at the following website for details:

http://www.redbooks.ibm.com/abstracts/sg247745.html

While many standards began as reference models for individual programming in the enterprises, it proved to be too expansive for single enterprises to catch up with the development of these standards. Enterprises need to catch up with new emerging standards and have to keep old standards alive as long as there is a single partner or internal system left that has to use it.

Thus, most enterprises make use of tools that help them to do this task. The toolmaking company ensures that they keep pace with the standards, while the user can spend more attention on their business.

You might notice that the same pattern that occurred on the level of individual protocols and on the level of standards might also become relevant for tools. If too many different tools are used, it might get hard for an enterprise to have all of the skills at hand when needed.

Security and non-repudiation

Punch cards or tapes used to be packed and sealed, and put into steel-boxes for transport. After transport, the receiver first checked the seal of the box, and then opened it. The cards received did not go directly to the card reader but were reviewed to ensure that they contained the correct category of data; for instance, only data records and no programs. After reading, the punch cards were kept as proof for later audits.

We rarely use physical media today, but communicate online. So the seals, locks, and audits must provide a virtual interpretation of the transport:

- Encryption and decryption of online content locks it from being read by unauthorized people.
- Digital signatures prove authenticity, that the content is not being changed or counterfeit, such as a physical seal.
- Storing the entire communication with signature serves as proof.

Because these security measures add a certain overhead to the communication, they must only be used when necessary, but never forgotten when needed. In many situations, it is required by law.

Layers

As an analogy, a courier driver would bring the data to the train station and hand it to the train chief for express expedition. None of the intermediary transporters had to care about the contents of the box. Their only task was to bring it to the labelled target address. Further, if the train car was shipped by ferry, the seamen would not care about the contents of the train, only about getting it to the other side.

Today, we might see situations where six or more protocols layers are wrapped over each other, much like a set of Russian stacking dolls. Each level needs appropriate tools to encode, decode, and wrap according to the rules expected. A familiar example would be a SOAP-Web service over HTTPS over TCP/IP, or sending files as MQ messages.

To the outside of the enterprise, the choice of transport depends on how your external trading partners are reached, and what both partners are willing to invest into the partnership. The more flexible you are in your IT connectivity, the more choices your business has to choose partners for reasons other than only being in reach.

Inside the enterprise, the decision is up to the enterprise itself. Standardization would be a good approach, because the more ways you use to connect, the higher the costs are. The fewer different networks and protocols you maintain internally, the cheaper it is for you to do the work.

Reliability and transaction security

In the vintage model, the operators at the sending end notified the receiving about the number of punchcards or records to be transferred. The receiving end checked the card reader for completeness of the transaction. The operators took care that all cards were read “exactly once, and only once”. If necessary, they repunched damaged cards or rejected the whole batch.

When we consider current integration technology, we ask for the degree of reliability and transactional safety the technology offers. Because this reliability has its costs in terms of latency and resource consumption, we have to decide where a high degree of reliability is necessary and, conversely, where a low degree can be used when a simpler solution is sufficient.

Timeliness of delivery

The partners in the vintage integration model synchronized their data every night. This form of cooperation still exists today as a nightly exchange of large files, suitable for instance to bootstrap master file data. The masterfile data is characterized by the fact that changes are rare. The enterprises often define the time when master data can change.

File based integration is well known and straighforward to understand, but not the fastest option. Reliability can be a problem, when unmanaged file based protocols such as FTP or SFTP are used. Protocols with less overhead, such as remote procedure call or messaging protocols, are more suitable.

Today there are many cases where it is vital to keep the synchronization gap as short as possible; for example, when propagating stock exchange data. In these times of online brokerage, getting the stock rate from last night is not helpful. It is the type of data that needs the shortest synchronization gap possible to get the latest rate.

The following questions need to be asked:

- How often must the systems communicate?
- How large are the chunks of information exchanged?
- Are users or systems waiting until their partner system has processed the request?

Concerning timeliness, we need to distinguish these parameters:

- Real time
- Near real-time
- Bulk and batch
- Store and forward
- Resend

Loose coupling: The need to separate

In the vintage model, the companies used single monolithic systems. Inside the enterprise, all users were working on the same database for all purposes. With growing requirements, this eventually led to severe problems and the systems became unmanageable. After recognizing that issues emerged from too much concentration of logic and data, as happens so often in history, just the opposite effect resulted; logic and data were distributed all over the enterprise. Today, we can learn from both eras, that too much or too little integration can both cause trouble.

When discussing integration, it is important to keep in mind cooperation of distinct systems in a way that they cooperate seamlessly only in a certain aspect, while staying as independent as possible. These smaller systems are more resilient to change and easier to maintain than the larger systems, because decoupling increases the flexibility. Decoupling is a major system design consideration to be kept in mind, particularly for internal communication.

Shared databases guarantee timeliness, but are problematic in terms of security and tight coupling systems through the use of a common data definition. *Remote procedure* call is also problematic, unless the coupling is loosened by a intermediate layers, as with Web services.

The coupling is always tight when systems communicate directly with each other, because they need to understand a common data format then. Intermediates such as integration hubs and brokers help to connect systems in a way that they remain loosely coupled in that sense. Loose coupling is most relevant when the speed of change in an enterprise is very high.

2.1.2 Conclusion

The basic principles might have changed less then expected, but there are some major differences today compared to the beginnings of electronic data interchange:

- Agility
- Scale
- Exposure

Today, we need to be very flexible and very fast, and we have to address integration tasks that are far beyond what IT specialists in the past might have imagined. And, with the advent of Internet connectivity, we are much more exposed to the risk of attacks.

2.2 Secure cooperation with business partners

The use of the term *cooperation* instead of *integration* emphasizes the obvious difference between external and internal communication. Between enterprises, there must be well defined borderlines to be compliant with both company's security requirements as well as with any legal mandates that each company must follow.

Security in B2B cooperation has the following meanings:

- *Data security*: Data is kept safe from being read or changed by unauthorized accesses.
- A*uthentication of partners and messages*: Provide access only to designated parties.
- *Authorization*: Grant access to only the resources allowed to be used by the partner.
- *Non repudiation*: Provide proof of data origination and receipt.
- *Threat protection*: Protect against the many external attacks that communicating over a public network can present; for example, Denial of Service, malicious content, and so on.

As described in 1.5, "Conceptual architecture" on page 8, we suggest connecting to partners through a combination of technical security and functional isolation:

- The technical security function of a perimeter network (DMZ) compares to the function of a guarded door. Those who want to enter can present their entry credentials, the doorman screens them for obvious security threats, but they do not do their business with the doorman.
- The gateway function is located at a place comparable to a counter behind the guarded door, where the actual business exchange takes place. Besides the security aspects that it obviously has, it separates the concerns and spheres of the business partners. Externals are neither entitled nor capable of interacting directly with back office functions. At the counter, they meet front office representatives, who know the external business as well as the internal and mediates for the back offices.

Products such as IBM WebSphere DataPower can provide both the perimeter security and gateway functions in a single device. It is due to the nature of an appliance form factor; most software solutions must distribute each function to remain secure.

2.2.1 Technical security function

In information security, the demilitarized zone (DMZ) or network perimeter is a small network situated between a trusted network and an untrusted one, usually between the LAN and the Internet. The function of a DMZ is to keep all services that have external access (such as HTTP, FTP, mail, and so on) separated from the local network, thus limiting the potential damage in case of compromise of any of these services by an attacker. To achieve this goal, the computers present in a DMZ must not contain any form of local network access.

This configuration is accomplished through the use of equipment in the firewall, which make the control of access between the local network, the Internet, and the DMZ (or in one generic model, between the two networks to be separated and the DMZ). The equipment in the DMZ can be a dedicated switch or a shared network switch, but in the latter case, it must be set up in separate virtual networks (VLANs) within the equipment.

DMZ systems

The system inside of the DMZ has the ability to connect to external partners using a wide range of standard and non-standard connection protocols. The protocols used can vary, depending on the technologies used in DMZ. Those systems in the DMZ are used as intermediaries for requests from clients. They act as proxies forwarding the requests or files to and from servers inside of the protected network zone. In many cases, they terminate the external connection and perform bridging between protocols on the front-side to protocol on the back-side.

The DMZ is also called a *perimeter* network. Like a perimeter, it allows you to be in contact with the outside, without being seen as a whole. You can also compare the perimeter zone to the manual process in the vintage model, as discussed in "Security and non-repudiation" on page 15. Any communication, such as incoming cards, are reviewed before they are read to the card reader and are allowed to change data in the core.

Requirements for DMZ systems

Systems exposed in the DMZ typically include these requirements:

- No files ought to be at rest in the DMZ unless they are encrypted.
- If connections are needed from systems in the DMZ to systems in the secured network, lock the inner firewall to only allow data to be passed from the DMZ system's IP and Port.
- Isolate internal and external connections on separate Ethernet adapters.
- Use IP filtering at the outer firewall to only accept data from known IP addresses.
- Perform session breaks and protocol termination in the DMZ.
- Key management; keep public keys of partners and your own private keys secure.

Security: IBM Sterling has the ability to provide added security by using software that limits the need to open holes through the inner firewall. Sterling Perimeter Services establishes outward-in connectively from within the secured network to the DMZ.

2.2.2 Gateway function

The gateway does what a front desk does in a shop organization. The gateway function is enabling the isolation of partner data from its own data. It assures a single view to the partner from the inside, much as the enterprise service bus assures a single view to the inside for partners and internal participants. Figure 2-2 shows this function.

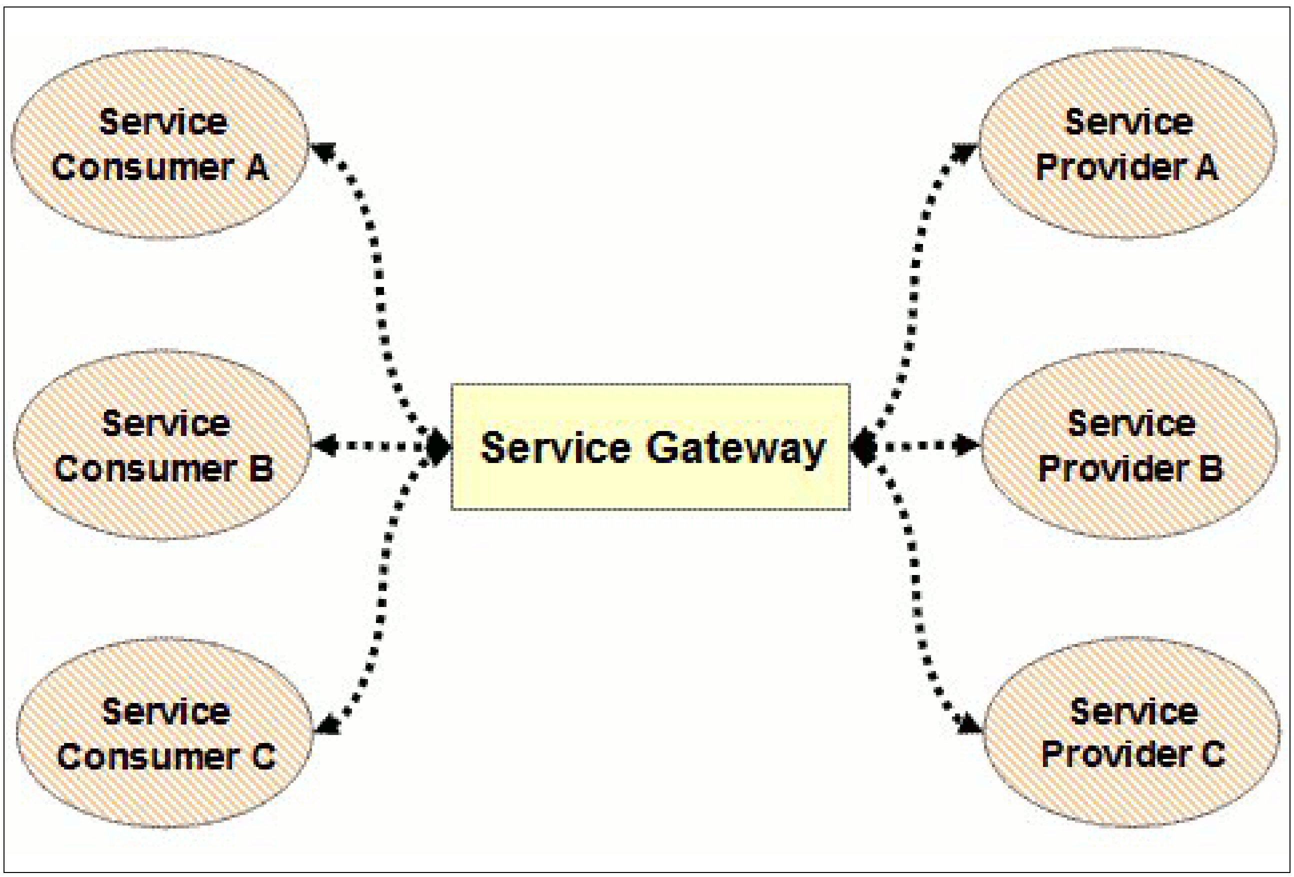


Figure 2-2 Gateway interaction pattern

Stopping proliferation of partner logic into the internal network

Business agility is the flexibility to add customers at a fast pace and change suppliers when the business needs it, not when the information technology is finally available. A major precondition to becoming agile is to define a place where all particular things about single customers or partners are handled. The gateway is the place designed for that task.

A single view of the partner

What does it mean when we say that a gateway maintains a single view of the partner regardless of its particularities? It means that we use an abstraction. We do not consider them as single partners, whether they are customer or supplier, but classify them into categories. The particularities are hidden under a single enterprise-wide facade. We do much the same when we define service abstraction in the internal network.

How service abstraction is done

The abstraction of different partner applications into a limited set of services happens on different levels:

1. Identify and assign: Identifying your abstract services candidates and assigning particular implementations of your partners to the internal service
2. Convert technically: Conversions of transport protocols and encoding
3. Translate semantics: Translations of field names and values

The first step only happens when you begin to work in a service oriented manner, and when new business fields are added. A simple example of this step is to abstract the ordering of goods from various suppliers into a single service definition. In the rare case, all of your suppliers would use the same EDI standards, but different transport protocols, you would only need to do technical conversions. In most cases, semantic translations are necessary.

> **Tip:** Prepare for future changes in your partner service definitions. Include explicit version numbers in your service definitions.

Non repudiation

Non repudiation has been discussed in the context of security aspects. This term refers to the archiving of information for the purpose of proofing the contents and time of a certain communication. That can be necessary in the result a dispute; for example, when a customer denies an order because they do not want to pay for goods you have delivered. In such cases, you are challenged to show proof of the order.

Technically, a widely used form of non repudiation proof is to save a copy (a hash) of the incoming message with the message digest that is generated with the private key of the sender. Because the private key is only known to the sender, and the hash (message digest) is a unique product of exactly the message to be proofed, it is a suitable method.

2.2.3 Technology choices

The technology choices for the perimeter network and gateway are shown in Table 2-1.

Table 2-1 Technology choices

	Sterling B2B Integrator	Sterling Secure Proxy	WebSphere DataPower XB62
DMZ component	Perimeter server proxy only	Proxy	Hardware appliance

	Sterling B2B Integrator	Sterling Secure Proxy	WebSphere DataPower XB62
Trusted network component	Configuration, mapping, translation, and so on. (See "Sterling B2B Integrator" on page 21)	Configuration Manager	Connect from a thin client for configuration. The XB62 typically sits in the DMZ and has no requirement for anything in the trusted zone; it can integrate directly to the internal ESB or application.
Protocols and standards supported	Wide range	Selected	Wide range. The only thing Sterling B2B Integrator has that XB62 does not have is Connect:Direct protocol.
Partner profiles	Yes	No	Yes
Strength	versatility and scalability	adds extra perimeter security	Low leniency, highly secure, ease of install, ease of maintenance
Partner Self service		Password management	Partner transaction visibility and resend capabilities
Cryptographic acceleration			Hardware based acceleration

The products mentioned Table 2-1 on page 20 are introduced in depth in the following sections.

Sterling B2B Integrator

IBM Sterling B2B Integrator offers perimeter security, gateway, and transformation engine in one product. It automates business processes shared with trading partners. The goal is automation of the complete "buy-sell-ship-pay" process that involves a range of documents and business processes, which include not just the buyer or seller of the goods/services, but also banks and third party logistics companies. The solution automates these processes and provides visibility into data and processes shared with external entities.

Sterling B2B Integration can address the following issues:

- Manual paper based processes
- Lack of visibility
- Limited internal resources
- Inefficient partner onboarding

Capabilities of Sterling B2B Integrator include the following features:

- Security features:
  - Identity management, including authorization and authentication
  - Perimeter security at DMZ traversal
  - Role-based data access and system operation
  - Secured mailboxing repository

  - Data transport security (SSL, SFTP/SSH) and data encryption (S/MIME and PGP) support
  - Non repudiation using the AS2 (Applicability Statement 2) or AS3 protocol
  - Digital signature support
  - Message- and transport-level security based on WS-Security 1.0 compliance, including WS-I Basic Profile 1.1 and Basic Security Profile 1.0

- Communication features:

  - B2B communication protocols: Web services (SOAP), S/FTP/S client and server, HTTP and HTTP/S, SMTP (Simple Mail Transfer Protocol), AS1, AS2, AS3 and RosettaNet, WebDAV, Zengin TCP/IP, IBM Sterling Connect:Direct, EBICS (Electronic Banking Internet Communication Standard)
  - Policy-based file transfer
  - Multi-gigabyte file handling
  - IPv6 compatible

- Business process management:

  - Graphical process modeling tool
  - Business process execution engine
  - Process abstraction (layered modeling and component reuse)

- Integration and transformation:

  - Multi-purpose data transformation engine
  - Traditional EDI: X12, EDIFACT, CII, TRADACOMS, and Verband der Automobilindustrie (VDA)
  - XML standards: OAGi, CIDX, PIDX, and RosettaNet
  - Internet standards for B2B data exchange: RosettaNet RNIF, ebXML, 1SYNC, and EBICS (France)
  - XSLT service to transform XML documents
  - Supports WTX translations
  - Graphic data mapping tool
  - Virtually unlimited file size (up to 50 gigabytes)
  - Validation of inbound and outbound data based on HIPAA rules defined for Level 1 – Level 6
  - Intelligent (content-based) routing
  - Interoperable with .Net 1.1/2.0, Axis 1.x/2.0, Xfire 1.2.6 and Java EES

- Community management:

  - Manage and grow trading partner communities
  - Centralized visibility into trading partner communities
  - Reduce error rates
  - Digital certificates deployment
  - Customizable partner configuration
  - Intelligent onboarding with partner self-provisioning

- Application extension and customization:

  - Web services: Support for SOAP, WDSL
  - Integrated Development Environment (IDE): Tool to speed custom app/dev work
  - Software Development Kit (SDK): Toolkit to create your own adapters to systems

- More than 300 comprehensive connectivity adapters:

  - Enterprise applications:

    SAP (BAPI, IDOC, and Netweaver), Oracle, Manugistics, PeopleSoft, Siebel, Vantive, JD Edwards, I2, IBM Sterling Connect:Direct, IBM Sterling Connect:Enterprise™, IBM Sterling Gentran:Server®, GXS, IBM

  - EAI messaging platforms
  - IBM WebSphere MQ, Oracle AQ, BEA Tuxedo, TIBCO Rendezvous, Vitria Businessware, WebMethods Enterprise, Microsoft MSMQ, JMS Queue and Topic Technology, JDBC, CORBA, LDAP, command line, file system, EJB, RMI, SNMP trap, IBM CICS®, JCA CCI, SSO, JSA, IM (Instant Messaging)
- Monitoring:
  - Monitoring with alerts for Sterling Integrator™ resources, including system status, database growth, and average business wait time
  - Running or restarting any failed or halted business process
  - Initiating key functions such as managing lock resolution for users and resources, initiating thread/ heap dumps, and turning logging on and off

Sterling File Gateway

Sterling File Gateway is an application for transferring files between partners using different protocols, file naming conventions, and file formats. It moves large and high-volume file transfers, with end-to-end visibility of file movement, in a process-oriented and highly-scalable framework. It also alleviates file transfer challenges, such as protocol and file brokering, automation, and data security.

Sterling File Gateway supports integration with Sterling B2B Integrator Mailbox, Sterling Control Center, Sterling Connect:Enterprise for UNIX server products, Sterling Secure Proxy and Sterling Connect:Direct. Sterling File Gateway is an add-on to the Sterling B2B Integrator platform with a unique application URL that provides single sign-on access to the Sterling B2B Integrator administrative console through menu selection.

Sterling File Gateway allows organizations to take complete control over file transfers with trading partners. Built on Sterling B2B Integrator, Sterling File Gateway offers a scalable architecture and a centralized file gateway with the capabilities necessary to monitor, administer, route, and transform high volumes of inbound and outbound files.

With Sterling File Gateway, the benefits of a standardized file transfer approach extend beyond the reliable and secure transmission of files with trading partners. A centralized gateway enables the consolidation of disparate file transfer activity. Intelligent routing and content-driven transformation capabilities help optimize file delivery processes. Subsequently, IT staff and users become more efficient, and platform consolidation helps reduce total cost of ownership.

How Sterling B2B Integrator and Sterling File Gateway work together

Within Sterling File Gateway, Sterling B2B Integrator is known as the B2B Console and is accessed from the Tools menu. Administrative functions, such as creating and managing user accounts, permission groups, and security keys for Sterling File Gateway, are handled in Sterling B2B Integrator.

Sterling File Gateway uses the following Sterling B2B Integrator communication adapters:

- FTP Server
- FTP Client
- SFTP Server
- SFTP Client
- HTTP Server
- HTTP Client
- Connect:Direct server
- Command Line adapter 2 (for PGP)

Sterling Secure Proxy

In the scenarios in this book, Sterling Secure Proxy acts as an application proxy between the external partners and the Sterling File Gateway server. It provides a high level of data protection between the external connections and an internal network. It allows the organizations to create firewall rules in order to prevent trading partners from obtaining direct access to their internal backend systems.

Sterling Secure Proxy is an application proxy that secures and shields your trusted network from external attacks by preventing direct communications between trading partners and internal servers. Sterling Secure Proxy provides demilitarized zone-based (DMZ) authentication, session breaks, and SSL terminations prior to allowing communications with the trusted network. It allows you to protect your trusted zone from unauthorized access by enforcing even tighter controls with multifactor authentication.

Capabilities of Sterling Secure Proxy include the following features:

- Application proxy:
  - Resides in the demilitarized zone (DMZ) and supports multiple DMZ environments
  - Meets customer requirements for all-electronic data transfer
  - Supports FTP, FTPS, HTTP, HTTPS, SSH/SFTP, PeSIT, and Connect:Direct protocols
  - Supports IBM Sterling Connect:Direct, IBM Sterling Connect Express and IBM Sterling B2B Integrator servers
- Firewall navigation:
  - Prevents inbound holes in the firewall
  - Minimizes rich targets in the DMZ by ensuring that files, user credentials and data are not stored in the DMZ
  - Establishes sessions from more-trusted to less trusted zones
  - Enforces internal and external security policies

Figure 2-3 provides an example.

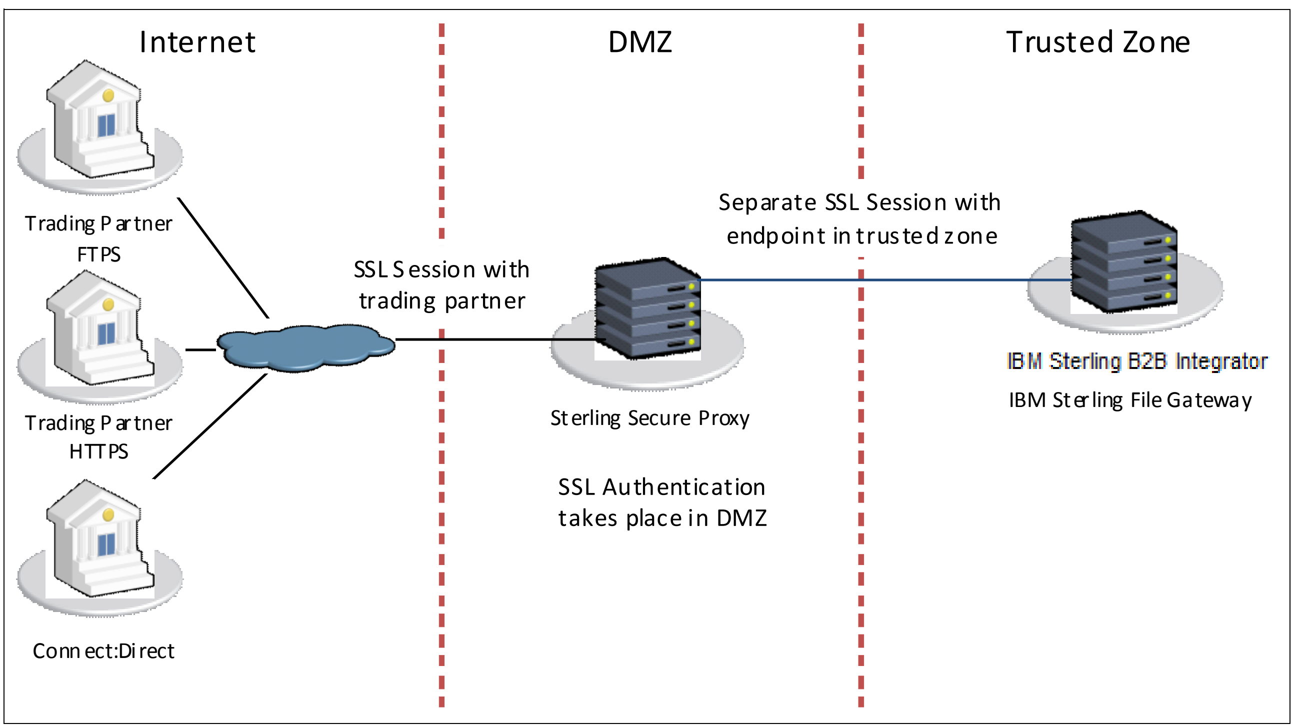


Figure 2-3 Sterling Secure Proxy Session breaks

- Perimeter security:
  - Session break and protocol validation: Prevents direct communications between internal and external sessions by establishing SSL session breaks in the DMZ, as shown in Figure 2-3.
  - Inspects protocol and sensitive control information, enabling configurable error handling for protocol violations.
  - Certificate validation and authentication: Authenticates incoming connections using the SSL or TLS protocol. Exchanges and validates certificates prior to allowing a separate connection to the trusted zone.
  - Multifactor authentication: Enforces tight controls with strong validation of trading partner identity in the DMZ using IP address, CRL checks, and custom lookups with options to interface with external user databases such as LDAP, Active Directory and IBM Tivoli® Access Manager.
  - Session limits and data encryption: Ensures business continuity and guards against Denial-of-Service attacks with support for SSL and TLS encryption algorithms.
  - User ID mapping: Protects internal applications by mapping trading partner user IDs and passwords to user IDs and passwords valid for internal systems.
- Clustering:
  - One central configuration manager pushes out configuration rules to multiple engines running in the DMZ, making it easy to scale.
  - Clustering for high availability and load balancing provides operational continuity and improved performance.

Sterling Secure Proxy is suitable for the technical part of security described in the DMZ. it does not provide the isolation features needed in a gateway. As a gateway, Sterling Connect:Direct and Sterling B2B Integrator can be used.

WebSphere DataPower XB62 appliance

Like Sterling B2B Integrator, WebSphere DataPower XB62 offers DMZ and gateway functionality. Because the WebSphere DataPower appliance is a firmware hardened appliance it provides a different approach by centralizing and consolidating the gateway features in the DMZ. You can quickly access new customers and new routes to market with standards-based trading partner management. In highly sensitive applications, cryptographic keys require the enhanced protection of certified FIPS 140-2 Level 2 or Level 3 hardware security modules (HSMs). The WebSphere DataPower XB62 appliance includes an HSM option. It augments the powerful authentication and SOA security capability of DataPower to further protect organizations' data.

Here are the main features of WebSphere DataPower XB62 appliance:

- Broad connectivity:
  - HTTP / HTTPS support (including SSL termination)
  - Supports WebSphere MQ 7 software-based high availability
  - Supports WebSphere MQ Channel exits for increased security
  - Secure File Transfer Protocol (SFTP) client and polling support
  - SFTP Server Service
  - Improved transactional integrity with IBM IMS™ Connect
  - FTP and FTPS support for both client and server side connections
  - Database connectivity
  - WebSphere Java Messaging Service
  - Tibco Enterprise Message Service (requires Option for Tibco Enterprise Message Service)
  - SMTP and POP support

- MQFTE Client Connections
- Network File System (NFS)

- Support for multiple authentication and authorization mechanisms:
  - WS-Security, WS-Security Policy, and Extensible Access Control Markup Language (XACML)
  - SAML 1.0, 1.1, and 2.0
  - S/MIME digital signature and encryption
  - XML digital signature and encryption
  - Integration with Tivoli Access Manager (requires Option for Tivoli Access Manager)
  - Full support for WS-Proxy validation of Message Transmission Optimization Mechanism (MTOM)-based SOAP messages
  - Increased WS-Security interoperability with WebSphere Application Server and Microsoft WCF framework
- Business to business:
  - Support for AS1, AS2, AS3, and ebMS
  - Trading partner management
  - Support for ebXML Collaboration Protocol Profile and Agreement v2.0
  - Support for EDI X12, EDIFACT and XML B2B payloads
  - B2B transaction visibility in an easy to use B2B Transaction Viewer
- Data transformation and validation:
  - Native XML Schema and WSDL validation
  - XSLT-based transformations
  - Any-to-any transformation using WebSphere Transformation Extender
  - Supports DPA maps created in WTX Design Studio
  - Supports type trees from the HL7 Industry pack
- Interoperability:
  - Flexible subscription support for service metadata stored in external repositories such as WebSphere Service Registry and Repository (WSRR)
  - Support for WS-Policy attachments authored within WSRR
  - Local mode support when integrating with Tivoli Access Manager
  - Full support for WS-Proxy validation of MTOM-based SOAP messages
  - Increased WS-Security interoperability with WebSphere Application Server
  - Integration with WebSphere Integration Developer for the development and deployment of XSLT transformations
  - Improved integration with WebSphere Transformation Extender
  - Integration with WebSphere Application Accelerator for Public Networks for optimal delivery of web and Web 2.0 applications across the public Internet
  - Integration to IBM z/OS®
  - Enhanced Integration to WebSphere MQ File Transfer Edition
  - Granular import and export capabilities
  - Deployment policies providing dynamic configuration changes upon import
  - Ability to configure the appliance in a Web GUI, using Command Line or using a SOAP
- Enhanced manageability:
  - Simplified backup and restore process
  - Java-based appliance management API with support for Jython
  - Granular logging capabilities
  - Probe to trouble process flow
  - SMTP Traps for monitoring appliance health and transaction statistics
  - Service Level Management allowing data shaping, throttling, or rejection

2.2.4 Small enterprises

With tight budgets and limited skill resources, small enterprises tend to choose a B2B solution that combines all of the needed functionality in one product and is easy to manage.

Sterling B2B Integrator offers a wide range of functionality with protocols and standards supported. It comes with perimeter security at DMZ traversal out of the box. The perimeter server comes as integral part of the product, but the component can be installed on a different system in DMZ, separated from the Sterling B2B Integrator server.

The perimeter service terminates the protocol in the DMZ and route the payload to the server using an internal protocol to Sterling B2B Integrator. It is configured only by the main Sterling B2B Integrator. Sterling B2B integrator calls the Perimeter Server from inside to establish the session. Data then flows between them in both directions.

Figure 2-4 provides details.

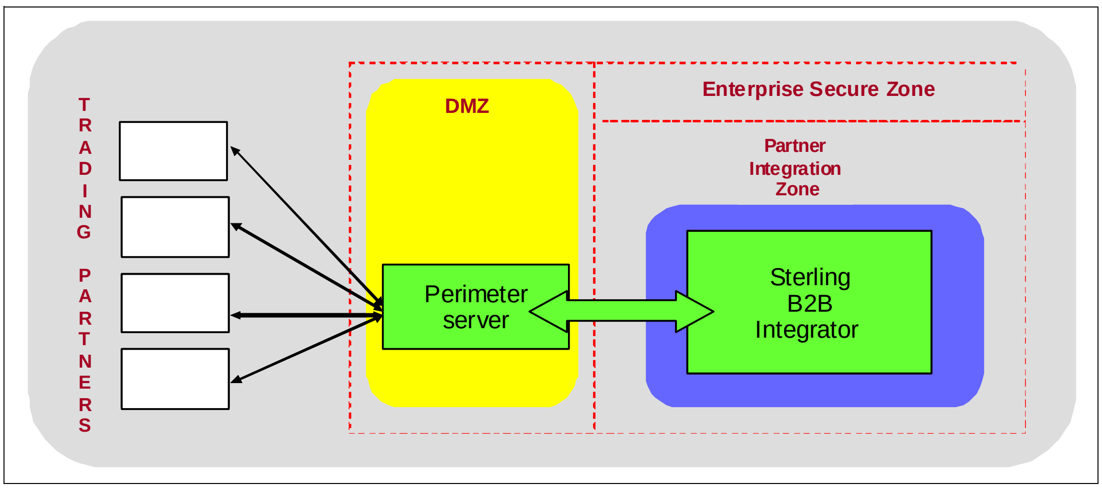


Figure 2-4 Sterling B2B Integrator perimeter server as session breaker

As IBM offers cloud-based, hosted Sterling B2B Services, it is also an alternative for small businesses who do not want to maintain their own installation and DMZ security. The hosted services guarantee defined levels of reaction to security threats and proactive security, such as updating to the newest levels, that otherwise can be hard to achieve for a small business.

2.2.5 Medium and large enterprises

Medium enterprises face different workloads and might need to scale up their infrastructure. They also can begin to combine products to make better use of product specific strengths, such as large enterprises.

Sterling B2B Integrator clustered

To scale for higher loads, additional instances of Sterling B2B Integrator can be added to work in a Sterling B2B integrator cluster. The two instances are much like a single logical node, and there is load balancing organized from the perimeter server and a tight band between the nodes, where load information is constantly exchanged. It happens through JMS communication.

Figure 2-5 provides details.

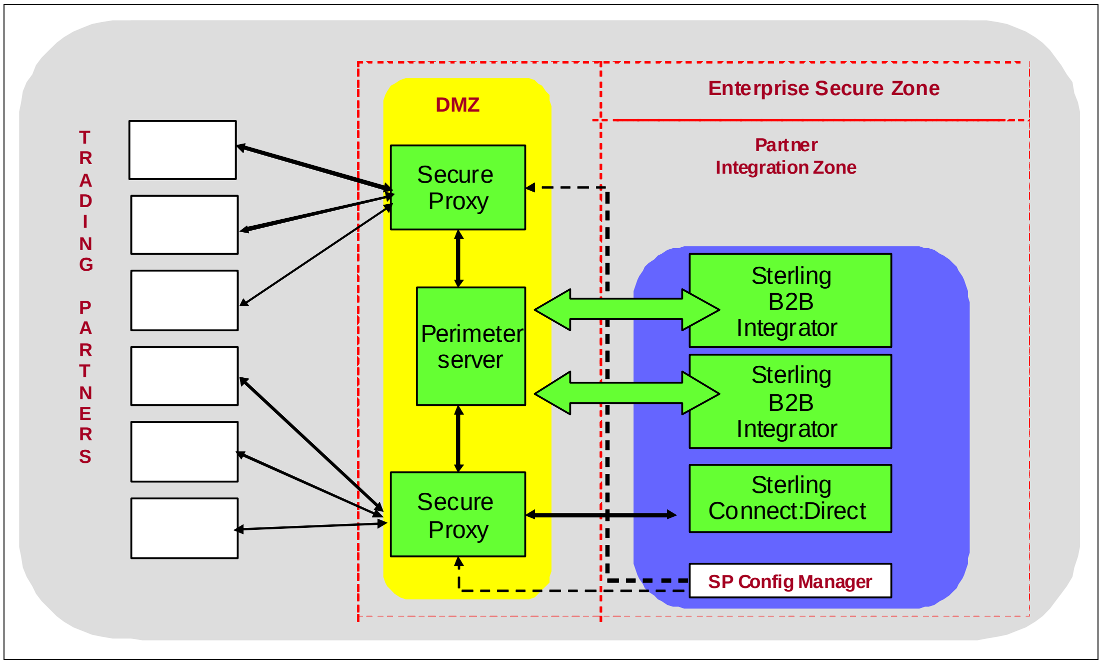


Figure 2-5 Sterling B2B Integrator clustered

Sterling Secure Proxy use

When a significant amount of the B2B load is lifted using the FTP, FTPS, HTTP, HTTPS, SSH/SFTP, PeSIT, and Sterling Connect:Direct protocols, the enterprise can use Sterling Secure Proxy to achieve additional perimeter security.

If a high amount of Connect:Direct file transfers is to be handled, the enterprise might want to use the standalone Connect:Direct server instead of the additional Sterling B2B Integrator Connect:Direct service. To place a Connect:Direct server into the DMZ would be a security risk, because it would expose files, user credentials, and data.

When the enterprise uses a multitude of clustered installations, Sterling Secure Proxy can distribute the configurations from a central Configurations Manager inside the trusted zone to all instances of Sterling Secure Proxy in the DMZ.

WebSphere DataPower XB62 use

The WebSphere DataPower B2B Appliance is well suited for use in the DMZ because it is a network security device that adds security gateway and integration functions in a tamper resistant hardware form factor. Additionally, it provides the option of being able to store private keys on a Hardware Security Module (HSM). Additionally, any sensitive data at rest in the DMZ on the device is stored AES encrypted.

The DataPower appliance can perform, not only some decryption and encryption tasks at very high velocity, but also other mappings close to or even at wire speed. For a number of a protocols and standards, it delivers a complete solution. With others, it can be used as security and speed enhancer in cooperation with Sterling B2B Integrator.

Because of its special hardened architecture as a firmware appliance, the DataPower is safe to do tasks in the DMZ that otherwise would not be done there.

Figure 2-6 provides details.

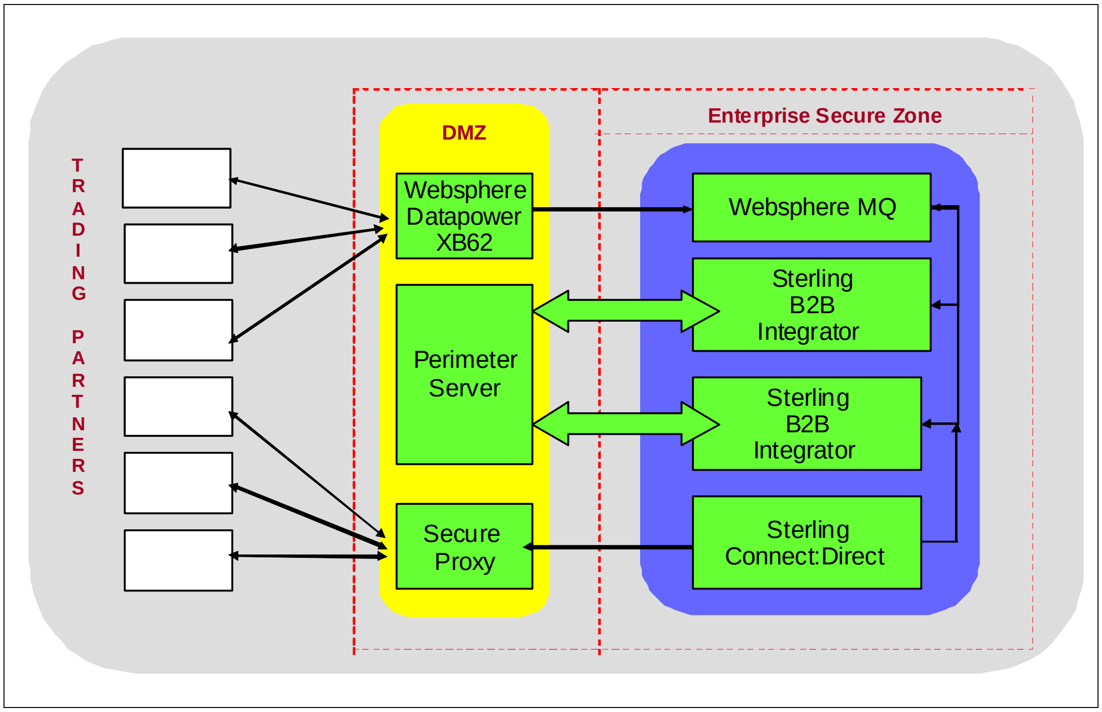


Figure 2-6 WebSphere DataPower XB62 in a DMZ perimeter zone

2.3 Seamless integration inside the enterprise

When dealing with internal integration, you need to determine how internal systems within the enterprise need to cooperate with each other and with the external partners. The technical setup of the communication is important, but it is even more important to consider the strategic and organizational aspects associated with it.

When talking about the seamless integration inside the enterprise, we need to consider two important concepts:

- Messaging
- Enterprise Service Bus

2.3.1 Connectivity matters: Messaging

If we consider connectivity in a modern medium or large enterprise, we might have to examine a significant number of different network protocols. Most of them are based on the IP protocol. Other base protocols, such as SNA, might only be relevant in smaller parts of the topology. But on top of the base of IP protocol, a large number of different protocols can be used. If you allow the connectivity to grow wild as projects demand it in the first request, you can end up with a connectivity tangle that is very hard to manage or can even be called unmanageable:

- Firewall administration becomes a tricky job; the rules list grows longer and longer.
- Loads generated by different tasks of different importance are transported at the same with undifferentiated service levels.

In particular, the situation described in the second bullet can lead to very bad effects: If a server hosting a customer order application is unresponsive because a router it is attached to is busy doing something that is not urgent (such as a database backup), the enterprise might pay fines for missing a service level agreement. There are different mitigations for such risks, such as separating the transport routes by adding hardware or managing communications more efficiently.

We think that the use of messaging can be an excellent choice for managing all communications by standardized means. Messaging can *wrap* other forms of communications and give the administrator a single point of management. Because messaging is not transient, communication can be replayed when needed. We discuss WebSphere MQ based messaging in this section. Many of the features mentioned here are also available in the Java based messaging, as in the WebSphere Application Servers Service Integration Bus.

Within MQ, you can distinguish the service levels for messages by the required reliability, speed, and security at different levels, but with a single point for maintenance and monitoring:

- You can vary the level of persistence from a hardened state. It is designed to recover messages also after severe errors, over a medium, where messages survive queue manager restarts to “not persistent,” where messages are not kept when the queue manager is stopped.
- The speed of messages can be varied by the batch size of transmission batches, priority flags, and the assignment of fast transmission means such as separate TCP/IP channels.
- Security is managed on a per message base by encrypting and signing payload from end-to-end, or on channel base, by encrypting and signing the whole channel traffic only between queue managers.
- Very large chunks of data, such as large files, can be split into a number of smaller messages, to prevent the large payload from blocking a communication channel completely for a long time.
- Messages can carry an “expiration” flag. For example, messages related to an online communication would expire when the online dialog has a timeout, to guarantee that no obsolete traffic clutters the network.
- If you want, message expiration can be reported to notify the operators.

These examples are taken from the toolbox that WebSphere MQ brings, helping you to wrap other protocols within WebSphere MQ. If you are not familiar with WebSphere MQ, read the following section for more information.

2.3.2 WebSphere MQ

WebSphere MQ is the market-leading messaging integration middleware product. Over more than 15 years, WebSphere MQ (or MQSeries® as it was known in earlier versions) has grown to provide flexible and reliable solutions that address the wide range of requirements introduced in the previous chapter.

A message queuing infrastructure built on WebSphere MQ technology provides an available, reliable, scalable, secure, and maintainable transport for messages with guaranteed once-only delivery. Many enhancements have been added to WebSphere MQ during its evolution in the marketplace:

- WebSphere MQ Clients: Enables an application to connect remotely or locally to a WebSphere MQ queue manager.
- Publish/Subscribe: Increases messaging capability from point-to-point messaging to a less coupled style of messaging.

- MQ Clusters: Allows multiple instances of the same service to be hosted through multiple queue managers, to enable load-balancing and fail-over and simplify administration.
- Secure Sockets Layer support: SSL protocol can be used to secure communication between queue managers or MQ Client.
- Diverse platforms: WebSphere MQ supports a wide range of operating system platforms.

Core concepts of WebSphere MQ

Data is transferred between applications in messages. A message is a container consisting of two parts:

- MQ Message Descriptor: Identifies the message and contains additional control information such as the type of message and the priority assigned to the message by the sending application.
- Message data: Contains the application data. The structure of the data is defined by the application programs that use it, and MQ is largely unconcerned with its format or content.

The nodes within a WebSphere MQ message queuing infrastructure are called queue managers. The queue manager is responsible for accepting and delivering messages. Multiple queue managers can run on a single physical server or on a wide network of servers across a large variety of different hardware and operating system platforms.

Each queue manager provides facilities for reliable messaging using both point-to-point and Publish/Subscribe styles.

The queue manager maintains queues of all messages that are waiting to be processed or routed. Queue managers are tolerant of failures and maintain the integrity of business-critical data flowing through the message queuing infrastructure.

The queue managers within the infrastructure are connected by logical channels over a communications network. Messages automatically flow across these channels from the initial producer of a message to the eventual consumer of that message based on the configuration of the queue managers in the infrastructure. Changes can be made to the configuration of queues and channels, and it is transparent to the applications.

Asynchronous messaging

Two applications that must communicate, whether hosted on the same machine or separate machines, might have originally been designed to do so directly and synchronously. It was a common messaging technique used prior to the introduction of WebSphere MQ.

In this case, the two applications exchange information by waiting for the partner application to become available and then sending the information. If the partner application is unavailable for any reason, including if it is busy performing communication with other applications, the information cannot be sent.

All intercommunication failures that can occur between the two applications must be considered individually by the applications, whether they are on the same machine or on different machines connected by a network. It requires a protocol for sending the information, confirming receipt of the information, and sending any subsequent reply.

Placing a WebSphere MQ infrastructure between the two applications allows communication to become asynchronous. One application places information for the partner in a message on a WebSphere MQ queue, and the partner application processes this information when it is available to do so. If required, It can then send a reply message back to the originator. The applications do not need to be concerned with intercommunication failures or recovery.

WebSphere MQ Client

WebSphere MQ Client is a light-weight component of WebSphere MQ that does not require the queue manager run-time code to reside on the client system. It enables an application running on the same machine as the client to connect to a queue manager that is running on another machine and perform messaging operations with that queue manager. Such an application is called a client and the queue manager is referred to as a server.

Using MQ Client is an effective way of implementing WebSphere MQ messaging and queuing. The benefits of doing this are as follows:

- There is no need for a licensed WebSphere MQ server installation on the client machine.
- Hardware requirements on the client system are reduced.
- System administration requirements on the client system are reduced.
- An application using MQ Client can connect to multiple queue managers on different machines.

Application programming interfaces (APIs)

Applications can use WebSphere MQ by several programming interfaces.

Message Queue Interface

The native interface is the Message Queue Interface (MQI). The MQI consists of the following features:

- Calls through which programs can access the queue manager and its facilities
- Structures that programs use to pass data to, and get data from, the queue manager
- Elementary data types for passing data to, and getting data from, the queue manager
- Classes in object-oriented languages for accessing data, the queue manager, and its facilities

Many programming languages and styles are supported depending on the software and hardware platform, for example, C, Java, and most other popular languages.

Standardized APIs

Utilizing a standardized API can add additional flexibility when accessing services through a message queuing infrastructure. This book uses the term standardized API to represent APIs that are not proprietary to an individual product, such as WebSphere MQ.

Here are some examples of standardized APIs that can be used to access services provided through a WebSphere MQ infrastructure:

- Java Message Service (JMS)
- IBM Message Service Client (XMS)

Wide adoption of these APIs can occur across multiple products. For example, the JMS API is an industry standardized API for messaging within the Java Enterprise Edition (Java EE) specification.

Reliability and data integrity

The intercommunication performed across channels between queue managers is tolerant of network communication failures, and WebSphere MQ assures once-only delivery of messages.

Persistent and non-persistent messages

Messages containing critical business data, such as receipt of payment for an order, need to be reliably maintained and must not be lost in the event of a failure. On the other hand, some messages might only contain query data, where the loss of the data is not crucial because the query can be repeated. In this case, performance might be considered more important than data integrity.

To maintain these opposite requirements, WebSphere MQ uses two type of messages, persistent and non-persistent:

- Persistent messages: WebSphere MQ does not lose a persistent message through network failures, delivery failures, or restart of the queue manager. Each queue manager keeps a failure-tolerant recovery log of all actions performed upon persistent messages. It is sometimes referred to as a journal.
- Non-persistent messages: WebSphere MQ optimizes the actions performed upon non-persistent messages for performance. Non-persistent message storage is based in system memory, so it is possible they can be lost in situations such as network errors, operating system errors, hardware failure, queue manager restart, and internal software failure.

Units of work

Many transactions performed by an application cannot be considered in isolation. An application might need to send and receive multiple messages as part of one overall action. Only if all of these messages are successfully sent or received, are any messages sent or received.

An application that processes messages might need to perform coordinated work against other resources as well as the WebSphere MQ infrastructure. For example, it can perform updates to information in a database based upon the contents of each message. The actions of retrieving the message, sending any subsequent reply, and updating the information in the database must only complete if all actions are successful.

These actions are considered to be a unit of work (UOW). Units of work performed by applications accessing a WebSphere MQ infrastructure can include sending and receiving messages as well as updates to databases. WebSphere MQ can coordinate all resources to ensure that a unit of work is only completed if all actions within that unit of work complete successfully.

2.3.3 Enterprise Service Bus (ESB): Enabler for Service Oriented Architecture

In software engineering, a Service Oriented Architecture (SOA) is a set of principles and methodologies for designing and developing software in the form of interoperable services. In order to implement an SOA, both applications and infrastructure must support the SOA principles. Enabling applications involves the creation of service interfaces to existing or new functions, either directly or through the use of adapters. Enabling the infrastructure at the most basic level involves the provision of capability to route and transport service requests to the correct service provider. The role of the Enterprise Service Bus is, in part, simply to enable the infrastructure in this way.

Value of the ESB concept

The true value of the Enterprise Service Bus (ESB) concept, however, is to enable the infrastructure for SOA in a way that reflects the needs of today's enterprise: to provide suitable service levels and manageability, and to operate and integrate in a heterogeneous environment.

The ESB must enable the substitution of one service implementation by another with no effect on the clients of that service. It requires both the service interfaces that are specified by SOA and that the ESB allows client code to invoke services in a manner that is independent of the service location and communication protocol that is involved.

ESB introduces features that can improve responsiveness, customer service, transaction time, and partner interactions. An ESB provides capabilities that enhance both direct connection between applications and routing requests among applications.

An ESB supports the concepts of SOA implementation in the following ways:

- Decoupling the consumer's view of a service from the implementation of a service
- Decoupling technical aspects of service interactions
- Integrating and managing services in the enterprise

Decoupling the consumer's view of a service from the actual implementation greatly increases the flexibility of the architecture. It allows the substitution of one service provider for another (for example, because another provider offers the same services for lower cost or with higher standards) without the consumer being aware of the change or without the need to alter the architecture to support the substitution.

This decoupling is better achieved by having the consumers and providers interact through an intermediary. Intermediaries publish services to consumers. The consumer binds to the intermediary to access the service, with no direct coupling to the actual provider of the service. The intermediary maps the request to the location of the real service implementation.

In an SOA, services are described as being loosely coupled. However, at implementation time, there is no way to loosely couple a service or any other interaction between systems. The systems must have some common understanding to conduct an interaction. Instead, to achieve the benefits of loose coupling, consideration must be given to how to couple or decouple various aspects of service interactions, such as the platform and language in which services are implemented, communication protocols used to invoke services, and data formats used to exchange input and output data between service consumers and providers.

Further decoupling can be achieved by handling some of the technical aspects of transactions outside of applications. It can apply the following aspects of interactions:

- How service interactions are secured.
- How the integrity of business transactions and data are maintained (for example, through reliable messaging, the use of transaction monitors, or compensation techniques).
- How the invocation of alternative service providers is handled in the event that the default provider is unavailable.

The role of the ESB is to fulfill these needs by providing the following functions:

- Map service requests from one protocol and address to another.
- Transform data formats.
- Support a variety of security and transactional models between service consumers and service providers and recognize that consumers and providers might support or require different models.
- Aggregate or disaggregate service requests and responses.
- Support communication protocols between multiple platforms with appropriate qualities of service.
- Provide messaging capabilities such as message correlation and publish/subscribe to support different messaging models such as events and asynchronous request/response.

Figure 2-7 shows the role of the ESB.

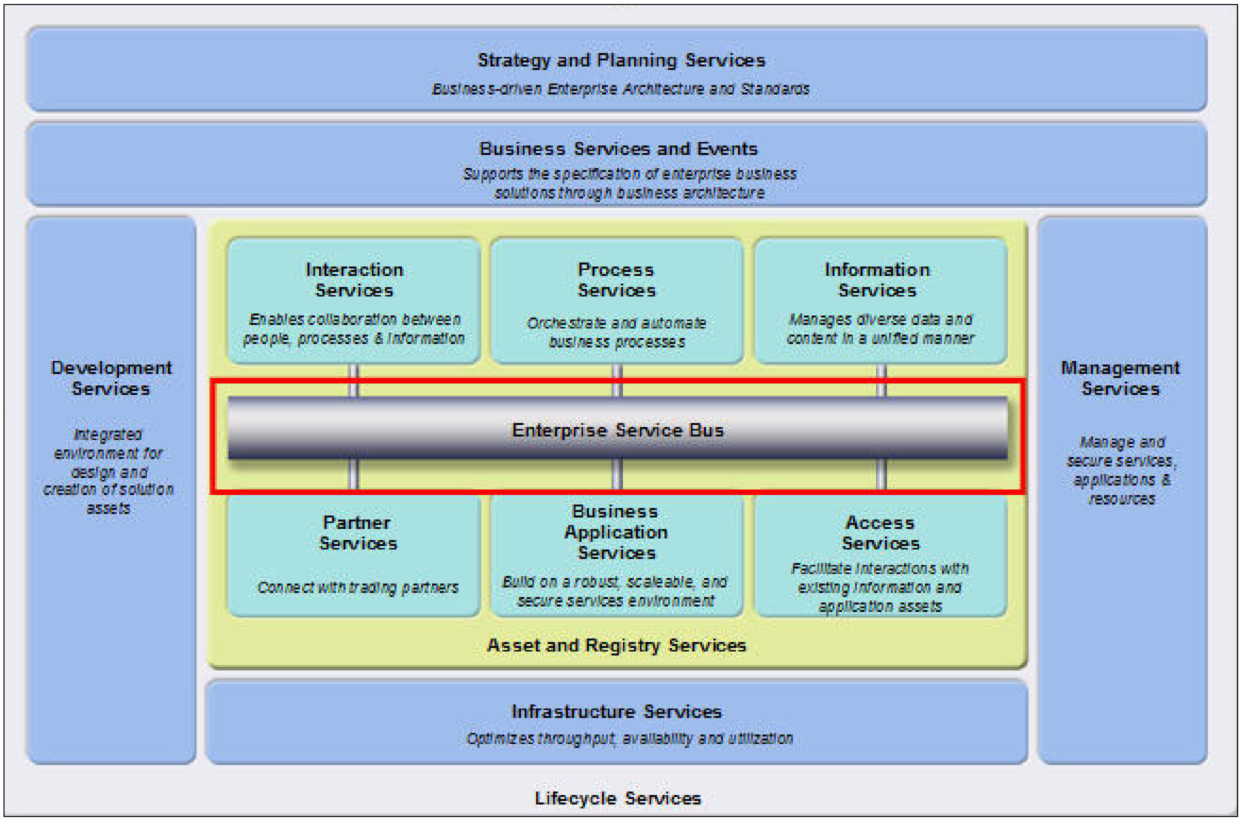


Figure 2-7 The role of the ESB

Application integration versus service integration

Again, invoking the front office scenario described in 2.2.2, "Gateway function" on page 19, we need to determine how the front desk clerk who is serving a customer can communicate with the back office. The front desk clerk has the following choices:

- Take care of the customer
- Pass the request to a another person
- Pass the request to a function such as another department
- Consult a manual to determine the best course of action

Questions for connectivity and format can only be answered afterwards: how the addressee can be reached (call personally, by phone, mail, and so on), which data needs to be passed, and in which form the data needs to presented.

The traditional approach of enterprise application integration often focuses on the question of how to connect applications only to each other. For example, it helps the front desk clerk to contact another person, but it does not help to place a loans request without knowing which person to call.

Obviously, it depends on the size of the enterprise and how much integration is needed. But there is no absolute criteria on how to classify a given enterprise. Other important factors are the dynamics of the enterprise, resources available, and regional distribution. Even small companies would be well advised to abstract their functions to services and implement an extendable enterprise service bus when they expect to grow in the long term, and for that expectation, decide on a strategic orientation in their IT.

2.3.4 IBM ESB products

There are many products that can be used to create an Enterprise Service Bus (ESB). IBM provides several options for creating an ESB, as shown in Figure 2-8. It allows customers to select ESB technology based on their specific needs. In many large organizations, more than one ESB technology can be used to create a hybrid ESB as required by differences in geography, technology, or for other reasons. Hybrid ESBs are especially likely when two large departments each embark on SOA separately and then find they need to interoperate with each other.

The three main IBM ESB products are WebSphere Message Broker, WebSphere ESB, and WebSphere DataPower SOA Appliances.

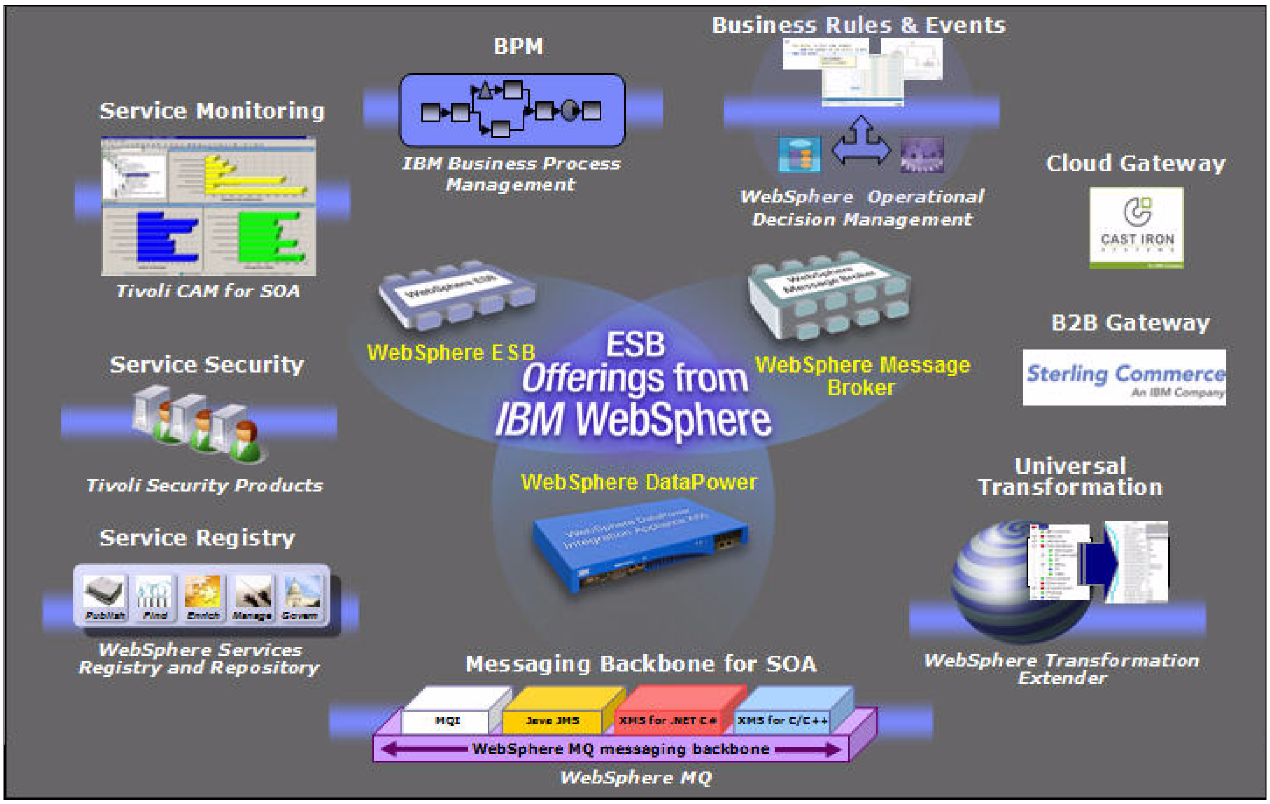


Figure 2-8 IBM ESB offerings

WebSphere Message Broker

WebSphere Message Broker offers capabilities that many customers use to create their ESB (or a central ESB in a hybrid solution). WebSphere Message Broker has its heritage in the WebSphere MQ messaging space, and is particularly well suited for environments heavy in MQ messaging. It offers interface definitions for WSDL and other message formats, mediation capabilities through message flows, and support for a range of communication formats, including WMQ and HTTP. WebSphere Message Broker also offers content based publish / subscribe interactions and managed topic spaces.

WebSphere Message Broker supports implementation of mediation patterns for a broad spectrum of service interaction endpoints. It supports a large number of industry standard message sets (some, but certainly not all using XML encodings) and enables support for additional, user-defined message formats.

The following typical customer requirements are well suited to a WebSphere Message Broker solution:

- Transactional
- Publish/subscribe
- ACORD, SWIFT, or COBOL copybook standard formats
- XML formatted data
- WS-* standard compliant
- WebSphere MQ messaging
- Complex transformations
- Complex Event Processing

WebSphere Message Broker provides numerous ESB connectivity options and any-to-any data transformation. It enables legacy applications and those that do not conform to standards to connect to an ESB.

WebSphere Message Broker is described in more detail in 4.3.1, "WebSphere Message Broker" on page 74.

WebSphere ESB

WebSphere ESB is a J2EE instantiation of an ESB. WebSphere ESB runs within a J2EE container in the IBM WebSphere Application Server. It is well suited to J2EE and WS-* standards-based applications, especially those applications that only need to communicate with other application servers. It also provides an excellent entry point into the SOA world, allowing basic web services to be utilized as a stepping-stone into a more robust SOA environment.

WebSphere ESB provides standards-based Web Services connectivity, JMS messaging, and service oriented integration. JMS applications for point-to-point and pub/sub messaging, and JAX-RPC service oriented applications can connect directly to WebSphere ESB, or messages can be delivered to WebSphere ESB across a variety of transports including WMQ, SOAP/HTTP and SOAP/JMS. WebSphere ESB implements a web services gateway which can mediate between SOAP/HTTP and SOAP/JMS based applications. Finally, it also provides an implementation of Universal Description Discovery and Integration (UDDI). The following typical customer requirements are well suited to a WebSphere ESB solution:

- J2EE implementations
- Web Services interfaces
- SOAP/HTTP

WebSphere DataPower SOA Appliance

WebSphere DataPower SOA Appliance is a hardware and software offering that provides a number of important functions: XML acceleration, security enforcement, and ESB functionality. DataPower has several important characteristics:

- Optimized hardware, firmware, and imbedded operating system
- A high level of assurance that the configuration is locked-down
- Reduced security vulnerabilities
- Hardware storage of encryption keys and a locked audit log
- No hard disks, CD ROMs, or USB ports
- Tamper-proof case that renders the machine unusable if opened
- Reduced operational complexity, as it is truly an SOA appliance

DataPower provides the fastest entry into the SOA world, while at the same time providing an enhanced security environment.

The following typical customer requirements are well suited to a DataPower solution:

- Security Gateway
- XML firewall, parsing and validation
- Basic routing
- Content-based routing

Protocol bridging (HTTP, WebSphere MQ clients, FTP, ODBC, etc.)

2.3.5 Technology choices

Table 2-2 shows the different approaches of seamless integration with the internal systems:

Table 2-2 Technology choices

	WebSphere ESB	**WebSphere Message Broker**	**WebSphere DataPower**
Possible complexity in transformations	High	High	Medium
Relative latency	Regular	Low	Close to zero
Out-of-the box Interoperability	Standards	Wide	Specialized
Extensibility	Very high (JCA)	High	Medium
Human task support	Yes, after upgrade to the WebSphere Process Server	No	No
Transaction manager	Internal coordinator WAS	Internal coordinator WMQ	External
Transactions	Regular and extended	Regular and extended	Regular and extended (JTA)
Pricing	Medium unit	Large unit	Low maintenance costs

For more information, see this website:

http://www-01.ibm.com/software/integration/wsesb/v6/faqs.html#provide

Technology selection criteria

In this section, we discuss the criteria that you need to keep in mind when developing your technology solutions.

Interaction style

In order to fully support the variety of interaction patterns that are required in a comprehensive SOA (such as request/response, publish/subscribe, and events), the ESB must ideally support in one infrastructure the three major styles of enterprise integration:

- SOAs in which applications communicate through reusable services with well-defined, explicit interfaces. Service-oriented interactions leverage underlying messaging and event communication models.
- Message-driven architectures in which applications send messages through the ESB to receiving applications.
- Event-driven architectures in which applications generate and consume messages independently of one another.

A consumer can implement a service invocation in different ways. From the consumer's point of view, the difference is as follows:

- Synchronous: The consumer uses a single thread to invoke the service; the thread sends the request, blocks while the service is running, and waits for the response.
- Asynchronous: The consumer uses a pair of threads to invoke the service; one thread sends the request, then a separate thread listens for and receives the response.
- Publish/subscribe: A service publishes messages on a specific topic. Multiple services (subscribers) can subscribe to this topic and receive the published messages.

An ESB can offer any combination of these invocation models for a single service, letting service consumers select a preferred invocation model.

Interaction style can influence the ESB implementation. IBM WebSphere Message Broker is especially suited to architectures in which the basic flow paradigm is asynchronous or pseudo-synchronous.

Statefulness

Certain situations might require an ESB to maintain state as messages traverse it. A simple example involves selecting a particular service endpoint from the context information of the last applicable message. A more complex example extends the ESB paradigm by detecting complex situations that involve a context-sensitive composition of messages and events (semantic context, temporal context, spacious-temporal context). This functionality is extremely powerful when applied to the processing of input from multiple event sources: from the perspective of a business, application, or infrastructure within different contexts. For example, such scenarios might involve Service Level Agreement alerts and compliance checking, which is applicable to the security, finance, banking, and insurance industries.

Hardware ESB implementations do not generally support stateful interactions, making statefulness a criterion for software ESB implementations. WebSphere Message Broker, uniquely, implements complex message processing directly in the product, making it the most suitable for use cases requiring this function.

Endpoints, standards, and protocols

Endpoints are consumers and services that interact through the ESB. Endpoints that employ standard technology and protocols (such as web services) allow the most flexibility in ESB technology selection. However, it is often not practical or cost-effective to restrict endpoints to these standards where large legacy investments exist. Similarly, while integrating packaged applications is possible using their proprietary APIs, it is often useful to delegate this integration to a middleware vendor. To address these concerns, adapters exist that can greatly simplify integration.

The use of adapters influences ESB product selection because adapters require a software runtime (they are not supported on hardware ESB implementations) and prerequisites for adapters vary.

Policy-based interaction management and dynamic service selection

An emerging ESB technology involves dynamic endpoint selection that exploits agent-based endpoint statistics such as server utilization and health. It is supported out of the box in certain IBM ESB products through integration with IBM Tivoli Composite Application Manager.

Message volume, size, and type

ESB products have different scalability characteristics. In general, hardware-based implementations scale quite well and have excellent support for industry standard data formats. However they do not provide the versatility of software implementations. WebSphere Message Broker performs very well, offers a very rich set of ESB functionality, and can be extended with custom components and adapters. WebSphere ESB has a rich set of support for industry standard formats, Java and Web Services standards, and can be extended with custom components and adapters.

Reliability, availability, and serviceability (RAS)

ESB products have different operational characteristics and they achieve scalability and high availability through different mechanisms and architectural footprints (one example is application server clustering versus OS-level clustering). It must be taken into consideration and the resulting solution architecture must balance RAS considerations against existing budget, skills within the organization and operational complexity.

Required mediations

ESB products differ in their ability to handle mediations without custom programming. Notably, WebSphere Message Broker has extensive support for standard data formats such as ACORD, SWIFT, and COBOL copybooks. If data formats are XML and Web Services, WebSphere DataPower SOA Appliances are an ideal fit, offering very high performance.

2.3.6 Small enterprises

The small enterprise must find solutions that fit its limited financial and human resources. It probably cannot afford to pay much for significant individual programming or adaptions.

Issues of ad-hoc integration

Small enterprises will probably choose highly integrated “custom off the shelves” software packages that offer most of the capabilities they need. It other words, they would make use of the shared database pattern for internal integration, as the vintage model companies did. Yet, there will be issues with their remaining integration needs. Examples include trying to solve their requirements ad-hoc with whatever converter, adapter, bridge, or transformation tool is in reach, such as a download from the Internet, ad-hoc scripting (when the IT person has the knowledge), or simply by manually transferring (copying and pasting) from one window to another.

If such a direct approach is chosen in an environment with eight components, it results in 42 bi-directional connections. Unlike the symbolic edges in Figure 2-9, the shape of the links would vary in terms of protocols and transports. It leads to operational complexity that can become expansive and dangerous.

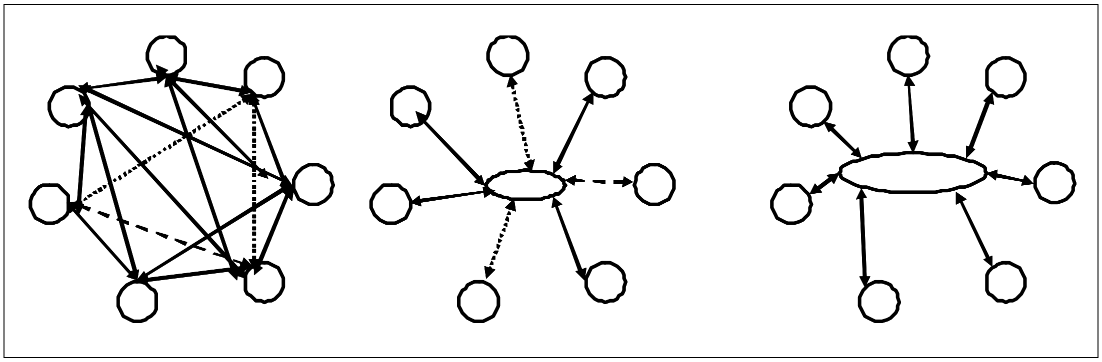


Figure 2-9 Direct connectivity, hub and spoke and bus approach

The most important consideration is that manual synchronization and most ad-hoc connectivity solutions do not guarantee transactional integrity.

Changes: *Transactional integrity* means that changes are propagated "exactly once, and only once" between systems.

Integrated cloud services

A viable alternative to homemade integration inside a small company can be buying an integrated cloud services solution from an external provider. The cloud attributes are very attractive for small businesses:

- Enhanced user experience
- Elastic scaling
- Automated provisioning
- Highly virtualized
- Flexible pricing

It is particularly true when the small enterprise uses a number of applications in the cloud and the integration data traffic stays in the cloud. If there are many interactions between applications and services hosted in different places, it would be problematic where the integration software is placed, because network latency becomes an issue. We assume that most of the services to be integrated are hosted in the cloud. Therefore, integration must also happen there.

The use of cloud based services that are not integrated have at least the same drawbacks as the non-integrated on-premises applications. Small enterprises tend to participate in public or shared cloud services as opposed to a hosted private cloud.

The cloud service provider uses highly developed integration software products (as we discuss in the following sections) to offer integration services for a higher number of clients, often combined with industry-specific extensions. There can be win/win situations when providers include some customization in order to better adapt their offering to the specific industry. The enterprise gets the service it needs at a good price and the cloud service provider improves its portfolio.

2.3.7 Medium and large enterprises

Medium and large enterprises who want to avoid the direct connection tangle often choose a "hub and spoke" integration approach where applications are connected by adapters to a broker that acts as intermediary. In the sample shown in Figure 2-9, the number of links is reduced from 42 to 7.

The concept of enterprise service bus (Figure 2-10) leverages hub and spoke architecture to the next level by the abstracting applications to services.

- More indirection
- More scalability
- More flexibility for different ways to do things

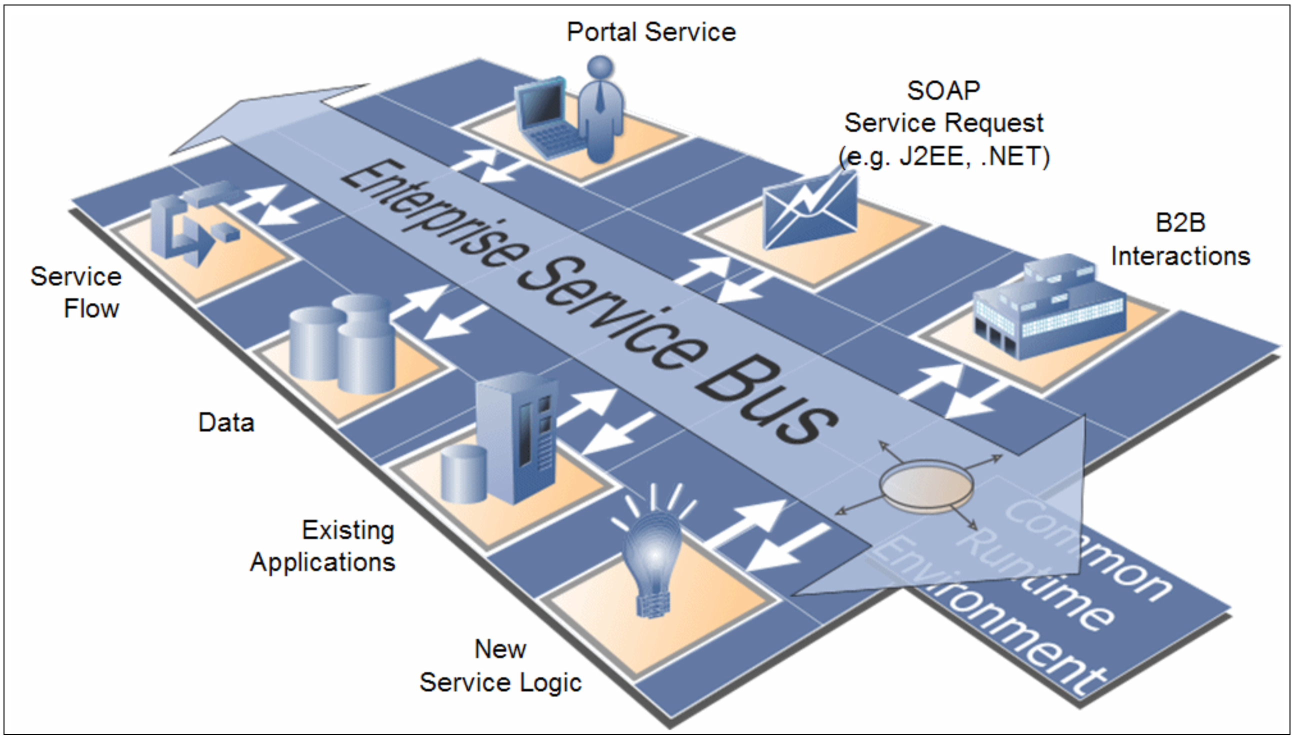


Figure 2-10 Enterprise Service Bus - Conceptual -

An ESB connects requestor and provider, decoupling the interactions between them in support of separation of concerns. An ESB is part of a connectivity infrastructure that enables decoupled, dynamic, adaptable service interaction.

An ESB is used to connect the service requester to the service provider so that messages can be routed between the two platforms. The ESB is a collection of software components that manage messaging from one part of the network to another. The ESB handles mismatches between the requesters and providers, including protocol, interface, or quality of service mismatches.

The ESB processes messages exchanged between the service endpoints (Figure 2-11). In contrast with regular business application components, the ESB is concerned with the flow of the messages through the infrastructure and not just with the business content of the messages. Rather than performing business functions, the ESB performs mediation capabilities including, routing, transformation, and logging operations on the messages.

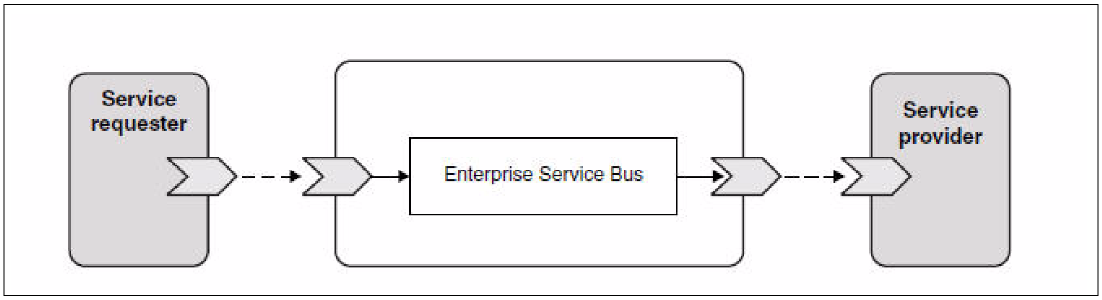


Figure 2-11 Messages exchanged between the service endpoints

Interposing the ESB between participants enables you to modulate their interaction through a logical construct called a *mediation*. Mediations operate on messages in-flight between requesters and providers. For example, mediations can be used to find services with specific characteristics that a requester is asking for, or to resolve interface differences between requesters and providers. For complex interactions, mediations can be chained sequentially.

An enterprise service bus, with mediations, performs the following actions between requester and service:

- *Routing* messages between services. An enterprise service bus offers a common communication infrastructure that can be used to connect services, and thereby the business functions they represent, without the need for programmers to write and maintain complex connectivity logic.
- *Converting* transport protocols between requester and service. An enterprise service bus provides a consistent, standards-based way to integrate business functions that use different IT standards. It enables integration of business functions that cannot normally communicate, such as to connect applications in departmental silos or to enable applications in different companies to participate in service interactions.
- *Transforming* message formats between requester and service. An enterprise service bus enables business functions to exchange information in different formats, with the bus ensuring that the information delivered to a business function is in the format required by that application.
- *Handling* business events from disparate sources. An enterprise service bus supports event-based interactions in addition to the message exchanges to handle service requests.

Hybrid ESB: One difference between medium and large enterprises is that, as described in 2.3.4, “IBM ESB products” on page 36, some large organizations use more than one ESB technology to create a hybrid ESB as required by differences in geography, technology, or for other reasons.

3

Connecting with your partners and managing your communication

In this chapter, we discuss typical scenarios for a company to exchange business data with trading partners. Several possible implementations are considered and described, depending on the business requirements. We also discuss managing exchanged business data.

This chapter includes the following sections:

3.1 What are the business requirements?

The scenarios that we discuss in this chapter take the following business requirements into consideration:

- Operation and performance characteristics, including these factors:
  - Onboarding partners:
    - The configuration necessary for onboarding an external trading partner
    - The need for installing a software on an external trading partner environment
    - The external trading partner skill to deploy and configure any software
    - Any outsourced services
    - The number of external trading partners
  - Operation and performance:
    - The need for encrypting data exchange
    - The need for doing data exchange without manual intervention
    - How often the data exchange occurs
    - The different file sizes that must be accommodated
    - The data exchange performance necessary to satisfy processing windows
    - Triggering processing after the data has been received
- Managing data exchange, including these factors:
  - Being able to get a warning if the connection or transmission fails
  - Being able to define what data exchange should happen. when it needs to happen, and get a warning if it does not occur
  - Being able to troubleshoot any problem from a central location

3.2 Typical scenarios and implementation

This section describes some sample scenarios showing different possible implementations for business data exchange and monitoring of the environment.

3.2.1 Sample scenario: IBM Sterling File Gateway

The scenario described in this section has the following assumptions:

- There are external trading partners (represented by Company A).
- There is a centralized location to which all external trading partners communicate (represented by Company B).

Scenario requirements

It is assumed that Company B has the following needs when communicating to external trading partners (Company A):

- Onboarding an external trading partner must require minimum configuration.
- There is no need to install any software on the external trading partner environment.
- Some external trading partners might not have personnel who are skilled to deploy or configure any software.
- Several different protocols need to be available to adapt to different external trading partners requirements.

- Existing software on the external trading partners environment must be accepted whenever possible.
- Different file sizes and amounts must be accommodated.
- It is necessary to stage data sent to some external trading partners.
- Data processing needs to be automated when received from an external trading partner.

Scenario implementation

IBM Sterling File Gateway can be used to satisfy the needs of Company B in the ways described in the following sections.

Onboarding an external trading partner must require minimum configuration

With IBM Sterling File Gateway, you can use easy-to-user graphical user interface to onboard partners and configure the various combinations of communication protocols to enable Sterling File Gateway operation. In this scenario, a new external trading partner (Company A) definition needs to be created. Previous definition templates can be reused.

There is no need to install any software on the external trading partner environment

IBM Sterling File Gateway provides a myFileGateway application that runs from a WebBrowser. From within myFileGateway, external trading partners can do these tasks:

- Upload or download files
- Search for routes in which they participated
- View recent activity and status for file transfers in which they participated
- Generate reports for activity in which they participated
- Change their password
- Subscribe to notifications

Several different protocols must be available to adapt to the requirements of different external trading partners

IBM Sterling File Gateway uses a wide range of protocols going into an external trading partner (Company A) server. These protocols are described in Table 3-1.

Table 3-1 Supported protocols by IBM Sterling File Gateway

Protocol	Encryption	Business Purpose
FTP/S	SSL	Widely available, easy to configure.
SFTP	SSH/SFTP	Widely available, easy to configure.
SFTP	SSH/SCP	SSH/SCP provides an alternative means to exchange information with trading partners who do not have SFTP clients. The SFTP Server adapter enables trading partners with SCP clients to exchange files with Application Mailboxes. To the external users, the Mailbox is a directory on which the user has privileges.
HTTP/S	SSL	Supports application myFileGateway.
Connect:DIrect	SSL/TLS STS	Connect:Direct uses a proprietary high performance protocol.

Existing software on external trading partners must be accepted whenever possible

If Company A already has an IBM Sterling Connect:Direct Server or a Sterling B2B Integrator Connect:Direct Adapter, it can be used to communicate to Sterling B2B Integrator Sterling File Gateway Connect:Direct Server adapter.

Different file sizes should be accommodated

If it is necessary for Company A (external trading partners) and Company B to exchange large size files and/or big amounts of files or they have a small processing window that other protocols could not meet, or need to exchange data without manual intervention, then IBM Sterling Connect:Direct could be used. See 3.2.2, "Sample scenario: IBM Sterling Connect:Direct" on page 50 below. IBM Sterling File Gateway supports Connect:DIrect protocol, as shown in Table 3-1 on page 47.

> **Tip:** IBM Sterling Connect:Direct supports several features providing a very high performance: multi-sessions, sessions retry, checkpoint/restart/ compression.

Need to stage data sent to some external trading partners

If Company A (external trading partners) are not available when Company B has data to send the data can be stored on Sterling B2B Integrator Mailbox. Next time Company A initiates a connection the data can be retrieved from Mailbox.

Need to automate data processing when received from an external trading partner

You have several options when integrating mailboxes with the rest of the application.

The most commonly used option is integrating the mailboxes with business processes. There are two primary methods you can use:

- Invoking the Mailbox services from a business process.
- Specifying an application business process using a routing rule.

See the following websites for more details:

http://help.sterlingcommerce.com/SB2BI52/topic/com.ic.customization.doc/SI_52_PDF/SI52_Mailbox_31923_22331.pdf

http://help.sterlingcommerce.com/SFG22/topic/com.ic.customization.doc/SFG22_Overview.pdf

Scenario details

Figure 3-1 describes this scenario.

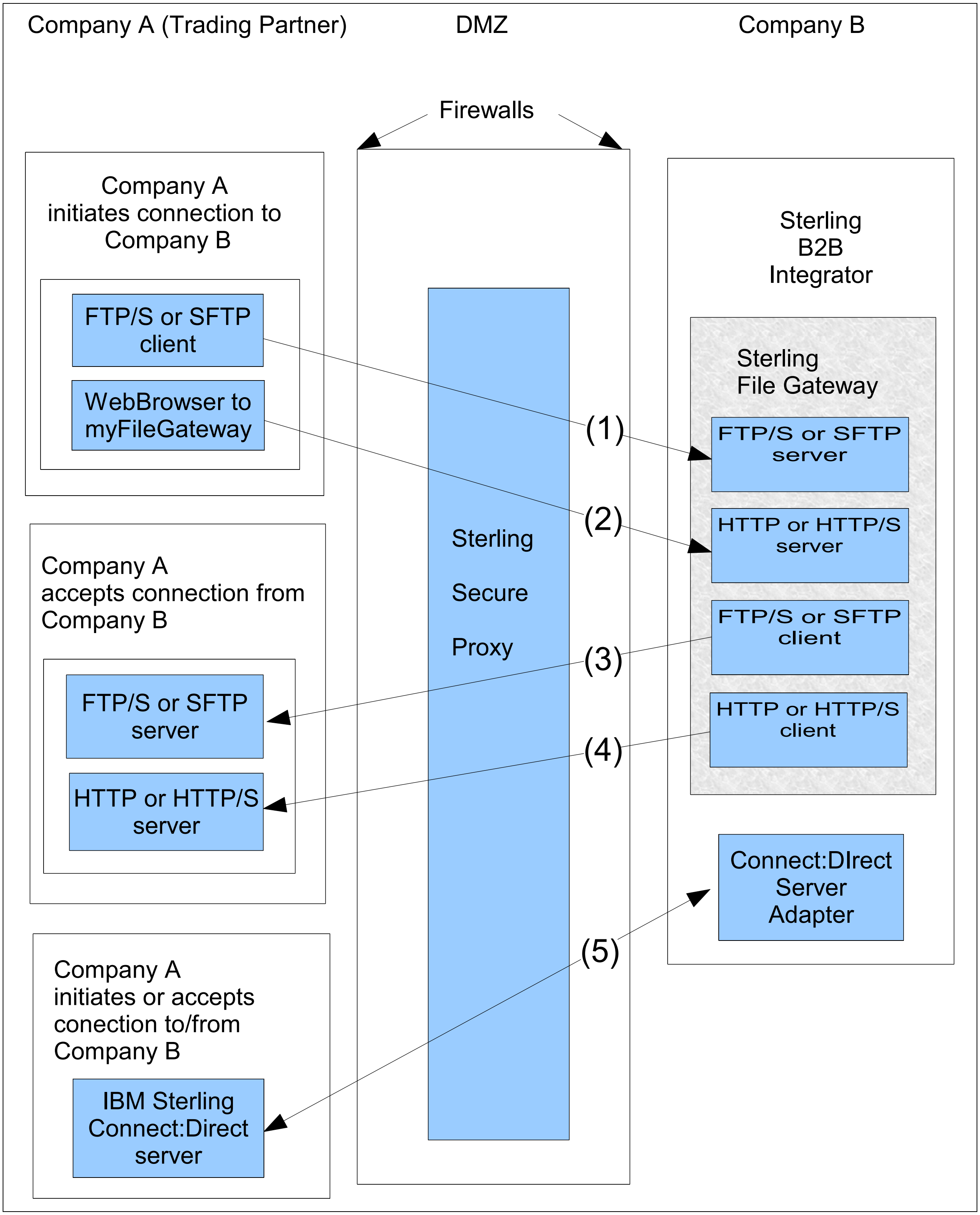


Figure 3-1 Sample IBM Sterling File Gateway scenario

This scenario has the following assumptions:

- External trading partners (Company A) are placed outside of the DMZ.
- Sterling Secure Proxy is placed inside the DMZ.
- Company B is placed internal to the DMZ.

The following text describes the flow of this scenario:

1. Arrow (1): An FTP/S or SFTP session initiated by external trading partners (Company A), traversing through Sterling Secure Proxy and reaching Sterling File Gateway FTP/S or SFTP server.
2. Arrow (2): An HTTP or HTTP/S session initiated by external trading partners (Company A), using myFileGateway application, traversing through Sterling Security Proxy and reaching Sterling File Gateway HTTP or HTTP/S server.
3. Arrow (3): An FTP/S or SFTP session initiated by Company B, traversing through Sterling Secure Proxy and reaching external trading partners (Company A) FTP/S or SFTP server.
4. Arrow (4): An HTTP or HTTP/S session initiated by Company B, traversing through Sterling Secure Proxy and reaching external trading partners (Company A) HTTP or HTTP/S server.
5. Arrow (5): A session initiated either by external trading partners) Company A or Company B, traversing through Sterling Secure Proxy. The participants of this session are external trading partners (Company A) IBM Sterling Connect:Direct server and Company B Sterling B2B Connect:Direct Server adapter.

3.2.2 Sample scenario: IBM Sterling Connect:Direct

The scenario described in this section has the following assumptions:

- There are external trading partners (represented by Company A).
- There is a centralized location to which all external trading partners communicate (represented by Company B).

Scenario requirements

It is assumed that Company B has the following needs when communicating to external trading partners (Company A):

- High volume data is exchanged with high number of files or very large files with high performance.
- High performance is required.
- Data exchange occurs without manual intervention.
- Detailed logging of data exchange operations is done.
- There is a need to automate data processing after it has been received from an external trading partner.

Scenario implementation

IBM Sterling Connect:Direct or the IBM Sterling B2B Connect:Direct Server Adapter can be used to satisfy the following needs of Company B.

High volume data is exchanged, high number of files, or very large files with high performance

IBM Sterling Connect:Direct supports several features providing very high performance: multi-sessions, sessions retry, checkpoint/restart/ compression, and data exchange without manual intervention. With IBM Sterling Connect:Direct, there is no limit on the file size to be transferred. IBM Sterling Connect:Direct supports FileAgent to automate data exchange operations. See the *IBM Sterling Connect:Direct File Agent Configuration Guide* at the following website for more information:

http://www.sterlingcommerce.com/documentation/home/MFT/ConnectDirect/File%20Agent/CD%20FileAgent14/CDFAConfigGuide.pdf

Data exchange without manual intervention

IBM Sterling B2B Connect:Direct Server Adapter operations can be triggered from within a Business Process without manual intervention.

Detailed logging of data exchange operations

IBM Sterling Connect:Direct statistics log every operation in detail. In general, the same information exists on the statistics of both IBM Sterling Connect:Direct instances that had communicated. IBM Sterling B2B Connect:Direct Server adapter generates Business Process statistics.

Need to automate data processing after it has been received from an external trading partner

IBM Sterling Connect:Direct exits or Process Language can be used to automate the processing of data received from an external trading partner

For more information, see the *IBM Sterling Connect:Direct Process Language Reference Guide* at this website:

http://www.sterlingcommerce.com/Documentation/Processes/CDProcessesLanguageRefGuide.pdf

IBM Sterling B2B Connect:Direct Server adapter can use Sterling File Gateway Mailbox services to automate the processing of data received from an external trading partner.

Scenario details

Figure 3-2 describes this scenario.

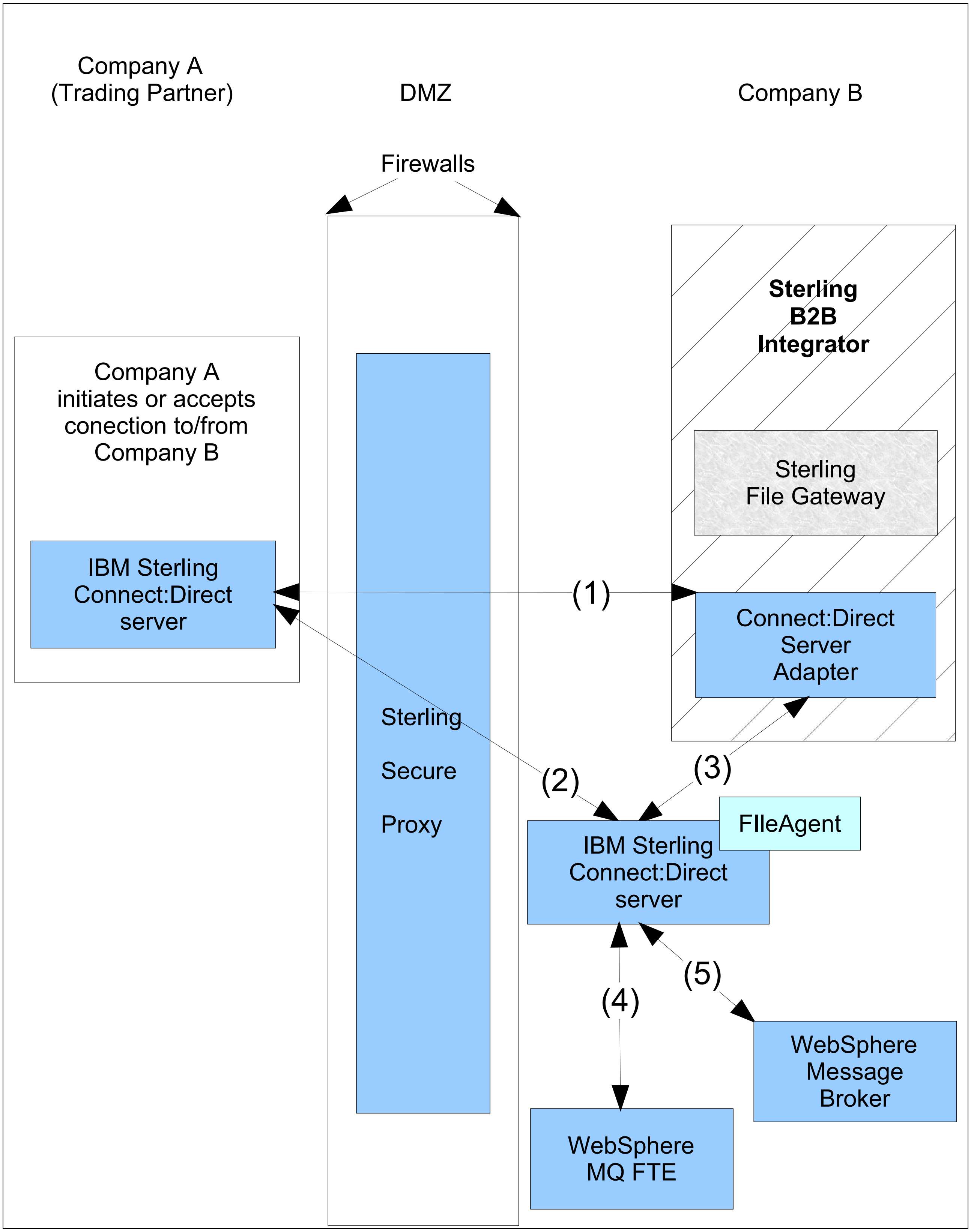


Figure 3-2 Sample IBM Sterling Connect:Direct scenario

This scenario has the following assumptions:

- External trading partners (Company A) are placed outside the DMZ.
- Sterling Secure Proxy is placed inside the DMZ.
- Company B is placed internal to the DMZ.

Here we describe the flow of this scenario:

1. Arrow (1): A session initiated either by external trading partners) Company A or Company B, traversing through Sterling Secure Proxy. The participants of this session are external trading partners (Company A) IBM Sterling Connect:Direct server and Company B IBM Sterling B2B Connect:Direct Server adapter
2. Arrow (2): A session initiated either by external trading partners) Company A or Company B, traversing through Sterling Secure Proxy. The participants of this session are external trading partners (Company A) IBM Sterling Connect:Direct server and Company B IBM Sterling Connect:Direct server.
3. Arrow (3): A session initiated by Company B internal to the DMZ. The participants of this session are Company B IBM Sterling Connect:Direct server and Company B IBM Sterling B2B Connect:Direct Server adapter. This session is shown to represent a possible connectivity internal to Company B.
4. Arrow (4): A session initiated by Company B internal to the DMZ. The participants of this session are Company B IBM Sterling Connect:Direct server and Company B WebSphere MQ FTE. This session is shown to represent a possible connectivity internal to Company B.
5. Arrow (5): A session initiated by Company B internal to the DMZ. The participants of this session are Company B IBM Sterling Connect:Direct server and Company B WebSphere Message Broker. This session is shown to represent a possible connectivity internal to Company B.

3.2.3 Sample IBM Sterling Control Center scenario

The scenario described in this section has the following assumptions:

- There are external trading partners (represented by Company A).
- There is a centralized location to which all external trading partners communicate (represented by Company B).

Scenario requirements

It is assumed that Company B has the following needs when communicating to external trading partners (Company A):

- Any failure in a data exchange will be immediately detected and notified.
- Any planned data exchange that does not occur will be notified before their Service Level Agreement (SLA) is not met.
- Availability of a centralized console with necessary data to determine problem cause or begin problem determination.

Scenario implementation

IBM SterlingControl Center can be used to satisfy Company B needs as described below.

Any failure in a data exchange will be immediately detected and notified

IBM Sterling Control Center manages IBM Sterling File Gateway and IBM Sterling Connect:Direct. You can define rules to follow up any data exchange operation that are critical to some SLA or processing window. Any failure in those operations will be immediately notified.

Any planned data exchange that does not occur will be notified before their Service Level Agreement (SLA) is not met

Service Level Criteria (SLC) can be defined for any planned data exchange that is critical to some SLA or processing window. If the planned data exchange fails or does not occur a notification will be sent.

Availability of a centralized console with necessary data to determine problem cause or begin problem determination

IBM Sterling Control Center gathers data about data exchange operations that can be accessed by their local or remote consoles. IBM Sterling Control Center can also be used to configure IBM Sterling Connect:DIrect servers.

For more information, see *IBM Sterling Control Center System Administration Guide* at this website:

http://www.sterlingcommerce.com/documentation/SCC53/PDF/SystemAdministration.pdf

Scenario details

Figure 3-3 describes this scenario.

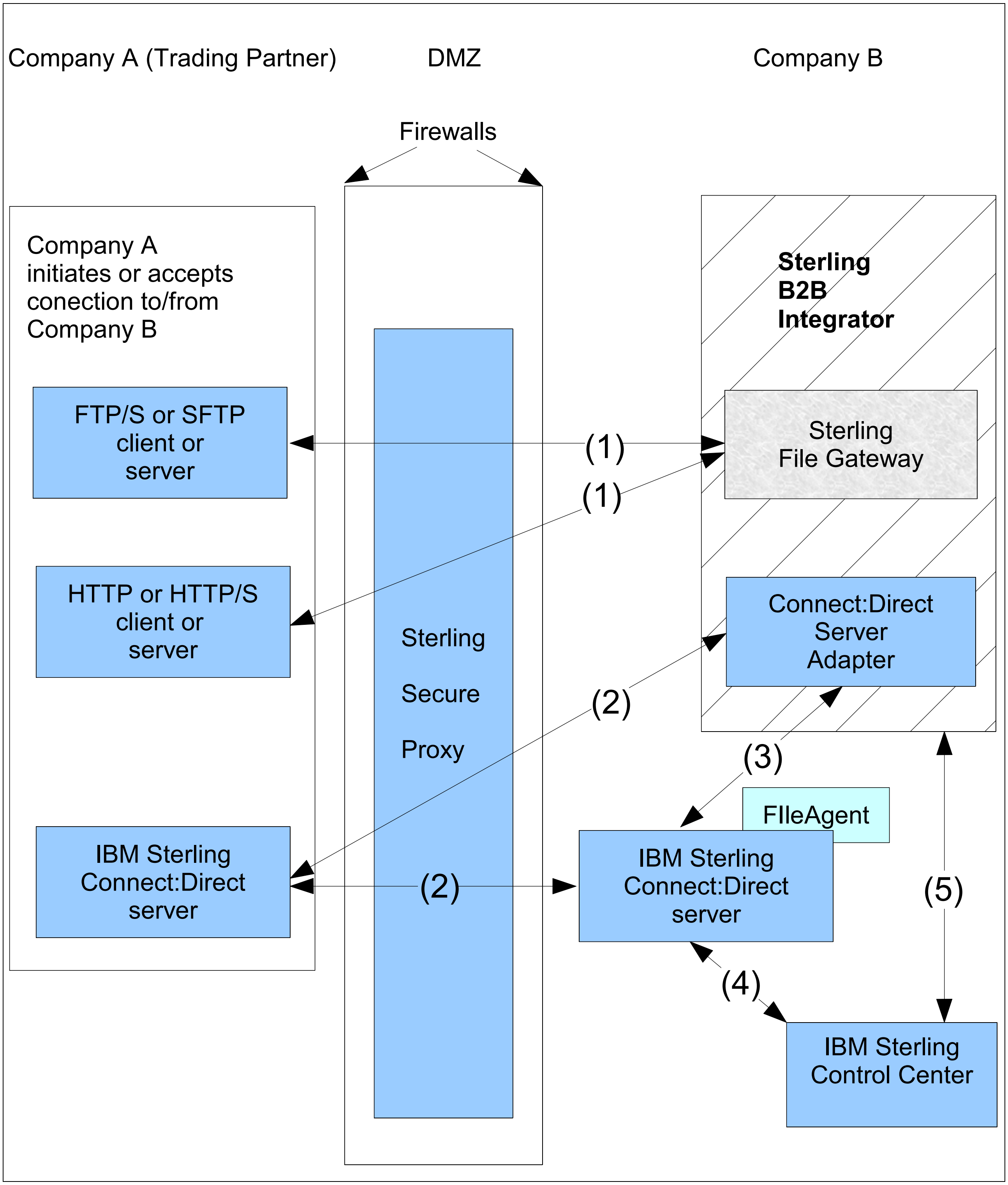


Figure 3-3 Sample IBM Sterling Control Center scenario

This scenario has the following assumptions:

- External trading partners (Company A) are placed outside the DMZ.
- Sterling Secure Proxy is placed inside the DMZ.
- Company B is placed internal to the DMZ.

Here we describe the flow of this scenario:

1. Arrow (1) represents:
   - An FTP/S or SFTP session initiated by external trading partners (Company A), traversing through Sterling Security Proxy and reaching Sterling File Gateway FTP/S or SFTP server.
   - An HTTP or HTTP/S session initiated by external trading partners (Company A), using myFileGateway application, traversing through Sterling Security Proxy and reaching Sterling File Gateway HTTP or HTTP/S server.
   - An FTP/S or SFTP session initiated by Company B, traversing through Sterling Secure Proxy and reaching external trading partners (Company A) FTP/S or SFTP server.
   - An HTTP or HTTP/S session initiated by Company B, traversing through Sterling Secure Proxy and reaching external trading partners (Company A) HTTP or HTTP/S server.
2. Arrow (2) represents:
   - A session initiated by external trading partners (Company A) traversing through Sterling Secure Proxy and reaching Company B Sterling B2B Integrator Connect:Direct Server Adapter.
   - A session initiated by external trading partners (Company A) traversing through Sterling Secure Proxy and reaching Company B IBM Sterling Connect:Direct Server.
3. Arrow (3) represents an internal session between Company B Sterling B2B Integrator Connect:Direct Server Adapter and Company B IBM Sterling Connect:Direct Server.
4. Arrow (4) represents IBM Sterling Control Center managing Company B IBM Sterling Connect:Direct Server operations.
5. Arrow (5) represents IBM Sterling Control Center managing Company B IBM Sterling B2B integrator operations.

3.2.4 Sample IBM Sterling B2B Cloud Services scenario

IBM Sterling B2B Cloud Services fill a void in the portfolio of on-premise software products by introducing software solutions ranging from on-demand to fully managed services, providing customers with the flexibility to quickly adapt as their business needs change and evolve. And the hybrid strategy of providing a combination of software on-premise and services in the cloud provides customers secure connectivity and collaboration with 100% of their business partners.

See Figure 3-4 for details.

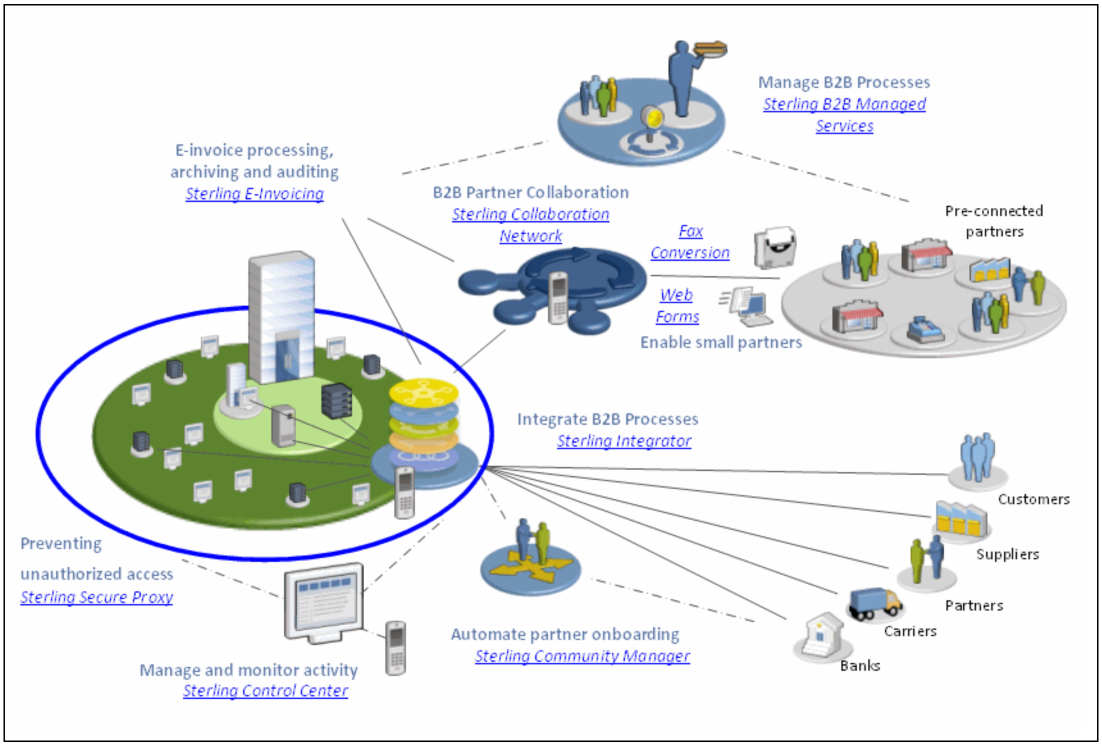


Figure 3-4 Sample IBM Sterling B2B Cloud Services scenario

Sample scenarios of adoption of IBM Sterling B2B Cloud Services are described in detail in Chapter 8, “IBM Sterling B2B Cloud Service scenarios” on page 313. Customers benefit from adopting the cloud services to meet the following business requirements:

- Reduce the time and complexity of securely and flexibly integrating 100% of partner communities, having little operational impact on IT staff.
- Rely on a highly scalable platform and elastic resource capacity to grow and shrink as needed, responding to dynamic business needs.
- Accelerate the ROI of B2B projects and use their internal resources where they need them most, and use outsourced services instead of establishing complex expertise in house.
- Provide centralized visibility into actionable information across key IT and business processes.

3.2.5 Sample WebSphere DataPower B2B Appliances XB62 scenario

In this section, we provide an overview of a banking integration scenario. In this case, we implement a one-way flow, which corresponds with an incoming payment from the hub Enterprise Resource Planning (ERP) system, that needs to be sent to a specific banking partner. A message comes through the DataPower device by WebSphere MQ (WMQ) queues and performs a message transformation to XML. This XML is then wrapped in AS3 and then sent to the specific partner’s backend. The DataPower device then waits for an asynchronous AS3 Message Disposition Notification (MDN) before the transaction is treated as complete.

All the messages between the partner and the provider are exchanged in encrypted format and signed as a consequence of the assumed trading manager agreement.

Figure 3-5 shows the flow from a high-level perspective.

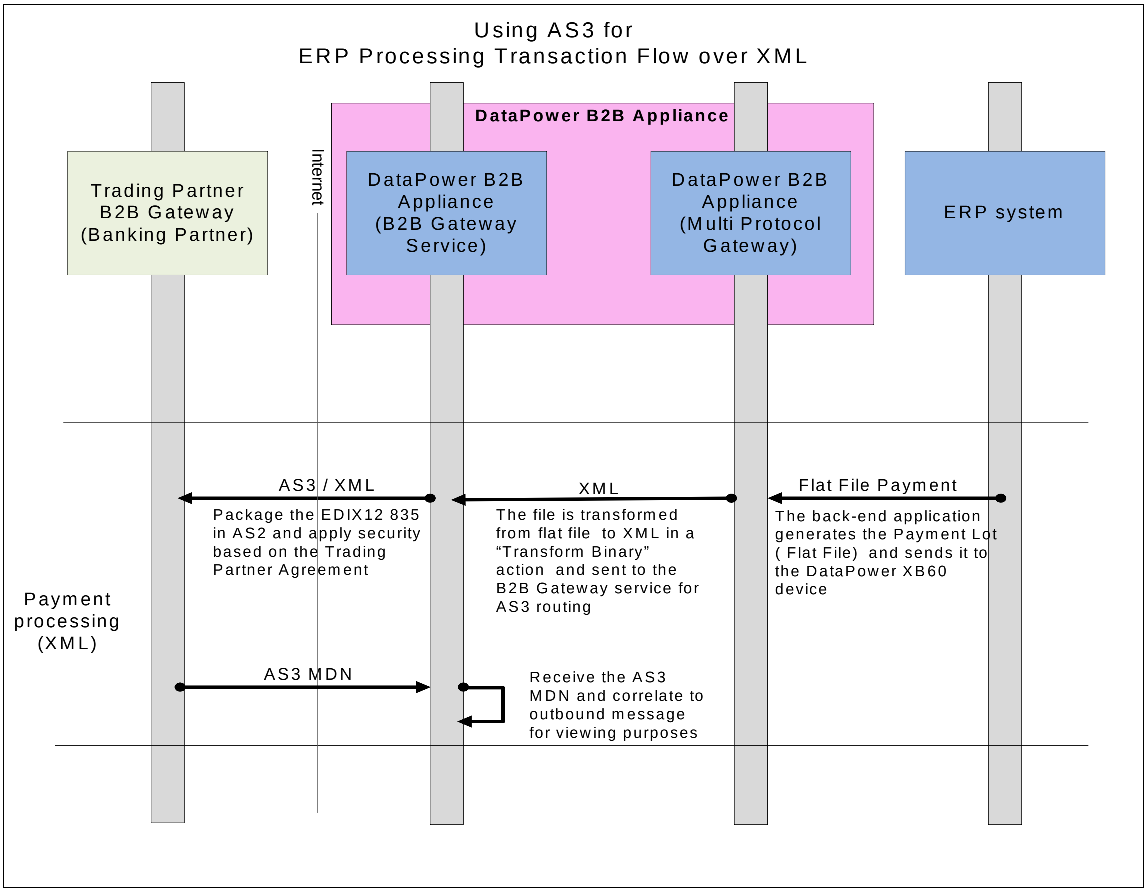


Figure 3-5 Scenario overview

More details of this scenario and implementation details can be found in *IBM WebSphere DataPower B2B Appliance XB60 Revealed*, SG247745.

3.3 Product information

In this section, we describe the products used in the scenarios described in this chapter.

3.3.1 IBM Sterling File Gateway

IBM Sterling FIle Gateway is an application for transferring files between partners using different protocols, file naming conventions, and file formats. It runs as an application on IBM Sterling B2B Integrator, to which it is fully integrated.

You can see *the Sterling File Gateway Overview* document at this website:

http://help.sterlingcommerce.com/SFG22/topic/com.ic.customization.doc/SFG22_Overview.pdf

IBM Sterling File Gateway features

IBM Sterling File Gateway has the following features:

- File/File name Transformations: Mapping of input to output file names, system-wide, group, and partner-specific policies; common file processing tasks such as compression/decompression, PGP encryption/decryption, and signing.
- File Transfer Visibility: Events are recorded for monitoring and reporting; detailed tracking for input-output file structure processing and dynamic route determination; ability to view and filter Sterling File Gateway data flows for all users.
- Replay/Redeliver: One-click replay/redeliver capability that allows users to reprocess a transmission from the beginning or to resend just the processed file to a specify delivery destination.
- Notifications: Partners and operators can subscribe to be notified about events by emails.
- Predefined business processes: Define common behaviors in file-transfer scenarios, reducing the need for customization.
- Extensibility: Custom event codes, protocols, facts, and consumer identification policies can be added to support unique scenarios.
- Broad communications Protocol Support: FTP, FTP/S, SSH/SFTP, SSH/SCP, and Sterling Connect:Direct are supported upon installation, and additional protocols (such as AS2, AS3, or Odette FTP) can be configured through use of the extensibility feature.
- Partner Interface (myFileGateway): Web browser-based interface that enables partners to upload/download files, subscribe to notifications of events, manage passwords, search and view file transfer activity, and generate reports about file transfer activity.
- Flexible Mailbox Structures: Ability to specify mailbox structures that leverage pattern matching policies and specify attributes that must be true of all partners or a subset of partners.
- Dynamic Routing: Consumer derived at run-time, either through mailbox structure, file name, business process-derived consumer name, or map-derived consumer name.
- Partner Onboarding: Easy-to-user graphical user interface to onboard partners and configure the various combinations of communication protocols to enable Sterling File Gateway operations.

IBM Sterling File Gateway integration with IBM Sterling B2B Integrator

Sterling File Gateway utilizes the Sterling B2B foundation. Administrative functions such as creating and managing user accounts, permission groups, and security keys for Sterling File Gateway are handled in Sterling B2B Integrator.

Sterling File Gateway utilizes the communication adapters of Sterling B2B Integrator, which include the following adapters:

- FTP Server adapter
- FTP Client adapter
- SFTP Server adapter
- SFTP Client adapter
- HTTP Server adapter
- HTTP Client adapter
- Connect:Direct Server adapter
- Command Line adapter 2 (for PGP)

IBM Sterling File Gateway Web Application myFileGateway

Sterling File Gateway provides a Web Application named myFileGateway that can be used to:

- Upload or download files
- Search for routes in which they participated
- View recent activity and status for file transfers in which they participated
- Generate reports for activity in which they participated
- Change their password
- Subscribe to notification

IBM Sterling File Gateway Mailboxes

Mailboxes can be used to stage date passing between internal systems and trading partners. Whenever data is produced when trading partners are not available to receive it, or trading partners send data that cannot be immediately processed, Mailboxes can be used to store data until it can be retrieved.

Routing is another important feature of Mailboxes. Routing provides Mailboxes with the ability to automate the processing of data received from trading partners.

IBM Sterling File Gateway integration with IBM Sterling Secure Proxy

IBM Sterling Secure Proxy can be deployed in a DMZ between IBM Sterling File Gateway and trading partners, as shown in Figure 3-6.

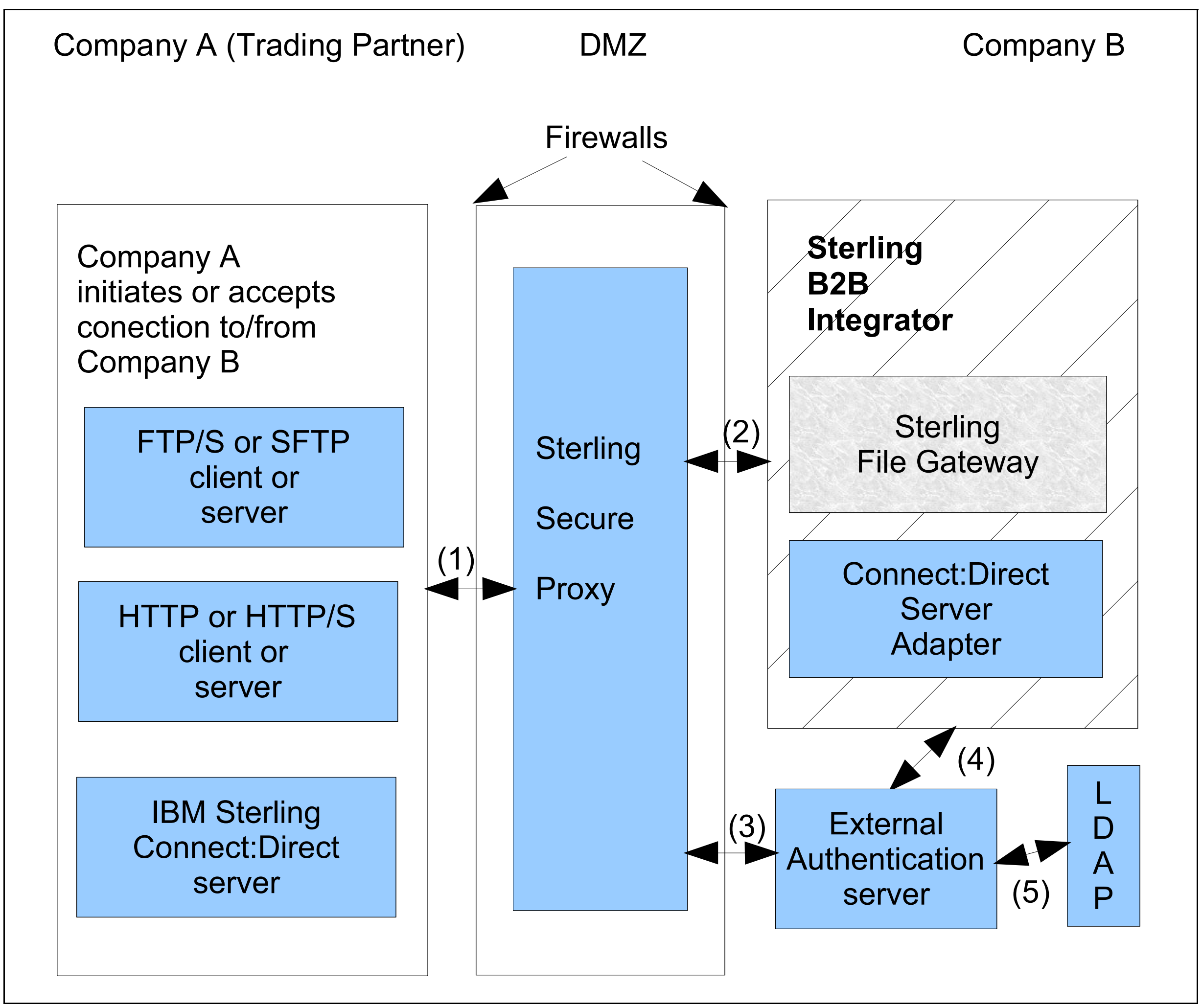


Figure 3-6 IBM Sterling File Gateway integration with IBM Sterling Secure Proxy

The following list explains the flow in this scenario:

1. Arrow (1): Any session initiated by external trading partners (Company A) reaching Sterling Secure Proxy. These sessions can be initiated by external trading partners (Company A) FTP/S, SFTP, HTTP or HTTP/s client or server or an IBM Sterling Connect:Direct server.
2. Arrow (2): Any session initiated by Company B) reaching Sterling Secure Proxy. These sessions can be initiated by Company B Sterling File Gateway FTP/S, SFTP, HTTP or HTTP/s client or server or an IBM Sterling Connect:Direct Server Adapter.
3. Arrow (3): Indicates use of an External Authentication server by Sterling Secure Proxy.
4. Arrow (4): Indicates use of an External Authentication server by Sterling B2B Integrator.
5. Arrow (5): Indicates use of LDAP by External Authentication server.

For more information, see *IBM Sterling Secure Proxy Planning and Installation* at:

http://help.sterlingcommerce.com/SSP34/index.jsp

IBM Sterling File Gateway integration with IBM Sterling Control Center

IBM Sterling Connect:Direct can be monitored by IBM Sterling Control Center, as shown in Figure 3-7.

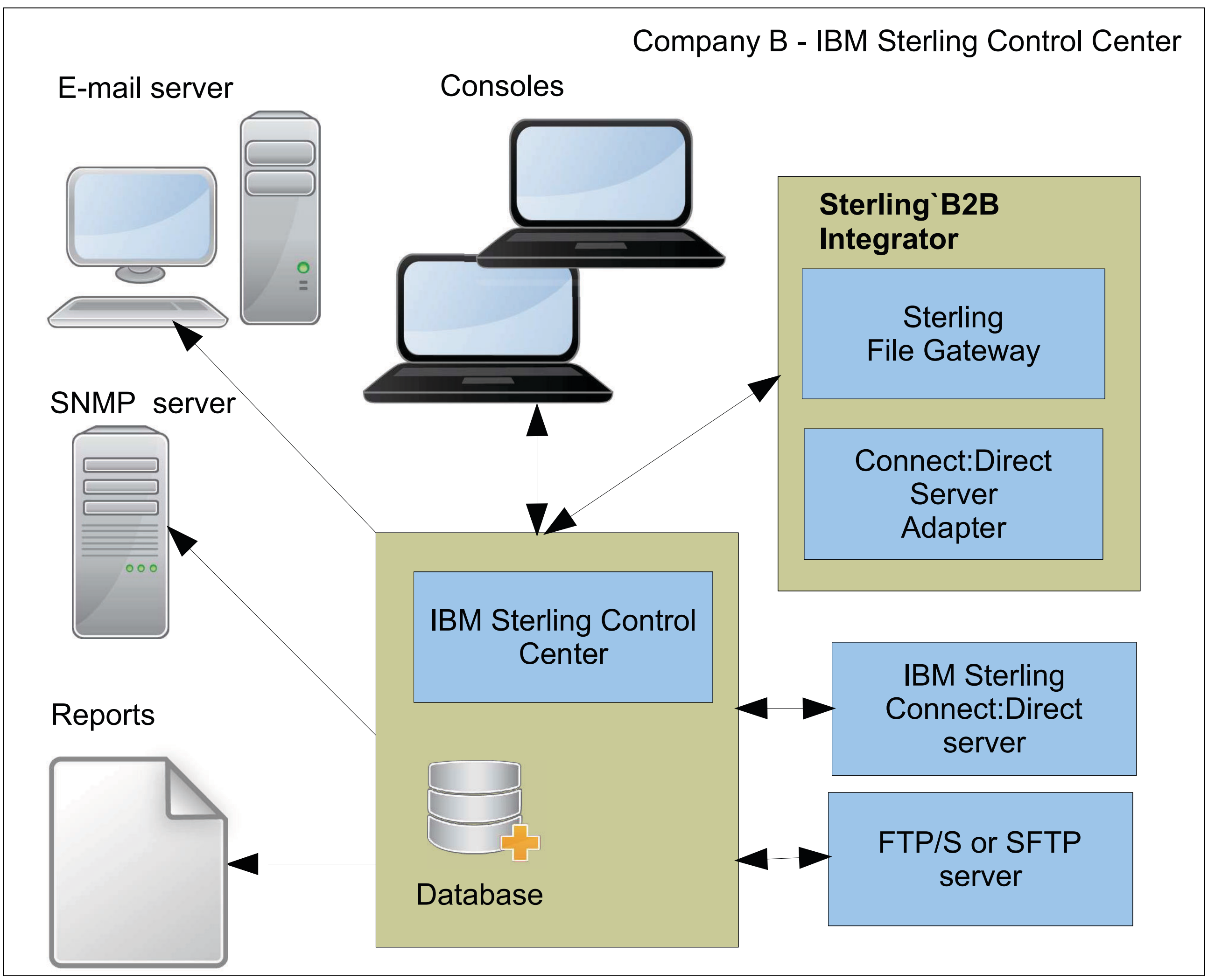


Figure 3-7 IBM Sterling File Gateway integration with IBM Sterling Control Center

For more information, see the *Monitoring with Sterling Control Center* manual at this website:

http://help.sterlingcommerce.com/SB2BI52/topic/com.ic.customization.doc/SI_52_PDF/SI52_SCC.pdf

3.3.2 IBM Sterling Connect:Direct

IBM Sterling Connect:DIrect provides a high performance file based information exchange solution. It can exchange high volume of files between several different platforms. IBM Sterling Connect:Direct is suitable for exchanging files of any sizes and any type that the underlying operating system supports. The number of files exchanged in a time period depends on the hardware resources available. IBM Sterling Connect:Direct can scale up to support larger amounts of data exchange without the need for an extensive reconfiguration.

IBM Sterling Connect:Direct is a peer-to-peer solution. An IBM Sterling Connect:Direct instance communicates to another IBM Sterling Connect:DIrect instance installed at any supported platform or to a Sterling Connect:Direct Server Adapter instance running under an IBM Sterling B2B Integrator.

IBM Sterling Connect:Direct supports the following features:

- File-based information exchange is assured: A proprietary protocol guarantees file integrity.
- Multi-sessions: Several sessions can take place with one or several remote IBM Sterling Connect:Direct instances simultaneously.
- Security: Exchanges occurs integrated with the underlying Operating System security.
- Compression: Compressing data improves performance on low speed lines.
- Encryption: Guarantees authentication, data integrity, and confidentiality.

For more information, see the *IBM Sterling Connect:Direct Product Overview* manual at this website:

http://www.sterlingcommerce.com/documentation/home/MFT/ConnectDirect/Common/CDOverview.pdf

IBM Sterling Connect:Direct also has integrations with two additional IBM products:

- WebSphere MQ File Transfer Edition
- WebSphere Message Broker

WebSphere MQ File Transfer Edition: Connect:Direct bridge

WebSphere MQ File Transfer Edition transfers files between systems in a managed and auditable way, regardless of file size or the operating systems used.

You can use WebSphere MQ File Transfer Edition to build a customized, scalable, and automated solution that enables you to manage, trust, and secure file transfers. WebSphere MQ File Transfer Edition eliminates costly redundancies, lowers maintenance costs, and maximizes your existing IT investments.

WebSphere MQ File Transfer Edition is built on WebSphere MQ, which provides assured, once-only delivery of messages between applications. You can take advantage of various features of WebSphere MQ. For example, you can use channel compression to compress the data that you send between agents over WebSphere MQ channels and use SSL channels to secure the data that you send between agents. Files are transferred reliably and can tolerate the failure of the infrastructure over which the file transfer is carried out.

From Version 7.0.4 of WebSphere MQ File Transfer Edition, you can transfer files to and from an existing IBM Sterling Connect:Direct network. Use the Connect:Direct bridge (Figure 3-8), which is a component of WebSphere MQ File Transfer Edition, to transfer files between WMQFTE and IBM Sterling Connect:Direct.

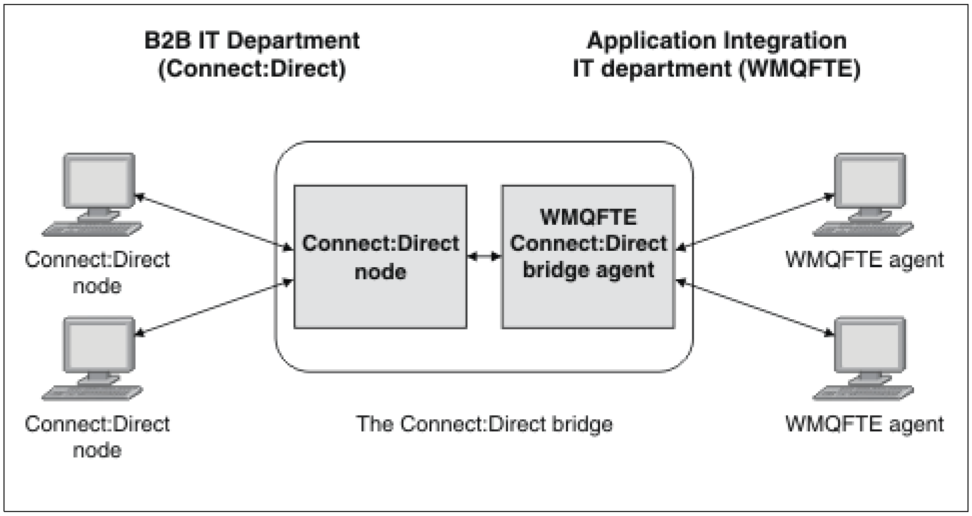


Figure 3-8 Connect:Direct bridge

Information about "The Connect:Direct bridge" can be found at the following website:

http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/topic/com.ibm.wmqfte.doc/cd_bridge_overview.htm

Information about "Configuring the Connect:Direct bridge" can be found at the following website:

http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/topic/com.ibm.wmqfte.doc/cd_bridge_configure.htm

WebSphere Message Broker: CD Nodes

WebSphere Message Broker support for IBM Sterling Connect Direct is introduced in WebSphere Message Broker 7.0.0.4 using APAR IC75621. More details about this APAR can be found at the following website:

http://www-01.ibm.com/support/docview.wss?uid=swg1IC75621

This feature will also be released in WebSphere Message Broker V8.

You can use the CDInput node in any flow that is designed to accept files from a IBM Sterling Connect:Direct network. You can use the CDInput node to extend WebSphere Message Broker support for file processing through its integration with IBM Sterling Connect:Direct.

On z/OS, when the CDInput node receives notification of the arrival of a dataset that it should process, the node copies that dataset into UNIX System Services temporarily, prior to processing.

The CDInput node is contained in the File drawer of the palette, and is represented in the WebSphere Message Broker Toolkit by the following icon (Figure 3-9):.

Figure 3-9 CDInput node

You can read more about CDInput node from the InfoCenter at the following website:

http://publib.boulder.ibm.com/infocenter/wmbhelp/v8r0m0/topic/com.ibm.etools.mft.doc/bc14020_.htm

You can use the CDOutput node to serialize the message tree to a file and then transfer it between two Connect:Direct servers. A directory under the work path within the execution group is used as the staging area, until the file is ready to be transferred.

After the file is transferred, it is deleted from the staging area. The CDOutput node is contained in the File drawer of the palette, and is represented in the WebSphere Message Broker Toolkit by the following icon (Figure 3-10):

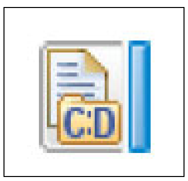


Figure 3-10 CDOutput node

You can read more about CDOutput node from the InfoCenter at the following website:

http://publib.boulder.ibm.com/infocenter/wmbhelp/v8r0m0/topic/com.ibm.etools.mft.doc/bc14015_.htm

3.3.3 IBM Sterling B2B Cloud Services

IBM Sterling B2B Cloud Services provide flexible solutions and expert services to optimize customers' dynamic business network and reduce the time and complexity of securely integrating with 100% of their business partners.

IBM Sterling File Transfer Service

IBM Sterling File Transfer Service is a cloud service for file-based business interactions with trading partners. As an alternative to on-premise software, IBM Sterling File Transfer Service allows customers to manage a single, secure, reliable connection to the cloud to reach their partners without the capital expense associated with on-premise software or the operational impact on their IT staff.

This offering fills a void in the portfolio by introducing a cloud service to replace on-premise FTP and home-grown solutions for exchanging of large, multi-gigabyte files with B2B partners. The features of IBM Sterling File Transfer Service are listed in Table 3-2.

Table 3-2 Service features

Capability	Description
Standards and protocols	Provides the following capabilities: ► Industry protocols: FTP, FTP/S, S/FTP, OFTP2 and AS2 ► Utilize Zip to compress files
Centralized monitoring and management	IBM Sterling InFlight Data Management visibility tool provides users: ► Reports and alerts for exception-based monitoring and event management ► Real-time views into file exchanges with trading partners for all files tracked from start to finish ► Timely and accurate audit trails for all file transfer activity ► Mobile access to search, view and track transfers
Security and governance	Provides the following capabilities: ► Connect to IBM Sterling File Transfer Service by a secure protocol, including IBM Sterling Connect:Direct ► PGP encryption secures files in-flight ► Single sign on user access ► Administrator IDs manage user access permissions and control screen level access ► Transfer logging provides a complete audit trail of file transfer activities
Scalability	IBM Sterling File Transfer Service scales with customers' demand with minimal impact on their resources and infrastructure: ► Flexible consumption models based on file sizes, volumes, and number of partners ► Process high volumes and large files within the capabilities for each protocol without compromising performance
File transfer automation	Provides the following capabilities: ► A single, secure connection with IBM Sterling File Transfer Service reaches partners ► Retry capabilities automate the replay and resend of failed transfers ► Intelligently route files based on sender, file name, and file type ► Protocol conversions in the cloud leverage partners' existing protocols
Outsourced services	Use IBM skilled personnel to manage customers' B2B file transfer environment with: ► Onboarding services to support their transition to IBM Sterling File Transfer Service ► The geographical coverage, language support, availability and level of expertise to meet their support needs, including phone, Web, and e-mail support ► Partner help desk services

IBM Sterling B2B Collaboration Network

IBM Sterling B2B Collaboration Network is a cloud platform for delivering B2B integration and visibility, enabling customers to rapidly connect, exchange, and manage business data with global partners. It acts as the communications link between customers and their business partners, and offers a growing list of on-demand application services.

IBM Sterling B2B Collaboration Network processes a variety of B2B integration requirements, provides flexibility to support varying needs, and improves speed to market for a range of hosted offerings, listed next:

- Exchange Services:

  Exchange Services enable customers to communicate securely with anyone, in any way, regardless of protocols, data formats, and preferred communication methods. Each customer has a single point of connection for business exchanges with many business partners, regardless of the type of communications protocol or data format required by each business partner.

- Translation Services:

  Translation Services enable customers to communicate with trading partners in any of the following translation relationships:

  - XML to EDI and EDI to XML
  - Proprietary data to EDI or XML
  - EDI or XML to proprietary data
  - EDI to EDI (including EDI data routed as e-mail attachments)
  - Flat-file layout to flat-file layout

- Community Services:

  Community Services interconnect to other networks and bridges to other exchanges and communities. Additionally, web-based services and fax services help customers integrate exchanges with trading partners who do not have EDI systems. The following are the available Community Services offerings:

  - IBM Sterling Web Forms
  - IBM Sterling Fax Conversion Services
  - IBM Sterling Managed AS2 Gateway
  - IBM Sterling Service Bureau for SWIFT
  - IBM Sterling Community Development Services

- Visibility Services:

  Visibility Services enable customers to see the process as their business data is routed through IBM Sterling B2B Collaboration Network to and from their trading partners. These services improve strategic decision making with real time visibility and alerts into customers' B2B operations. The following Visibility Services offerings are available:

  - IBM Sterling Document Tracking
  - IBM Sterling InFlight Data Management
  - IBM Sterling Report Visibility
  - IBM Sterling Supply Chain Visibility

- Application Services:

  Application Services provide specific industry and business functionality. The following Application Services offerings are available:

  - IBM Sterling e-Invoicing
  - IBM Sterling Transportation Management System
  - IBM Sterling Supplier Portal
  - IBM Sterling Data Synchronization Manager

IBM Sterling Web Forms

IBM Sterling Web Forms is a customizable website operated by IBM that enables customers' trading partners to create and exchange business documents using the Internet. It helps customers integrate smaller trading partners, who do not have EDI capabilities, into their electronic trading community.

To replace paper business documents with electronic commerce, customers sponsor Web Forms communities that allow their smaller partners to receive, complete, and send electronic business documents from IBM Sterling Web Forms website. Web Forms communities function as follows:

- A Web Forms community consists of a large company that sponsors the service and its smaller business partners that are members of the community.
- The sponsor sends a data file, in either EDI or XML format, to IBM Sterling B2B Collaboration Network, which routes the document to IBM Sterling Web Forms system. Web Forms translates the data format into the Web Forms proprietary format, and posts it on the Web Forms site for the member to retrieve.
- Members use a web browser to access the Web Forms site and complete an HTML-based document, also called a Web Form.
- Sponsors use a browser to access the Sterling Web Forms administration site, where they can create messages for their members, generate reports about their community's activity, change login information given to new members, and so on.

3.3.4 IBM Sterling Control Center

IBM Sterling Control Center provides centralized management, monitoring and notification for several IBM Sterling products:

IBM Sterling Control Center monitoring and notification

IBM Sterling Control Center monitors operations on IBM Sterling products and issue real-time notifications in case any problem is detected,

Some benefits of IBM Control Center monitoring/notification functions are as follows:

- No processing exception goes undetected, thus preventing business impacts.
- It helps processing windows to be met.
- It avoids situations where a Service Level Agreement (SLA) is not respected.

Here are the main examples of the operations monitored by IBM Sterling Control Center:

- IBM Sterling Connect:Direct Server Status
- IBM Sterling Connect:Direct processes activities
- IBM Sterling Connect:Direct File Agent File Agent Service Initialization Error
- Process not submitted by File Agent within a time period
- IBM Sterling B2B Integrator Perimeter Server Status
- IBM Sterling B2B Integrator Adapter Status
- IBM Sterling B2B Integrator Business Process activities
- IBM Sterling B2B Integrator AFT File Transfer Activities
- IBM Sterling B2B Integrator Sterling File Gateway (SFG) activities
- File Transfer Protocol (FTP) activities

- IBM Sterling Control Center can notify when an IBM Sterling Connect:DIrect Secure+ certificate is about to expire.
- IBM Sterling Control Center can notify when and IBM Sterling Connect DIrect license is about to expire.

For more information, see the Sterling Integrator manual, *Monitoring with Sterling Control Center,* at this website:

http://help.sterlingcommerce.com/SB2BI52/topic/com.ic.customization.doc/SI_52_PDF/SI52_SCC.pdf

IBM Sterling Control Center Service Level Criteria

Another important feature of IBM Sterling Control Center is the Service Level Criteria (SLC). SLCs define an event that should occur in a time period. For instance, an IBM Sterling Connect:DIrect process is expected to be executed every Monday between 03:00PM and 04:30PM. If process execution does not occur in that time window, an SLC can generate a notification so that measures could be taken to fix the problem.

For more information, see the *IBM Sterling Control Center System Administration Guide* at this website:

http://www.sterlingcommerce.com/documentation/SCC53/PDF/SystemAdministration.pdf

IBM Sterling Control Center Notification

IBM Sterling Control Center notifications can be made available in the following ways:

- IBM Sterling Control Center consoles. There is a local console, and remote consoles run under a web browser.
- SMTP (e-mails) can be sent to administration or operation users.
- SNMP traps can be sent to a suitable application.
- Emits JMS to IBM MQ Broker, which can be accessed by WebSphere.

Each user has restricted access to some events only.

4

Routing and transforming messages

This chapter describes the components used to integrate message based routing and transformation in the scenarios demonstrated in Part 2, “Scenarios based on the proposed solution architecture” on page 101. In addition to describing the components used, we discuss recommended practices, alternative methods, and alternative products.

We also describe how WebSphere Transformation Extender, WebSphere Message Broker, WebSphere DataPower Appliance, and Sterling B2B Integrator can all work together to accomplish functional requirements to route, transform, and translate messages from source to target, regardless of the protocols and formats that make up the environment.

This chapter includes the following sections:

4.1 What are the business requirements?

What a business requires in terms of transformation and routing is always based on the kind of business being done. Requirements can depend on these factors:

- Nature of existing services
- Kind of data to be transferred
- Communication protocol
- Industry standards
- Business monitoring and intelligent routing

Before we proceed further, we need to understand the meaning of *mediation*. Simply put, mediation can mean either of the following types of intervention:

- An intervention in a dispute in order to resolve it
- An intervention in a process or relationship

In the process of mediation, it is required that the parties involved are willing to communicate and negotiate with each other. Then it becomes the responsibility for a mediator to enable mediation between the involved parties.

In a real world scenario, assume that some business is being conducted between Chinese and American parties. The Chinese business representative only speaks and understands Chinese, and the American representative only speaks and understands English. For both of them to directly communicate with each other is next to impossible, so they would need a translator who can perform language translation or *mediation* between the two business representatives.

Figure 4-1 illustrates the language mediation task performed by the language translator.

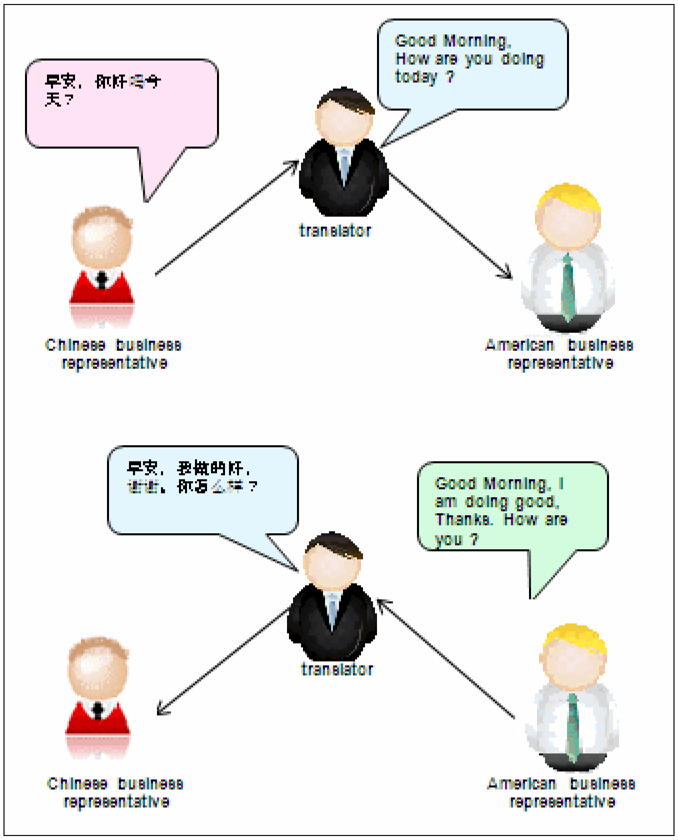


Figure 4-1 Language mediation by translator

If we replace business representatives by software services, which need to communicate with each other for business integration, there is a need for a software mediation module to help integrate these business services. The mediation module needs to enable this integration with the following features:

- Content transformation
- Communication protocol switching
- Message routing

Figure 4-2 illustrates the mediation module components.

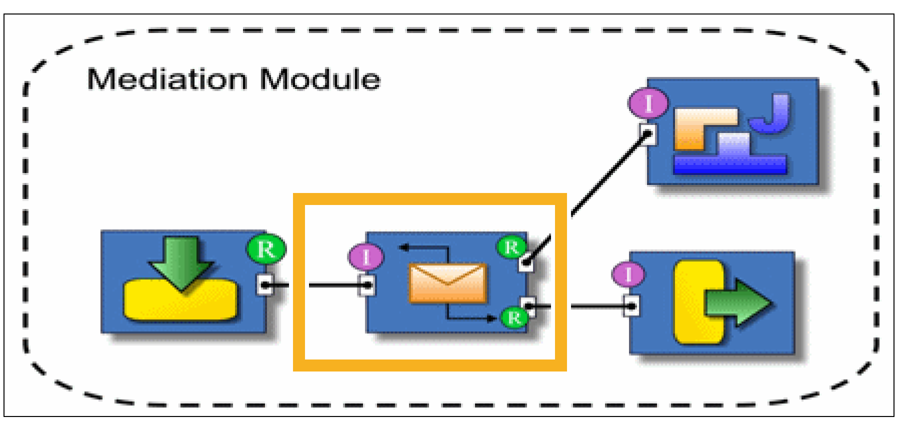


Figure 4-2 Software mediation module

To illustrate the responsibilities of the mediation module, we use an example of a simple mediation that provides stock quotes. A client application provides a query containing a stock symbol and customer ID to the mediation module which processes the query. The customer's subscription level is determined, and depending on the level of subscription, the query is routed to the appropriate service provider. The quote that is returned from the service provider is converted into the customer's preferred currency before it is returned to the client application.

We are using a mediation module because we want to use different interfaces from two external service providers, and expose a single interface to the client application. We need to build the service quickly with the ability to change the application on demand, and without modeling a business process. We also want the ability to change the service provider without disrupting the service. Figure 4-3 shows the complete mediation.

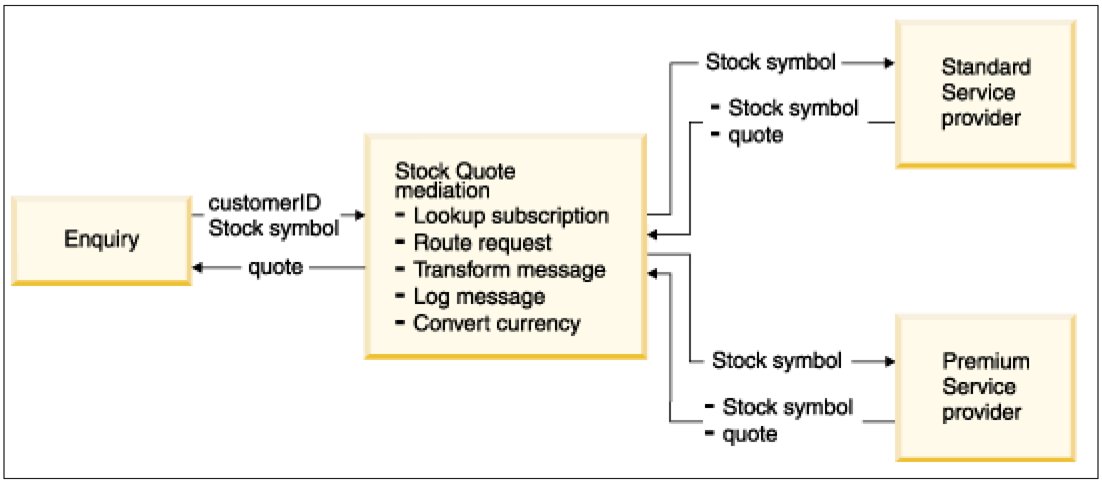


Figure 4-3 Stock quotes mediation module

4.2 Message based integration choices

In this section, we discuss message based integration patterns. More details about these patterns can be found at the following websites:

https://www.ibm.com/developerworks/wikis/display/esbpatterns/Message-based+Integration

http://www.ibm.com/developerworks/library/ws-enterpriseconnectivitypatterns/index.html

Mediation infrastructure can help an existing messaging infrastructure by providing an environment for building and deploying “infrastructure level” message based “applications”. Examples of such “applications” include routing and transformation services. In a message oriented model, we are typically much more focused around the data that is flowing through the system and the set of actions that are applied to this in-flight data; that is, we have more of a producer-to-consumer view. While it is not an absolute distinction, this line of thinking can influence the context in which we seek to apply each pattern listed next:

- Message router pattern
- Message translator pattern
- Message bridge pattern
- Message aggregator pattern

- Message splitter pattern
- Message request/response correlator
- Message router pattern

The message router pattern can be used to provide a strong level of decoupling between applications or services which need to exchange data by enabling data sent from one application to be routed to one of several potential target applications based on various conditions.

Context-based routers select an appropriate target based on the identity of the sender or on some aspect of the data carried in a protocol header. Content-based routers make a selection based on the content of the message payload. Load-based routers use information about the application load on the various target systems.

4.2.1 Message translator pattern

The message translator pattern allows data from one application to be mapped into the data format required by another application without either application being aware that such mapping is needed or taking place. This pattern covers anything from direct mapping to highly complex transformations possibly involving lookups and cross references.

4.2.2 Messaging bridge pattern

The messaging bridge pattern maps data from one transport mechanism to another without modifying the format or content of the message payload. An implementation of this pattern must also handle the mapping between the different addressing schemes which might be used by the separate messaging systems.

A frequent example of the use of this pattern is the bridging between JMS implementations from different vendors.

4.2.3 Message aggregator pattern

The message aggregator pattern addresses the need to take multiple messages from one or more applications and merge them into a single piece of data to be propagated as a new message. The inbound messages can come from independent applications or an be asynchronous response messages from a set of applications which received requests from an implementation of the message splitter pattern (described below in “Message splitter pattern” on page 73).

An implementation of this pattern can simply concatenate the individual source messages or can incorporate some more sophisticated set of data mapping capabilities. It must also be able to handle the failure of an expected inbound message to arrive within a prescribed time period and the subsequent late arrival of this message.

4.2.4 Message splitter pattern

The message splitter pattern extracts subsets of a message which are then sent as separate messages to multiple target applications. A set of mapping rules define how the original inbound message is to be broken apart into constituent pieces.

4.2.5 Message request/response correlator pattern

A typical message broker system is often processing multiple requests from the same stream or queue in a parallel, asynchronous manner. When such requests are being routed to a further system (or systems), there is a requirement to correlate any response messages to the original request. The message request/response correlator pattern provides a solution to this particular problem.

4.3 Product information

In this section we discuss the IBM products used for routing and transforming the messages.

4.3.1 WebSphere Message Broker

WebSphere Message Broker is a platform-independent based ESB that provides universal connectivity. It can be used to integrate disparate applications and is designed to transform various formats of data between any type of applications using a number of supported communications protocols or distribution methods. It is used where there is a need for high-performance and complex integration patterns.

WebSphere Message Broker V8.0 offers simplicity and productivity in terms of developing and managing the WebSphere Message Broker environment. WebSphere Message Broker plays a critical role in SOA and offers a wide range of SOA scenarios in which it can be integrated. The dynamic operational management of WebSphere Message Broker enables administrators to effectively understand and modify broker behavior, which thus enables them to respond quickly to business requirements. WebSphere Message Broker is supported on a large range of platforms and environments.

Installing and configuring WebSphere Message Broker V8.0 is significantly simplified with the removal of the Configuration Manager, the User Name Server components, and the removing the requirement for a system database.

For additional information about these changes, go to the following website:

http://publib.boulder.ibm.com/infocenter/wmbhelp/v8r0m0/topic/com.ibm.etools.mft.doc/bb23110_.htm

Processing logic in WebSphere Message Broker is implemented using message flows. Through message flows, messages from business applications can be transformed and routed to other business applications. Message flows are created by connecting nodes together. A wide selection of built-in nodes are provided with WebSphere Message Broker. These nodes perform tasks that are associated with message routing, transformation, and enrichment. The base capabilities of WebSphere Message Broker are enhanced by SupportPacs that provide a wide range of additional enhancements.

Message routing

Routing a message involves sending an incoming message to a destination that is based on a criteria. The destination can be predefined (static) or based on information that is obtained at the time of the message flow (dynamically). A common routing pattern includes a dynamic lookup of the destination based on the incoming message type and routing the message to that destination. This routing pattern typically consists of the steps shown in Figure 4-4.

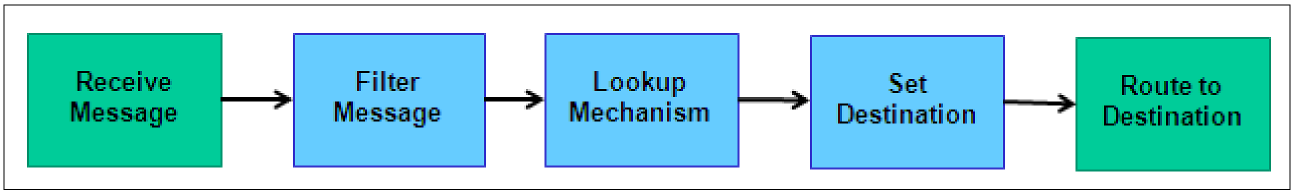


Figure 4-4 A typical routing mechanism

Packaged with WebSphere Message Broker are a variety of nodes through which connectivity is provided for both standards and non-standards-based applications and services. Routing can be point-to-point or based on matching the content of the message with a pattern that is specified in a node.

Aggregation is an advanced form of message routing. With aggregation, a request message is received, and multiple new request messages are generated. Each new message is routed to its destination using a request-reply interaction. WebSphere Message Broker tracks the process, collecting each response and recomposing them into a single output message.

Figure 4-5 shows the nodes available in the WebSphere Message Broker palette for enabling message based routing.

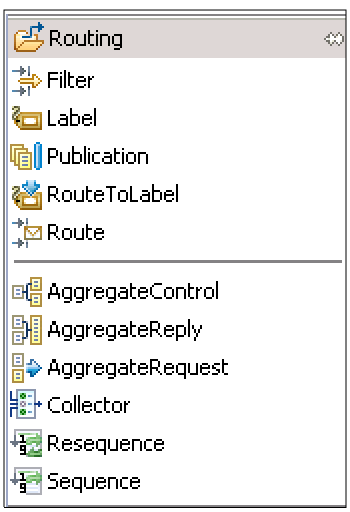


Figure 4-5 Routing nodes in WebSphere Message Broker

Routing can also be performed programmatically using Compute Node, Java Compute Node, PHP Node, and .Net Compute node.

Message transformation

One of the key capabilities of WebSphere Message Broker is the transformation and enrichment of in-flight messages. This capability enables business integration without the need for any additional logic in the applications, for example, an application that generates messages in a custom format can be integrated with an application that only recognizes XML. This capability provides a powerful mechanism to unify organizations because business information can now be distributed to applications that handle completely separate message formats.

In WebSphere Message Broker, message transformation and enrichment are dependent upon a broker understanding the structure and content of the incoming message. Self-defining messages, such as XML messages, contain information about their own structure and format. However, before other messages, such as custom format messages, can be transformed or enhanced, a message definition of their structure must exist. The WebSphere Message Broker Toolkit contains facilities for defining messages to the WebSphere Message Broker.

Using parsers and message sets, WebSphere Message Broker can validate and check that incoming messages comply with the format that is defined in the message set. A flow can be constructed to reject and handle non-compliant messages. Additionally, complex manipulation of message data can be performed using extended SQL (ESQL), Java, and PHP facilities, which are provided in the WebSphere Message Broker Toolkit.

WebSphere Transformation Extender can be integrated into the WebSphere Message Broker ESB solution to extend the existing capabilities and to simplify transformation development.

Applications typically use a combination of messages, which includes those that are defined by the following structures or standards:

- C and COBOL data structures
- Industry standards such as X12, ACCORD AL3, EDIFACT, SWIFT, EDI or HL7
- XML DTDs or schemas
- SOAP
- CSV
- IDoc
- User Defined

Mediation flows must be able to transform a message from one format to another with acceptable throughput. Messages in WebSphere Message Broker can be transformed using one of the following ways:

- Compute node (ESQL)
- XSLTransform node (Extensible Stylesheet Language Transformations)
- Mapping node (graphical)
- JavaCompute node (Java)
- Runtime message set and definition
- PHPCompute node (PHP)
- WebSphere TX node

Along with the nodes in the *Transformation* tab, an additional WTX node is also available in the WebSphere Message Broker toolkit, after the IBM WebSphere Transformation Extender Design Studio is installed on the same development workstation. Figure 4-8 on page 80 illustrates the WebSphere TX node in the WebSphere Message Broker toolkit.

When a message arrives from a transport protocol wired to the message flow runtime, the message bit stream is parsed using a physical format, such as XML. See Figure 4-6.

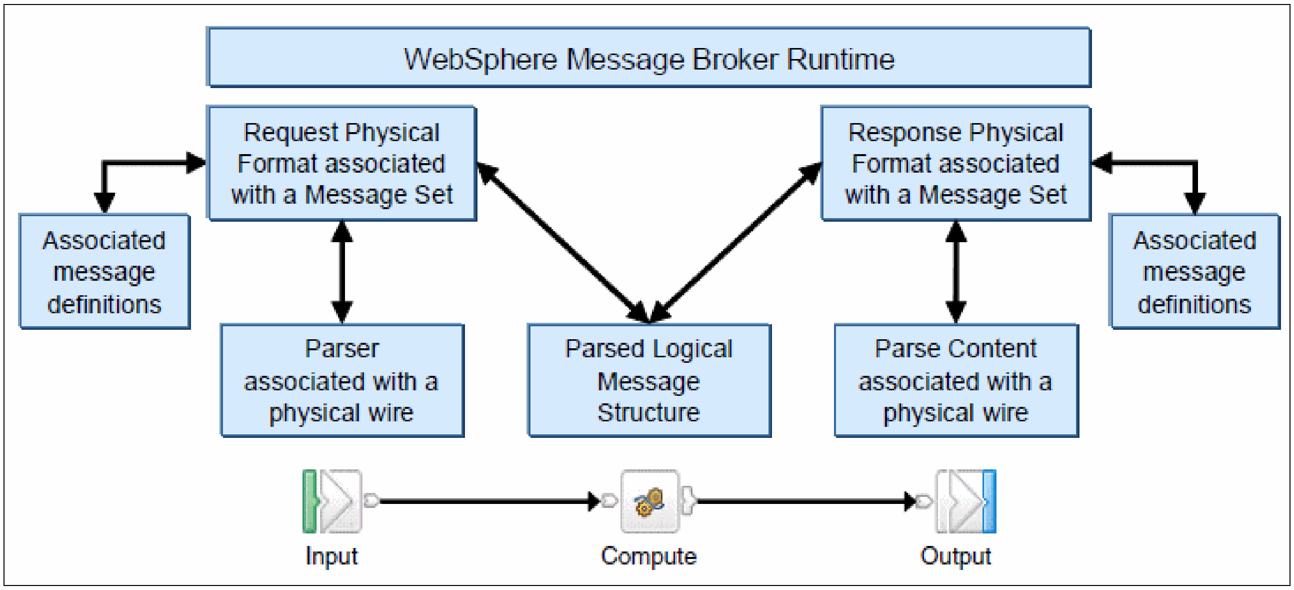


Figure 4-6 WebSphere Message Broker runtime architecture

When the message format is known, the broker parses an incoming message bit stream using the message set and definition that are defined on the flow configuration and converts it into a logical message tree for later manipulation. After the message is processed by the message flow, the broker converts the message tree back into a message bit stream. The transformation includes reformatting the message, concatenating the strings, or changing the element values.

The following physical formats are supported by the broker run time:

- XML:

  This format is the default runtime configuration. The message structure is validated and transformed using the parser specification that is defined inside the message flow.

- Text (TDS):

  The Text or Tagged Delimited String (TDS) physical format is designed to model messages that consist only of text strings. Examples of TDS messages are those that conform to the ACORD AL3, EDIFACT, HL7, SWIFT, and X12 standards. The TDS physical format allows a high degree of flexibility when defining message formats and is not restricted to modeling specific industry standards. Therefore, you can use the TDS format to model your own messages.

- Binary (CWF):

  The Binary or Custom Wire Format (CWF) is a physical representation of a message that is composed of a number of fixed format data structures or elements, which are not separated by delimiters. The CWF physical format is typically used to describe messages that are mapped to a C structure, a COBOL copybook, or any other programming language data structure definition.

Message set: A message set can have one or more physical formats on each XML, TDS, and CWF format.

WebSphere Message Broker typically supplies a range of parsers to parse and write message formats. Some message formats are self-defining and can be parsed without reference to a model. An example of a self-defining message format is XML. In XML, the message itself contains metadata and data values, enabling an XML parser to understand an XML message even if no model is available.

Most message formats, however, are not self-defining. That is, a binary message that originates from a COBOL program and a SWIFT formatted text message do not contain sufficient metadata to enable a parser to understand the message. The parser must have access to a model that describes the message to parse it correctly.

Table 4-1 lists the supported parsers for WebSphere Message Broker.

Table 4-1 WebSphere Message Broker parsers

Parser name	Parser description
MRM	For modeling a wide range of messages, including XML, fixed-format binary, and formatted text. This domain is usually used for enrichment of the message transformation; for example, creating COBOL copybook using CWF.
XMLNSC XMLNS XML	For messages that conform to the W3C XML standard. The input bitstream must be a well-formed XML document that conforms to the W3C XML specification. XMLNSC is the preferred domain for generic XML messages, and messages using XML namespaces.
DataObject IDOC	Used for messages going in and out of WebSphereAdapter nodes. The DataObject parser parses the business objects that are received from EIS. It is guided by XML schemas that model the EIS business objects. IDOC is implemented by using the SAP BAPI and the IDoc SAP format.
MIME	For handling multipart MIME messages, such as SOAP with attachments or RosettaNet.
BLOB	A parser that is used with messages that do not need to be interpreted in a logical message tree. The run time internally interprets the message as a BLOB bit stream. It is commonly used with messages that do not have a well-defined element tree structure, such as the ISO8583 standard.
JMSMap JMSStream	For modeling messages that are produced by the implementations of JMS standard. JMSMap domain can be used when handling JMS messages of type MapMessage. JMSStream domain can be used when handling JMS messages of type StreamMessage.
SOAP	Creates WSDL-based logical tree format to work with Web Services and validates incoming messages against a WSDL definition.

Message models

Message modeling is a way to predefine the message formats that applications use. WebSphere Message Broker uses message models to automatically parse and write message formats.

The components of a message model are as follows:

- Message set projects
- Message sets
- Message definition files
- Message categories

The majority of the model content is described by message definition files. These files use the XML schema to represent the messages. Each message definition file describes both the logical structure of the messages and the physical format or formats that describes the appearance of the message bit stream during transmission.

If you are using the MRM domain, you must provide physical format information. This information tells the MRM parser how to parse the message bit stream. If you are using one of the XML domains, physical format information is not needed.

However, if your messages are self-defining and do not require modeling, there are still advantages to modeling them:

- Enhanced parsing of XML messages:

  Although XML is self-defining, without a model all data values are treated as strings. If a model is used, the parser knows the data type of data values and can cast the data accordingly.

- Improved productivity when writing ESQL:

  When you create ESQL programs for WebSphere Message Broker message flows, the ESQL editor can use message models to provide code completion assistance.

- Drag-and-drop message maps:

  When you create message maps for WebSphere Message Broker message flows, the Mapping editor uses the message model to populate its source and target views. Without message models, you cannot use the Mapping editor.

- More efficient way to implement websites:

  PHP is a dynamic scripting language that is most frequently used to implement websites. PHPCompute node provides the ability to transform and route messages in WebSphere Message Broker V8.0.

- Runtime validation of messages:

  Without a model, it is not possible for a parser to check that input and output messages have the correct structure and data values.

- Reuse of message models in whole or in part can be done by creating new messages based on existing messages.
- Automatic generation of documentation.
- Provision of version control and access control for message models by storing them in a central repository.

Tip: When creating output messages, the MRM parser can automatically generate the XML declaration and other XML constructs based on options in the model, which simplifies the transformation logic. For more information about when to use MRM or XMLNS(C) domains, search on “Which XML parser should you use” in the WebSphere Message Broker Information Center at the following website:

http://publib.boulder.ibm.com/infocenter/wmbhelp/v8r0m0/index.jsp?topic=/com.ibm.etools.mft.doc/ad70530_.htm

Message transformation nodes

After a logical message structure is created, the implementing flow has full access to the parsed elements inside of the message. In this sections, we describe the nodes that can be used to transform a message within a mediation flow. Figure 4-7 illustrate nodes available in WebSphere Message Broker palette to perform message transformation.

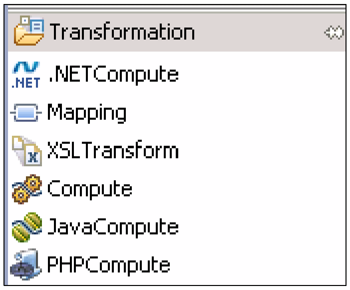


Figure 4-7 Transformation nodes in WebSphere Message Broker

Figure 4-8 illustrates the WTX Map node for message transformation. WTX Map node uses WTX maps for message transformation. WTX maps can be deployed on execution group or can be used from an external location.

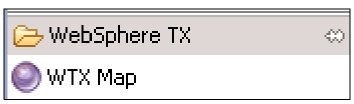


Figure 4-8 WTX Map node

Using WTX nodes for tranformation: There are multiple nodes available in WebSphere Message Broker for routing and transformation of business data. A list of these nodes was also discussed in 4.3.1, “WebSphere Message Broker” on page 74.

As a best practice, it is recommended to use WTX node for tranformation, with WTX Industry standard packs. WTX offers multiple industry standard packs, for example:

- WebSphere Transformation Extender pack for ACORD
- WebSphere Transformation Extender pack for HL7
- WebSphere Transformation Extender pack for HIPAA
- WebSphere Transformation Extender pack for SEPA
- WebSphere Transformation Extender pack for EDIFACT
- WebSphere Transformation Extender pack for SWIFT
- WebSphere Transformation Extender pack for X12

To read more details about WTX node, see the following websites:

WTX Map node overview:

http://publib.boulder.ibm.com/infocenter/wtxdoc/v8r2m0/index.jsp?topic=/com.ibm.websphere.dtx.wtx4wmb.doc/concepts/c_wtx4wmb_node_overview.htm

Designing, developing, and deploying WebSphere Transformation Extender:

http://www.ibm.com/developerworks/websphere/library/techarticles/0907_gupta/0907_gupta.html

Chapter 5, “Health Insurance scenario” on page 103 illustrates usage of WTX node in WebSphere Message Broker with WebSphere Transformation Extender pack of HIPAA.

4.3.2 WebSphere Transformation Extender

WebSphere Transformation Extender is an Enterprise Application Integration tool used for translation and transformation of messages from any source format to any target format, and allows messages from one or many sources to be translated and transformed to one or many targets at the same time. In this section, we discuss the WebSphere Transformation Extender products that we use to implement the scenarios in Part 2 of this book, as well as the methods and practices we used to implement the components of the solution where WebSphere Transformation Extender is deployed.

WebSphere Transformation Extender Design Studio

WebSphere Transformation Extender Design Studio is the development tool used to design and develop the maps used for transformation and translation tasks. We do not get into details of development, because the purpose of this book is integration, but we do describe a few features of the Design Studio that accelerate development, testing, and delivery of the maps that you need in your solution.

After you have developed and tested your map on your local platform, you need to do preliminary testing in the target environment with your target integration software. The Design Studio makes testing your map on the target very simple in the case of Sterling B2B Integrator and the WebSphere DataPower Appliance. WebSphere Message Broker requires more effort, because the ability to test on the broker from the Design Studio is not built in.

Testing a map on the Sterling B2B Integrator server

"WebSphere Transformation Extender for Integration Servers" must be installed before you can execute a map on the Sterling B2B Integrator server. The Design Studio gives you the ability to test and deploy maps from within Design Studio on the server as well, as long as you have "WebSphere Transformation Extender for Integration Servers" installed on the development platform in addition to the server. It is not required that you deploy a map to the server before you are able to test it on the server from Design Studio.

In order to test or deploy a map from the Design Studio on the server where Sterling B2B Integrator is installed, you must first configure the connectivity settings for the Test Map service and the Check In map service in the Design Studio. These two services can share the same connectivity settings. It is done by navigating to **Window** → **Preferences** → **Transformation Extender** → **Map** → **Sterling B2B Integrator.** (Figure 4-9). You would then populate the information. If you need assistance on how to configure the Design Studio for the appliance, help is available by navigating to the WebSphere Transformation Extender help window.

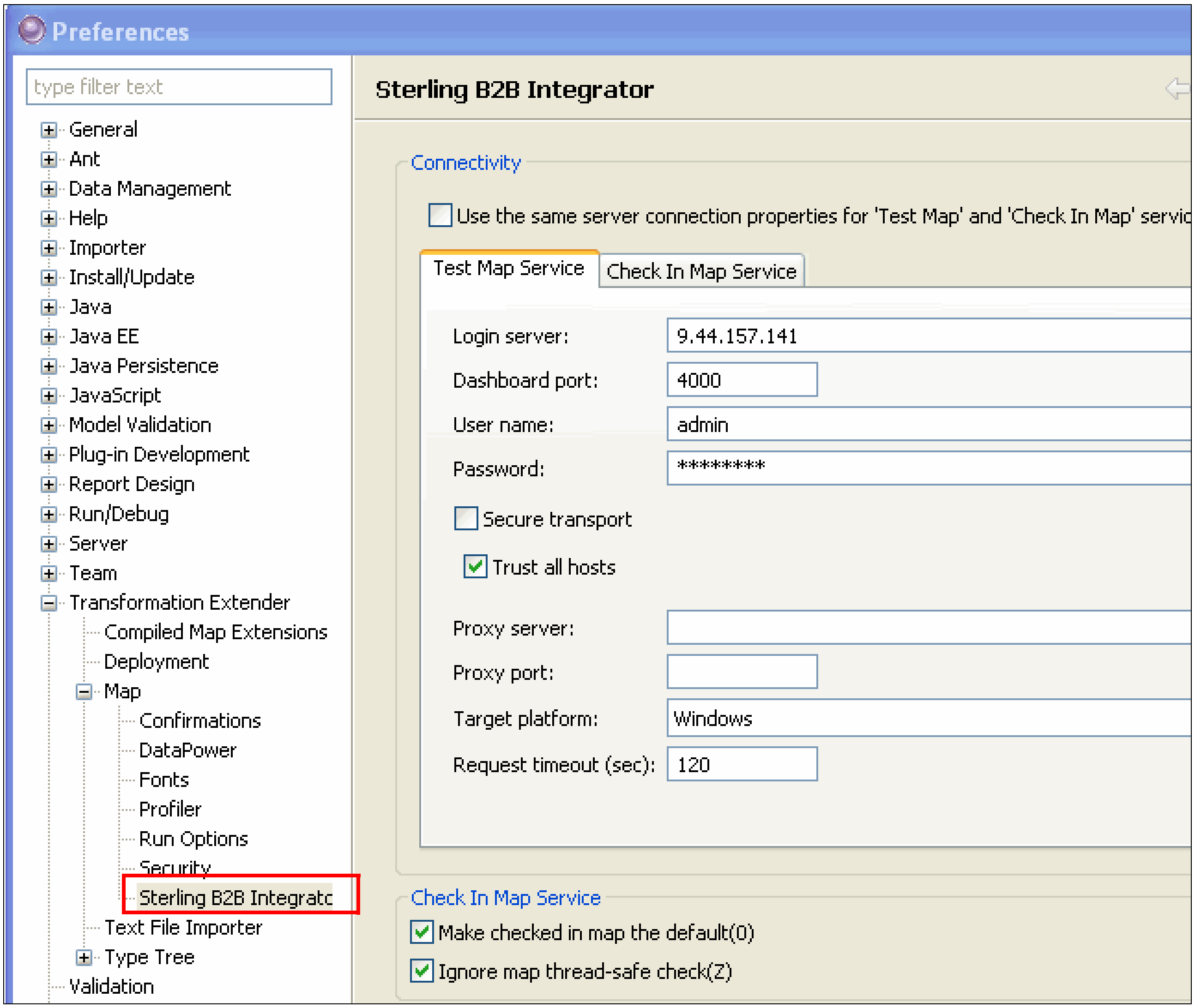


Figure 4-9 WebSphere Transformation Extender Sterling B2B Integrator preference

After the configuration has been completed, you can then execute the map on the server from Design Studio by either right-clicking the executable map in the Composition or Outline view and then selecting **Run on Sterling B2B Integrator** or by clicking the executable map and using the key combination “Ctrl+Shift+R”.

You can deploy the map by performing the same steps as described for executing the map, except that you would choose **Deploy to Sterling B2B Integrator** or use the key combination “Ctrl+Shift”D”. See Figure 4-10.

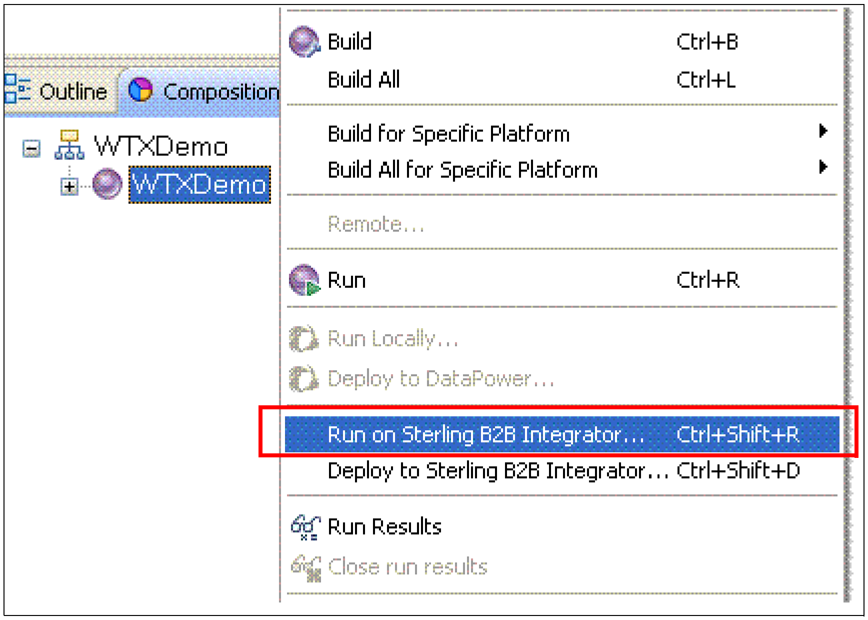


Figure 4-10 Test or deploy map on Sterling B2B Integrator server

Testing a map on the WebSphere DataPower Appliance

Integration: The integration of WebSphere Transformation Extender with the WebSphere DataPower Appliance, discussed in this section, describes the integration available in WebSphere Transformation Extender v8.3.x.

At the time this book was written, WebSphere Transformation Extender v8.4.0 did not expose the WebSphere DataPower Appliance integration capability. Integration similar to what is discussed, along with extended capabilities, will be available in v8.4.x when v5.0 of the WebSphere DataPower Appliance is released.

“WebSphere Transformation Extender for Integration Servers” is not required to be installed in order to test or deploy maps on the WebSphere DataPower Appliance. The appliance ships with the runtime in the firmware, and the Design Studio has the ability to test the map on the appliance, and deploy to the appliance built in.

In order to test or deploy a map on the WebSphere DataPower Appliance, you must configure the connectivity settings for the TX-Test XML Firewall Service and the XML Management Interface Service in the Design Studio. You do it by navigating to **Window** → **Preferences** → **Transformation Extender** → **Map** → **DataPower.** (Figure 4-11). You then populate the information. If you need assistance on how to configure the Design Studio for the appliance, help is available by navigating to the WebSphere Transformation Extender help window.

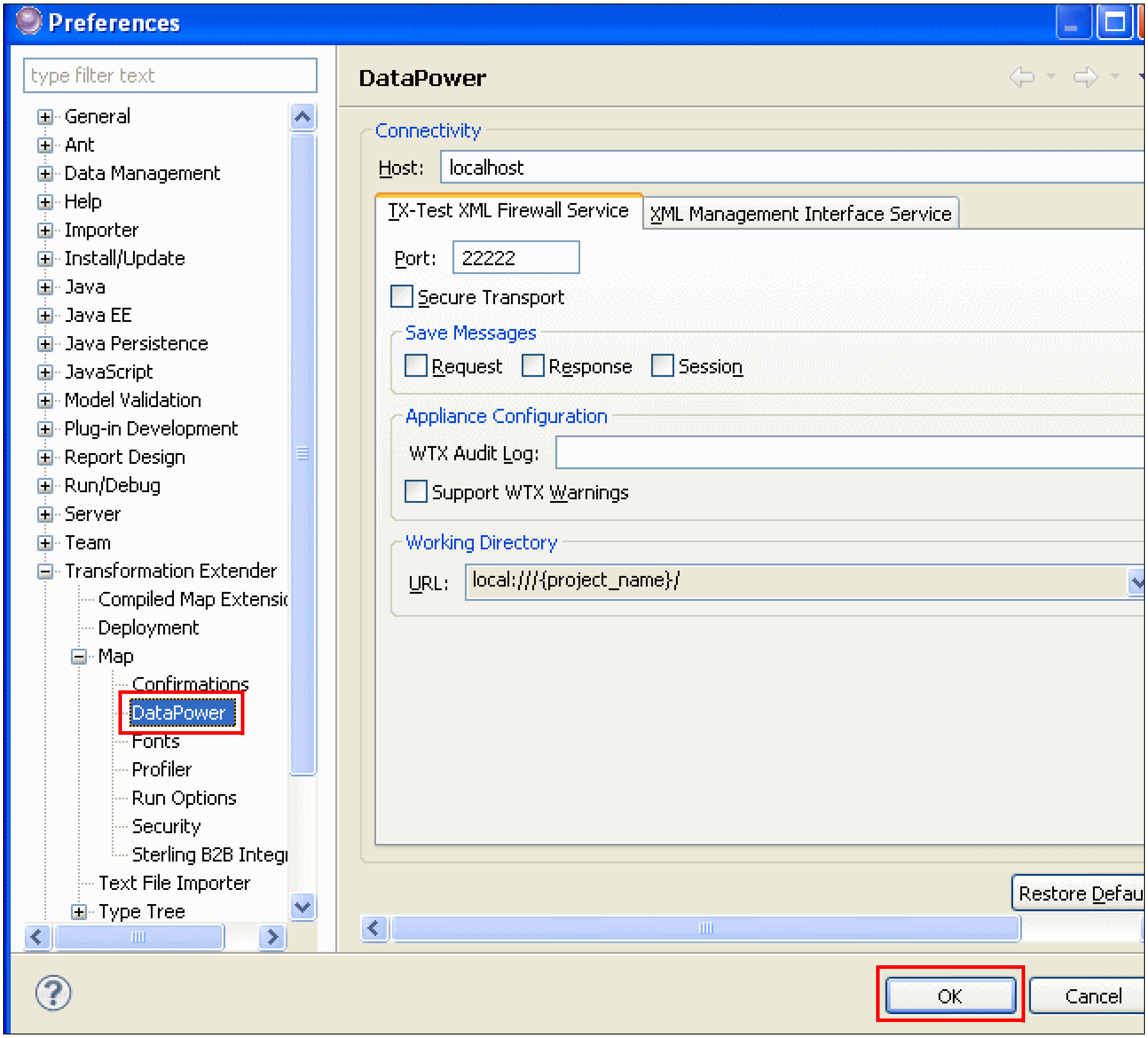


Figure 4-11 WebSphere Transformation Extender DataPower preference

After the configuration has been completed, you can then execute the map on the appliance from the WebSphere Transformation Extender Design Studio. However, only maps built to be executed on the appliance can be tested this way. The reason for it is that not all capabilities available on other server platforms are available on the appliance. You build a map for the appliance by navigating to Map Settings and setting the “MapRuntime” attribute of the map to WebSphere DataPower through the drop-down list. See Figure 4-12.

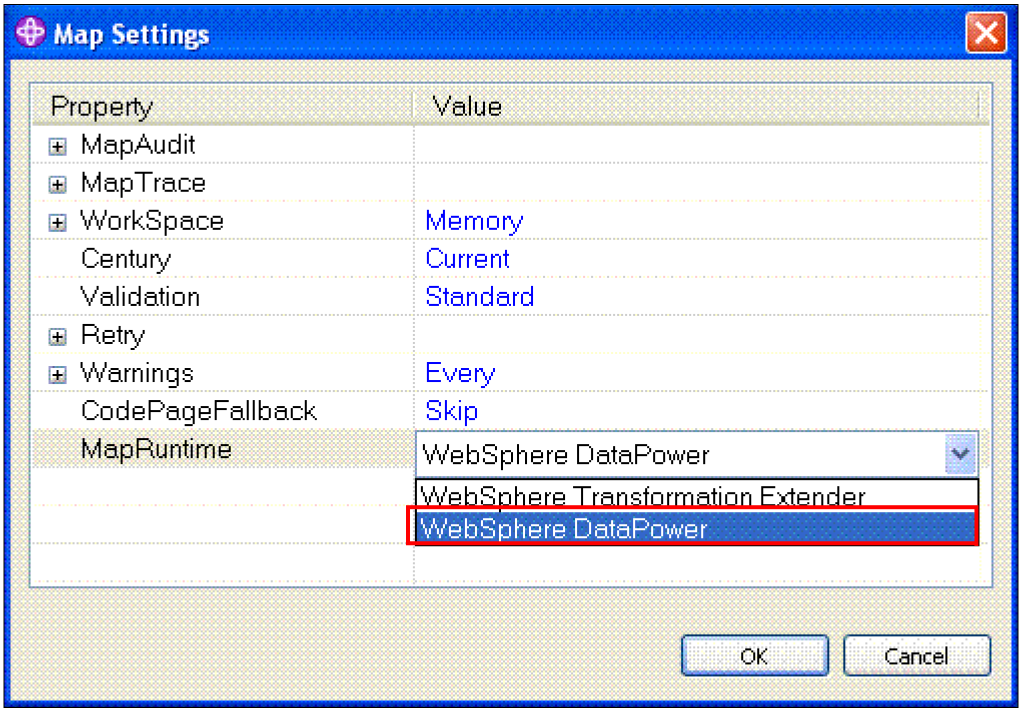


Figure 4-12 Design Studio map setting to build map for WebSphere DataPower Appliance

After it is done, you would build the executable map as you normally would, and the build process would create a map specifically built to run on the appliance, with the extension `.dpa`. After the map has been built, you can test it locally, test it on the appliance from the Design Studio, or deploy it to the appliance. You test locally by selecting **Run Locally** from the context menu or by right-clicking the executable map in the Composition or Outline view. Selecting this option does not execute the map on the appliance.

You can test the appliance by clicking the normal **Run** icon after the map has been built for the appliance and the connectivity settings have been configured. Finally, you can deploy the map to the appliance by selecting **Deploy to DataPower** from the context menu by right-clicking the executable map in the Composition or Outline view. See Figure 4-13.

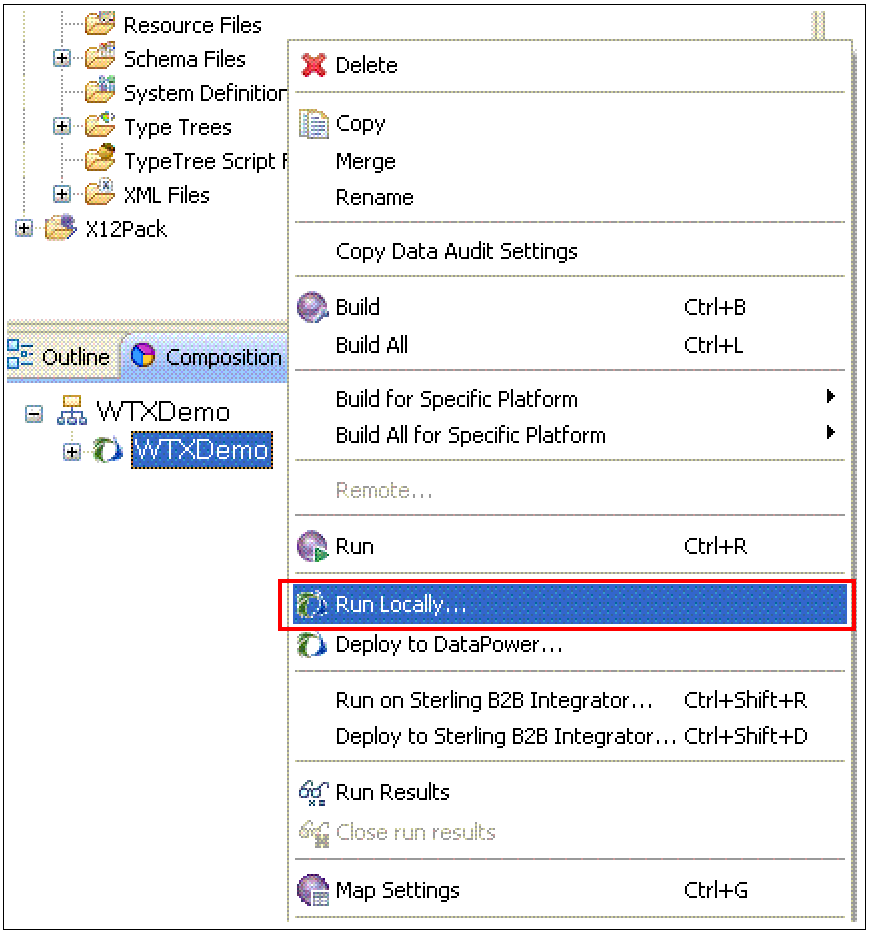


Figure 4-13 Test or deploy map on the WebSphere DataPower Appliance

Testing a map on WebSphere Message Broker

The Design Studio does not allow you to test from the designer on the target WebSphere Message Broker platform as it does for the WebSphere DataPower Appliance or the Sterling B2B Integrator server. However, like maps that execute on the Sterling B2B Integrator, maps intended to execute from WebSphere Message Broker will execute as they do on the local Design Studio environment as long as the map has been built for the environment that hosts the broker. Unit testing a map on the broker is discussed in “Testing maps in a WebSphere Message Broker runtime environment” on page 88.

Importing data format definitions

Along with the ability to execute maps on practically any platform, another of the most powerful features of WebSphere Transformation Extender is the ability to represent any type of message format through a type tree definition.

A type tree is the proprietary data definition structure used by WebSphere Transformation Extender to parse inbound data, and to build outbound data. In order to expedite generation of type trees, WebSphere Transformation Extender Design Studio provides the ability to import many standard data definitions into type trees. Valid xml schema definitions can either be imported into type trees, or used natively to define input and output. It is recommended that native xsd be used unless additional data definition properties need to be defined that are not supported as native xsd, or you intend to process broken or illegal XML.

In addition to the WebSphere Transformation Extender type importers that allow you to import the formats shown in Figure 4-14, application specific importers are available as add-on components to support formats like SAP IDocs, SAP BAPI, SAP ALE, and more.

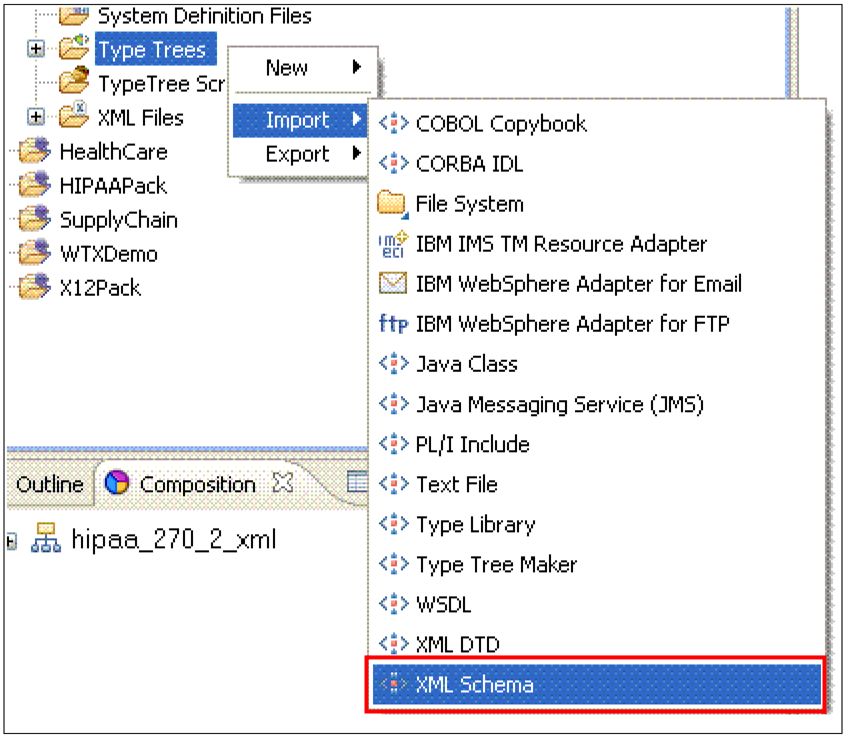


Figure 4-14 WebSphere Transformation Extender Type Importers

WebSphere Transformation Extender for Integration Servers

WebSphere Transformation Extender for Integration Servers allows WebSphere Transformation Extender translation capabilities and components to be ready to deploy in message flows, mediation flows, and business processes. WebSphere Transformation Extender complements the native capabilities of these IBM products, and can process large documents and messages with more complex formats that would prove more difficult using the native capabilities of the products. Integration with the following IBM products are packaged as WebSphere Transformation Extender for Integration Servers:

- WebSphere Message Broker
- Sterling B2B Integrator
- WebSphere Enterprise Service Bus

WebSphere Transformation Extender Industry Packs

WebSphere Transformation Extender Industry Packs provide out-of-the-box capabilities to integrate a range of industry standard data formats with your enterprise infrastructure. Industry packs enable developers to accelerate the delivery of transformation solutions by providing predefined type tree templates and conversion maps. In some cases they provide validation maps and tools to facilitate implementation in projects that require conformance to mandatory and advisory guidelines required by the regulatory body or industry service provider.

WebSphere Transformation Extender Industry Packs used in conjunction with WebSphere Transformation Extender provide organizations with an infrastructure with these features:

- Enables compliance with government and industry mandates.
- Controls administrative costs.
- Streamlines business processes.
- Facilitates accuracy and timeliness of information.
- Reuses existing business systems.
- Adapts to new technologies as they emerge.
- Integrates multiple systems and standards.
- Automates cross-organizational exchanges.

The scenarios described in this book make use of the HIPAA EDI and the EDI X12 industry packs.

HIPAA EDI

WebSphere Transformation Extender Pack for HIPAA EDI includes ready-to-execute template definitions for the complete ASC X12N standard for HIPAA including HIPAA 4010 and HIPAA 5010 addendum, as well as CMS (formerly HCFA) formats for legacy NSF, UB-92 Claims and Coordination of Benefits. The Pack features compliance validation checking, and handling of claims attachments. Also supports the National Provider ID requirement and Clinical Data Architecture release 2.0 for claims attachments support.

HIPAA EDI addresses the administrative simplification aspects of HIPAA legislation - the standardization of electronic patient health, administrative, and financial data. HIPAA regulations affect payers, health plans, clearinghouses, and those providers who conduct financial and administrative transactions electronically.

In the health insurance scenario discussed in Chapter 5, “Health Insurance scenario” on page 103, located in Part 2 of this book, the HIPAA version 5010 type tree that enforces type 4 compliance that ships with the HIPAA EDI industry pack is used to define the inbound services eligibility request. In addition, the HIPAA compliance_check maps that ship with the pack are used to generate the negative acknowledgement for non-compliant messages.

EDI X12

WebSphere Transformation Extender Pack for X12 includes pre built definitions for the ASC X12 transaction sets. ASC X12 represents multiple business domains including finance, government, health care, insurance, and transportation.

In the supply chain scenario discussed in Chapter 7, “Supply Chain scenario using AS2 and EDI” on page 261, the X12 version 4010 type tree is used to represent the inbound purchase order. In addition, the X12 compliance check maps that ship with the pack are used to generate the negative acknowledgement for non-compliant purchase orders.

Methods and practices

In the following sections, we describe some common methods and practices for routing and transforming the messages.

Testing maps in a WebSphere Message Broker runtime environment

There are times when you might want to test a map in the WebSphere Message Broker runtime environment without having to execute the entire message flow leading up to the map. It can be done by setting up a map test flow to invoke an executable map whose sole purpose is to execute other maps.

This section discusses how to accomplish the testing. We assume that you have a basic understanding of how to use the “RUN” map rule, and a basic understanding of runtime parameters that control the execution of WebSphere Transformation Extender maps.

The “RUN” map rule is the means by which an executable map can invoke another executable map at runtime. “RUN” also allows for any runtime parameter, such as card overrides, audit switches, and trace switches to be at the time the “RUN” command is executed.

From the WebSphere Message Broker perspective, you need a simple message flow that monitors a test queue, and initiates the flow when a message is dropped onto the queue. The purpose of the flow is to take this message, and pass it to the map. The information in the message forces the map to execute the map you want to test. See Figure 4-15.

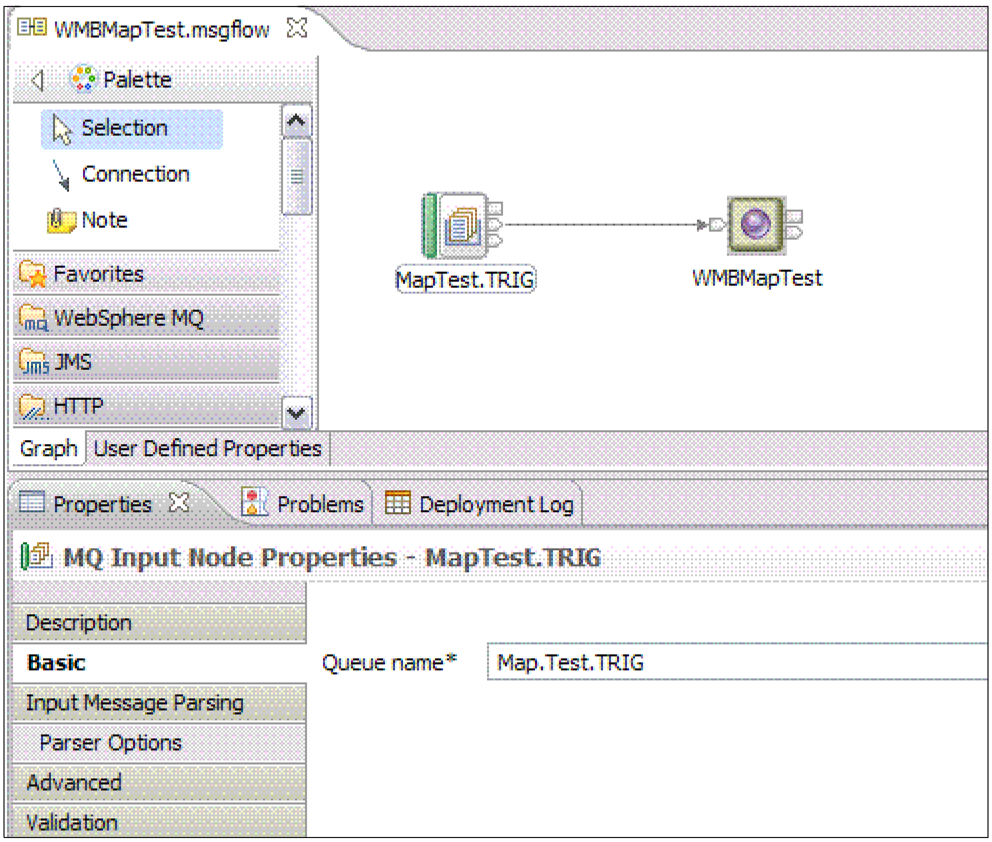


Figure 4-15 Map unit test flow

The map that is executed by the broker is made up of one input card and one output card. For the purpose of this discussion, the input card is defined using an xml schema definition. The xml contains four attributes, as shown in Example 4-1 and Example 4-2:

- The map to run, including the path
- Command line options, including the audit and trace switches
- Input overrides, including the overrides for any input card
- Output overrides, including the overrides for any output card

Example 4-1 XML schema definition

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="TestMapInfo">
    <xs:complexType>
```

```
      <xs:sequence>
        <xs:element name="MapToRun" type="xs:string" />
        <xs:element minOccurs="0" name="CmdOptions" type="xs:string" />
        <xs:element minOccurs="0" name="InputOverride" type="xs:string" />
        <xs:element minOccurs="0" name="OutputOverride" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Example 4-2 XML input data

```
<?xml version="1.0" encoding="UTF-8"?>
<TestMapInfo>
<MapToRun>C:\map\hipaa_270_2_xml.mmc</MapToRun>
<CmdOptions> -AE -TI </CmdOptions>
<InputOverride> -if1 C:\data\hipaa_270_5010a1.dat </InputOverride>
<OutputOverride> -of1 C:\data\output.msg </OutputOverride>
</TestMapInfo>
```

The output card definition uses the type tree shown in Figure 4-16.

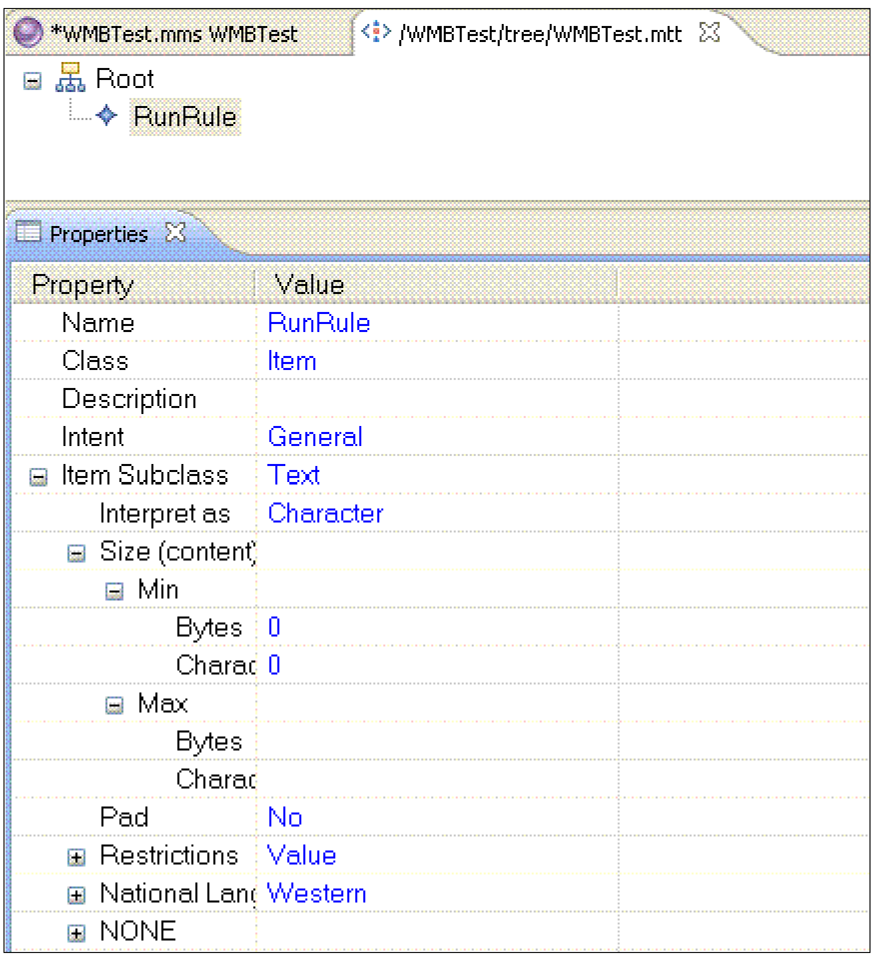


Figure 4-16 Type tree used for test map

After the broker flow delivers the message to the WebSphere Transformation Extender map node, the broker invokes the map that calls the map to be tested. This map takes the data passed in, and uses that data to set up the run command to call the map being tested.

- See the map rule used to execute the map passed into the flow in Figure 4-17 on page 91.
- See the input card definition for this map using the xsd from in Figure 4-18 on page 92.
- See the output card definition for this map using the type tree from Figure 4-16 on page 90 in Figure 4-19 on page 93.

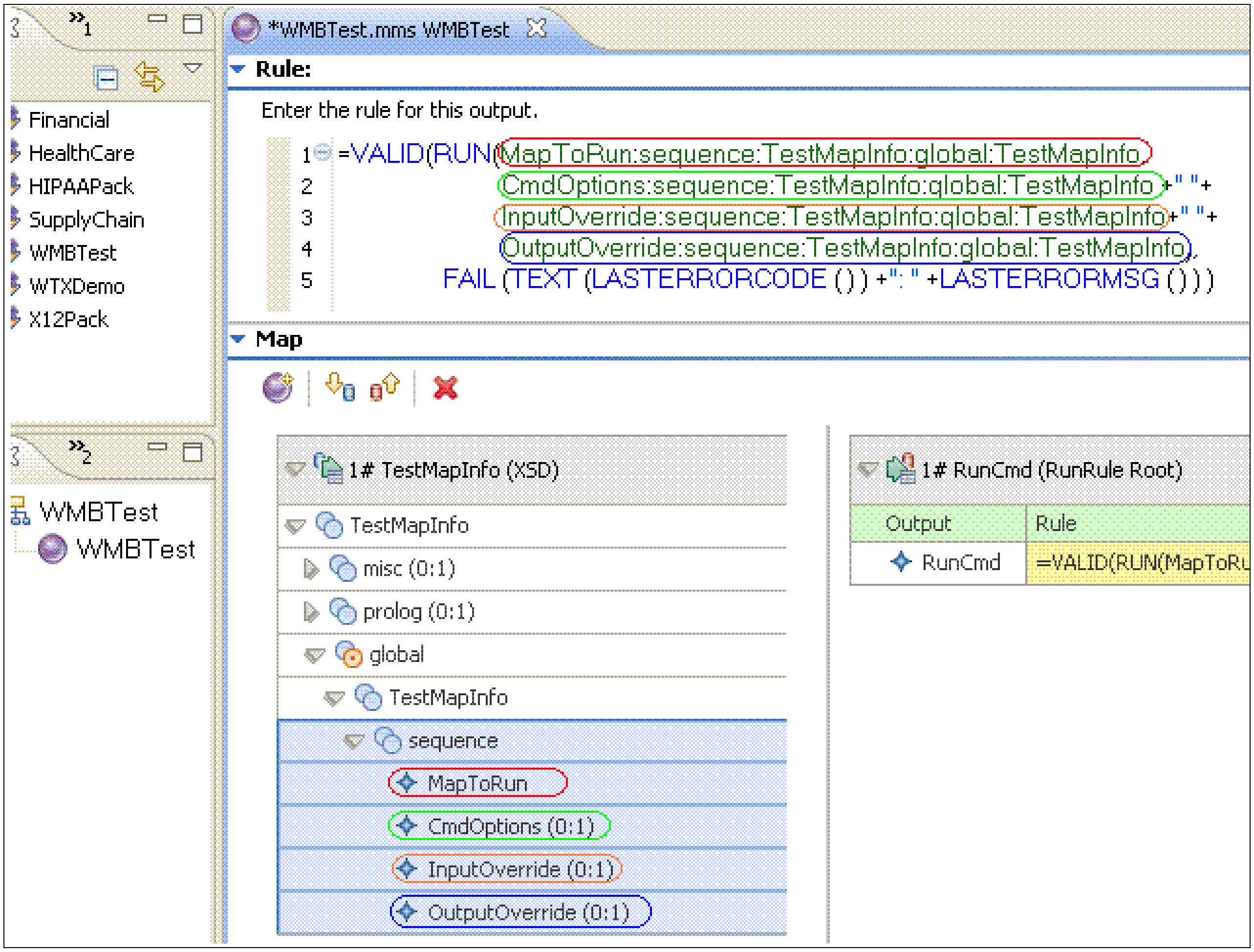


Figure 4-17 RUN map rule

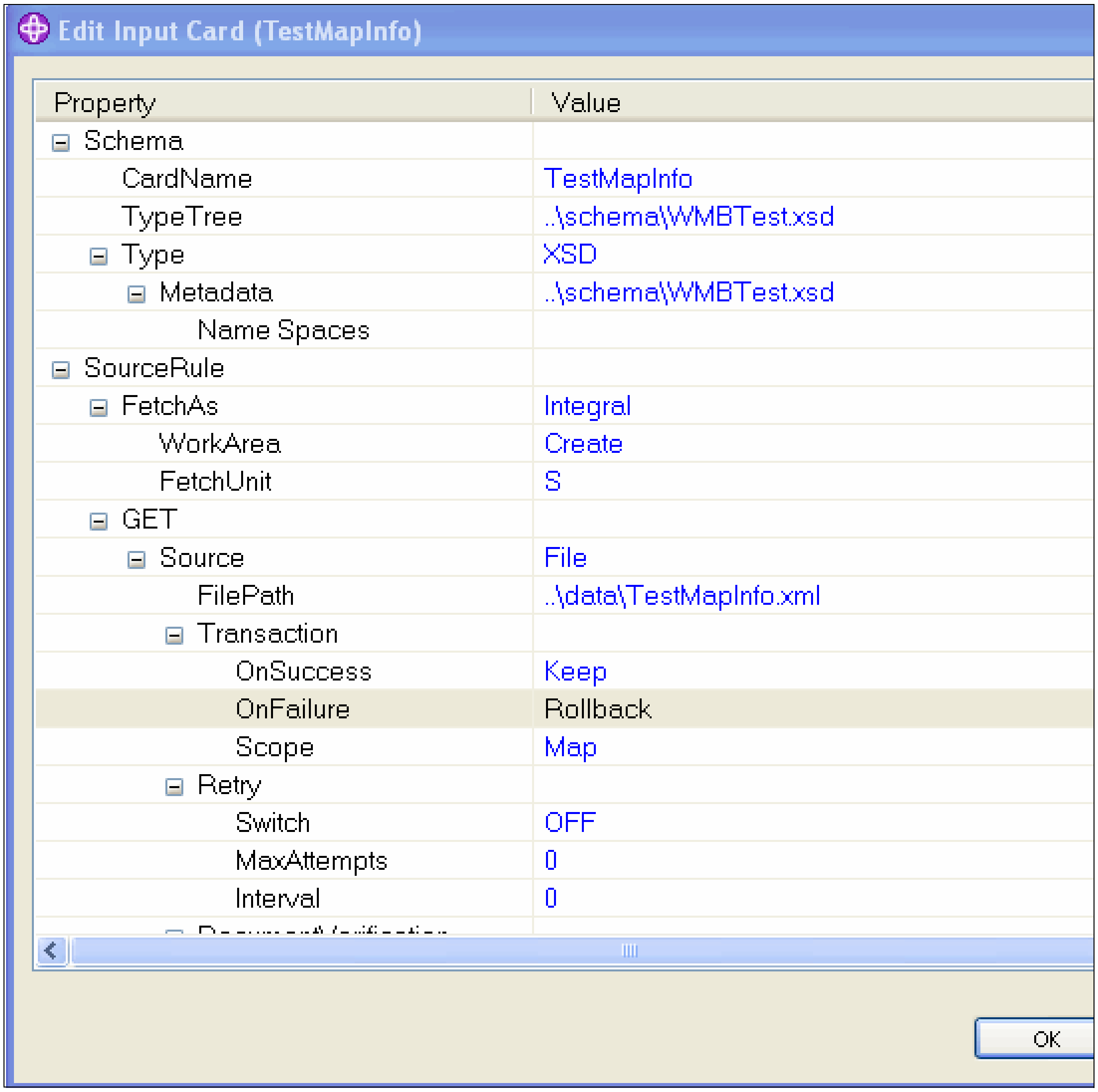


Figure 4-18 Input card

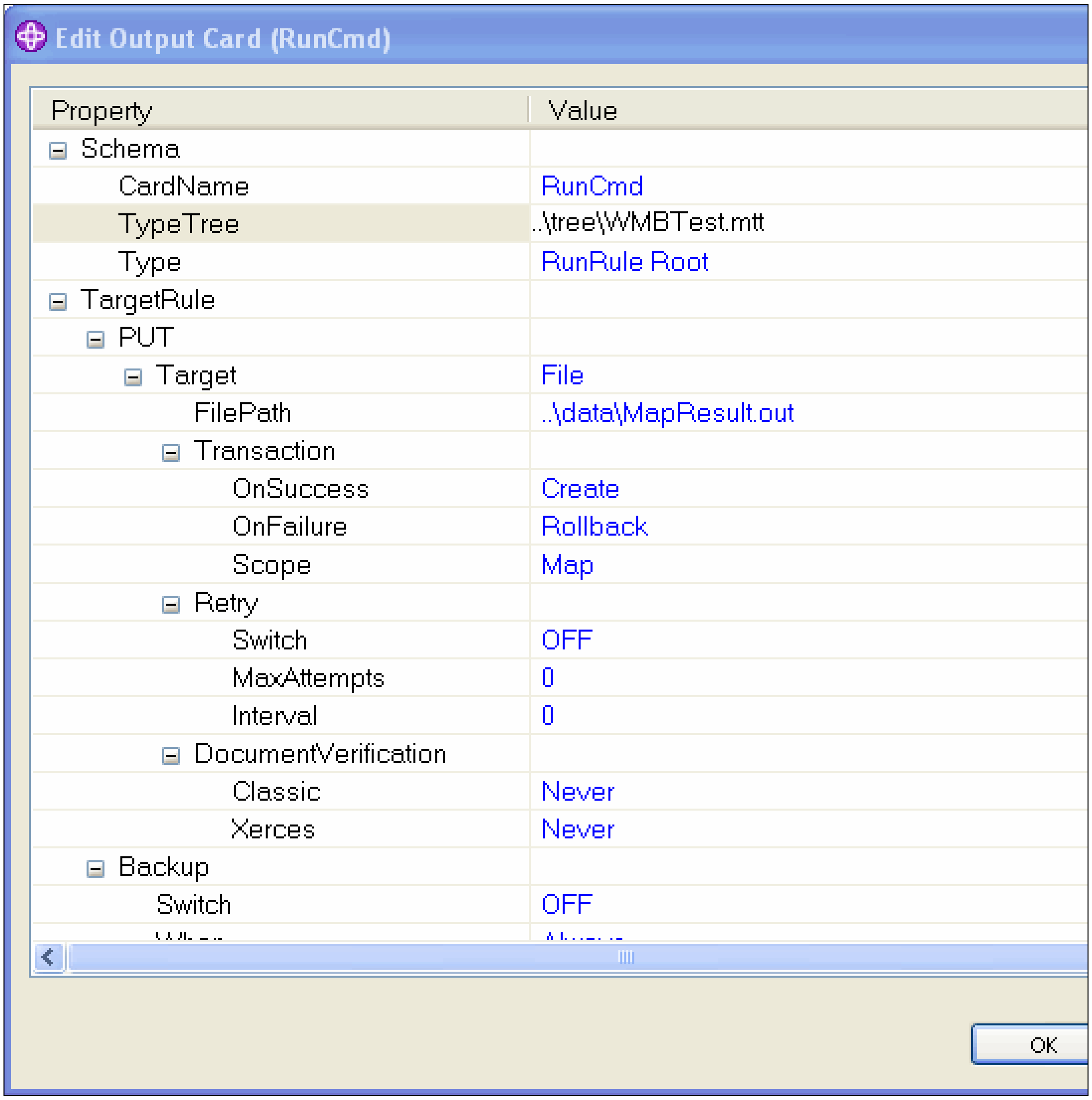


Figure 4-19 Output card

Trimming type trees

Trimming a type tree greatly improves the performance of parsing input of a WebSphere Transformation Extender map. Trimming is normally done to type trees shipped with an industry pack. The reason is that industry pack type trees can include many versions and transactions that you do not use in your solution.

If you execute a map with an untrimmed type tree, WebSphere Transformation Extender tries to parse each and every transaction until it successfully parses the inbound message with the correct definition. By trimming the type tree, unnecessary attempts to parse data are avoided improving performance. Trimming a tree does not affect the output performance, but it minimally simplifies mapping, because unused types do not need to be mapped to “NONE”

You trim type trees using the WebSphere Transformation Extender Design Studio Type Designer. From inside the type designer, you select the types not needed and delete them from the type tree. Recommended practice is to make a copy of the original tree, and to trim the copy, keeping the original as it was when it was originally shipped.

To illustrate the result of trimming a type tree, see Figure 4-20, where we take the type tree `hipaa_x12_type_4.mtt` that ships with the industry pack and show it side by side with the type tree `hipa_x12_type_4_trimmed.mtt` used in Chapter 5, “Health Insurance scenario” on page 103.

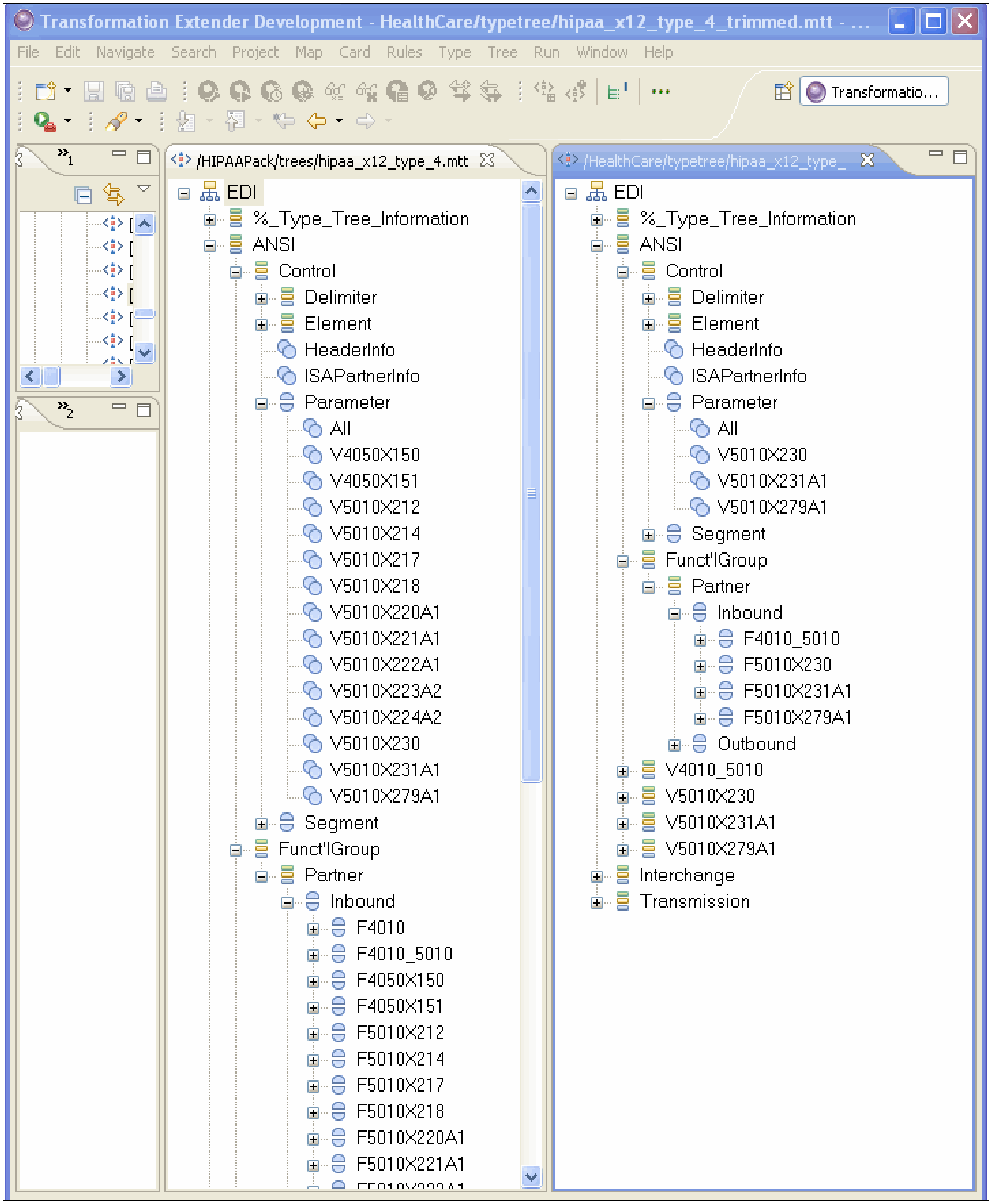


Figure 4-20 Comparing trimmed and non-trimmed type trees

Canonical data format

During the implementation of the scenarios in this book, we use the concept of a canonical data format. The purpose of the canonical data format is to provide a consistent format for data used for like purposes, which makes for easier expansion and maintenance in the future.

In order to process the information sent to you by your trading partners, you need to understand the format of the data they are sending. In a perfect world, all of your trading partners would send you data using the same format, and the application, or applications you use to process this information would accept the data without any changes. In other words, everyone would speak the same language. Unfortunately, that is seldom the case, even with the definition of industry standards for communicating information electronically, your trading partners might be sending data using the standard, but at a different revision level, which still causes the same issues.

Add to that, the fact that the information you receive is being used by more than one application. Each of those applications also expect the data to be in a different format, and you can have many translation processes that you need to build and maintain.

Figure 4-21 shows how complicated such a solution can be. It is an illustration of incoming data being sent using four different formats to three different applications. It also shows that each application creates a response in each format. In order to accomplish the requirements shown in Figure 4-21, Note that twenty-four translations would be required. Four source formats translated to three application formats would give you twelve translations, and then you would need to translate each of the three application formats back to the source formats in order to respond for 12 more formats, giving you a total of twenty-four translations.

Now suppose that one of the source formats is modified. To implement this change would require you to change six translation processes. If an application format changed, it would require eight translation process changes. Adding a new source format would require six new translations, and adding a new application would require eight. As you can see, it would not take long before this type of point-to-point solution became very complex and burdensome.

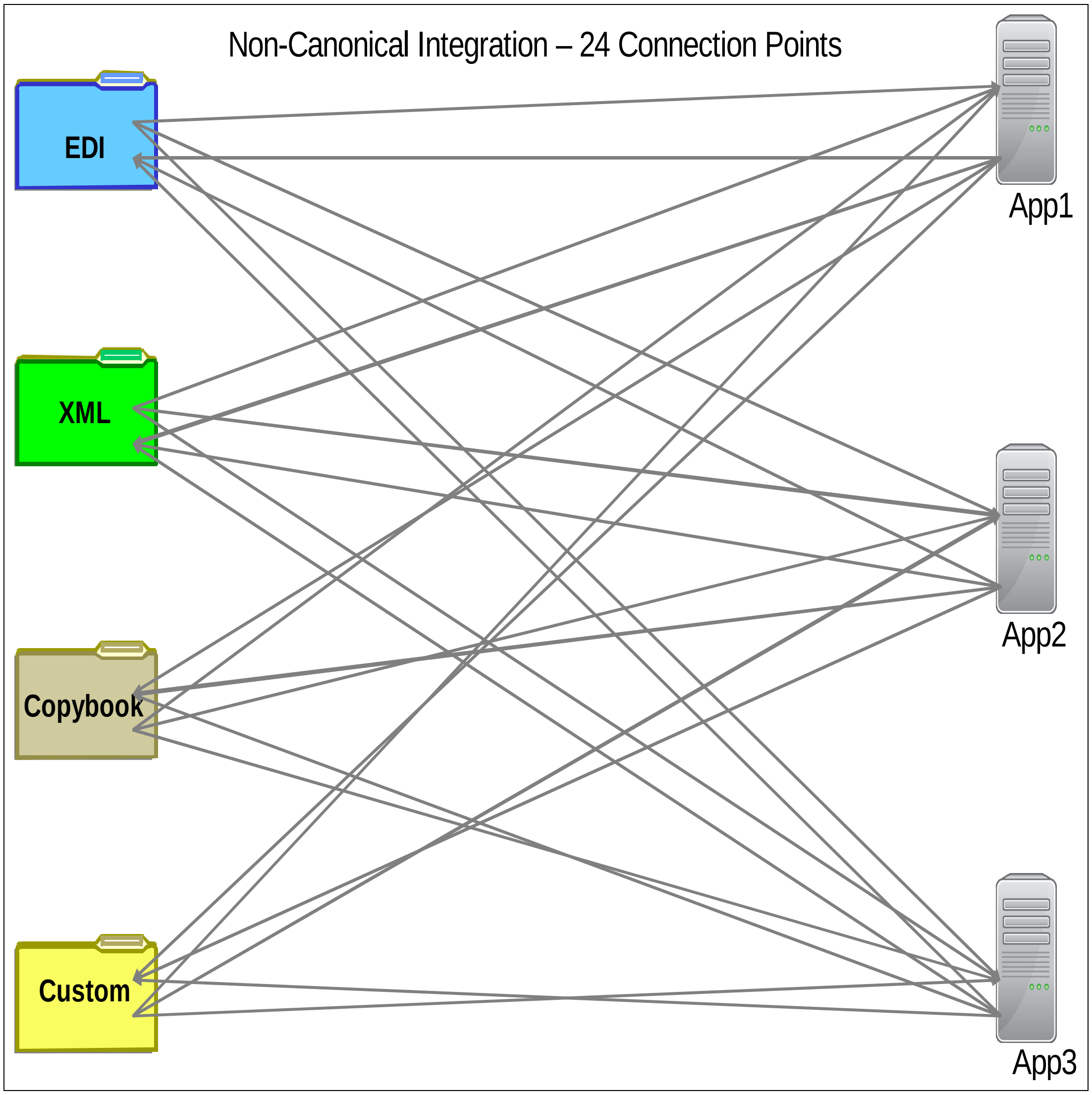


Figure 4-21 Data translation without an intermediate canonical format

Next we look at implementing the same solution using a canonical data format as an intermediate translation format. As you can see in 4.3.3, “WebSphere DataPower XB62 appliance” on page 99, the complexity is reduced dramatically. Now, instead of twenty-four translations there are fourteen; two each for the four source formats and the three application formats.

When one of the source or target formats changes, at most it affects two translations: the translation to and the translation from the canonical format. Before it affected six or eight translations, depending on which format changed. If you add a new format or a new application, again it would only require two new translations.

Another advantage is that you would gain a deeper understanding of how to make changes, because all data flows to or from the same format. It is why the approach used in the scenarios in this publication, as well as the recommendation for building transformation and translation within the enterprise service bus, is to make use of canonical data structures. As you can see, sometimes the canonical structure is either the target or the application format, in which case your work is cut in half.

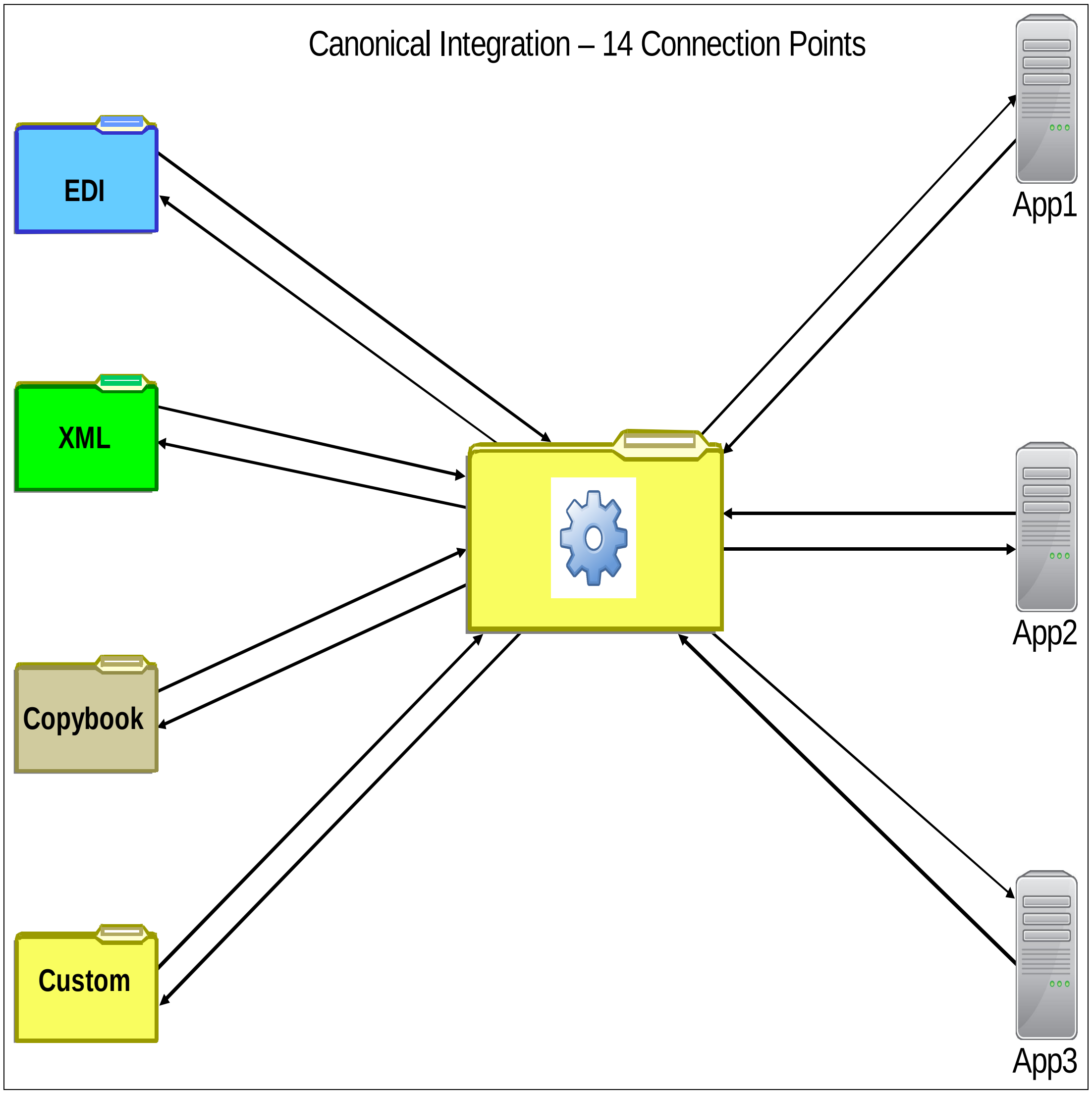


Figure 4-22 Data translation with a canonical intermediate format

Recommended practices when implementing WebSphere Transformation Extender

Because WebSphere Transformation Extender can be integrated with so many product portfolios, recommended practices can vary based on the environment where it is being used. Some of the recommended practices that are discussed in the various scenarios covered in this book, such as “Canonical data format” on page 95 are not limited to use with WebSphere Transformation Extender. Other recommended practices discussed in this book, such as “Trimming type trees” on page 93, that make parsing more efficient, are recommended regardless of runtime environment.

In order to access the WebSphere Transformation Extender recommended practices that best meet your needs, you need to make use of the help facility that is included with your Design Studio, the WebSphere Transformation Extender Online Library, if you have it installed on your workstation. Information can be found at the following websites:

IBM Developer Works articles:

http://www.ibm.com/developerworks/websphere/library

The WebSphere Tranformation Extender product page:

http://www-01.ibm.com/software/integration/wdatastagetx/

The IBM WebSphere Transformation Extender Developer Works Forum:

http://www.ibm.com/developerworks/forums/forum.jspa?forumID=829

A specific recommended practice that we discuss here is the use of the map rule function “PUT” with the adapter alias “WIRE”. This map rule function is used in two of the scenarios discussed in part 2. The syntax for this map rule function is shown in Example 4-3.

Example 4-3 Using “PUT” with the “WIRE” adapter alias in Message Broker and Sterling Business Integrator

```
PUT(adapter_alias, adapter_command, data_to_send_to_adapter)
PUT(“WIRE”, ”OutputTarget”, DATA) /* Sends data when map completes */
PUT(“WIRE”, ”OutputTarget -NOW”, DATA) /* Sends data immediately */
```

Using the “PUT” map rule function, as in this example, is valid with both WebSphere Message Broker and in Sterling B2B Integrator. Although the purpose for using this map rule function is the same with both, the behavior of the map rule function differs between the two.

PUT(“WIRE”,....) is used when you want to split a message up into individual parts to be processed separately. The Financial Services extended scenario, uses the PUT map rule function with the “WIRE” adapter alias in a map invoked by WebSphere Message Broker, to split a message containing multiple payments records into individual payments records to be processed separately. The Health Insurance scenario in Chapter 5 converts an inbound HIPAA 837P message into an xml version of the message with each claim contained in its own xml envelope to be processed individually. It is done using the PUT map rule function with the “WIRE” adapter alias within a map that is invoked by Sterling B2B Integrator.

When using the map rule function PUT(“WIRE”, “OutputTarget”, DATA) with WebSphere Message Broker, each execution of the PUT places the DATA contained in the third parameter of the command on the wire connected to the Output Terminal of the Map Node named in the second parameter of the command. Each individual instance of DATA is processed individually by the remainder of the message flow.

When using this same map rule function with Sterling Business Integrator, you must somehow differentiate the name defined in parameter 2, for each time PUT is invoked. It is normally done by concatenating an index and the second parameter together. It results in the creation of tags with the concatenated name referencing the DATA in parameter 3 of the map rule function. Failure to follow this practice results in only the last instance of DATA passed by the third parameter of the map rule function being available upon completion of the map.

Example 4-4 shows the proper use of the PUT map rule when invoked from Sterling B2B Integrator, and an excerpt from ProcessData showing the result.

Example 4-5 shows the improper use of the PUT map rule when invoked from Sterling B2B Integrator, and an excerpt from ProcessData showing the result.

The same HIPAA 837P, containing two claims, was used in both examples.

Example 4-4 Proper use of PUT when invoked by Sterling B2B Integrator

```
PUT("WIRE", "OutputTarget_"+FILLLEFT(Idx,"0",3), DATA)
<ProcessData>
  <PrimaryDocument SCIObjectID="572:2406457:135400a450b:scb104-5:node1"/>
  <output_document_001 SCIObjectID="629:2406473:135400a450b:scb104-5:node1"/>
  <output_document_002 SCIObjectID="732:2406479:135400a450b:scb104-5:node1"/>
  <WTX_MAP_EXECUTION_AUDIT_LOG_346995
SCIObjectID="366:2406482:135400a450b:scb104-5:node1"/>
</ProcessData>
```

Example 4-5 Improper use of PUT when invoked by Sterling B2B Integrator

```
PUT("WIRE", "OutputTarget_"+FILLLEFT(Idx,"0",3), DATA)
<ProcessData>
  <PrimaryDocument SCIObjectID="572:2406457:135400a450b:scb104-5:node1"/>
  <output_document SCIObjectID="517:2406467:135400a450b:scb104-5:node1"/>
<WTX_MAP_EXECUTION_AUDIT_LOG_346995
SCIObjectID="366:2406482:135400a450b:scb104-5:node1"/>
</ProcessData>
```

4.3.3 WebSphere DataPower XB62 appliance

WebSphere DataPower B2B Appliance XB62 is a purpose-built hardware B2B-enabled ESB for simplified deployment and hardened security with the ability to quickly transform data between a wide variety of formats, including XML, industry standards, and custom formats. The device provides core B2B functions, including AS2 and AS3 messaging, partner profile administration, routing of electronic data interchange (EDI), XML, and binary payloads, auto archiving and purging of B2B transactions, and B2B transaction viewing capabilities.

The ESB functions include routing, bridging, transformation, and event handling. Data transformation and validation features include native XML Schema and WSDL validation, XSLT-based transformations, any-to-any transformation using WebSphere Transformation Extender, DPA maps created in IBM WebSphere Transformation Extender Design Studio, and type trees from the HL7 Industry pack.

WebSphere DataPower XB62 appliance provides a reliable, performance-oriented solution to many integration challenges. Because it is not limited to handling just XML, it resonates with IT organizations that need to benefit from the connectivity of SOA deployments but must also deal with managing a combination of multiple proprietary, industry, company-specific, and existing data formats.

4.3.4 Sterling B2B Integrator

Sterling B2B Integrator enables the secure integration of complex B2B processes with diverse partner communities.

IBM Sterling B2B Integrator offers perimeter security, gateway, and transformation engine in one product. It includes the following integration and transformation features:

- Multi-purpose data transformation engine:
  - Traditional EDI: X12, EDIFACT, CII, TRADACOMS, and Verband der Automobilindustrie (VDA)
  - XML standards: OAGi, CIDX, PIDX, and RosettaNet
  - Internet standards for B2B data exchange: RosettaNet RNIF, ebXML, 1SYNC, and EBICS (France)
  - XSLT service to transform XML documents
- WebSphere Transformation Extender translations support
- Graphic data mapping tool
- Virtually unlimited file size (up to 50 gigabytes)
- Validation of inbound and outbound data based on HIPAA rules defined for Level 1 – Level
- Intelligent (content-based) routing
- Interoperable with .Net 1.1/2.0, Axis 1.x/2.0, Xfire 1.2.6 and Java EES

The various adapters and services that come with Sterling B2B Integrator can be employed in the routing and transformation of messages in many different ways. One of the most common is to use the Mailbox functions of Sterling B2B Integrator, in conjunction with Routing Rules, to move messages from one Mailbox to another. It allows trading partners to upload documents, which can be translated and then routed automatically to another Mailbox for another partner or internal system to retrieve and perform additional processing.

The implementation of many of these routing functions can be simplified by using the Sterling File Gateway application. Sterling File Gateway also provides robust tools for viewing the documents or files, as well as the routing status of the messages in the application.

For more information about the Mailbox functions of Sterling B2B Integrator, see the Mailbox overview in the documentation at this website:

http://publib.boulder.ibm.com/infocenter/sb2bi/v5r2/nav/6_4.

For detailed information about Sterling File Gateway, see the documentation for that product at this website:

http://publib.boulder.ibm.com/infocenter/sfg/v2r2/index.jsp.

Part 2

Scenarios based on the proposed solution architecture

In this part of the book, we describe four scenarios that demonstrate ways to integrate IBM Sterling and IBM WebSphere portfolio products. Each scenario implementation is discussed. Each implementation is executed using a different combination of products and/or product capabilities from the available products in the IBM Sterling and WebSphere portfolios.

5

Health Insurance scenario

This chapter demonstrates an example scenario that uses several different systems to receive, check, transform, and route a HIPAA EDI 837 Health Care Claim transaction. They then respond back to the source system with an EDI 999 acknowledgement.

Because we assume that the claim will be processed by an existing backend application or process, our scenario ends when we deliver the request to the destination that performs this task.

Beyond the acknowledgement of the receipt of the message, this scenario does not present the integration required to deliver a response message back to the trading partner that generated the initial message. It is an extension to the scenario, which could be conceived and executed by the reader.

This chapter includes the following sections:

Tip: If you want to implement the scenario presented in this chapter in your own environment, you can download the Project Interchange file for the applications used in this scenario from the ITSO FTP site. For download instructions, see Appendix A, “Additional material” on page 333.

5.1 Business value

During an interaction between a patient and a healthcare provider, there is a large amount of data that is collected and exchanged in order to provide the proper services for the patient. Much of the data collected can be shared with other entities, for any of various reasons:

- Patient encounter reporting
- Payment processing
- Plan eligibility
- Coordination of benefits
- Health care claim status
- Referral authorization

A very common need is to prepare and send the information relating to the encounter, and possibly claim information, to an insurance company or other entity for reporting purposes or payment of the claim.

While this encounter or claim can be processed manually, with a worker preparing the document manually and submitting it by mail to the entity requesting it, a much more time and cost effective approach would be to automate these requests, thereby eliminating much of the wait time for the information while also ensuring higher accuracy.

The provider office could simply format the document according to a defined structure (such as the HIPAA EDI 837), and then transmit this request electronically to the company or entity providing the benefits. The receiving company could then, using a composite system with various points of integration, accept and process that message, and send back a response to the originating system to acknowledge the receipt of the message. This would cut down on manual processing steps, reduce the wait time, and help eliminate errors in the request.

We explore an example solution to this scenario in this chapter, demonstrating how various products can be integrated to provide a seamless and high performance solution for processing these requests.

5.2 Prerequisites: Technical and infrastructure

There are prerequisites both in order to fully understand the scenario and to successfully implement it in your own infrastructure.

5.2.1 Software prerequisites

In order to be able to run this scenario, you must have the following components installed:

- Sterling Secure Proxy v3.4
- Sterling B2B Integrator v5.2.3
- Sterling File Gateway v2.2.3
- WebSphere MQ v7.0.1.5
- WebSphere Message Broker version v8.0.0.0
- WebSphere Transformation Extender Design Studio v8.4
- WebSphere Transformation Extender for Integration Servers v8.4
- WebSphere Transformation Extender Pack for HIPAA EDI v4.4.0

Figure 5-1 illustrates the configuration of the environment used to implement this scenario, and represents the systems of the fictitious company “ITSOHealthcare Corporation.”

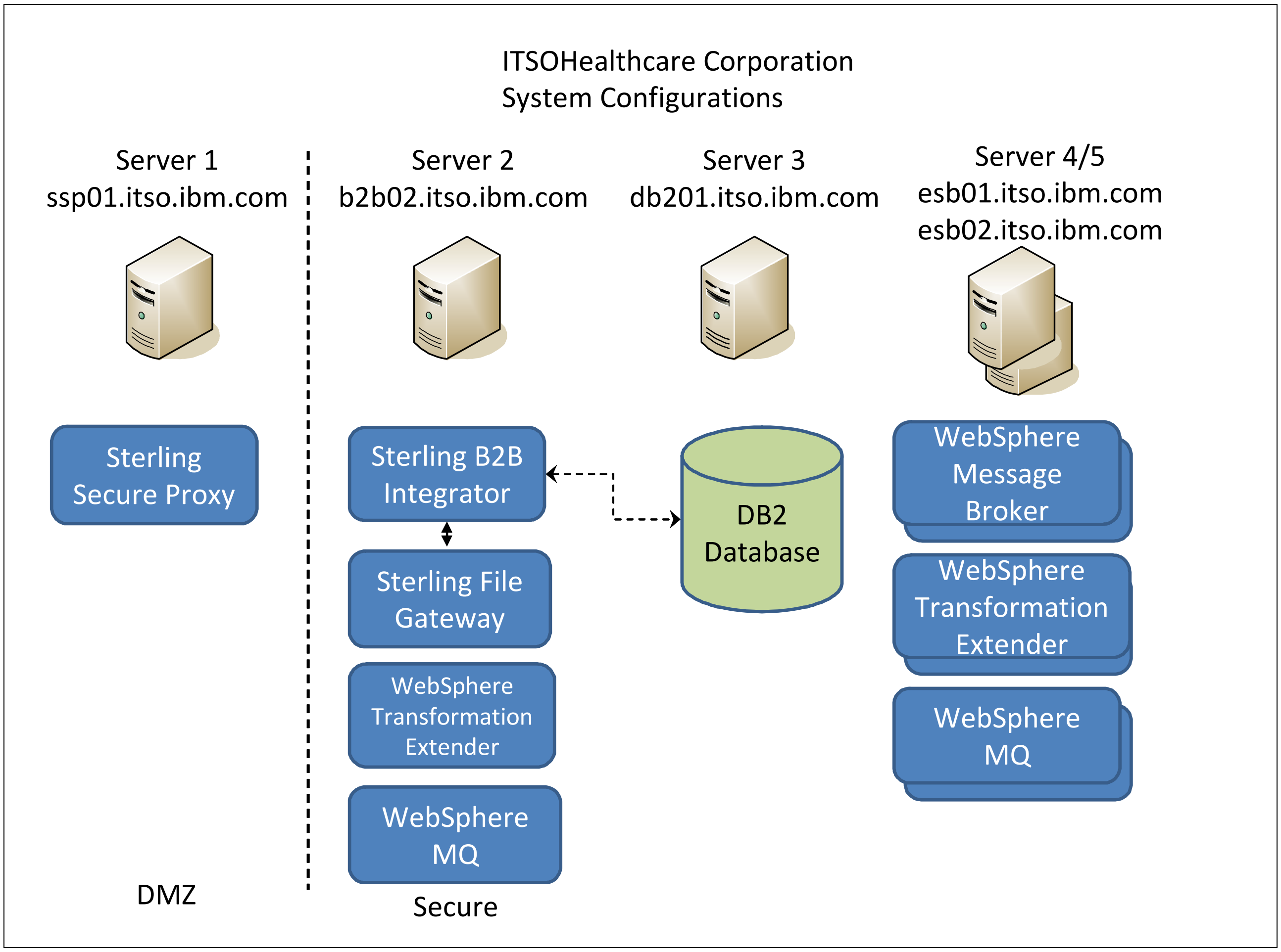


Figure 5-1 System configuration for ITSOHealthcare

For the purposes of this scenario, we use five servers:

- The first server is in the DMZ and has Sterling Secure Proxy installed.
- The second server has Sterling B2B Integrator and Sterling File Gateway.
- The third server has an IBM DB2® database instance installed to support Sterling B2B Integrator.
- The fourth and fifth servers are load balanced backend ESB integration servers and have WebSphere Message Broker and WebSphere Transformation Extender installed in them.

Servers two, four, and five also have WebSphere MQ for communication between the Broker components and also Sterling B2B Integrator integration.

5.2.2 Skills prerequisites

To fully implement and understand this scenario, you must be familiar with these tasks:

- Configuring Sterling Secure Proxy to communicate with Sterling B2B Integrator
- Configuring Sterling B2B Integrator to receive and process AS2, X12 and HIPAA messages
- Configuring Sterling File Gateway to receive, handle, and route messages

- Configuring either Sterling File Gateway or Sterling B2B Integrator to move messages to a WebSphere MQ queue
- Configuring WebSphere MQ infrastructure
- Configuring WebSphere Transformation Extender to check messages for compliance and to transform them into a format for the backend system
- Configuring WebSphere Message Broker infrastructure and develop message flows

5.3 Presenting the healthcare scenario

In this section, we provide an overview of our health care scenario. For this particular scenario, we implement a process flow where a Health Care Claim (HIPAA EDI 837) is received by our fictitious company, which is named “ITSOHealthcare Corporation” for the purpose of this scenario. As part of the process, we receive the file, check it for compliance, and respond back to the originator with an acknowledgement.

As shown in Figure 5-2, the scenario is composed primarily of the inbound flow, where the message is moved through the various components of the solution, on its way to the eventual destination in a backend application. There are some components of the outbound flow that are incorporated, but for the sake of clarity, we are focusing on the integration of the various systems that are working with the inbound message.

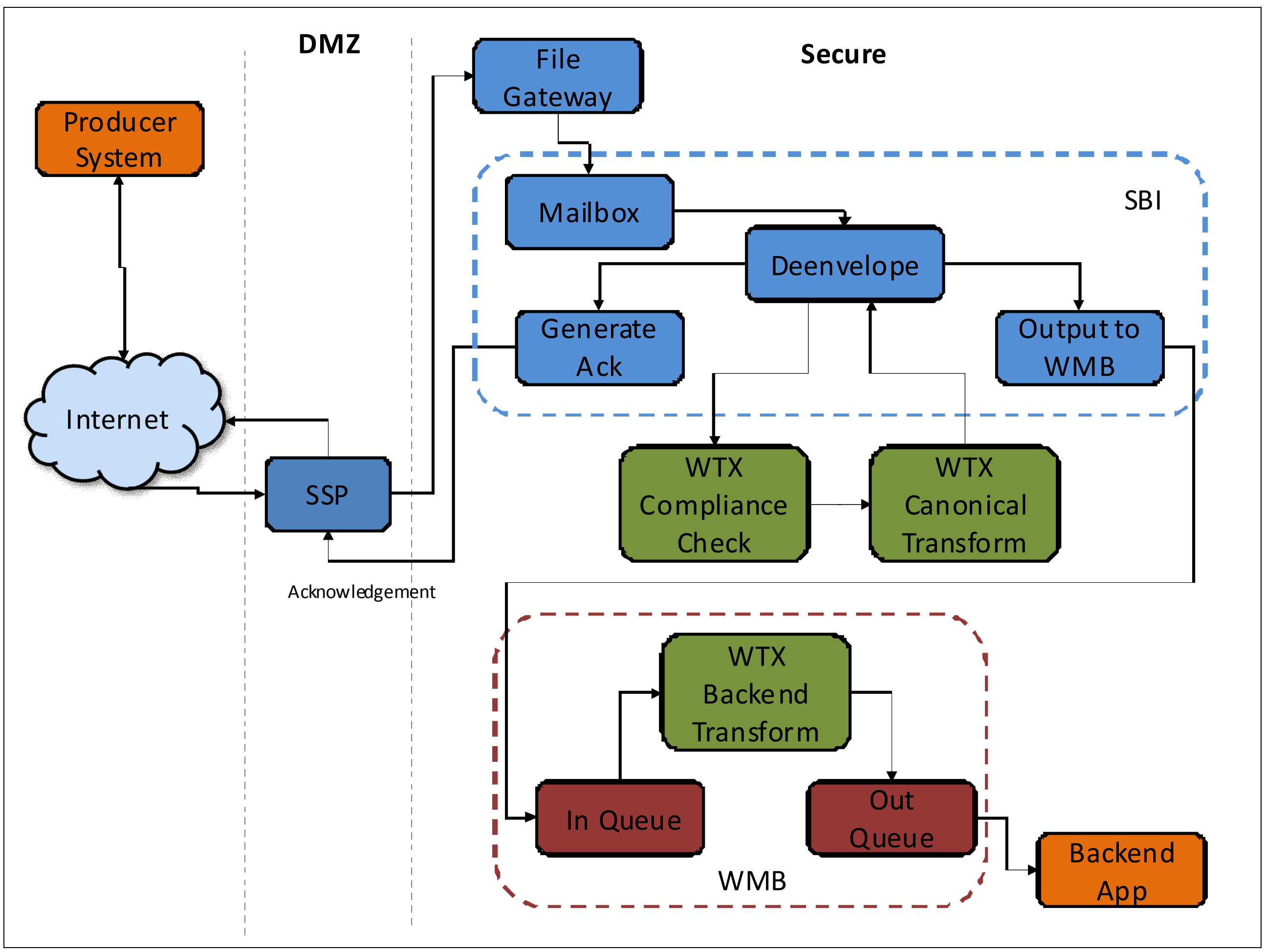


Figure 5-2 Overall process flow, both inbound and outbound

There are two primary areas of the inbound process, and then the rudimentary outbound flow. We look at each in more detail in the following sections.

5.3.1 Inbound flow part 1: Sterling Secure Proxy/Sterling B2B Integrator/Sterling File Gateway

Figure 5-3 shows the first part of the flow, which begins with the producer system creating the message, and ends with the message being routed out of Sterling B2B Integrator.

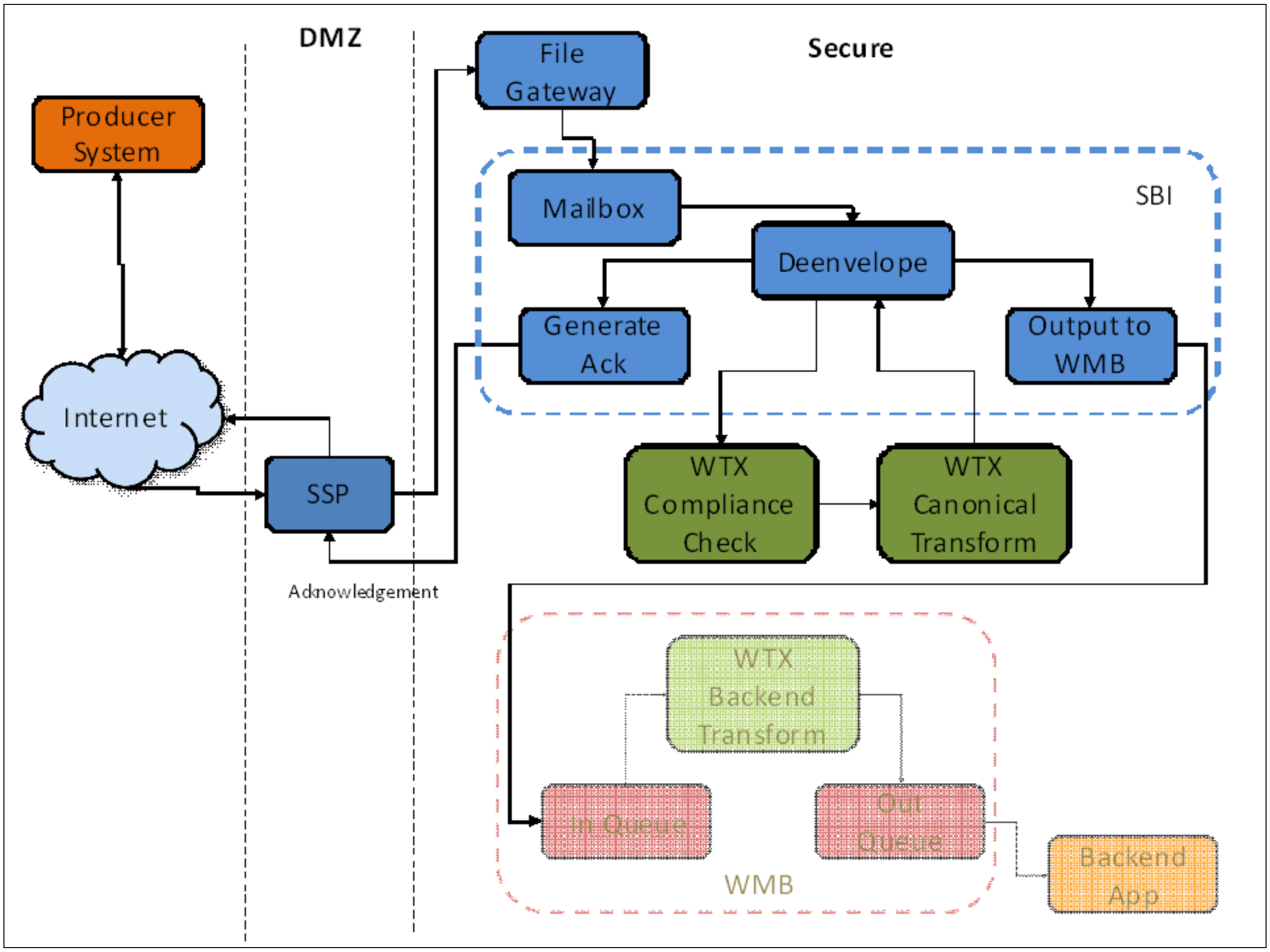


Figure 5-3 Part 1 of the inbound flow

The flow begins in the partner's system, where a HIPAA EDI 837 message is generated as part of the health care claim request on the behalf of a client. The partner's system would then transmit this message by public networks, and it would be received in the DMZ by Sterling Secure Proxy (SSP).

After SSP reviews the message, and determines that it is allowed into the Secure zone, it then passes the message on to Sterling File Gateway (SFG) for routing. SFG interrogates the message to ensure that it contains a valid transaction for that particular partner, and it then routes that message into a mailbox for further processing.

When the message is received in the mailbox, the Sterling B2B Integrator system then begins the processing of the message. It extracts the message from the mailbox, and then calls the service that de-envelopes the message. De-enveloping the document generates the acknowledgment for the partner, which is routed back to them as part of the outbound flow.

When the de-enveloping occurs, it calls a WebSphere Transformation Extender map, which checks the message for compliance with the EDI standards in use. If it passes the compliance check, it then calls another WebSphere Transformation Extender map, which converts the message from the current format into the canonical format for additional processing. As part of this processing, the system generates one message per claim in the document.

As the final step in this portion of the processing, Sterling B2B Integrator then places the resulting messages onto a WebSphere MQ queue for that processing.

5.3.2 Inbound flow part 2: WebSphere Message Broker/WebSphere Transformation Extender

Figure 5-4 demonstrates the second portion of the flow. It begins with the messages being placed into the WebSphere MQ queue, and ends with the final message arriving in the backend application.

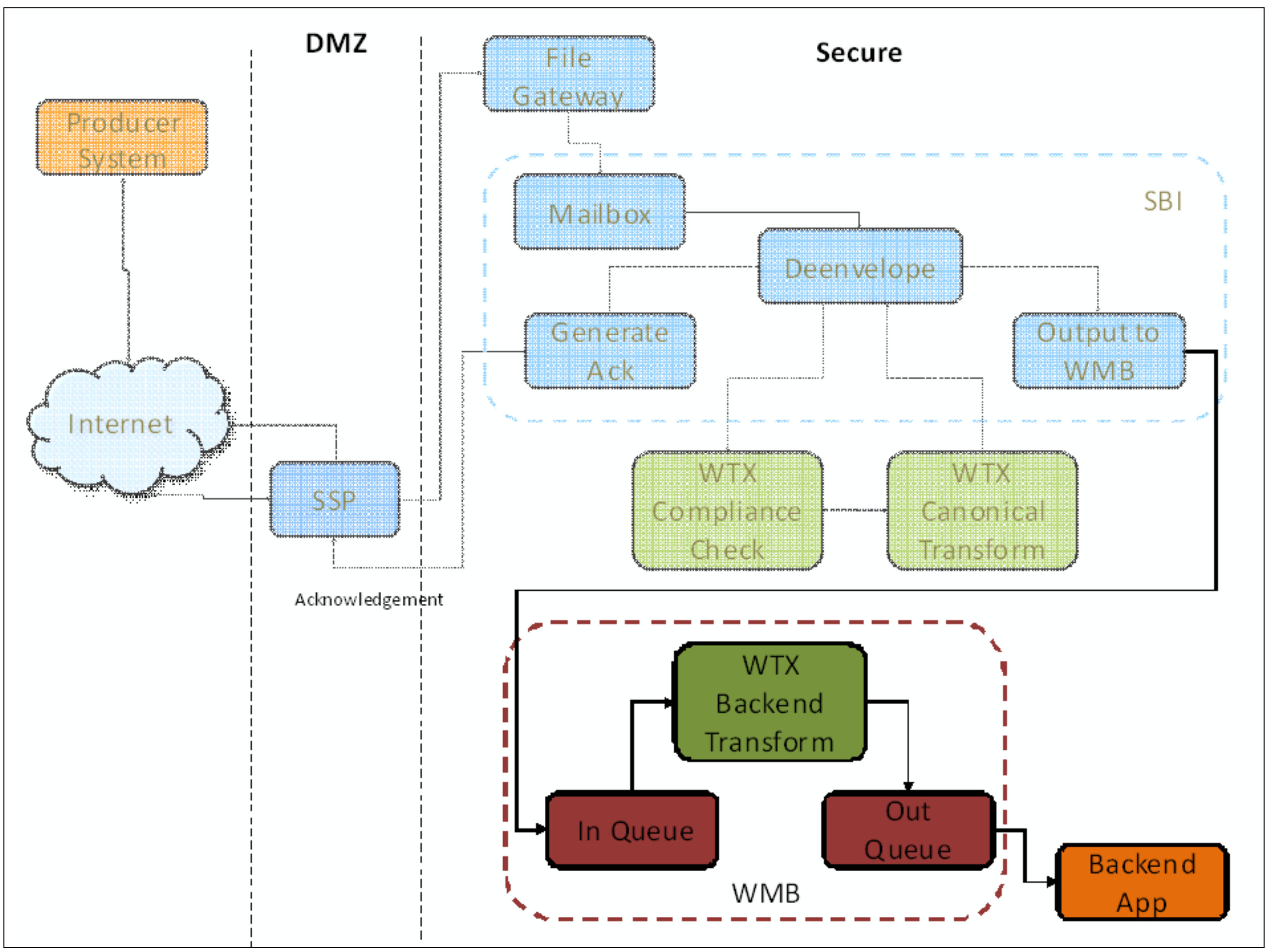


Figure 5-4 Part 2 of the inbound flow

After Sterling B2B Integrator completes the de-enveloping and preliminary transformation of the messages, it places them into the In Queue for WebSphere Message Broker (WMB).

Upon arriving on the queue, WebSphere Message Broker then calls a WebSphere Transformation Extender node, to convert the message from the current canonical format into a format that is compatible with the existing backend application.

After this transformation is complete, WebSphere Message Broker then moves the message into the Out Queue. This queue feeds the message into the backend application for additional processing.

5.3.3 Outbound flow

As mentioned earlier, the outbound, or return, flow is simplified in this scenario to concentrate on the integration of the various components as they move and process the inbound message. However, the simple outbound portion, as depicted, should be reviewed.

After the receipt of the message into Sterling B2B Integrator, the message is de-enveloped, and as part of this process, Sterling B2B Integrator generates an acknowledgement (in this case, an EDI 999). It is routed back to the partner to demonstrate that the message was successfully received, as well as any compliance errors generated.

Eventually, the backend application can be expected to generate a response to the messages, which need to be routed back to the partner. This return message needs to be processed through a similar flow (although in reverse). As it is out of the scope of the current scenario, we leave the conceptualization of this more detailed outbound flow to the reader.

5.4 Configuring the scenario

In this section, we look at an overview and then a detailed description of how we configure this Healthcare Integration scenario. In this scenario, we configure our various products to accept an inbound HIPAA EDI 837 Health Care Claim request by AS2. We route the resultant message through Sterling File Gateway. Then we use Sterling B2B integrator, WebSphere MQ, WebSphere Message Broker, and WebSphere Transformation Extender to prepare that message for processing in the existing backend system.

5.4.1 Scenario outline

For simplicity, we use the following fictitious companies:

ITSOHealthcare Corporation: The company for which this integration scenario is being implemented. It is a large healthcare company that receives and processes Health Care Claim (HIPAA EDI 837) requests from its partners.

ITSORetail Corporation: An external trading partner that is generating a Heatlh Care Claim (HIPAA EDI 837) request for one of their clients. In this case, this external partner is using software that accepts the created message and transmits it over AS2 to the systems of our healthcare company.

Here is a list of the steps to be performed in configuring this scenario.

1. Configure Sterling Secure Proxy.
2. Configure AS2 in Sterling B2B Integrator.
3. Configure Sterling File Gateway.
4. Configure message processing in Sterling B2B Integrator.
5. Configure WebSphere MQ.
6. Configure WebSphere Transformation Extender.
7. Configure WebSphere Message Broker.

5.4.2 Scenario implementation

In all of the following steps, we assume that the application in question has been successfully installed and tested to ensure that it functions properly and that a nominal performance configuration has been put in place for each to accommodate the anticipated processing load.

For detailed information about installing the applications, see the *Installation Guide* and/or the *Performance Tuning Guide* (if applicable) for each of the products.

Step 1: Configure Sterling Secure Proxy

To prepare Sterling Secure Proxy for use in this scenario, the following tasks must be performed:

- Configure a Policy
- Configure a Netmap
- Configure an HTTP Adapter

For the purposes of configuring Sterling Secure Proxy, it should be noted that our AS2 connection is not seen as any different than any other HTTP communications. Thus, we configure the various components as we would any other HTTP session.

For more information if required, see the documentation for Sterling Secure Proxy:

`http://www.sterlingcommerce.com/documentation/home/MFT/SSP/SSP.html`

It contains comprehensive details on configuring the application for common usage tasks.

We now review each of these steps in detail for this scenario.

Configure a Policy

Use the following procedure to define a Policy for our HTTP connection.

1. Log into the user interface for Sterling Secure Proxy.
2. Select **Configuration** from the menu bar and then select **Actions** → **New Policy** → **HTTP Policy**.

3. Select a name for the new Policy, and then click **Save**. Figure 5-5 shows an example of the Policy created for this example.

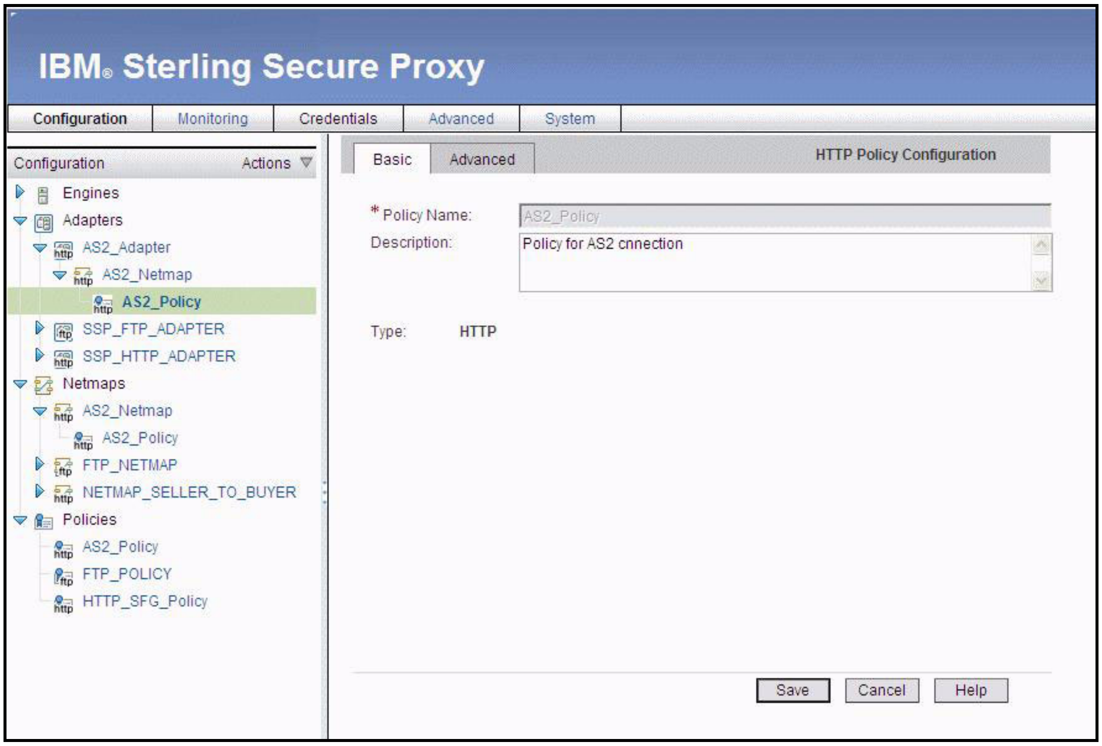


Figure 5-5 Creation of the Policy in Sterling Secure Proxy

Configure a Netmap

After we have created our new Policy, we must then create a Netmap. The Netmap defines the inbound connection information from the external trading partner, and the outbound connection information in order to communicate with Sterling B2B Integrator. These values are stored in the Netmap, and in turn, this Netmap is associated with both a Policy and an HTTP Adapter.

Use the following procedure to create a Netmap.

1. Log into the user interface for Sterling Secure Proxy.
2. Select **Configuration** from the menu bar and then select **Actions** → **New Netmap** → **HTTP Netmap**.
3. Enter a name for the Netmap.
4. To define an inbound node definition, select the **Inbound Nodes** tab and click **New**.
5. Specify the following values:
   - Inbound Node Name
   - Peer Address Pattern
   - Policy
6. Click **OK**.
7. To define an outbound node definition, select the **Outbound Nodes** tab and click **New**.
8. Specify the following values:
   - Outbound Node Name
   - Primary Destination Address
   - Primary Destination Port

9. Click **OK**.
10. Click **Save**.

Figure 5-6 shows the creation of the inbound node for the new Netmap.

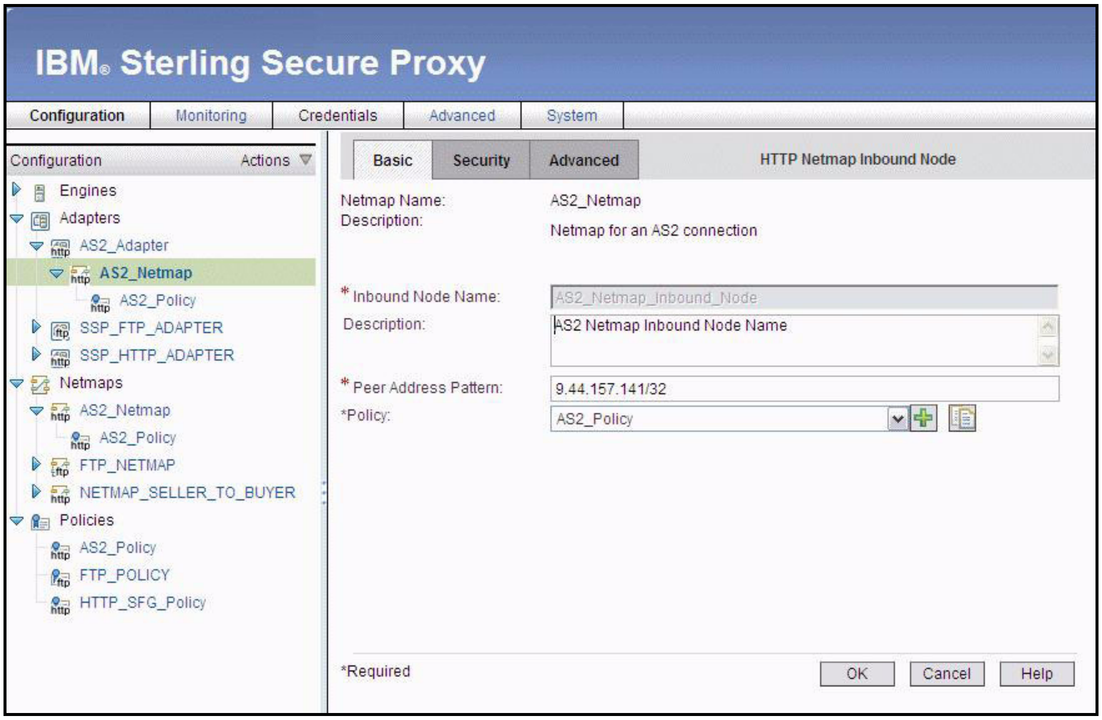


Figure 5-6 Details of the inbound node in the Netmap

Figure 5-7 shows the creation of the outbound node for the new Netmap.

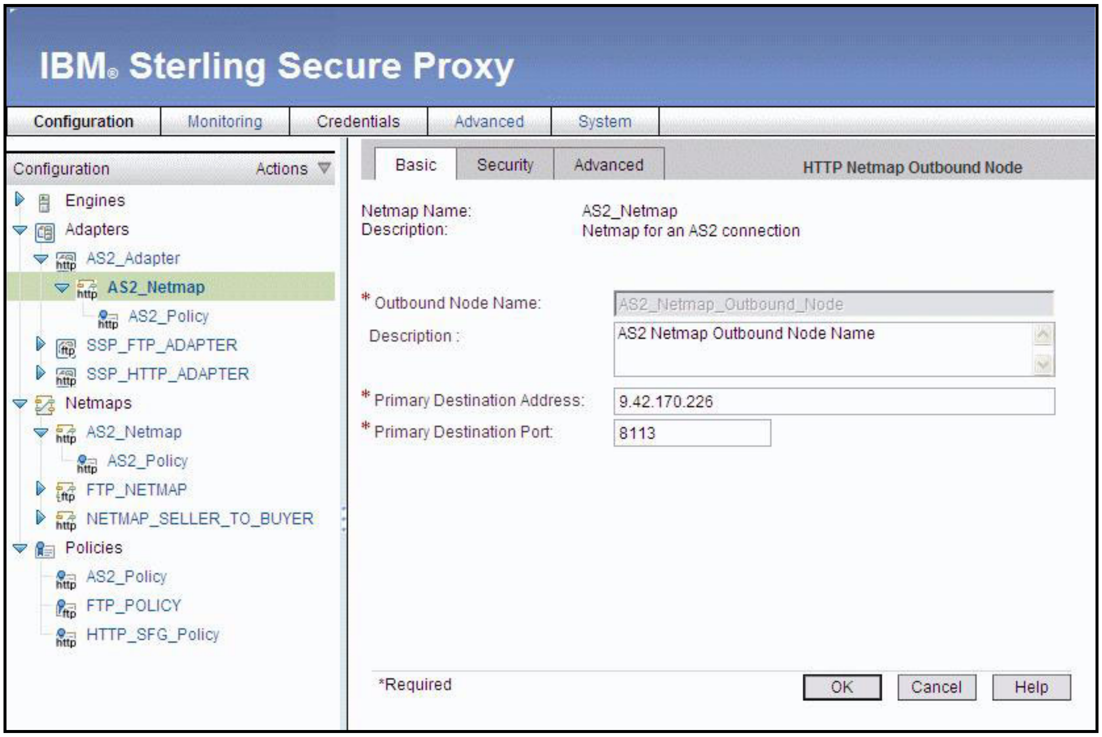


Figure 5-7 Details of the outbound node in the Netmap

Configure an HTTP Adapter

After the Policy and Netmap are created, the last step in configuring Sterling Secure Proxy is to define the HTTP Adapter to use in our scenario.

In Sterling Secure Proxy, the HTTP Adapter definition specifies the information necessary for HTTP connections both to and from the application.

Use the following procedure to define the HTTP Adapter for this scenario.

1. Login to the Sterling Secure Proxy user interface.
2. Select **Configuration** from the menu bar and then select **Actions** → **New Adapter** → **HTTP**.
3. Specify the proper values for the following items:
   – Adapter Name
   – Listen Port
   – Netmap
   – Standard Routing Node
   – Engine
4. Click **Save**.

Figure 5-8 shows our HTTP Adapter configured for this scenario.

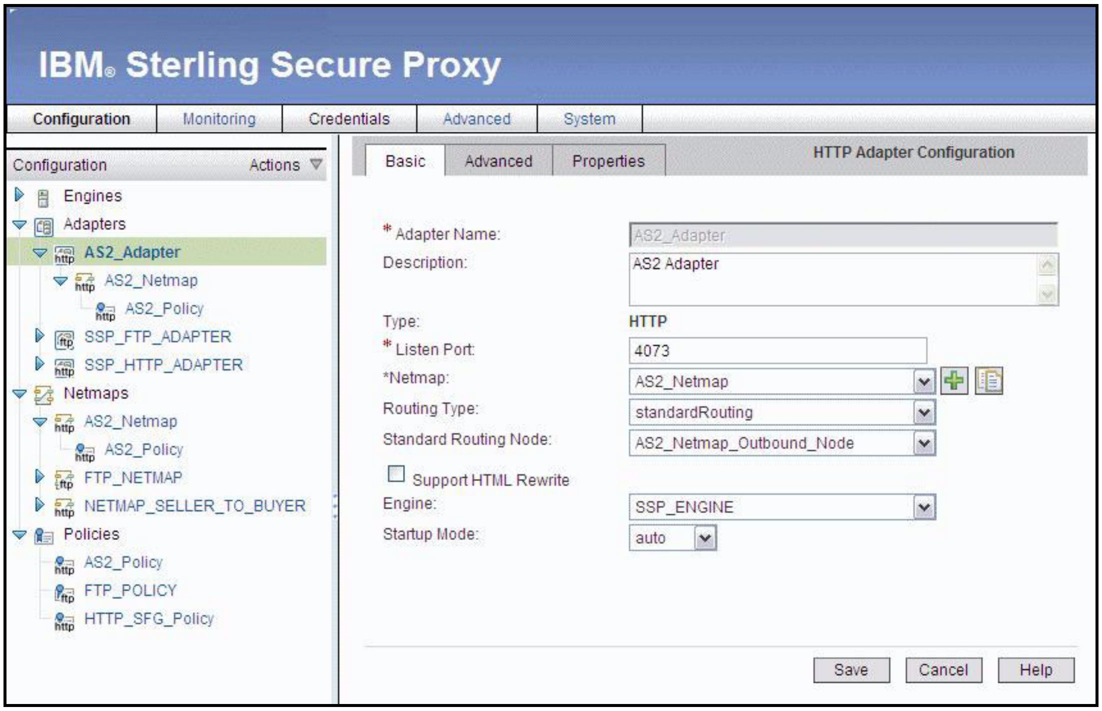


Figure 5-8 Configuration of the HTTP Adapter in Sterling Secure Proxy

After we complete the creation of the Policy, the Netmap, and the HTTP Adapter, we can then move on to the configuration of Sterling B2B Integrator.

Step 2: Configure AS2 in Sterling B2B Integrator

To prepare Sterling B2B Integrator for this scenario, the following tasks must be performed:

- Configure the certificates for the Organization and AS2 Partner.
- Configure the AS2 Organization.
- Configure the AS2 Trading Partner.
- Create the AS2 Trading Relationship.
- Test the AS2 Setup to ensure that it functions correctly.

The Sterling B2B Integrator documentation contains much more detail about configuring the application for AS2 communications. For full information about setting up AS2 in particular, see the “Standards Library Using AS2” sections of the Sterling B2B Integrator documentation:

http://help.sterlingcommerce.com/SB2BI52/topic/com.ic.customization.doc/SI_52_PDF/Stds801_Using_AS2.pdf

Next we detail each of these steps for our scenario.

Configure the certificates for the Organization and AS2 Partner

For AS2 communication to occur, the trading partners need to create and exchange digital certificates.

We must create the key certificate, and then share that certificate with our trading partner to allow them to decrypt the AS2 messages that we intend to transmit to them. There are many ways to create this certificate, both in Sterling B2B Integrator and through the Sterling Certificate Wizard.

For this scenario, we generate a simple self-signed certificate, and then present that to our trading partner as the key certificate for communications. In most real life situations, we would be using an actual CA certificate which must be checked into Sterling B2B Integrator, and also shared with the partner.

Our trading partner, in this case the fictitious ITSORetail Corporation, is also required to supply us with their certificate to check into the Trusted store in Sterling B2B Integrator.

When we have the certificate from the partner, we need to save it to a location we can reach from the server, and then navigate to **Trading Partners** → **Digital Certificates** → **Trusted** in the Sterling B2B Integrator dashboard. **Select Check In New Certificate**, provide the required information, and then save the certificate to the Trusted store.

Figure 5-9 shows the summary for the certificate used in this example.

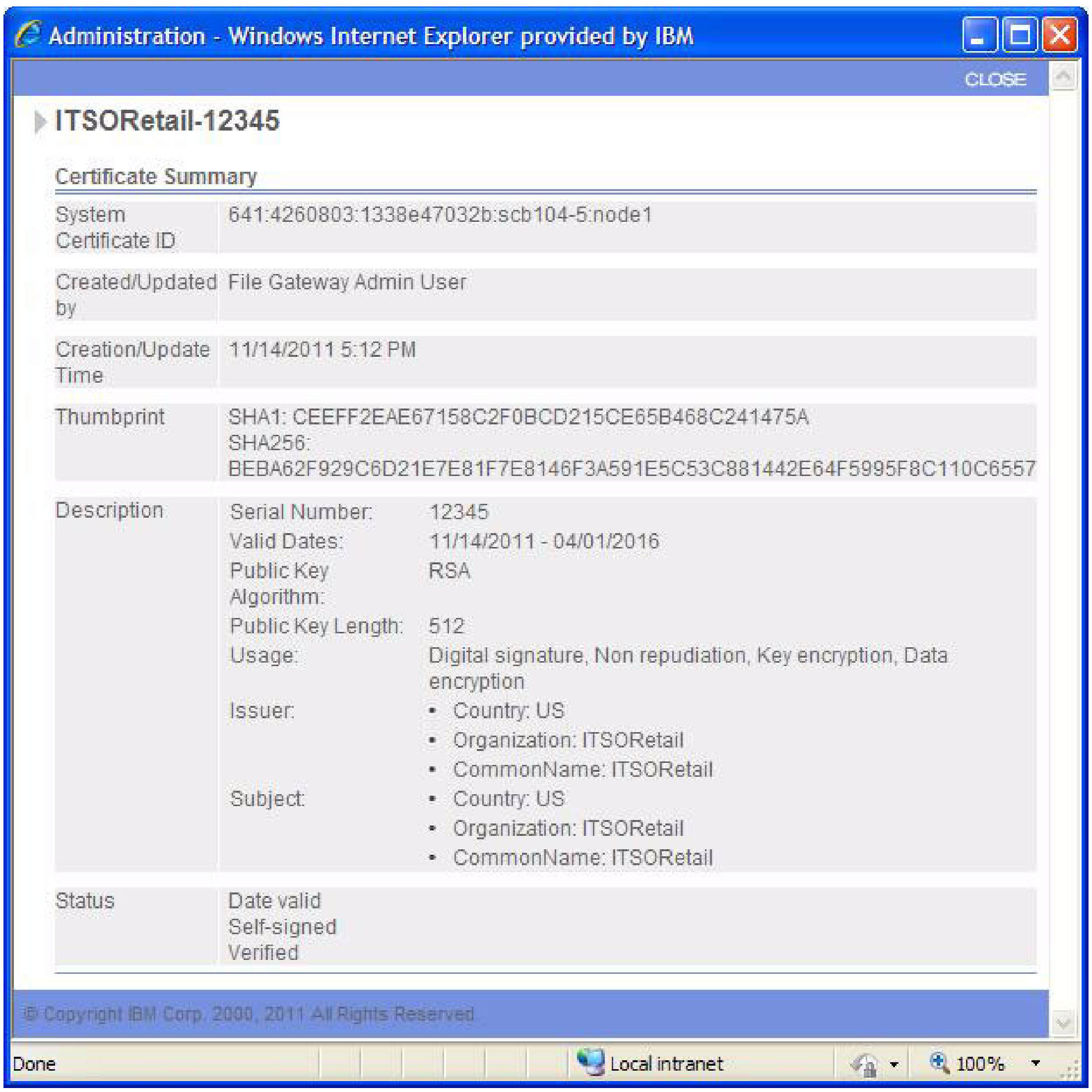


Figure 5-9 Configuration of the Trusted Certificate provided by ITSORetail Corporation

In this example, we are not actually using any certificates generated by a certificate authority. All of the certificates in use have been generated by Sterling B2B Integrator itself, which is sufficient for presenting this scenario and demonstrating the AS2 communications.

Configure the AS2 Organization

After the certificates have been created, exchanged, and checked in, we must now create our Organization. The Organization represents the internal group that is receiving and sending AS2 messages, and we must provide information about our fictitious ITSOHealthcare Corporation to configure the Organization.

To simplify the creation of the Organization, as well as the Trading Partner and Relationship, Sterling B2B Integrator has an AS2 Wizard that you can use to create and configure the proper values. This wizard properly creates the required profiles, identities, transports, channels, packaging, business processes, and schedules to implement AS2. All of these items can be manually configured from the respective menu choices in the Sterling B2B Integrator dashboard, but using the AS2 Wizard can greatly simplify these tasks.

To reach the AS2 Wizard, log into the Sterling B2B Integrator dashboard and navigate to **Trading Partner** → **AS2**. To create the Organization, select **Create New AS2 partner or organization** on the first option screen that is displayed. We must select **Organization**, because the system does not allow us to create a Partner until we have created an Organization.

Figure 5-10 shows the settings for the Organization created for ITSOHealthcare Corporation.

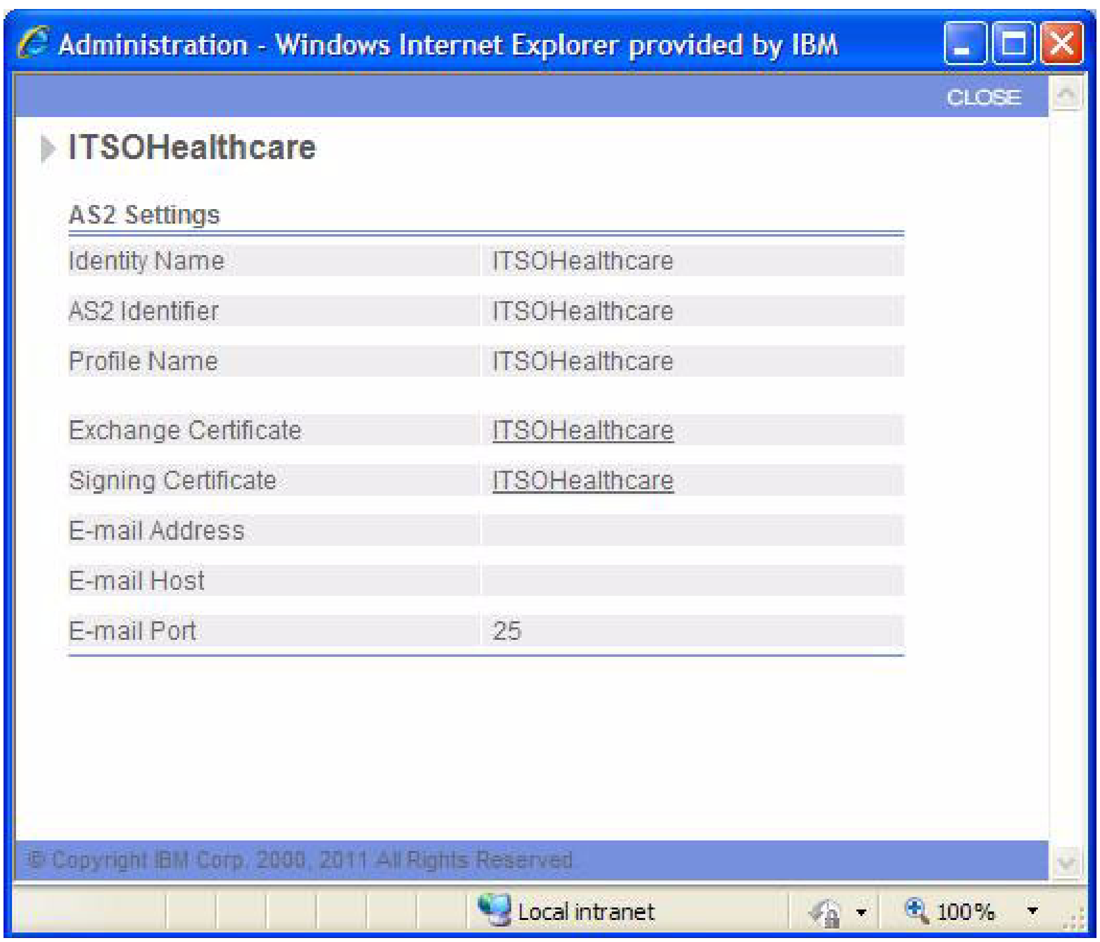


Figure 5-10 Configuration of the ITSOHealthcare Corporation AS2 Organization

Tip: In this simple example, we are using the same certificate for the exchange and the signing, and we have left the email information blank. These would normally be filled out for a real Organization, to allow them to receive notifications from Sterling B2B Integrator.

Configure the AS2 Trading Partner

After the Organization is created, we must now create a Trading Partner for AS2.

Again, log into the Sterling B2B Integrator dashboard, and navigate to **Trading Partner** → **AS2**. We again select **Create New AS2 partner or organization**. This time, however, on the first option screen, we select **Partner** as the type of Profile we are creating.

Fill out the required fields, including the certificate that was received from the partner and checked into Sterling B2B Integrator.

Figure 5-11 shows the configuration for the ITSORetail Corporation Profile.

Figure 5-11 Configuration for the AS2 Profile for ITSORetail Corporation

Note that the Profile is using the Trusted certificate we checked in earlier, and that the End Point is configured for the delivery of messages. It is a fictitious IP address for the purposes of this example.

Configure the AS2 Trading Relationship

In order to link the Organization with the Partner, we must now create a Relationship between them. In Sterling B2B Integrator, there can be multiple Organizations and multiple Partners, and the Relationship defines the linkage between them.

Log into the Sterling B2B Integrator dashboard, and navigate to **Trading Partner** → **AS2** and select **Create New AS2 trading relationship**.

For the purpose of this scenario, it is particularly important to select mailboxes as the storage type of the inbound and outbound AS2 messages. It ensures that we are able to use Sterling File Gateway for routing of the messages after they are received by Sterling B2B Integrator and placed in the proper mailbox.

Figure 5-12 shows the configuration screen where the option is presented to select mailboxes to store the AS2 messages.

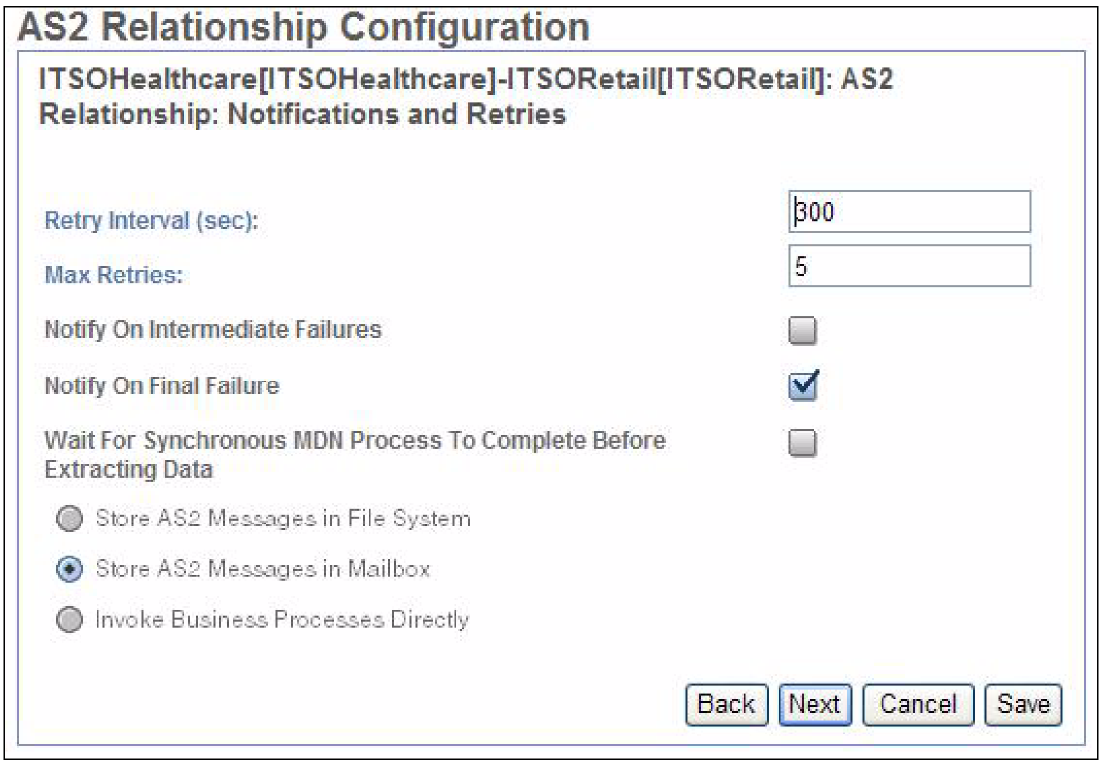


Figure 5-12 Screen to configure mailbox storage of AS2 messages

Figure 5-13 shows the screen where we can change the mailboxes that are used by Sterling B2B Integrator for storing these messages. When using AS2 messages with Sterling File Gateway, it is best to accept the default locations.

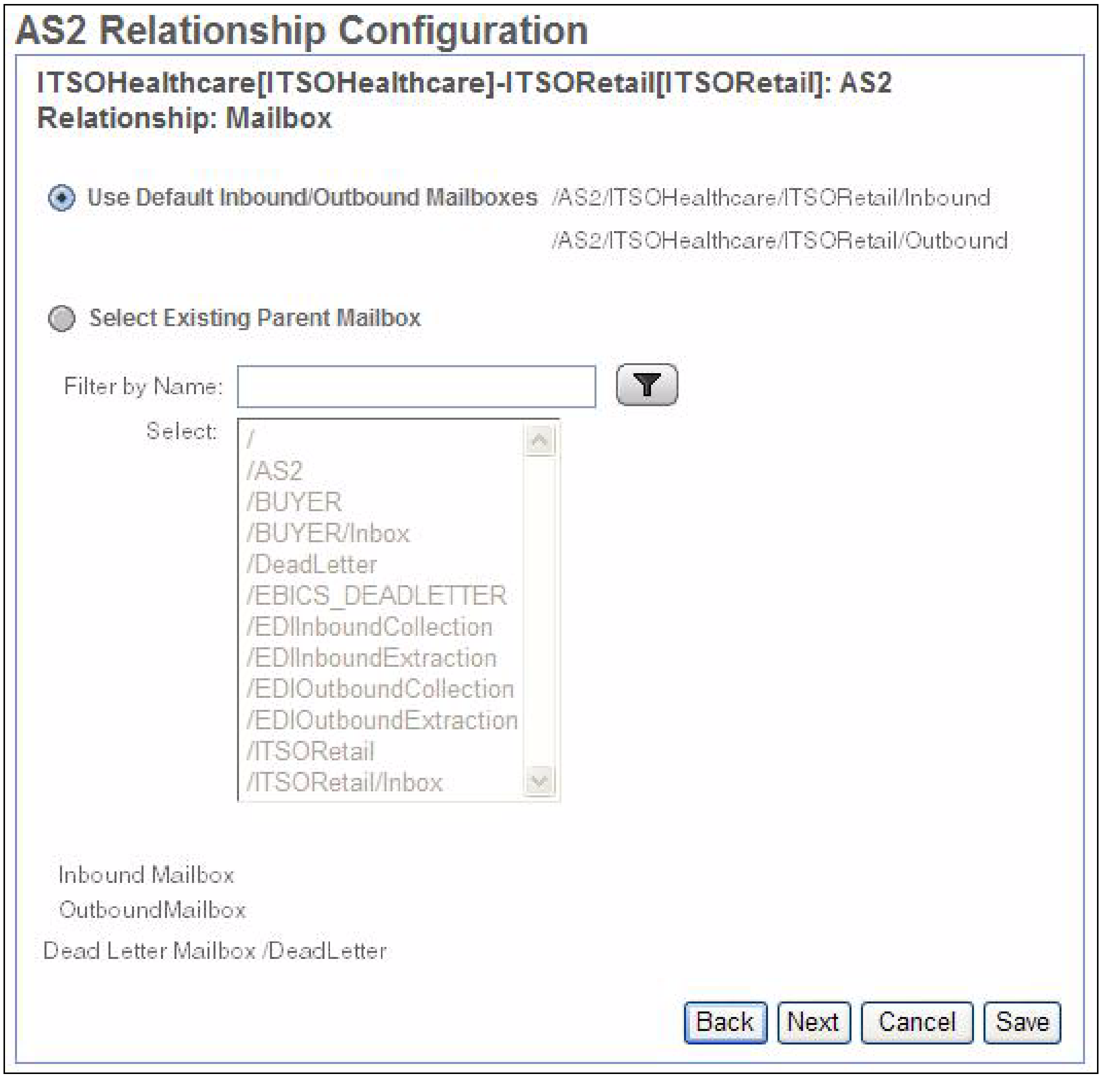


Figure 5-13 Selection of the mailboxes for the Relationship

Test the AS2 Setup to ensure that it functions correctly

After the Organization, Partner and Relationship have been created, we can then test the AS2 communications between Sterling B2B Integrator and the trading partner.

Login to the Sterling B2B Integrator dashboard, and navigate to **Trading Partner** → **AS2**. Select **Search Relationships**, and enter the values to search on, which in this case would be “ITSOHealthcare” for the Organization and “ITSORetail” for the Partner. On the Summary screen that is displayed, there is a test icon displayed under the Select heading.

Figure 5-14 shows an example of this screen for the Organization and Partner in this scenario.

Figure 5-14 Summary screen after searching for Relationship

Clicking the test icon initiates a communications test with the partner, and confirms that our trading relationship has been configured correctly and is functioning normally. If any errors are encountered, we must go back and correct them before proceeding to the next step.

Figure 5-15 shows an example running this test with the partner. Note that the last step, the receipt of the MDN, failed. It is an example of a failure that would need to be corrected before the partner could exchange messages.

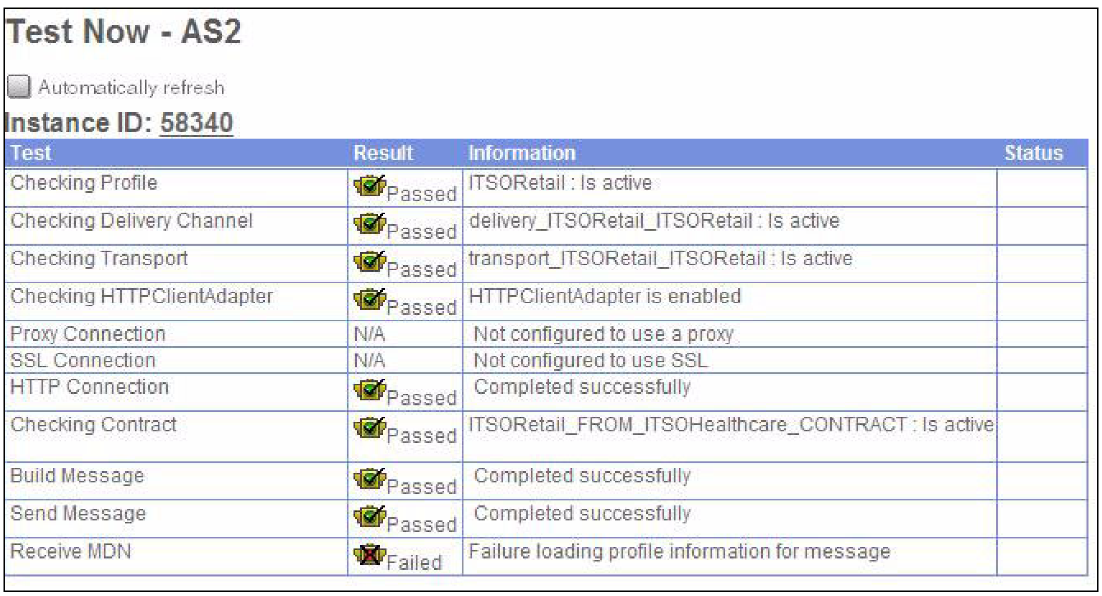


Figure 5-15 Example of the results when testing an AS2 partner configuration

This concludes the setup for Sterling B2B Integrator for AS2 communications. We now move on to Sterling File Gateway.

Step 3: Configure Sterling File Gateway for routing

To prepare Sterling File Gateway for this integration scenario, the following tasks must be performed:

- Configure the Community
- Configure the Partner
- Configure the AS2 Partners Group
- Create the Routing Channel Template for AS2 Inbound
- Create the Routing Channel Template for AS2 Outbound
- Create the Route Channel for AS2 Inbound
- Create the Route Channel for AS2 Outbound

Next we look at each of these tasks in detail.

Configure the Community

The first step in configuring Sterling File Gateway is to configure the community to be used for the fictitious ITSOHealthcare Corporation. This community is used as a way of organizing all trading partners into a logical grouping, which can speed up the onboarding process for the partner. In addition, the community can also be used to help define the protocols and methods used by those partners.

Use the following procedure to create the community used in this scenario.

1. Login to the Sterling File Gateway dashboard.
2. Navigate to **Participants** → **Communities**.
3. Under the **Select** menu, select the link to add a new community.
4. Enter the name for the community, and select the options for the partner both initiating and listening for protocol connections.

Figure 5-16 shows the community created for “ITSOHealthcase Corporation.”

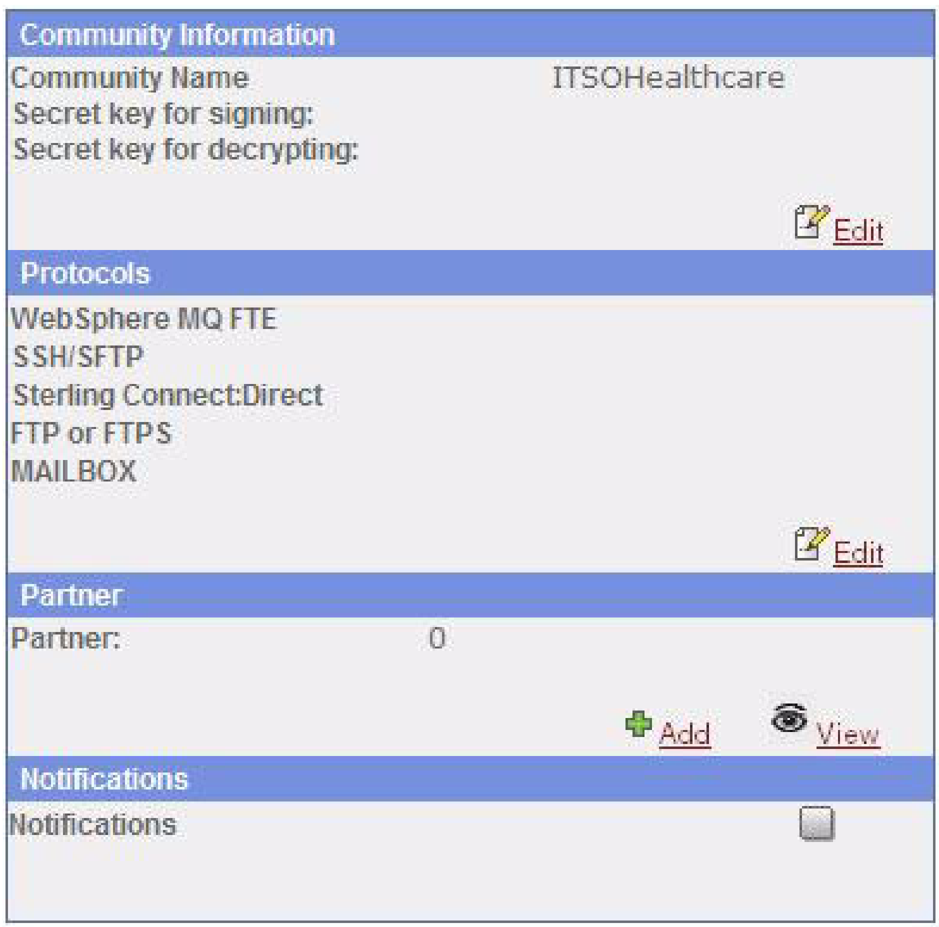


Figure 5-16 Sterling File Gateway Community for ITSOHealthcare Corporation

Note that this community is showing a Custom Protocol for WebSphere MQ FTE. There is another option for configuring this scenario, which we discuss in more detail later.

Configure the AS2 Partner

The configuration of the AS2 Partner, in this case "ITSORetail Corporation," is done through the normal Sterling B2B Integrator dashboard, and is described in detail in the previous step.

There are two important considerations when creating this partner:

- Ensure that the partner is configured to use mailbox storage of the AS2 messages to allow Sterling File Gateway to route them.
- Ensure that the AS2 partner is using the default mailboxes for the storage of the messages, which simplifies the configuration of the routing channel template.

There is one additional task that needs to be completed. When an AS2 partner is created, there is a corresponding user created in Sterling B2B Integrator. This user must be added to the File Gateway Partners Users group.

1. Login to the Sterling B2B Integrator dashboard.
2. Navigate to **Accounts** → **User Accounts** and search for the user that corresponds to the partner. In this scenario, this would be "ITSORetail."
3. Edit this account, and on the Groups page, add File Gateway Partner Users, and then save the changes.

Figure 5-17 shows this group added for our "ITSORetail" AS2 partner.

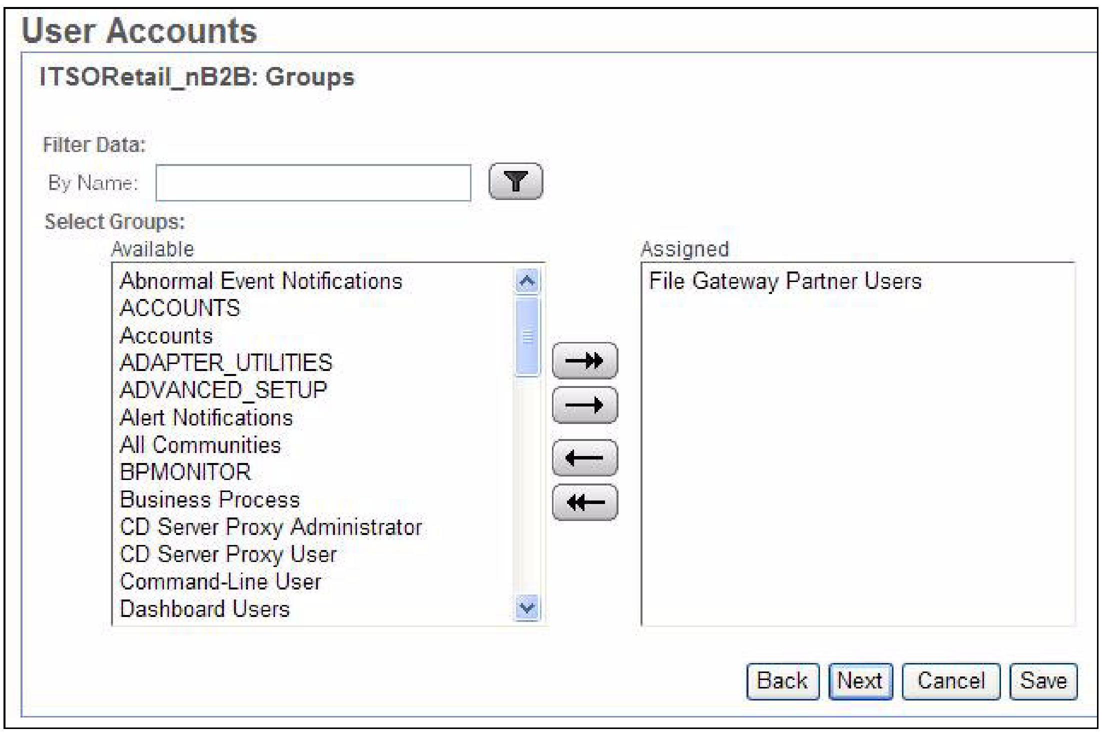


Figure 5-17 Adding the File Gateway Partner Users group to the AS2 partner user

An important fact to remember is that the AS2 partner must continue to be configured with Sterling B2B Integrator, rather than managed in Sterling File Gateway. Hence, any changes to the profile would need to be done from the Sterling B2B Integrator dashboard.

Configure the AS2 Partners group

After the partner is configured and added to the user group, we create a partner group to assist in organizing AS2 partners in Sterling File Gateway. It is not a required step, particularly in this case with only one AS2 partner, but it can simplify the management of the partners.

Use the following procedure to configure the AS2 Partners group.

1. Login to the Sterling File Gateway dashboard.
2. Navigate to **Participants** → **Groups**.
3. Create a new group named AS2 Partners.
4. Select Add Partners, and add the AS2 partner to the group. In this scenario, we are adding the “ITSORetail” partner to the group “AS2 Partners.”

Figure 5-18 shows an example of the group, with the new AS2 Partners group highlighted, and the ITSORetail partner added to the group.

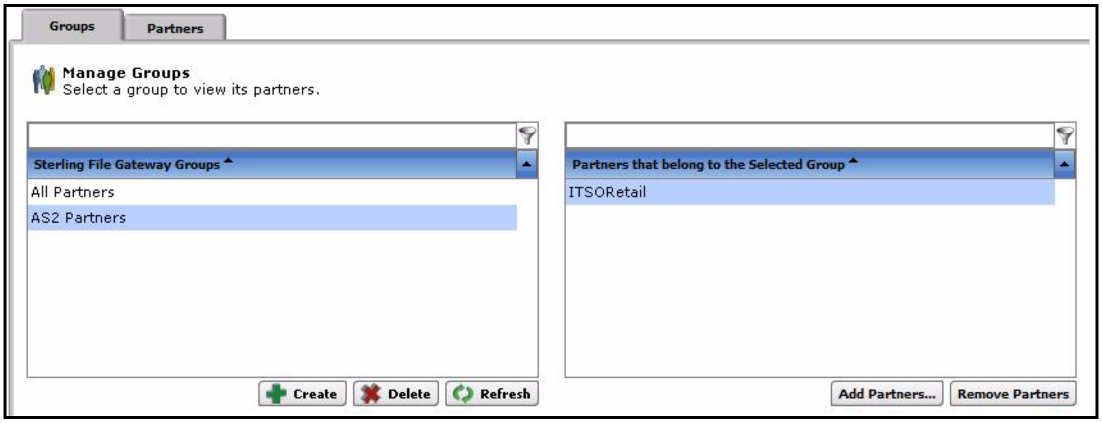


Figure 5-18 ITSORetail AS2 partner added to the AS2 Partners group

As noted above, it is not a required step, but it can make managing the partners easier, particularly as the number of partners increases.

Create the Routing Channel Template for AS2 Inbound

In order to route the messages, we need to create two routing channel templates, one for inbound and one for outbound. The routing channel template provides the definition and structure for the routing of files through Sterling File Gateway, and it is required before the actual routes can be created. It also controls the protocols and formats for the files, as well as the partners that can participate in the movement of files.

A routing channel template can be either static, where the values are defined at creation time and are linked with a particular resource, or dynamic, which allows Sterling FIle Gateway to determine, based on the definition, the consumer for the file. In this scenario, for AS2 transfers, we are creating a static routing channel template, but one that can be used for any AS2 partner.

First, we create the inbound template. Use the following procedure.

1. Login to the Sterling File Gateway dashboard.
2. Navigate to **Routes** → **Templates**.
3. Create a new template named AS2 Inbound.
4. Select Static as the type of template, and do not specify any special character handling.
5. Select AS2 Partners as the Producer Group, and All Partners as the Consumer Group.
6. For the Producer Mailbox Path, enter “/AS2/ITSOHealthcare/${ProducerName}/Inbound”.
7. Under Producer File Structures, select Unknown and use ‘.+’ as the regular expression, and no facts are required.

8. For the Consumer Mailbox Path, enter “/${ConsumerName}/Inbox”.
9. For the Consumer File Structure, select Unknown, with a file name format of `${ProducerFilename}`.

Figure 5-19 shows the routing channel template created for AS2 Inbound.

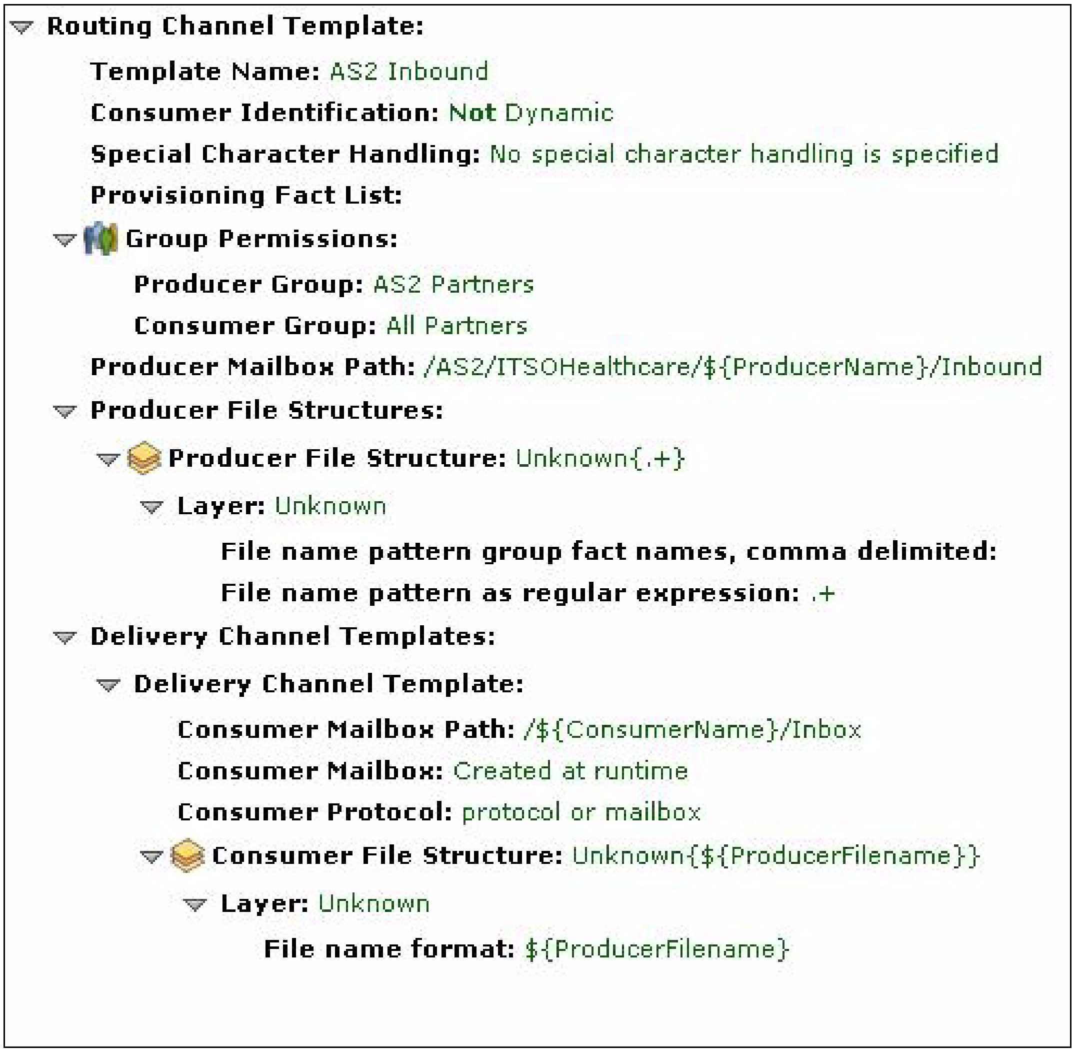


Figure 5-19 Routing Channel Template for AS2 Inbound

Create the Routing Channel Template for AS2 Outbound

The next step is to create a very similar routing channel template, but this time for outbound AS2 messages.

We now create the inbound template. Use the following procedure.

1. Login to the Sterling FIle Gateway dashboard.
2. Navigate to **Routes** → **Templates**.
3. Create a new template named AS2 Outbound.
4. Select Static as the type of template, and do not specify any special character handling.
5. Select All Partners as the Producer Group, and AS2 Partners as the Consumer Group.
6. For the Producer Mailbox Path, enter /${ProducerName’.

7. Under Producer File Structures, select Unknown and use ‘.+’ as the regular expression, and no facts are required.
8. For the Consumer Mailbox Path, enter “/AS2/ITSOHealthcare/${ProducerName}/Outbound”.
9. For the Consumer File Structure, select “Unknown,” with a file name format of `${ProducerFilename}`.

Figure 5-20 shows the routing channel created for AS2 Outbound.

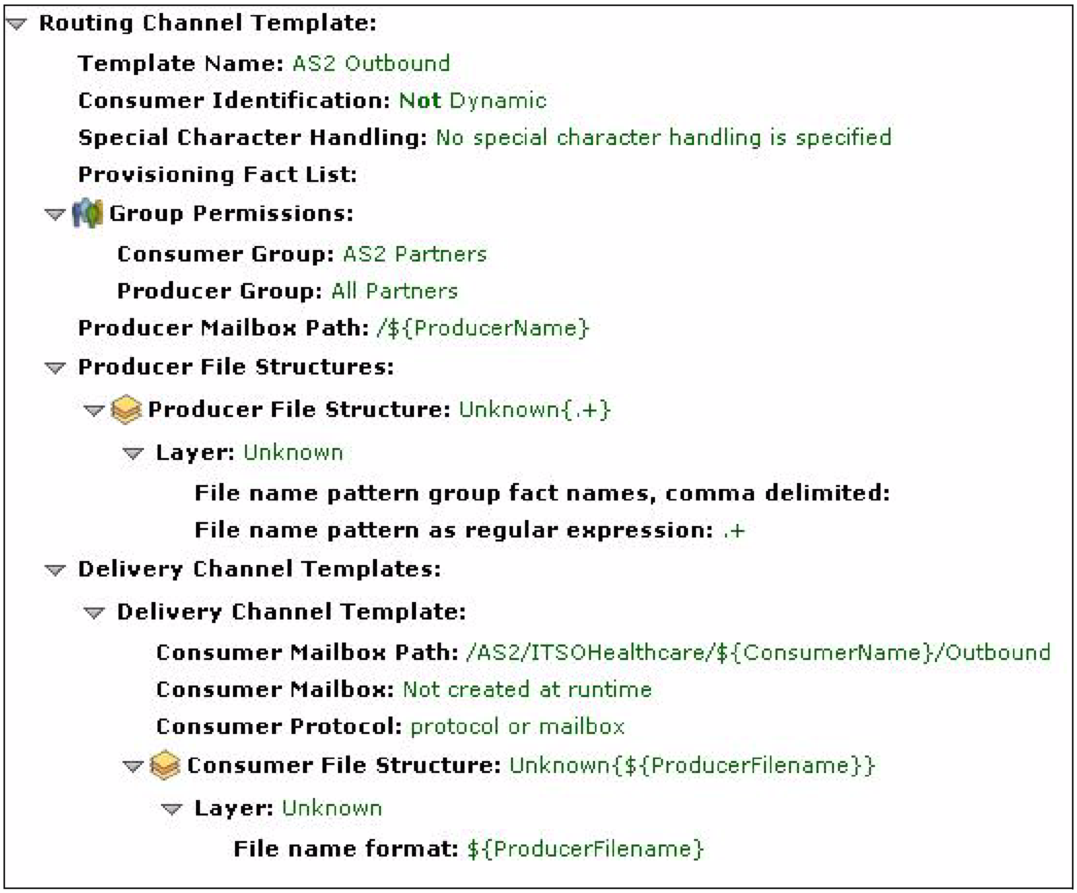


Figure 5-20 Routing Channel Template for AS2 Outbound

Create the Route for AS2 Inbound from ITSORetail

After the templates are created, we must now create an actual route that uses those templates to do the actual routing of the files.

Use the following procedure to create the inbound route.

1. Login to the Sterling File Gateway dashboard.
2. Navigate to **Routes** → **Channels**.
3. Click **Create** to create the new Channel.
4. For the Routing Channel Template, select the AS2 Inbound template.
5. For the Producer, select the AS2 partner, in this case, the ITSORetail AS2 partner created earlier.
6. For the Consumer, select any other partner, which does not have to be an AS2 partner.
7. For this scenario, we select the same partner, to simplify the configuration in the next step.

Figure 5-21 shows the routing channels created in Sterling File Gateway, with the new inbound route for the ITSORetail AS2 partner highlighted.

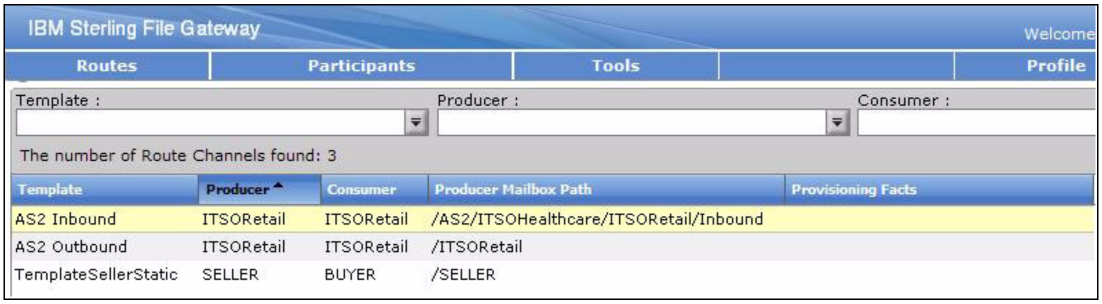


Figure 5-21 Routing Channels with the ITSORetail inbound channel highlighted

Create the Route for AS2 Outbound to ITSORetail

The final step in the Sterling File Gateway configuration is to configure the route for AS2 Outbound. It is quite similar to the AS2 Inbound Route, but allows a message to be routed back to Sterling B2B Integrator for transmission by AS2 to the partner. We could eventually use this outbound route as a method for returning a message back from the backend application to the trading partner.

Use the following procedure to create the outbound route.

1. Login to the Sterling File Gateway dashboard.
2. Navigate to **Routes** → **Channels**.
3. Click **Create** to create the new Channel.
4. For the Routing Channel Template, select the AS2 Outbound template.
5. For the Producer, select any partner, which is not required to be an AS2 partner. In this case, we select ITSORetail, which simplifies the configuration in the next step.
6. For the Consumer, also select the ITSORetail partner, which is the AS2 partner that the message will be routed to for transmission by Sterling Integrator by AS2.

Figure 5-22 shows the routing channels created in Sterling File Gateway, with the new outbound route for the ITSORetail AS2 partner highlighted.

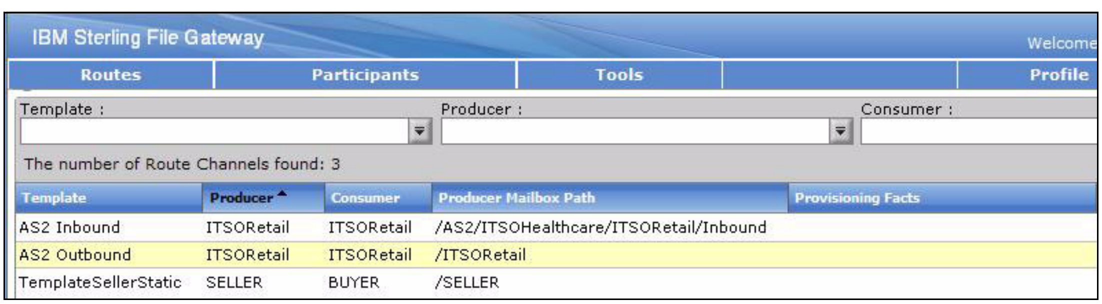


Figure 5-22 Routing Channels with the ITSORetail outbound channel highlighted

As mentioned earlier, this outbound route is not used as part of this scenario, but is left as an exercise for the reader.

Alternate configuration for Sterling File Gateway using a Custom Protocol

After the message moves through Sterling File Gateway and is processed by Sterling B2B Integrator, there are two ways that we could move this message to the WebSphere MQ for additional processing and eventual delivery to the backend application:

- As part of the business process that is called when Sterling B2B Integrator de-envelopes the message, use the WebSphere Suite Adapters in Sterling B2B Integrator to place the message into the WebSphere MQ queue for the next step of the processing.
- After the message is processed by Sterling B2B Integrator, place it back into a maibox, and define a route that uses a Custom Protocol in Sterling File Gateway to move the message to WebSphere MQ for the next step in processing.

In this scenario, as a way of demonstrating some of the additional capabilities of Sterling B2B Integrator, we are using the first option, and we configure these components in the next step.

For more information about configuring a Custom Protocol for Sterling File Gateway, see the documentation for that product. In particular, the section on “Extending the Capabilities” has a detailed description of implementing a Custom Protocol, as well as some examples.

In addition, the IBM Redbooks *IBM Sterling Managed File Transfer Integration with WebSphere Connectivity for a Multi-Enterprise Solution*, SG247927 has a very detailed example of implementing a Custom Protocol in Chapter 7, “External transfers using IBM WebSphere Message Broker and IBM Sterling File Gateway’.

Step 4: Configure message processing in Sterling B2B Integrator

Now that Sterling File Gateway has been configured to route the message to the proper location, we need to configure Sterling B2B Integrator to process the message and deliver it to the WebSphere MQ queue for the additional processing in WebSphere Message Broker.

As mentioned in the previous section, the final delivery of the message to the queue could also be accomplished by using the Custom Protocol functions of Sterling File Gateway. In this scenario, however, we use the Sterling B2B Integrator to demonstrate its ability to interact with WebSphere MQ.

In order to accomplish these integrations, we need to complete the following tasks:

- Configure the inbound Envelopes for the trading partner.
- Create a Routing Rule to check the mailbox for a message and execute the initial business process.
- Specify the correct maps in the envelopes for the compliance check and for the transformation to the canonical format.
- Create a Business Process that sends the document to the WebSphere MQ queue when called.
- Configure the outbound Envelopes to return the acknowledgement (EDI 999) back to the trading partner.
- Create and schedule a Business Process to clean up expired mailbox messages.

We now review each of these tasks and the component in detail.

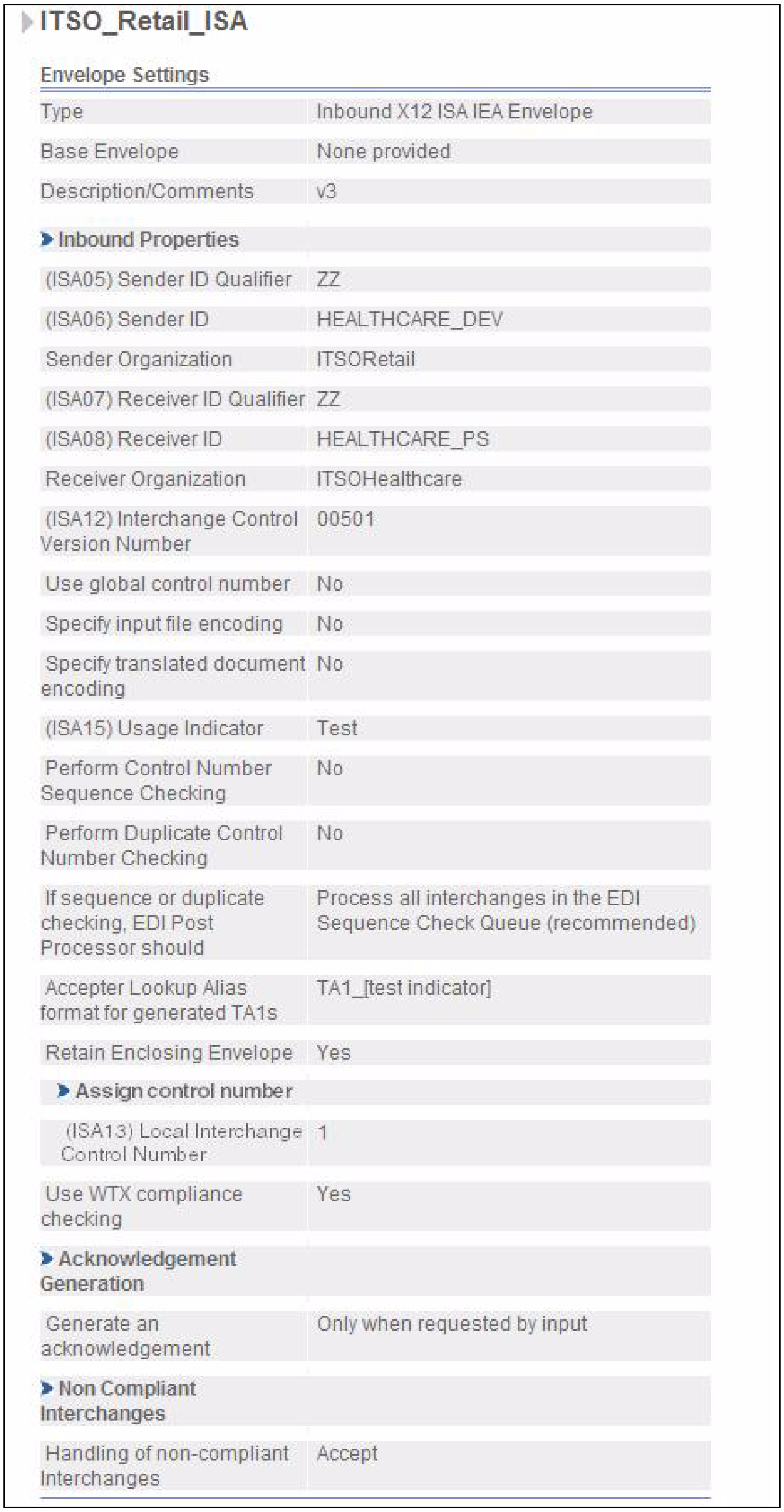
Configure the inbound envelopes for the trading partner

In order to process the message from the trading partner, we need to configure document envelopes in Sterling B2B Integrator to allow the system to de-envelope the message and pass the resultant document on for the additional processing required.

There are three distinct envelopes that need to be created for this message, which correspond to the usual X12 standard:

- ISA-IEA
- GS-GE
- ST-SE

Figure 5-23 is the ISA-IEA envelope configured to process our inbound message from the trading partner. Note that in the envelope configuration, we have simplified this example by using only local (as opposed to global) control numbers, and we have also disabled duplicate and sequence checking.



ITSO_Retail_ISA

Envelope Settings

Type	Inbound X12 ISA IEA Envelope
Base Envelope	None provided
Description/Comments	v3
Inbound Properties	
(ISA05) Sender ID Qualifier	ZZ
(ISA06) Sender ID	HEALTHCARE_DEV
Sender Organization	ITSORetail
(ISA07) Receiver ID Qualifier	ZZ
(ISA08) Receiver ID	HEALTHCARE_PS
Receiver Organization	ITSOHealthcare
(ISA12) Interchange Control Version Number	00501
Use global control number	No
Specify input file encoding	No
Specify translated document encoding	No
(ISA15) Usage Indicator	Test
Perform Control Number Sequence Checking	No
Perform Duplicate Control Number Checking	No
If sequence or duplicate checking, EDI Post Processor should	Process all interchanges in the EDI Sequence Check Queue (recommended)
Accepter Lookup Alias format for generated TA1s	TA1_[test indicator]
Retain Enclosing Envelope	Yes
Assign control number	
(ISA13) Local Interchange Control Number	1
Use WTX compliance checking	Yes
Acknowledgement Generation	
Generate an acknowledgement	Only when requested by input
Non Compliant Interchanges	
Handling of non-compliant Interchanges	Accept

Figure 5-23 Inbound ISA-IEA envelope configured for ITSORetail Corporation

Figure 5-24 is the GS-GT envelope for this trading partner. Again note that we are using local control numbers rather than global, and we are not doing duplicate or sequence checking for simplicity. Note also in this envelope that we specify to return an acknowledgement to our trading partner, which generates the EDI 999 that we need to return.

ITSO_Retail_GS

Envelope Settings

Type	Inbound X12 GS GE Envelope
Base Envelope	None provided
Description/Comments	v6 - change to 999
Inbound Properties	
(GS01) Functional ID Code	HC
(GS02) Application Sender's Code	DEVELOPMENT
Sender Organization	ITSORetail
(GS03) Application Receiver's Code	PROFSERV
Receiver Organization	ITSOHealthcare
Use global control number	No
Assign control number	
(GS06) Local Group Control Number	1
(GS07) Responsible Agency Code	X
(GS08) Version, Release, Industry Identifier Code	005010X222A1
Perform Control Number Sequence Checking	No
Perform Duplicate Control Number Checking	No
Retain Enclosing Envelope	Yes
Functional Acknowledgement	
Handling of non-compliant Groups	Accept
Generate an Acknowledgement when this group is received.	Yes
Acknowledgement Settings	
Acknowledgement Format	999
Acknowledgement Detail Level	Group Level only
Send acknowledgement immediately	Yes
Accepter Lookup Alias format for generated 997/999s	99X

Figure 5-24 Inbound GS-GT envelope configured for ITSORetail Corporation

Figure 5-25 is the ST-SE envelope for our message. It is in this envelope that we configure the maps to be used to check the compliance and to transform the data contained in the message. Again, we are using local control numbers and have disabled duplicate and sequence checking.

ITSO_Retail_837_ST

Envelope Settings

Type	Inbound X12 ST SE Envelope
Base Envelope	None provided
Description/Comments	v5
Inbound Properties	
Sender ID	DEVELOPMENT
Sender Organization	ITSORetail
Receiver ID	PROFSERV
Receiver Organization	ITSOHealthcare
Transaction Set ID Code	837
Use global control number	No
Assign control number	
Local Transaction Set Control Number	1
Implementation Convention Reference(versions 4012 and above)	005010X279A1
Group Version Release ID Code	005010X279A1
Test Indicator	Test
Perform Control Number Sequence Checking	No
Perform Duplicate Control Number Checking	No
Retain Enclosing Envelope	Yes
Batch transactions received within a functional group into one output document	Yes
Translation/Compliance Checking	
Translate and/or Compliance Check Document	Yes
Map Name Mode	
Map Name Mode	Specify
Map Name	
Map Name	hipaa_837p_5010_toxml (WTX)
Compliance Checking	
Handling of non-compliant Transactions	Accept compliant and non-compliant transactions
Translate transactions	Yes

Figure 5-25 Inbound ST-SE envelope for ITSORetail Corporation

Also note that we have disabled the batching of like documents. It allows the system to create one document for each claim in the original message, and allows Sterling B2B Integrator to place one message onto the WebSphere MQ queue for each of those documents.

Create a Routing Rule to check the mailbox for a message

After the message is routed by Sterling File Gateway, it is placed in the mailbox `/ITSORetail/Inbox`. In order to trigger the business process that continue the processing of the message, we need to create a Routing Rule that can watch this mailbox, and then initiate the business process configured in the next section.

In the Sterling Integrator Dashboard, navigate to **Deployment** → **Mailboxes** → **Routing Rules**, and then select **Create a new Routing Rule**.

The Routing Rule should be configured to execute automatically on the mailbox `/ITSORetail/Inbox`, and it should run the business process named “EDIBootstrap” when it finds a message in the mailbox. This business process is a default process that is shipped as part of the base installation of Sterling B2B Integrator.

Figure 5-26 shows the Configuration screen for the new Routing Rule named ITSOHealthcare_Route.

Figure 5-26 Routing Rule configuration summary screen

With the Routing Rule configured to be evaluated automatically, it is executed when the scheduled evaluation occurs in Sterling B2B Integrator. Because of the addition of Sterling FIle Gateway, the system evaluates all automatic rules continuously, which ensures that the message is routed almost as soon as it is placed in the mailbox.

Specify the WebSphere Transformation Extender maps in the envelopes

When the de-enveloping runs in Sterling B2B Integrator, part of the process can include checking the compliance of the message, as well as transforming or translating the message into a different format.

In this scenario, we are using a HIPAA EDI 837 message, and as part of the processing of that message, we need to check the compliance to the published standards. If there is any variance from those standards, we need to inform our trading partner by returning an acknowledgement to them which lists the variance. For this, we are using the EDI 999 document type.

When checking the compliance of the message to the standard, there are various levels that the compliance can be compared to. In the native Sterling B2B Integrator application, we can check Level 4, 5, and 6 of compliance, but if the user wants to check compliance Level 1, 2, or 3, they can do so by using integration with WebSphere Transformation Extender compliance check, which supports compliance check Levels 1 to 4 out of the box.

In Sterling B2B Integrator, the compliance level is specified in the ST-SE envelope, and is controlled by selecting “yes” to the option of “Perform HIPAA compliance check” in the setup of the envelope. When “yes” is selected, the user can then specify the compliance check level.

> **Important:** If you are using WebSphere Transformation Extender transformation maps in the X12 de-envelope service, then you need to use the WebSphere Transformation Extender HIPAA compliance check maps for compliance checking regardless of the level.

One final step that is required to enable the use of the WebSphere Transformation Extender map as the compliance check is to enable its use in the ISA-IEA envelope for the message. The user must select “yes” for the option “Use WTX compliance checking.”

Along similar lines, the user can also call a WebSphere Transformation Extender map to do the transformation of the data into the canonical format as part of the de-enveloping of the message. After the WebSphere Transformation Extender map is deployed into Sterling B2B Integrator, the user can select it as the map to be called from the ST-SE envelope for the transformation, as it would be for any other map in Sterling B2B Integrator.

For more information about these settings, see the Pack for HIPAA EDI documentation at this website:

http://publib.boulder.ibm.com/infocenter/wtxdoc/v8r4m0/index.jsp?topic=%2Fcom.ibm.websphere.dtx.packhealthcareref.doc%2Fconcepts%2Fc_pack_healthcareref_Overview_of_the_Packs_for_Healthc.htm

Create a Business Process to send the document to the WebSphere MQ queue

When the EDIBootstrap process executes, called by the routing rule set up earlier, it calls the pre-configured EDIDeenvelope service, which in turn then calls the pre-configured X12DeenvelopeUnified business process. This process locates and uses the envelopes configured previously. The ST-SE envelope calls the business process that output the de-enveloped and transformed message to the WebSphere MQ queue for processing by WebSphere Message Broker.

Because we have the batching disabled in the ST-SE envelope, the X12DeenvelopeUnified process invokes the defined process for each of the transactions in the original message. In this case, it calls the process we are creating to put each resulting document onto the proper WebSphere MQ queue.

Figure 5-27 is the graphical view of this business process. Note that this process is quite simple. It basically opens a session and queue, puts the message, and then closes the queue and session.

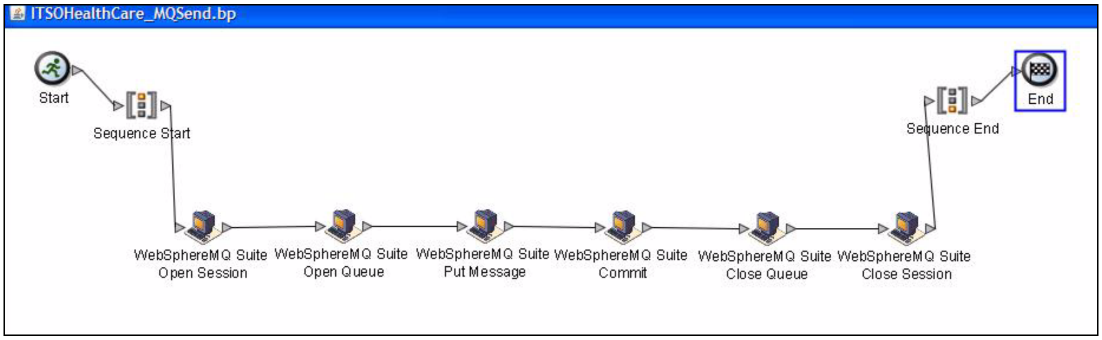


Figure 5-27 ITSOHealthCare_MQSend business process in graphical view

For clarity, this business process does not contain an "on fault" for the process flow. Even though this business process is quite simple, it is usually a good practice to anticipate potential failure modes and provide a method for resolving them, if possible. We leave the conception and addition of the "on fault" to the reader as an additional exercise.

Configure the outbound Envelopes to return the acknowledgement

In order to return the acknowledgement to the trading partner, we must specify in the GS-GE envelope to generate the acknowledgement, and then we must also configure outbound envelopes for the resultant acknowledgment.

In this scenario, we are returning the X12 standard EDI 999 to acknowledge the receipt of this message, and also to report on any compliance errors from the message.

Similar to the envelopes created for the inbound message, we must configure an envelope on each standard level for this EDI 999:

- ISA-IEA
- GS-GE
- ST-SE

As these envelopes are quite similar to those already presented for the inbound portion of the scenario, except of course for outbound rather than inbound, we do not replicate them here.

Create and schedule a Business Process to clean up expired messages

Sterling B2B Integrator does not come with a pre-configured process that remove mailbox messages that are expired. In order to ensure that we are removing messages that are no longer needed, we need to create and schedule a business process that can review the mailboxes used in this scenario, and then delete any messages that are either no longer extractable or expired based on their age.

This business process is much simpler than the prior process, in that it simply reviews the mailboxes in use in this scenario and then deletes any message in the mailbox that is no longer extractable.

Figure 5-28 shows the steps of this process.

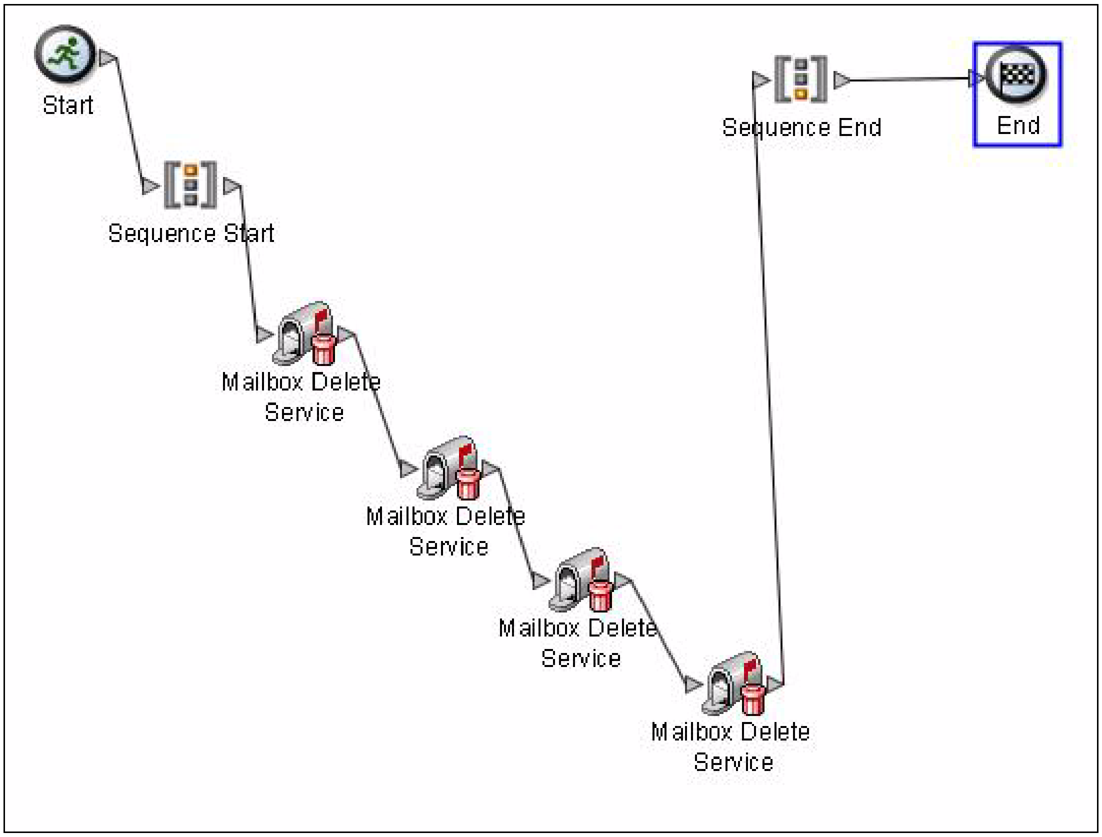


Figure 5-28 Graphical representation of the business process ITSOHealthcare_MBX_Cleanup

In order to ensure that the mailboxes are kept clean of expired messages, we need to schedule this process to run on a regular interval.

Figure 5-29 displays the schedule configuration for this business process.

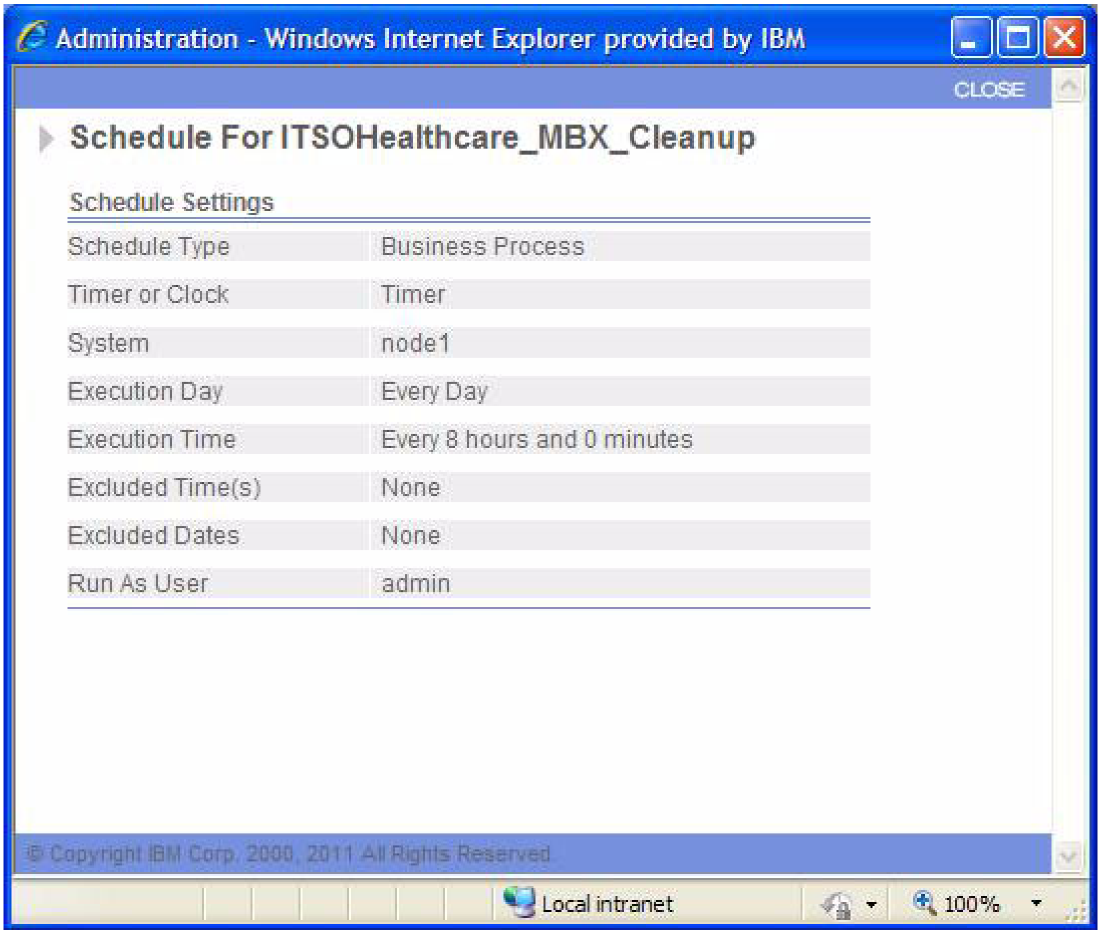


Figure 5-29 Configuration of the schedule for the process ITSOHealthcare_MBX_Cleanup

Note that this schedule is currently set to execute every 8 hours. Because this cleanup is simply removing messages that are expired, the longer schedule interval should be sufficient.

Step 5: Configure WebSphere MQ

In this scenario, we use WebSphere MQ for integration on various infrastructure components. For it to happen, a base connectivity infrastructure should be created. We need to perform the following tasks before any further backend integration:

- Create queue managers.
- Create listener objects on the queue managers.
- Create queue manager clusters.
- Create server connection channels.
- Create queues for application.

Create queue managers

Queue Managers could be created using the **crtmqm** command. Example 5-1 shows the syntax of this command.

Example 5-1 The crtmqm command syntax

```
crtmqm [-z] [-q] [-c Text] [-d DefXmitQ] [-h MaxHandles]
       [-md DataPath] [-g ApplicationGroup] [-ss | -sa | -sax | -si]
       [-t TrigInt] [-u DeadQ] [-x MaxUMsgs] [-lp LogPri] [-ls LogSec]
       [-lc | -ll] [-lf LogFileSize] [-ld LogPath] QMgrName
```

For more information about all of the parameters for the **crtmqm** command, see this website:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/topic/com.ibm.mq.amqzag.doc/fa15650_.htm

ESB01 Queue Manager for Backend ESB-01

The command in Example 5-2 is executed on server `esb01.itso.ibm.com` to create queue manager ESB01.

Example 5-2 Create queue manager ESB01

```
>crtmqm ESB01
WebSphere MQ queue manager created.
Directory 'C:\Program Files (x86)\IBM\WebSphere MQ\qmgrs\ESB01' created.
Creating or replacing default objects for ESB01.
Default objects statistics : 68 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
```

ESB02 Queue Manager for Backend ESB-02

The command in Example 5-3 is executed on server `esb02.itso.ibm.com` to create queue manager ESB02.

Example 5-3 Create queue manager ESB02

```
>crtmqm ESB02
WebSphere MQ queue manager created.
Directory 'C:\Program Files (x86)\IBM\WebSphere MQ\qmgrs\ESB02' created.
Creating or replacing default objects for ESB01.
Default objects statistics : 68 created. 0 replaced. 0 failed.
Completing setup.
Setup completed
```

B2B02 Queue Manager for Sterling and ESB integration

The command in Example 5-4 is executed on server `b2b02.itso.ibm.com` to create queue manager B2B02.

Example 5-4 Create queue manager B2B02

```
>crtmqm B2B02
WebSphere MQ queue manager created.
Directory 'C:\Program Files (x86)\IBM\WebSphere MQ\qmgrs\B2B02' created.
Creating or replacing default objects for B2B02.
Default objects statistics : 68 created. 0 replaced. 0 failed.
Setup completed.
```

Figure 5-30 illustrates the WebSphere MQ topology used for implementation in this scenario.

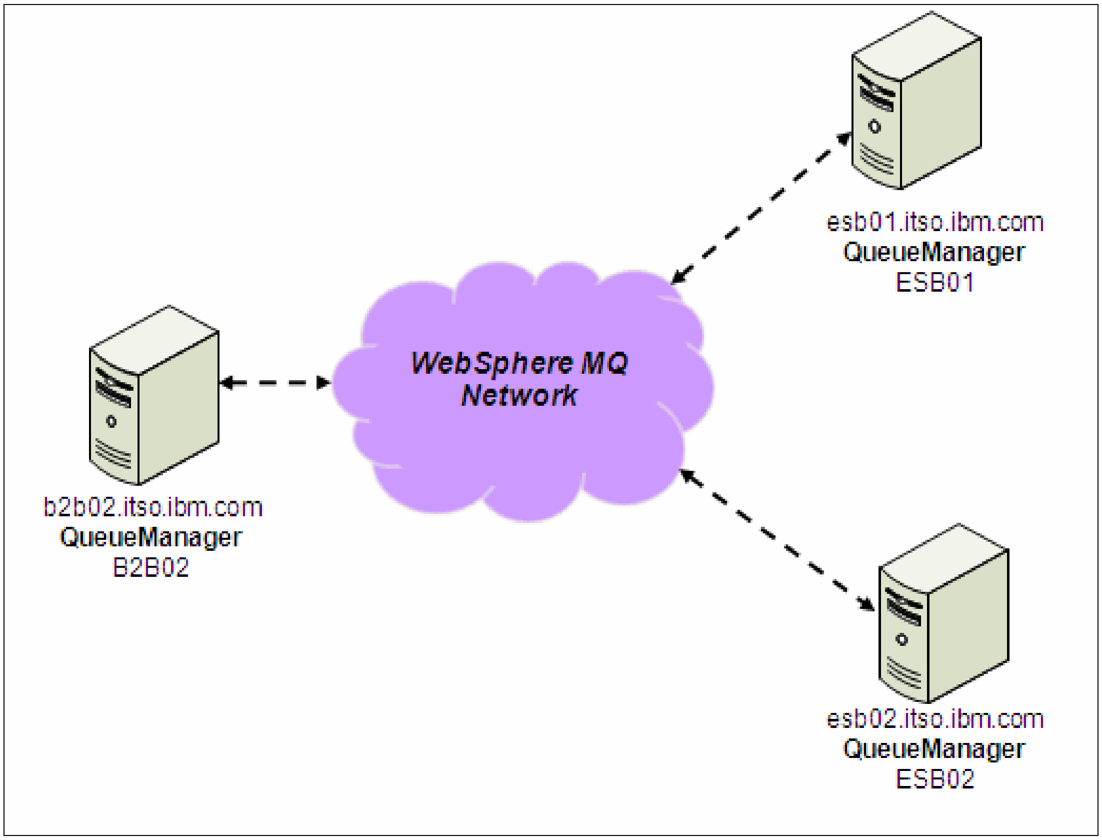


Figure 5-30 WebSphere MQ topology

Now that all of the queue managers are created, we need to start them using the **strmqm** command. Example 5-5 shows the syntax of this command.

Example 5-5 The syntax of crtmqm command

```
strmqm [-z] [-a | -c | -r | -x] [-d none|minimal|all] [-f]
       [-ns] [-ss | -si] [QMgrName]
```

For more information about all of the parameters for the **strmqm** command, see the following website:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/topic/com.ibm.mq.amqzag.doc/fa16090_.htm

The commands used to start the various queue managers are shown in Example 5-6.

Example 5-6 Start queue manager commands

```
>strmqm ESB01
>strmqm ESB02
>strmqm B2B02
```

Create listener objects on the queue manager

In this section, we create listener objects on various queue managers. We do this by issuing commands on the WebSphere MQ **runmqsc** console. Also, the listeners control is set to be queue manager, so that listeners start/stop based on the control of the queue manager.

For more information about defining listeners, see the following website:

http://publib.boulder.ibm.com/infocenter/wmqv6/v6r0/topic/com.ibm.mq.csqzaj.doc/sc11050_.htm

Example 5-7 to Example 5-9 illustrate commands to create listeners on various queue managers.

Example 5-7 Create listener on the queue manager ESB01

```
DEFINE LISTENER ('ESB01.LISTENER')
TRPTYPE(TCP) IPADDR('esb01.itso.ibm.com')
PORT(1414) BACKLOG(0) DESCR('ESB01 Queue Manager Listener')
CONTROL(QMGR) REPLACE
```

Example 5-8 Create listener on the queue manager ESB02

```
DEFINE LISTENER ('ESB02.LISTENER')
TRPTYPE(TCP) IPADDR('esb02.itso.ibm.com')
PORT(1414) BACKLOG(0) DESCR('ESB02 Queue Manager Listener')
CONTROL(QMGR) REPLACE
```

Example 5-9 Create listener on the queue manager B2B02

```
DEFINE LISTENER ('B2B02.LISTENER')
TRPTYPE(TCP) IPADDR('b2b02.itso.ibm.com')
PORT(1414) BACKLOG(0) DESCR('B2B02 Queue Manager Listener')
CONTROL(QMGR) REPLACE
```

After all of the listeners are created, use the command shown in Example 5-10 to start the listeners from the **runmqsc** console.

Example 5-10 Start queue manager listener objects

```
START LISTENER(‘listener name’)
```

For more information about this command, go to the following website:

http://publib.boulder.ibm.com/infocenter/wmqv6/v6r0/topic/com.ibm.mq.csqzaj.doc/sc13170_.htm

Configure a Queue Manager cluster

To set up message communication between ITSO Healthcare queue manager, the following channels are required:

- Channel from ESB01 to ESB02 and B2B02
- Channel from ESB02 to ESB01 and B2B02
- Channel from B2B02 to ESB01 and ESB02

If all of the ITSO Healthcare systems are members of an MQ cluster, WebSphere MQ can manage the channels automatically. Using an MQ cluster results in simpler administration and configuration of the MQ network. The benefits of MQ clusters quickly increase as the number of systems grows. Using MQ clusters, the number of channels that must be manually defined is greatly reduced.

For scenarios with larger numbers, queue managers using MQ clusters offer even greater administrative savings. Thus, ITSO Healthcare decides to set up an MQ Cluster ITSO.CLUSTER. The queue managers participating in ITSO.CLUSTER are listed in Table 5-1.

Table 5-1 Participating queue manager in wmq cluster ITSO.CLUSTER

Queue Manager	Repository
ESB01	Full Repository
ESB02	Full Repository
B2B02	Partial Repository

Figure 5-31 illustrates the cluster channels created on these queue managers.

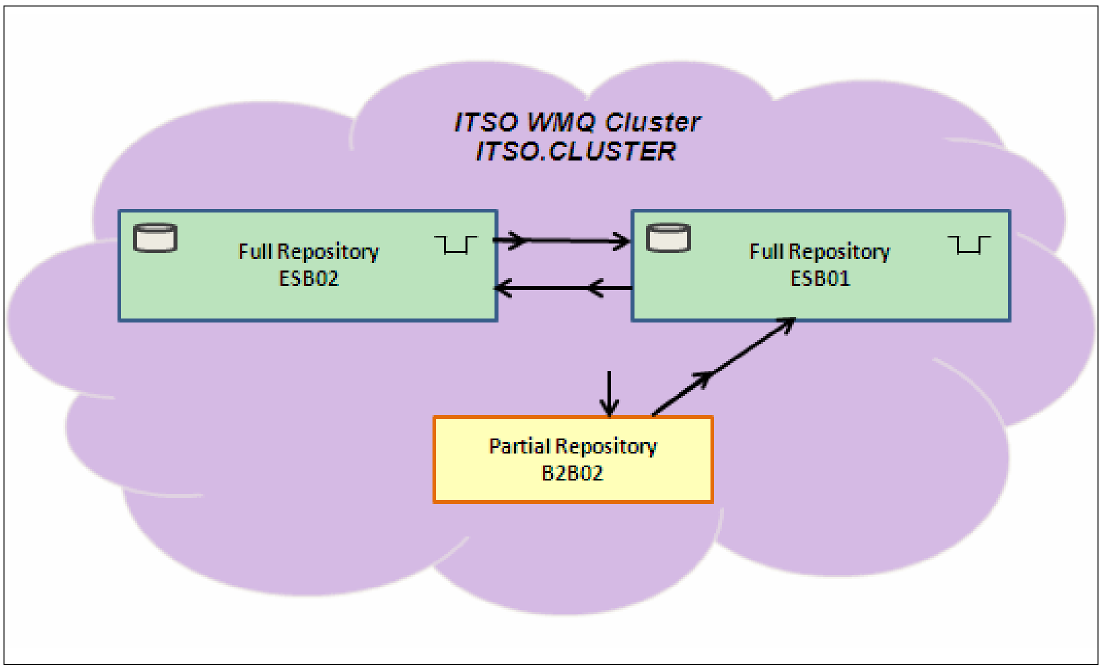


Figure 5-31 Cluster channels in MQ Cluster ITSO.CLUSTER

Alter queue manager definitions to add to the repository

On each queue manager that is to hold a full repository, you need to alter the queue manager definition using the ALTER QMGR command and specifying the REPOS attribute. Issue the command shown in Example 5-11 from the **runmqsc** console of the queue manager ESB01 and ESB02. These two queue managers are changed to a full repository.

Example 5-11 Change queue manager to full repository

```
ALTER QMGR REPOS('INVENTORY.CLUSTER')
```

Create a cluster channel on ESB01

Create a cluster receiver and cluster sender channel on the full repository queue manager ESB01. Example 5-12 shows the commands used to create these channels.

Example 5-12 Cluster channel on ESB01

```
DEFINE CHANNEL (TO.ESB01) CHLTYPE (CLUSRCVR) TRPTYPE (TCP) CONNAME
('esb01.itso.ibm.com(1414)') CLUSTER (ITSO.CLUSTER) REPLACE

DEFINE CHANNEL (TO.ESB02) CHLTYPE (CLUSSDR) TRPTYPE (TCP) CONNAME
('esb02.itso.ibm.com(1414)') CLUSTER (ITSO.CLUSTER) REPLACE
```

Create a cluster channel on ESB02

Create a cluster receiver and cluster sender channel on the full repository queue manager ESB02. Example 5-13 shows the commands used to create these channels.

Example 5-13 Cluster channel on ESB02

```
DEFINE CHANNEL (TO.ESB02) CHLTYPE (CLUSRCVR) TRPTYPE (TCP) CONNAME
('esb02.itso.ibm.com(1414)') CLUSTER (ITSO.CLUSTER) REPLACE

DEFINE CHANNEL (TO.ESB01) CHLTYPE (CLUSSDR) TRPTYPE (TCP) CONNAME
('esb01.itso.ibm.com(1414)') CLUSTER (ITSO.CLUSTER) REPLACE
```

Create a cluster channel on B2B02

Create a cluster receiver and cluster sender channel on the partial repository queue manager B2B02. Example 5-14 shows the commands used to create these channels.

Example 5-14 Cluster channel on B2B02

```
DEFINE CHANNEL (TO.B2B02) CHLTYPE (CLUSRCVR) TRPTYPE (TCP) CONNAME
('b2b02.itso.ibm.com(1414)') CLUSTER (ITSO.CLUSTER) REPLACE

DEFINE CHANNEL (TO.ESB01) CHLTYPE (CLUSSDR) TRPTYPE (TCP) CONNAME
('esb01.itso.ibm.com(1414)') CLUSTER (ITSO.CLUSTER) REPLACE
```

Queue managers are added to the repository and channels are created between the cluster components. Now we can also manage this cluster from WMQ explorer. Use the following procedure.

1. On the server esbo1.itso.ibm.com, select **Start** → **Programs** → **IBM WebSphere MQ** → **WebSphere MQ Explorer**, and click **WebSphere MQ Explorer** to launch the Eclipse GUI.
2. In the Navigator view of WebSphere MQ Explorer, expand **Queue Manager Cluster** → **ITSO.CLUSTER**.

3. Figure 5-32 shows the participating queue managers in MQ cluster **ITSO.CLUSTER.**

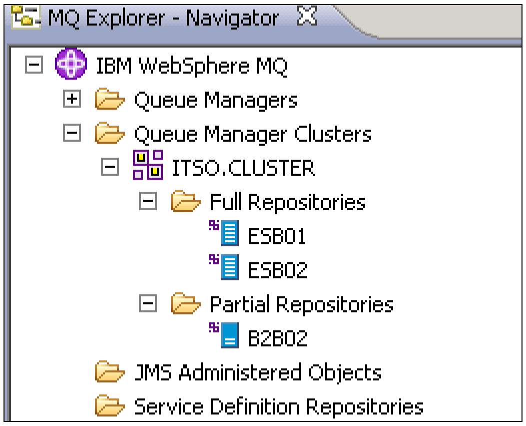


Figure 5-32 Cluster administration from WMQ explorer

For more information about WebSphere MQ clusters, go to the following website:

http://publib.boulder.ibm.com/infocenter/wmqv6/v6r0/topic/com.ibm.mq.csqzah.doc/qc10120_.htm

Create a server connection channel

MQI channels are used by applications in client mode (rather than bindings mode) to connect to queue managers. MQI channels are bidirectional; they carry WebSphere MQ API calls, (for example, "GET a message from queue XYZ") from the application to the queue manager. They also carry responses to those calls from the queue manager back to the application.

Sterling B2B Integrator uses server connection channel to connect B2B02 queue manager. Example 5-15 shows the command to be used from the **runmqsc** console to create the server connection channel **B2B.HEALTH.SFG** on queue manager B2B02.

Example 5-15 Create server connection channel

```
DEFINE CHANNEL('B2B.HEALTH.SFG') CHLTYPE(SVRCONN) REPLACE
```

Create queues for applications

You must first define the cluster queues on the queue manager, which will be accessible from any queue manager in the cluster. Example 5-16 shows the command to be executed from the **runmqsc** console of queue manager ESB01 and ESB02.

After executing this command, a cluster queue HEALTH.B2B.ESB.REQUEST.QC is created on ESB01 and ESB02. This queue is used for accepting incoming messages from HIPAA 837 message from Sterling B2B Integrator and later messages are processed by the backend ESB message flow.

Example 5-16 Cluster queue for accepting HIPAA 837 message

```
DEFINE QLOCAL (HEALTH.B2B.ESB.REQUEST.QC) DEFBIND (NOTFIXED) CLWLUSEQ (ANY)
CLUSTER (ITSO.CLUSTER) REPLACE
```

Example 5-17 shows the command to define local queue HEALTH.B2B.ESB.BACKEND.QL on queue manager ESB01 and ESB02 for sending messages to backend application in canonical format.

Example 5-17 Local queue for sending messages to backend application

```
DEFINE QLOCAL (HEALTH.B2B.ESB.BACKEND.QL) REPLACE
```

It is a good practice to define alias queues for clients, instead of providing details of physical cluster and local queues on the queue manager. We define alias queues to be used by Sterling B2B integrator, WebSphere Message Broker and the backend application.

Example 5-18 shows the command to define alias queue HEALTH.B2B.ESB.RQEQUEST. Sterling B2B integrator uses this queue to send HIPAA 837 messages. This alias queue, in turn, sends messages to target cluster queue HEALTH.B2B.ESB.REQUEST.QC on queue manager ESB01 and ESB02.

Example 5-18 Alias queues on queue manager B2B02

```
DEFINE QALIAS (HEALTH.B2B.ESB.REQUEST) CLUSTER (ITSO.CLUSTER) DEFBIND(NOTFIXED)
TARGQ (HEALTH.B2B.ESB.REQUEST.QC) REPLACE
```

Example 5-19 illustrates creation of alias queue HEALTH.B2B.ESB.BACKEND in queue manager ESB01, this queue is defined for putting messages from target local queue HEALTH.B2B.ESB.BACKEND.QL.

Example 5-19 Alias queues on queue manager ESB01

```
DEFINE QALIAS (HEALTH.B2B.ESB.BACKEND)TARGQ (HEALTH.B2B.ESB.BACKEND.QL) REPLACE
```

Example 5-20 illustrates creation of alias queue HEALTH.B2B.ESB.BACKEND in queue manager ESB02, this queue is defined for putting messages from target local queue HEALTH.B2B.ESB.BACKEND.QL.

Example 5-20 Alias queues on queue manager ESB02

```
DEFINE QALIAS (HEALTH.B2B.ESB.BACKEND)TARGQ (HEALTH.B2B.ESB.BACKEND.QL) REPLACE
```

To learn more about local queues, go to the following website:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/topic/com.ibm.mq.amqzag.doc/fa11220_.htm

To learn more about clustered queues and queue manager, go to the following website:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/topic/com.ibm.mq.csqzah.doc/qc10680_.htm

To learn more about alias queues, go to the following websites:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/topic/com.ibm.mq.csqzal.doc/fg10990_.htm

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/topic/com.ibm.mq.amqzag.doc/fa11330_.htm

With queues created to be used by Sterling B2B Integrator, WebSphere Message Broker, and backend application the WebSphere MQ, the configuration steps are complete.

Step 6: Configure WebSphere Transformation Extender

As stated in the introduction to this book, it is assumed that any WebSphere Transformation Extender component that is needed has already been installed. For any scenario using WebSphere Transformation Extender, the Design Studio must be installed on a development platform in order to create the maps needed for translation. In this scenario, WebSphere Transformation Extender for Integration Servers and the WebSphere Transformation Extender Industry Pack for Health Care have also been installed on the development platform, as well as the server that will host WebSphere Message Broker flows.

For this scenario, two translation maps are needed, one to transform the incoming HIPAA 837P message into canonical form, and a second to transform the canonical message into the target format. In addition to these transformations, the compliance check map system that ships with the health care industry pack is used to perform compliance checking and generate the HIPAA 999 acknowledgment using the Sterling B2B Integrator de-envelope service.

The compliance check map system that ships with the industry pack allows for compliance checking from level 1 compliance through level 7 compliance. For this scenario, we chose to use level 4 compliance checking. For details on how to set the compliance level in the compliance check map, and how to configure the Sterling B2B Integrator and WebSphere Transformation Extender to perform HIPAA compliance check, see the documentation that ships with the industry pack and the help menu in the Design Studio.

HIPAA 837P to canonical xml

The map used for translating the HIPAA 837P message to the canonical form is composed of one input card and two output cards, as shown in Figure 5-33 on page 146. This map is deployed to the Sterling B2B Integrator to make it available for use within the WTXMap service.

The input card is defined using a trimmed tree from the health care pack, as shown in Figure 5-34 on page 146 for the input card definition and Figure 5-35 on page 147 for the type tree. The type tree is customized based on the instructions in chapter 15 of the documentation shipped with the industry pack.

The first output card is defined using the canonical form `xsd`. Figure 5-36 on page 148 shows the output card definition. The second output card is defined using a type tree that consists of a blob item that has been nested inside layers of groups to mimic the envelope structure of the canonical xml. This was done to facilitate the process of splitting individual claims into separate xml interchanges. Figure 5-37 on page 149 shows the definition of output card 2.

In the case of a compliant message, the canonical form of the message, split into a single xml interchange per claim, is passed back to the business process. A reference to each singleton xml claim is put into ProcessData. The business process then passes each singleton xml claim to the broker flow for further processing. In the case of a non-compliant message processing terminates. In either case Sterling B2B Integrator returns the 999 acknowledgement to the trading partner.

> **Tip:** The WebSphere Transformation Extender map rule function “PUT” with the adapter alias “WIRE” is used to split the xml into singleton claims. Usage of this function is described in Chapter 4, “Routing and transforming messages” on page 69, as shown in Example 4-3 on page 98.

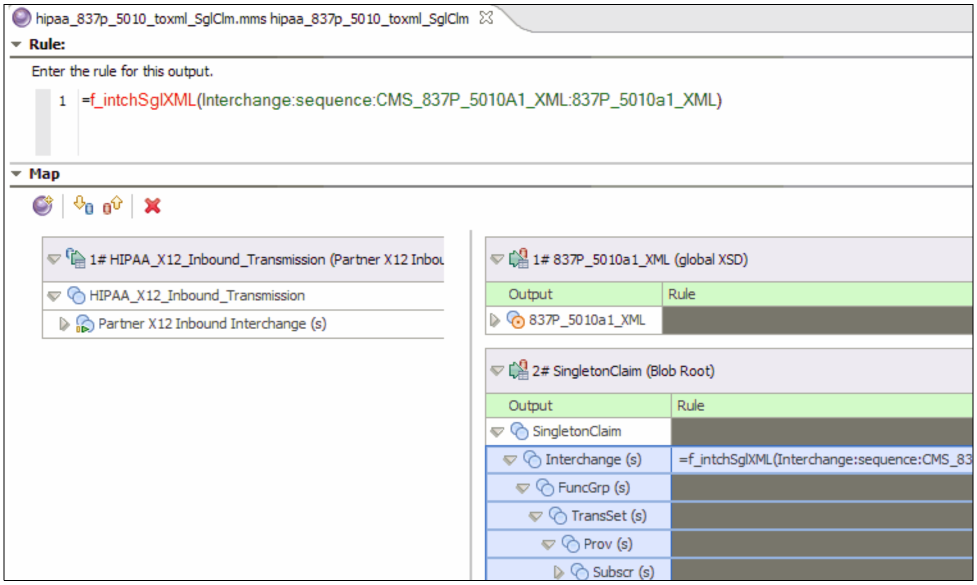
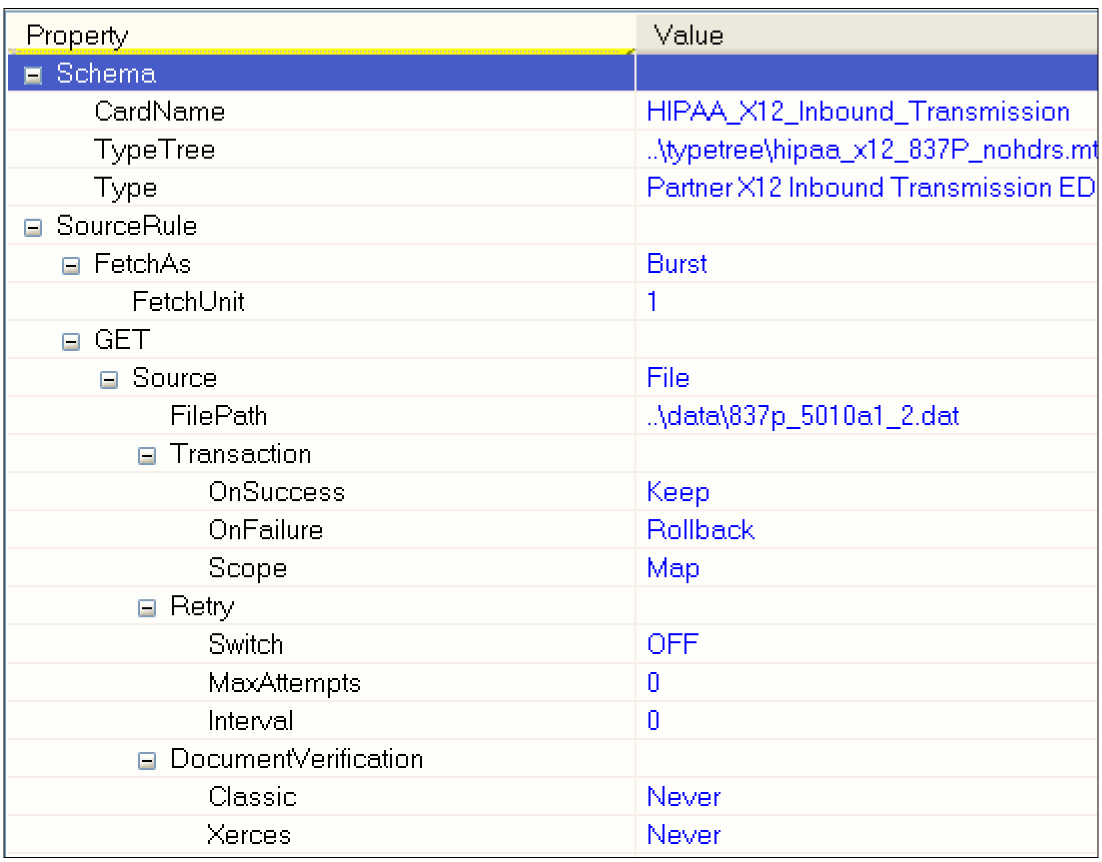


Figure 5-33 HIPAA_837p_5010_toxml_SglClm.mms



Property	Value
Schema	
CardName	HIPAA_X12_Inbound_Transmission
TypeTree	..\typetree\hipaa_x12_837P_nohdrs.mt
Type	Partner X12 Inbound Transmission ED
SourceRule	
FetchAs	Burst
FetchUnit	1
GET	
Source	File
FilePath	..\data\837p_5010a1_2.dat
Transaction	
OnSuccess	Keep
OnFailure	Rollback
Scope	Map
Retry	
Switch	OFF
MaxAttempts	0
Interval	0
DocumentVerification	
Classic	Never
Xerces	Never

Figure 5-34 Input card definition

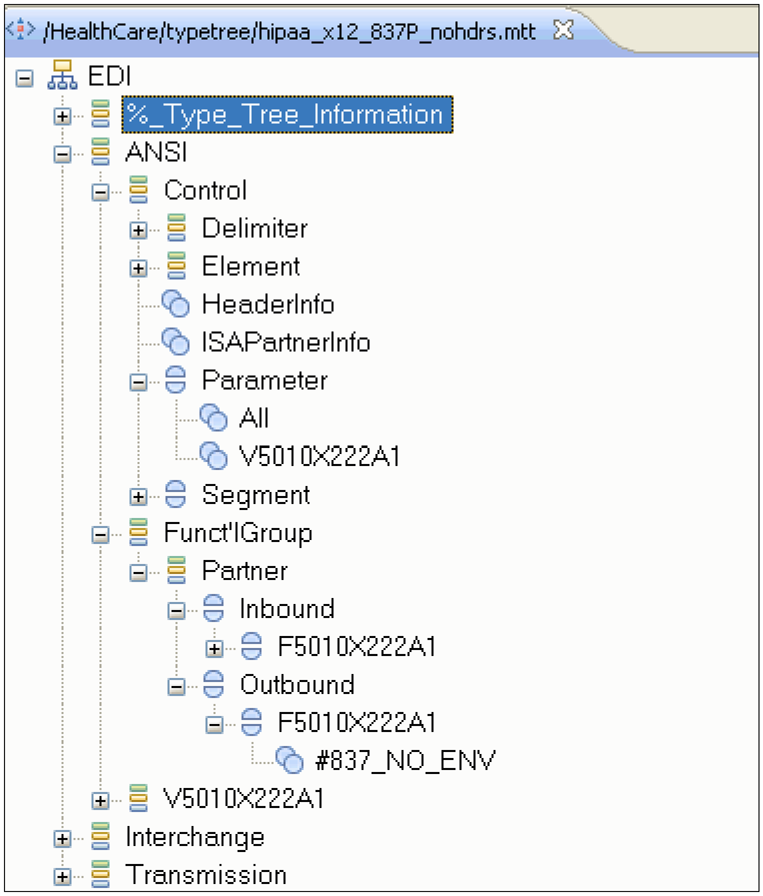


Figure 5-35 Trimmed HIPAA type tree

Property	Value
⊟ Schema	
CardName	canonical_xml
TypeTree	..\schema\HealthCanonical.xsd
⊟ Type	XSD
⊟ Metadata	..\schema\HealthCanonical.xsd
Name Spaces	
⊟ TargetRule	
⊟ PUT	
⊟ Target	Sink
⊟ DocumentVerification	
Classic	Never
Xerces	Never
⊟ Backup	
Switch	OFF
When	Always
⊟ BackupLocation	File
Directory	Map
FileName	Unique
SyntaxCard	No

Figure 5-36 Output Card 1

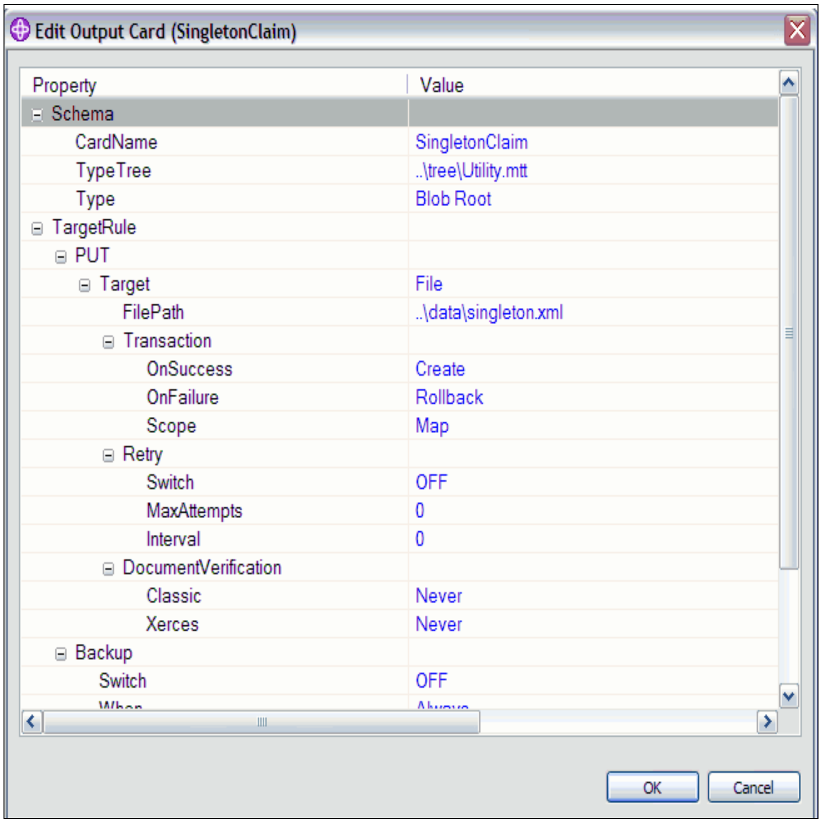


Figure 5-37 Output Card 2

Canonical to backend format

The map used for translating the canonical message to the backend format is composed of one input card and one output card, as shown in Figure 5-38 on page 150. Because the purpose of this map is to convert the canonical message created in the map described in the section "HIPAA 837P to canonical xml" on page 145, the input card for this map is defined exactly as the output card 1 of the HIPAA 837P to canonical xml map was defined, using the same xml schema definition file. Figure 5-39 on page 151 shows the card definition.

One of the things to notice is that the "Document Verification" attribute is set to "Well Formed (Xerces Only). It is because we do not need to validate the xml against the actual schema, because the xml was created internally. We only need to verify that the xml is well formed.

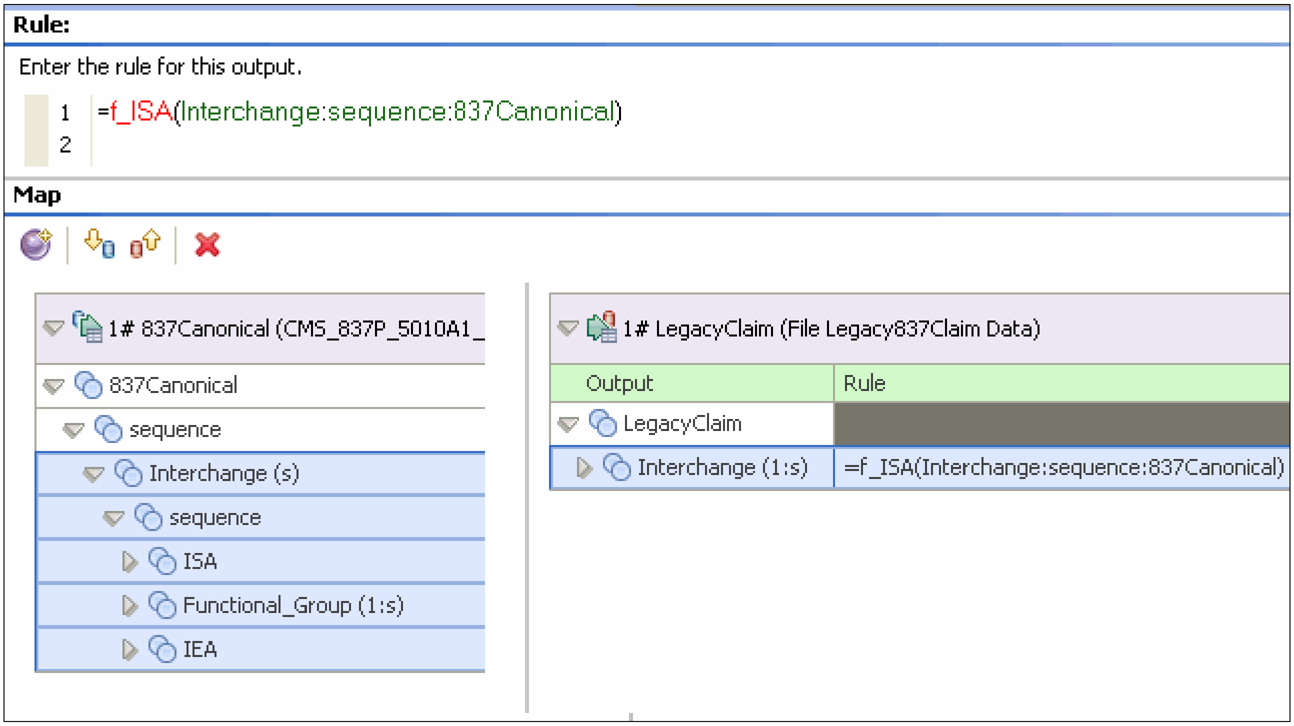


Figure 5-38 Canonical xml to backend format map

Property	Value
Schema	
CardName	canonical_xml
TypeTree	..\schema\HealthCanonical.xsd
Type	XSD
Metadata	..\schema\HealthCanonical.xsd
Name Spaces	
SourceRule	
FetchAs	Integral
WorkArea	Create
FetchUnit	S
GET	
Source	File
FilePath	..\data\837p_5010a1.xml
Transaction	
OnSuccess	Keep
OnFailure	Rollback
Scope	Map
Retry	
Switch	OFF
MaxAttempts	0
Interval	0
DocumentVerification	
Classic	Never
Xerces	Well Formed (Xerces only)
Backup	

Figure 5-39 Input card for canonical to backend map

The output card is defined by a custom type tree. In many cases, for similar scenarios, the backend format can be imported using one of the type importers described in Chapter 4, “Routing and transforming messages” on page 69 in Part 1 of this book, such as from a COBOL copybook. In our example, the format was given to us in a type specification document and the type tree was created by hand.

The output card definition is shown in Figure 5-40 on page 152 and the type tree we built is shown in Figure 5-41 on page 153. The result of executing this map is the canonical xml is translated into the delimited record format required by the backend application, as shown in Example 5-31 on page 175.

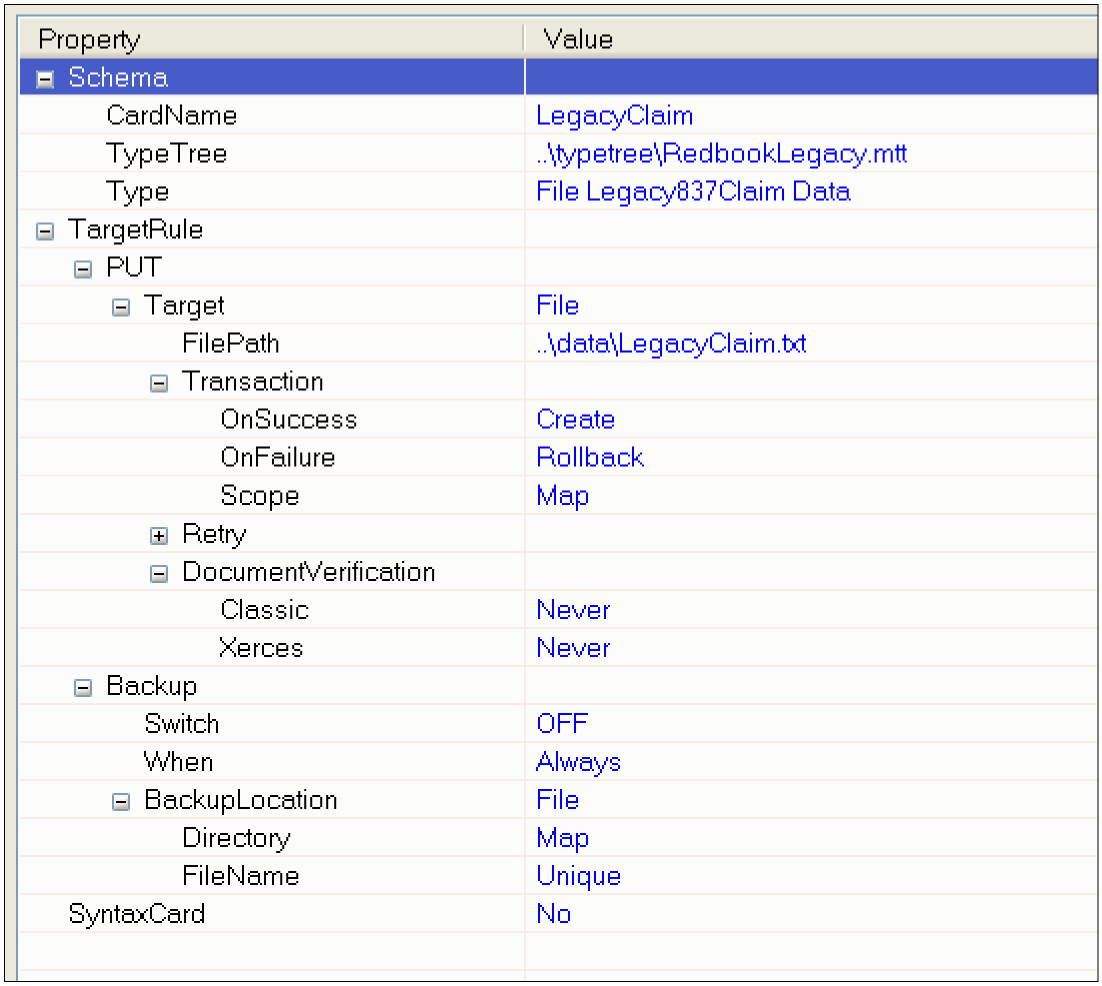


Figure 5-40 Output card definition

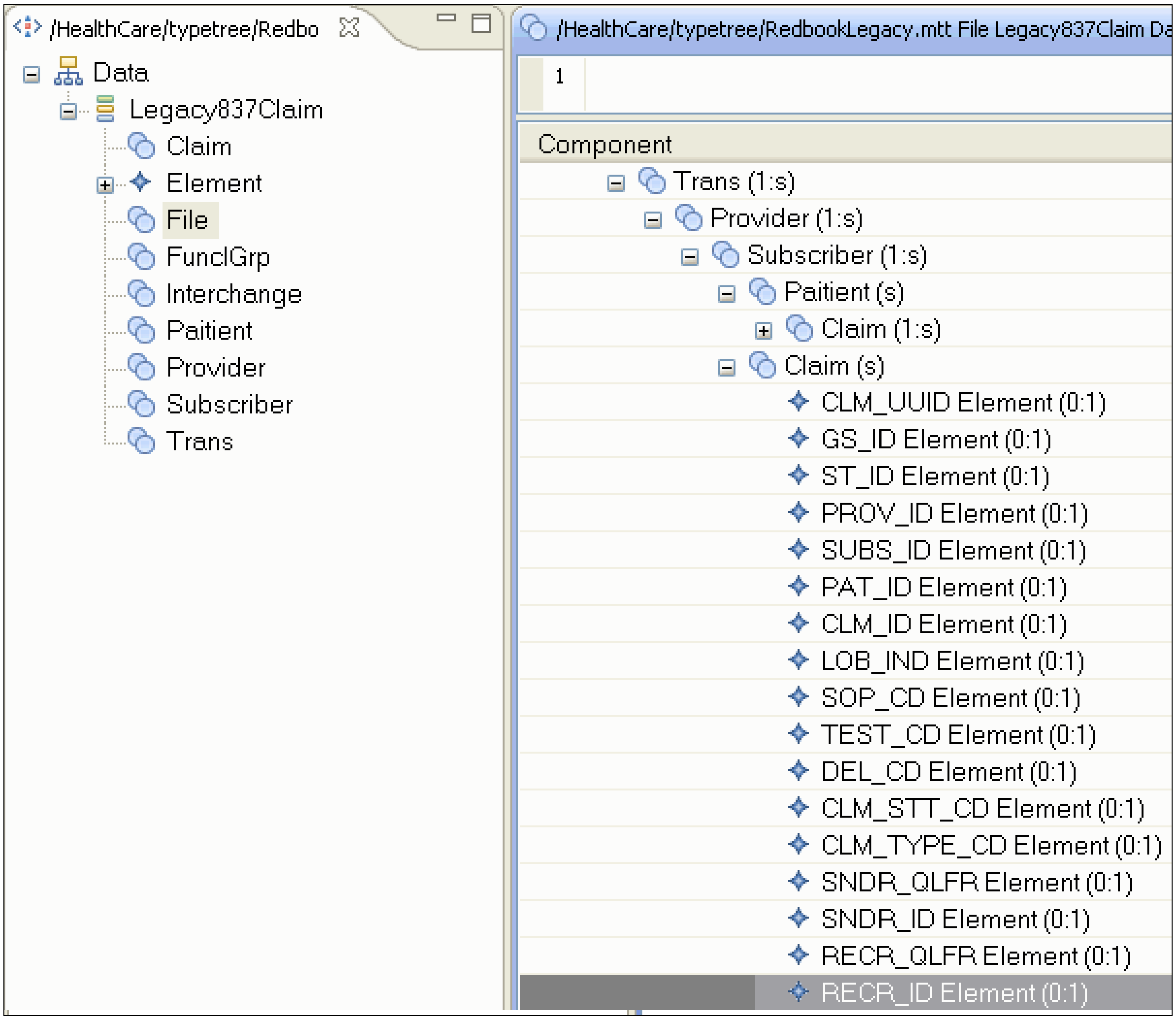


Figure 5-41 Type tree for backend format

Step 7: Configure WebSphere Message Broker

In this scenario, we use WebSphere Message Broker to transform and route incoming HIPAA messages arriving from Sterling B2B Integrator. Message flows deployed on WebSphere Message Broker read HIPAA 837 XML messages on the WebSphere MQ queue, and later transform and put incoming messages into legacy format for the backend application.

We use the following steps to configure WebSphere Message Broker for this scenario:

- Create a broker
- Create execution group on broker
- Create and configure message flow
- Deploy message flow

Create a Broker

A Message Broker can be created using the **mqsicreatebroker** command. Example 5-21 shows the syntax of this command.

Example 5-21 The mqsicreatebroker command syntax

```
mqsicreatebroker brokerName -i serviceUserId -a servicePassword -q
queueManagerName [-g configurationChangeTimeout] [-k internalConfigurationTimeout]
[-w workPath] [-e sharedWorkpath] [-l userLilPath] [-t] [-m] [-v
statisticsMajorInterval][-P httpListenerPort] [-c icuConverterPath] [-y
ldapPrincipal -z ldapCredentials] [-x userExitPath] [-o operationMode] [-s
adminSecurity] [-d MQService]
```

To learn more about all parameters for the mqsicreatebroker command, see the following website:

http://publib.boulder.ibm.com/infocenter/wmbhelp/v8r0m0/topic/com.ibm.etools.mft.doc/an07080_.htm

ESB01 broker for backend ESB-01

The command in Example 5-22 is executed on server `esb01.itso.ibm.com` to create broker ESB01.

Example 5-22 Create broker ESB01

```
mqsicreatebroker ESB01 -i <username> -a <password> -q ESB01
```

ESB02 broker for backend ESB-02

The command in Example 5-23 is executed on server `esb02.itso.ibm.com` to create broker ESB02.

Example 5-23 Create broker on ESB02

```
mqsicreatebroker ESB02 -i <username> -a <password> -q ESB02
```

Figure 5-42 shows the WebSphere Message Broker topology used for the implementation of the Healthcare scenario.

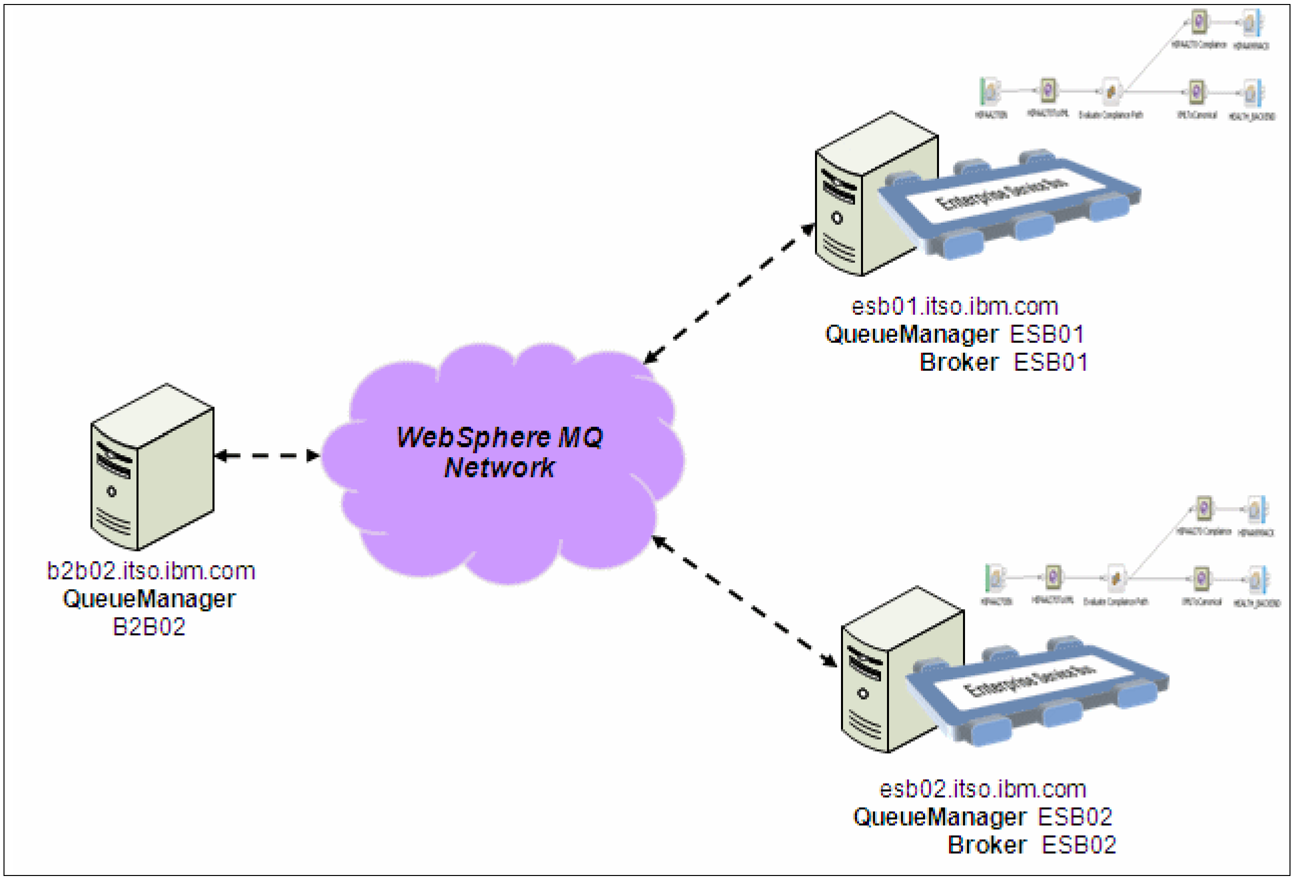


Figure 5-42 WebSphere Message Broker topology

Now that all of the brokers are created, we start them by using the **mqsistart** command, as shown in Example 5-24.

Example 5-24 The syntax for mqsistart command

```
mqsistart <brokerName>
```

To learn more about the **mqsistart** command, go to the following website:

http://publib.boulder.ibm.com/infocenter/wmbhelp/v8r0m0/topic/com.ibm.etools.mft.doc/an07230_.htm

The commands used to start various brokers are shown in Example 5-25.

Example 5-25 Start broker commands

```
>mqsistart ESB01
>mqsistart ESB02
```

Create an execution group on the broker

In this section, we create execution groups on various brokers by issuing the **mqsicreateexecutiongroup** command. An execution group is a named grouping of message flows that have been assigned to a broker. The broker enforces a degree of isolation between message flows in distinct execution groups by ensuring that they run in separate address spaces, or as unique processes.

An execution group process is also known as a DataFlowEngine (DFE); this term is typically used in problem determination scenarios (trace contents, diagnostic messages, and so on). A DFE is created as an operating system process, and has a one-to-one relationship with the named execution group. If more than one message flow runs within an execution group, multiple threads are created within the DFE process.

Example 5-26 shows the syntax for the **mqsicreateexecutiongroup** command.

Example 5-26 The syntax for mqsicreateexecutiongroup command

```
mqsicreateexecutiongroup brokerSpec -e egName [-w timeoutSecs] [-v traceFileName]
```

To learn more about the **mqsicreateexecutiongroup** command, go to the following website:

http://publib.boulder.ibm.com/infocenter/wmbhelp/v8r0m0/topic/com.ibm.etools.mft.doc/an26000_.htm

We create the execution group ITSOHealth on brokers ESB01 and ESB02. Example 5-27 and Example 5-28 shows the commands to be executed on the brokers.

Example 5-27 Create execution group ITSOHealth on ESB01

```
>mqsicreateexecutiongroup ESB01 -e ITSOHealth
```

Example 5-28 Create execution group ITSOHealth on ESB02

```
>mqsicreateexecutiongroup ESB02 -e ITSOHealth
```

Figure 5-43 shows the execution group ITSOHealth created on Broker ESB01.

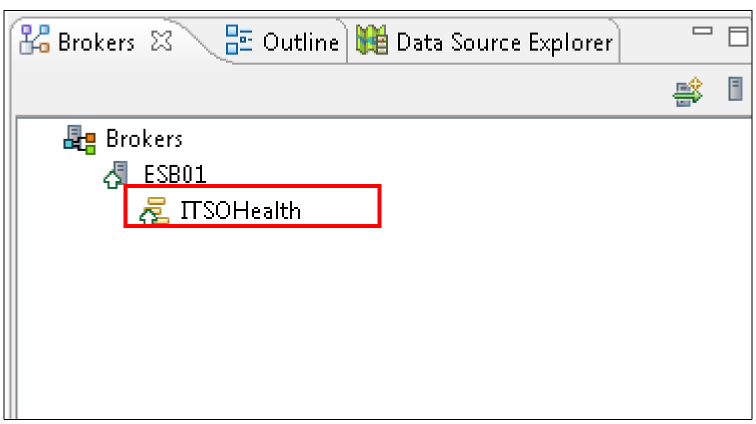


Figure 5-43 Execution group ITSOHealth on broker ESB01

Create and configure a message flow

We start by creating a new application, but before that, we launch the WebSphere Message Broker v8.0.0.0 toolkit, as described in the following procedure.

1. On the server esbo1.itso.ibm.com, select **Start** → **All Programs** → **IBM WebSphere Message Broker toolkit** → and click **WebSphere Message Broker toolkit 8.0** to launch the Eclipse GUI.
2. In the workspace wizard, enter the workspace location and click **OK.**
3. After the workspace is launched, close the **Welcome** tab and enter in “Broker application development perspective.”

4. Figure 5-44 illustrates the broker development view in the top left corner of the Broker application development perspective. Click **New Application** to open a new application wizard.

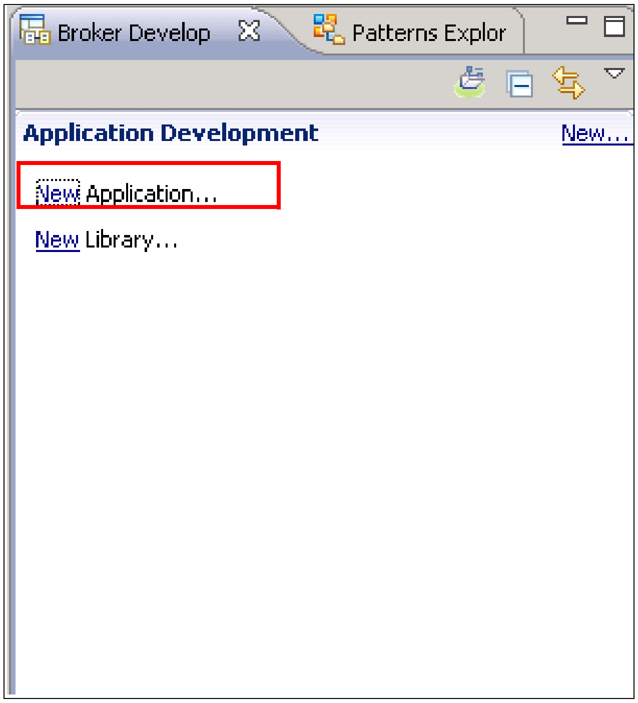


Figure 5-44 Create new message broker application project

5. Figure 5-45 shows the New Application wizard. Enter the application name as “HealthCareESB” and then click **Finish**.

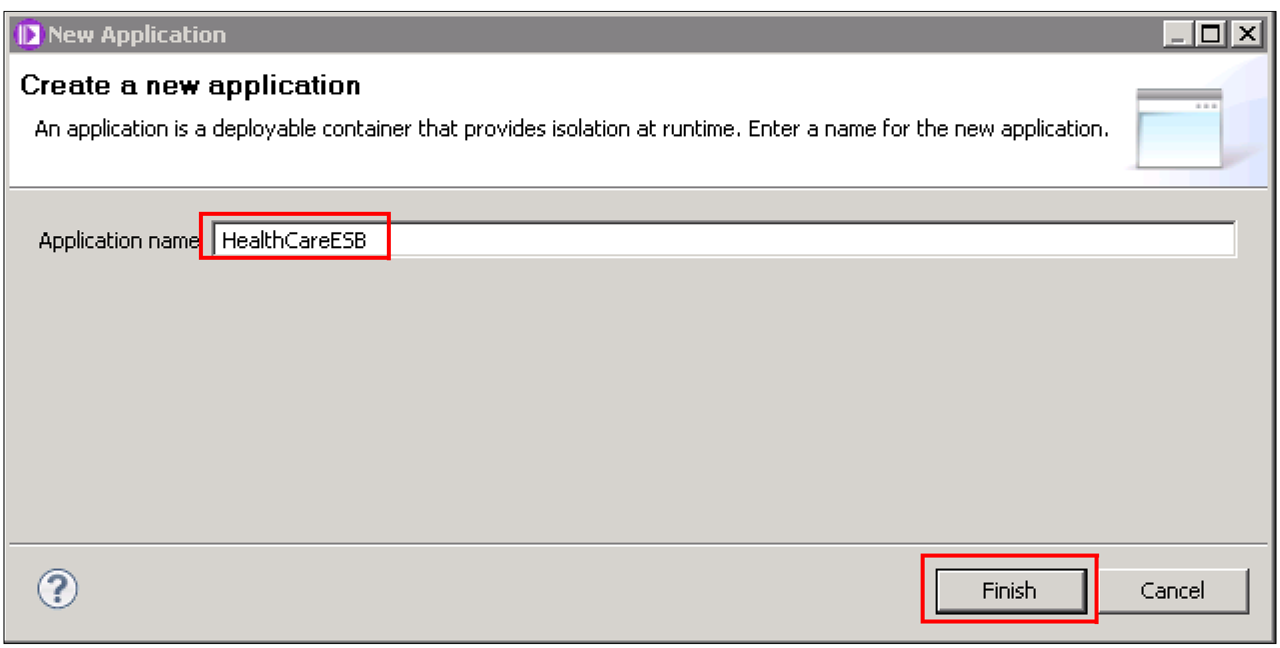


Figure 5-45 Create new application

6. Figure 5-46 shows the new message broker application project HealthCareESB created in the broker development view. Click **New** → **Message Flow**.

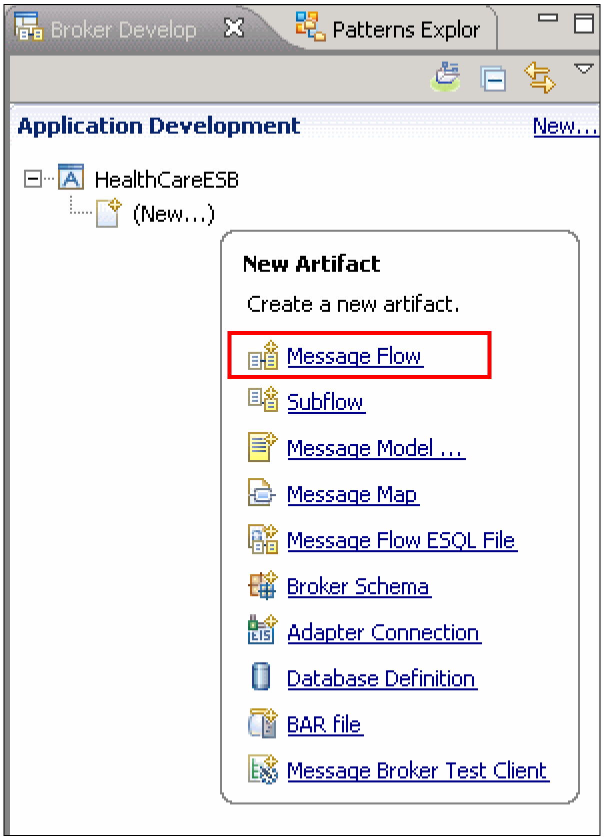


Figure 5-46 Create new message flow

7. Figure 5-47 shows the New Message Flow creation wizard. Enter the message flow name "HIPAA837" and then click **Finish**.

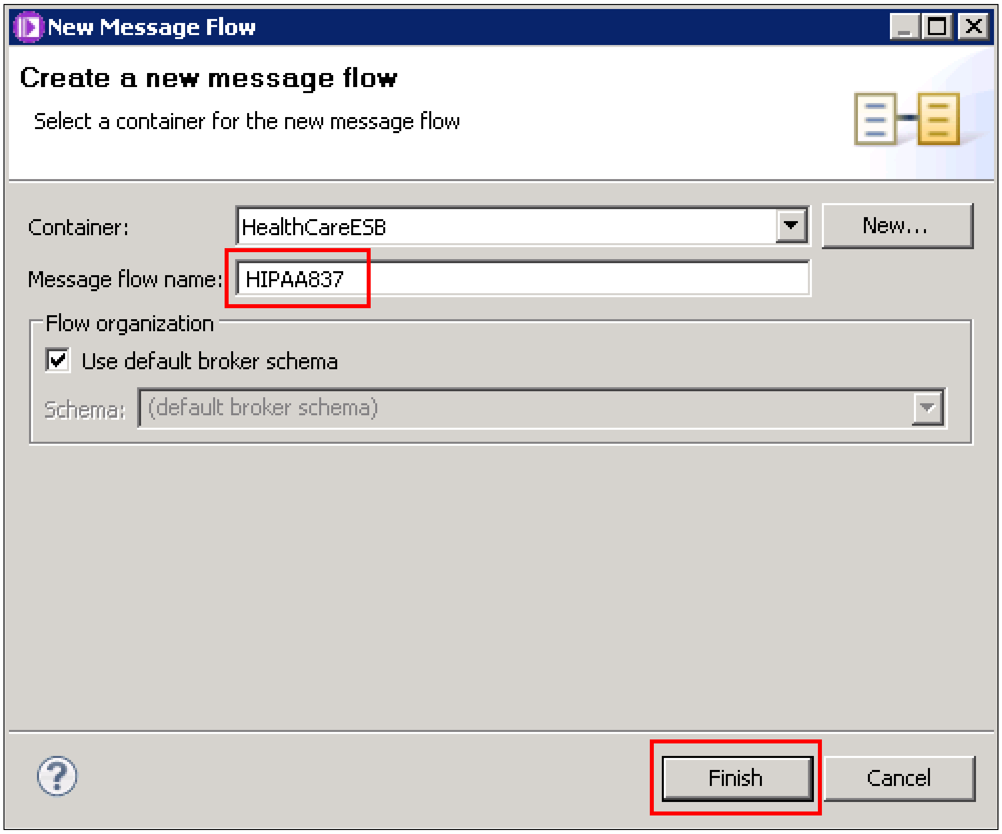


Figure 5-47 Create HIPAA837 message flow for SI and backend integration

8. Figure 5-48 shows the message flow editor for message flow HIPAA837. We now proceed by dragging and dropping the nodes in Table 5-2.

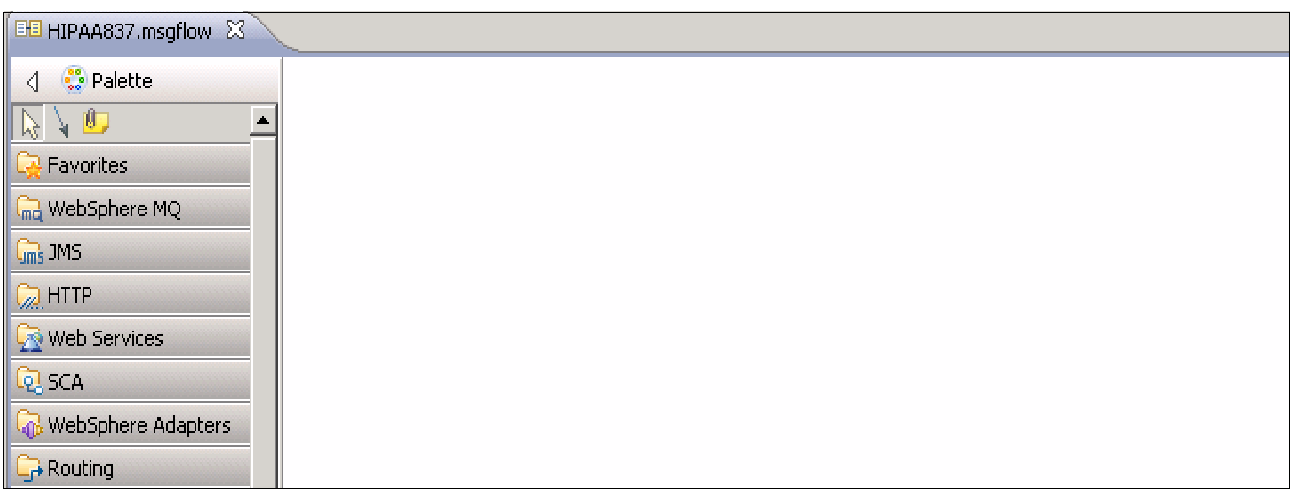


Figure 5-48 Create HIPAA837 message flow functionality

Table 5-2 lists the node types to be dragged from the pallet and then renamed to the corresponding node name in the message flow editor area.

Table 5-2 Nodes in message flow HIPAA837

Node name	Node type
HIPAA837IN	MQInput Node
XML To Legacy	WTX Node
HEALTH_BACKEND	MQOutput Node

Table 5-3 shows the terminals used to connect the nodes listed in Table 5-2.

Table 5-3 Connection between nodes

Source node	Output terminal	Destination node	Input terminal
HIPAA837IN	Out	XML To Legacy	in
XML To Legacy	out1	v	in

9. After the nodes are renamed and connected using the details mentioned in Table 5-2 and Table 5-3, the connected nodes message flow should look similar to Figure 5-49.

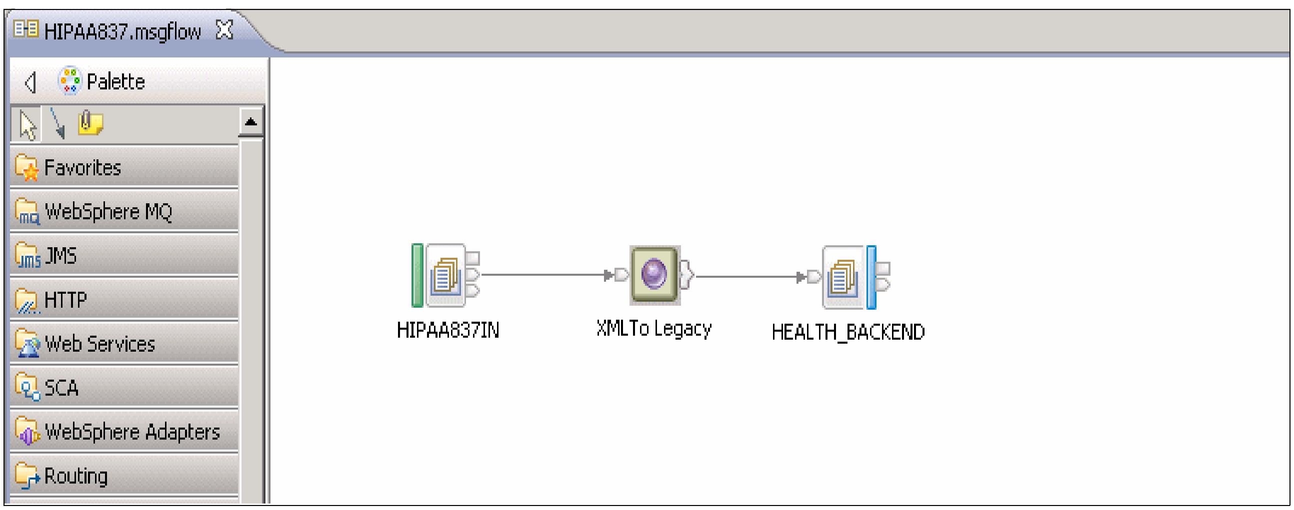


Figure 5-49 Connected nodes in message flow HIPAA837

10. Configure the properties of the nodes as shown in Table 5-4.

Table 5-4 Node properties

Node name	Property name	Property value
HIPAA837IN	Queue name	HEALTH.B2B.ESB.REQUEST
XML To Legacy	Local compiled map	<external path to map>/837Canonical_toLegacy.mmc
	Map server location	<external path to map>/837Canonical_toLegacy.mmc
	Input card	1
	Output card	1
HEALTH_BACKEND	Queue name	HEALTH.B2B.ESB.BACKEND

Save the flow, and proceed to create the broker archive (BAR) file for deployment.

Map properties: The message flow derives the WTX map properties from different sources, and does the search for the properties in the order listed:

1. **Run time:**
   - Source: LocalEnvironment
   - Override the map server location and input card to wire map properties at run time in the LocalEnvironment.
   - For information about how the LocalEnvironment works, see the related topics in the WebSphere Message Broker product documentation that is published on the WebSphere Message Broker Library Web page: `http://www.ibm.com/software/integration/wbimessagebroker/library`
2. **Deploy time:**
   - Source: broker archive (bar) file
   - Override the cache map, map name, map server location, and input card to wire map properties at deploy time in the bar file using the Configure tab of the Properties view, which is under the Manage tab in the Broker Archive editor.
3. **Design time:**
   - Source: WTX Map node
   - Override the map and card properties at design time in the Map Settings tab of the WTX Map node.

At run time, override the following map properties by using the WebSphere Message Broker LocalEnvironment tree from a prior node in the message flow, such as a Compute node, which can change the LocalEnvironment to override parameters in the tree that is passed to the WTX Map node:

dynamicMap property:

LocalEnvironment tree path: LocalEnvironment.WTX.DynamicMap

Purpose: You can override a WebSphere Transformation Extender map by populating the WebSphere Message Broker local environment tree with the binary data of a compiled map, stored, for example, in WebSphere Service Registry and Repository (WSRR).

Result: When a compiled map is stored in WSRR, you can use a node such as the WebSphere Message Broker Compute node, Java Compute node, or RegistryLookup node to populate the local environment tree with the binary data of the compiled map. A compiled map in the local environment tree overrides both a dynamic path to a map in the local environment tree

MapServerLocation property

LocalEnvironment tree path: LocalEnvironment.WTX.MapServerLocation

Purpose: To run the compiled map located in a different location than the location configured on the WTX Map node.

Result: WebSphere Message Broker overrides the map server location with the location you specified in the LocalEnvironment tree, and the WTX Map node runs the map from this location instead.

Note continued: (WTX map properties)

CardNumberToWire property

LocalEnvironment tree path: LocalEnvironment.WTX.InputCardNumberToWire

Purpose: To specify which input card of the map should be wired.

Result: The specified map input card receives its data from the prior node in the message flow.

Deploy a message flow

Prior to deployment, a broker archive (BAR) file should be created to compile the message flow. Use the following procedure.

1. Create a new BAR file as shown in Figure 5-50.

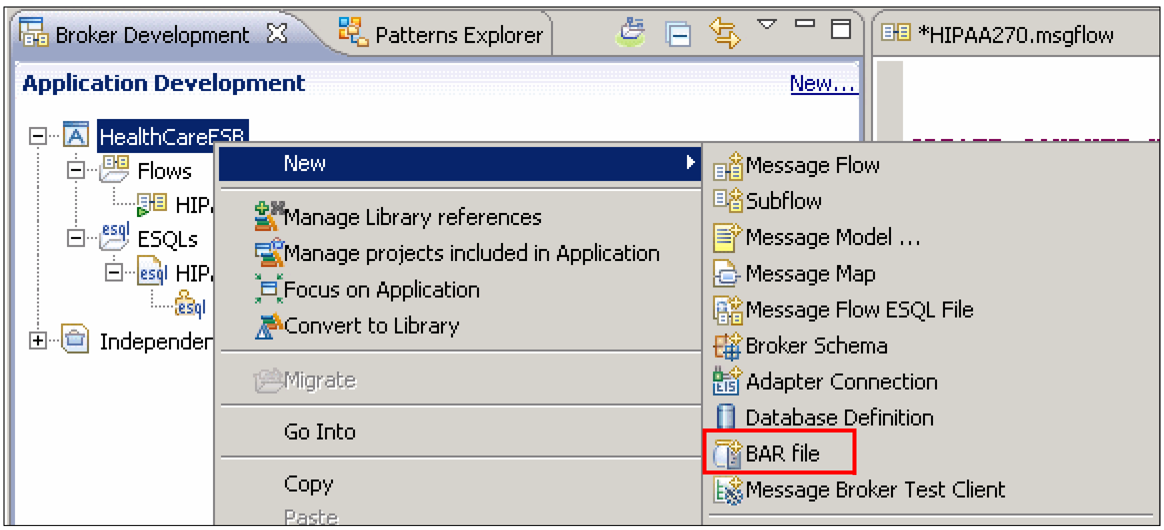


Figure 5-50 Create new BAR file

2. The New BAR file wizard opens. Enter the BAR file name as "HealthCareESB" and click **Finish.** Figure 5-51 shows the New BAR file wizard.

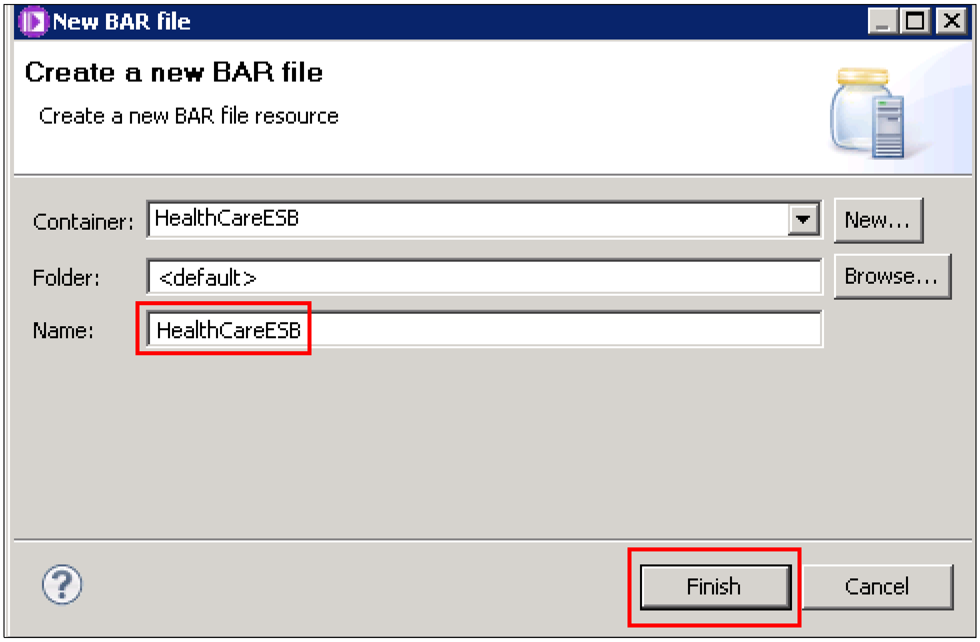


Figure 5-51 New BAR file wizard

3. After the BAR file is created, it opens in the BAR file editor, as shown in Figure 5-52. Click the **Prepare** tab in this editor and select the HealthCareESB application to be compiled in this BAR file.

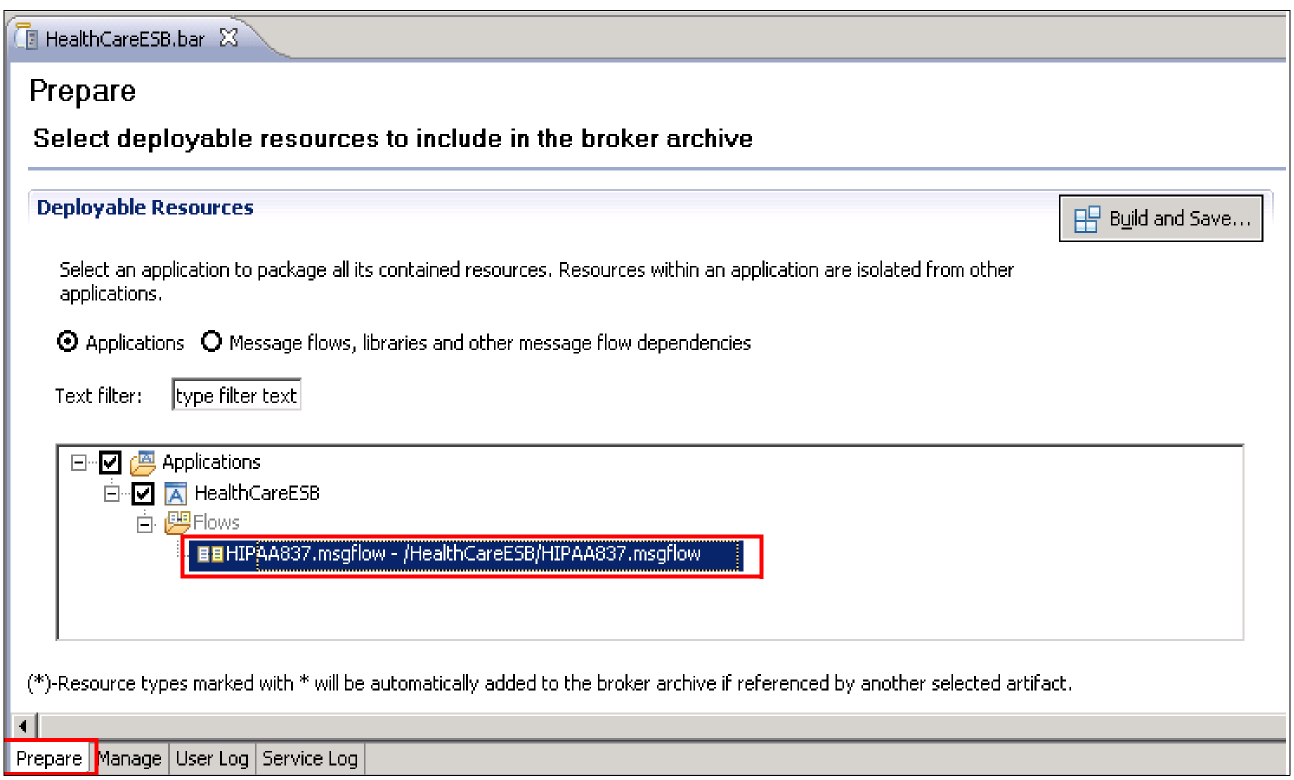


Figure 5-52 Select application in BAR file editor

4. Figure 5-53 shows how to build the BAR file. Click the **Manage** tab and then click the **Rebuild** button. If there are no compilation errors, the BAR file is built. Save the BAR file before deployment.

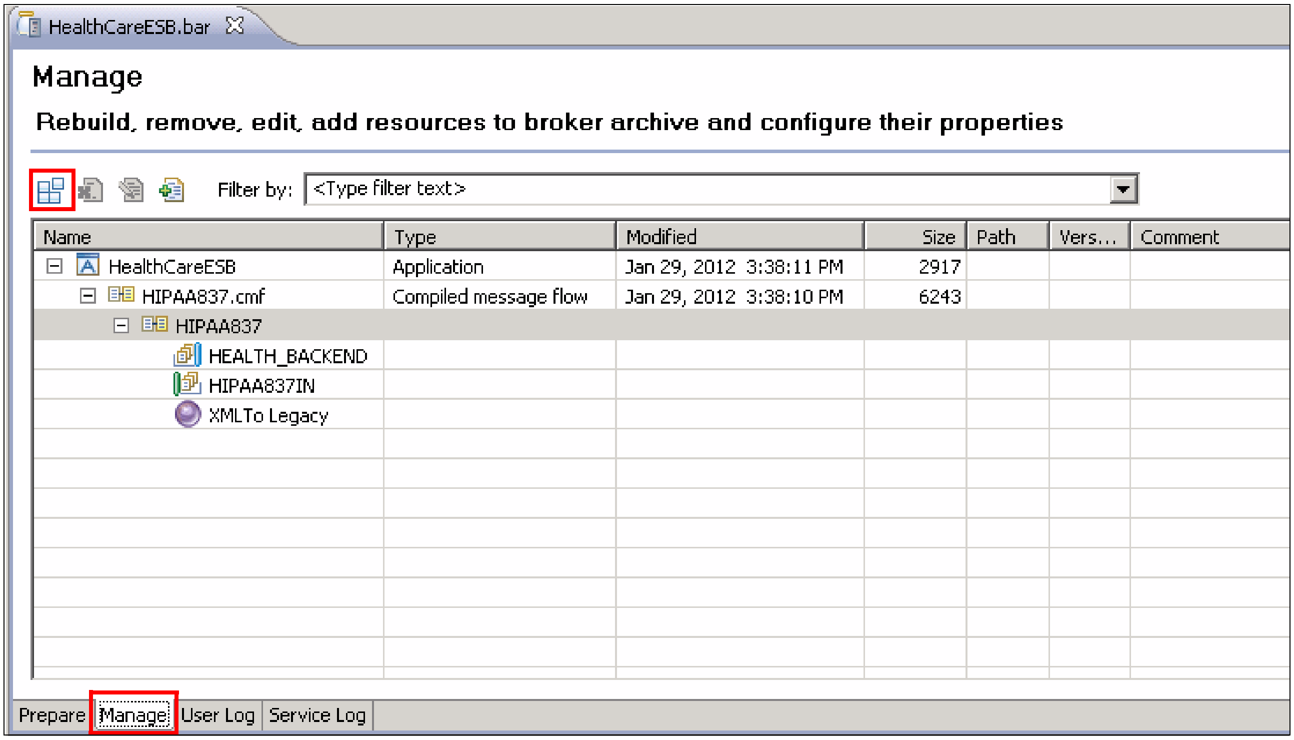


Figure 5-53 Build the BAR

5. After the BAR file is created, deploy it on broker ESB01 on `esb01.itso.ibm.com` and ESB02 on `esb02.itso.ibm.com`. Drag and drop the BAR file on execution group HealthCareESB on broker ESB01. Figure 5-54 shows the application deployed on the execution group. Transfer the BAR file to `esb02.itso.ibm.com` and deploy it on broker ESB02.

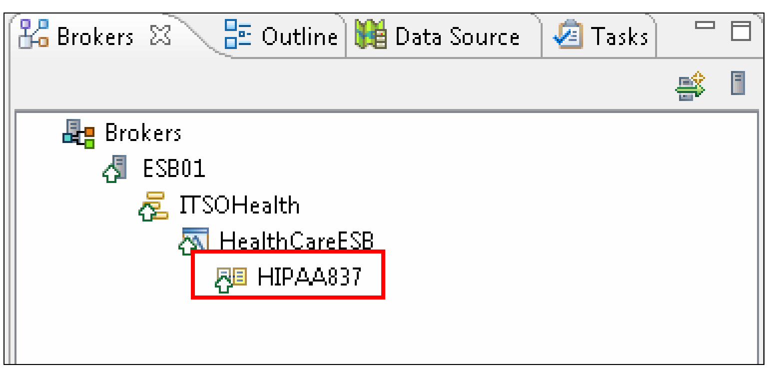


Figure 5-54 Message flow deployed on execution group ITSOHealth

The next main step is to perform the complete integration testing.

5.5 Testing the scenario

Now that the various applications are configured properly, it is time to test the integration of these systems. We need to track the message from the receipt at Sterling Secure Proxy through to the eventual delivery as the output from WebSphere Message Broker to the backend system.

In order to demonstrate this integration scenario, we track the delivery, receipt, and processing of a single Health Care Claim (HIPAA EDI 837) as it moves through the system. We view the original message, as received from the trading partner, and then we review the intermediate documents that are prepared by the various transformations. We end with the final document delivered to the backend application. We also briefly look at the acknowledgement (EDI 999) that is returned to the trading partner to confirm receipt of the message and detail any compliance errors.

5.5.1 Original message

The process is initiated by our fictitious trading partner, ITSORetail Corporation, creating a Health Care Claim (HIPAA EDI 837) request on behalf of a client. This request is formatted properly, and is then transmitted by the AS2 protocol by the public Internet, until it finally connects to our Sterling Secure Proxy engine in the DMZ.

Example 5-29 is the example message we are using in this test.

Example 5-29 Example HIPAA EDI 837 message

```
ISA*00*          *00*          *ZZ*HEALTHCARE_DEV *ZZ*HEALTHCARE_PS
*070612*1041*^*00501*000000001*0*T*:
GS*HC*DEVELOPMENT*PROFSERV*20070612*1041*42*X*005010X222A1
ST*837*1323*005010X222A1
BHT*0019*00*244579*20061015*1023*CH
NM1*41*2*PREMIER BILLING SERVICE*****46*TGJ23
PER*IC*JERRY JAMES*TE*3055552222*EX*231
NM1*40*2*KEY INSURANCE COMPANY*****46*66783JJT
HL*1**20*1
PRV*BI*PXC*203BF0100Y
NM1*85*2*BEN SANGFORD SERVICE*****XX*9876543210
N3*234 SEAWAY ST
N4*MIAMI*FL*33111
REF*EI*587654321
NM1*87*2
N3*2345 OCEAN BLVD
N4*MAIMI*FL*33111
HL*2*1*22*1
SBR*P**2222-SJ******CI
NM1*IL*1*SMITH*JANE****MI*JS00111223333
NM1*PR*2*KEY INSURANCE COMPANY*****XV*999996666
N4*MAIMI*FL*33111
REF*FY*KA6663
HL*3*2*23*0
PAT*19
NM1*QC*1*LARKINS*THEODORE
N3*236 N MAIN ST
N4*MIAMI*FL*33413
DMG*D8*19730501*M
```

```
CLM*26463774*100***11:B:1*Y*A*Y*I
REF*D9*17312345600006351
HI*BK:0340*BF:V7389
LX*1
SV1*HC:99213*40*UN*1***1
DTP*472*D8*20061003
LX*2
SV1*HC:87070*15*UN*1***1
DTP*472*D8*20061003
LX*3
SV1*HC:99214*35*UN*1***2
DTP*472*D8*20061010
LX*4
SV1*HC:86663*10*UN*1***2
DTP*472*D8*20061010
SE*42*1323
ST*837*1324*005010X222A1
BHT*0019*00*0123*19981015*1023*RP
NM1*41*1*PREMIER*BILLING SERVICE****46*TGJ23
PER*IC*BRENT RICE*TE*3055552222*EX*231
PER*IC*TERRY SPANGLER*TE*3055552222*EX*231
NM1*40*2*REPRICER SERVICES INC*****46*66783JJT
HL*1**20*1
PRV*BI*PXC*207Q00000X
CUR*85*CAD
NM1*85*1*PREMIER BILLING SERVICE
N3*234 SEAWAY ST
N4*TAMPA*FL*33111
REF*SY*888555222
NM1*87*2
N3*2345 OCEAN BLVD
N4*MIAMI*FL*33111
HL*2*1*22*0
SBR*P*18*12312A******HM
PAT*****D8*19970314
NM1*IL*1*OWENS*TED****MI*000221111
N3*236 N MAIN ST
N4*MIAMI*FL*33413
DMG*D8*19430501*M
NM1*PR*2*ALLIANCE HEALTH AND LIFE INSURANCE*****XV*741234
N4*KANSAS CITY*MO*64108
CLM*26462967*100***11:B:1*Y*A*Y*Y*P*EM
DTP*431*D8*19981003
DTP*439*D8*20070106
PWK*03*BM***AC*DMN0012
PWK*04*EL***AC*DMN0012
PWK*05*EM***AC*DMN0012
PWK*06*FX***AC*DMN0012
PWK*07*FT***AC*DMN0012
PWK*08*AA
PWK*09*AA
PWK*11*AA
PWK*13*AA
PWK*15*AA
REF*D9*17312345600006351
```

```
NTE*ADD*NOTE TEST
HI*BK:0340*BF:V7389
NM1*82*1*KILDARE*BEN****XX*112233334
PRV*PE*PXC*203BF0100Y
NM1*77*2*ABC CLINIC*****XX*581234567
N3*2345 OCEAN BLVD
N4*MIAMI*FL*33111
LX*1
SV1*HC:99213*40*UN*1***1
PWK*BT*BM***AC*DMN0012
PWK*CB*EL***AC*DMN0012
PWK*CK*EM***AC*DMN0012
PWK*CT*FX***AC*DMN0012
PWK*D2*FT***AC*DMN0012
PWK*DA*AA
PWK*DB*AA
PWK*DG*AA
PWK*DJ*AA
PWK*DS*AA
DTP*472*D8*19981003
LX*2
SV1*HC:99214*15*UN*1***1
PWK*03*BM***AC*DMN0012
PWK*04*EL***AC*DMN0012
PWK*05*EM***AC*DMN0012
PWK*06*FX***AC*DMN0012
PWK*07*FT***AC*DMN0012
PWK*08*AA
PWK*09*AA
PWK*11*AA
PWK*13*AA
PWK*15*AA
DTP*472*D8*19981003
LX*3
SV1*HC:87072*35*UN*1***2
PWK*21*BM***AC*DMN0012
PWK*A3*EL***AC*DMN0012
PWK*A4*EM***AC*DMN0012
PWK*AM*FX***AC*DMN0012
PWK*AS*FT***AC*DMN0012
PWK*B2*AA
PWK*B3*AA
PWK*B4*AA
PWK*BR*AA
PWK*BS*AA
DTP*472*D8*19981003
LX*4
SV1*HC:86663*10*UN*1***2
PWK*EB*BM***AC*DMN0012
PWK*HC*EL***AC*DMN0012
PWK*HR*EM***AC*DMN0012
PWK*I5*FX***AC*DMN0012
PWK*IR*FT***AC*DMN0012
PWK*LA*AA
PWK*M1*AA
```

```
PWK*MT*AA
PWK*NN*AA
PWK*OB*AA
DTP*472*D8*19981010
SE*99*1324
GE*2*42
IEA*1*000000001
```

5.5.2 Delivery to Sterling File Gateway and routing of the message

Sterling Secure Proxy confirms the identification of the sender, and if correct, then it routes the message to our Sterling B2B Integrator for processing by the standard AS2 business processes.

Sterling B2B Integrator receives an HTTP POST message on the URI defined for inbound AS communications, and this in turn initiates the EDIIntParse business process, which interrogates the message and retrieves the required values from the configured AS2 Partner profile by several sub-processes.

After Sterling B2B Integrator confirms the values and the message, it then deposits this message in the `/AS2/ITSOHealthcare/ITSORetail/Inbound` mailbox for Sterling File Gateway to route. It also sends a properly formatted MDN back to the partner, by Sterling Secure Proxy, to indicate successful receipt of the inbound message.

Because Sterling File Gateway has a Routing Rule that runs on all defined producer mailboxes continuously, the message placed in the `/AS2/ITSOHealthcare/ITSORetail/Inbound` mailbox is seen and routed to the `/ITSORetail/Inbox` almost immediately upon arrival.

Figure 5-55 on page 168 shows the routed message (highlighted) with some of the details of the route also displayed.

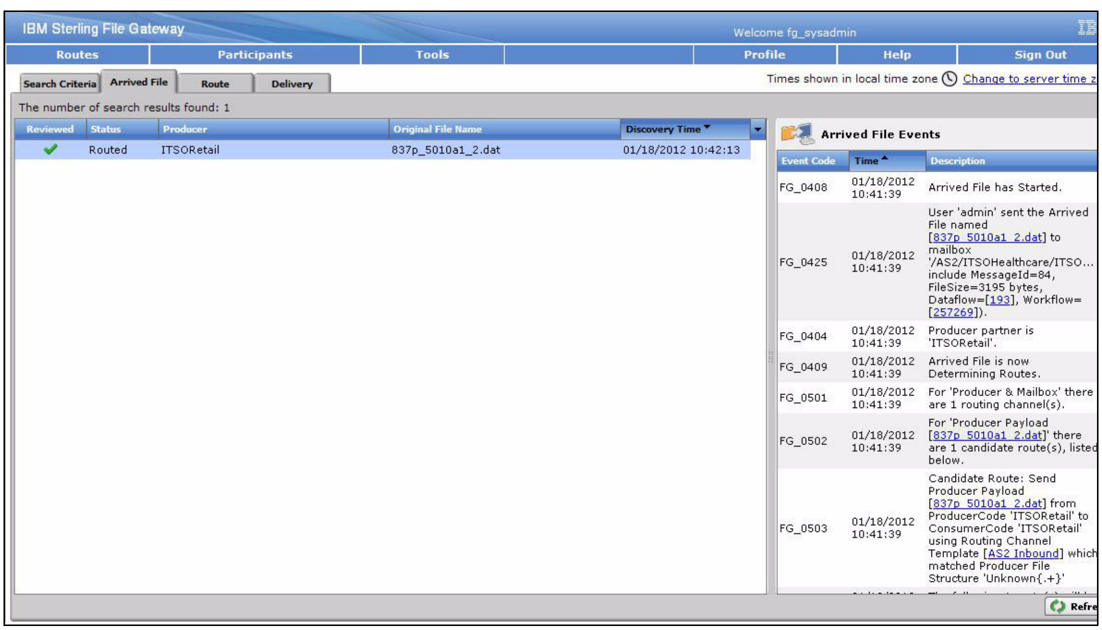


Figure 5-55 Successful route of the HIPAA EDI 837 message in Sterling File Gateway

As mentioned previously, Sterling File Gateway places the message in the mailbox `/ITSORetail/Inbox`, which prepares it for the next processing step, which will now proceed in Sterling B2B Integrator.

5.5.3 Processing of the message in Sterling B2B Integrator

Figure 5-56 shows the mailbox `/ITSORetail/Inbox` prior to the next step in the processing of the message.

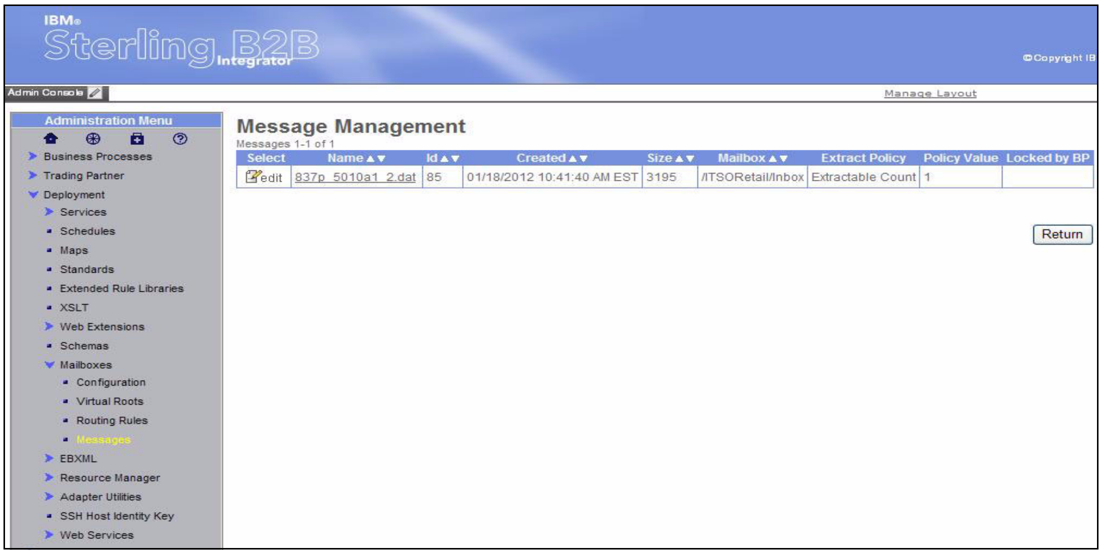
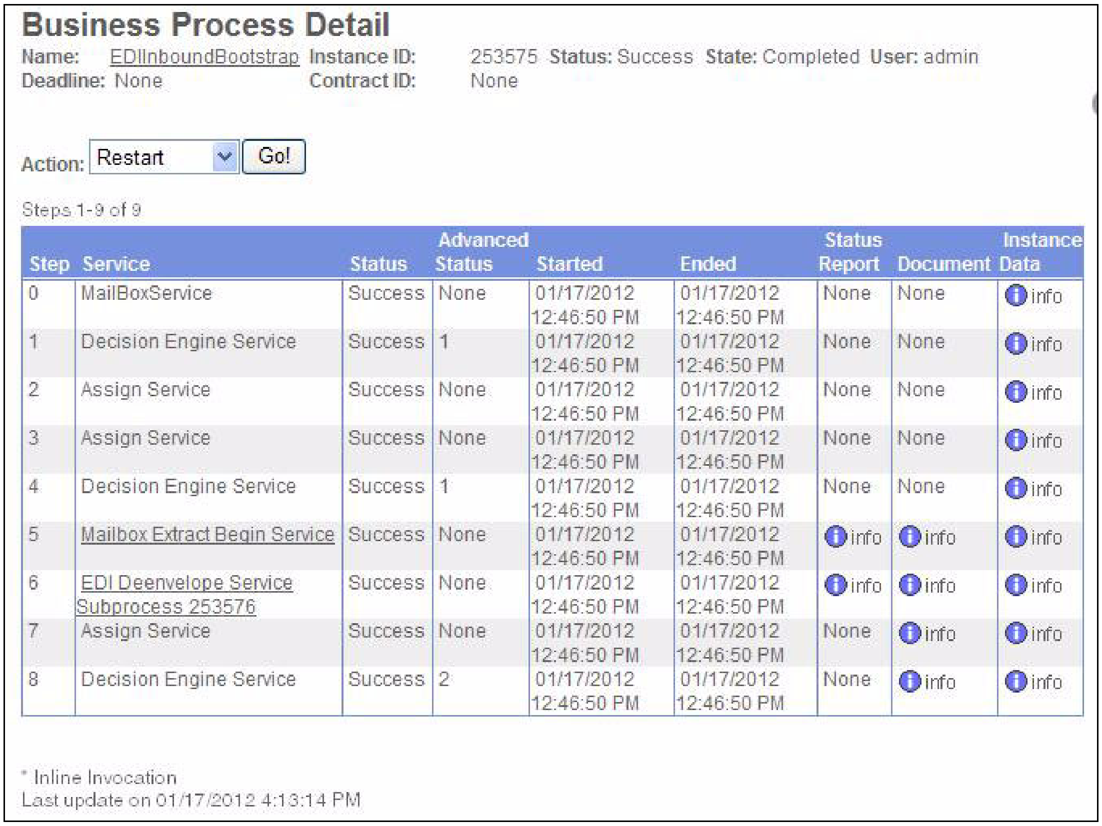


Figure 5-56 Mail /ITSORetail/Inbox prior to the Routing Rule running

Because of the routing rule configured to watch this mailbox, the system detects the message after it is placed in the mailbox and then trigger the business process EDIInboundBootstrap.

This business process extracts the message from the `/ITSORetail/Inbox` mailbox and then, as part of the process, invoke the EDIDeenvelope business process, which is another process that is shipped as a standard part of Sterling B2B Integrator.

Figure 5-57 shows the steps of a successful execution of the business process.



Business Process Detail

Name: EDIInboundBootstrap Instance ID: 253575 Status: Success State: Completed User: admin
Deadline: None Contract ID: None

Action: Restart Go!

Steps 1-9 of 9

Step	Service	Status	Advanced Status	Started	Ended	Status Report	Document	Instance Data
0	MailBoxService	Success	None	01/17/2012 12:46:50 PM	01/17/2012 12:46:50 PM	None	None	info
1	Decision Engine Service	Success	1	01/17/2012 12:46:50 PM	01/17/2012 12:46:50 PM	None	None	info
2	Assign Service	Success	None	01/17/2012 12:46:50 PM	01/17/2012 12:46:50 PM	None	None	info
3	Assign Service	Success	None	01/17/2012 12:46:50 PM	01/17/2012 12:46:50 PM	None	None	info
4	Decision Engine Service	Success	1	01/17/2012 12:46:50 PM	01/17/2012 12:46:50 PM	None	None	info
5	Mailbox Extract Begin Service	Success	None	01/17/2012 12:46:50 PM	01/17/2012 12:46:50 PM	info	info	info
6	EDI Deenvelope Service Subprocess 253576	Success	None	01/17/2012 12:46:50 PM	01/17/2012 12:46:50 PM	info	info	info
7	Assign Service	Success	None	01/17/2012 12:46:50 PM	01/17/2012 12:46:50 PM	None	info	info
8	Decision Engine Service	Success	2	01/17/2012 12:46:50 PM	01/17/2012 12:46:50 PM	None	info	info

* Inline Invocation
Last update on 01/17/2012 4:13:14 PM

Figure 5-57 Successful execution of EDIInboundBootstrap

This service then invokes the X12DeenvelopeUnified business process, which does the actual de-enveloping of the message, and the compliance check and initial transformation to the canonical format.

Figure 5-58 is the successful execution of the X12DeenvelopeUnified business process, which in turn invokes the next process in the chain.

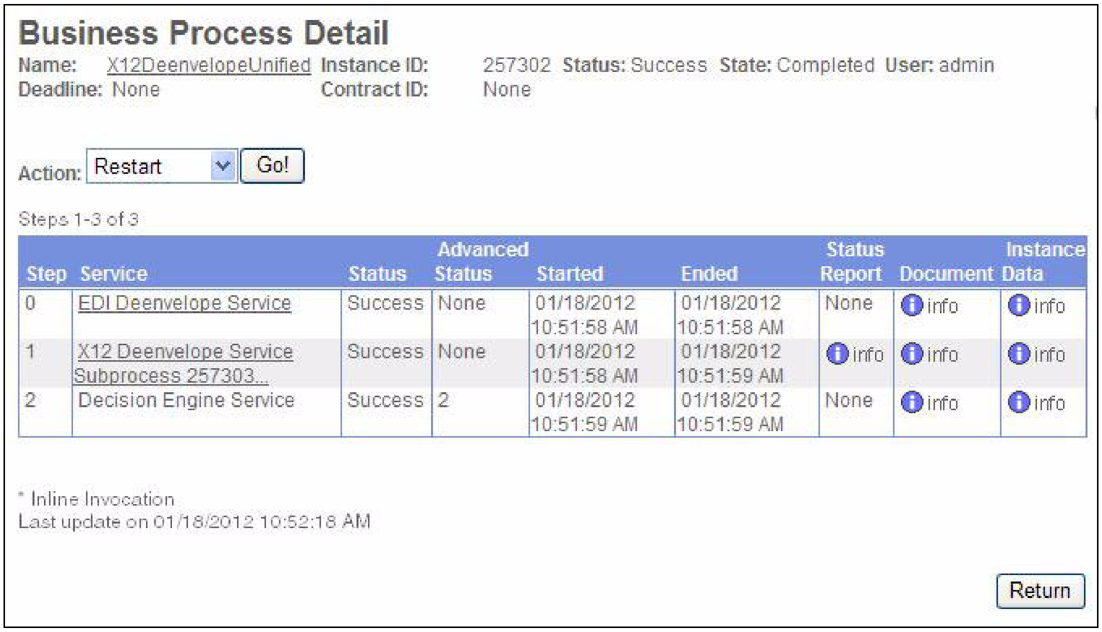


Figure 5-58 Successful execution of X12DeenvelopeUnified

The X12DeenvelopeUnified business process completes the de-enveloping of the message, generates the acknowledgement, and finally triggers the last business process in the chain, which will finally send the message on to the WebSphere MQ queue for further processing.

Figure 5-59 shows successful execution of the business process ITSOHealthCare_MQSend, which completes this portion of the test.

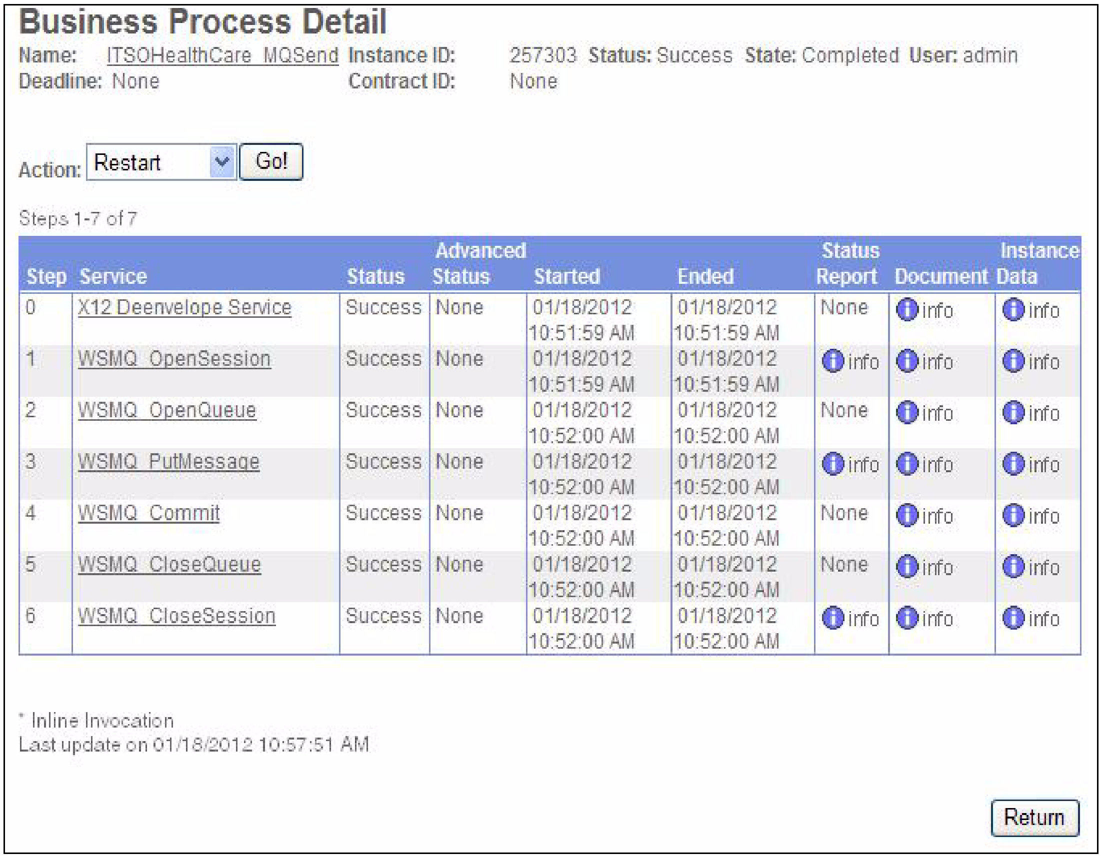


Figure 5-59 Successful execution to move the message to the WebSphere MQ queue

5.5.4 Processing of the message in WebSphere Message Broker

After the broker picks the inbound 837 message up from the queue, it then routes the message to a WTX map node and converts the XML message in backend legacy format. Example 5-30 illustrates the incoming XML message which will be transformed in legacy format, as shown in Example 5-31 on page 175.

Example 5-30 Excerpt of the canonical xml after transformation from HIPAA 837

```
<CMS_837P_5010A1_XML>
<Interchange>
<ISA>
<Record_Identifier_84>
<Loop_Id>
</Loop_Id>
<Loop_Seq_Num>
</Loop_Seq_Num>
<Segment_Id>ISA</Segment_Id>
<Segment_Seq_Num>1</Segment_Seq_Num>
</Record_Identifier_84>
<AUTH_INFO_QUAL>00</AUTH_INFO_QUAL>
<AUTH_INFO>
</AUTH_INFO>
```

```xml
<SEC_INFO_QUAL>00</SEC_INFO_QUAL>
<SEC_INFO>
</SEC_INFO>
<INTRCHG_ID_QUAL>ZZ</INTRCHG_ID_QUAL>
<INTRCHG_SENDR_ID>HEALTHCARE_DEV</INTRCHG_SENDR_ID>
<INTRCHG_ID_QUAL2>ZZ</INTRCHG_ID_QUAL2>
<INTRCHG_RECVR_ID>HEALTHCARE_PS</INTRCHG_RECVR_ID>
<INTRCHG_DATE>2007-06-12T00:00:00</INTRCHG_DATE>
<INTRCHG_TIME>10:41:00</INTRCHG_TIME>
<REPITION_SEPARATOR>^</REPITION_SEPARATOR>
<INTRCHG_CTRL_VERS_NO>00501</INTRCHG_CTRL_VERS_NO>
<INTRCHG_CTRL_NO_286>1</INTRCHG_CTRL_NO_286>
<ACK_REQ>0</ACK_REQ>
<USAGE_IND>T</USAGE_IND>
<COMP_ELE_SEP>:</COMP_ELE_SEP>
</ISA>
<Functional_Group>
<GS>
<Record_Identifier_84>
<Loop_Id>
</Loop_Id>
<Loop_Seq_Num>
</Loop_Seq_Num>
<Segment_Id>GS</Segment_Id>
<Segment_Seq_Num>1</Segment_Seq_Num>
</Record_Identifier_84>
<FUNCT_ID_CD>HC</FUNCT_ID_CD>
<APP_SENDR_CD>DEVELOPMENT</APP_SENDR_CD>
<APP_RECVR_CD>PROFSERV</APP_RECVR_CD>
<FUNCTIONAL_GROUP_DATE>2007-06-12T00:00:00</FUNCTIONAL_GROUP_DATE>
<FUNCTIONAL_GROUP_TIME>10:41:00</FUNCTIONAL_GROUP_TIME>
<GRP_CTRL_NO_272>42</GRP_CTRL_NO_272>
<RESP_AGENCY_CD>X</RESP_AGENCY_CD>
<VERS_ID_CD>005010X222A1</VERS_ID_CD>
</GS>
<DATE_TRANSMISSION_RECEIPT>
<Record_Identifier_84>
<Loop_Id>
</Loop_Id>
<Loop_Seq_Num>
</Loop_Seq_Num>
<Segment_Id>DTP</Segment_Id>
<Segment_Seq_Num>1</Segment_Seq_Num>
</Record_Identifier_84>
<DATE_TIME_QUAL_202>+RC</DATE_TIME_QUAL_202>
<DATE_TIME_PERIOD_FORMAT_QUAL_201>D8</DATE_TIME_PERIOD_FORMAT_QUAL_201>
<TRANSMISSION_RECEIPT_DATE>2007-06-12T00:00:00</TRANSMISSION_RECEIPT_DATE>
</DATE_TRANSMISSION_RECEIPT>
<CLAIM_SOURCE_INDICATOR>
<Record_Identifier_84>
<Loop_Id>
</Loop_Id>
<Loop_Seq_Num>
</Loop_Seq_Num>
<Segment_Id>REF</Segment_Id>
```

```
<Segment_Seq_Num>1</Segment_Seq_Num>
</Record_Identifier_84>
<REF_ID_QUAL_354>+PR</REF_ID_QUAL_354>
<CLAIM_SOURCE_INDICATOR>O</CLAIM_SOURCE_INDICATOR>
</CLAIM_SOURCE_INDICATOR>
<Transaction_Set>
<ST>
<Record_Identifier_84>
<Loop_Id>
</Loop_Id>
<Loop_Seq_Num>
</Loop_Seq_Num>
<Segment_Id>ST</Segment_Id>
<Segment_Seq_Num>1</Segment_Seq_Num>
</Record_Identifier_84>
<TRANS_SET_ID_CD>837</TRANS_SET_ID_CD>
<TRANS_SET_CTRL_NO_385>1323</TRANS_SET_CTRL_NO_385>
<IMPL_GUIDE_VERSION_NAME>005010X222A1</IMPL_GUIDE_VERSION_NAME>
</ST>
.
.
.
</Service_Line>
</Claim_424>
</Subscriber_HL>
</Provider_HL>
<SE>
<Record_Identifier_84>
<Loop_Id>
</Loop_Id>
<Loop_Seq_Num>
</Loop_Seq_Num>
<Segment_Id>SE</Segment_Id>
<Segment_Seq_Num>1</Segment_Seq_Num>
</Record_Identifier_84>
<TRANS_SEGMENT_COUNT>99</TRANS_SEGMENT_COUNT>
<TRANS_SET_CTRL_NO_385>1324</TRANS_SET_CTRL_NO_385>
</SE>
</Transaction_Set>
<GE>
<Record_Identifier_84>
<Loop_Id>
</Loop_Id>
<Loop_Seq_Num>
</Loop_Seq_Num>
<Segment_Id>GE</Segment_Id>
<Segment_Seq_Num>1</Segment_Seq_Num>
</Record_Identifier_84>
<NO_OF_TRANS_SETS_INCLUDED>2</NO_OF_TRANS_SETS_INCLUDED>
<GRP_CTRL_NO_272>42</GRP_CTRL_NO_272>
</GE>
</Functional_Group>
<IEA>
<Record_Identifier_84>
<Loop_Id>
```

```
</Loop_Id>
<Loop_Seq_Num>
</Loop_Seq_Num>
<Segment_Id>IEA</Segment_Id>
<Segment_Seq_Num>1</Segment_Seq_Num>
</Record_Identifier_84>
<NO_OF_INCLUDED_FUNCT_GRPS>1</NO_OF_INCLUDED_FUNCT_GRPS>
<INTRCHG_CTRL_NO_286>1</INTRCHG_CTRL_NO_286>
</IEA>
</Interchange>
</CMS_837P_5010A1_XML>
```

After the broker receives the canonical structure back from the map, it then routes the canonical message to the appropriate target. In this scenario, there is only one targeted backend application for the canonical message, but in real world scenarios, there could be other backend systems and services that also use the canonical message as input.

5.5.5 Final message delivery to backend application

Example 5-31 on page 175 shows the result of transforming the canonical xml message into the target backend format. In this scenario, the target format is a message containing fixed length header and detail records.

Example 5-31 Target backend format

```
2012011612225485900000000000000000001|42|1323|9876543210|JS00111223333|JS0011122333
3|1|P|||OPEN|A|ZZ|HEALTHCARE_DEV|ZZ|HEALTHCARE_PS|2007-06-12|10:41:00|00501||100|
4|TGJ23|||26463774|999996666|||||||||||||||1|0340
2012011612225501500000000000000000001|42|1324||000221111|000221111|1|P|||OPEN|A|ZZ
|HEALTHCARE_DEV|ZZ|HEALTHCARE_PS|2007-06-12|10:41:00|00501|112233334|100|4|TGJ23||
|26462967|741234||||1|XX|112233334||||KILDARE||||||1|0340
```

After the message is successfully delivered to our backend application, the scenario ends with the understanding that additional processing needs to occur in our backend application on the message.

5.5.6 Delivery of the acknowledgement to the trading partner

As part of the de-enveloping of the HIPAA EDI 837 message in 5.5.3, “Processing of the message in Sterling B2B Integrator” on page 169, one of the steps in the X12DeenvelopeUnified process was to create an acknowledgement for the message.

In this scenario, we return an EDI 999 message to the trading partner. The EDI 999 works much like the more common EDI 997, but also details any compliance errors that are discovered in the original message.

When the de-envelope runs, it invokes the process EDIEnvelopeUnified, which uses the defined envelopes to prepare the EDI 999 message for transmission back to the trading partner. As part of this enveloping, it also deposits the message into the `/ITSORetail` mailbox, and from there, Sterling File Gateway routes the resulting message, by AS2, to the trading partner.

Example 5-32 is the example EDI 999 that is returned to our trading partner when processing the example HIPAA EDI 837 message.

Example 5-32 Example EDI 999 message.

```
ISA*00*          *01*          *ZZ*HEALTHCARE_PS  *ZZ*HEALTHCARE_DEV
*120118*1051*^*00501*000000007*0*T*:~
GS*FA*DEVELOPMENT*PROFSERV*20120118*1051*17*X*005010~
ST*999*0001~
AK1*HC*42*005010X222A1~
AK9*E*2*2*2~
SE*4*0001~
GE*1*17~
IEA*1*000000007~
```

5.6 Conclusion

This scenario demonstrates the integration of several IBM products into what appears to be, at first glance, a fairly simple inbound flow of messages, with a corresponding return message to our partner if the message is not compliant.

However, even this simple flow can dramatically reduce the processing time and resources needed to deal with the messages in question. By automating this processing, we can improve overall throughput and reduce wait times on the trading partner's end, while also taking advantage of streamlined processing and reduced workloads on the staff of the company implementing this solution.

This scenario is only one possible use of the interaction of the systems in question, and could be expanded upon to deal with any number of different requirements for the quick processing of messages between two trading partners.

In particular, the integration of these various systems could provide an example to a company that only has part of these systems currently deployed. This scenario shows the simplicity of connecting these systems into an integrated whole, and as mentioned, this scenario can be expanded upon to deal with messages of nearly any type.

6

Financial Services scenario

This chapter provides a sample scenario involving two companies: a company acting as a seller, another playing the role of the buyer, and a financial institution, a bank. The bank acts as an intermediary and processes the payment on behalf of the buyer and also responds back to the seller to acknowledge that the payment has been processed correctly.

The seller sends an invoice in a proprietary flat file to the buyer, then the buyer receives and transforms the document into XML format and routes it to the bank. The bank processes the file internally to accomplish the payment and then sends a copy of the XML file back to the seller as an acknowledgement of the payment received. See also Figure 6-4 on page 181.

We assume that the bank internal processing of the payment is accomplished by an existing internal processing system. So our scenario demonstrates the file routing and transformation within a circular flow that starts from the seller, goes to the buyer, and from here to the bank. The process ends when the bank delivers an acknowledgement file back to the seller.

A second scenario in this chapter extends the above scenario to demonstrate the integration of Sterling B2B Integrator with WebSphere Message Broker.

This chapter includes the following sections:

Tip: If you want to implement the scenario presented in this chapter in your own environment, you can download the Project Interchange file for the applications used in this scenario from the ITSO FTP site. For download instructions, see Appendix A, "Additional material" on page 333.

6.1 Business value

IBM provides comprehensive integration solutions that help you optimize your dynamic business network. Implementing this Financial scenario leads to the following business values:

- *Improve business agility* by securely and flexibly integrating with any partner, any system, anywhere. It is crucial to making vital connections with customers, suppliers, and others in your B2B community.
- *Improve operational efficiency* by seamlessly automating manual business and IT processes inside and outside your enterprise. Thus it saves you staff, time, and money, which can be used to grow your business in other key areas.
- *Improve business performance* by providing visibility into actionable information across your key business and IT processes.
- *Empower data movement* by providing assured file delivery with visibility, control, and governance for all data movement.

6.2 Prerequisites: Technical and infrastructure

You need to meet the following prerequisites in order to understand and implement this scenario.

6.2.1 Software prerequisites

The implementation of this scenario is based on the following products that therefore need to be installed in order to implement this solution:

- IBM Sterling B2B Integrator version 5.2.3
- IBM Sterling File Gateway version 2.2.3
- IBM Sterling Connect:Direct version 4.6
- IBM Sterling Connect:Direct File Agent version 1.4
- IBM Sterling Secure Proxy version 3.4
- IBM Sterling Control Center version 5.3
- IBM WebSphere Transformation Extender Design Studio version 8.4
- IBM WebSphere Transformation Extender for Integration Servers version 8.4

Figure 6-1 illustrates the configuration details for the environment used in the buyer network for this scenario.

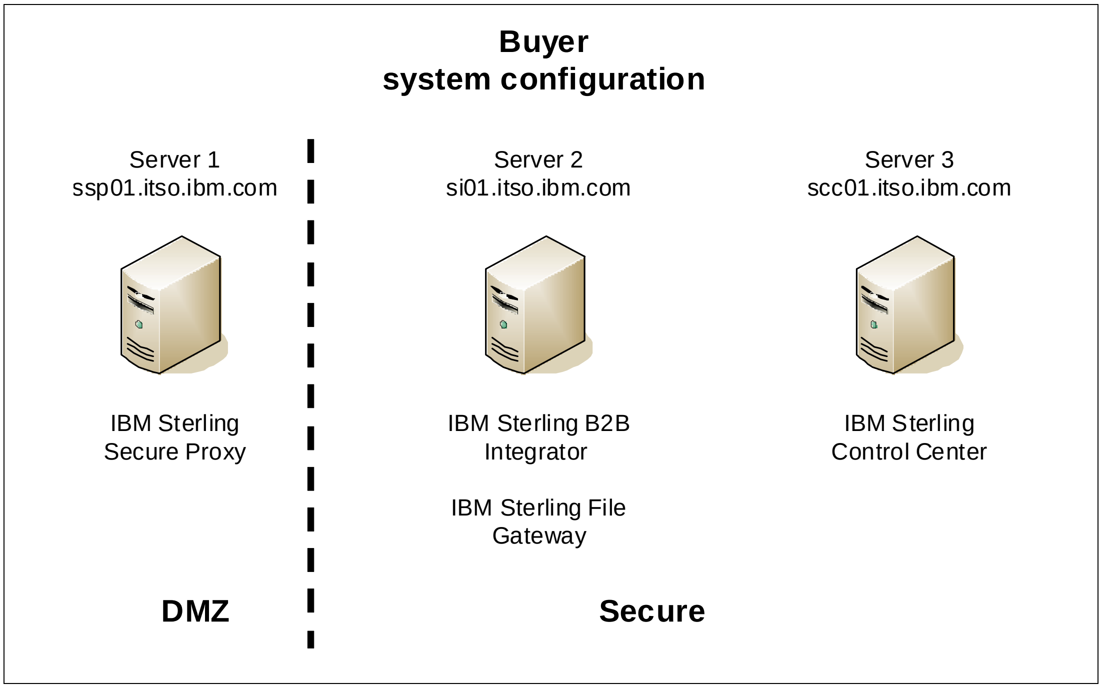


Figure 6-1 Buyer system configuration

Figure 6-2 illustrates the configuration details for the environment used in the seller company for this scenario.

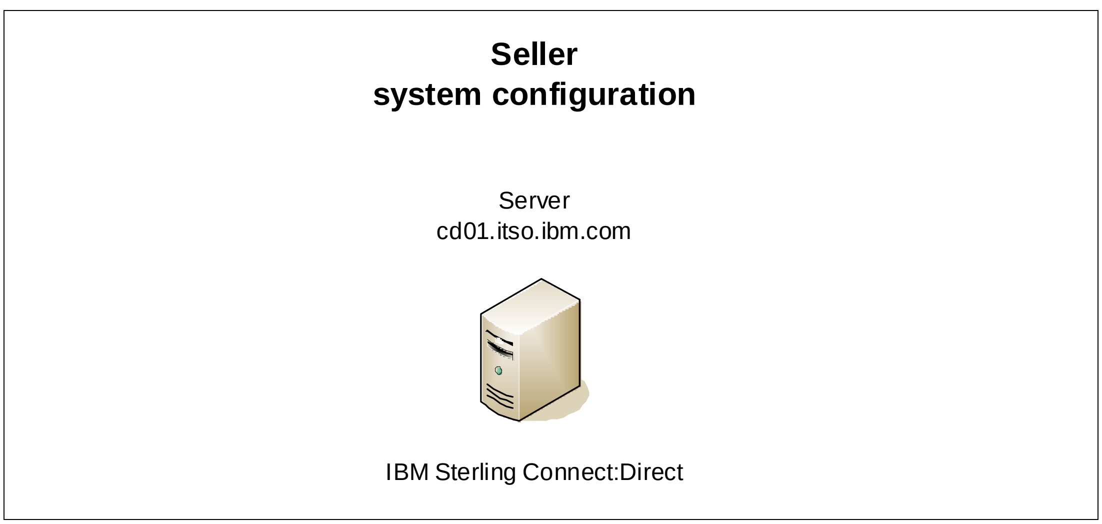


Figure 6-2 Seller system configuration

Figure 6-3 illustrates the configuration details for the environment used in the bank for this scenario.

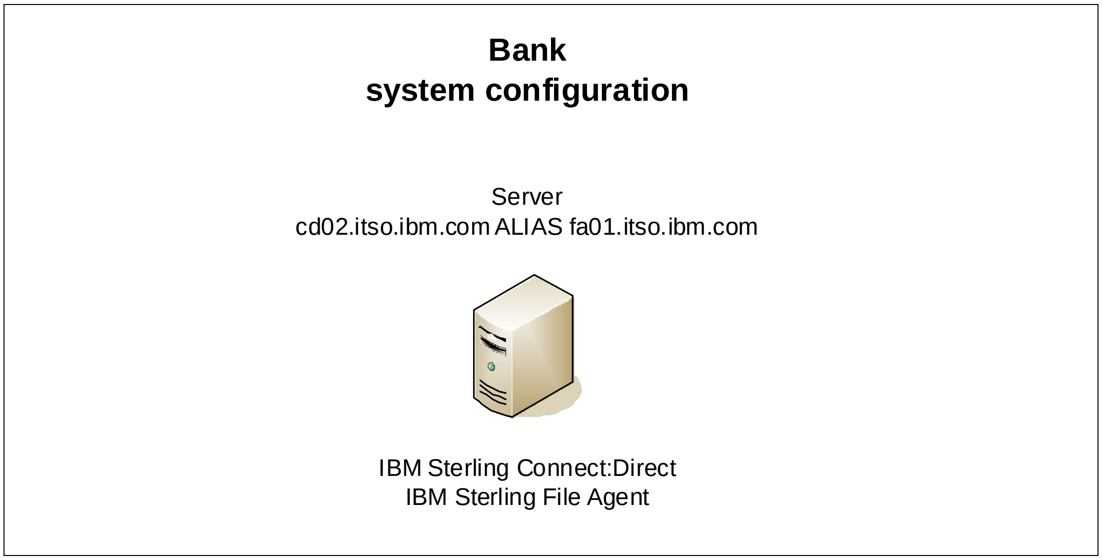


Figure 6-3 Bank system configuration

6.2.2 Skills prerequisites

In order to fully implement and understand this scenario, we assume that you are familiar with the following activities:

- Installing and configuring IBM Sterling Secure Proxy to communicate with Sterling B2B Integrator, Sterling File Gateway, and Sterling Connect:Direct.
- Installing and configuring IBM Sterling File Gateway to receive, handle, and route messages.
- Installing and configuring IBM Sterling Connect:Direct and its File Agent to send, receive, and automatically route messages.
- Installing and configuring IBM Sterling B2B Integrator to receive, handle, translate, and send messages.
- Installing and configuring IBM WebSphere Transformation Extender Design Studio V8.4. This map design tool is used to develop the map used to translate and transform data to the desired format.
- Installing and configuring IBM WebSphere Transformation Extender for Integration Servers V8.4. It is the runtime engine that allows execution of the WebSphere Transformation Extender map on Sterling B2B Integrator.

6.3 Presenting the Financial Services scenario

This section provides an overview of the data flow for the scenario described in Figure 6-4.

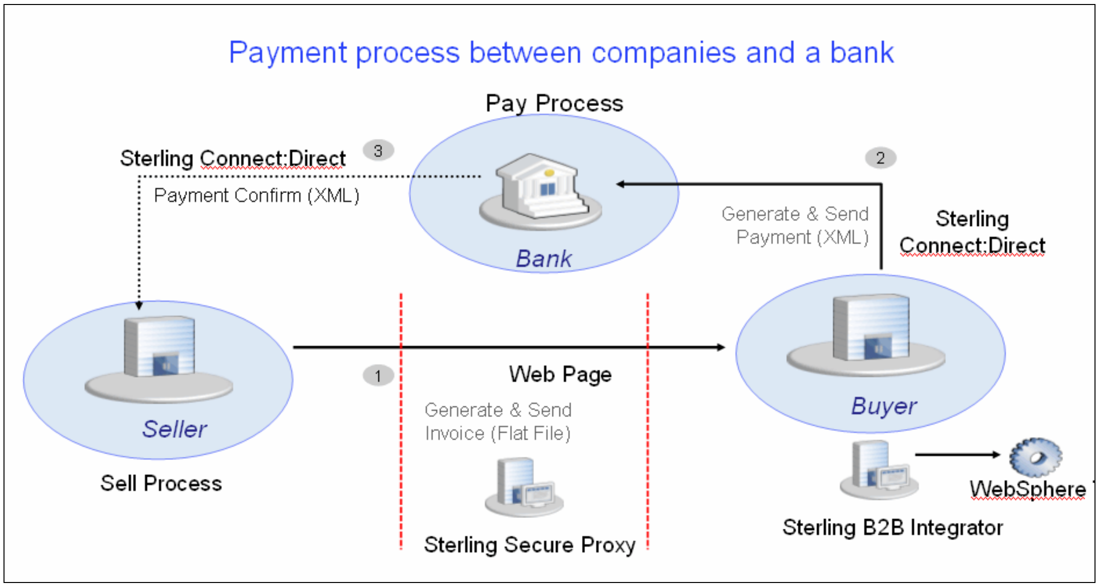


Figure 6-4 Payment Process

For this particular scenario, we are implementing a sample flow that can be described by the following steps:

1. The seller sends an invoice in a flat file to the buyer by http protocol using the web interface IBM Sterling myFileGateway. The file is uploaded to a dedicated seller mailbox on the Sterling File Gateway server installed at the buyer side, and the connection is secured through Sterling Secure Proxy (Figure 6-5).

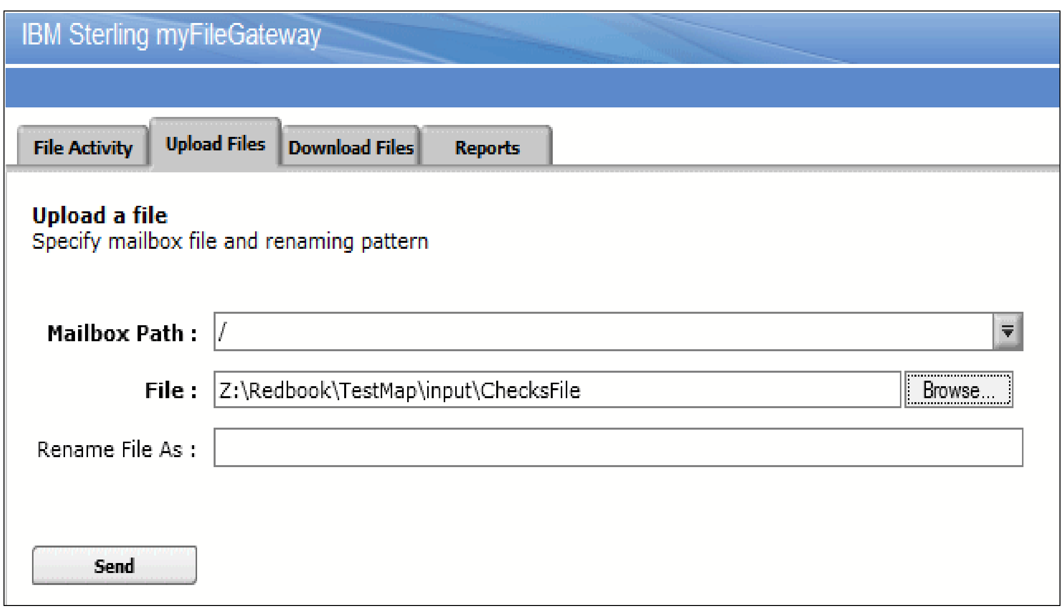


Figure 6-5 myFileGateway file upload

2. The buyer receives the invoice on the seller mailbox, and kicks off a Business Process. It generates and sends the payment to the bank in XML format using the WTX Map service first for generating the XML file, and then the Connect:Direct protocol to send it to the bank (Figure 6-6).

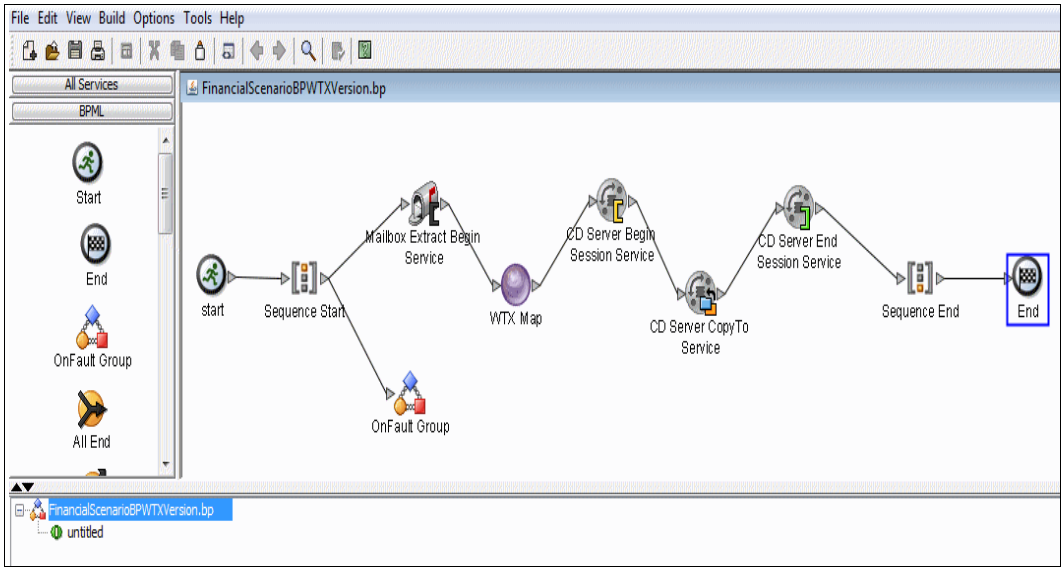


Figure 6-6 Business Process with WTX Map service

3. The bank receives the payment on its Connect:Direct server, then processes it on their legacy systems and sends a copy of the XML file to the seller by Connect:Direct as an acknowledgement of the payment received. The latter is automated by mean of the File Agent, which is monitoring the Connect:Direct directory where the file comes in, and kicks off a Connect:Direct process to send the acknowledgement to the seller. See Figure 6-7 and Figure 6-8.

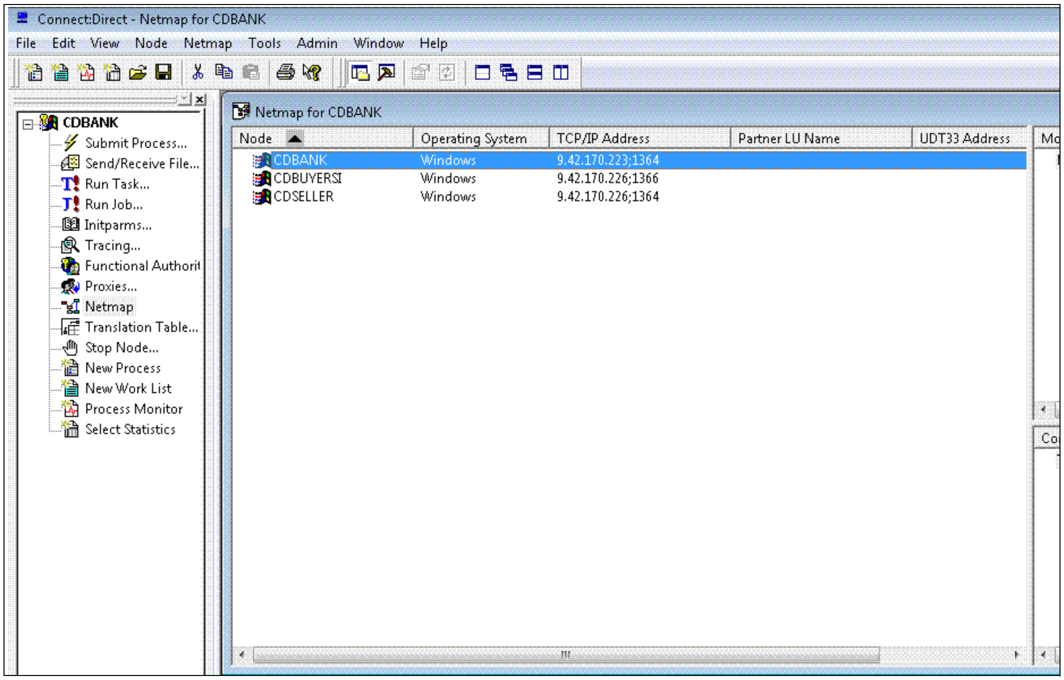


Figure 6-7 Connect:Direct bank's node: CDNODE

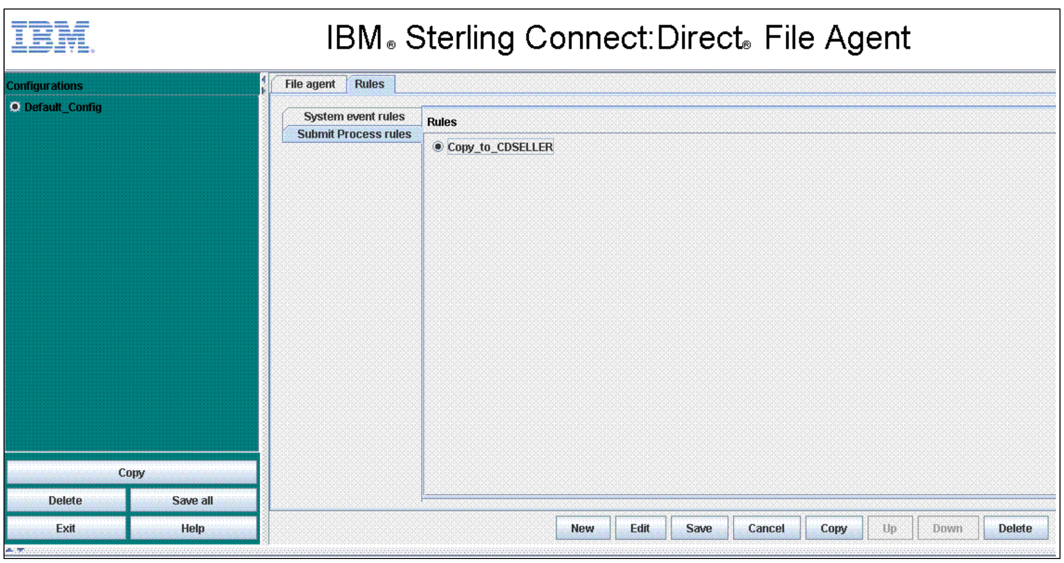


Figure 6-8 IBM Sterling Connect:Direct File Agent

4. The buyer has a Control Center instance to monitor all of the communications (Figure 6-9).

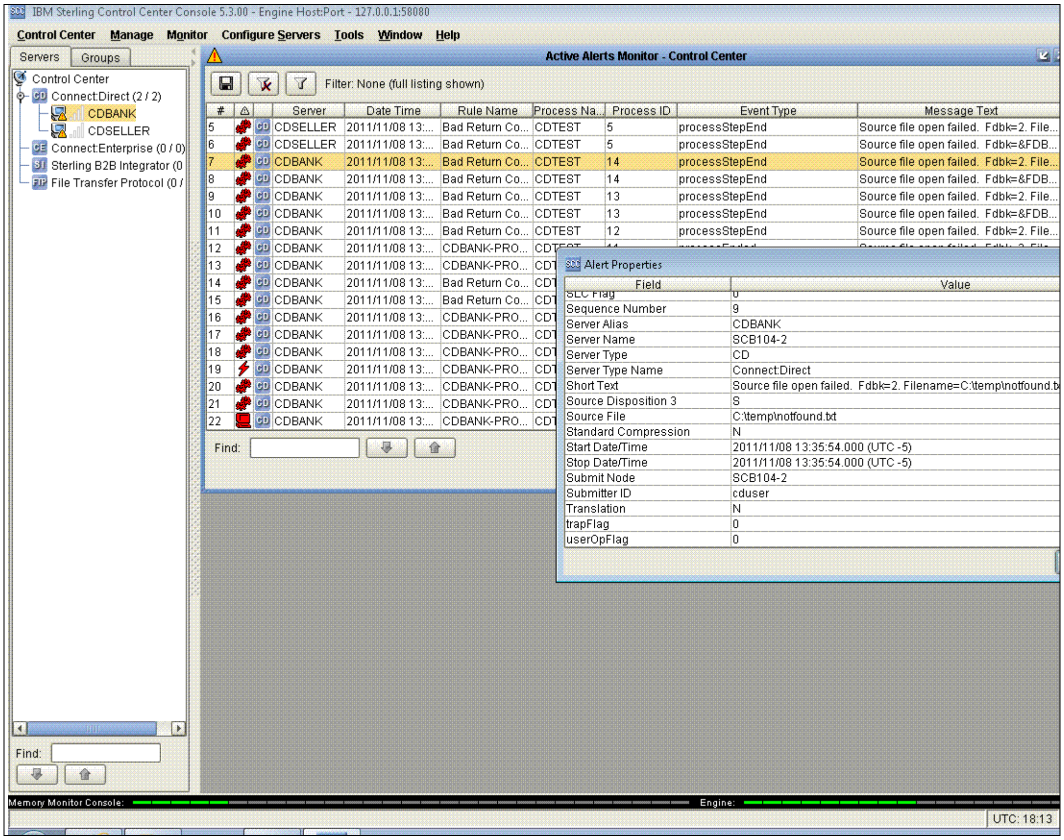


Figure 6-9 IBM Sterling Control Center

Tip: We are demonstrating only a single invoice transaction flow in this section. In most cases, a real financial service scenario contains multiple transactions that represent a full conversation. For example, a buyer sends a purchase order. The seller can respond with an advanced shipping notice and an invoice that is related to the order that was received, then a delivery notice after the payment, and so on.

6.3.1 Alternate scenarios

In step 2 on page 182, we described the use of the WTX Map service for the translation. The WTX Map service requires you to install WebSphere Transformation Extender V8.4 on the same machine as the Sterling B2B Integrator V5.2.3 Server, and configure the Sterling B2B Integrator PATH and CLASSPATH to point to your WebSphere Transformation Extender installation.

It is a very interesting point that shows how Sterling B2B Integrator integrates with WebSphere Transformation Extender seamlessly and effectively.

However, Sterling B2B Integrator includes a number of embedded translation services such as the proprietary Translation service. So of course it is also possible to modify the scenario described in 6.3, “Presenting the Financial Services scenario” on page 181, and for instance, to use the embedded Translation service instead of the WTX Map service as depicted in Figure 6-10.

Tip: WTX Map service provides more flexibility with the WebSphere Transformation Extender maps. For example, the WTX Map service supports multiple inputs and/or outputs, but the Translation service does not.

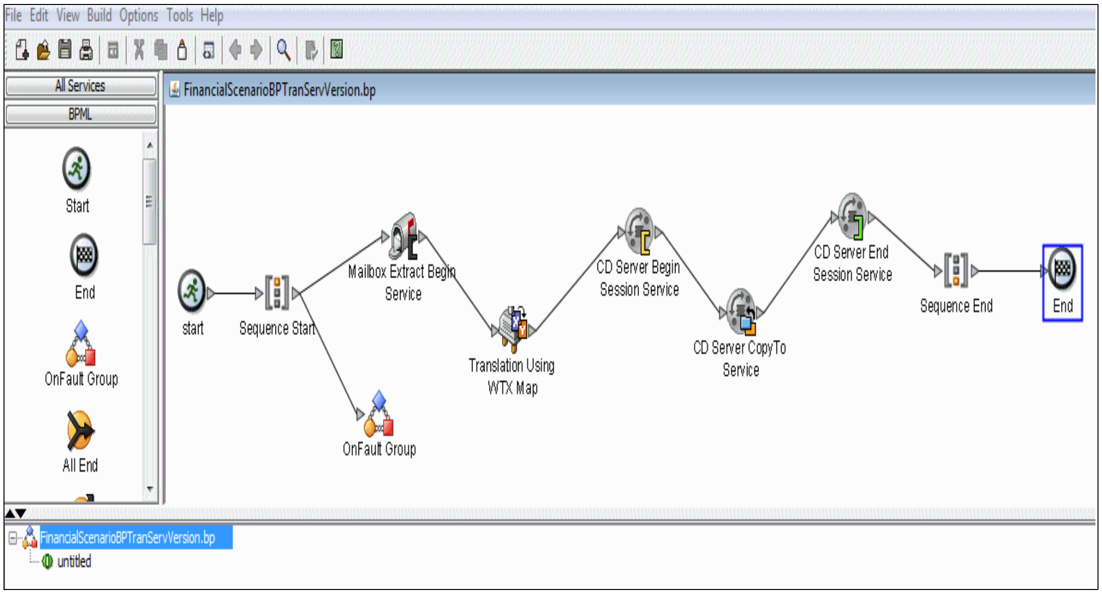


Figure 6-10 Business Process with Translation service

This map shows another interesting aspect of the integration between Sterling B2B Integrator and WebSphere Transformation Extender. The same map used to translate the document by the WTX Map service is reusable to perform the transformation through the embedded Translation Service.

6.4 Configuring the scenario

This section illustrates the installation parameters for the products needed to implement this scenario (listed in 6.2.1, “Software prerequisites” on page 178) and their configurations.

6.4.1 Installing and configuring the IBM Sterling B2B Integrator

Table 6-1 shows the parameters used to install the product on a Windows server in the secure zone of the buyer company network.

Table 6-1 IBM Sterling B2B Integrator installation parameters

Parameter name	Value
Installation Directory	C:\IBM\SI
JDK directory	C:\Program Files\Java\jdk1.6.0_26
JCE jar file	C:\Users\IBM_ADMIN\Desktop\unrestricted.zip
Install jar	C:\Users\IBM_ADMIN\Desktop\SI_5020300.jar
IP address	localhost
Initial port	8080

Parameter name	Value
Passphrase	*********
E-Mail address	administrator@mail.itso.ibm.com
SMTP Mail Server	localhost
Database Vendor	DB2
Database user name	si
Database password	*********
Database catalog name	SI52
Database host	db2.itso.ibm.com
Database port	50000
Database driver	C:\db2jcc.jar C:\db2jcc_license_cu.jar

Special considerations

The following considerations apply:

- The installation directory is the location where SI needs to be installed. Make sure that no space is given in the folder location.The installation assumes the presence of a DB2 database for Sterling B2B Integrator. If you do not have DB2 installed, follow the procedures in the *DB2 Installation Manual* at this website:

  http://public.dhe.ibm.com/ps/products/db2/info/vr97/pdf/en_US/DB2InstallingServers-db2ise972.pdf

- During the pre-installation phase of Sterling B2B Integrator, a DB connectivity test is required. Note that it is mandatory and, unless completed successfully, the installation does not start.

Important: The installation script creates tables and indexes. Certain tables require a page size of 32K. You must have a table space to accommodate such tables on DB2. If you don not create a table space on DB2, you get the installation failure shown in Figure 6-11.

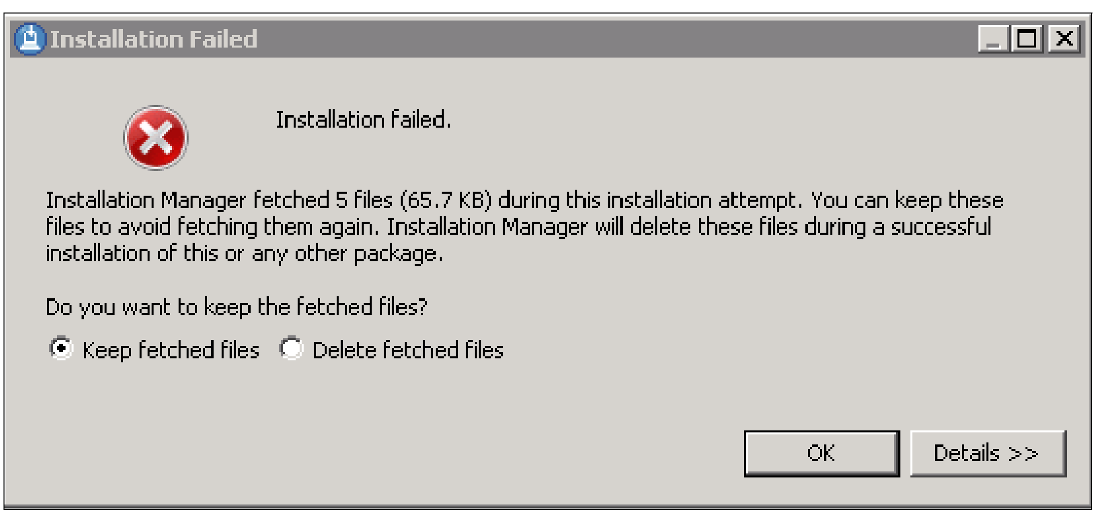


Figure 6-11 DB2 Error message

Before installing the product on your environment, for more information, check the Sterling B2B Integrator 5.2.3 documentation with special attention to the "System Requirements" section and the *Installation Guide* available at this website:

http://help.sterlingcommerce.com/SB2BI52/index.jsp

Configuring IBM Sterling B2B Integrator

We need to implement a Sterling B2B Integrator Business Process to extract the message from the mailbox to which the seller uploads the invoice, then to translate the message into XML and route it to the bank by the Connect:Direct protocol. Optionally, a copy of the same XML content can be sent as a message to a WebSphere MQ queue manager for further processing in the buyer enterprise service bus. It is discussed in more detail as a possible extension of the Financial Services scenario in 6.6, "Extended scenario" on page 238.

In order to use the Connect:Direct services in Sterling B2B Integrator, we need to configure a Connect:Direct Server Adapter instance first, which in its turn requires to configure the Nodes, to define a Netmap and the Netmap Cross Reference.

Figure 6-12 and Figure 6-13 illustrate the configurations for the Nodes, the Netmap, and the Netmap Cross Reference used for the subsequent Connect:Direct Server Adapter configuration.

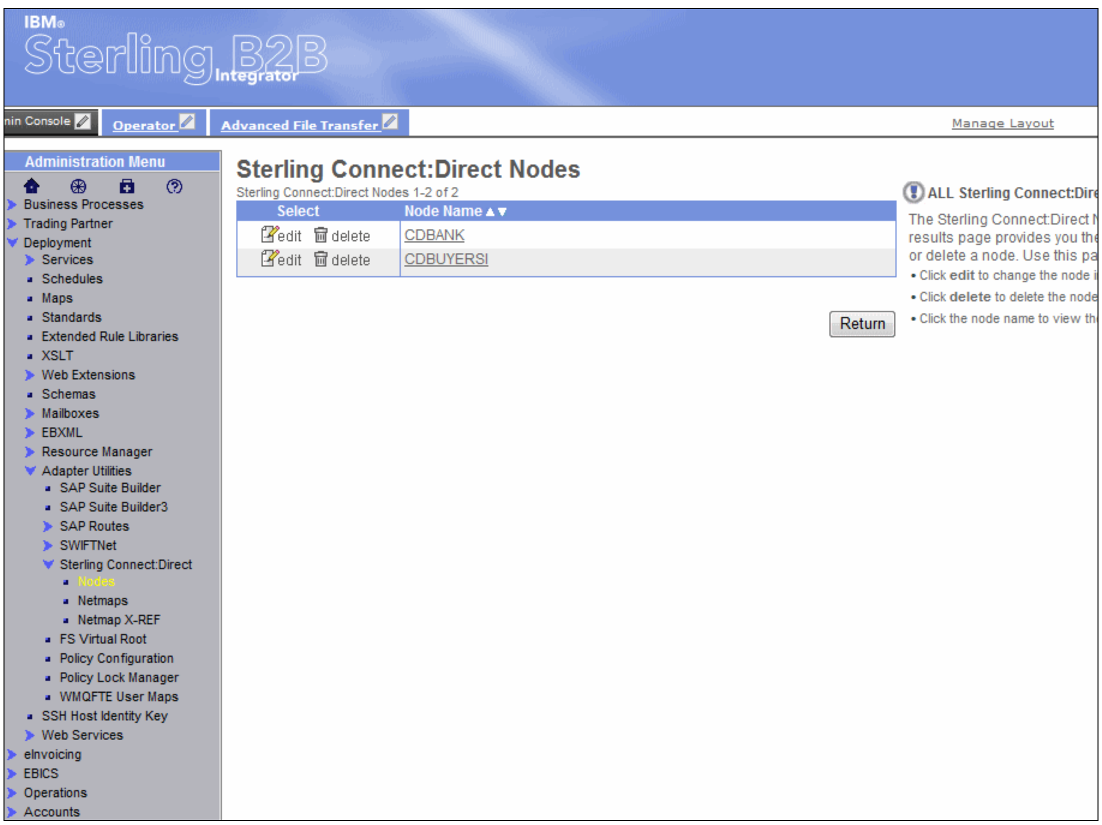


Figure 6-12 Connect:Direct nodes

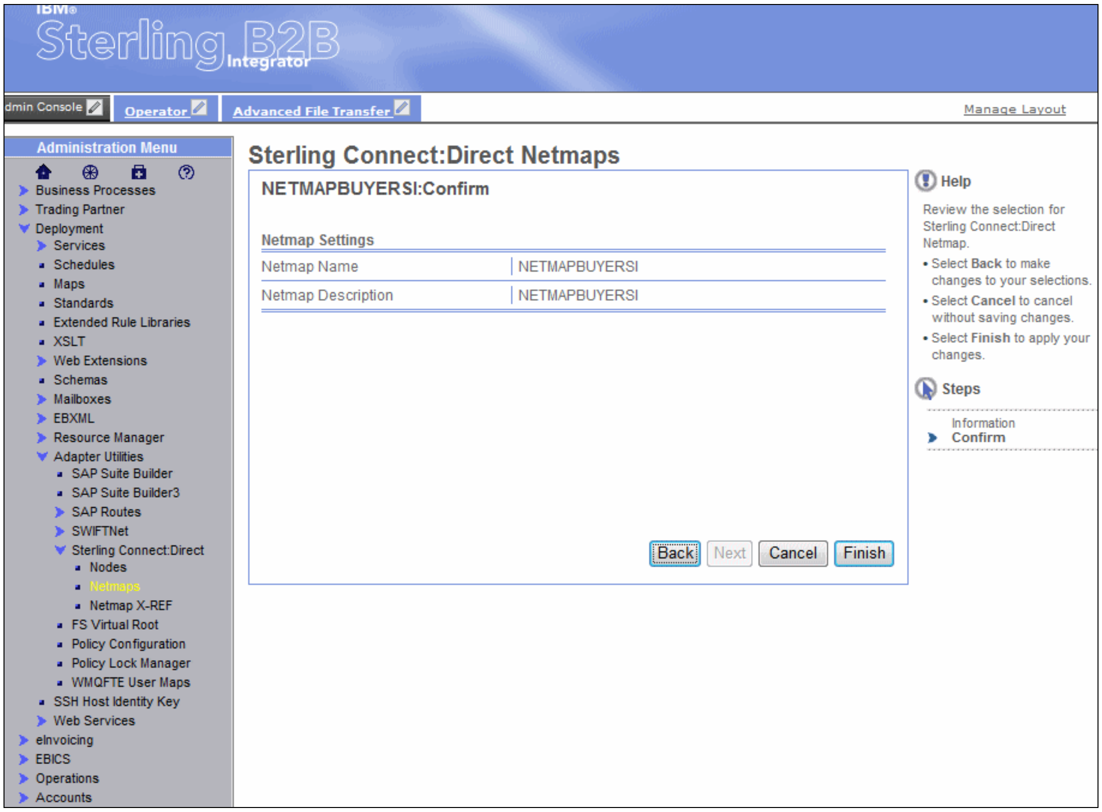


Figure 6-13 Connect:Direct Netmap

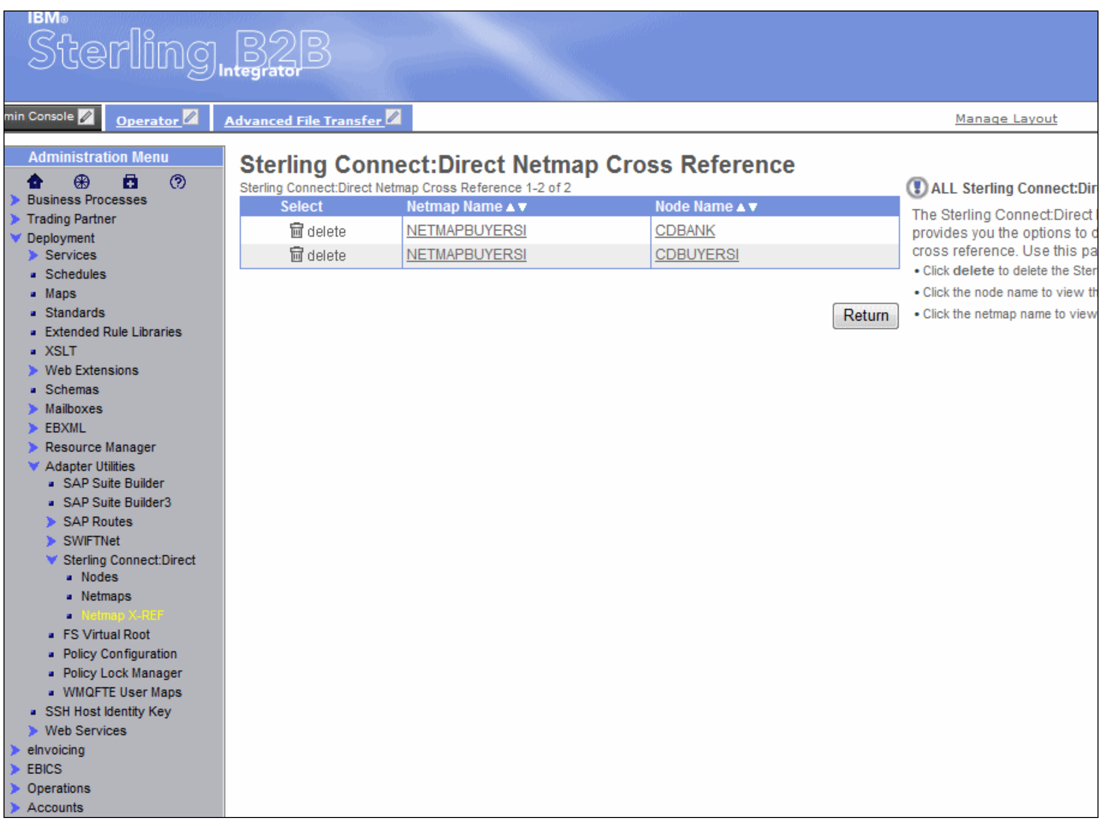


Figure 6-14 Connect:Direct Netmap Cross Reference

Table 6-2 illustrates the parameters used for the Connect:Direct Server Adapter Configuration. It includes the parameters used to configure the nodes illustrated in Figure 6-12 on page 187.

Table 6-2 Sterling B2B Integrator Connect:Direct Server Adapter node: CDBUYERSI

Service settings	
Service Type	Sterling Connect:Direct Server Adapter
Description	CDBUYERSI
System Name	CDBUYERSI
Group Name	None
Sterling Connect:Direct Server Node Name	CDBUYERSI
Sterling Connect:Direct Perimeter Services Option	node1 & local
Sterling Connect:Direct Server Port	1366
Firewall Ports	None provided
Max locally initiated (pnode) sessions allowed	5
Max remotely initiated (snode) sessions allowed	5
Document Storage	File System
NetMap Check	No
Buffer-size for Copy	32768
Number of short-term session retry attempts	5
Interval between short-term session attempts (seconds)	5
Number of long-term session retry attempts	5
Interval between long-term session attempts (minutes)	1
Retry Remote File Allocation Errors	No
Max Session Establishment Timeout value in Seconds	600
Max Socket Read Timeout value in Seconds	90
Server Start Option	Warm
RunTask Business Process Name	None provided
RunJob Business Process Name	None provided
Max Run Task Forwarding Timeout value in Seconds	300
Max Run Job Forwarding Timeout value in Seconds	30
Encryption using Secure+	Disabled
Enable Netmap Node Override	Yes

Service settings	
Sterling Connect:Direct Server Netmap	NETMAPBUYERSI
Number of Nodes in Netmap	2
Connect:Direct Server Node	Name: CDBANK Host: 9.42.170.223 Port: 1364 Max Pnode sessions: 5 Max Snode sessions: 5 Alternate Comm Info: Secure+ Option: disabled Cipher Suites: none
Connect:Direct Server Node	Name: CDBUYERSI Host: 9.42.170.226 Port: 1366 Max Pnode sessions: 5 Max Snode sessions: 5 Alternate Comm Info: Secure+ Option: disabled Cipher Suites: none

The Business Process shown in Example 6-1 is an example of implementation that uses the WTX Map service for the communication.

Example 6-1 Business Process with WTX Map service

```
Description: Version 1.0
Business Process Definition:

<process name = "FinancialScenarioBPWTXVersion">
  <sequence name="Sequence Start">
    <operation name="Mailbox Extract Begin Service">
      <participant name="MailboxExtractBegin"/>
      <output message="MailboxExtractBeginServiceTypeInputMessage">
        <assign to="." from="*"></assign>
        <assign to="CommitNow">Yes</assign>
        <assign to="MessageId"
from="/ProcessData/RoutingRequest/RoutingRequest/MessageId[last()]/text()
"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="WTX Map">
      <participant name="WTXMapService"/>
      <output message="WTXMapServiceTypeInputMessage">
        <assign to="." from="*"></assign>
        <assign to="MapName">WTXDemo</assign>
        <assign to="in1">//PrimaryDocument</assign>
        <assign to="out1">//PrimaryDocument</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
```

```
      </input>
    </operation>

    <operation name="CD Server Begin Session Service">
      <participant name="CDServerBeginSession"/>
      <output message="CDServerBeginSessionServiceTypeInputMessage">
        <assign to="LocalCDNodeName">CDBUYERSI</assign>
        <assign to="ProcessName">SNDTOCD</assign>
        <assign to="RemoteCDNodeName">CDBANK</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="CD Server CopyTo Service">
      <participant name="CDServerCopyTo"/>
      <output message="CDServerCopyToServiceTypeInputMessage">
        <assign to="RemoteFileName">c:\CDBANKCOMM\CDBUYERIN\bankCheck.xml</assign>
        <assign to="SessionToken" from="SessionToken/SessionId/text()"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="CD Server End Session Service">
      <participant name="CDServerEndSession"/>
      <output message="CDServerEndSessionServiceTypeInputMessage">
        <assign to="SessionToken" from="SessionToken/SessionId/text()"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <onFault>
      <sequence name="Sequence Start">
        <assign name="Assign" to="OnFaultError">En error occurred in the 
sequence</assign>
        <operation name="CD Server End Session Service">
          <participant name="CDServerEndSession"/>
          <output message="CDServerEndSessionServiceTypeInputMessage">
            <assign to="SessionToken" 
from="SessionToken/SessionId/text()"></assign>
            <assign to="." from="*"></assign>
          </output>
          <input message="inmsg">
            <assign to="." from="*"></assign>
          </input>
        </operation>
```

```
      </sequence>
    </onFault>
  </sequence>
</process>
```

As discussed in section 6.3.1, “Alternate scenarios” on page 184, it is also possible to use the Translation Service for the translation step as shown in the Business Process in Example 6-2.

Example 6-2 Business Process with Translation Service

```
Description: Version 1.0
Business Process Definition:

<process name = "FinancialScenarioBPTranServVersion">
  <sequence name="Sequence Start">
    <operation name="Mailbox Extract Begin Service">
      <participant name="MailboxExtractBegin"/>
      <output message="MailboxExtractBeginServiceTypeInputMessage">
        <assign to="." from="*"></assign>
        <assign to="CommitNow">Yes</assign>
        <assign to="MessageId"
from="/ProcessData/RoutingRequest/RoutingRequest/MessageId[last()]/text()
"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="Translation Using WTX Map">
      <participant name="Translation"/>
      <output message="TranslationTypeInputMessage">
        <assign to="." from="*"></assign>
        <assign to="map_name">WTXDemo</assign>
        <assign to="output_report_to_process_data">Yes</assign>
        <assign to="output_to_process_data">No</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="CD Server Begin Session Service">
      <participant name="CDServerBeginSession"/>
      <output message="CDServerBeginSessionServiceTypeInputMessage">
        <assign to="LocalCDNodeName">CDBUYERSI</assign>
        <assign to="ProcessName">SNDTOCD</assign>
        <assign to="RemoteCDNodeName">CDBANK</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="CD Server CopyTo Service">
```

```
      <participant name="CDServerCopyTo"/>
      <output message="CDServerCopyToServiceTypeInputMessage">
        <assign to="RemoteFileName">c:\CDBANKCOMM\CDBUYERIN\bankCheck.xml</assign>
        <assign to="SessionToken" from="SessionToken/SessionId/text()"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="CD Server End Session Service">
      <participant name="CDServerEndSession"/>
      <output message="CDServerEndSessionServiceTypeInputMessage">
        <assign to="SessionToken" from="SessionToken/SessionId/text()"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <onFault>
      <sequence name="Sequence Start">
        <assign name="Assign" to="OnFaultError">En error occurred in the
sequence</assign>
        <operation name="CD Server End Session Service">
          <participant name="CDServerEndSession"/>
          <output message="CDServerEndSessionServiceTypeInputMessage">
            <assign to="SessionToken"
from="SessionToken/SessionId/text()"></assign>
            <assign to="." from="*"></assign>
          </output>
          <input message="inmsg">
            <assign to="." from="*"></assign>
          </input>
        </operation>

      </sequence>
    </onFault>
  </sequence>
</process>
```

The third Business Process example shown in Example 6-3 shows how a copy of the same XML content for the bank can be also sent as a message to a WebSphere MQ queue manager for further processing in the Buyer enterprise service bus as described in 6.6, “Extended scenario” on page 238.

Example 6-3 Business Process with MQ communication

```
Description: Version 1.2
Business Process Definition:

<process name = "FinancialScenarioBPWTXInvokeExtension">
  <sequence name="Sequence Start">
    <operation name="Mailbox Extract Begin Service">
      <participant name="MailboxExtractBegin"/>
      <output message="MailboxExtractBeginServiceTypeInputMessage">
        <assign to="." from="*"></assign>
        <assign to="CommitNow">Yes</assign>
        <assign to="MessageId"
from="/ProcessData/RoutingRequest/RoutingRequest/MessageId[last()]/text()
"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="WTX Map">
      <participant name="WTXMapService"/>
      <output message="WTXMapServiceTypeInputMessage">
        <assign to="." from="*"></assign>
        <assign to="MapName">WTXDemo</assign>
        <assign to="in1">//PrimaryDocument</assign>
        <assign to="out1">//PrimaryDocument</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="Invoke Business Process Service">
      <participant name="InvokeBusinessProcessService"/>
      <output message="InvokeBusinessProcessServiceTypeInputMessage">
        <assign to="." from="*"></assign>
        <assign to="INVOKE_MODE">INLINE</assign>
        <assign to="WFD_NAME">FinancialScenarioMQExtension</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <sequence name="CDSequence">
      <operation name="CD Server Begin Session Service">
        <participant name="CDServerBeginSession"/>
        <output message="CDServerBeginSessionServiceTypeInputMessage">
          <assign to="LocalCDNodeName">CDBUYERSI</assign>
          <assign to="ProcessName">SNDTOCD</assign>
          <assign to="RemoteCDNodeName">CDBANK</assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
```

```
          <assign to="." from="*"></assign>
        </input>
      </operation>

      <operation name="CD Server CopyTo Service">
        <participant name="CDServerCopyTo"/>
        <output message="CDServerCopyToServiceTypeInputMessage">
          <assign
to="RemoteFileName">c:\CDBANKCOMM\CDBUYERIN\bankCheck.xml</assign>
          <assign to="SessionToken" from="SessionToken/SessionId/text()"></assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>

      <operation name="CD Server End Session Service">
        <participant name="CDServerEndSession"/>
        <output message="CDServerEndSessionServiceTypeInputMessage">
          <assign to="SessionToken" from="SessionToken/SessionId/text()"></assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>

      <onFault>
        <sequence name="Sequence Start">
          <assign name="Assign" to="OnFaultError">En error occurred in the
sequence</assign>
          <operation name="CD Server End Session Service">
            <participant name="CDServerEndSession"/>
            <output message="CDServerEndSessionServiceTypeInputMessage">
              <assign to="SessionToken"
from="SessionToken/SessionId/text()"></assign>
              <assign to="." from="*"></assign>
            </output>
            <input message="inmsg">
              <assign to="." from="*"></assign>
            </input>
          </operation>

        </sequence>
      </onFault>
    </sequence>
    <onFault>
      <assign to="Error">Error on First Sequence</assign>
    </onFault>
  </sequence>
</process>
```

Example 6-4 shows the Business Process invoked by the Business Process in Example 6-3 on page 194 for the actual communication with the WebSphere MQ queue manager.

Example 6-4 Business Process invoked for the actual MQ communication

```
Description: Version 1.0
Business Process Definition:

<process name = "FinancialScenarioMQExtension">
  <sequence name="PutFileInMQ">
    <operation name="WSMQ Open Session">
      <participant name="WSMQ_OpenSession"/>
      <output message="WSMQOpenSessionInputMessage">
        <assign to="wsmq_channel">B2B.FINANCE.SBI</assign>
        <assign to="wsmq_hostname">9.42.170.226</assign>
        <assign to="wsmq_port">1414</assign>
        <assign to="wsmq_qmanager">B2B02</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="WSMQ Open Queue">
      <participant name="WSMQ_OpenQueue"/>
      <output message="WSMQOpenQueueInputMessage">
        <assign to="wsmq_MQOO_type">PUT</assign>
        <assign to="wsmq_qname">FINANCE.B2B.ESB.REQUEST</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="WSMQ Put Message">
      <participant name="WSMQ_PutMessage"/>
      <output message="WSMQPutMessageInputMessage">
        <assign to="wsmq_MQMD_expiry">0</assign>
        <assign to="wsmq_MQMD_msgType">DATAGRAM</assign>
        <assign to="wsmq_qname">FINANCE.B2B.ESB.REQUEST</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="WebSphereMQ Suite Commit">
      <participant name="WSMQ_Commit"/>
      <output message="WSMQCommitInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
```

```
      </input>
    </operation>

    <operation name="WSMQ Close Queue">
      <participant name="WSMQ_CloseQueue"/>
      <output message="WSMQCloseQueueInputMessage">
        <assign to="wsmq_qname">FINANCE.B2B.ESB.REQUEST</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="WSMQ Close Session">
      <participant name="WSMQ_CloseSession"/>
      <output message="WSMQCloseSessionInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <onFault>
      <sequence>
        <operation name="WebSphereMQ Suite Close Queue">
          <participant name="WSMQ_CloseQueue"/>
          <output message="WSMQCloseQueueInputMessage">
            <assign to="." from="*"></assign>
            <assign to="wsmq_qname">FINANCE.B2B.ESB.REQUEST</assign>
          </output>
          <input message="inmsg">
            <assign to="." from="*"></assign>
          </input>
        </operation>

        <operation name="WebSphereMQ Suite Close Session">
          <participant name="WSMQ_CloseSession"/>
          <output message="WSMQCloseSessionInputMessage">
            <assign to="." from="*"></assign>
          </output>
          <input message="inmsg">
            <assign to="." from="*"></assign>
          </input>
        </operation>

      </sequence>
    </onFault>
  </sequence>
</process>
```

6.4.2 Installing and configuring the IBM Sterling File Gateway

This section describes the installation and the configuration of Sterling File Gateway at the Buyer company as it is required to allow the Seller company to send the invoice to Buyer using myFileGatway.

Sterling File Gateway 2.2.3 requires Sterling B2B Integrator 5.2.3, so the following installation steps assume that you have already installed Sterling B2B Integrator 5.2.3.

> **Tip:** In this section we are presenting a summary of the installation steps performed to install Sterling File Gateway version 2.2.3 for this specific scenario. However, if you want to replicate the installation in your own environment, make sure to see the *IBM Sterling File Gateway Installation Guide* at this website, as there are important instructions to follow:
>
> `http://help.sterlingcommerce.com/SFG22/index.jsp`

Installation procedure

Follow these steps:

1. Stop Sterling B2B Integrator 5.2.3.
2. Close all command prompt windows.
3. Install the Sterling File Gateway jar file:
   a. Open a command prompt window.
   b. Run the following command:

      ```
      install_dir\bin\InstallService.cmd install_dir\packages\filegateway_2020300
      ```

Configuring IBM Sterling File Gateway

In order to configure Sterling File Gateway to be ready to accept connections from the Seller through myFileGateway, we need to complete the tasks described in the following sections:

Configuration steps

Follow these steps:

1. From the menu Participants:
   a. Create community: FINANCIAL
   b. Create partners:
      - BUYER
      - SELLER
2. From the menu Routes:
   a. Create Template: TemplateSellerStatic.
   b. Create Channel: Creating a channel builds a routing between the producer and the consumer that use the template created previously.

Figure 6-15 illustrates the parameters used for the community configuration.

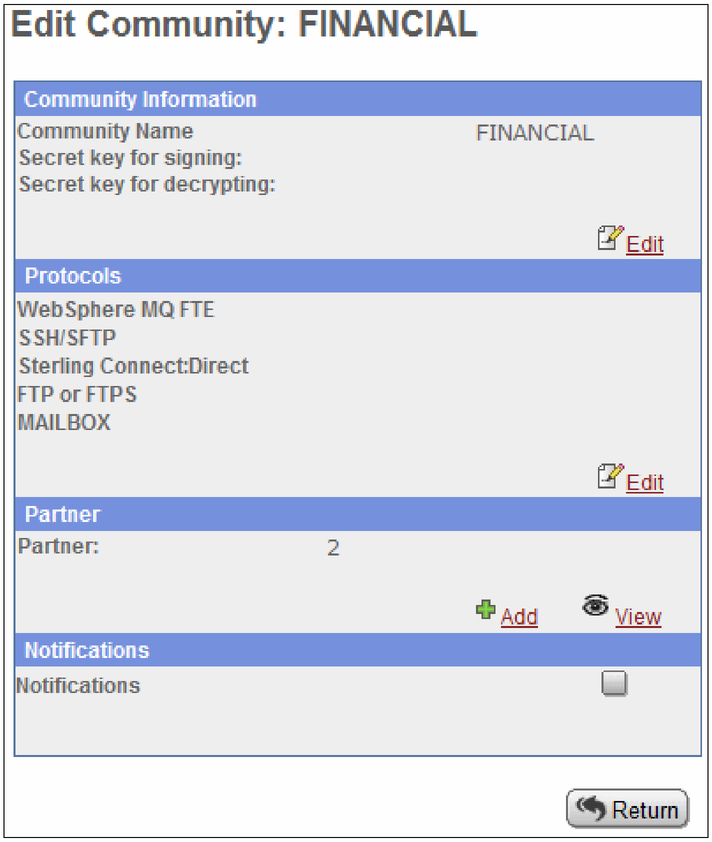


Figure 6-15 Community: FINANCIAL

Figure 6-16 and Figure 6-17 show the parameters used for the partners setup: BUYER and SELLER.

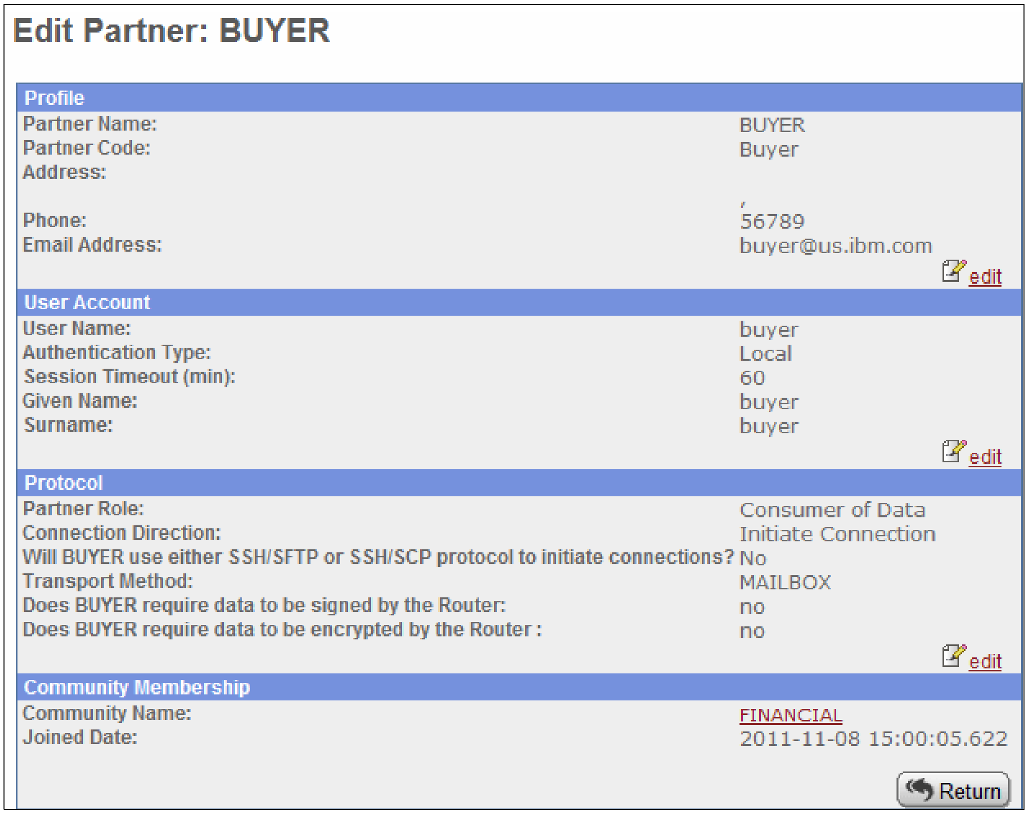


Figure 6-16 Partner: BUYER

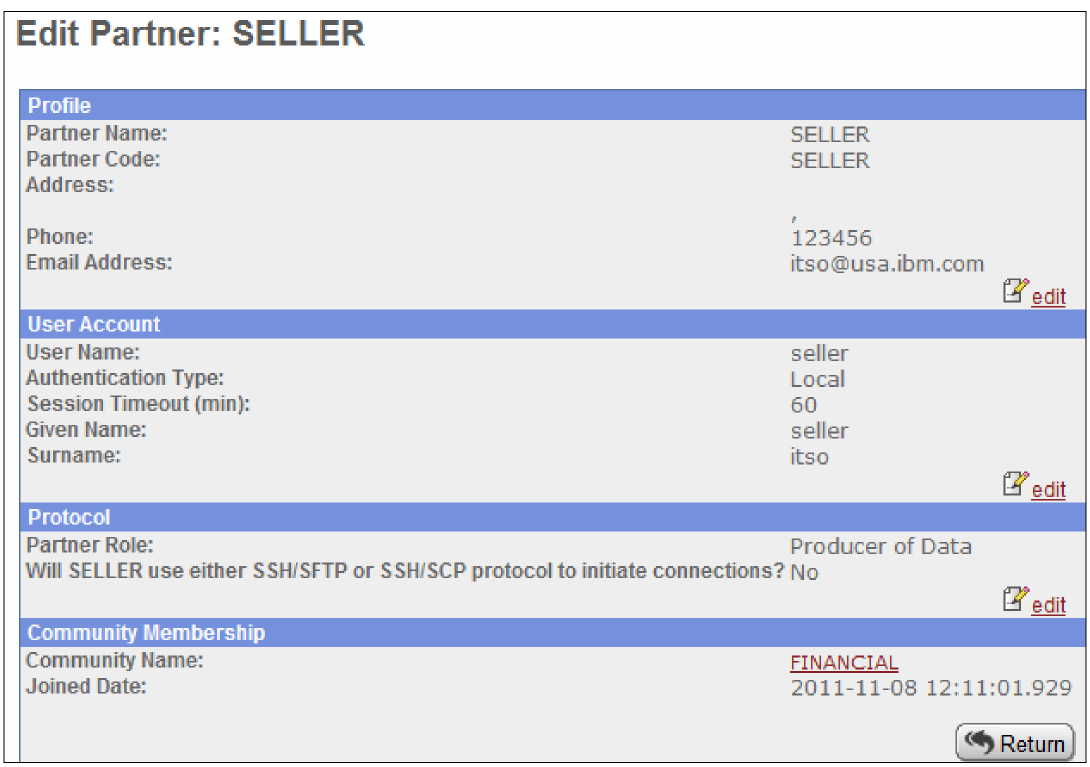


Figure 6-17 Partner: SELLER

Figure 6-18 illustrates the parameters used for creating the template.

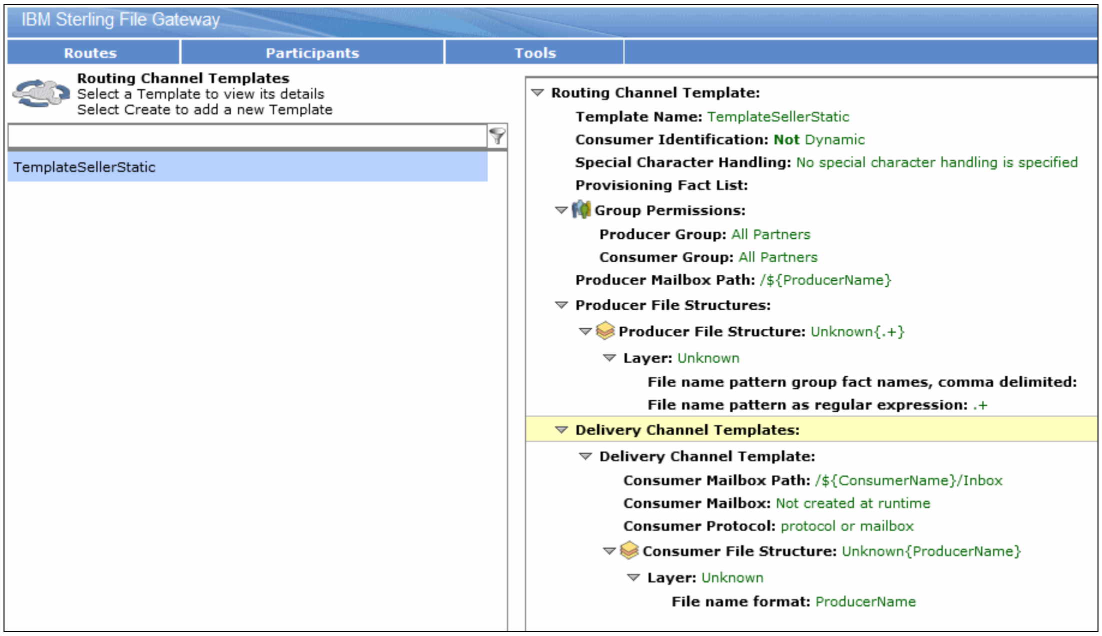


Figure 6-18 Template: TemplateSellerStatic

Figure 6-19 shows the routing channel using the previous template.

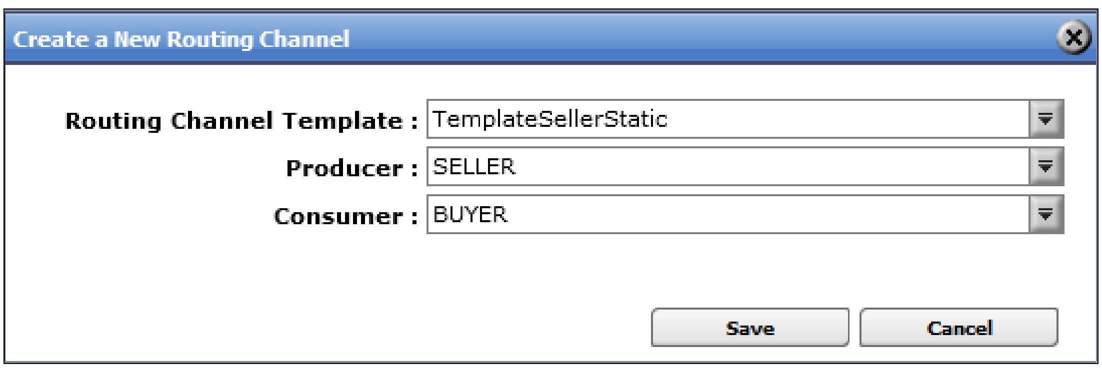


Figure 6-19 Routing Channel

6.4.3 Installing and configuring the IBM Sterling Connect Direct

This section describes the installation and the configuration of IBM Sterling Connect:Direct at the Bank company. It is required to allow the Buyer company to send the Payment to the Bank company using IBM Sterling Connect:Direct and also to allow the Bank company to send payment acknowledgement to the Seller company.

Instructions: In this section we are presenting a summary of the installation steps performed to install IBM Sterling Connect:Direct version 4.6 for this specific scenario. However, if you want to replicate the installation in your own environment, see the *IBM Sterling Connect:Direct Installation Guide*, as there are important instructions to follow:

http://www.sterlingcommerce.com/documentation/home/MFT/ConnectDirect/Windows/CD%20Windows46/CDWindowsGettingStarted.pdf

Installation procedure

Follow these steps to install IBM Sterling Connect Direct:

1. IBM Sterling Connect:Direct needs a database for its process queue and logs. Its installation package has MSSQL Express available, but several other databases are supported. The database must be installed prior to IBM Sterling Connect:Direct installation.
2. Run the setup.exe from IBM Sterling Connect:Direct product directory: `C:\SterlingConnect Direct V5.1.0\Direct V 4.6.00 for Microsoft Windows\CDWin4600\Server and Requester\setup.exe.`
3. The installation asks for several parameters. Make sure to select a "Custom" installation so that it will be possible to enter a Local Node Name and the IP and port-number that this IBM Sterling Connect:Direct instance will use. Otherwise, default values are used.

Configuring IBM Sterling Connect:Direct

In order to configure IBM Sterling Connect:Direct to be ready to accept connections from the Buyer and start connections to Seller, the following steps must be completed.

Configuration steps

Follow these steps to configure IBM Sterling Connect Direct:

1) Logon to the C:D Requester giving the credentials of the user-id used for installing IBM Sterling Connect:DIrect.

2) Select **Netmap**.

3) Create nodes for companies BUYER and SELLER as shown in Table 6-3.

Table 6-3 Nodes for companies BUYER and SELLER

Connect:Direct Server Node	Name: CDSELLER Host: 99.42.170.226 Port: 1364 Max Pnode sessions: 5 Max Snode sessions: 5 Alternate Comm Info: Secure+ Option: disabled Cipher Suites: none
Connect:Direct Server Node	Name: CDBUYERSI Host: 9.42.170.226 Port: 1366 Max Pnode sessions: 5 Max Snode sessions: 5 Alternate Comm Info: Secure+ Option: disabled Cipher Suites: none

Figure 6-20 shows the Sterling Connect:Direct Configuration.

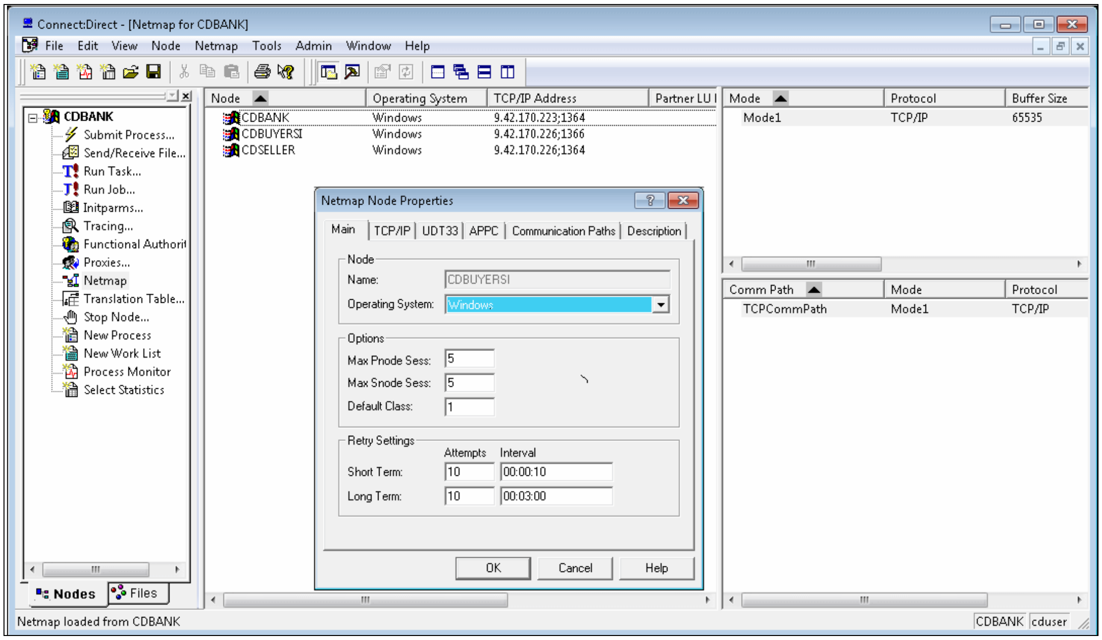


Figure 6-20 Sterling Connect:Direct Configuration

4) Select **Proxies**.

5) Create proxies for the companies BUYER and SELLER as shown in Table 6-4.

Table 6-4 Proxies for the companies BUYER and SELLER

Connect:Direct Server Node	Remote Userid: ANY Remote Node: CDSELLER Local Userid: cduser Local Password: *****
Connect:Direct Server Node	Remote Userid: ANY Remote Node: CDBUYERSI Local Userid: cduser Local Password: *****

Figure 6-21 shows the user proxies. Figure 6-22 shows the window for editing user proxies. Figure 6-23 shows Edit User Proxies - Directories window.

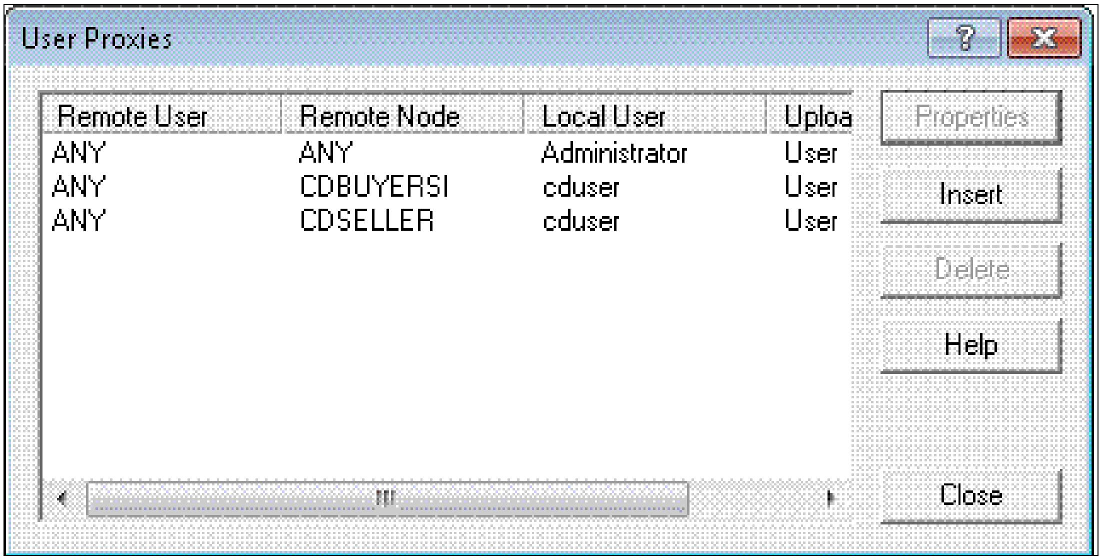


Figure 6-21 User Proxies

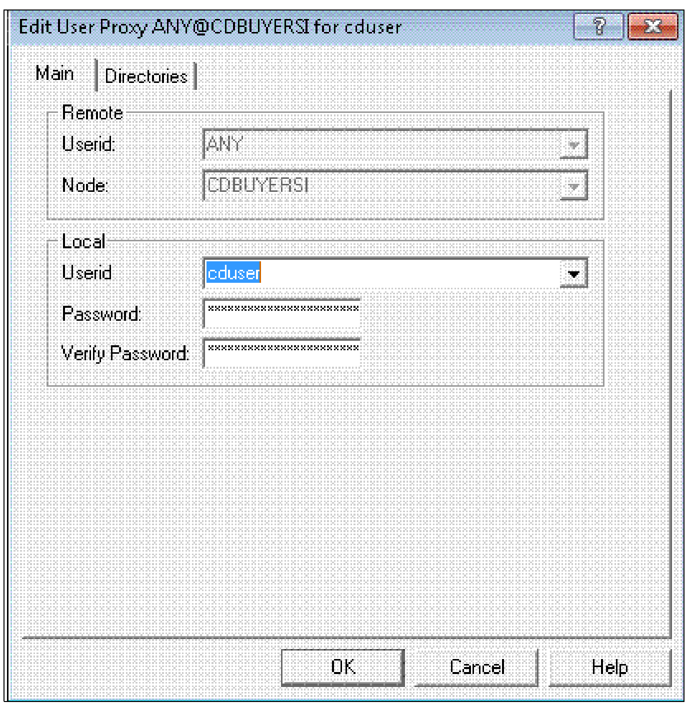


Figure 6-22 Edit User Proxies - Main

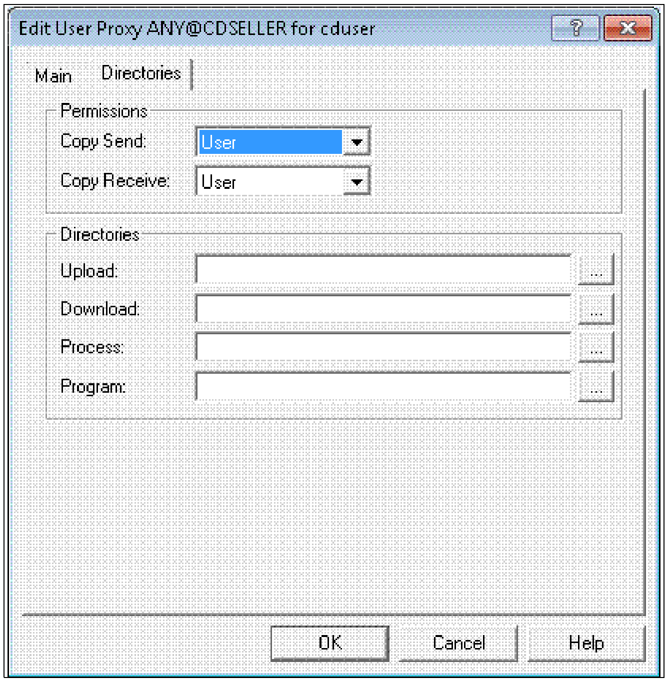


Figure 6-23 Edit User Proxies - Directories

6.4.4 Installing and configuring the IBM Sterling Connect:Direct File Agent

Use the steps in this section to install and configure IBM Sterling Connect:Direct File Agent.

Installation procedure

Follow these steps:

1. Run the FileAgent.exe from IBM Sterling Connect:Direct product directory:

   ```
   C:\SterlingConnect Direct V5.1.0\Direct V 4.6.00 for Microsoft
   Windows\CDWin4600\Server and Requester\FileAgentInstall.exe.
   ```

2. The installation asks for the directory where to install File Agent.

Configuring IBM Sterling Connect:Direct File Agent

In this specific scenario, IBM Sterling Connect:Direct File Agent is used to monitor the directory where Buyer sends a copy of the XML file, which is an acknowledgement of the payment received by the Bank. File Agent then submits a process to the Bank IBM Sterling Connect:DIrect to forward the XML file to the Seller.

Configuration steps

Follow these steps:

1. Open the IBM Sterling Connect:Direct File Agent graphical user interface (Figure 6-24).
2. Select the **File agent** tab and **Default_config**.
3. Click **Edit**.
4. Type the following information:
   a. The Userid for API. It is the IBM Sterling Connect:Direct user that the File Agent uses to logon to the Bank IBM Sterling Connect:Direct. In this scenario, it should be “cduser.”
   b. The Password for API. It is the cduser Windows password.

5. API host DNS name: It is the Bank IBM Sterling Connect:Direct IP: 9.42.170.223
6. API port: It is the Bank IBM Sterling Connect:Direct Bank port-number: 1364
7. Watched directories: It is the directory where Buyer Sterling B2B Integrator Connect:Direct Server adapter puts acknowledgement XML: `c:\CDBANKCOMM\CDBUYERIN\`.

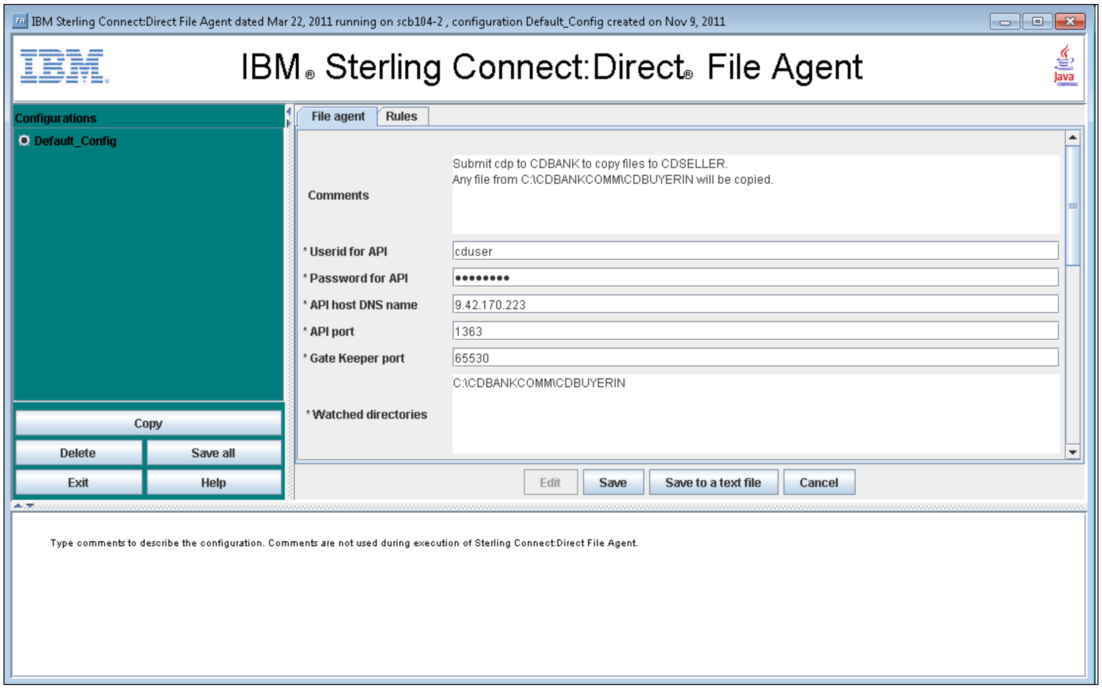


Figure 6-24 Watched directories

8. Create new "Submit process rules," as shown in Figure 6-25 and Figure 6-26.

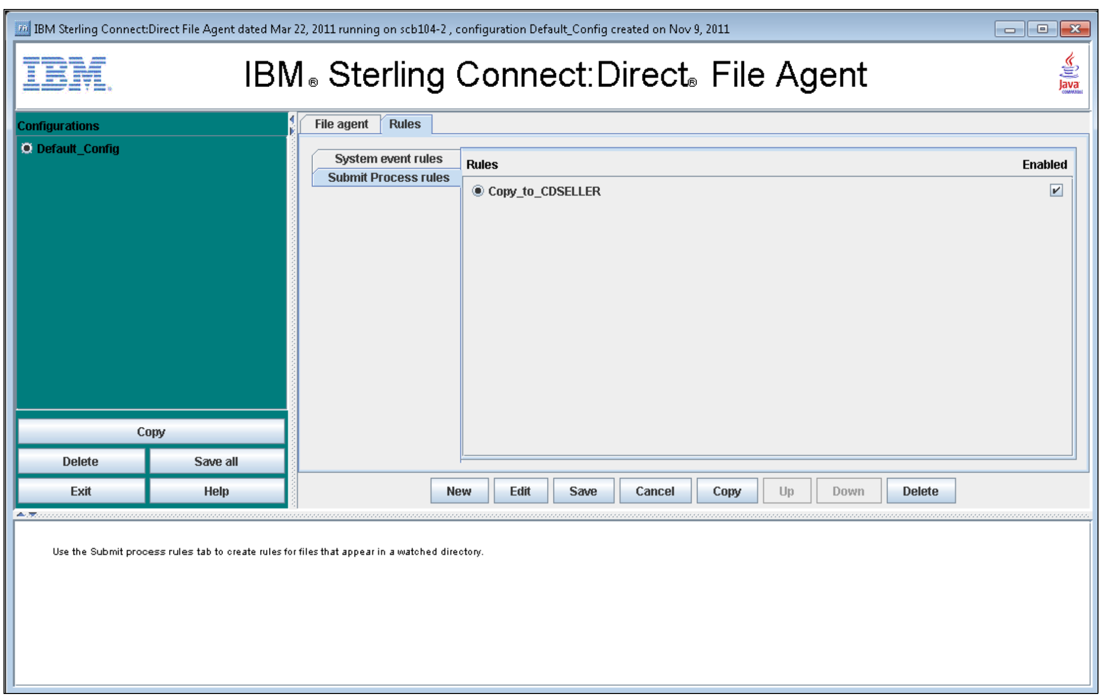


Figure 6-25 Submit Process Rule

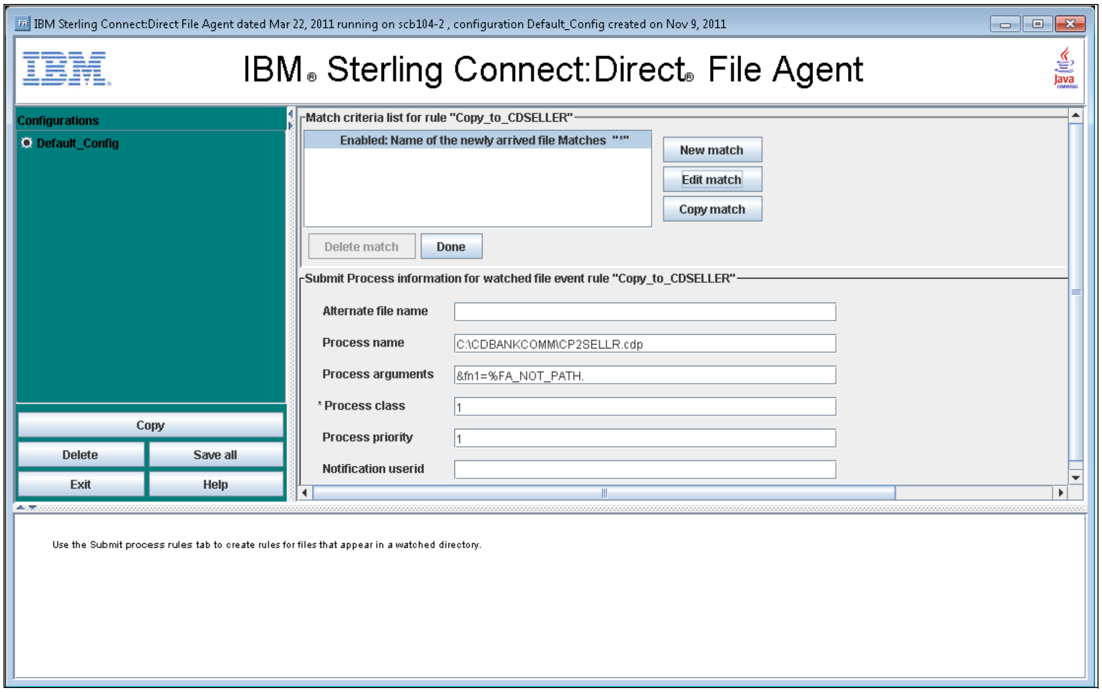


Figure 6-26 Match criteria

9. In this specific scenario, any file arriving at the watched directory whose name ends with .XML must submit a process, as shown in Figure 6-27.

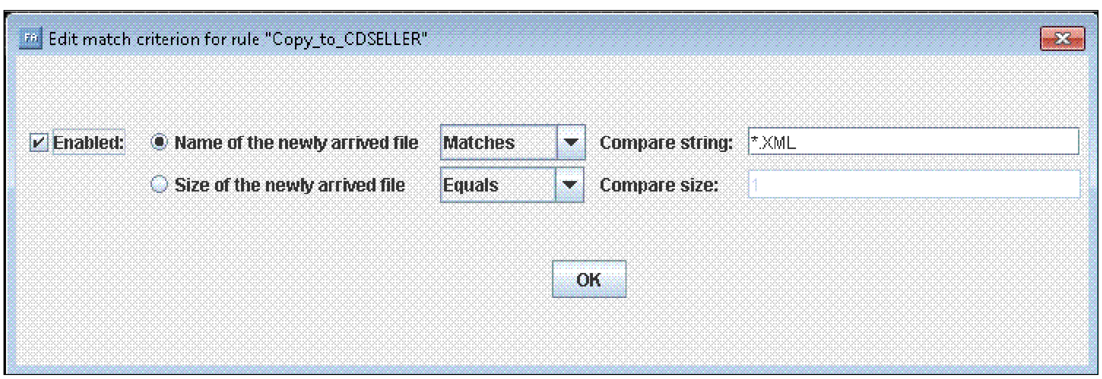


Figure 6-27 Edit match criterion

10. Example 6-5 shows a sample process to be submitted by File Agent:

    The following process refers to “On the Submit Process Rules” window after selecting the “Copy_to_CDSELLER” rule to be edited. Then the Submit Process information for watched file event rule Copy_to_CDSELLER window is shown.

    The argument ‘Process name C:\CDBANKCOMM\CP2SELLR.cdp’ contains the location and name of the process to be submitted and the argument ‘Process arguments &fn1=%FA_NOT_PATH.’ contains the variable to be passed to the process. In this example, the process variable &fn1 is replaced by the name of the file to be copied. The process deletes the file copied from the origin directory if the copy operation succeeds.

Example 6-5 Sample process

```
/*BEGIN_REQUESTER_COMMENTS
    $PNODE$="CDBANK" $PNODE_OS$="Windows"
    $SNODE$="CDSELLER" $SNODE_OS$="Windows"
    $OPTIONS$="WDOS"
  END_REQUESTER_COMMENTS*/

CP2SELLR PROCESS
   &fn1=default
   &fn2=default
   SNODE=CDSELLER

CPBS1 COPY
   FROM (FILE="C:\CDBANKCOMM\CDBUYERIN\&fn1")
   TO (  FILE="C:\CDSELLERCOMM\CDBANKIN\&fn1" DISP=RPL)
IF (CPBS1 EQ 0) THEN
   RUN TASK PNODE (PGM=Windows)
      SYSOPTS="cmd(DEL) ARGS(C:\CDBANKCOMM\CDBUYERIN\&fn1)"
EIF
PEND
```

For more details on how to configure Sterling File Agent, see the *IBM Sterling Connect:Direct File Agent Configuration Guide* at the following website:

http://www.sterlingcommerce.com/documentation/home/MFT/ConnectDirect/File%20Agent/CD%20FileAgent14/CDFAConfigGuide.pdf

6.4.5 Installing and configuring the IBM Sterling Secure Proxy

This section describes the configuration of the IBM Sterling Secure Proxy at the Buyer company. We considered it as required to allow the Seller company to send the invoice to Buyer using myFileGatway in a secure manner through the DMZ. Figure 6-28 depicts this configuration.

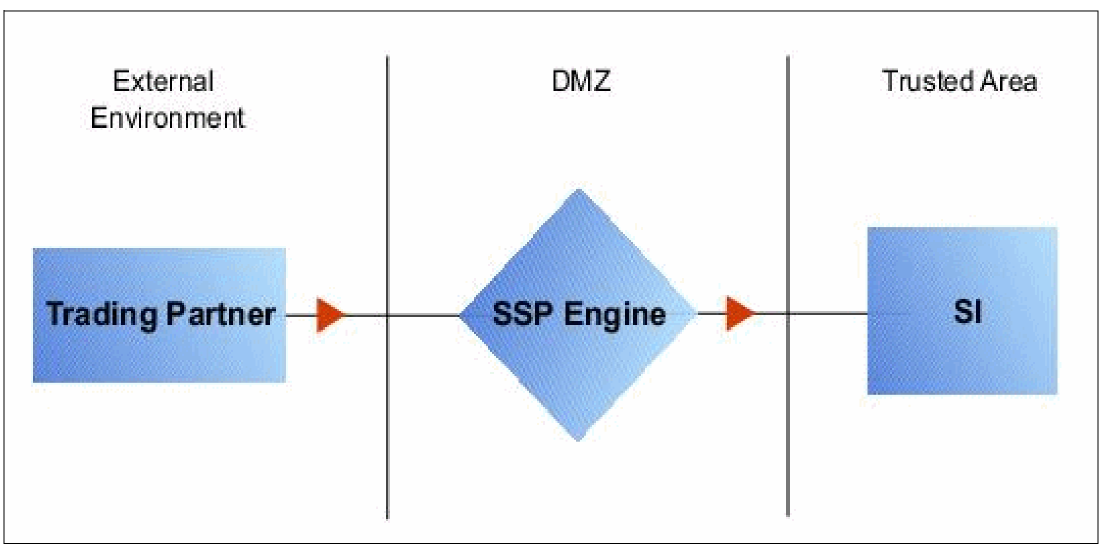


Figure 6-28 Sterling Secure Proxy at the Buyer company

In the following installation steps, we assume that you have already installed Sterling Secure Proxy 3.4.

> **Important:** Before installing the product on your environment, if you need more information, see the *Sterling Secure Proxy 3.4* documentation at this website:
>
> http://www.sterlingcommerce.com/documentation/home/MFT/SSP/SSP.html
>
> Pay special attention to the "System Requirements" section and the *Installation Guide*.

Configuring an IBM Sterling Secure Proxy

In this section we describe how to configure an IBM Sterling Secure Proxy.

Configuration steps

In order to configure IBM Sterling Secure Proxy to be ready to accept connections from the Seller through myFileGateway, we need to complete the following steps.

Create an Engine Definition

The engine lives in the DMZ and runs the proxy adapters that manage client communication requests to servers in your trusted zone. To perform this function, the engine receives configuration information from CM (Configuration Manager). Use CM to create an engine definition that contains configuration information for the engine.

Before you configure the engine, gather the information in Table 6-5 that you require to configure the engine. After you configure the engine, validate the configuration by ensuring that CM can view the engine.

Table 6-5 Engine configuration

CM field	Feature/value	Value
Engine Name	Name of the engine	
Engine Host	IP address of the engine	
Engine Listen Port	Port number of the engine	

Follow these steps to define an engine. See Figure 6-29.

Select **Configuration** from the menu bar.

1. Click **Actions** → **New Engine**.
2. Specify the following values:
   – Engine Name
   – Engine Host
   – Engine Listen Port
3. Click **Save**.

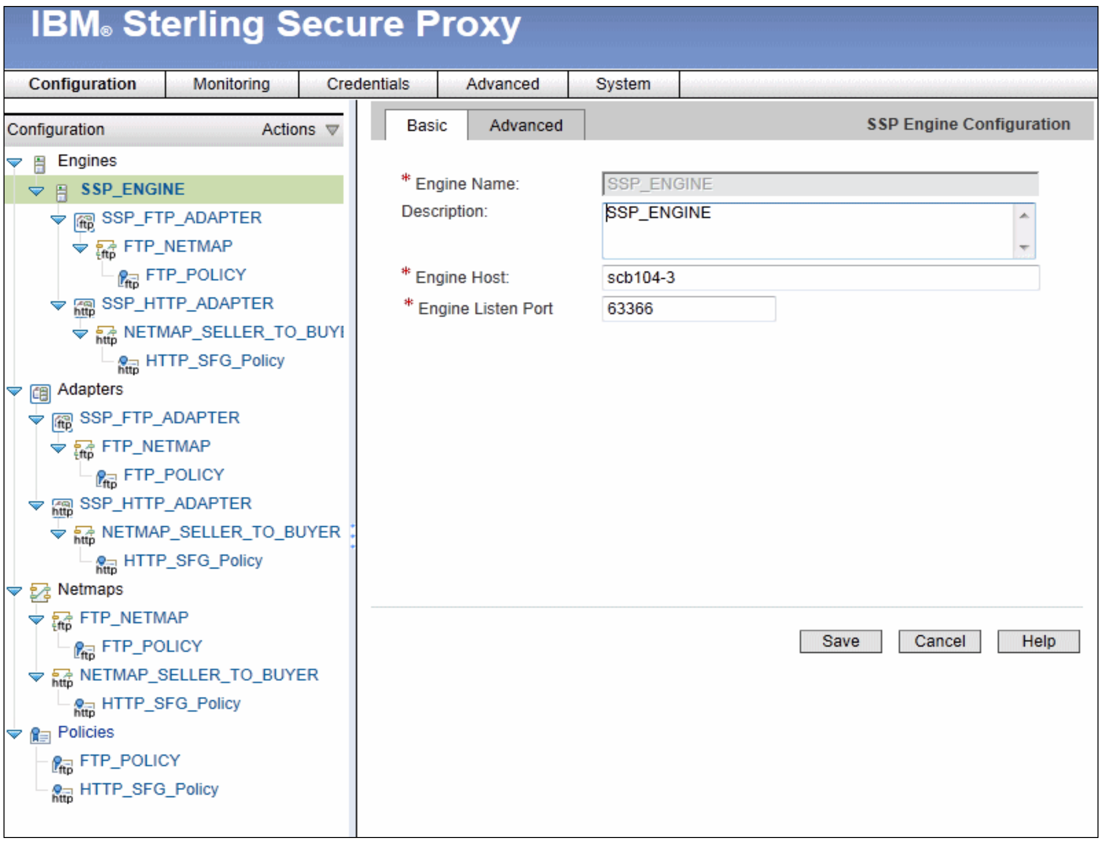


Figure 6-29 Engine definition

Create an HTTP Policy

The HTTP Policy defines how you impose controls to authenticate a trading partner (in our case BUYER company) trying to access the IBM Sterling File Gateway over the public Internet.

Follow these steps to define an HTTP Policy. See Figure 6-30.

1. Click **Configuration** from the menu bar.
2. Click **Actions** → **New Policy** → **HTTP Policy**.
3. Type a Policy Name.
4. Click **Save**.

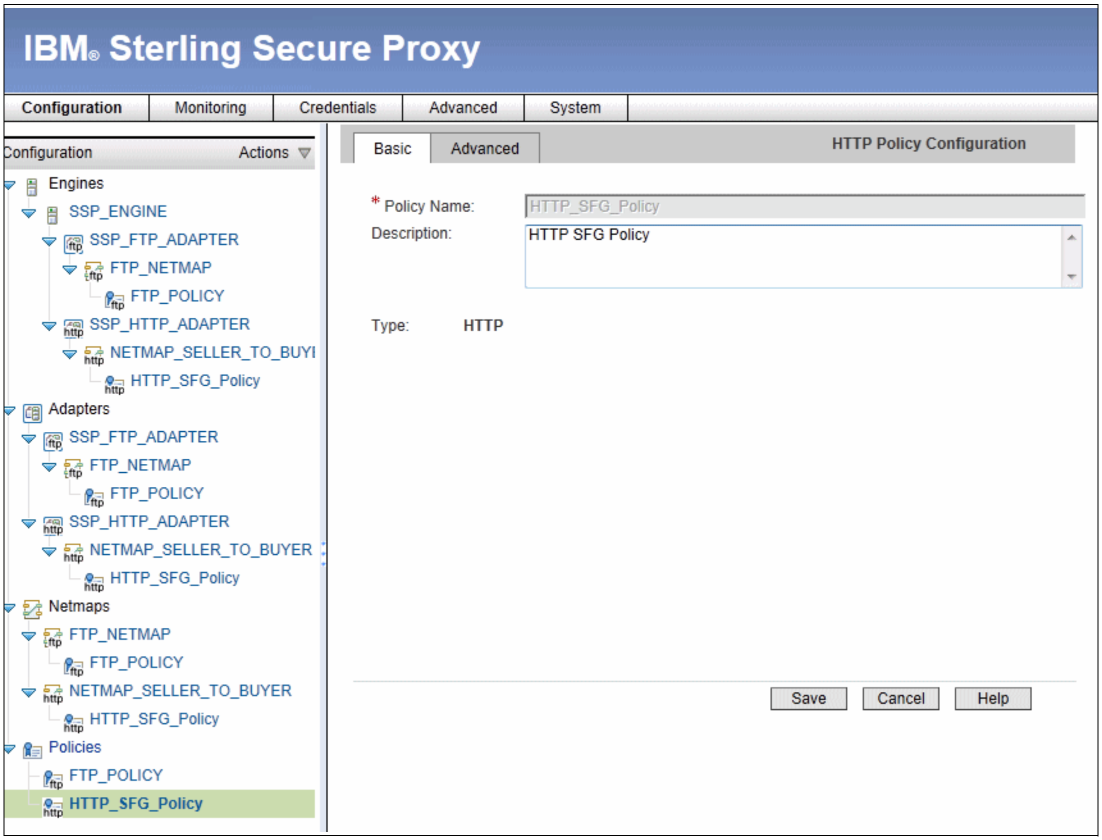


Figure 6-30 Policy Configuration

Create an HTTP Netmap

You define inbound connection information for your external trading partners and outbound connection information for the IBM Sterling File Gateway that Sterling Secure Proxy connects to. These values are stored in a netmap. The netmap is associated with a policy and an adapter.

Before you begin this procedure, be sure that you have created a policy to associate with the netmap.

Follow these steps to create a netmap and define inbound and outbound nodes.

1. Click **Configuration** from the menu bar.
2. Click **Actions** → **New Netmap** → **HTTP Netmap**.
3. Type a Netmap Name.
4. To define an inbound node definition, click the **Inbound Nodes** tab and click **New**. See Figure 6-31.
5. Specify the following values:
   - Inbound Node Name
   - Peer Address Pattern
   - Policy

   Tip: If you have not defined a policy, click the green plus sign to define one.
6. Click **OK**.

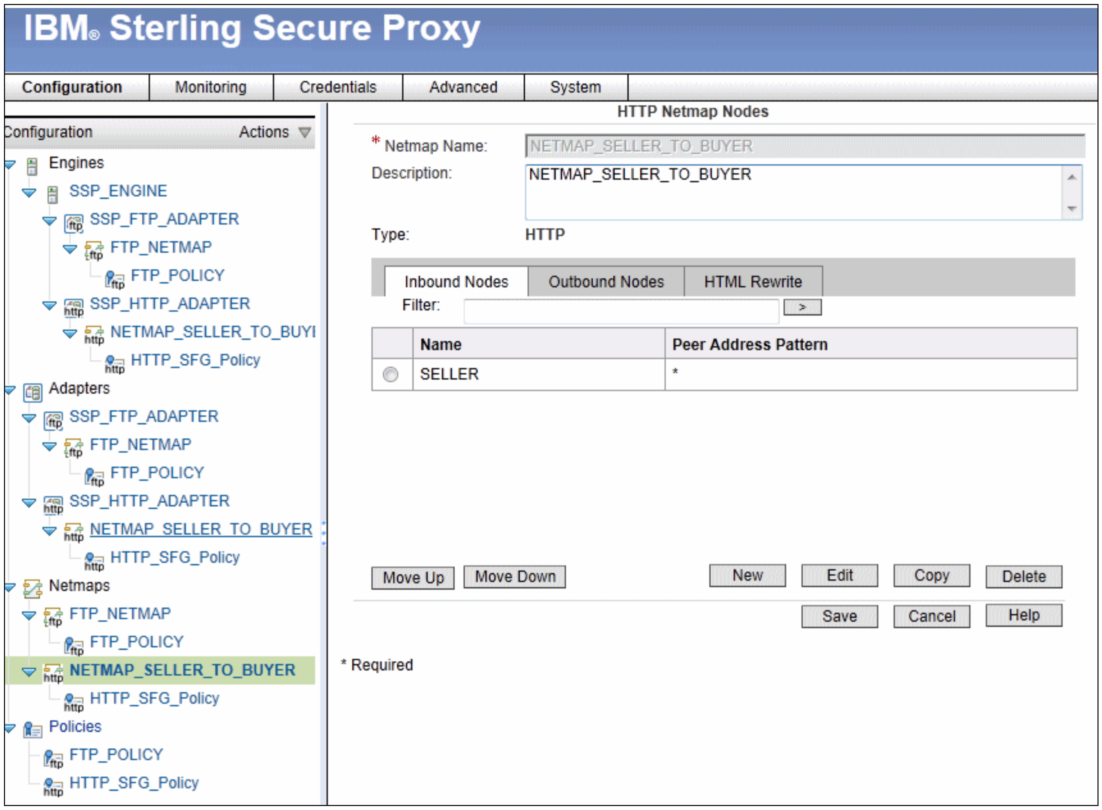


Figure 6-31 HTTP Netmap Inbound Node

7. To define an outbound node definition, click the **Outbound Nodes** tab and click **New.** See Figure 6-32.
8. Specify the following values:
   – Outbound Node Name
   – Primary Destination Address
   – Primary Destination Port
9. Click **OK**.
10. Click **Save**.

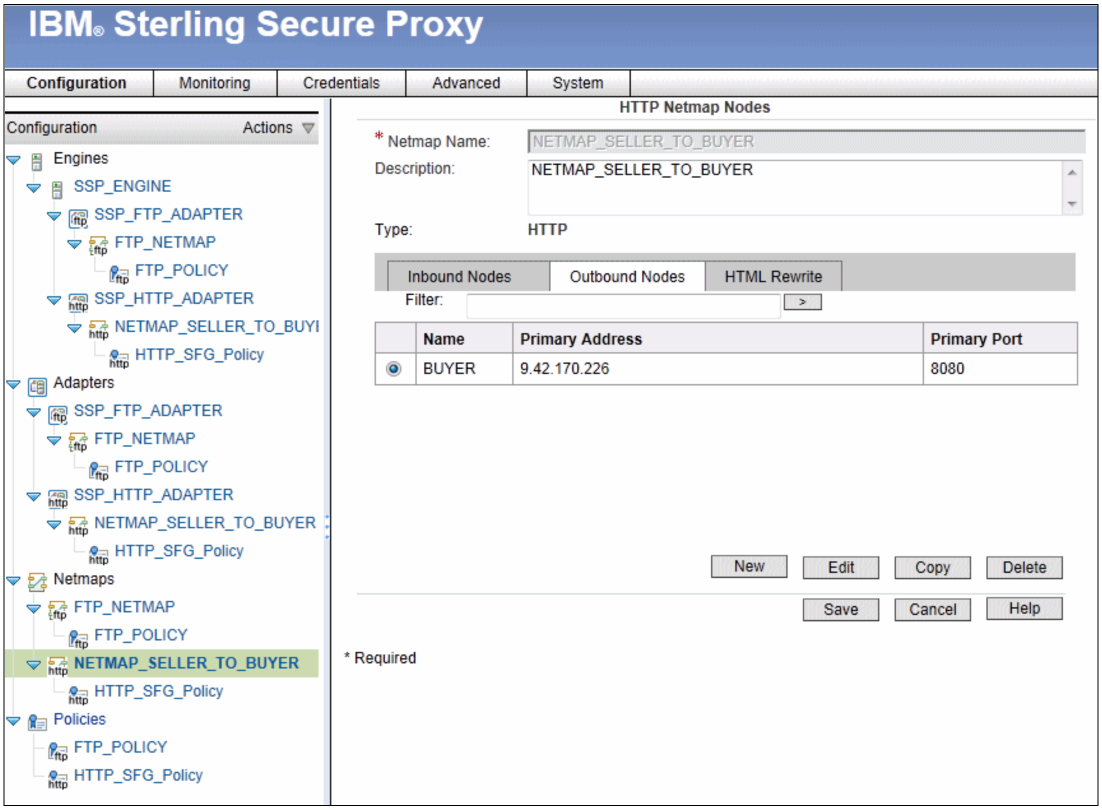


Figure 6-32 HTTP Netmap Outbound Node

Define the HTTP adapter used for the connection

An HTTP adapter definition specifies system-level communications information necessary for HTTP connections to and from Sterling Secure Proxy. You can create multiple adapter definitions depending on your need.

Before you begin this procedure, be sure that you have created the following definitions:

- A netmap to associate with the adapter.
- An engine definition to associate with the adapter. See the following website to Install or Upgrade Sterling Secure Proxy on UNIX or Linux or Install or Upgrade Sterling Secure Proxy on Windows for instructions:

  `http://www.sterlingcommerce.com/documentation/home/MFT/SSP/SSP.html`

Use the following steps to define an HTTP adapter. See Figure 6-33.

1. Click **Configuration** from the menu bar.
2. Click **Actions** → **New Adapter** → **HTTP Reverse Proxy**.
3. Specify values for the following items:
   – Adapter Name
   – Listen Port
   – Netmap
   – Standard Routing Node
   – Engine
4. Click **Save**.

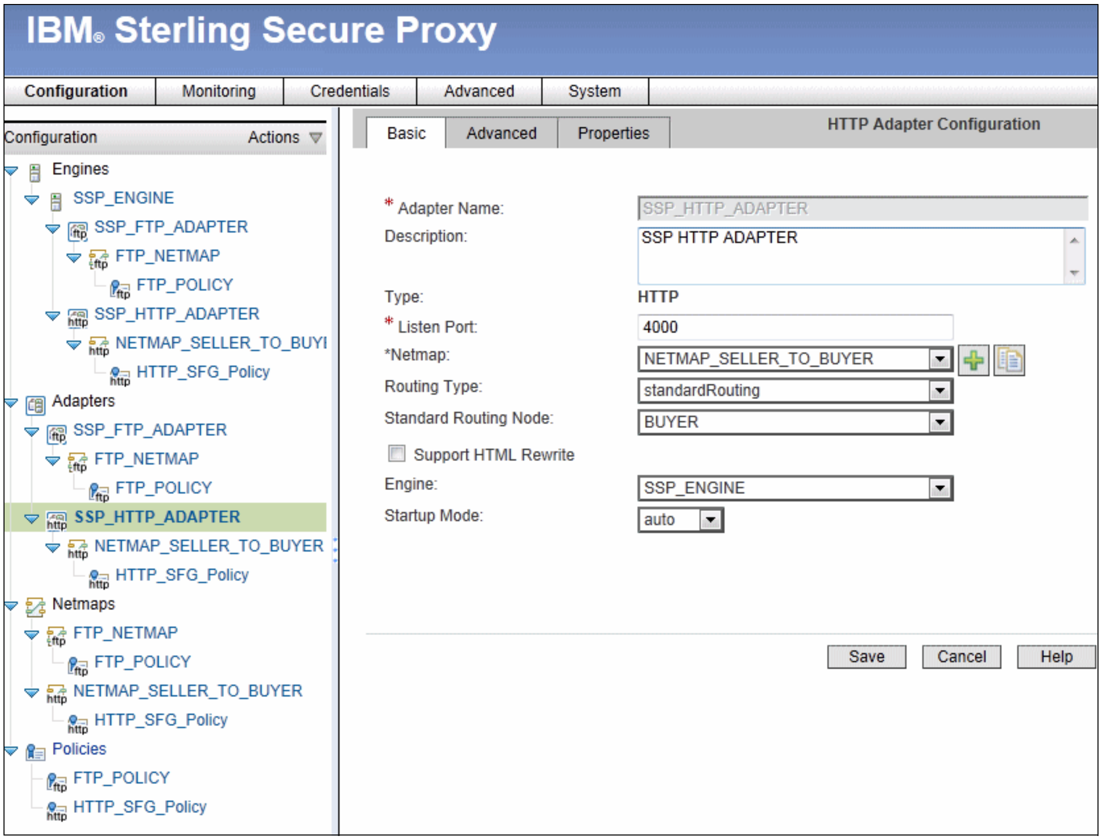


Figure 6-33 HTTP Adapter Configuration

6.4.6 Installing and configuring the IBM Sterling Control Center

This section describes the installation and the configuration of IBM Sterling Control Center at the Buyer company. It is required to allow the Buyer company to manage IBM Sterling File Gateway connections with Seller and also to manage IBM Sterling Connect:Direct connections with the Bank.

> **Tip:** In this section, we are presenting a summary of the installation steps performed to install IBM Sterling Control Center version 5.3 for this specific scenario. However, if you want to replicate the installation in your own environment, make sure to see the *IBM Sterling Control Center Getting Started Guide*, as there are important instructions to follow:
>
> `http://www.sterlingcommerce.com/documentation/SCC53/PDF/GettingStarted.pdf`

Installation procedure

Follow these steps to install IBM Sterling Control Center.

1. IBM Sterling Control Center needs a database to hold its configuration, the alerts that it generates, and the logs collected from managed IBM Sterling B2B Integrator and IBM Sterling Connect:Direct. Several databases are supported. The database must be installed and schemas defined prior to IBM Sterling Control Center installation.
2. Run the **`CCinstall.exe`** from IBM Sterling Control Center product directory:
   `C:\IBM Sterling Control Center\IBM Sterling Control Center V 5.3.00 for Windows, English (CZZ2KEN)\CZZ2KEN\Windows.`

See the IBM Sterling Control Center Getting Started Guide for an explanation of the required installation parameters:

http://www.sterlingcommerce.com/documentation/SCC53/PDF/GettingStarted.pdf

Configuring IBM Sterling Control Center

In this specific scenario, IBM Sterling Control Center is used to monitor the sending of the payment message from Buyer to Bank and the XML acknowledgement from Bank to Seller.

Configuration steps

Follow these steps:

1. Open the IBM Sterling Control Center graphical user interface. See Figure 6-34.
2. Select **Manage** → **Add Server** to define Bank and Seller IBM Sterling Connect:Direct.

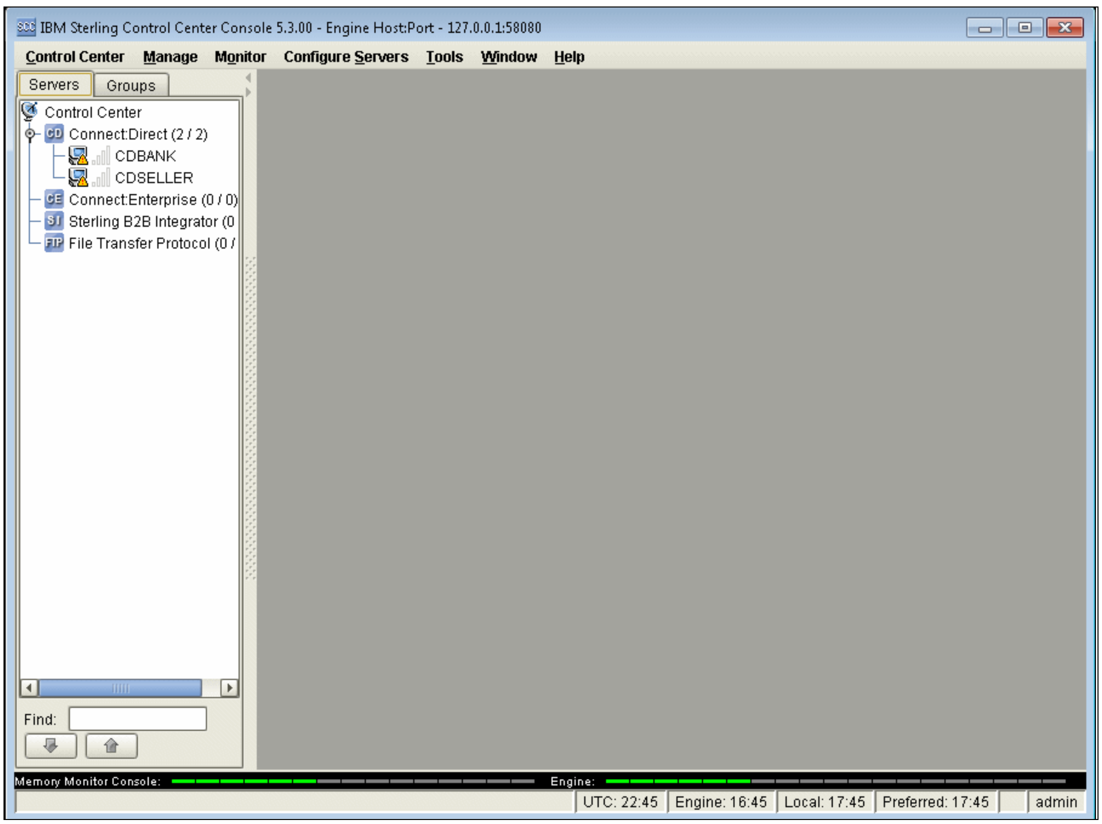


Figure 6-34 Sterling Control Center configuration

Figure 6-35 shows the server properties.

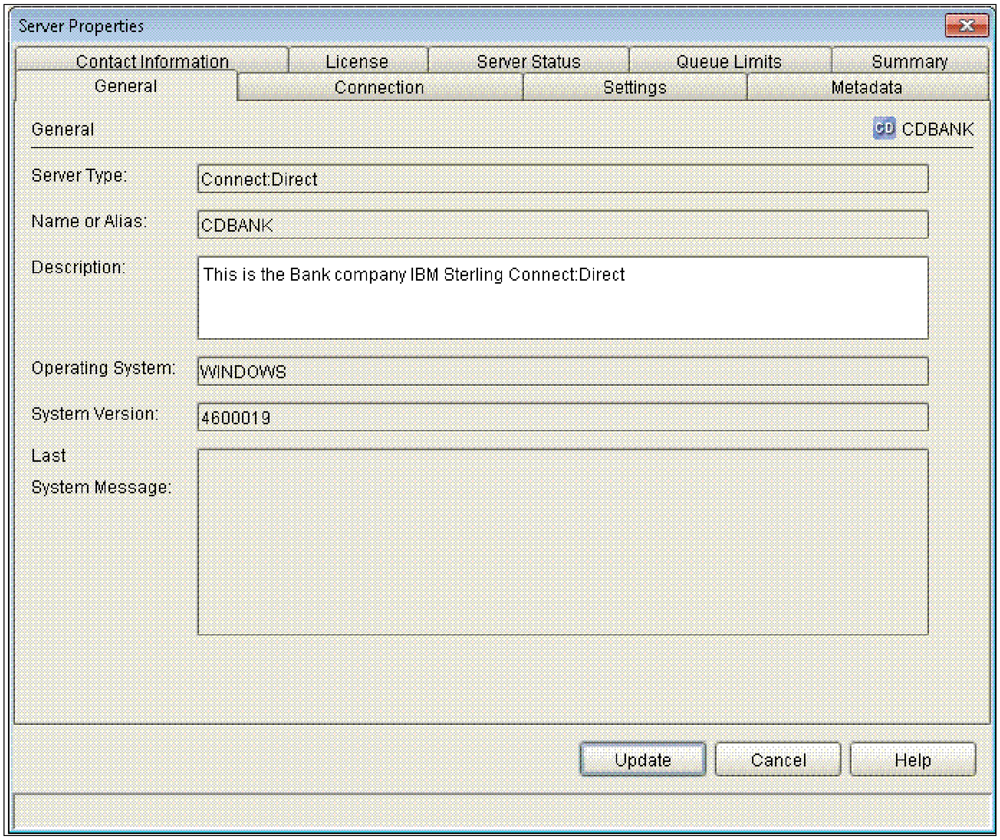


Figure 6-35 Server Properties

3. Select **Manage** → **Rules and Actions** → **Rules** to define the rules to monitor Bank and Seller transmissions. See Figure 6-36.

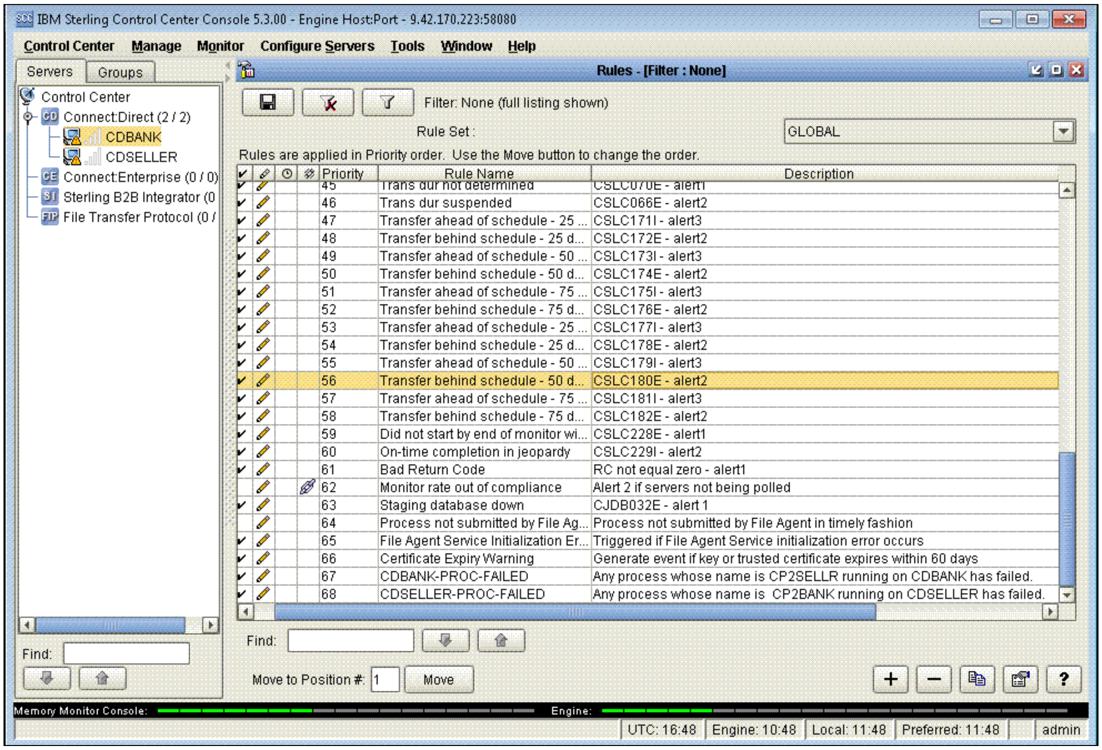


Figure 6-36 Rules to monitor Bank and Seller transmissions

Figure 6-37 shows the rule properties.

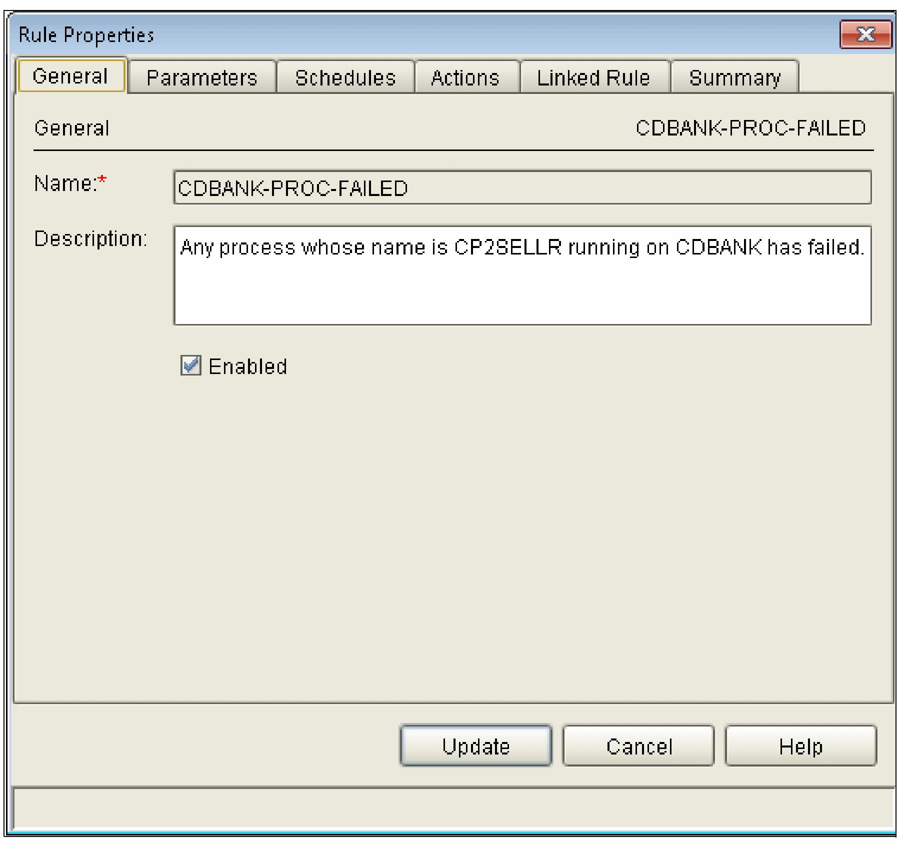


Figure 6-37 Rule Properties

4. Select **Monitor** → **Alerts** to see new alerts generated by IBM Sterling Control Center. See Figure 6-38.

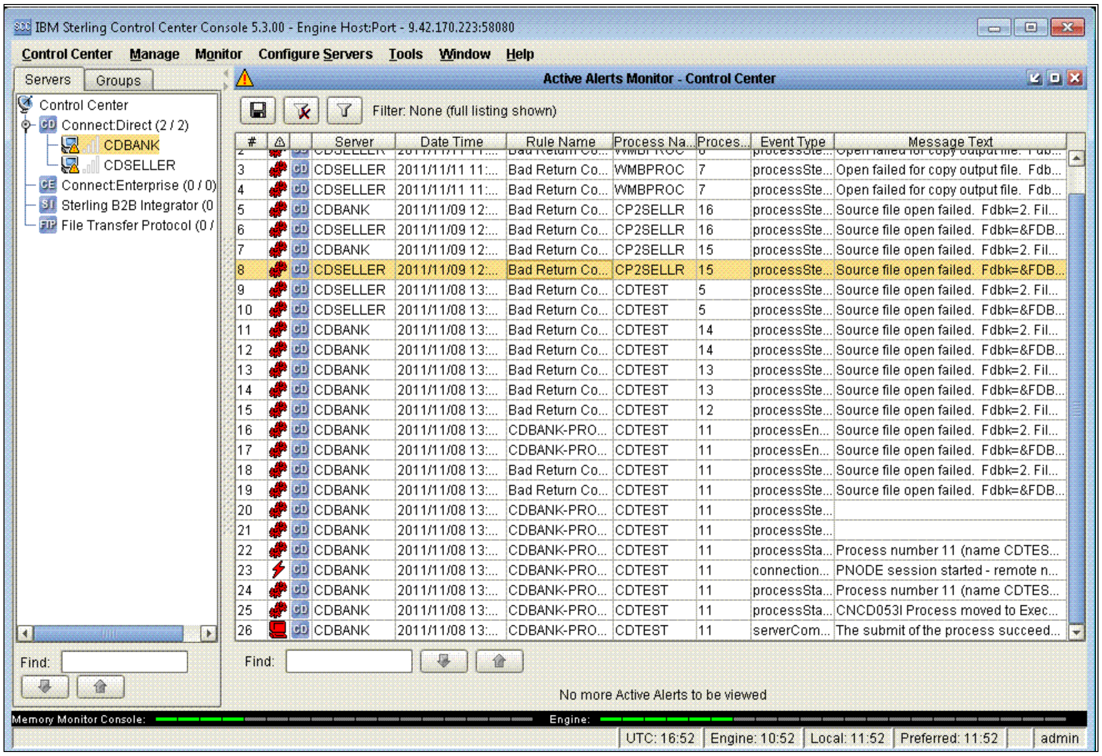
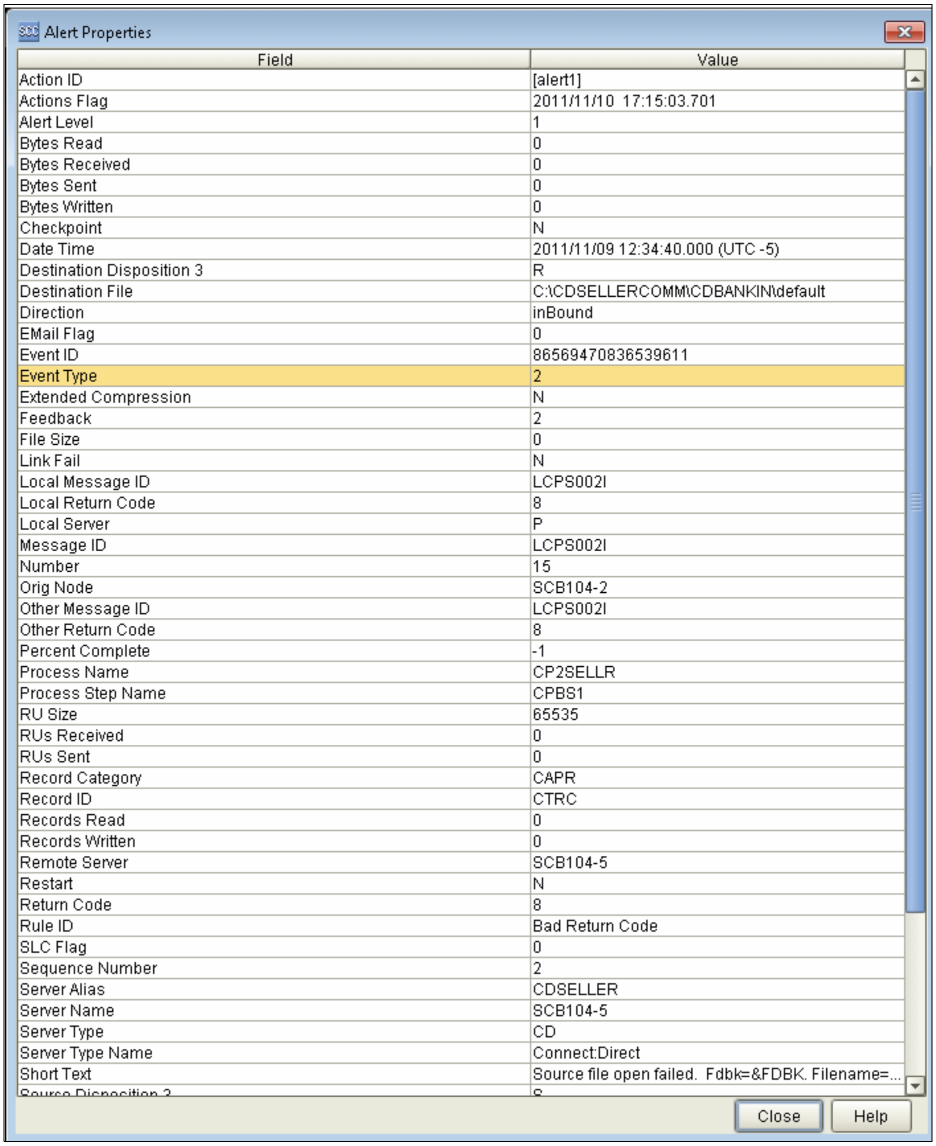


Figure 6-38 Active Alert Monitor

Figure 6-39 shows the alert properties.



Alert Properties

Field	Value
Action ID	[alert1]
Actions Flag	2011/11/10 17:15:03.701
Alert Level	1
Bytes Read	0
Bytes Received	0
Bytes Sent	0
Bytes Written	0
Checkpoint	N
Date Time	2011/11/09 12:34:40.000 (UTC -5)
Destination Disposition 3	R
Destination File	C:\CDSELLERCOMM\CDBANKIN\default
Direction	inBound
EMail Flag	0
Event ID	86569470836539611
Event Type	2
Extended Compression	N
Feedback	2
File Size	0
Link Fail	N
Local Message ID	LCPS002I
Local Return Code	8
Local Server	P
Message ID	LCPS002I
Number	15
Orig Node	SCB104-2
Other Message ID	LCPS002I
Other Return Code	8
Percent Complete	-1
Process Name	CP2SELLR
Process Step Name	CPBS1
RU Size	65535
RUs Received	0
RUs Sent	0
Record Category	CAPR
Record ID	CTRC
Records Read	0
Records Written	0
Remote Server	SCB104-5
Restart	N
Return Code	8
Rule ID	Bad Return Code
SLC Flag	0
Sequence Number	2
Server Alias	CDSELLER
Server Name	SCB104-5
Server Type	CD
Server Type Name	Connect:Direct
Short Text	Source file open failed. Fdbk=&FDBK. Filename=...
Source Disposition 2	S

Close Help

Figure 6-39 Alert Properties

5. Select **Monitor** → **Handled Alerts** to see alerts previously checked by someone.
6. Select **Queued Process Activity** to see IBM Sterling Connect:Direct processes on the queue, These are processes running or waiting to run (Figure 6-40).

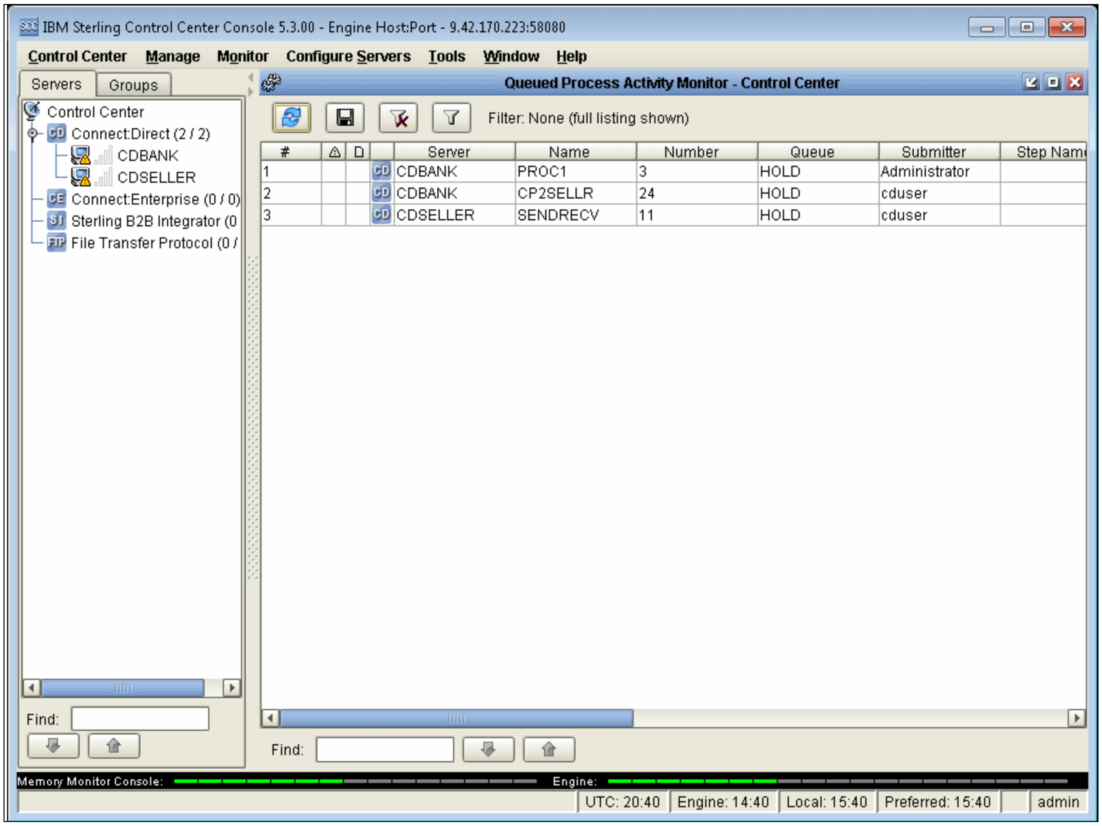


Figure 6-40 Queued Process Activity Monitor

7. Select Completed Process Activity to see IBM Sterling Connect:Direct process statistics (Figure 6-41).

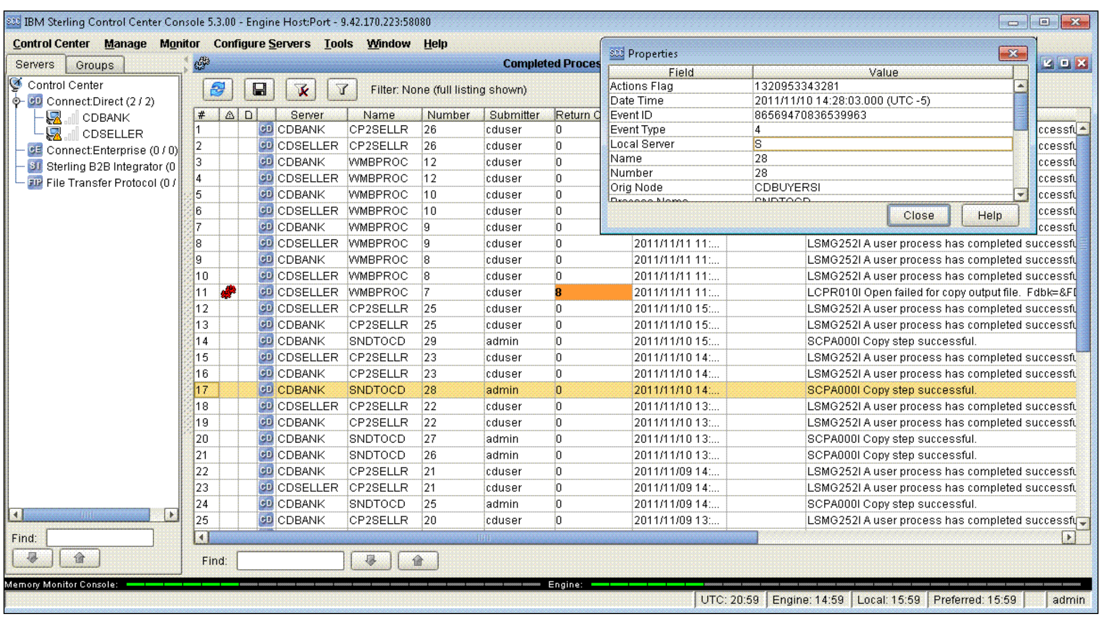


Figure 6-41 Completed Process Activity

Also see the *IBM Sterling Control Center Configuration Management Guide* at this website:

http://www.sterlingcommerce.com/documentation/SCC53/PDF/ConfigMgmt.pdf

Configuring Sterling Control Center for Sterling B2B Integrator

Although not used in this specific scenario, it is also possible to use Sterling Control Center to monitor a Sterling B2B Integrator instance. This section presents the configuration steps that you need to make if you want to test this additional feature.

Configuration steps for IBM Sterling Control Center

Follow these steps:

1. Open the IBM Sterling Control Center graphical user interface. See Figure 6-42.
2. Select **Manage** → **Add Server** to define Buyer IBM Sterling B2B Integrator.

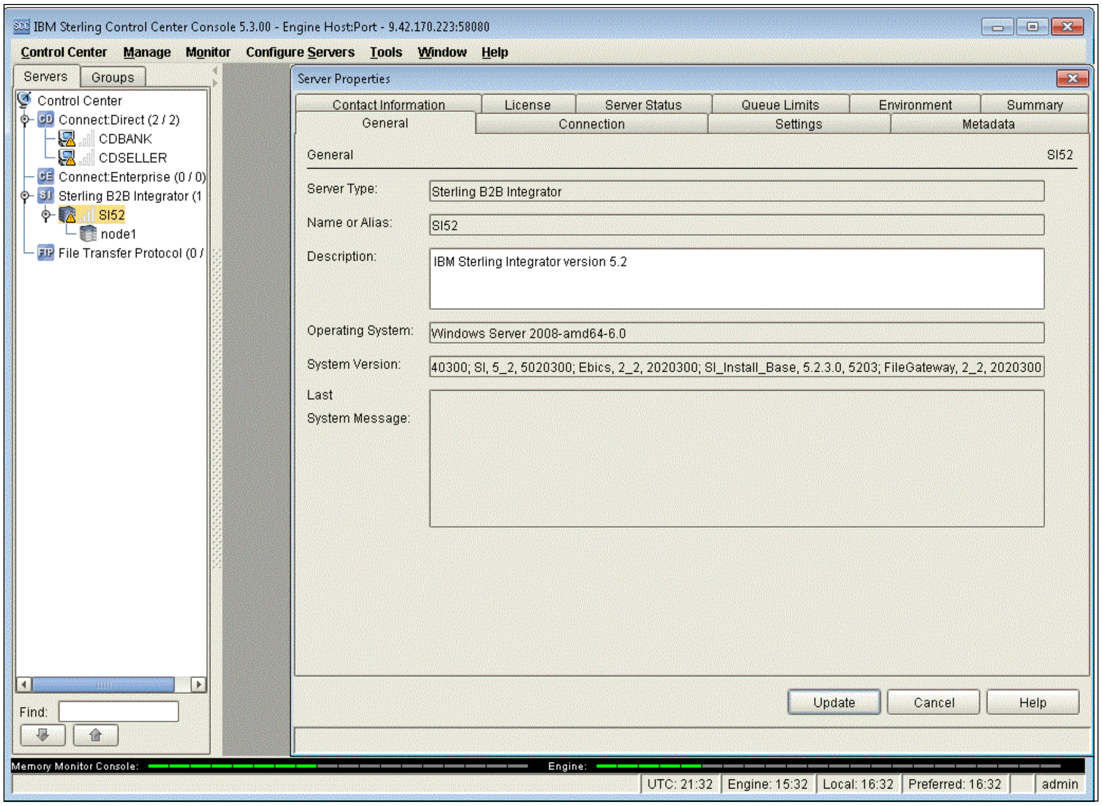


Figure 6-42 Adding Sterling B2B Integrator to Sterling Control Center

Figure 6-43 shows the Connection window.

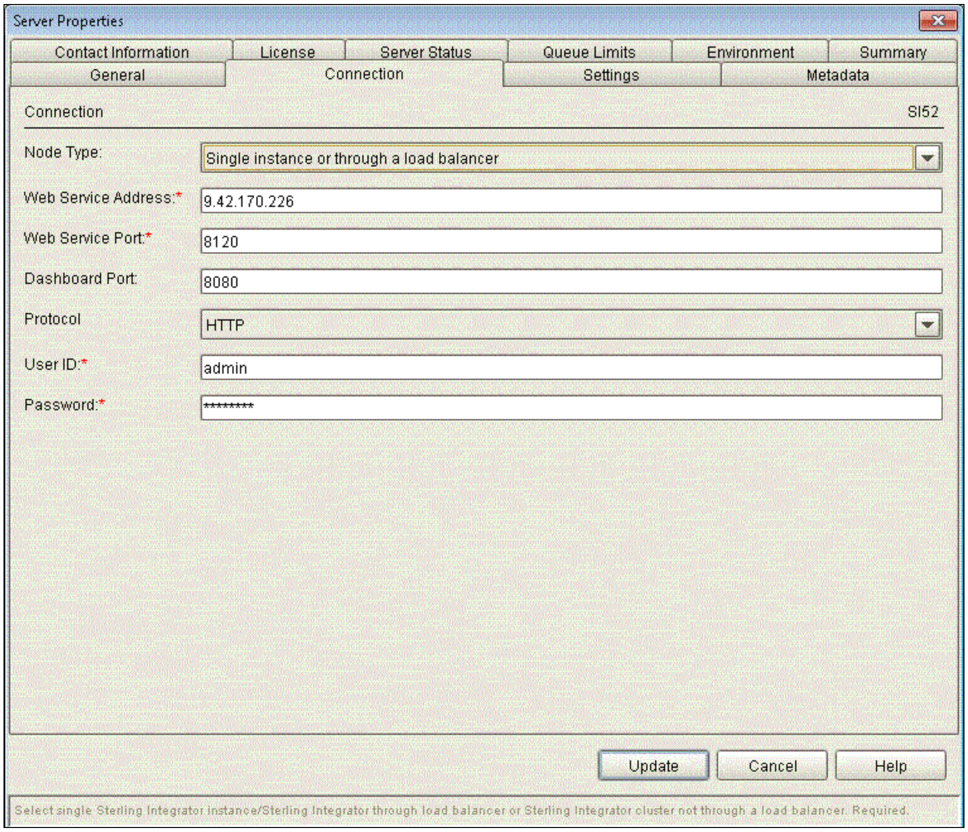


Figure 6-43 Connection

The “Web Service Address” and “Web Service Port” parameters that you need here are defined in the section named “Configurations steps for IBM Sterling B2B Integrator” on page 228.

Figure 6-44 shows the Select Protocols window.

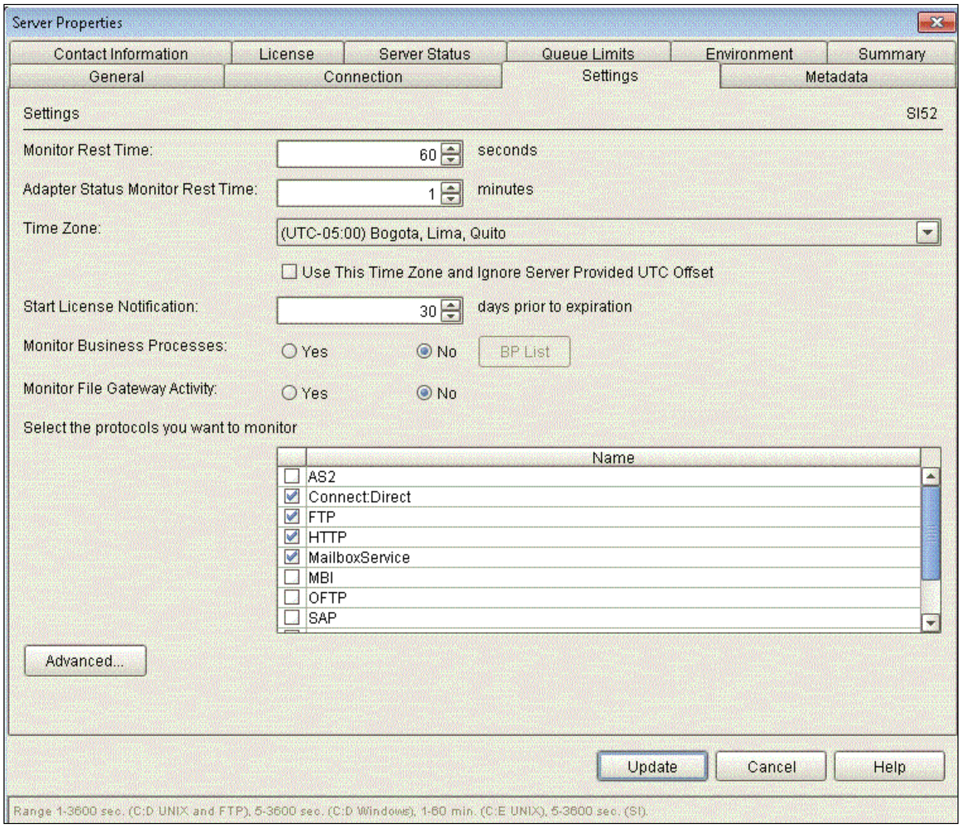


Figure 6-44 Select protocols

Tip: You can find detailed information about how to configure IBM Sterling Control Center at the following website:

http://www.sterlingcommerce.com/documentation/SCC53/PDF/GettingStarted.pdf

3. Select **Manage** → **Rules and Actions** → **Rules** to define the rules to monitor Buyer IBM Sterling B2B Integrator. In this example we show a rule to generate an alert whenever an IBM Sterling B2B Integrator adapter is disabled. See Figure 6-45.

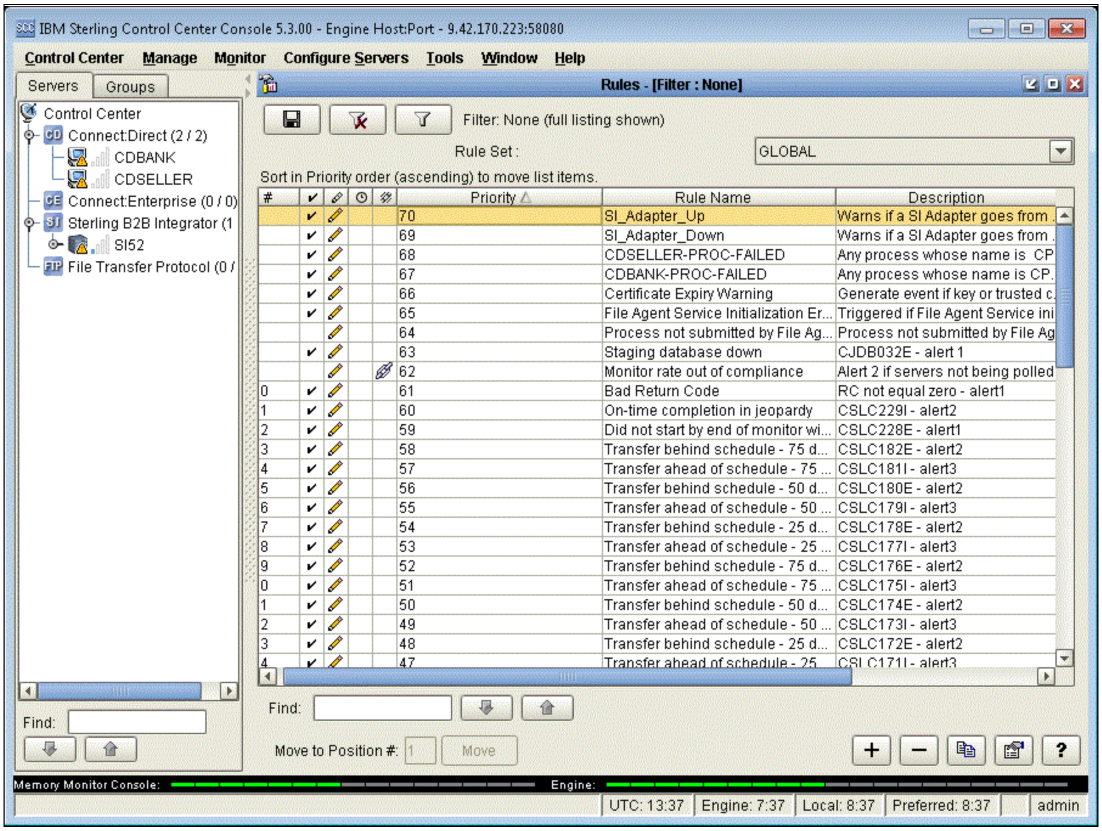


Figure 6-45 Rule list

Figure 6-46 shows the Rule Properties window.

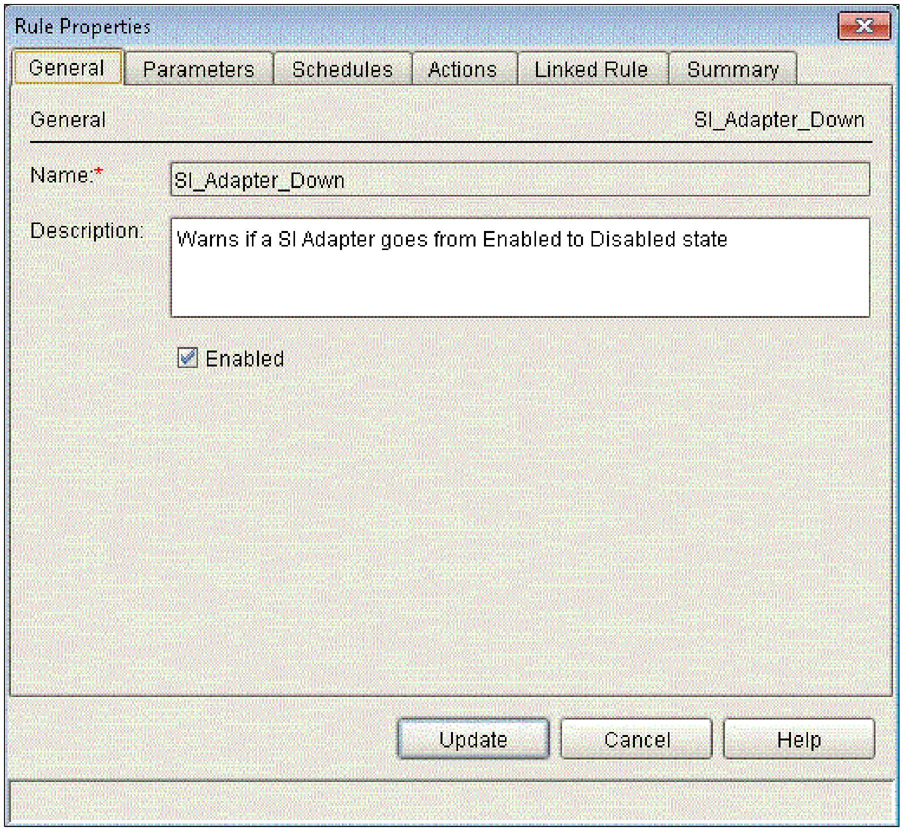


Figure 6-46 Rule Properties: General

Figure 6-47 shows the parameters of the rule properties.

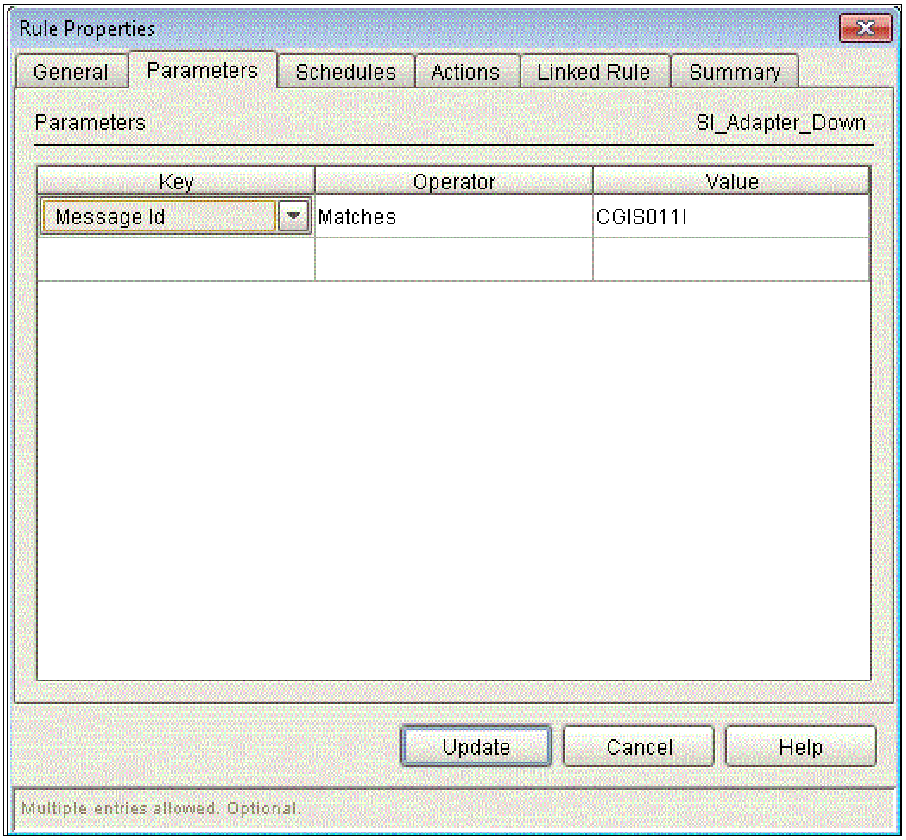


Figure 6-47 Rule Properties: Parameters

4. Select **Monitor** → **Active Alerts** to see the alerts that have been generated by IBM Sterling Control Center which have not yet being processed by any operator.

   Figure 6-48 and Figure 6-49 show the list of Active Alerts at some moment. If you click a specific alert, another window is opened showing its detailed information.

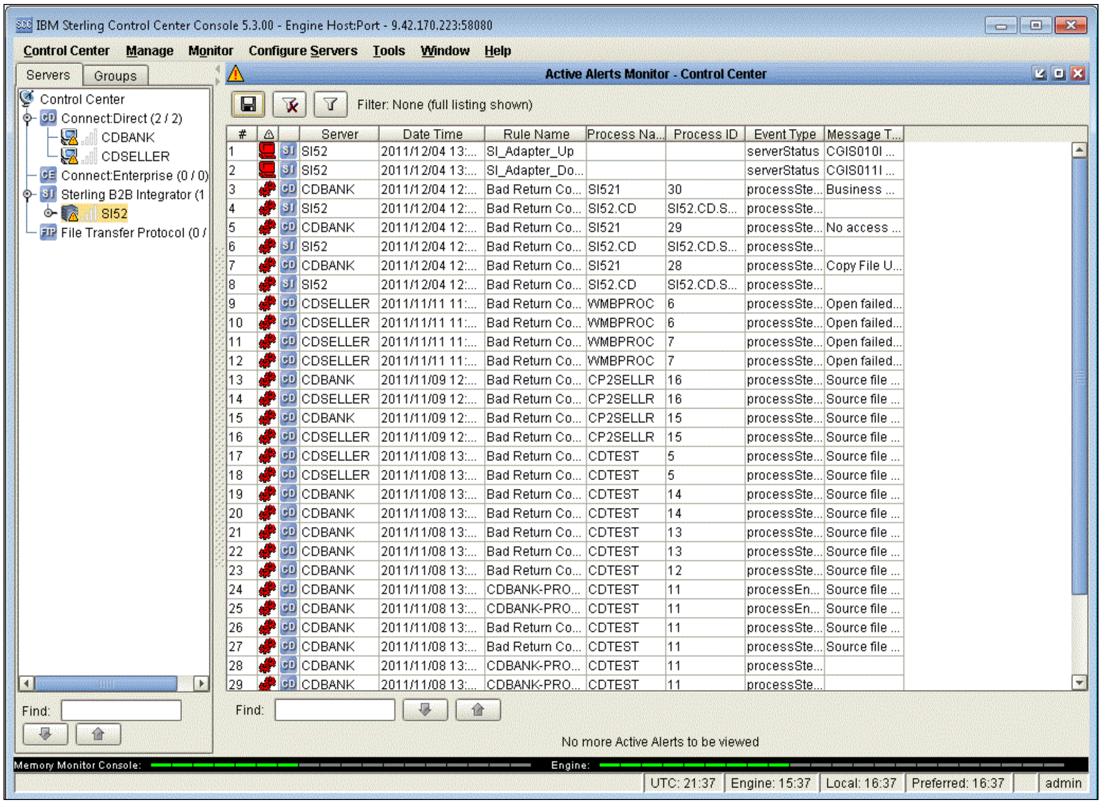


Figure 6-48 Active Alerts Monitor

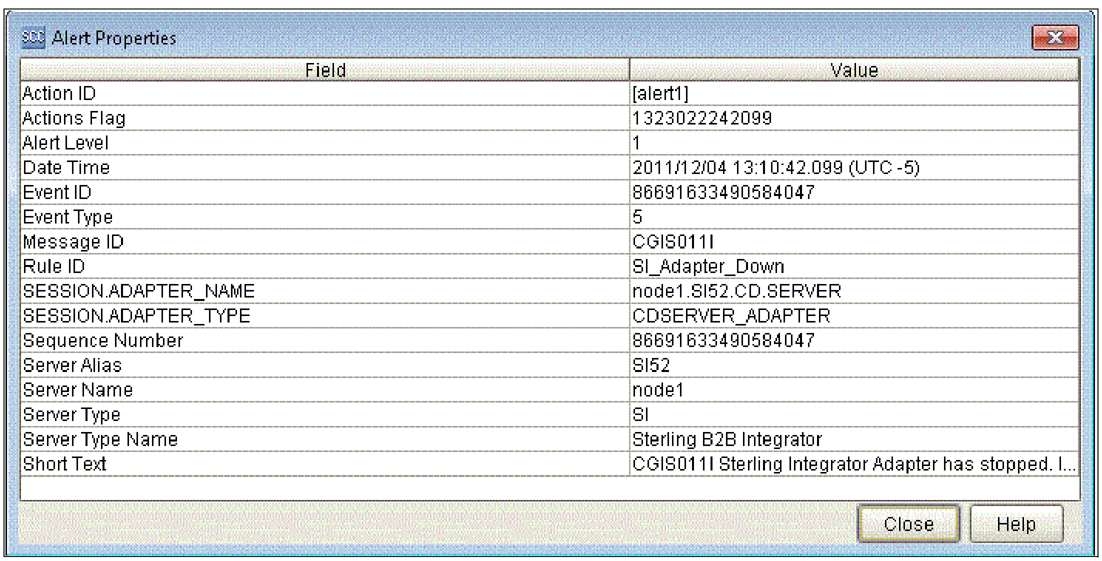


Figure 6-49 Alert Properties

Tip: Detailed information about how to create IBM Sterling Control Center Rules is available on the *IBM Sterling Control Center - How-To Guide* at this website:

`http://www.sterlingcommerce.com/documentation/SCC53/PDF/HowTo.pdf`

Configurations steps for IBM Sterling B2B Integrator

The detailed description of the steps necessary to configure Web Services to allow IBM Sterling Control Center to manage IBM Sterling B2B Integrator can be found in this manual (see the following website):

IBM Sterling B2B Integrator - Monitoring with Sterling Control Center in the paragraph, "Enable Sterling Integrator for Monitoring by Sterling Control Center."

http://help.sterlingcommerce.com/SB2BI52/topic/com.ic.customization.doc/SI_52_PDF/SI52_SCC.pdf

Additional information about Web Services can be found in this manual (see the following website):

IBM Sterling B2B Integrator - Web Services

http://help.sterlingcommerce.com/SB2BI52/topic/com.ic.customization.doc/SI_52_PDF/SI52_WebServices.pdf

Figure 6-50 shows the parameters used on this specific scenario.

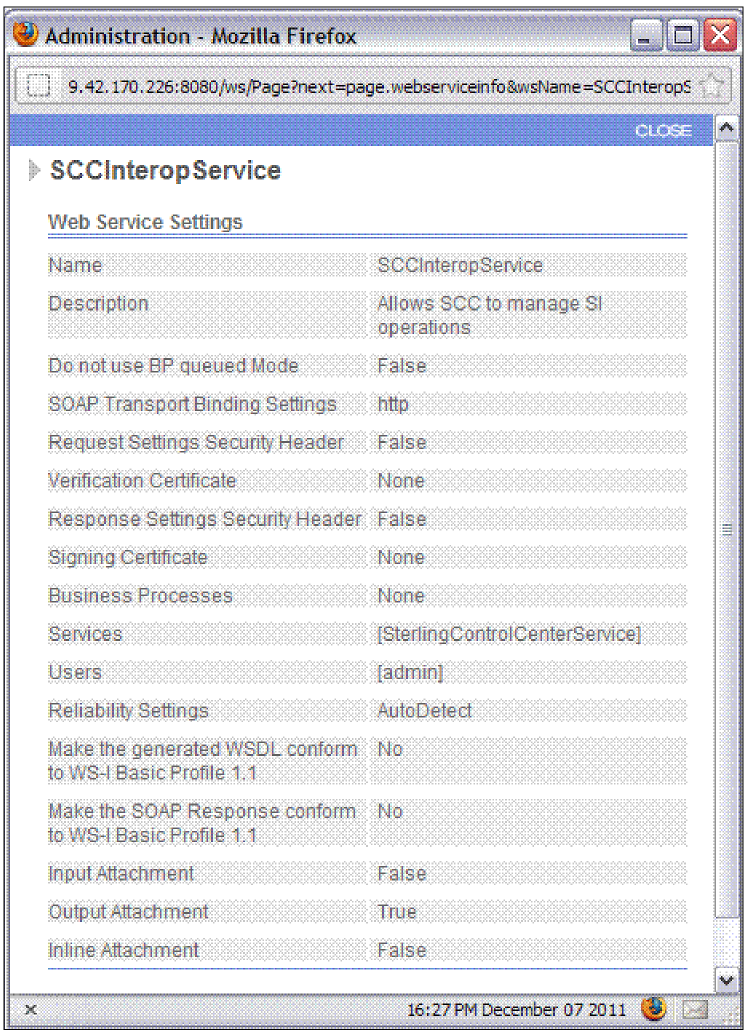


Figure 6-50 Web Service Settings

This information can be obtained by navigating on the IBM Sterling B2B Integrator dashboard (Figure 6-51) to **Deployment** → **Web Services** → **Manage** and typing `SCCInteropService` in the Search box, and clicking **Go**. Then click the **`SCCInteropService`** link.

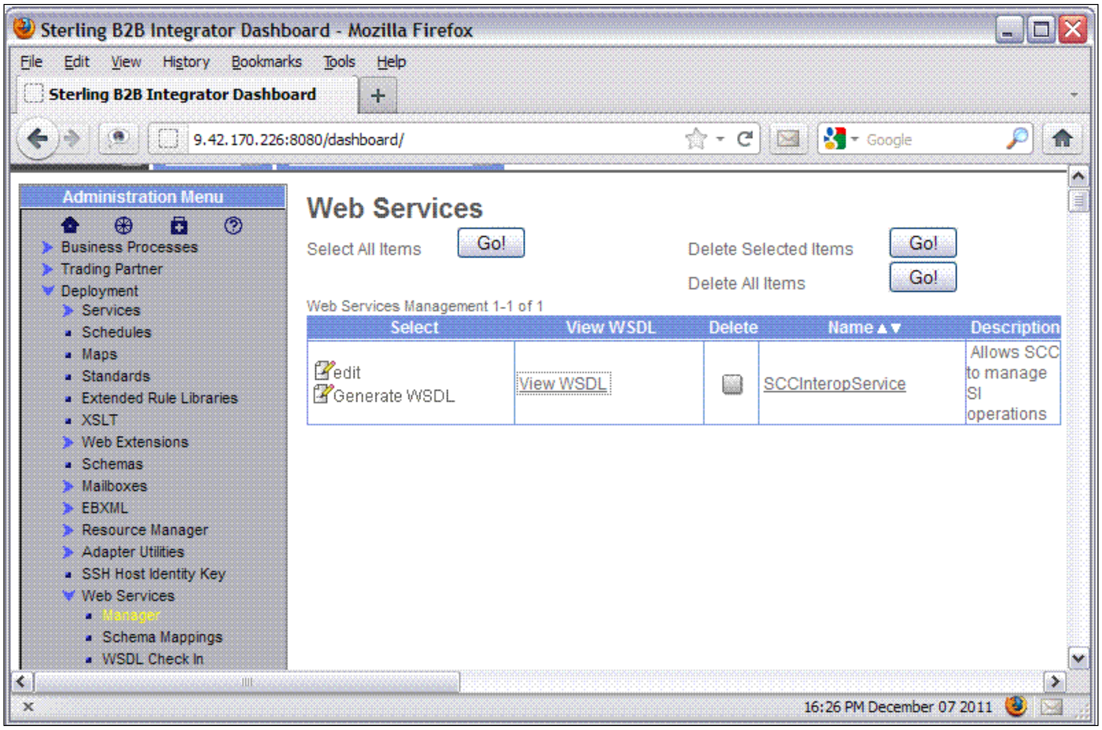


Figure 6-51 Web Services Manager

Web Service Address and Web Service Port

Figure 6-52 shows the IP address and port number that SCCInteropService is listening to; these parameters are necessary for configuring IBM Sterling Control Center. To obtain this information, click the **`View WSDL`** link as shown in Figure 6-51.

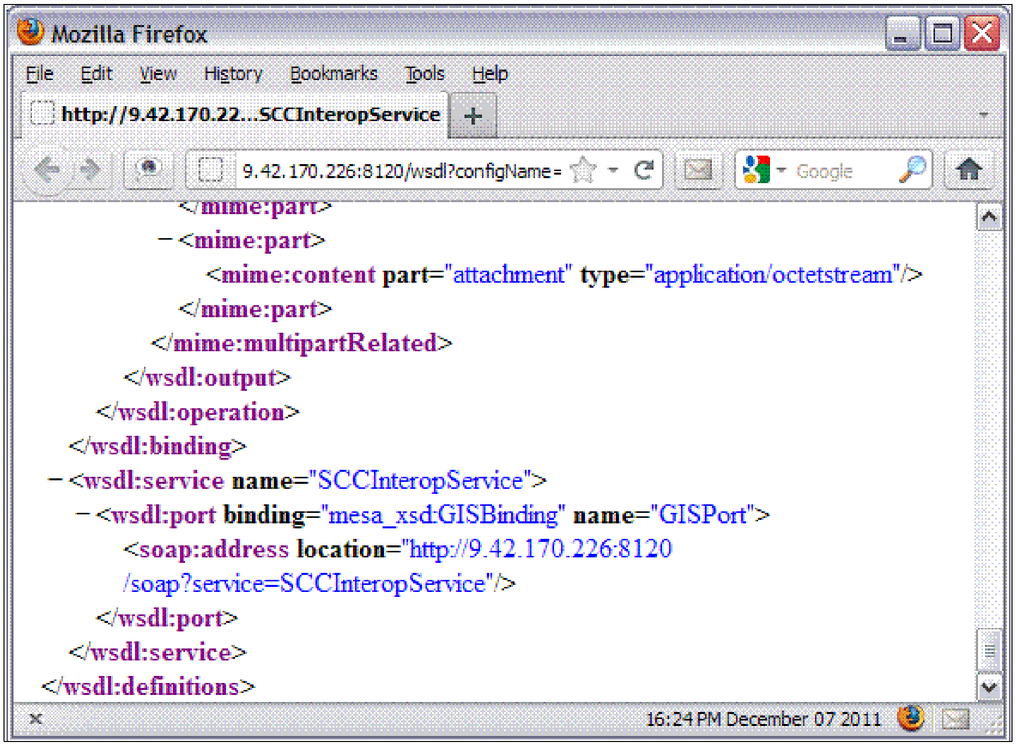


Figure 6-52 View WSDL

6.4.7 Installing and configuring the IBM WebSphere Transformation Extender Design Studio

As stated in the introduction to this book, it is assumed that any WebSphere Transformation Extender component that is needed has already been installed. For any scenario using WebSphere Transformation Extender, the Design Studio must be installed on a development platform in order to create the maps needed for translation. In this scenario, WebSphere Transformation Extender for Integration Servers has also been installed on the development platform, as well as on the server that will host the Sterling B2B Integrator. Because the inbound message used for this scenario is of custom format, no industry packs are needed for this scenario.

Setting up the Design Studio

Details for setting up the Design Studio to integrate and deploy maps to Sterling B2B Integrator are discussed in detail in Chapter 4, "Routing and transforming messages" on page 69 of this book. Here we discuss the map used to translate the inbound message to the desired format for this scenario.

Translating the message with WebSphere Transformation Extender

For this scenario, a single translation maps is needed to transform the incoming custom payment message into canonical form. For this scenario, no further translation is needed as the canonical form chosen is also the format used by the partner to whom the message is sent. Transforming the canonical message to a target format is discussed in "Step 6: Configure WebSphere Transformation Extender" on page 145.

The inbound message is described in Example 6-6.

Example 6-6 Inbound message format

```
The payments message contains information for 1 to many checks.
```

A check is represented by a fixed length row of 141 byes terminated by CR/LF. The composition of check data is described in Table 6-6.

Table 6-6 Inbound payment message format

FIELD	POSITION	SIZE	TYPE	COMMENT
ABA Route Number	00	9	INTEGER	Right Justify Zero Fill
From Account Number	09	15	INTEGER	Right Justify Zero Fill
Check Number	24	4	INTEGER	Right Justify Zero Fill
Payment Amount	28	10	DECIMAL(10,2)	Right Justify Zero Fill Decimal separator "."
Check Date	38	8	DATE	MMDDCCYY
To Account Number	46	15	INTEGER	Right Justify Zero Fill
Description/Memo	61	80	CHARACTER	Left Justify Space Fill

Using the type designer from the WebSphere Transformation Extender Design Studio, we create a custom type tree that follows the format described above. Figure 6-53 shows the type tree definition.

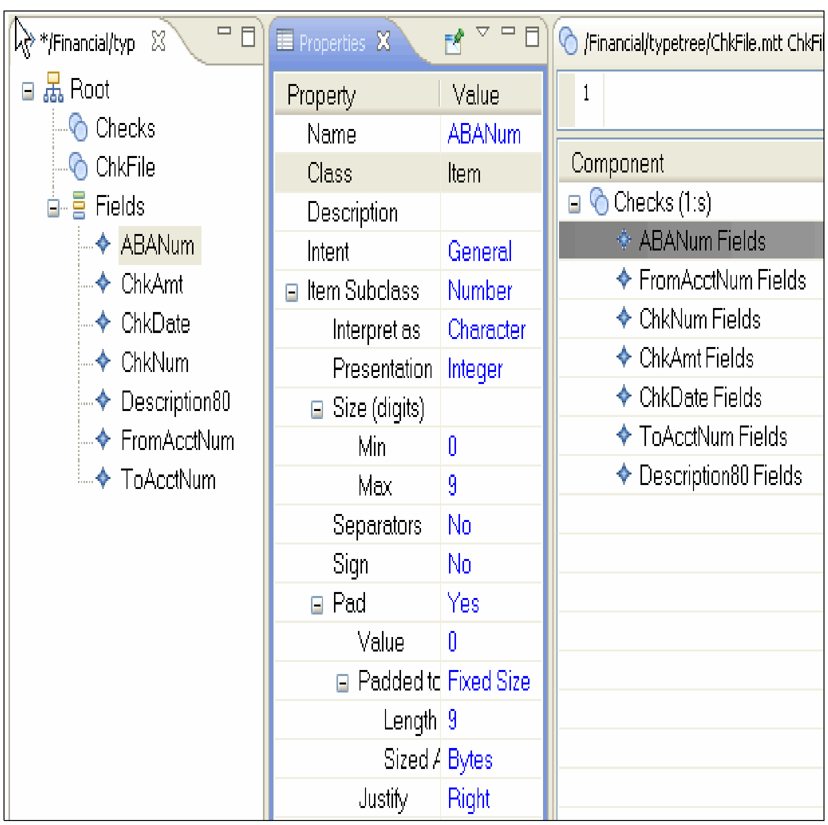


Figure 6-53 Type tree for inbound payment message

The target format for the map is defined using an xml schema definition. The content of the schema is shown in Example 6-7.

Example 6-7 Payment target schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="ChkFile">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="Checks">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ABANum" type="xs:integer" />
              <xs:element name="FromAcctNum" type="xs:integer" />
              <xs:element name="ChkNum" type="xs:integer" />
              <xs:element name="ChkAmt" type="ChkAmt_Type" />
              <xs:element name="ChkDate" type="xs:dateTime" />
              <xs:element name="ToAcctNum" type="xs:integer" />
              <xs:element name="Description80" type="Description80_Type" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="ChkAmt_Type">
    <xs:restriction base="xs:decimal">
      <xs:totalDigits value="10" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="Description80_Type">
    <xs:restriction base="xs:string">
      <xs:maxLength value="80" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Using the map designer from the WebSphere Transformation Extender Design Studio, we define the input card using the type tree described by Figure 6-53 on page 231.

The output card using the xml schema is shown here in Example 6-7.

Figure 6-54 shows the input card definition.

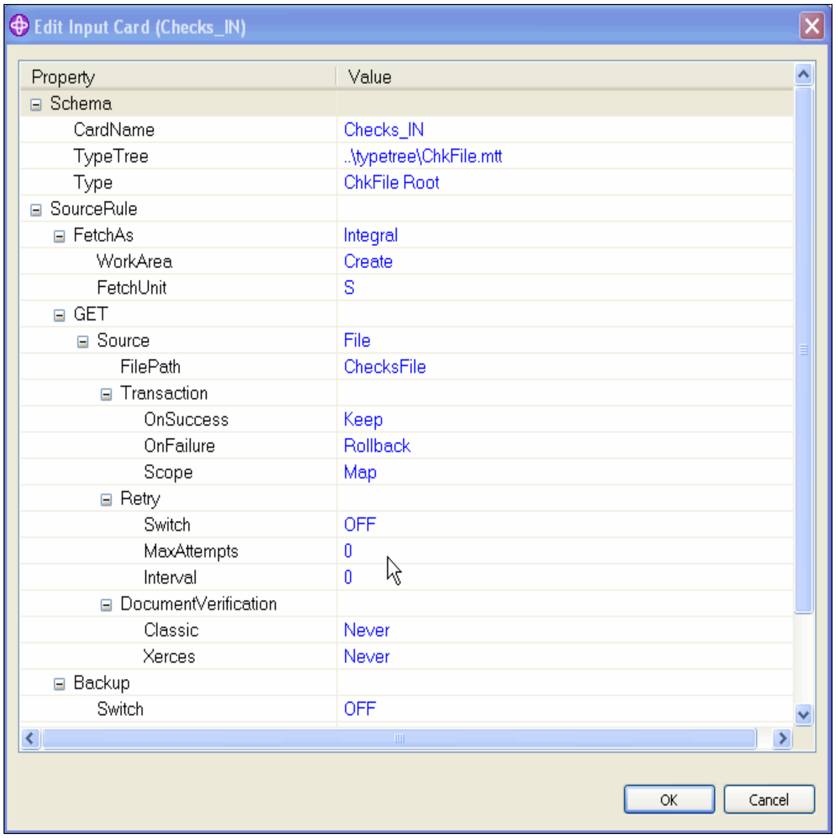


Figure 6-54 Input card definition

Figure 6-55 shows the output card definition.

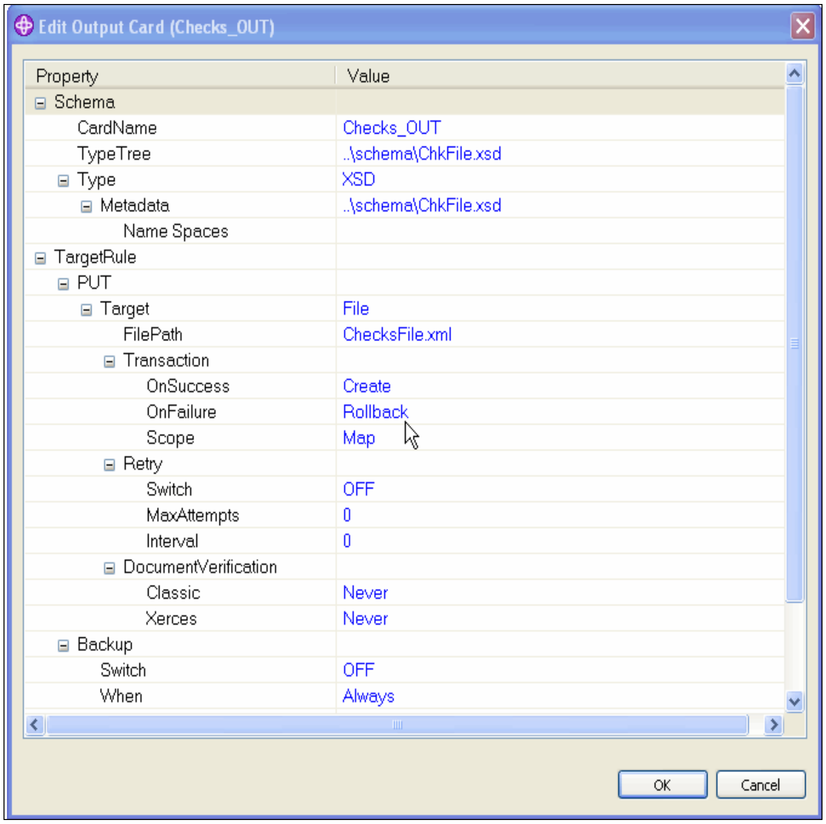


Figure 6-55 Output card definition

The map shown in Figure 6-56 loops through each payment record in the input and transforms it into an xml payment record.

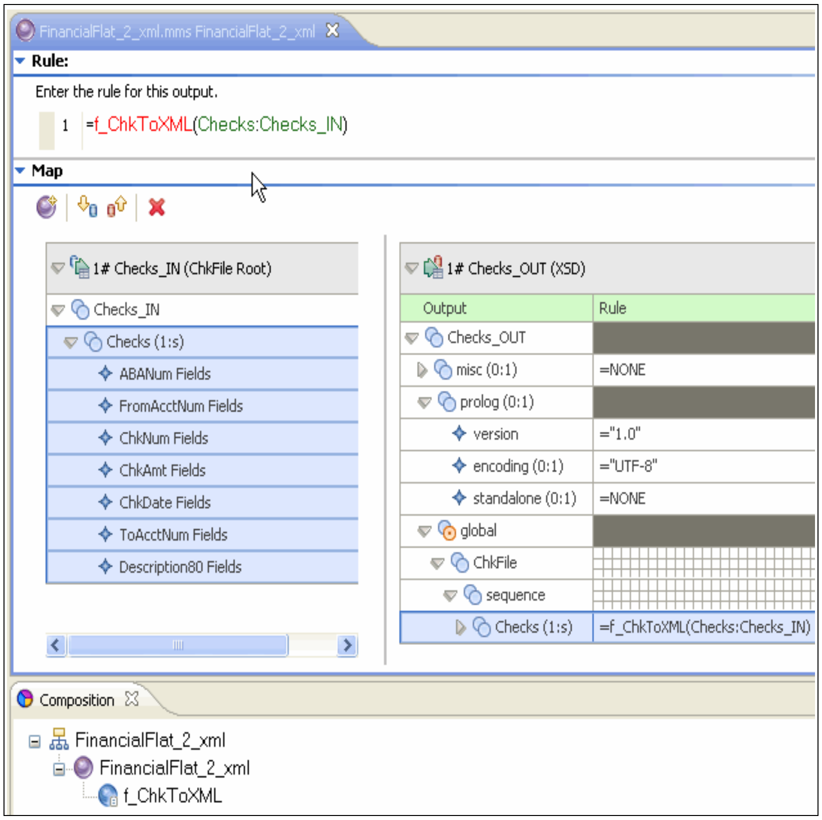


Figure 6-56 Payments translation map

After the map is created and built, it is tested on, and deployed to the Sterling B2B Integrator as described in Chapter 4, “Routing and transforming messages” on page 69 of this book.

6.4.8 Installing and configuring the IBM WebSphere Transformation Extender for Integration Servers

This topic has been covered earlier in this book. For details, see Chapter 4, “Routing and transforming messages” on page 69.

6.5 Testing the scenario

In this section, we provide a breakdown of the actions required to test the scenario.

As described in 6.3, “Presenting the Financial Services scenario” on page 181, the data flow starts with a manual intervention; indeed, the process starts when the Seller uploads an invoice file to the Buyer. The file is routed to the Seller mailbox in the Sterling B2B Integrator system at the Buyer, subsequently the entire process is automated in the Seller Sterling B2B Integrator and also at the Bank because the File Agent in the Bank Connect:Direct monitors the inbound directory and, as soon as a file arrives to that directory, Sterling File Agent sends a file to the Seller as an acknowledgement of the payment received.

In order to upload the file using myFileGateway, the Seller must know the login and password, as shown in Figure 6-57.

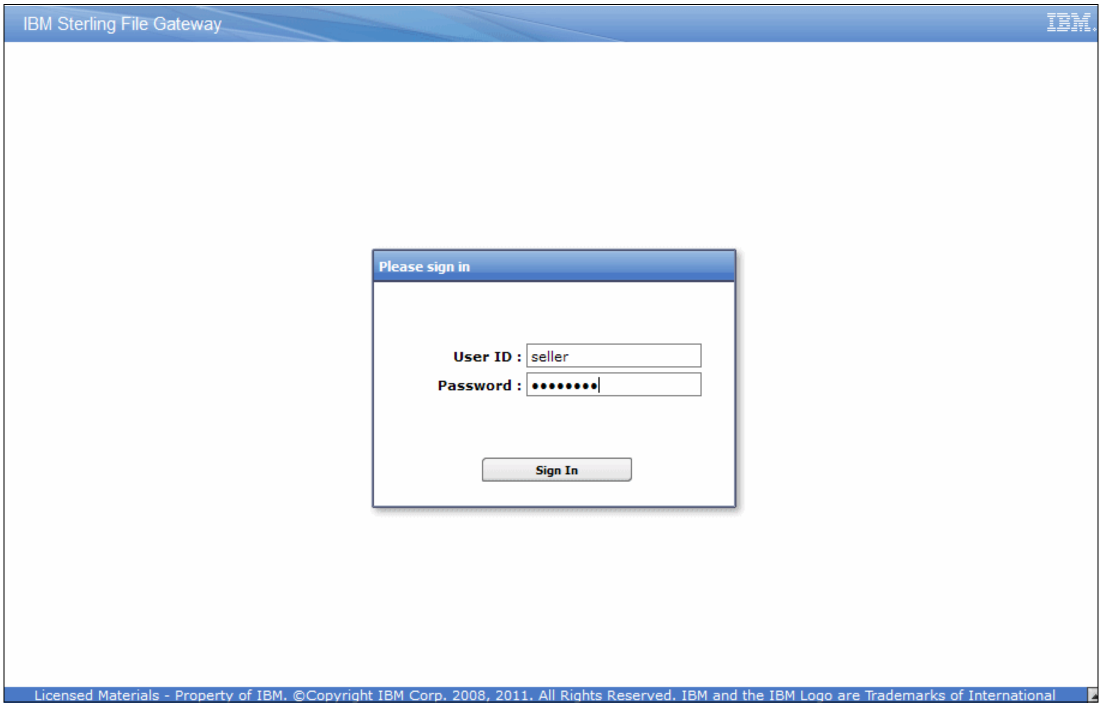


Figure 6-57 IBM Sterling File Gateway

The next steps are completely automated: when the file comes in the mailbox /BUYER/Inbox at the Buyer, a Routing Rule set up for this purpose kicks off the appropriate Business Process for further processing.

We implemented the Routing Rules shown in Figure 6-58.

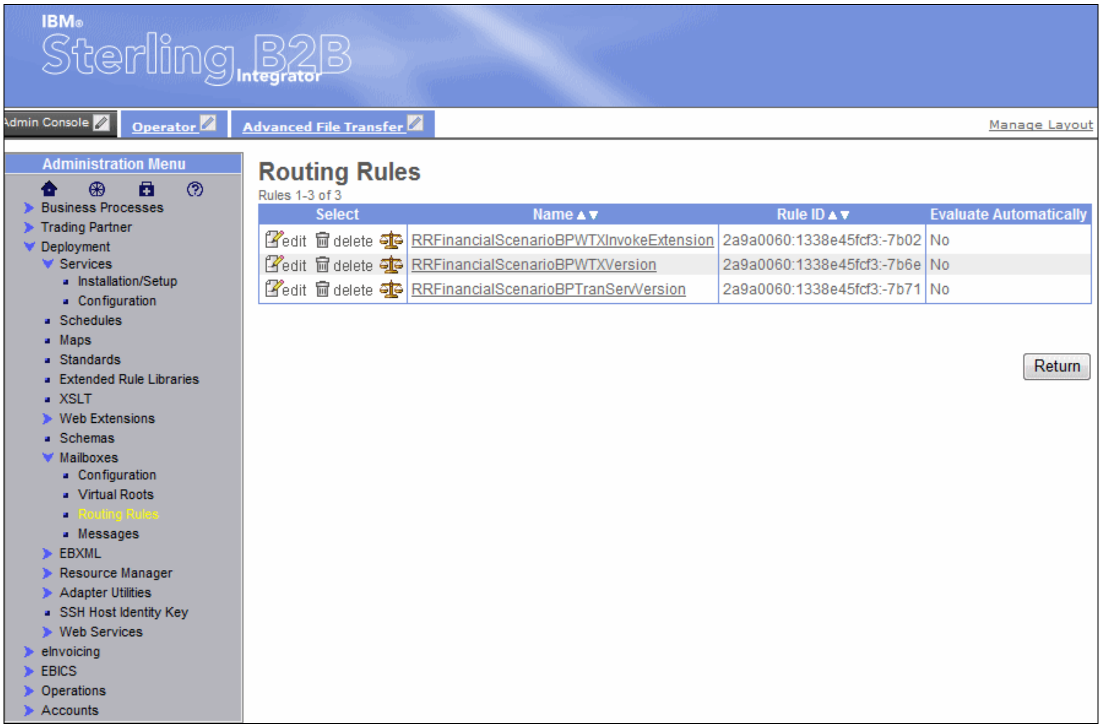


Figure 6-58 Routing rules

With these Routing Rules, you can test the corresponding Business Processes independently, as each Routing Rule monitors the /BUYER/Inbox mailbox and then starts the corresponding Business Process:

- The Routing Rule RRFinancialScenarioBPWTXVersion calls the Business Process FinancialScenarioBPWTXVersion from Example 6-1 on page 190 which implements the base Financial Scenario.
- The Routing Rule RRFinancialScenarioBPTranServVersion calls the Business Process FinancialScenarioBPTranServVersion from Example 6-2 on page 192 which implements the base Financial Scenario but with the Translation service in place of the WTX Map service.
- The Routing Rule RRFinancialScenarioBPWTXInvokeExtension calls the Business Process FinancialScenarioBPWTXInvokeExtension from Example 6-3 on page 194 which implements the Extended Scenario described in section 6.6, “Extended scenario” on page 238.

Depending on which scenario you want to test, you can enable the corresponding Routing Rule according to the above schema.

Any test ends when a copy of the XML file is routed by the bank Connect:Direct process by File Agent to the seller inbox directory C:\CDSELLERCOMM\CDBANKIN\.

6.6 Extended scenario

As an extension to the scenario, we consider the buyer's partner integration systems to connecting to his backend systems using the own internal integration layer, which is an enterprise service bus as discussed in 6.3.1, "Alternate scenarios". The extension is depicted in Figure 6-59.

6.6.1 Business value

The extension provides additional business value by seamlessly integrating the external partner connections to backend systems in the own enterprise of the buyer, by using WebSphere MQ messaging in a chain of transaction-safe operations. The particular value of the enterprise service bus is not only in allowing this single sample connection, but in supplying a unique, normalized mean for all the connectivity needs of the enterprise. It reduces the complexity and maintenance cost compared to using a different transport. See Figure 6-59 for a description of this scenario.

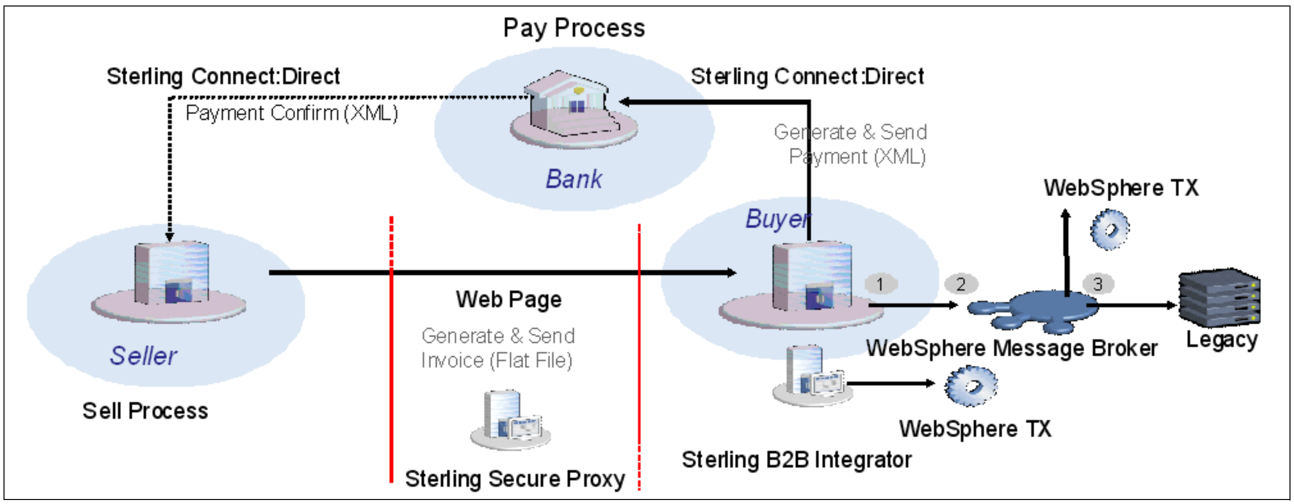


Figure 6-59 Extended Scenario with enterprise service bus connection

6.6.2 Additional prerequisites

The connect to the enterprise service bus requires additional software and skills.

Software prerequisites

In addition to the software mentioned in 6.2.1, "Software prerequisites", you need WebSphere MQ and Message Broker software.

Skills prerequisites

In addition to the skills mentioned in 6.2.2, "Skills prerequisites", you need some administration and development skill regarding WebSphere MQ and WebSphere Message Broker. General knowledge of messaging in the staff that deals with Sterling B2B Integrator is advantageous.

6.6.3 Overview of the extension scenario

The overview picture of Figure 6-59 on page 238 shows the scenario connected by IBM WebSphere MQ through the enterprise service bus to the legacy backend resources, right end. The regular message flow is from left to right:

1. Sterling B2B Integrator opens a connection to WebSphere MQ queue manager and puts (writes) a message to the IN queue.
2. The WebSphere MQ queue manager on B2B02 transfers the message to the queue manager on host ESB01.
3. The IBM Message Broker flow deployed for this scenario gets (reads) the message from the local IN queue. A WebSphere Translation Extender map designed for that purpose transforms the message to the backend format. Depending on the account number the responsible branch is chosen: content based routing and routing.

In detail, the setup uses two WebSphere MQ queue managers, five application queues and three channel definitions, as shown in Figure 6-60. These are the WebSphere MQ objects which are directly relevant for the applications and their configuration.

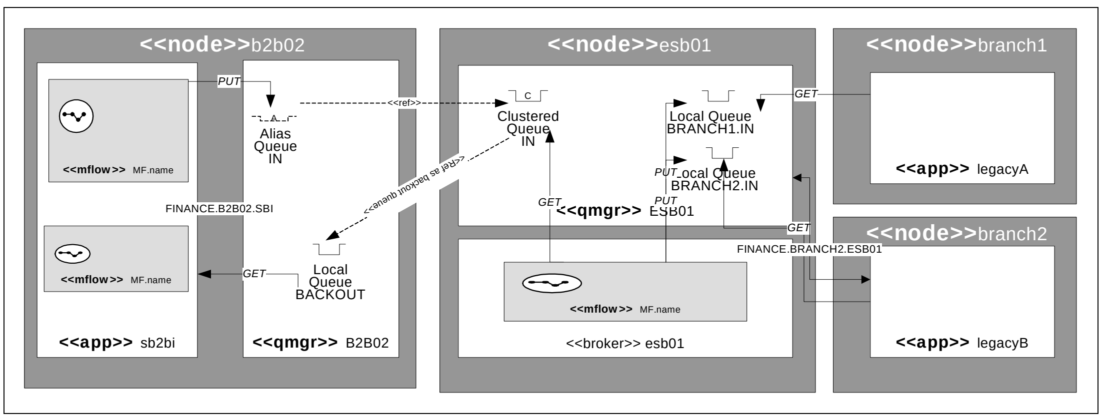


Figure 6-60 Applications, queue managers and queues

There are some other configuration steps related to the basic set-up of the messaging and enterprise service bus infrastructure, which belong to the realm of the messaging network administration:

- Create queue managers.
- Create a server connection channel.
- Create a WebSphere MQ queue manager cluster.
- Connect the queue managers in the cluster so that they can write to queues defined on other queue managers without additional administration tasks.
- Create and setup brokers.

The messaging network must be operated as an application neutral infrastructure: for example, a TCP network. It means that the objects created to set up basic infrastructure can also be used by other applications, if there are no technical reasons against it.

6.6.4 Alternate configurations

In this section, we discuss some different choices and configurations, in particular, in the WebSphere MQ topology:

- Use of WebSphere MQ distributed queueing instead of clustering
- Remote WebSphere MQ queue manager
- Remote WebSphere Transformation Extenders
- Choosing different ESB technology

6.6.5 Configuring the extended scenario

To set up the extension scenario, we need to configure Sterling B2B Integrator to send messages to WebSphere MQ, and the WebSphere MQ queue managers, the broker and the WebSphere TX extension to deal with these messages.

Configuring Sterling B2B Integrator

Most of the connection from Sterling B2B Integrator to the WebSphere MQ queue manager is configured in Sterling B2B integrator. See the description in "Configuring IBM Sterling B2B Integrator" on page 187. Figure 6-61 shows the business process used to write a message to the WebSphere MQ queue manager.

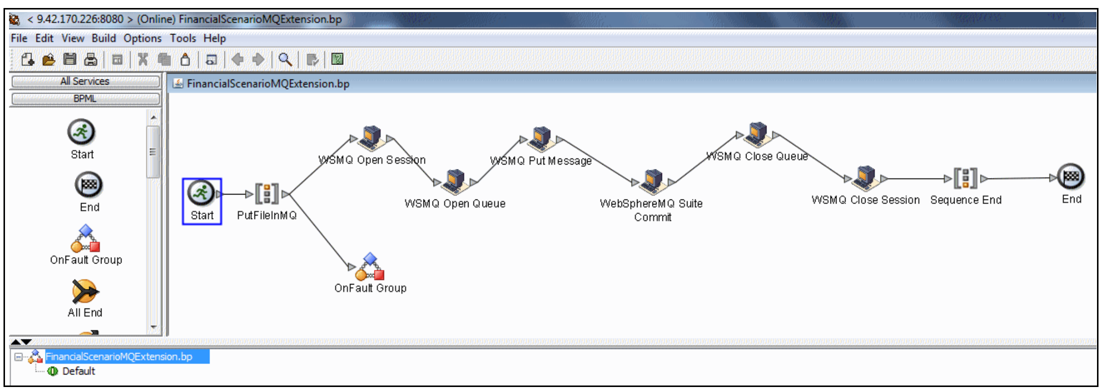


Figure 6-61 Sterling B2B Integrator business process accessing WebSphere MQ

Configuring WebSphere MQ

We assume that WebSphere MQ is already installed on your system. We give a short introduction to the use of WebSphere MQ and comment on some configuration parameters in the following sections. Because WebSphere MQ is a very large system, we recommend that new users study the IBM manuals and Redbooks publications to gain more insight.

To administrate a local WebSphere MQ installation on Windows or Linux, you can use the WebSphere MQ Explorer, call it from the Windows start menu or the command **`strmqcfg`**.

Creating queue managers

To create a local queue manager, you can use a wizard in the WebSphere MQ Explorer. For a very basic configuration, not more than the name of the queue manager is needed. Nevertheless, we discuss some hints about configuration collected in the two wizard pages shown in Figure 6-62 and following.

Choose a convenient, comprehensive and short queue manager name. We recommend to use a name with a maximum of 8 characters and to use upper case only. Upper case is in general a good choice for MQ object names. You can use the hostname if you want to see WebSphere MQ as network infrastructure, or a functional name, that allows moving queue managers.

You can make your new queue manager the default queue manager on the system, which for convenience can be called without naming it explicitly, though we do not recommend, because the shortcut can be misleading.

A default transmission queue is not required for the topologies used in this book. Default transmission queues are a mean to implement default routes from peripheral queue managers to a central queue manager in hub and spoke topologies.

We propose to set up the queue manager with a dead letter queue where messages go that cannot be delivered. We use the predefined queue SYSTEM.DEAD.LETTER.QUEUE as dead letter queue for the queue manager. If you choose a different name, be sure to define the queue. If you do not declare and define a dead letter queue, the queue manager closes any connection that tries to send an undeliverable message. This can be good in certain particular scenarios, in most of the cases it would just stop the operation without good reason. In most cases the SYSTEM.DEAD.LETTER.QUEUE can be used.

The numerical values for the maximum handle limit, the maximal count of uncommitted messages and the trigger interval should be unchanged unless your application requires. You can change these values when needed. See Figure 6-62.

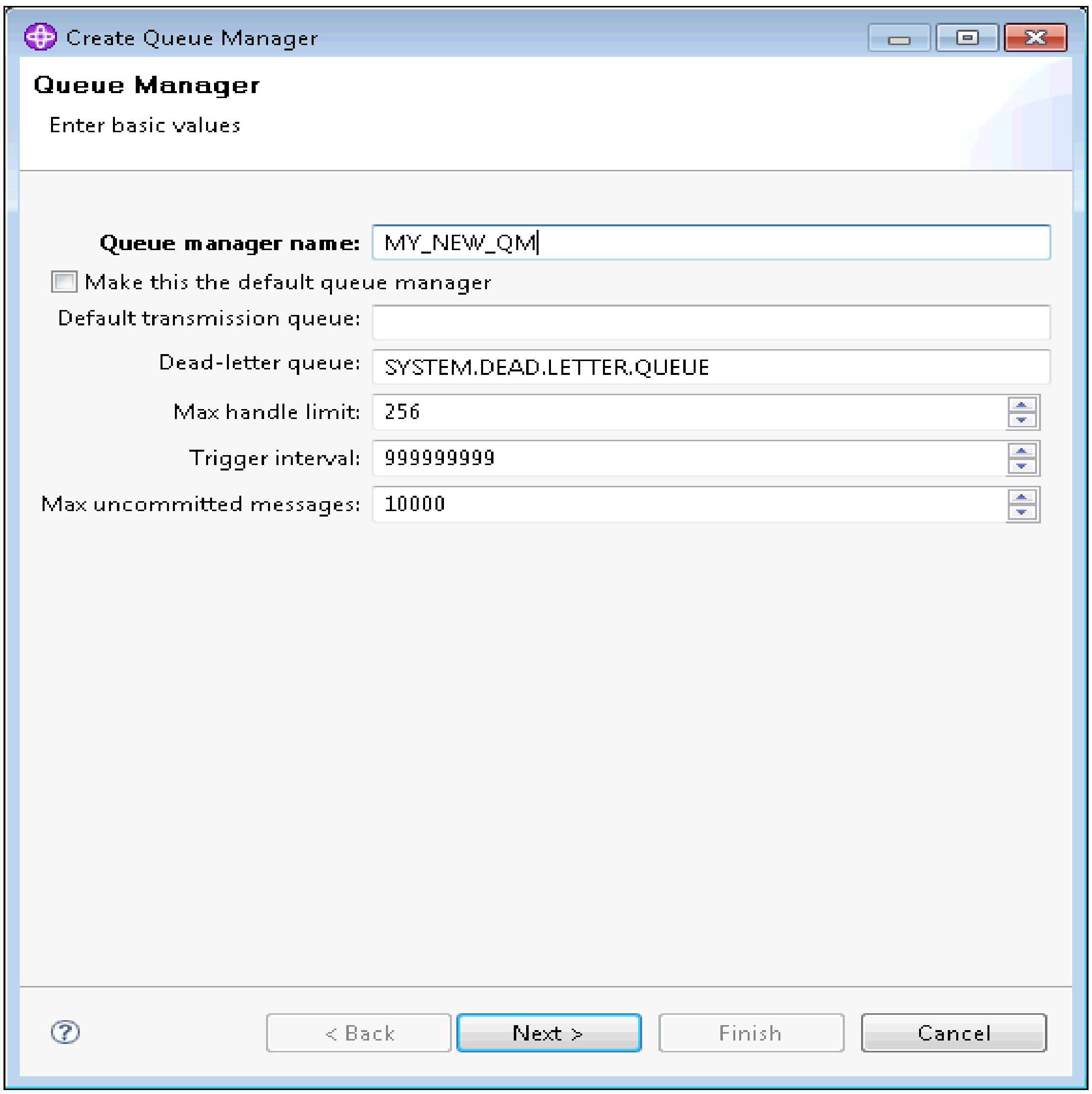


Figure 6-62 Queue manager creation by wizard, first page

The second page of the wizard, shown in Figure 6-63, requests important settings for transaction logging and storage.

Transaction logging does not mean error reporting, as in a logfile. In this context, something very alike the redo logs of a database is meant. Productive queue managers should be operated in linear logging mode to allow media recovery. In this case, the transaction logs are kept to help as message backup in case of queue corruption, which can for instance be caused by a failing disk. Development and test queue managers should use circular logging, which requires less administration tasks then linear logging.

If you use internal disks, for performance reasons the data and log paths should be located on different physical disks. The physical media location is not that relevant if you use SAN storage which is written through a battery buffered cache, though data and log should also then use different sets of physical disks, in this case for safety reasons.

The capacity of the log you need to provide depends on the volume of data you want to queue and move through the queue manager.

The volume of data relevant for this calculation are the messages in open transactions at one time. While exact calculations can become quite difficult, a rule of thumb is to take the payload and add 300 byte for message header and another 300 byte logspace for any get or put operation.

The log capacity is (calculated primary logs + secondary logs) * (logfilepages * 4 KB).

Primary log space is created at queue manager start, secondary space when needed. Changes of the numbers become effective after queue manager restart, you can adapt this value easily to growing needs. An attribute you have to consider carefully is the “logfile size” (queue manager attribute LogFilePages), because you cannot change it during the queue manager lifecycle. If you choose a value that proves to be too small, you could only increase it by recycling the whole queue manager: a change that is critical for production environments.

On the third and forth wizard page, you can keep the defaults, unless you have to change the default port 1414 because there is already another queue manager on your system. In this case you can just use 1415 or any other free port.

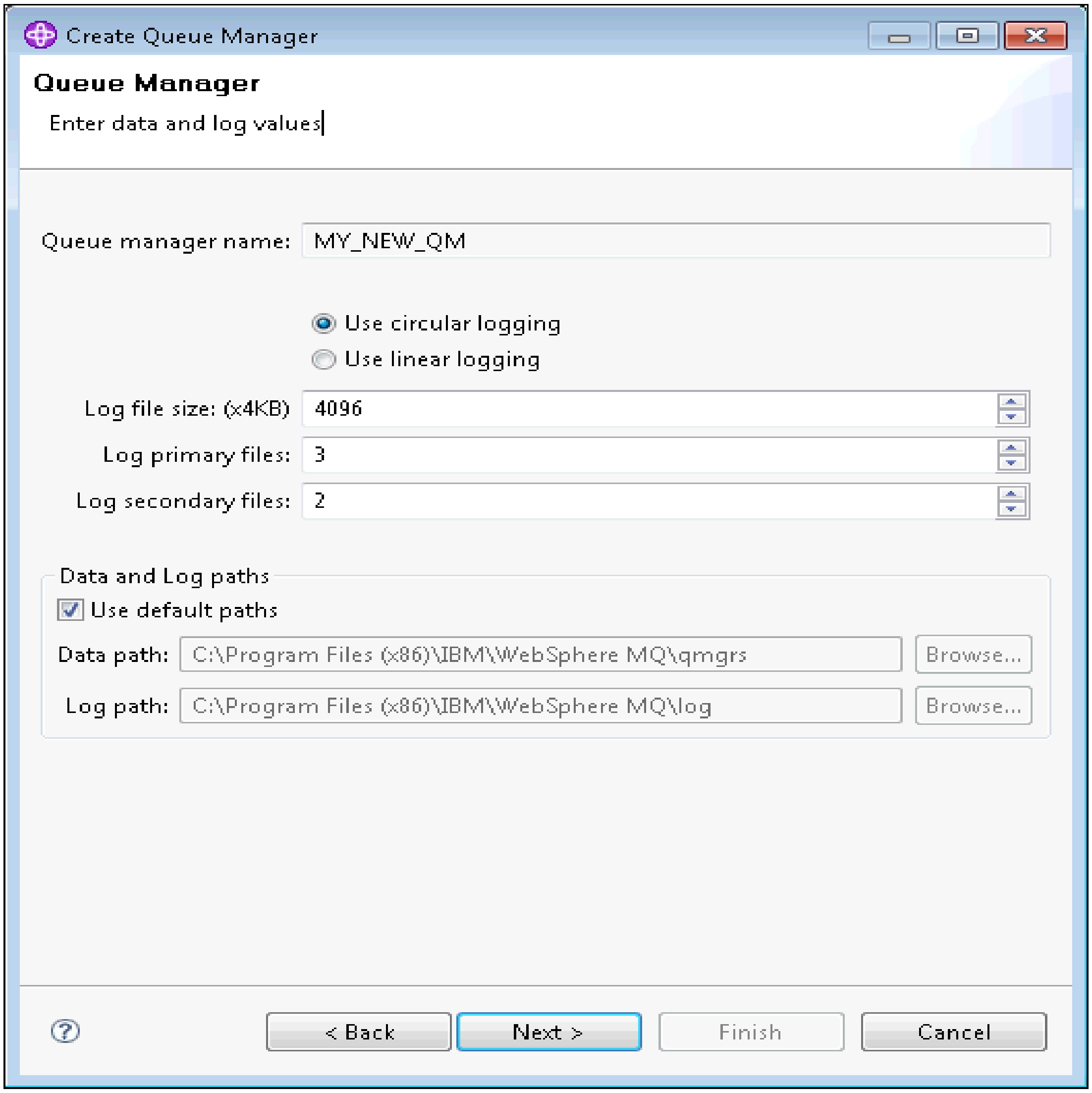


Figure 6-63 Queue manager creation wizard, second page

After you have created the queue manager, install the WebSphere MQ SupportPac MS03 to save the queue managers definitions frequently. SupportPacs are extensions, that are available from the IBM website free of charge. Many of them are supported like the product.

Tip: Download the free WebSphere MQ SupportPac MS03 to save queue manager definitions from the following website and run it frequently:

`http://www-01.ibm.com/support/docview.wss?uid=swg24000673`

The save queue manager SupportPac is essential to keep track of changes in the queue manager. It saves queue and other definitions, but not the messages in the queues.

Connecting applications to the queue manager

After you have created your queue mangers, you provide connectivity to the applications entitled to use it. We demonstrate connecting applications to the queue manager through TCP/IP. You create a WebSphere MQ *server connection channel* object to allow it, as shown in Figure 6-64.

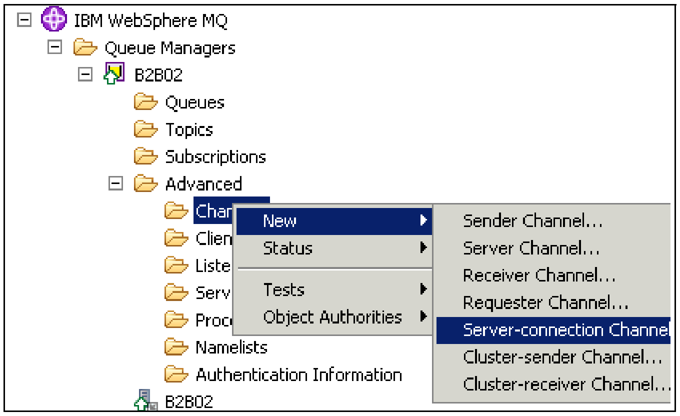


Figure 6-64 Creating a server connection channel from WebSphere MQ Explorer

Like all WebSphere MQ channel definitions, the server connection channel is basically a name to define a certain access to the queue manager. The connecting party, like the application, connects to the listener port of the queue manager and passes the name of the channel it wants to connect to. Both, application and queue manager have to use matching channel names. You can see the same pattern again in cluster channels and distributed queueing channels, described in section.

Because a B2B gateway is a security sensitive place, in particular, we need to think about securing the channel against unauthorized use. We make these distinctions:

- Authentication: Identifying who is trying to access the queue manager:
  - By IP
  - By SSL
  - By username/password combination
  - By a token from a security service
- Authorization: Determining what the identified party can do:
  - To connect, and access message in specified queues:
    - To *browse* (read) or *get* (read consuming) the message
    - To *put* (write) a new message
  - Or even to change the configuration and access rights

By default, WebSphere MQ believes in the user names that remote systems present when connecting. Therefore, it is crucial for the security of a WebSphere MQ queue manager network to limit access to the MQ System objects. Otherwise unauthorized parties could pretend to be MQ administrator. The means to secure a new channel is to fill the *mcauser* attribute of the channel; the queue manager replaces the user names of incoming messages presented by the sending system with the *mcauser* value. The parameters inserted in the server connection channel wizard are listed in Figure 6-65.

```
NAME: B2B.FINANCE.SBI
comment (“DESCR”): Chl for B2B Integrator for Finance
transmission Protocol (“TRPTYPE”): TCP
copy from existing channel (“LIKE”): SYSTEM.DEF.SVRCONN
technical user (“MCAUSER”): sbiuser
```

Figure 6-65 Parameters for the creation of a server connection channel

Before you can use the channel, you have to add the declared *mcauser* as a system user (windows) or group name (unix) and authorize it in WebSphere MQ's object authorization manager. It is depicted for a Windows operating system in Figure 6-66. You can access the object authorization manager from the WebSphere MQ Explorer, as shown in Figure 6-67, or create the access rights with the utility command **setmqaut** from the command line. We first grant the user a basic read-only access, and add the "put" and "get" authorities needed to write to and read from specific queues later.

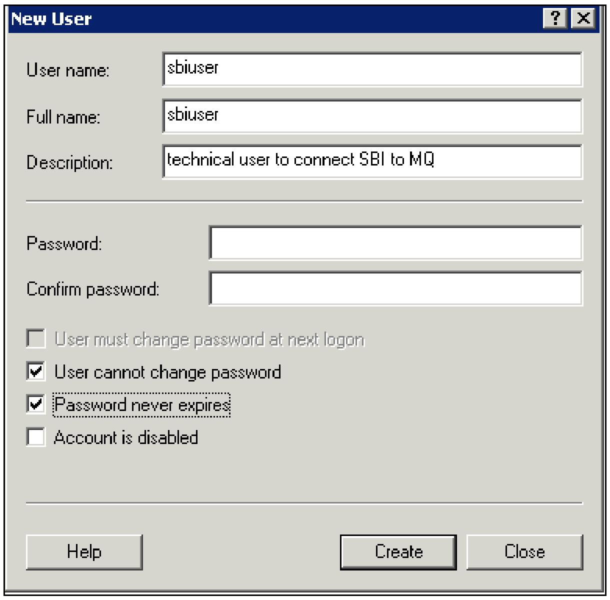


Figure 6-66 The technical mcauser must be created on operating system level

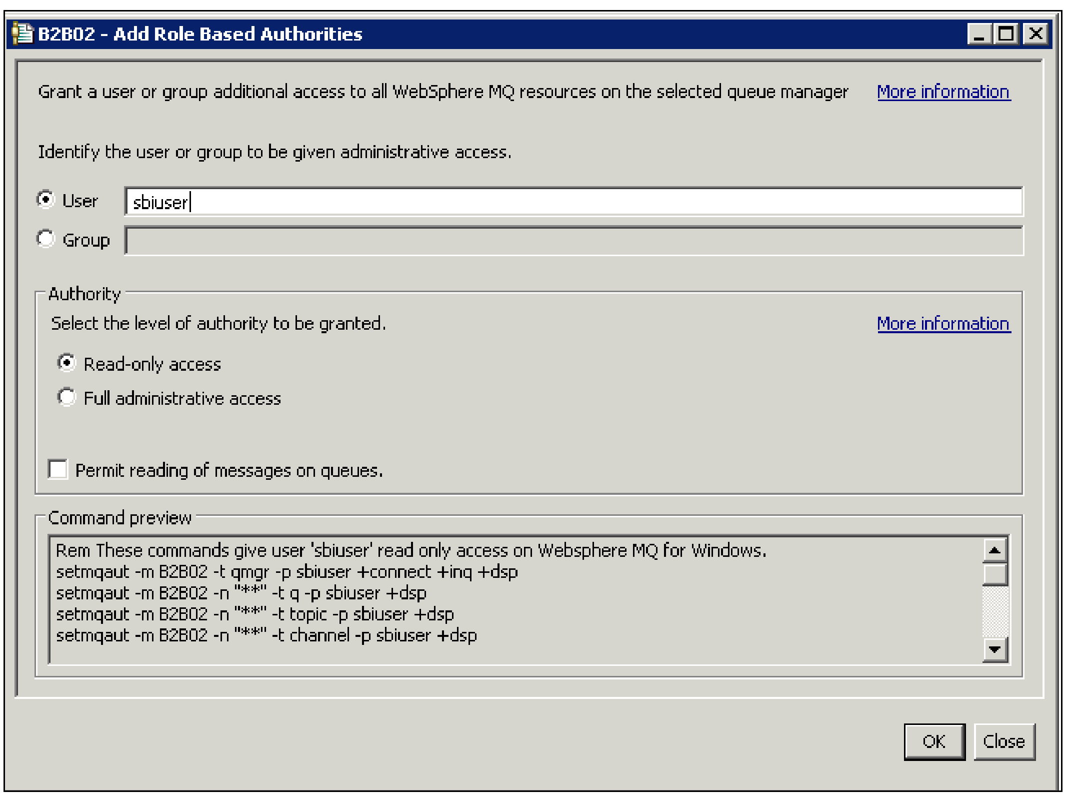


Figure 6-67 Grant basic read only access to a technical user

As an alternative, a fast and convenient way to attach only local applications to a queue manager hosted on the same node is the binding mode. The dialog is shown in Figure 6-68.

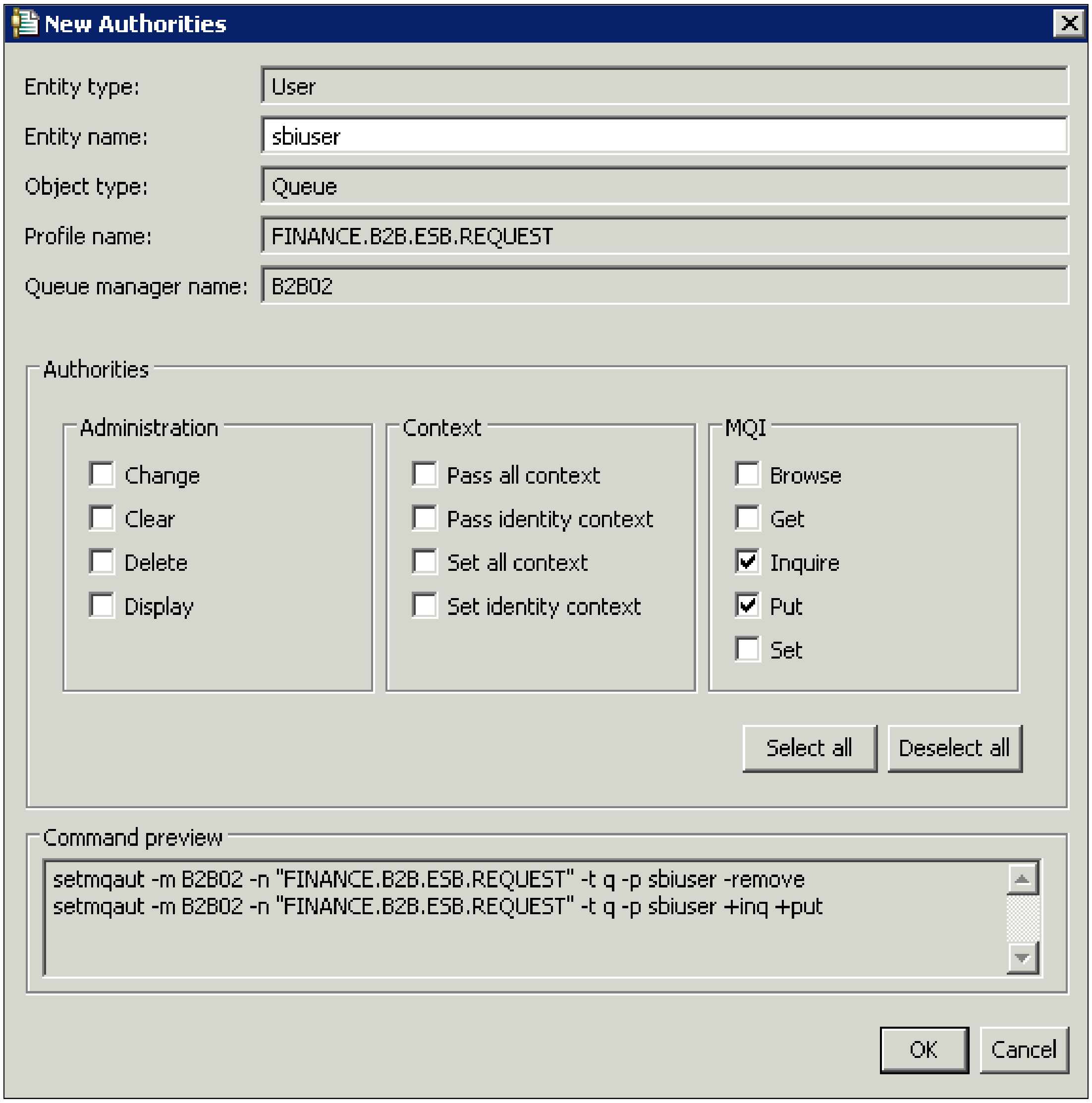


Figure 6-68 Grant put and inquire access

Connecting queue managers in a cluster

After creating queue managers, you want to connect the queue managers so that they can exchange messages. We demonstrate in the following connecting queue managers through a WebSphere MQ queue manager cluster, which organizes the distribution of messages in the cluster with low administration overhead, and is capable of load balancing.

Within a WebSphere MQ queue manager cluster, applications can write to queues that are hosted on remote queue managers, not only to the queue manager they are directly connecting.

To define a queue manager cluster, as Cluster ITSO shown in Figure 6-69, we define two queue managers as the full repositories of the cluster. For all cluster members two cluster channels are defined, a cluster receiver channel to itself (like TO.B2B02) and a cluster sender channel object to one of the repositories (like TO.ESB01). A cluster repository queue manager itself will have a sender channel to the other repository. The cluster receiver channel definitions serve as templates used by all queue managers in the cluster to auto-define sender channels to that queue manager.

Queue manager clustering is a powerful and versatile technology. You can define multiple clusters for differentiated types of use, for example, with different service levels:

- Unencrypted or encrypted channels
- Fastpath or slow transport of large messages during low traffic times

You could (and in large organizations, should) use clusters overlapping in a gateway to separate zones.

The WebSphere MQ queue manager cluster is completely different than physical clustering on the hardware level, which is often used to secure a single queue manager with services like IBM AIX PowerHA (former HACMP) or comparable products of other sources. Queue manager cluster and physical cluster are complimentary concepts that fit together well; queue manager clusters scale the performance up, while physical cluster tales care of fail over safety, the second now often replaced by multi instance queue manager.

There is an alternative configuration called MQ distributed queuing. While clustering is a feature-rich, easy to use, and mature technology, distributed queueing is the basic way of connecting queue managers, as shown in Figure 6-69. We discuss the principles of distributed queuing in “Use of distributed queueing connections” on page 250.

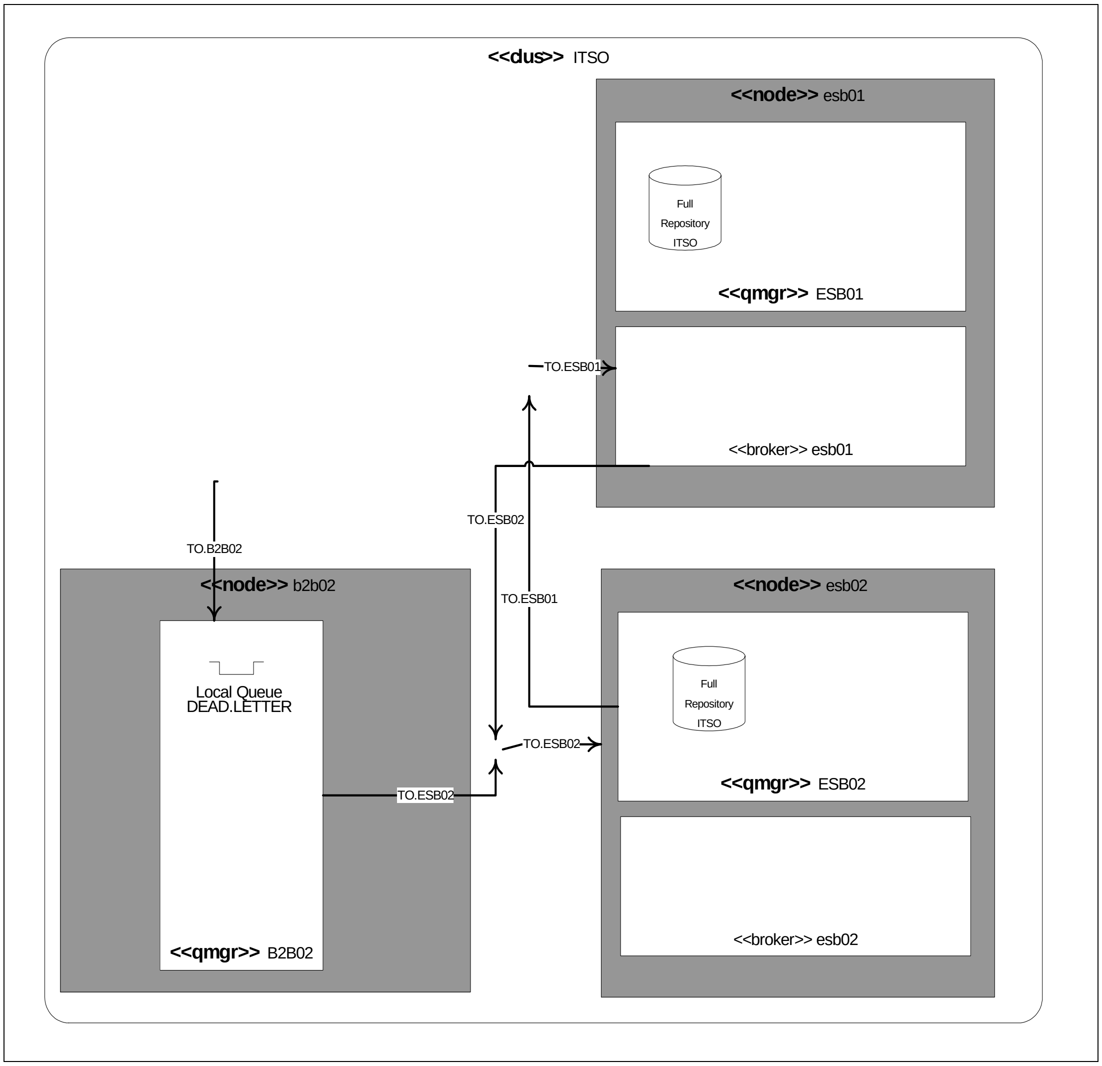


Figure 6-69 Queue manager cluster configuration for cluster ITSO

Configure queues

WebSphere MQ organizes the actual store and forward of messages in queues. Some of the categories of queues are explained here:

- A *local* queue is the basic type of queue. It is the only one that stores messages physically, and the only type an application can read from.
- An *alias* queue is a facade to a queue. Alias queues can carry specific access rights to discriminate read and right access, and can help to implement different naming concepts both suitable for network administration and application operation at the same time.
- A *cluster* queue is a queue that is writable from all queue managers of a cluster. If there are two ore more instances of a cluster queue (queues with same name) available in the same cluster, the senders distribute messages among them for load balancing.
- There is also the *remote queue*, which is a specific form of alias queue that points to a foreign queue manager. It is writable only. It is a concept of distributed queueing, the other form of cooperation between queue managers when you don't use clustering. It is accompanied by a *transmission queue,* local queues that serve as temporary store for messages before they are shipped to the target queue manager.

As shown in Figure 6-70, we create an alias queue on queue manager B2B02 as direct target for Sterling B2B Integrator. This alias points to a cluster queue that is located on queue manager ESB01. The cluster mechanism forwards the messages from queue manager B2B02 to the local queue on queue manager ESB01, where the middleware processing happens in a WebSphere Message Broker flow. We discuss that very soon in detail. At the moment we want to point out the possibility of load balancing at that place.

If the load increases and requires more processing power than the physical node ESB01 can supply, you can scale up and load balance to other esb-nodes by just creating another local queue of the same name on queue manager ESB02 and deploying the same message flows there, too.

The second queue, the B2B gateway queue manager, is a local queue that serves an error and backout queue for the scenario. If the middleware cannot process messages, for instance, because they are malformed, it bounces such indigestible (“poisoned”) messages back to the origin, because we do not want the esb queue managers to be clogged, and because the root cause of the error is easier to determine closer to the origin of the erroneous message.

Last, we define two local queues for the output to the legacy systems. We do not spell this out, but the messages could also be forwarded to other nodes. Everything is possible from client connection (as we demonstrate), cluster queue and distributed queueing.

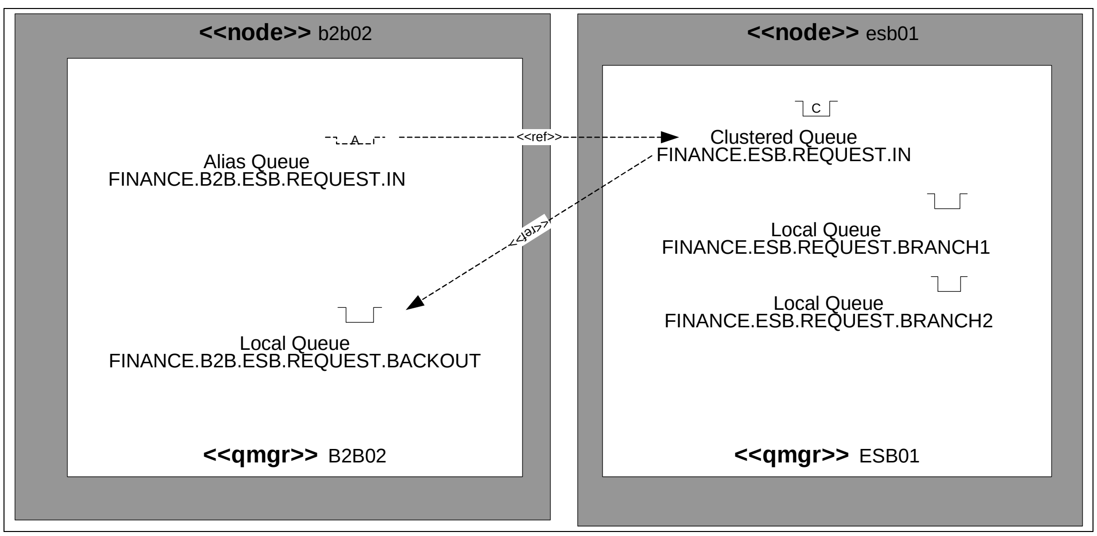


Figure 6-70 Queue topology with full queue names

Throughout the whole train, we need to consider the message size and the maximal queue depth, that is the number of messages stored in the queue manager:

- Maximal message size is an attribute of queue manager, queues, and channels. We need to adjust the values in all of them if we want to work with larger messages. It is also good practice to separate the paths of fast and heavy messages.
- The maximal queue depth (MAXDEPTH) is the number of messages a queue can store. It determines how long the queue manager can buffer for a dysfunctional subsequent system, for instance in planned or unplanned maintenance breaks.

Our declared objective is to be work without interruption, or at least without letting our B2B partners know when the processing is interrupted. So we want to supply so much room in the queue that messages can queue as long as wanted and needed to bridge the maintenance window.

Use of distributed queueing connections

Distributed queueing is the basic approach to connect WebSphere MQ queue managers. It takes some more manual configuration steps than clustering, and offers less functionality, in particular no load balancing. After being configured, the distributing queueing setup is easy to maintain and simple to understand. Most important, because of simplicity, distributed queueing is very suited for the use in external partner communication and in perimeter zones.

We do not spell out the whole scenario above as an alternative to clustering, but explain the use of distributed queueing with a realistic sample that shows how our security gateway queue manager B2B01 can be connected to the queue manager B2B02.

To allow messages to be put on a queue on a remote queue manager, we need to create at least on the near (sender) queue manager, B2B01:

1. A *remote queue* definition: FINANCE.B2B01.B2B02.IN
2. A *transmission queue* definition: B2B02
3. A *sender channel definition:* B2B01.B2B02

And on the remote (receiver) queue manager, B2B02:

4. A matching receiver channel definition: B2B01.B2B02
5. The queue which is target of the transfer. This queue can be local, alias, remote, or cluster queue.

For use in a security sensitive zone, you can use a server and requester channel pair instead of sender and receiver. The message flow with server and requester is like in sender and receiver from letter “s” to letter “r”, but the connection is not initiated by the sending party, but by the (receiving) requester. It is most suitable in situations where a queue manager in the DMZ needs to transfer messages to a queue manager in the trusted zone.

It defines a one way route from sender B2B02 to receiver queue manager ESB01. If you also want to send messages the way back, you need to create corresponding definitions.

The connection between queue managers is done by sender and receiver channels. Like all WebSphere MQ channels, the definitions must match by name. We propose to define the channel names like this:

```
<SENDER_QM>.<RECEIVER_QM>[.<XX>]
```

Channel names are limited to 20 characters. Remember that we recommended to keep queue manager names shorter than nine characters, although MQ allows queue manager names to be longer. It was because of this channel naming pattern, which is a kind of de facto standard in the WebSphere MQ world. The two characters qualifier space can be used to give a counter or qualifiers signaling things like encryption or compression.

Configuring WebSphere Message Broker

WebSphere Message Broker is a routing and transformation application working on top of WebSphere MQ. While WebSphere MQ is the messaging backbone of an enterprise service bus, WebSphere Messages Broker provides connectivity to other protocols and standards and a number of custom translation capabilities.

In our sample scenario, WebSphere Message Broker has the task to mediate between the B2B gateway and two different legacy backend applications. The backends use different data formats and have slightly different responsibilities. The ESB, say WebSphere Message Broker is responsible to deal with the particularities of these backends because we decided for architectural reasons that the B2B gateway must not deal with details of backend implementations: one of the simple truths of service oriented architecture.

Figure 6-71 shows the Message Broker Toolkit.

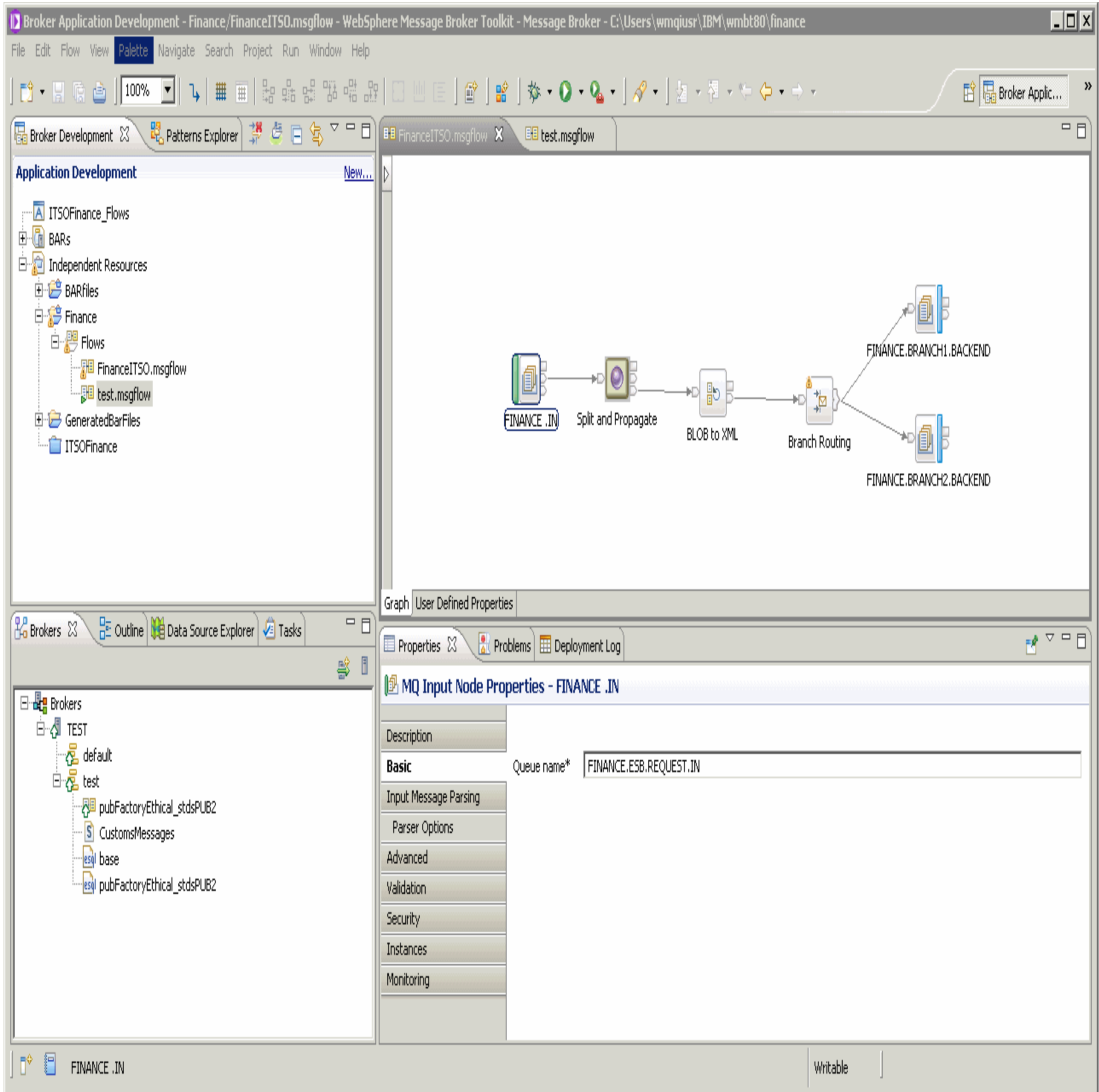


Figure 6-71 Message Broker Toolkit overview

You can think of the WebSphere Message Broker as a platform or application server. The applications are developed and packaged in a specific eclipse based build time environment called Message Broker Toolkit. Deployment is also done from this tool or from the Message Broker Explorer, a plug-in into the WebSphere MQ Explorer suitable for Broker administrators. The applications are deployed to the broker runtime component. You can separate the runtime, and also scale up to the limit of the physical OS, by using several execution groups for different applications.

Message broker flow development

A message broker flow basically needs input and output nodes at least. In our scenario case, both source and target are WebSphere MQ queues. As shown in Figure 6-72, the flow consists of six nodes.

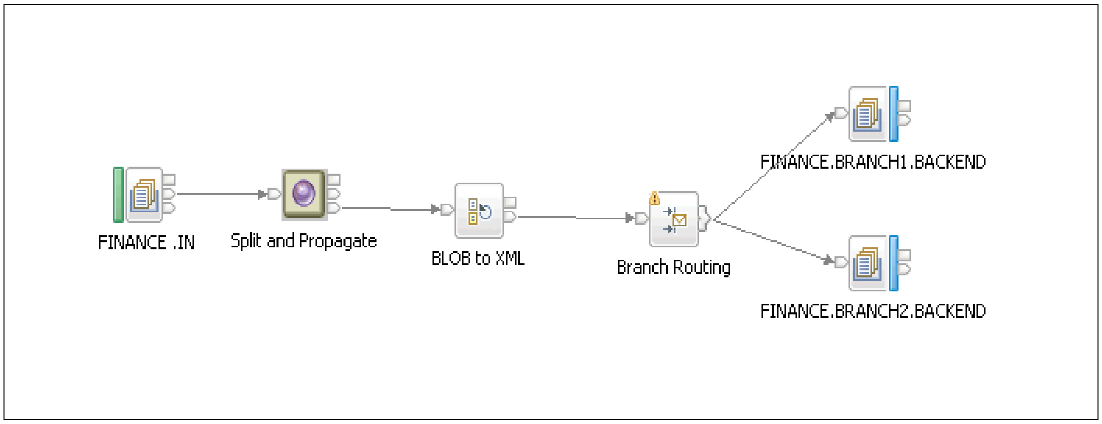


Figure 6-72 The message flow

These six nodes are as follows:

1. The input node specifies the queue where messages come from and the parser used to interpret the message. This could be a XML parser or a parser for custom wire formats. In our scenario we do not parse the message on input but read it as a binary large object- a BLOB, because we just pass the message as a whole to the following WTX node.
2. The WTX node has the purpose to split a compound message containing an unknown number of single check - notifications into single messages. These messages are XML messages as well. We explain the details of the WTX map in “WebSphere Transformation Extender configuration” on page 254. At this place, in the WebSphere TX node, we only have to reference the place where Message Broker can find the TX map on the server.
3. The parse node parses the output messages that the WTX node produced to make it handy for the further processing in the broker flow. In rare cases it can be clever to work directly on a BLOB, or parse only partial when needed. In most cases it is good to parse to the XML tree in advance, like we do here.
4. The route node makes a content based choice: Checks with an account number smaller than 2222222 go to the first Branch 1, the others to the second branch.
5. MQ output nodes that point to the branches’ queues.
6. MQ output nodes that point to the branches’ queues.

Message broker flow deployment

After developing the message flow in the Message Broker Toolkit, the flow needs to be deployed to the broker runtime environment on the server, as shown in Figure 6-73. Message flows are deployed to execution groups, which are the WebSphere Message Broker artefact for a completely distinct execution environment running its own jvm instance and native operation system processes.

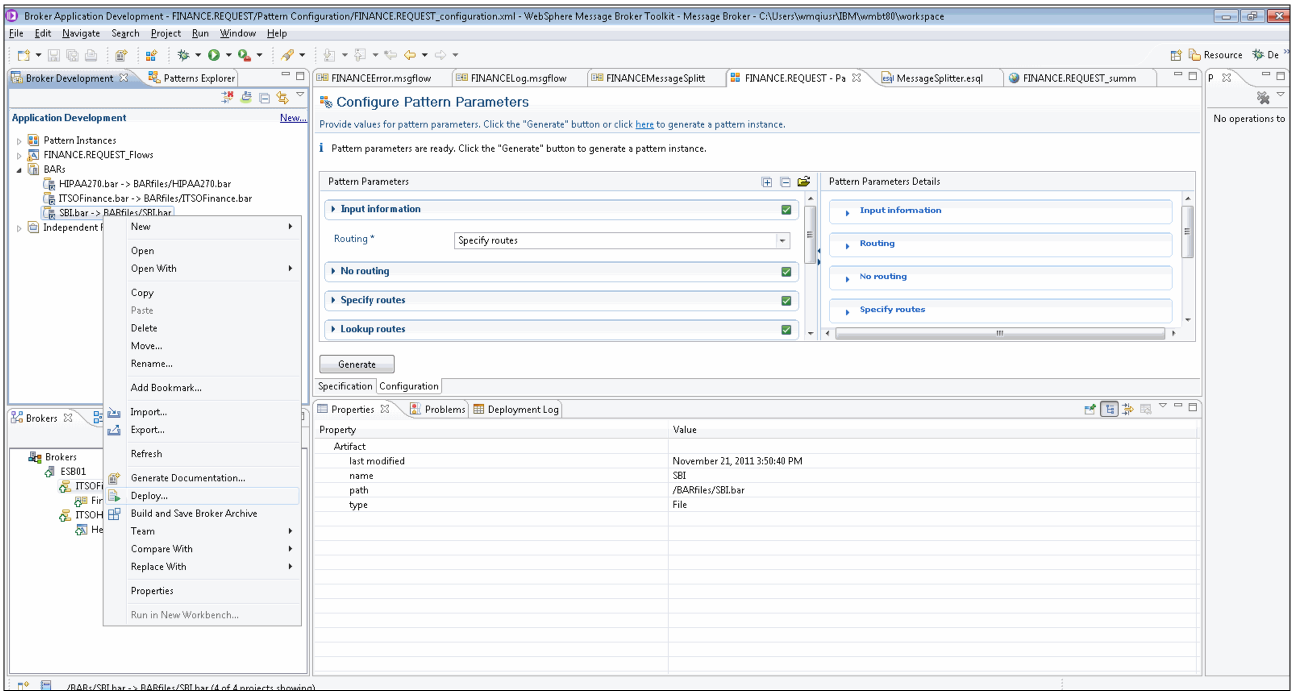


Figure 6-73 Deploy a barfile to the broker runtime

In our example, we run only a single execution group called “default” on the server. In real world operations, you might have some distinct execution groups for different projects of your enterprise, or representing for different clients (to design for isolation) or deploy message flows to some execution groups in parallel (to design for high loads).

WebSphere Transformation Extender configuration

For the extended financial services scenario, WebSphere Transformation Extender is used to split the xml payments message into individual payments and then passed back to WebSphere Message Broker to continue processing. Because both input and output are defined by the xml schema definition described earlier in the chapter, no other type tree is needed.

The same XSD is used to define both the input card, as shown in Figure 6-74, and the output card, as shown in Figure 6-75.

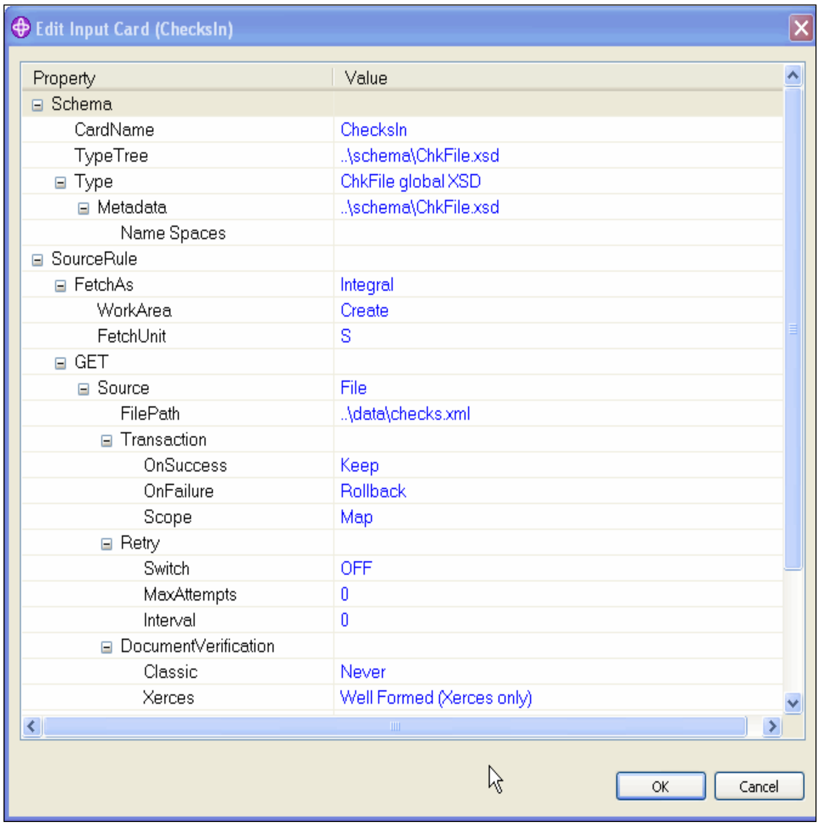


Figure 6-74   Input card for payments split map

You might notice that we set the document verification attribute to Well Formed (Xerces Only) in the input card because the xml is generated internally, and we know that it adheres to the xsd. Because WebSphere Transformation Extender does not perform verification on output, it is not necessary to set this attribute on the output card.

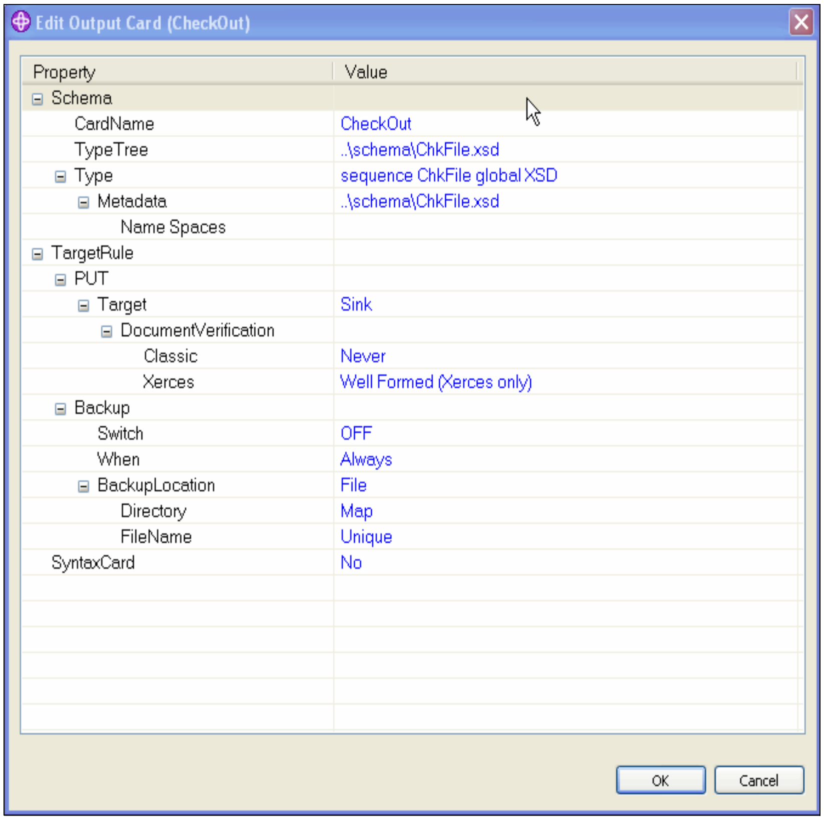


Figure 6-75 Output card for payments split map

One thing that you might notice about the output card is that we do not choose the entire document as our output type. Because we are only interested in individual payment records, the type definition for the output card is the element (sequence) that contains the repeating group of payments (Checks).

Figure 6-76 and Figure 6-77 show the two-step process for splitting the payment message into individual payments.

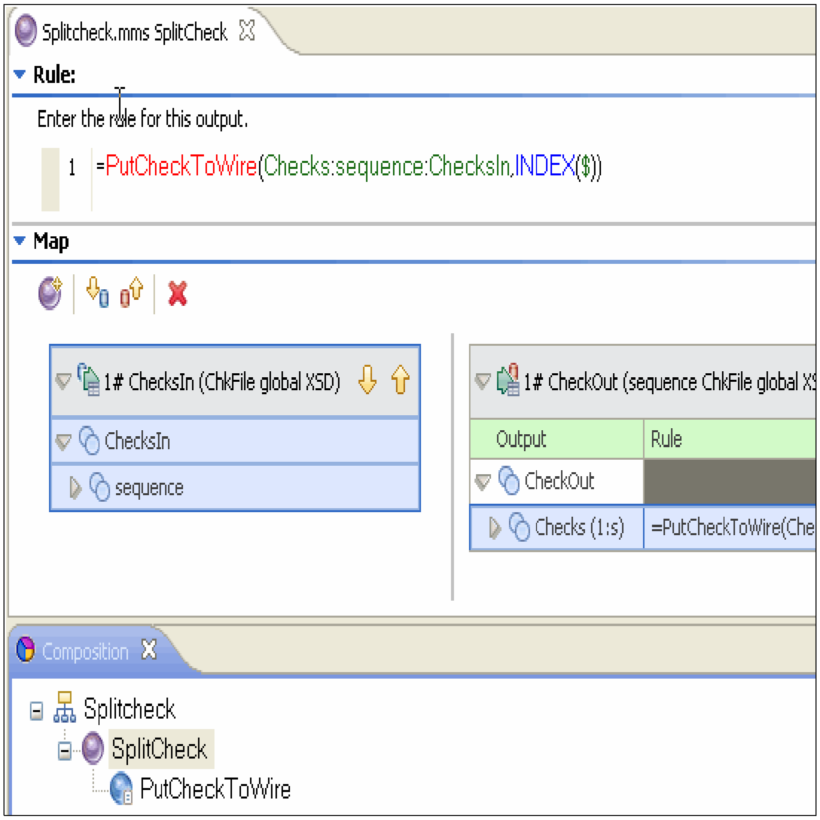


Figure 6-76 Executable map passing individual payment to the functional map

Figure 6-76 shows the executable map calling the functional map named PutCheckToWire, passing the xml element Checks. Checks contains the payment information to be used by the target application or service. You might notice that we also pass a second parameter to the functional map using the syntax INDEX($). It is done solely to expedite testing the map. By passing in the current index of the payment, we can output each payment individually to a separate file and review the results to make sure that the correct data is passed to the output terminal from the functional map after we deploy to the target platform.

Figure 6-77 shows the functional map used to output each Checks record to the named output terminal using the WIRE adapter with the PUT rule. In the second line of the rule, you see a comment containing the rule you would use to test using the FILE adapter as described in the earlier narrative discussing the executable map.

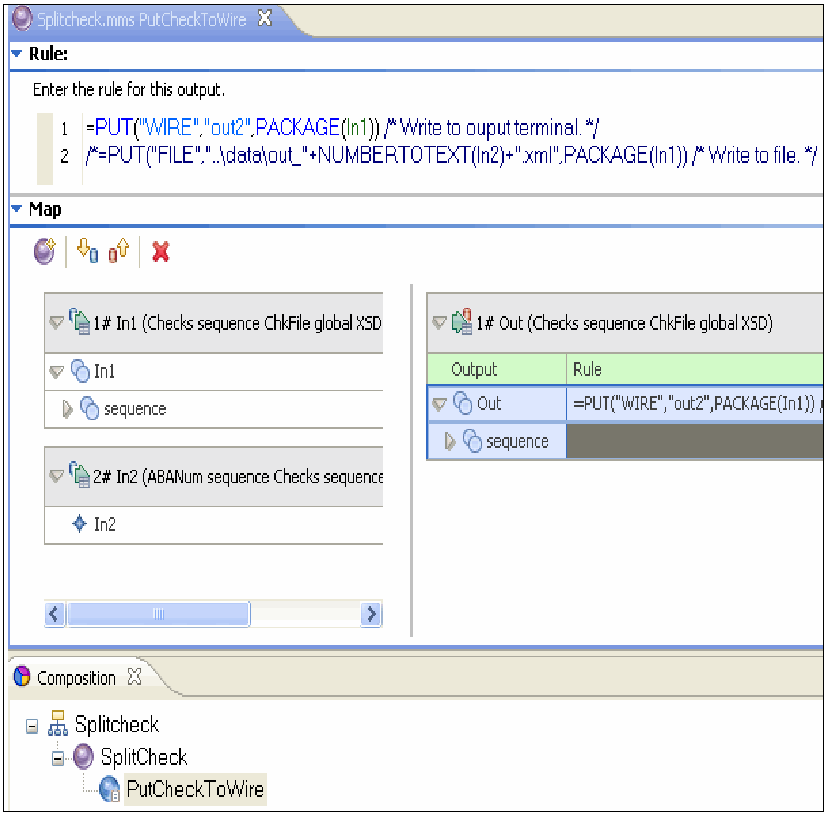


Figure 6-77 Using the PUT rule to pass each individual payment to the broker output terminal

When the map in Figure 6-77, executes it receives the xml message shown in Example 6-8.

Example 6-8 inbound xml payment message.

```
<?xml version="1.0" encoding="UTF-8"?>
<ChkFile>
<Checks>
<ABANum>183260101</ABANum>
<FromAcctNum>222225622010087</FromAcctNum>
<ChkNum>1</ChkNum>
<ChkAmt>800</ChkAmt>
<ChkDate>2011-05-05T00:00:00</ChkDate>
<ToAcctNum>660007045551212</ToAcctNum>
<Description80>Acct# 660007045551212 House Pmt</Description80>
</Checks>
<Checks>
<ABANum>111111111</ABANum>
<FromAcctNum>222222222222222</FromAcctNum>
<ChkNum>3333</ChkNum>
<ChkAmt>444.44</ChkAmt>
<ChkDate>2011-05-15T00:00:00</ChkDate>
<ToAcctNum>660678912345678</ToAcctNum>
<Description80>Description of 2nd check in list</Description80>
```

```
</Checks>
<Checks>
<ABANum>123456789</ABANum>
<FromAcctNum>333333333333</FromAcctNum>
<ChkNum>4011</ChkNum>
<ChkAmt>1000</ChkAmt>
<ChkDate>2011-05-25T00:00:00</ChkDate>
<ToAcctNum>660123456789012</ToAcctNum>
<Description80>Description of 3rd check in list</Description80>
</Checks>
</ChkFile>
```

It then splits the content of the message into individual messages, one for each payment, or Checks record. Example 6-9 shows the result of executing the map based on the input shown in Example 6-8.

Example 6-9 Split messages after map is executed.

```
Message 1
...........
<Checks>
<ABANum>183260101</ABANum>
<FromAcctNum>222225622010087</FromAcctNum>
<ChkNum>1</ChkNum>
<ChkAmt>800</ChkAmt>
<ChkDate>2011-05-05T00:00:00</ChkDate>
<ToAcctNum>660007045551212</ToAcctNum>
<Description80>Acct# 660007045551212 House Pmt</Description80>
</Checks>
...........
Message 2
...........
<Checks>
<ABANum>111111111</ABANum>
<FromAcctNum>222222222222222</FromAcctNum>
<ChkNum>3333</ChkNum>
<ChkAmt>444.44</ChkAmt>
<ChkDate>2011-05-15T00:00:00</ChkDate>
<ToAcctNum>660678912345678</ToAcctNum>
<Description80>Description of 2nd check in list</Description80>
</Checks>
...........
Message 3
...........
<Checks>
<ABANum>123456789</ABANum>
<FromAcctNum>333333333333</FromAcctNum>
<ChkNum>4011</ChkNum>
<ChkAmt>1000</ChkAmt>
<ChkDate>2011-05-25T00:00:00</ChkDate>
<ToAcctNum>660123456789012</ToAcctNum>
<Description80>Description of 3rd check in list</Description80>
</Checks>
...........
```

As each individual record is split out, it is passed to the output terminal of the broker map node that is named, by literal, in the map rule. The broker continues processing each message as it is output to the terminal, while the map continues to split out additional messages. This scenario shows each message passed to the same named output terminal as it is split. However, it is also possible to dynamically name the output terminal, if that functionality is required. The only requirement is that the output terminal named in the map rule, be defined for the map node. After the last message written to the output terminal, the map completes and broker continues processing.

6.7 Conclusion

In spite of its simplicity, the scenario described in this chapter illustrates two interesting integration aspects between IBM Sterling and IBM WebSphere portfolios.

The first aspect is the integration between WebSphere Transformation Extender Maps in Sterling B2B Integrator which is achievable in two ways, using the WTX Map service in a Business Process or using a WebSphere Translation Extender map in the proprietary Translation service.

The second integration aspect concerns the communication between Sterling B2B Integrator and a WebSphere Message Broker infrastructure through MQ, which is the recommended approach to implement such a communication.

In general, it is achievable because Sterling B2B Integrator includes the WebSphereMQ Suite, a group of services and a communications adapter that provide maximum flexibility and functionality by enabling you to script a complete MQ session using a business process.

The suite also includes an adapter that you can use for asynchronous receiving - the WebSphereMQ Async Receiver adapter.

Sterling B2B Integrator still includes the original WebSphereMQ adapter, which might serve your needs better if your MQ communications are fairly basic.

As stated above, MQ communication is the recommended practice to integrate Sterling Business Integration Suite with WebSphere Message Broker infrastructure. However, because the latest fix pack of WebSphere Message Broker v.7.0, it is also possible to establish a bridge with a dedicated Connect:Direct server that needs to be installed by the WebSphere Message Broker, in the same server. You then establish a connection with a Connect:Direct Server Adapter instance of Sterling B2B Integrator.

Lastly, because both Sterling B2B Integrator and WebSphere Message Broker support the most common communication protocols, like HTTP, HTTPS, or FTP, and now Connect:Direct, it is even possible to establish a connection using one of those protocols. However, this would not be a recommended practice because it won't make use of MQ whereas MQ is the preferred communication mean.

7

Supply Chain scenario using AS2 and EDI

This chapter provides an example scenario describing how to use the WebSphere DataPower B2B Appliance XB62 in conjunction with Sterling Integrator, WebSphere MQ, and WebSphere Transformation Extender to consume and process an AS2 packaged message that contains an EDI purchase order.

This chapter includes the following sections:

7.1 Business value

The EDIINT B2B pattern is commonly used in supply chain scenarios where companies want to securely exchange EDI data over the Internet. The Applicability Statements (AS1, AS2, AS3) provide a mechanism to securely transport data over a public network. They provide encryption, signatures, and identity of the sender and receiver as well as providing for non-repudiation of origin and receipt. Although EDIINT stands for EDI over the Internet, over the years it has evolved to be payload agnostic and is commonly used to envelope any payload data format. This pattern as depicted in this example demonstrates the ability of the B2B appliance to provide edge security in and to consume an AS2 message from a trading partner that contains an EDI payload. It further demonstrates the IBM capability to integrate to Sterling Integrator and WTX for downstream document processing.

The business value of using WebSphere DataPower in conjunction with both Sterling Integrator and WebSphere Transformation Extender to deliver this pattern is significant. Each product compliments the others, providing combined functionality encompassing a much broader range of B2B flow scenarios than could have been accomplished when used individually.

Next we provide is a brief description of the key value of each product:

- The IBM WebSphere® DataPower® B2B Appliance redefines the boundaries of middleware by extending the IBM SOA Foundation with a specialized, consumable, dedicated SOA appliance that combines business-to-business (B2B) standards, simplified integration, superior performance, and hardened security for SOA implementations. It is meticulously designed to augment all phases of the SOA life cycle and implementation.
- Sterling Integrator is a transaction engine and toolkit that allow the user to define, create, implement, and manage process flows. These flows, in turn, allow for the processing, routing, translation, and storage of high volumes of inbound and outbound messages or files. They can also interact and integrate with both internal systems and external business partners.
- WebSphere Transformation Extender WTX is the IBM universal transformer, designed to be the Enterprise Data Transformation standard. Its consistent data transformation and enrichment capabilities reduce application development and maintenance costs, standardize components, provide reuse of both assets and skill sets across the enterprise, and decrease the time to market of new applications.

7.2 Prerequisites: Technical and infrastructure

There are some prerequisites necessary in order to fully understand the scenario and to set it up successfully in your own infrastructure.

7.2.1 Software prerequisites

In order to be able to run this scenario, you must have installed the following components:

- WebSphere DataPower B2B Appliances XB62 (Appliance)
- ibm sterling B2B integrator
- WebSphere Transformation Extender for Integration Servers
- WebSphere Transformation Extender Design Studio
- WebSphere Transformation Extender Industry Pack for EDI-X12
- WebSphere MQ

7.2.2 Skills prerequisites

In order to fully implement and understand this scenario, you need to be familiar with:

- B2B Messaging (AS2) and EDI-X12 (basic knowledge)
- WebSphere DataPower B2B Appliances XB62 basic concepts (*IBM WebSphere DataPower B2B Appliance XB60 Revealed*, SG24-7745)
- IBM Sterling B2B Integrator basic concepts
- WebSphere MQ basic concepts
- WebSphere Transformation Extender basic mapping techniques

7.3 Presenting the scenario

This section provides an overview of the data flow for this scenario. For this particular scenario, we are implementing a two-way flow where we receive an AS2 message that corresponds with an incoming purchase order (X12-850 format). We send an AS2 message back to the partner that corresponds to a functional acknowledgement (EDI 997 format).

> **Tip:** It is important to point out that we are only demonstrating a single EDI transaction flow and its acknowledgement in this section. In most cases, a customer's supply chain pattern might contain multiple transactions that represent a full conversation. For example, a purchase order might be responded to with an advanced ship notice (X12-856) and an invoice (X12-810) that are related to the order that was received.

7.3.1 Primary scenario

As you can see in Figure 7-1, the scenario can be divided into two logical data flows:

- The inbound flow, where we receive the AS2 message containing the X12-850 purchase order from the trading partner
- The outbound flow, where we send the corresponding AS2 message containing the X12-997 functional acknowledgement to the trading partner that was generated by Sterling Integrator

Next we describe each step as the data flows through the XB62, pictured in Figure 7-1:

1. An EDI-850 payload is wrapped in an AS2 message envelope and sent to the B2B appliance sitting in the DMZ.
2. The B2B appliance receives the AS2 message, verifies the partner information, and unpackages the AS2 envelope.
3. The B2B Gateway routes the X12-850 to the MQ queue defined in the internal partner profile's destination setting.
4. After the X12-850 is successfully written to the MQ queue the B2B Gateway generates and sends an AS2 message disposition notification (MDN) back to the sending partner's system to inform it that the message was successfully received and the B2B Gateway marks the X12-850 transaction as complete.
5. Sterling Integrator picks up the X12-850.
6. Sterling Integrator validates its using WTX, generates an X12-997, routes the X12-997 to an out queue that is linked to the XB62 B2B Gateway.

7. Sterling Integrator uses a map created using the WTX X12 Industry pack to transform the X12-850 to XML.
8. The B2B Gateway picks up the X12-997 from the out queue.
9. The B2B Gateway identifies the sending and receiving partner from the ISA header, applies the attributes as defined in the receiving partner profile and wraps the X12-997 in an AS2 messaging envelope as defined in the external partner profile's destination setting.
10. The B2B Gateway routes the AS2 message carrying the X12-997 payload to the external trading partner's configured destination.
11. The external trading partner sends a message disposition notification (MDN) to the B2B Gateway to inform it that the message was successfully received.
12. The B2B Gateway logs the X12-997 transaction as complete.

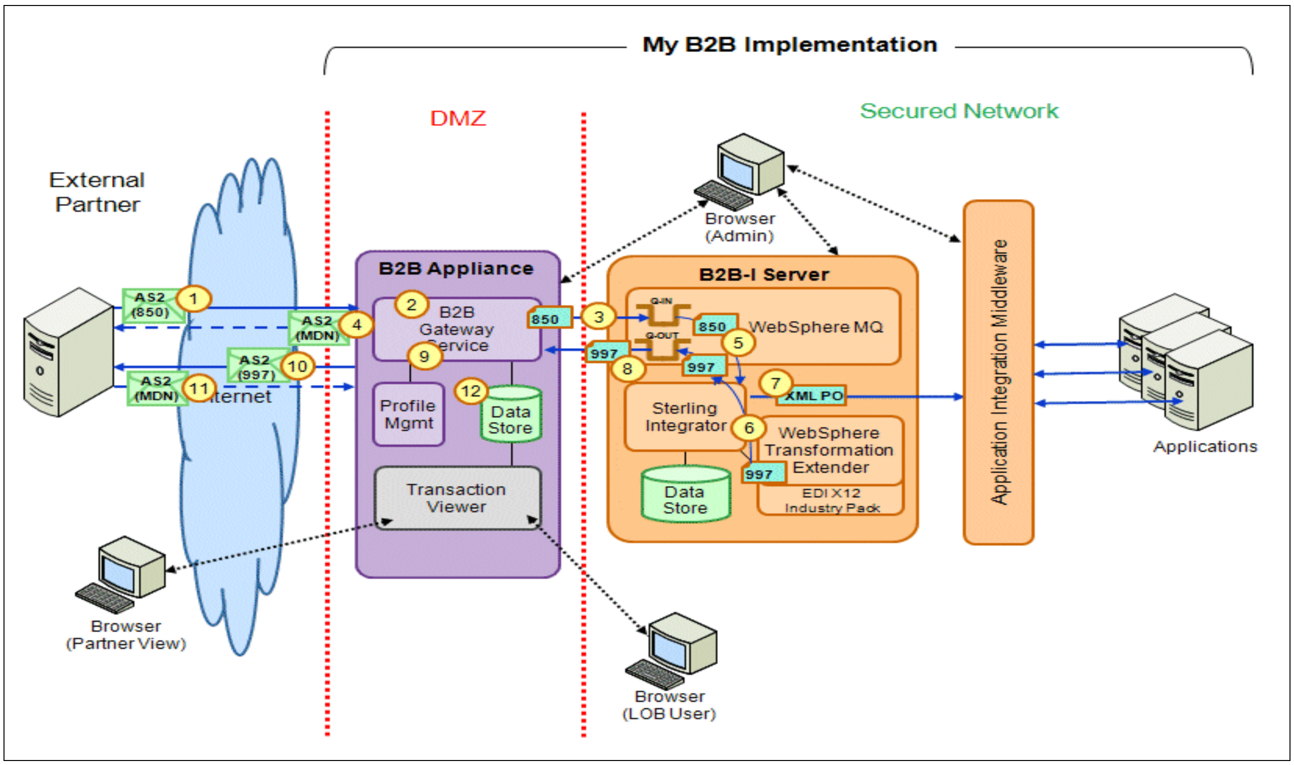


Figure 7-1 Supply Chain - purchase order data flow

7.3.2 Alternate scenarios

The primary scenario just described demonstrates how Sterling can integrate with IBM products to give the customer the most comprehensive capabilities possible to meet their most challenging B2B integration needs. However, not all customers need to support complex B2B flows and could potentially benefit from an alternative less comprehensive solution. This section lists some of the possible alternatives.

- If your company has no need to provide comprehensive security and B2B integration capabilities in the DMZ and only need to proxy data to Sterling Integrator, you can place Sterling Secure Proxy or a standard TCP/IP based proxy in the DMZ instead of DataPower.

- If your company standardized on WebSphere for Internal integration, you can use WebSphere DataPower in the DMZ and WebSphere Message Broker in conjunction with WebSphere Transformation Extender in the secured network to meet your needs.
- If your company standardized on Gentran Integration for EDI, you can use it in this scenario instead of WebSphere Transformation Extender.

7.4 Configuring the scenario

This section describes how to configure each product used in the primary scenario depicted in Figure 7-1 on page 264 as well as testing the both the inbound and outbound data flows.

The following artifacts are needed to support this configuration and can be downloaded using the instructions found in Appendix A, “Additional material” on page 333.

XB62 artifacts:

- `SCS_DataPowerArtifacts.zip` (contains DataPower Import file to fully configure the simulated partner, the X.509 certificates used in the profiles and the EDI test file).
- `SCS_WTXArtifacts1.zip` (contains the compliance artifacts needed to execute the X12 compliance maps. When extracted, the zip creates the directory structure, and puts all configuration files and map files in the proper location. The zip should be extracted at the root directory of the server that hosts the Sterling B2B Integrator).
- `SCS_WTXArtifacts2.zip` (contains the Project Interchange needed to create the Extender Project needed to build the maps needed for this scenario. From within WebSphere Transformation Extender Design Studio, import this zip as a Project Interchange).

7.4.1 Configuring the IBM WebSphere B2B Appliance XB62

This section contains the steps involved in configuring the WebSphere DataPower B2B Appliance XB62 to trade AS2 messages with your trading partners, More specifically, it covers these tasks:

- Import of the simulated Trading Partner B2B hub into the XB62
- Creation of an Application Domain to be used to house your B2B hub:
  - Creation of your internal profile, which is composed of your company's trading information and your keys and certificates.
  - Creation of the trading partner's external profile, which is composed of your partner's trading information and their public certificate.
  - Configuration of the XB60 B2B Gateway for trading AS2 messages:
    - Creation of an AS2 Front-side Protocol Handler to be used for receiving AS2 messages from trading partners
    - Creation of an HTTP Front-side Protocol Handler to be used for receiving files from the backend systems
    - Association of profiles with the B2B Gateway
    - Configuration of an data archive process to keep the system free of outdated transaction data

Installation: The installation of the XB62 is not covered in this book. For information about how to install DataPower Appliances, see the *DataPower Common Installation Guide*. For the purpose of this book, we configure a B2B Gateway in a single domain to act as your B2B hub. The Partner's B2B hub is pre-configured and is imported into its own Domain to allow us to simulate a Trading Partner without the need for another machine.

This book assumes that you are using your own XB62 and have Admin access to the device. All IP addresses, ports, queues, object names, and so on, reflected are for the purpose of this example only. Your actual configuration can differ and contain settings specific to your environment.

Dependencies: This configuration is dependent on a connection to an MQ Queue Manager for delivering files to Sterling Integrator. Be sure you have access to an MQ Queue Manager and the queues that you intend to use prior to configuring the MQ Queue Manager object in DataPower.

After you have completed this section, you can expect to have a better understanding of what it takes to configure the XB60 for AS2 communication.

Import the simulated trading partner domain

The DataPower XB62 has the ability to logically separate configurations by allowing multiple DataPower Domains to be configured on the device. We use this ability to create a simulated trading partner to trade AS2 with. Unlike traditional B2B Software, a separate instance and/or install of the B2B Gateway is not needed to simulate trading partners:

1. Launch your Internet browser and log on to the XB62's Web GUI as *Admin* into the *default* domain. You need Admin privileges to configure many of the objects.
2. From the console, click the `Import Configuration` link as shown in Figure 7-2.

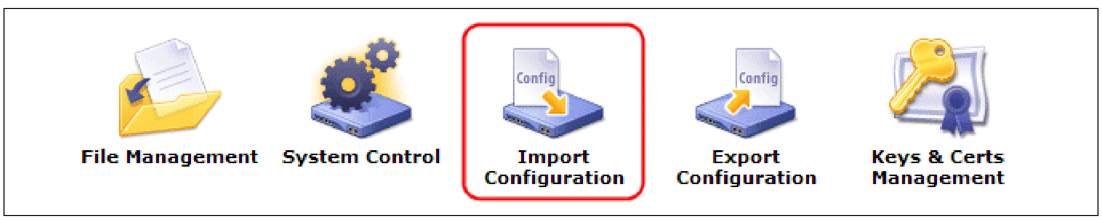


Figure 7-2 Import Configuration

3. In the Import Configuration view (Figure 7-3) be sure that the **ZIP** radio button is selected and click the **Choose File** button. Navigate to the directory where you placed the additional materials for this chapter, and select the `SCS_PARTNER_DOMAIN.zip` file. Click the **Next** button to continue.

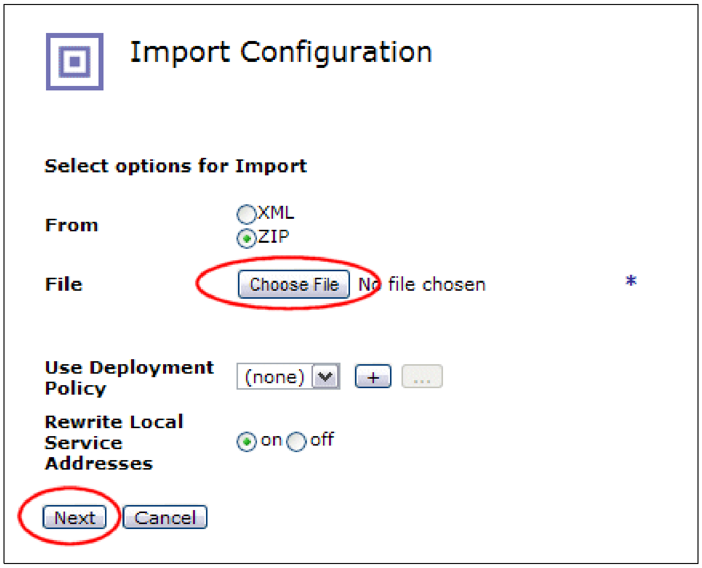


Figure 7-3 Import Configuration - Browse File

4. The Import Configuration view (Figure 7-4) displays information regarding the domain to be imported. Be sure to check the box next to the *Partner* domain and click **Next** to continue.

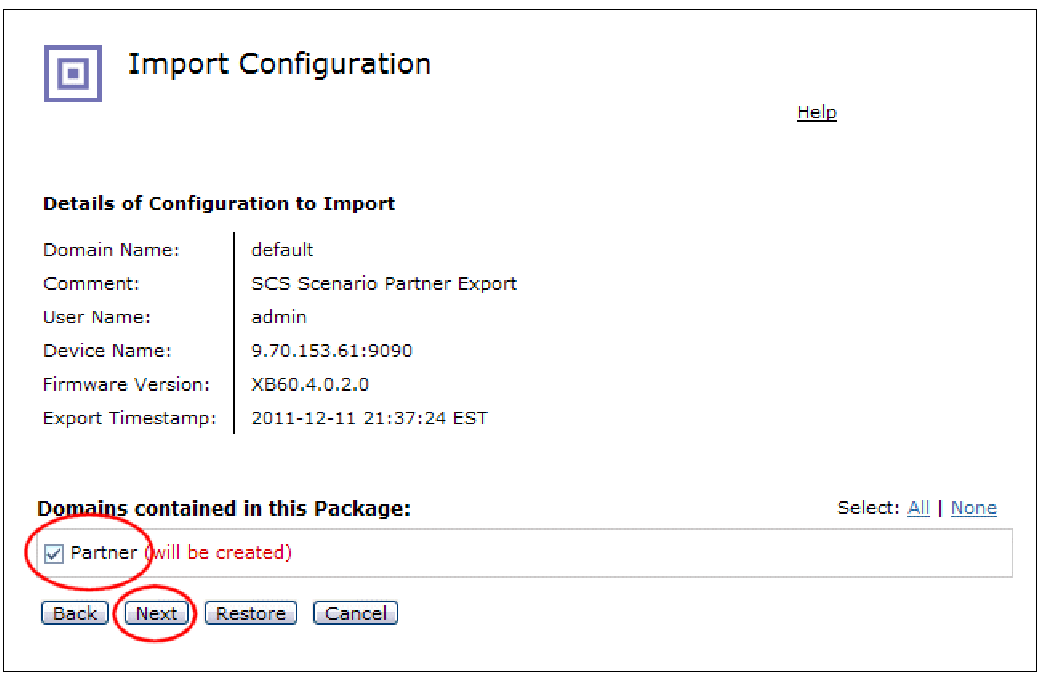


Figure 7-4 Import Configuration - Select Partner domain

5. The Import Configuration view displays a list of objects to be imported, a list of objects that already exist and a list of files that can differ from existing files. The objects that we need should already be checked, so click the **Import** button to continue.
6. The Import Configuration view displays the results of the import. Click the **Done** button to complete the import.
7. Click the `Save Config` link in the top menu bar just to the left of the Domain drop-down.

Configure the Partner Domain

DataPower has a built in security mechanism that prevents the user from having the ability to export keys and certificates from the device. Because DataPower does not allow the export of keys and certificates, we must associate the keys and certificates used for trading AS2 to the internal and external partner profiles associated with the Partner gateway.

> **Tip:** In the Partner's domain, your profile. mycompany, is an external profile. It is because the partner's B2B Gateway sees you as an external entity, and only your public certificates are available to the partner. Later in this chapter, you can create your own Domain where the partner profile is external and your profile is internal.
>
> If you are new to DataPower B2B, the concept of internal and external partner profiles might be confusing. For a good explanation of this concept, see section 7.1.1 in the book, *IBM WebSphere DataPower B2B Appliance XB60 Revealed*, SG24-7745.

Follow these steps:

1. Change into the **`Partner`** domain by clicking the drop-down next to *Domain* on the top menu bar. See Figure 7-5.

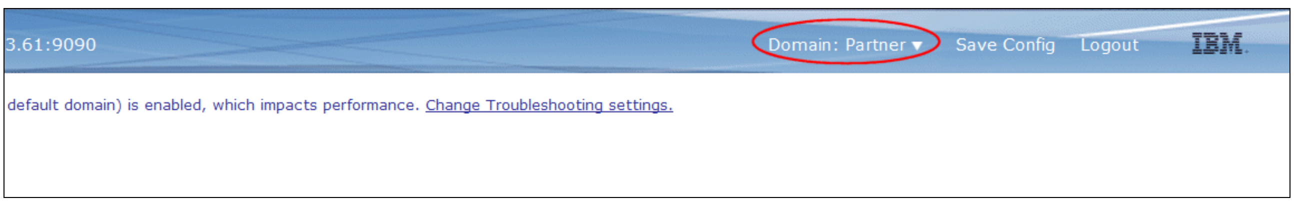


Figure 7-5 Change Domains

2. From the *Control Panel*, click the **`B2B Partner Profiles`** link as seen in Figure 7-6. The profile list as seen in Figure 7-7 shows both the internal and external profiles are down.

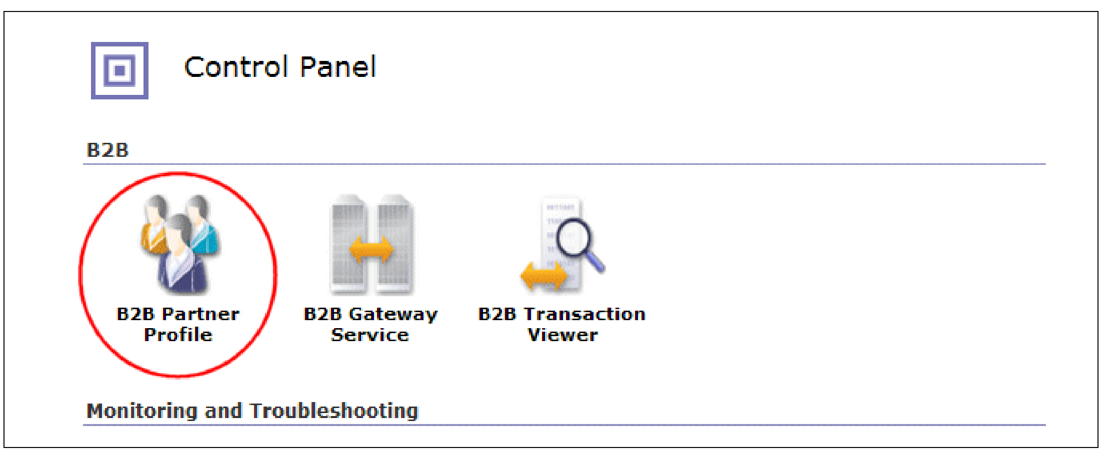
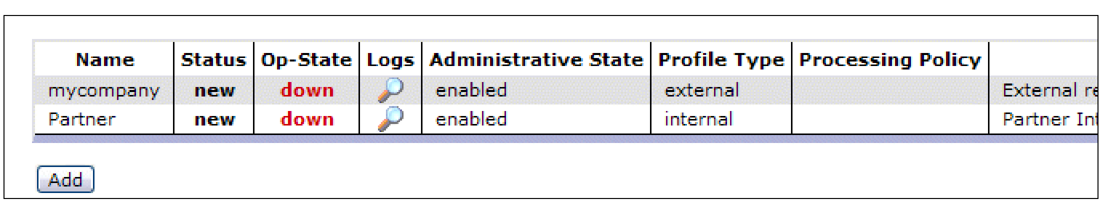


Figure 7-6 Control Panel - Profile Link



Name	Status	Op-State	Logs	Administrative State	Profile Type	Processing Policy	
mycompany	new	down		enabled	external		External re
Partner	new	down		enabled	internal		Partner Int

Add

Figure 7-7 Profile List View - down

3. Click **mycompany** in the *Name* column to edit the profile.

   c. In the Configure B2B Partner Profile view click the **AS Settings** tab.

   d. In the Inbound Security section, click the **Edit (...)** button next to the Inbound Signature Validation Credential field. See Figure 7-8.

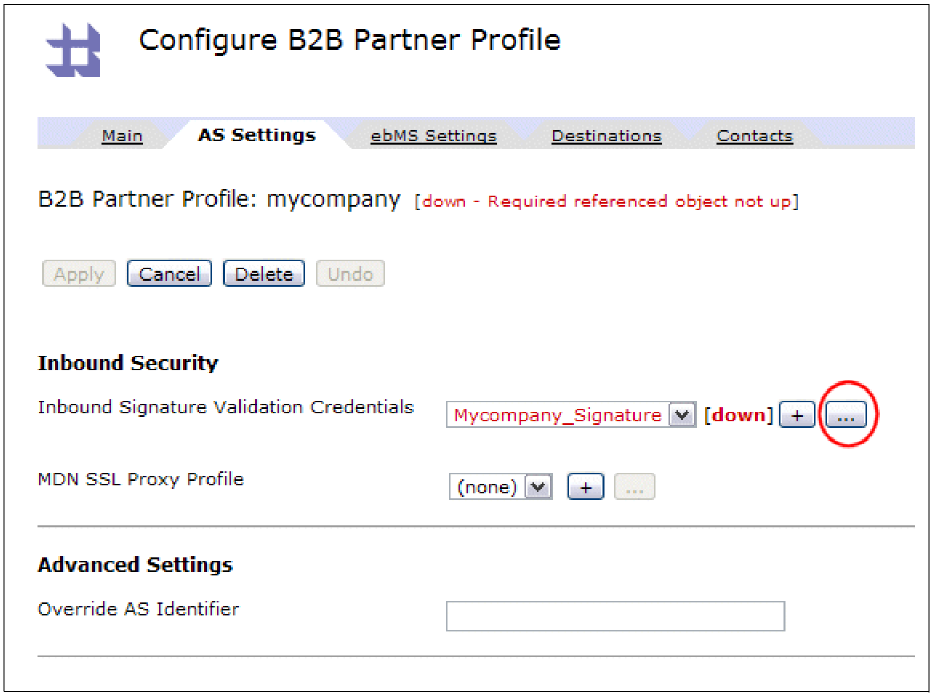


Figure 7-8 Configure mycompany AS Settings

   e. In the Configure Crypto Validation Credentials, view, click the `pencil` just to the right of Mycompany_Cert in the Certificates field.

   f. In the Configure Crypto Certificate view, click the **Upload** button next to the File Name field.

   g. In the Upload File to Directory cert:/// view, click **Choose File** just under the File to Upload field. Navigate to the location where you unzipped the `SCS_DataPowerArtifacts.zip` file, select the `mycompany-sscert.pem` file and click **Open**.

   h. Be sure **Overwrite Existing File** is checked, then click the **Upload** button at the bottom of the view.

   i. Click **Continue** in the upload success window. It puts you back into the Configure Crypto Certificate view.

   j. Click the **Apply** button in the Configure Crypto Certificate view. It puts you back into the Configure Crypto Validation Credentials view.

   k. Click the **Apply** button in the Configure Crypto Validation Credentials view. It puts you back into the **AS Settings** tab of the Configure B2B Partner Profile view.

   l. Click the **Apply** button in the Configure B2B Partner Profile view. It puts the mycompany profile in an Up state.

   m. Click **Save Config** on the top menu bar, just to the right of the Domain drop-down.

4. To quickly return to the profile list view, click the **B2B Partner Profile** portion of the Configure B2B Partner Profile at the top of the view, as shown in Figure 7-9.

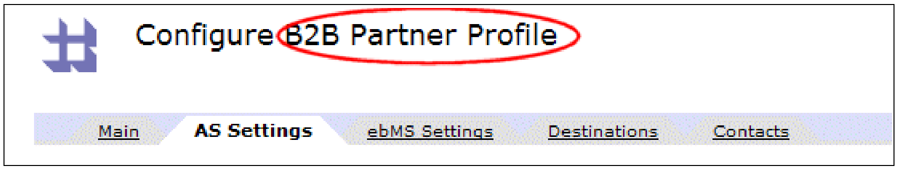


Figure 7-9 Return to the profile list view

5. Click the **Partner** profile to edit it:
   a. In the Configure B2B Partner Profile view, click the **AS Settings** tab.
   b. In the Inbound Security section, leave the Require Signature and Require Encryption boxes unchecked and click the **edit (...)** button next to Inbound Decryption Identification Credential field. See Figure 7-10.

Figure 7-10 Configure Partner AS settings

   c. In the Configure Crypto Identification Credentials view, click the **edit (...)** button next to the Crypto Key field.
   d. In the Configure Crypto Key view, click the **Upload** button next to the File Name field.
   e. In the Upload File to Directory cert:/// view, click **Choose File** under the File to Upload field. Navigate to the location where you unzipped the `SCS_DataPowerArtifacts.zip` file, select the `partner-privkey.pem` file, and click **Open**.
   f. Be sure **Overwrite Existing File** is checked, then click the **Upload** button at the bottom of the view.
   g. Click **Continue** in the upload success window. It puts you back into the Configure Crypto Key view.
   h. Click the **Apply** button in the Configure Crypto Key view. It puts you back into the Configure Crypto Identification Credentials view.

i. In the Configure Crypto Identification Credentials view, click the **edit (...)** button next to the Certificate field.
j. In the Configure Crypto Certificate view, click the **Upload** button next to the File Name field.
k. In the Upload File to Directory cert:/// view, click **Choose File** under the File to Upload field. Navigate to the location where you unzipped `SCS_DataPowerArtifacts.zip`, select the `partner-sscert.pem` file, and click **Open**.
l. Be sure Overwrite Existing File is checked, then click the **Upload** button at the bottom of the view.
m. Click **Continue** in the upload success window. It puts you back into the Configure Crypto Certificate view.
n. Click the **Apply** button in the Configure Crypto Certificate view. It puts you back into the Configure Crypto Identification Credentials view.
o. Click the **Apply** button in the Configure Crypto Identification Credentials view. It puts you back into the AS Settings tab of the Configure B2B Partner Profile view.
p. Click the **Apply** button in the Configure B2B Partner Profile view. It puts the Partner profile in an Up state.
q. Click **Save Config** on the top menu bar, just to the right of the Domain drop-down.

Next: You have completed the import and configuration of your simulated trading partner. You are now ready to create your application domain to be used to house your profiles and your B2B Gateway service.

Create your Application Domain

In this section you are creating a domain to be used to house the configuration needed to support your B2B flows.

Tip: If you already understand how to configure the DataPower XB62 for connecting to your AS2 trading partners, you can bypass this section by importing `SCS_MYDOMAIN_DOMAIN.zip` from the location where you unzipped `SCS_DataPowerArtifacts.zip`. Be sure to import the appropriate certificates in the AS Settings tabs for both profiles, just like you did for the Partner domain.

Follow these steps:

1. Change back into the default domain by clicking the drop-down next to Domain on the top menu bar.
2. In the left navigation menu, search on the word “Application” and select **Application Domain** as seen in Figure 7-11.

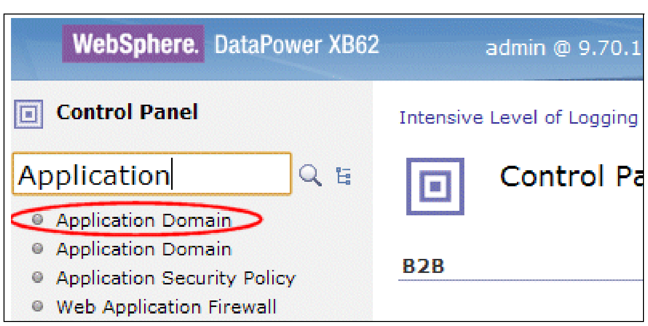


Figure 7-11 Left navigation menu - search for Application Domain

3. In the Configure Application Domain view, **Main** tab, enter a descriptive name in the Name field. In this example, we used MyDomain.
4. Check the Enable Auditing and Enable Logging boxes at the bottom of the view and click the **Apply** button to create the domain.
5. Click **Save Config** on the top menu bar, just to the right of the Domain drop-down.
6. Switch into your newly created domain by clicking the drop-down next to Domain in the top menu bar and selecting MyDomain.

Tip: From this point forward, all of your configuration is done in MYDOMAIN. For the purpose of this lab, we stay logged in as admin, however, in a real world deployment, you normally create a User ID and assign appropriate rights to the user that enables them to create and configure objects in this domain, restricting access to the default domain to only the primary Admin user.

Create an MQ Manager Object

Follow these steps:

1. In the left navigation menu, search on MQ and select MQ Queue Manager.
2. In the Configure MQ Queue Manager list view click in the Add button to create a Queue Manager Object.
3. Configure the **Main** tab as described in the following steps. See Figure 7-12.
   a. In the Name field, enter a descriptive name for the MQ Queue Manager object; in this example we used B2B02.
   b. Check **enabled** in the Administrative State field.
   c. Optionally add comments that describe this MQ Queue Manager object.
   d. In the Host Name field, enter the IP address or host name and port of the MQ server you are connecting to; in this example we used 9.42.170.226:1414.
   e. In the Queue Manager Name field, enter the name of the MQ Queue Manager your are connecting to; in this example we used B2B02.
   f. In the Channel Name field, enter the name of the MQ connection channel you are connecting to; in this example we used B2B.SUPPLYCHAIN.DP.
   g. Enter the username used to connect to the channel that is being used. In this example, we used an internal name that has the appropriate rights on the MQ Server.

Tip: The best practice for connecting to MQ is to leave the user empty in the DataPower config, and to insert a local technical user with limited access in the MQ config. Authorization is managed by the rights of this user and Authentication would best be done through SSL.

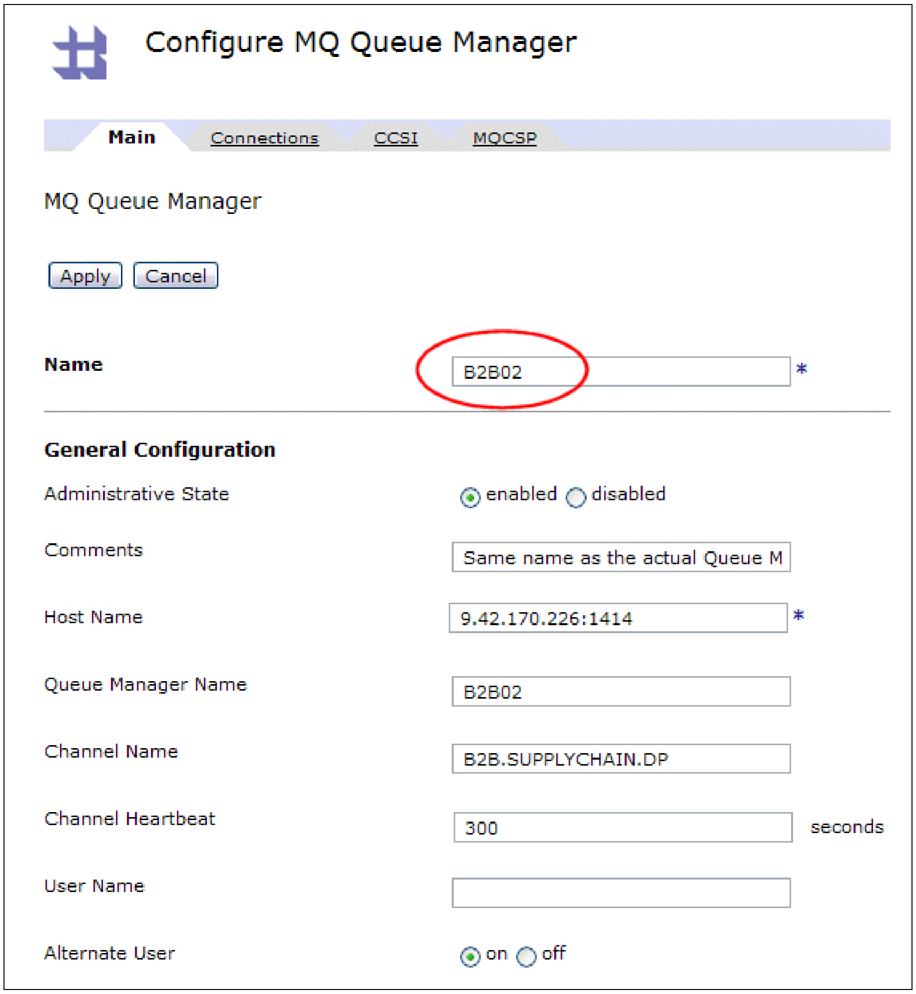


Figure 7-12 MQ Queue Manager Configuration - Main tab

4. Configure the **Connections** tab as described in the following steps. See Figure 7-13.
   a. Click the **Connections** tab; the Name field carries over to the Connections view; do not change it.
   b. In the Retry Behavior section, change the retry settings as described here:
      i. Be sure the Automatic Retry is set to **on**.
      ii. Set the Retry Interval to 60 seconds.
      iii. Set the Retry Attempts to 3 attempts.
      iv. Set the Long Retry Interval to 600 seconds
      v. Set the Reporting Interval to 300 seconds.
      vi. Leave the default values in all remaining fields.

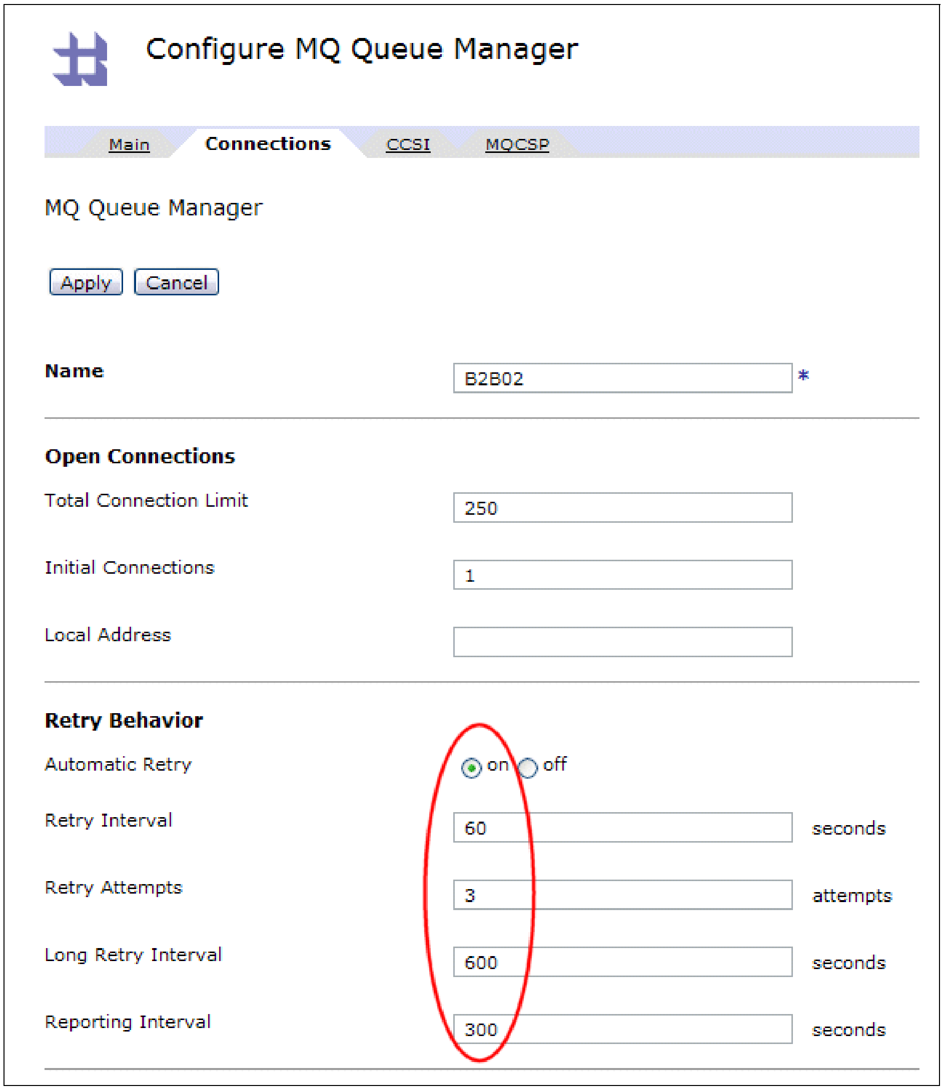


Figure 7-13 MQ Queue Manager Configuration - Connection tab

5. Click the **Apply** button at the top left of the Configure MQ Queue Manager view to complete the MQ Queue Manager configuration.
6. Click **Save Config** on the top menu bar, just to the right of the Domain drop-down.

> **Tip:** If DataPower can connect to your queue manager, it is in an Up state. Otherwise, if it is in a Down state, you cannot write the files to the MQ backend as demonstrated in this example until you resolve the connectivity issue.

> **Next:** You have completed the configuration of a MQ Queue Manager object to be used in our example as the back-side integration exchange point. You are now ready to create your internal partner profile.

Create your internal profile

In this section, you are creating your company's profile to be used for identifying yourself to your external trading partner, for applying trading partner policy or agreement attributes to the connection, and for integrating to the backend system; WebSphere MQ in this example:

1. Click the **B2B Partner Profile** icon in the Control Panel view.
2. In the Configure B2B Partner Profile list view, click the **Add** button to create your internal profile.
3. Configure the **Main** tab as described in the following steps. See Figure 7-14.
   a. In the Name field, enter a descriptive name for your Internal Profile; in this example we used mycompany.
   b. Choose **enabled** in the Administrative State field.
   c. Optionally add comments that describe this profile.
   d. Choose **Internal** in the Profile Type field.
   e. In the Partner Business IDs field, enter your IDs; in this example, we used ZZMYCOMPANY, which matches the EDI file that we test later in this chapter. Type in the ID and click the **Add** button.
   f. Leave all remaining fields set to the default values.

Figure 7-14 Configure internal partner profile - Main tab

> **Tip:** Do not click the **Apply** button at this point. We still have some required configuration needed to complete this profile.

4. Configure the *AS Settings* tab as described in the following steps:

   a. Click the **AS Settings** tab. The Name field carries over to the AS Settings view. Do not change it.

   b. In the Inbound Security section, leave the Require Signature and Require Encryption boxes unchecked and click the **plus sign** icon to create a new Inbound Decryption Identification Credential. See Figure 7-15.

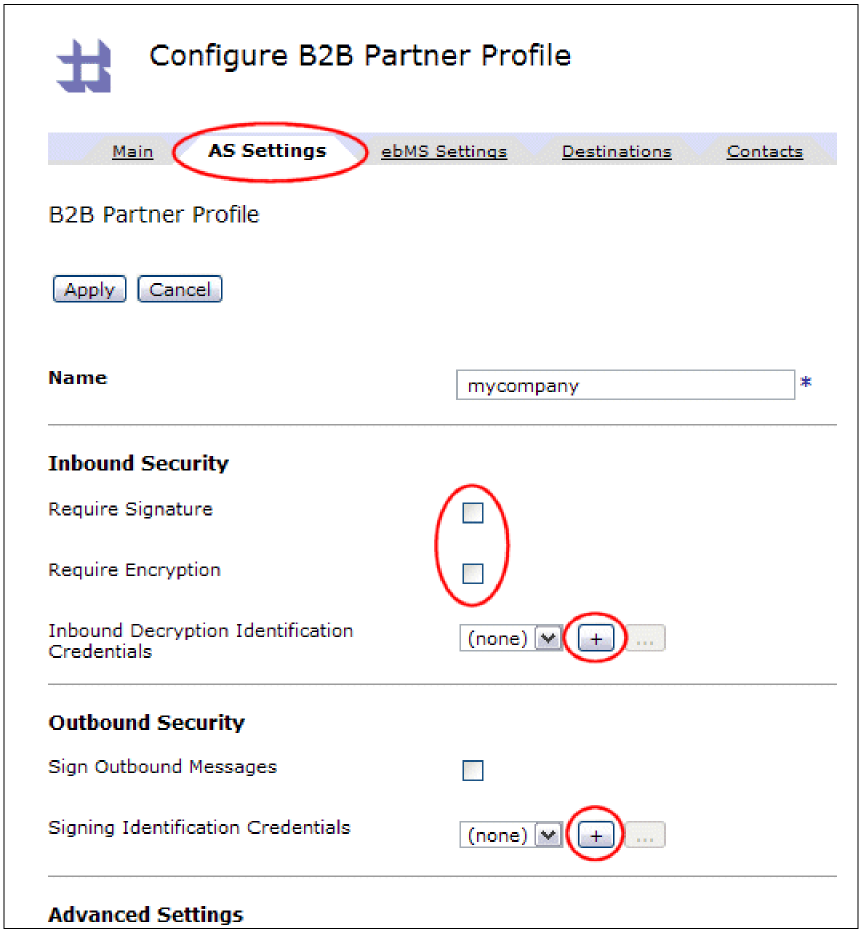


Figure 7-15 Configure internal partner profile - AS Settings tab

   c. In the Configure Crypto Identification Credentials view Name field, enter a descriptive name for this credential. In this example, we used mycompany_decrypt. See Figure 7-16.

   d. Choose **enabled** in the Admin State field.

   e. Click the **plus sign** icon next to the Crypto Key field to create/upload the Crypto Key.

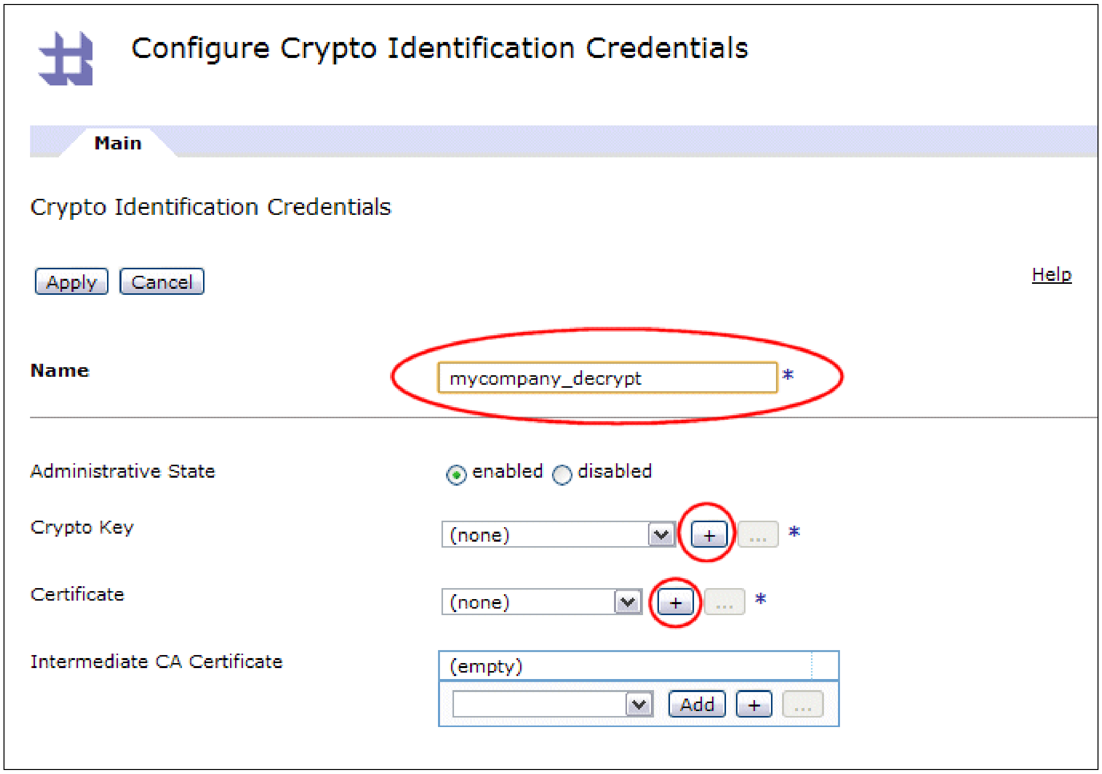


Figure 7-16 Configure Crypto Identification Credentials

f. In the Configure Crypto Key view Name field, enter a descriptive name for this key. In this example, we used mycompany_privkey.

   i. Choose **enabled** in the Admin State field.

   ii. Click the **Upload** button in the File Name field.

   iii. In the Upload File to Directory cert:/// view, click **Choose File** under the File to Upload field. Navigate to the location where you unzipped the `SCS_DataPowerArtifacts.zip` file, select the `mycompany-privkey.pem` file and click **Open**.

   iv. Be sure Overwrite Existing File is checked, then click the **Upload** button at the bottom of the view.

   v. Click **Continue** in the upload success window. It puts you back into the Configure Crypto Key view.

   vi. Enter the password used for the mycompany-privkey.pem file in both fields. In this example, the key password is “datapower”.

   vii. Click the **Apply** button in the Configure Crypto Key view. It puts you back into the Configure Crypto Identification Credentials view.

g. Now that we are back in Configure Crypto Identification Credentials view, we need to upload the certificate that is associated with the key. Click the **plus sign** icon next to the Certificate field to create/upload the Certificate. See Figure 7-16.

h. In the Configure Crypto Certificate view, Name field, enter a descriptive name for this cert. In this example, we used “mycompany_cert”.

   i. Choose **enabled** in the Admin State field.

   ii. Click the **Upload** button in the File Name field.

iii. In the Upload File to Directory cert:/// view, click **Choose File** under the File to Upload field and navigate to the location where you unzipped the `SCS_DataPowerArtifacts.zip`, select the `mycompany-sscert.pem` file and click **Open**.

iv. Be sure Overwrite Existing File is checked then click the **Upload** button at the bottom of the view.

v. Click **Continue** in the upload success window. It puts you back into the Configure Crypto Certificate view.

vi. Leave the Password fields blank (they are not needed for the certificate) and take the defaults for everything else.

vii. Click the **Apply** button in the Configure Crypto Certificate view. It puts you back into the Configure Crypto Identification Credentials view.

i. In the Configure Crypto Identification Credentials view, leave the Intermediate CA Certificate field empty because we are using Self-Signed Certificates.

j. After both credentials are configured, click the **Apply** button, which puts you back into the AS Settings view.

k. In the Outbound Security section be sure the Sign Outbound Messages box is checked and click the **plus sign** button to create a new Signing Identification Credential. See Figure 7-17.

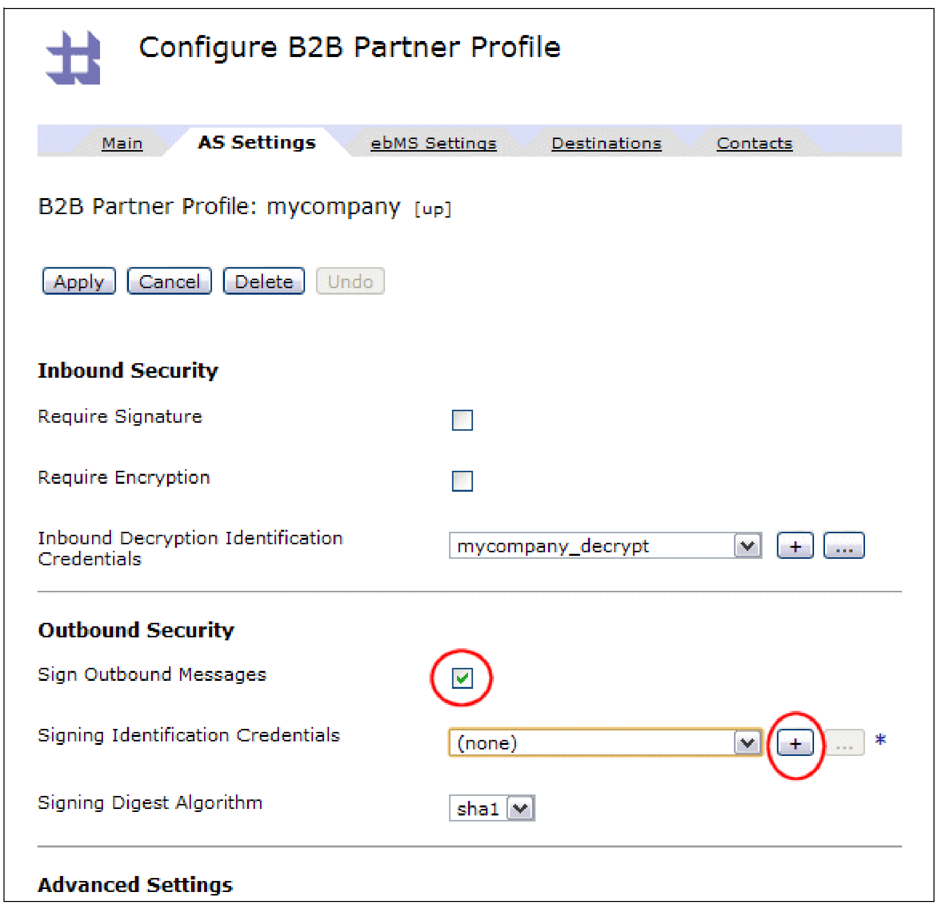


Figure 7-17 Outbound Security - Signing Identification Credentials

l. In the Configure Crypto Identification Credentials view, Name field, enter a descriptive name for this credential; in this example we used mycompany_signature_cred.

   i. Choose **enabled** in the Admin State field.

   ii. Because we using the same key to decrypt and sign we can use the key and cert that we already imported, click the drop-down next to the Crypto Key and the Certificate fields and select the same credentials we used for inbound security.

   iii. In the Configure Crypto Identification Credentials screen, leave the Intermediate CA Certificate field empty because we are using Self-Signed Certificates.

   iv. After both credentials are configured as seen in Figure 7-18, click the **Apply** button, which puts you back into the AS Settings screen.

   v. In the AS Settings view, leave all of the remaining fields set to the default values.

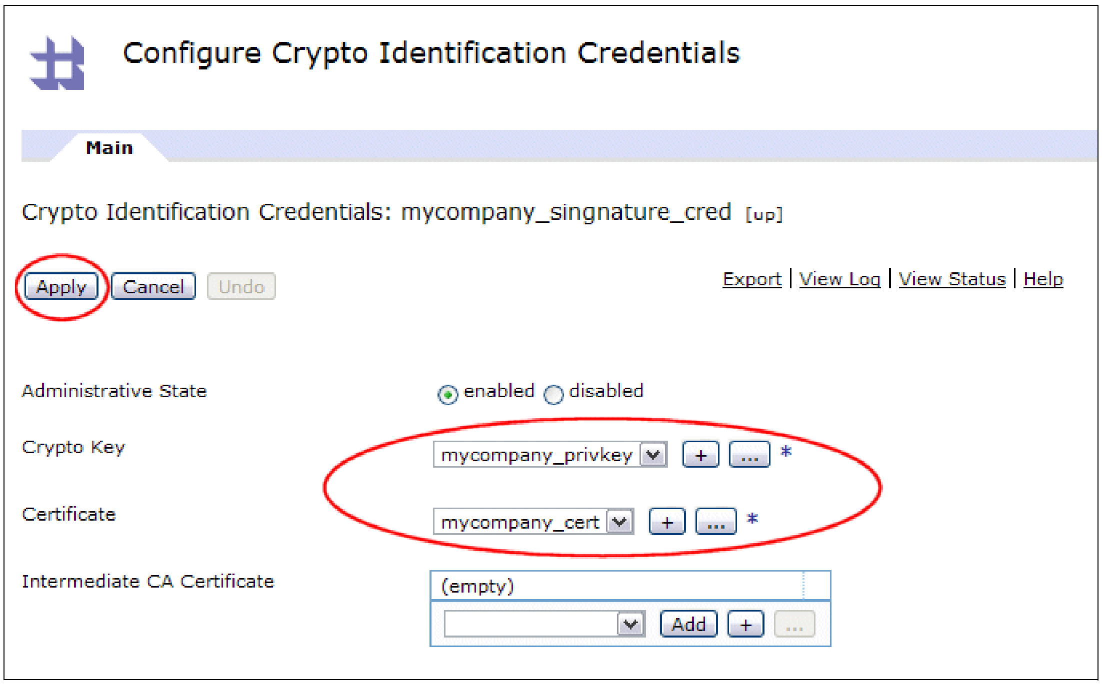


Figure 7-18 Outbound Security - Configure Crypto Identification Credentials - completed

> **Tip:** Do not click the **Apply** button at this point. We still have some required configuration needed to complete this profile.

5. Configure the **Destinations** tab as described in the following steps:

   a. Click the **Destinations** tab. The Name field carries over to the Destinations view. Do not change it. See Figure 7-19.

   b. In the Destinations section, click the **Add** button to add a destination to this profile. Because this profile is an internal profile the destination is a system or application inside the private network. For this example, we are going to use MQ as an Integration exchange point.

   c. Enter a descriptive name in the Destination Name field; in this example we used “mycompany_MQ_B2B02”.

   d. Be sure all of the boxes are checked under the Enabled Document Type section.

e. In the Connection Section, Destination URL drop-down, select dpmq:// and enter "`B2B02/?RequestQueue=SUPPLY.B2B.REQUEST`".

f. In the Destination Box, click the **Apply** button at the bottom left to save the destination to the list.

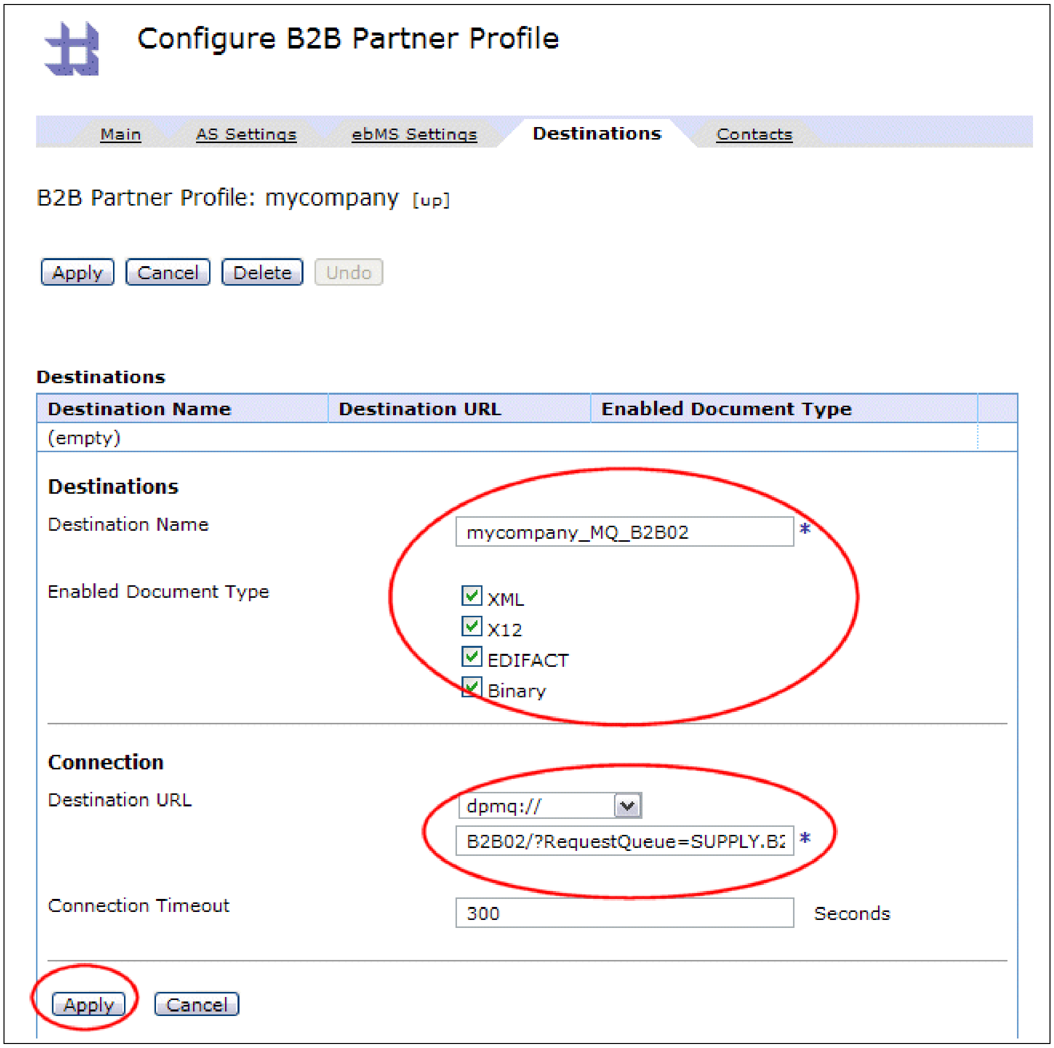


Figure 7-19 Mycompany Internal Profile - Destination Tab

6. Optionally configure the **Partner Profile Contacts** tab.
7. Click the **Apply** button in the top left corner of the Configure B2B Partner Profile view to save your internal profile.
8. Click **Save Config** on the top menu bar, just to the right of the Domain drop-down.

Next: You have completed the configuration of your internal profile, you are now ready to create the partner's external profile.

Create the Partner's external profile

In this section, you are creating your trading partner's profile to be used for verifying their identity, for applying trading partner policy or agreement attributes to the connection and for integrating to your external trading partner over a specified protocol, AS2 in this example.

1. If you are not already in the B2B Partner Profile list view, click the **B2B Partner Profile** icon in the Control Panel view.
2. In the Configure B2B Partner Profile list view click the **Add** button to create your partner's external profile.
3. Configure the **Main** tab as described in the following steps. See Figure 7-20.
   a. In the Name field, enter a descriptive name for your Internal Profile. In this example, we used "partner".
   b. Choose **enabled** in the Administrative State field.
   c. Optionally add comments that describe this profile.
   d. Choose **External** in the Profile Type field.
   e. In the Partner Business IDs field, enter your IDs; in this example we used ZZYOUR COMPANY which matches the EDI file that we test later in this chapter, type in the ID and click the **Add** button.
   f. Leave all remaining fields set to the default values.

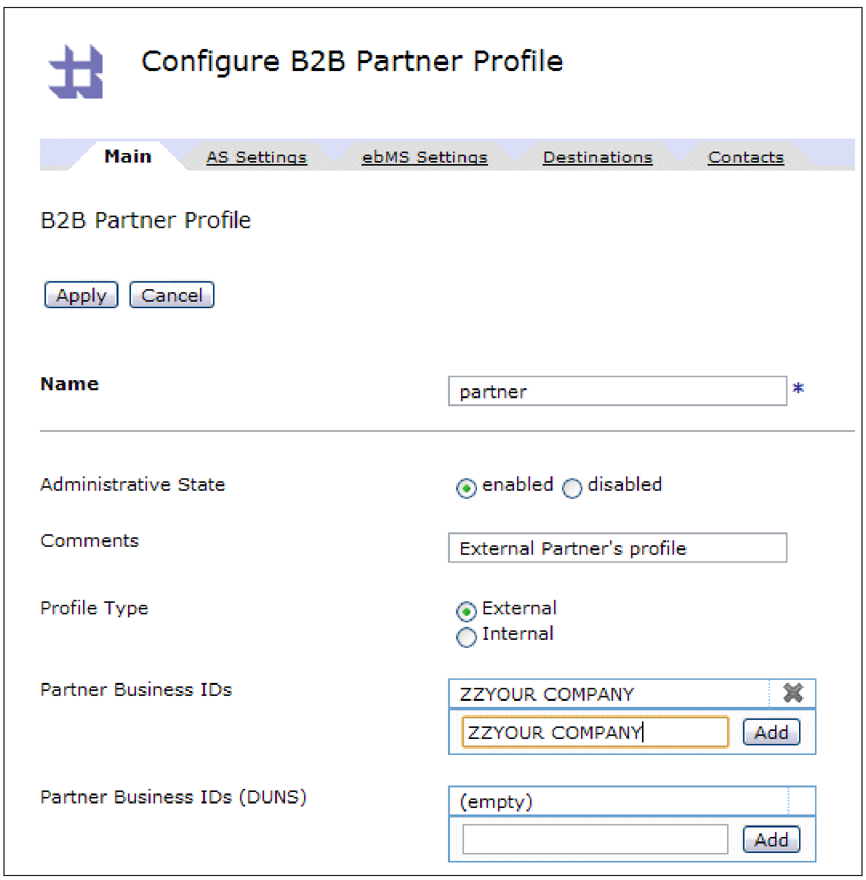


Figure 7-20 Configure external partner profile - Main tab

> **Tip:** Do not click the **Apply** button at this point. We still have some required configuration needed to complete this profile.

4. Configure the AS Settings tab as described in the following steps.
   a. Click the **AS Settings** tab; the Name field carries over to the AS Settings view, do not change it.
   b. In the Inbound Security section, click the **plus sign** button to create a new Inbound Signature Validation Credential. See Figure 7-21.

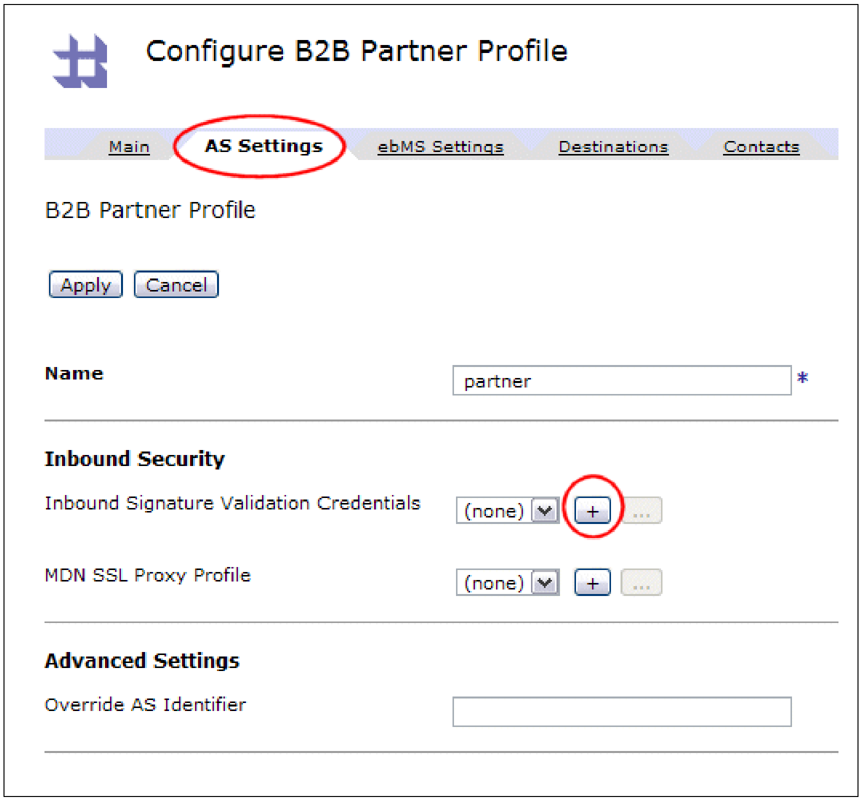


Figure 7-21 Configure external partner profile - AS Settings tab

   c. In the Configure Crypto Validation Credentials view, Name field, enter a descriptive name for this credential. In this example, we used “partner_signval”. See Figure 7-22 on page 283.
   d. Choose **enabled** in the Admin State field.
   e. Click the **plus sign** button next to the **Add** button in the Certificates field to create/upload the partner’s certificate.
   f. In the Configure Crypto Certificate view, enter a descriptive name for the crypto certificate. In this example, we used “partner_cert”.
   g. Click the **Upload** button next to the File Name field.
   h. In the Upload File to Directory cert:/// view, click **Choose File** just under the File to Upload field. Navigate to the location where you unzipped the `SCS_DataPowerArtifacts.zip` file, select the `partner-sscert.pem` file, and click **Open**.
   i. Be sure Overwrite Existing File is checked then click the **Upload** button at the bottom of the view.
   j. Click **Continue** in the upload success window. It puts you back into the Configure Crypto Certificate view.
   k. Leave the password fields blank, public certs do not need passwords.

l. Click the **Apply** button in the Configure Crypto Certificate view. It puts you back into the Configure Crypto Validation Credentials view.

m. Set Use CRL to **Off**.

n. Click the **Apply** button in the Configure Crypto Validation Credentials view. It puts you back into the **AS Settings** tab of the Configure B2B Partner Profile view.

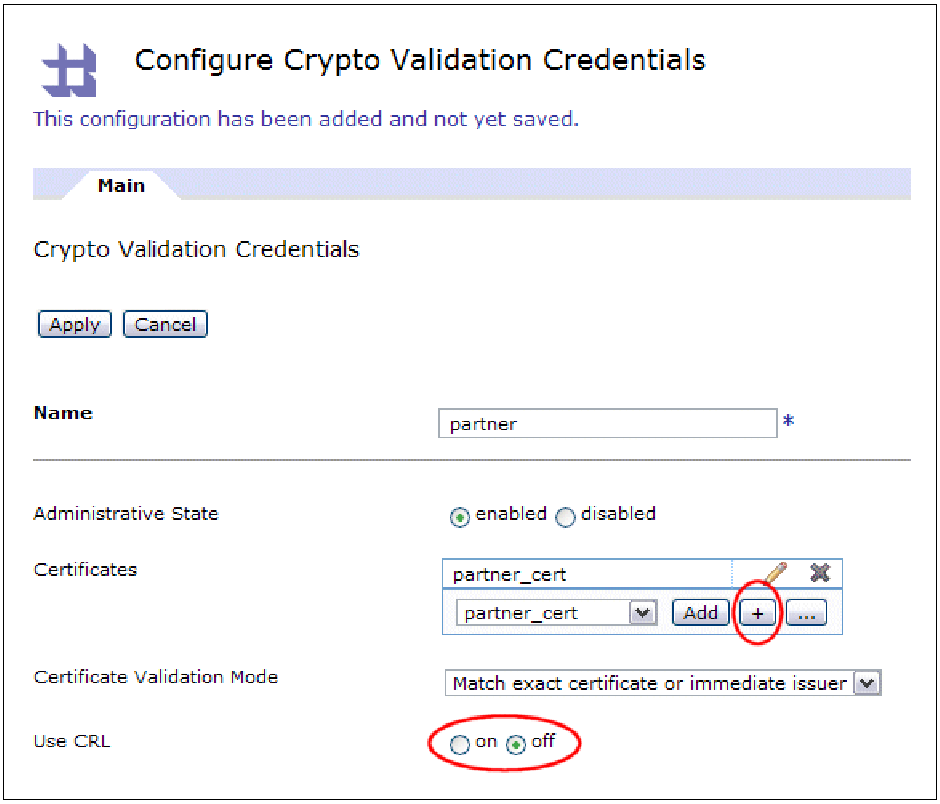


Figure 7-22 Inbound Security - Configure Crypto Validation Credentials

> **Tip:** Do not click the **Apply** button at this point. We still have some required configuration needed to complete this profile.

5. Configure the **Destinations** tab as described in the following steps:
   a. Click the **Destinations** tab; the Name field carries over to the Destinations view. Do not change it. See Figure 7-23 and Figure 7-24.
   b. In the Destinations section, click the **Add** button to add a destination to this profile. Because this profile is an external profile the destination is the trading partner's URL, For this example we are going to use AS2 as the preferred method of exchanging files.
   c. Enter a descriptive name in the Destination Name field. In this example, we used "Partner_AS2_31001".
   d. Be sure all of the boxes are checked under the Enabled Document Type section.
   e. In the Connection Section, Destination URL drop-down select AS2:// and enter the IP address or host name of your XB62. For this example, we used 127.0.0.1:31001. Also change the Connection Timeout to 120 seconds.
   f. In the AS Outbound Security section check the box next to Encrypt Messages, use the drop-down to select partner_cert and use the default value for Encryption Algorithm.
   g. In the Advanced AS Behavior section, check the box next to Request MDN, change the Time to Acknowledge to 120 and check the box next to Request Signed MDN.
   h. Take the default values for all other fields.
   i. In the Destination Box, click the **Apply** button at the bottom left to save the destination to the list.

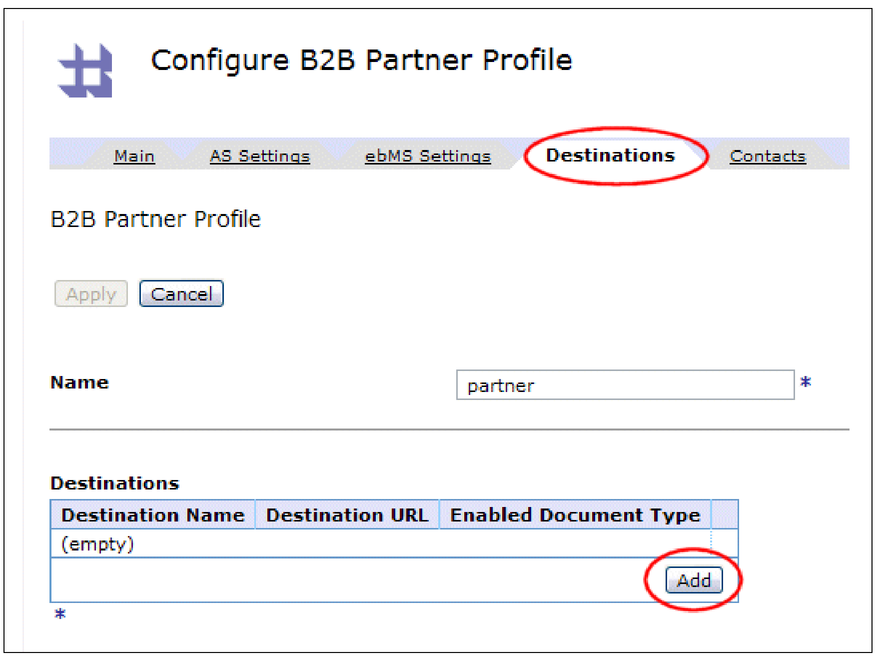


Figure 7-23 Mycompany External Profile - Configure Destination Tab

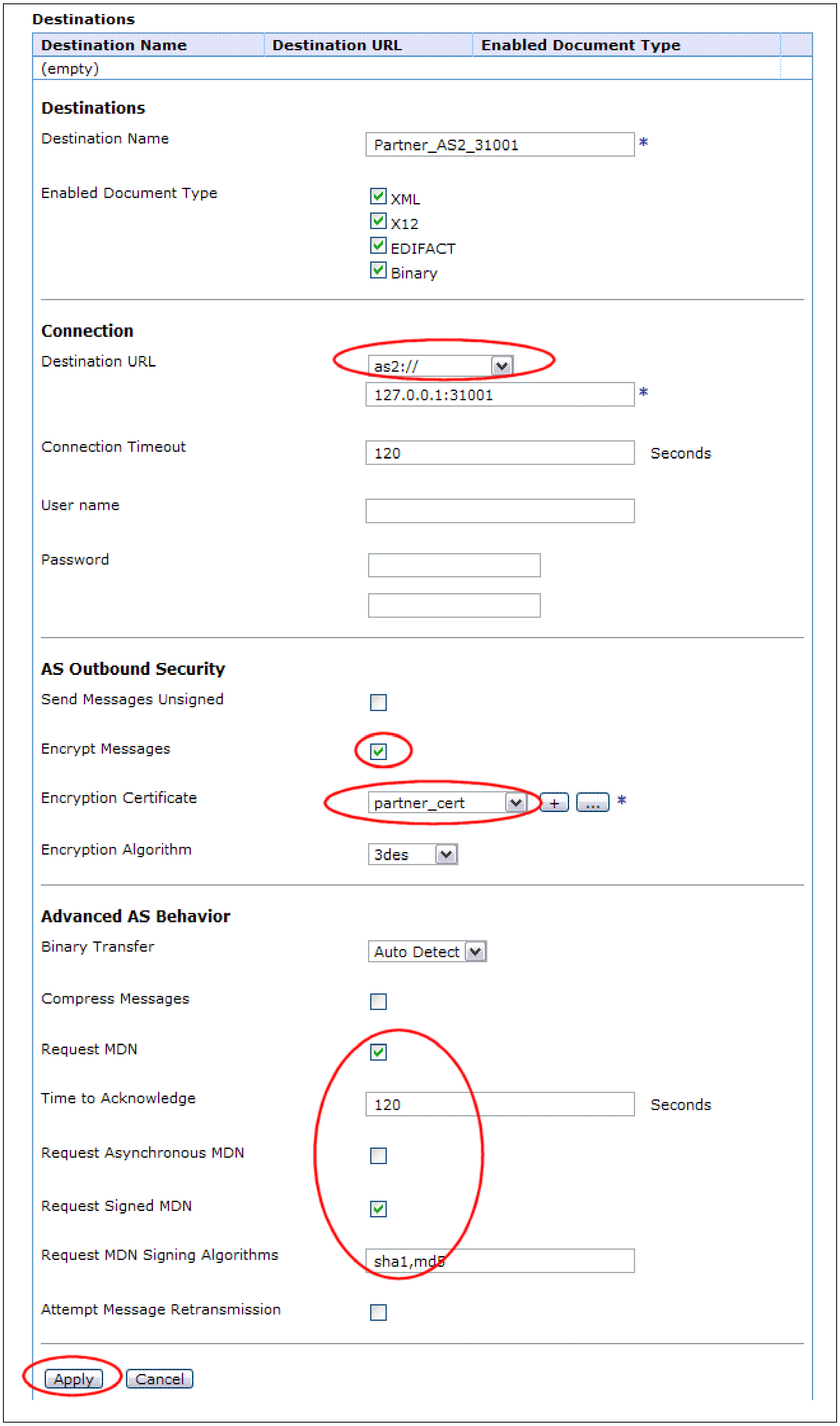


Figure 7-24 Mycompany External Profile - Configure Destination Tab - AS2

6. Optionally, configure the **Partner Profile Contacts** tab.
7. Click the **Apply** button in the top left corner of the Configure B2B Partner Profile view to save the partner's external profile.
8. Click **Save Config** on the top menu bar, just to the right of the Domain drop-down.

Next: You have completed the configuration of your partner's external profile. You are now ready to create your B2B Gateway Service.

Create your B2B Gateway Service

The B2B Gateway service is the engine that ties everything together. It uses profile management to ensure that all data that passes through the service is associated with a trading partner agreement between you and your external partner. This service is capable of natively extracting business ID's from EDI-X12, EDIFACT and XML files, if the files do not have business IDs to extract or are of a format the Gateway does not understand, they are treated as Binary and the business IDs can be set using a Routing Pre-processor stylesheet.

For this example we are using EDI-X12 payloads. For more information about how to configure Binary trading relationship, see Chapters 12 and 13 in the Redbooks publication, *IBM WebSphere DataPower B2B Appliance XB60 Revealed*, SG24-7745.

1. From the Control Panel, click the **B2B Gateway Service** icon, or search on "B2B" in the left navigation menu and select B2B Gateway Service.
2. In the Configure B2B Gateway list view, click the **Add** button.
3. Configure the **Main** tab as described in the following steps. See Figure 7-25.
   a. Enter the B2B Gateway name in the Name field. In this example we used MY_B2BGATEWAY.
   b. Choose **enabled** in the Administrative State field.
   c. Optionally add comments that describe this gateway.
   d. Take the defaults for the Document Storage Location and XML Manager fields.

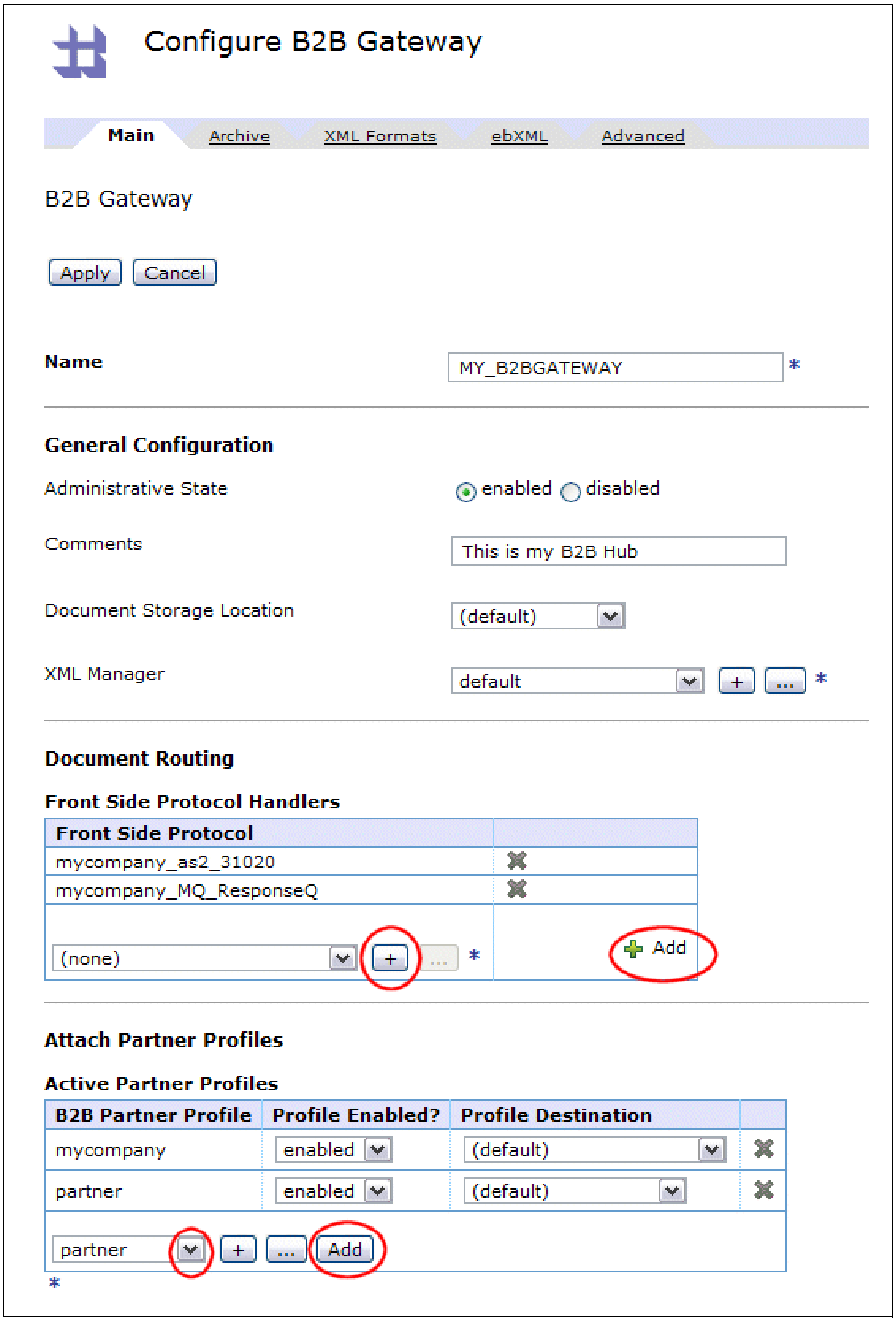


Figure 7-25 Configure MyCompany B2B Gateway Service

e. In the Document Routing section, create and configure an AS2 Front Side Handler; this handler is used to receive AS2 messages and MDNs from the trading partner, Click the **plus sign** in the Front Side Protocol Handlers list and select AS2 Front Side Handler. See Figure 7-26.

   i. In the **Main** tab, Name field, enter a descriptive name for this listener. In this example, we used mycompany_as2_31020.

   ii. Choose **enabled** in the Admin State field.

   iii. Optionally add comments that describe this handler.

   iv. Use 127.0.0.1 in the Local IP Address field.

Tip: In production deployments, you want to use a specific IP address that is associated to the Ethernet port being used for external communications. In this example, we are using two gateways on the local machine so we can use the local host IP.

v. In the Port Number field, enter an available port number, in this example we used 31020.

vi. Use the default values for all other fields and click the **Apply** button.

f. In the Front Side Protocol Handlers section, click (**+ Add**) to add the listener to the Front Side Protocol list.

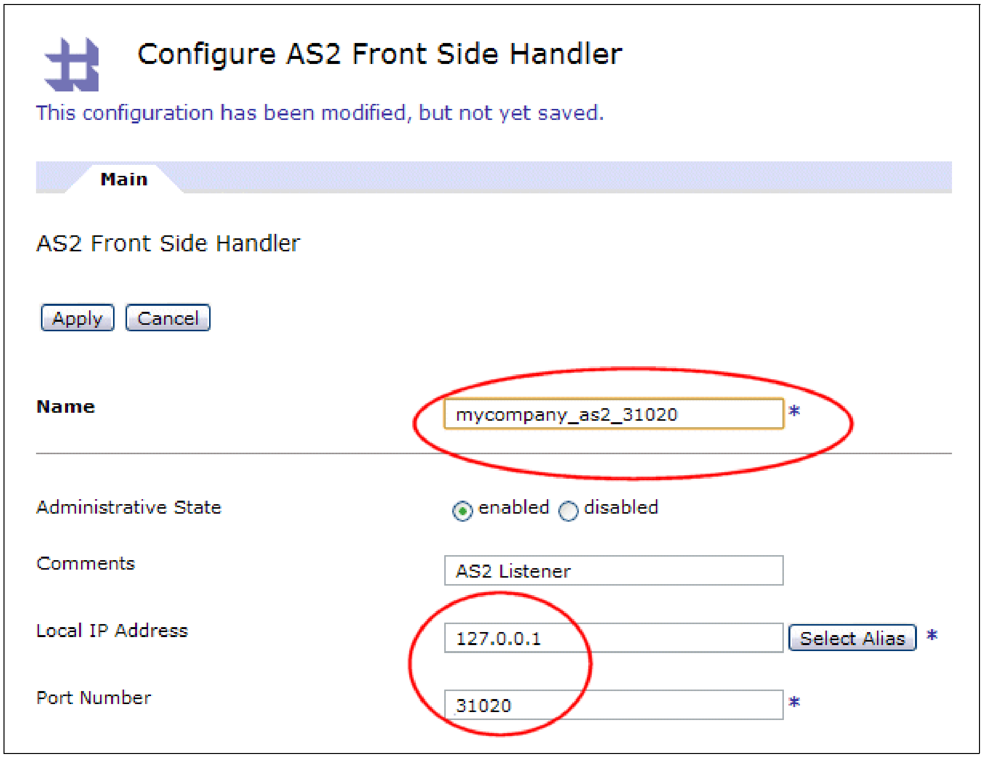


Figure 7-26 Configure AS2 Front Side Protocol Handler

g. In the Document Routing section, create and configure a MQ Front Side Handler. This handler is used to receive the 997's from the Sterling Integrator through MQ, Click the **plus sign** in the Front Side Protocol Handlers list and select MQ Front Side Handler. See Figure 7-27.

i. In the **Main** tab, Name field, enter a descriptive name for this listener, in this example we used mycompany_MQ_ResponseQ.

ii. Choose **enabled** in the Admin State field.

iii. Optionally add comments that describe this handler.

iv. In the Queue Manager field use the drop-down and choose the B2B02 Queue Manager object we created in the "Create an MQ Manager Object" on page 272 section of this book.

v. Enter the name of the Get Queue. This queue is the same name as you configured in your MQ implementation, in this example we used SUPPLY.B2B.RESPONSE.

vi. Use the default values for all other fields and click the **Apply** button.

h. In the Front Side Protocol Handlers section click (**+ Add**) to add the listener to the Front Side Protocol list.

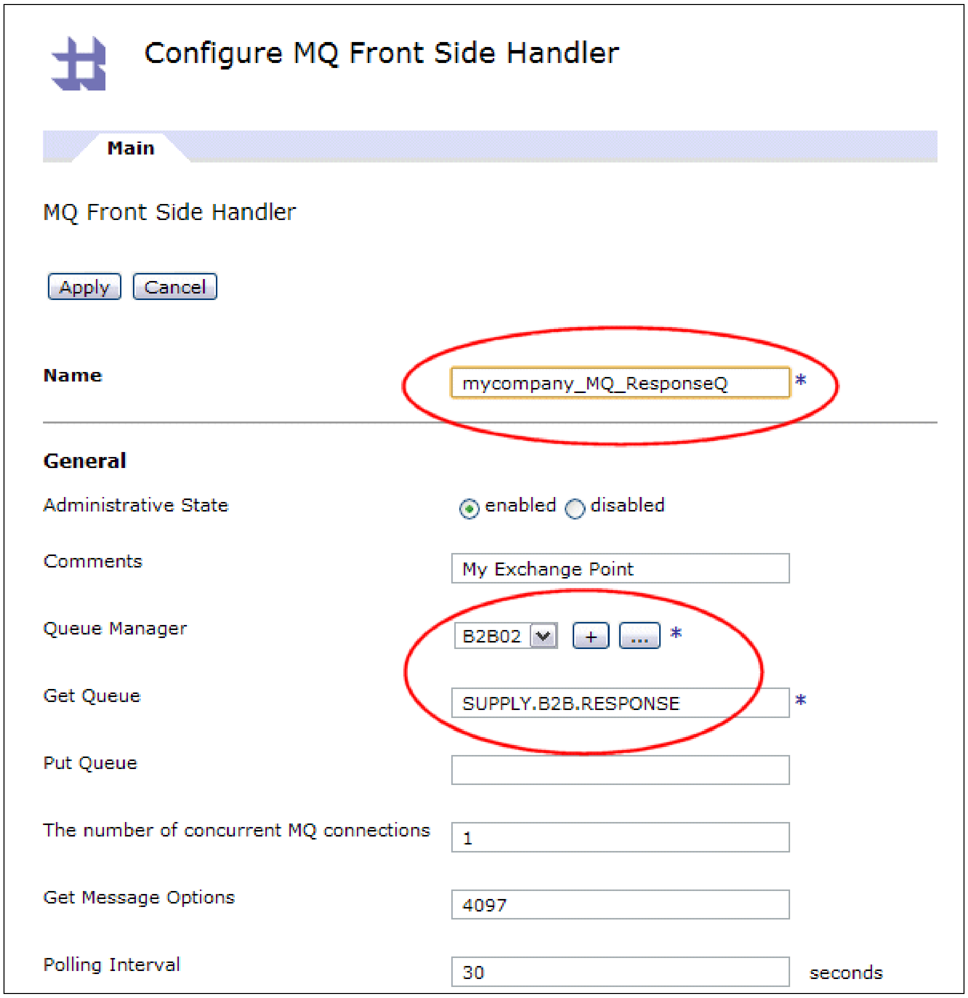


Figure 7-27 Configure MQ Front Side Handler

i. In the Attach Partner Profiles section, click the drop-down, select partner from the list, and click the **Add** button to add the external partner profile.

j. In the Attach Partner Profiles section, click the drop-down again, select **mycompany** from the list, and click the **Add** button to add the internal partner profile.

k. Do not use Active Profile Groups, they are not needed in this example.

> **Tip:** Do not click the **Apply** button at this point. We still have some required configuration needed to complete this profile.

4. Configure the **Archive** tab to purge documents as described in the following steps. See Figure on page 303.

> **Tip:** The **Archive** tab is used to automatically keep the B2B document and metadata storage areas clean. There are two modes; Archive and Purge, and Purge Only.

a. Click the **Archive** tab. The Name field carries over to the Archive screen. Do not change it.

b. In the Archive Mode field, use the drop-down and select **Purge Only**. We do not need to Archive files in this example.

c. Use the defaults for all of the other fields.

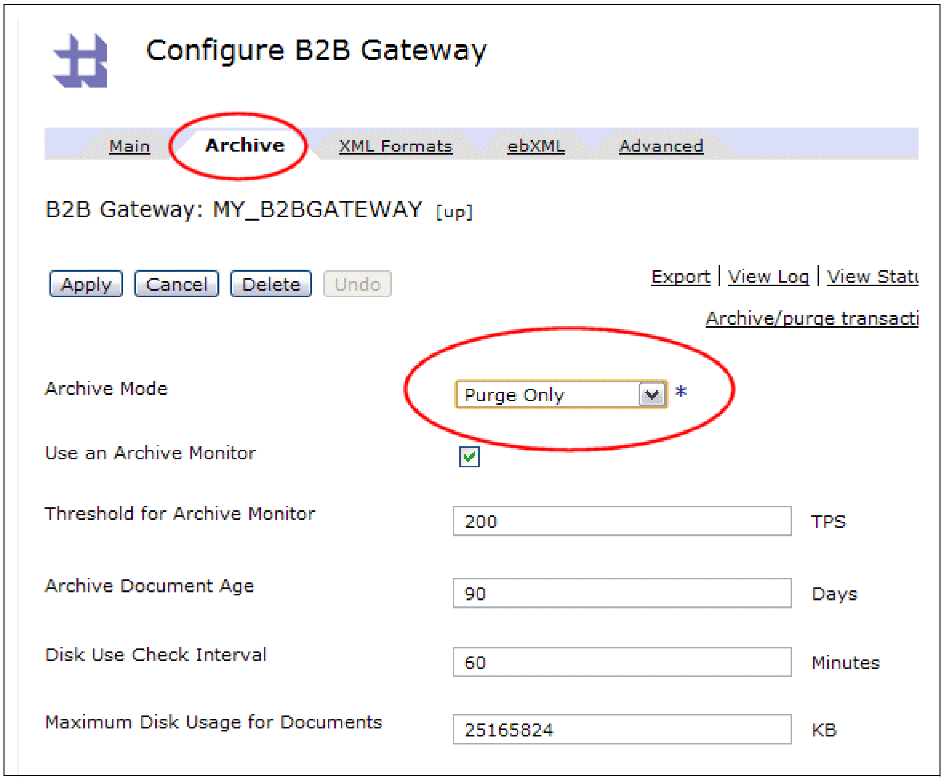


Figure 7-28 Configure the Archive tab

5. The remaining tabs are not required and are not used in this example. Click the **Apply** button at the top left to save the gateway.

6. Click **Save Config** on the top menu bar, just to the right of the Domain drop-down.

Next: You have completed the configuration of your B2B Gateway Service, you are now ready to configure Sterling Integrator and WTX.

7.4.2 Configuring the IBM WebSphere Transformation Extender

As stated in the introduction to this book, it is assumed that any WebSphere Transformation Extender component that is needed has already been installed. For any scenario using WebSphere Transformation Extender, the Design Studio must be installed on a development platform in order to create the maps needed for translation. In this scenario, WebSphere Transformation Extender for Integration Servers has also been installed on the development platform, as well as on the server that is intended to host the Sterling B2B Integrator.

Setting up the Design Studio

Details for setting up the Design Studio to integrate and deploy maps to Sterling B2B Integrator are discussed in detail in "Step 6: Configure WebSphere Transformation Extender" on page 145. Here we discuss the map used to translate the inbound message to the desired format for this scenario. We must also install the industry pack on the development platform and copy the library file edisvu.dll that comes with the industry pack, compliance check maps and other related files to the WebSphere Transformation Extender install folder on the server hosting Sterling B2B Integrator. See *EDI Compliance Checking* documentation for details at http://publib.boulder.ibm.com/infocenter/wtxdoc/v8r4m0/index.jsp?topic=%2Fcom.ibm.websphere.dtx.edicc.doc%2Ftopics%2Fg_edi_compliance_check_overview.htm.

Tip: A platform specific version of the library file, edisvu, is provided in the pack for each platform supported to integrate the WebSphere Transformation Extender Pack for EDI X12.

Translating the message with WebSphere Transformation Extender

For this scenario, a single translation map is needed to transform the incoming EDI X12 850 message into canonical xml form. After translation to canonical xml form, no further translation is needed, as the xml output from the transformation map is dropped by Sterling B2B Integrator onto the file system to be consumed by other processes. Transforming a canonical message to a target format is discussed in Chapter 6 of this book. The type tree representing the inbound EDI X12 850 message is a trimmed version of the type tree artifact supplied in the WebSphere Transformation Extender Industry Pack for EDI X12 (Figure 7-29).

This scenario takes advantage of the built in integration in the Sterling B2B Integrator De-envelop service to execute WebSphere Transformation Extender maps. In order to take advantage of this integration, additional editing of the type tree beyond the trimming methodology discussed earlier in the document is required. The steps for customizing a type tree for integration with the Sterling B2B Integrator De-envelope service is described *Pack for EDI* documentation that ships with the x12 industry pack and also found at this website:

http://publib.boulder.ibm.com/infocenter/wtxdoc/v8r4m0/index.jsp?topic=%2Fcom.ibm.websphere.dtx.packediref.doc%2Ftopics%2Fg_pkediref_Introduction.htm

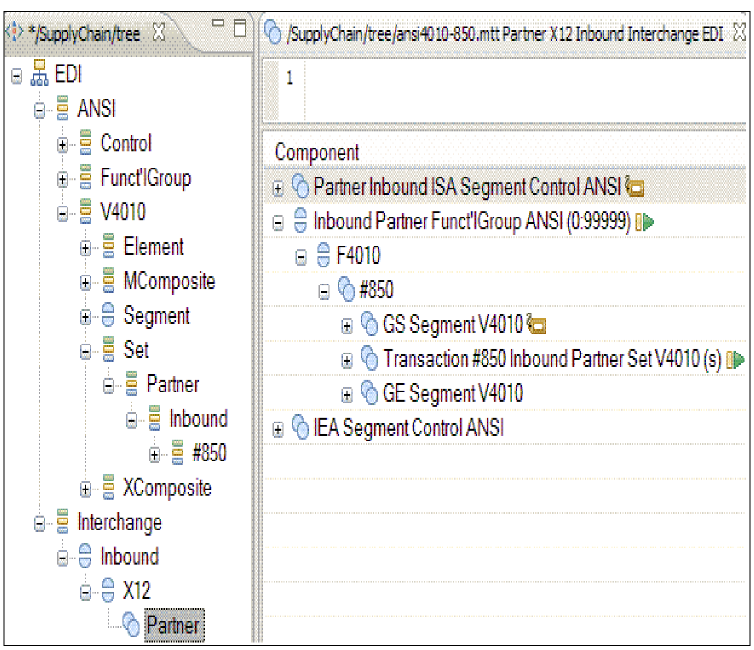


Figure 7-29 ansi4010-850.mtt.mtt

The target format is xml. However, instead of using the native xml schema definition to represent the output format, we chose to import the xml schema definition and generate a type tree to represent the xml. The reason that you would import a schema instead of using it it in its native form, is because sometimes there are rules related to elements in the xml that cannot be represented in a schema. Example 7-1 here shows the schema imported to create the type tree and Figure 7-30 on page 295 shows the type tree that was generated.

Example 7-1 XML Schema representation of the purchase order

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
   <xs:element name="OrderFile">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="FileHeader">
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="Sender">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element ref="Contact"/>
                           </xs:sequence>
                        </xs:complexType>
                     </xs:element>
                     <xs:element name="Receiver">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element ref="Contact"/>
                           </xs:sequence>
                        </xs:complexType>
                     </xs:element>
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
              <xs:element name="Order" maxOccurs="unbounded">
            <xs:complexType>
               <xs:sequence>
                  <xs:element ref="Header"/>
                  <xs:element ref="DetailLoop" maxOccurs="unbounded"/>
                  <xs:element ref="Trailer"/>
                 </xs:sequence>
               </xs:complexType>
      </xs:element>
                     </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="Header">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="PO_Purpose"/>
            <xs:element name="PO_Type"/>
            <xs:element name="PO_No"/>
            <xs:element name="PO_Date" type="xs:date"/>
            <xs:element ref="SendTo" minOccurs="0"/>
            <xs:element ref="Purchaser" minOccurs="0"/>
```

```
            <xs:element ref="Vendor" minOccurs="0"/>
            <xs:element ref="Consignee" minOccurs="0"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="DetailLoop">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="ItemNumber"/>
            <xs:element name="Quantity" type="xs:decimal"/>
            <xs:element name="UOM"/>
            <xs:element name="UnitPrice" type="xs:decimal"/>
            <xs:sequence>
               <xs:element name="SubDetail" maxOccurs="10">
                  <xs:complexType>
                     <xs:sequence>
                        <xs:element name="ProductQualifier"/>
                        <xs:element name="ProductDesc"/>
                     </xs:sequence>
                  </xs:complexType>
               </xs:element>
            </xs:sequence>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="Trailer">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="ItemCount" type="xs:integer" default="0"
minOccurs="0"/>
            <xs:element name="HashTotal" type="xs:decimal" default="0"
minOccurs="0"/>
            <xs:element name="AmountQualifier" minOccurs="0"/>
            <xs:element name="Amount" type="xs:decimal" default="0"
minOccurs="0"/>
            <xs:element name="DebitCredit" minOccurs="0"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="SendTo">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="Address"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="Purchaser">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="Address"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="Vendor">
      <xs:complexType>
```

```
            <xs:sequence>
               <xs:element ref="Address"/>
            </xs:sequence>
         </xs:complexType>
      </xs:element>
      <xs:element name="Consignee">
         <xs:complexType>
            <xs:sequence>
               <xs:element ref="Address"/>
            </xs:sequence>
         </xs:complexType>
      </xs:element>
      <xs:element name="Address">
         <xs:complexType>
            <xs:sequence>
               <xs:element name="Name"/>
               <xs:element name="Id"/>
               <xs:element name="Street" minOccurs="0" maxOccurs="2"/>
               <xs:element name="City" minOccurs="0"/>
               <xs:element name="State" minOccurs="0"/>
               <xs:element name="ZipCode" minOccurs="0"/>
               <xs:element name="Country" minOccurs="0"/>
            </xs:sequence>
         </xs:complexType>
      </xs:element>
      <xs:element name="Contact">
         <xs:complexType>
            <xs:sequence>
               <xs:element name="Id"/>
               <xs:element name="Qualifier"/>
            </xs:sequence>
         </xs:complexType>
      </xs:element>
   </xs:schema>
```

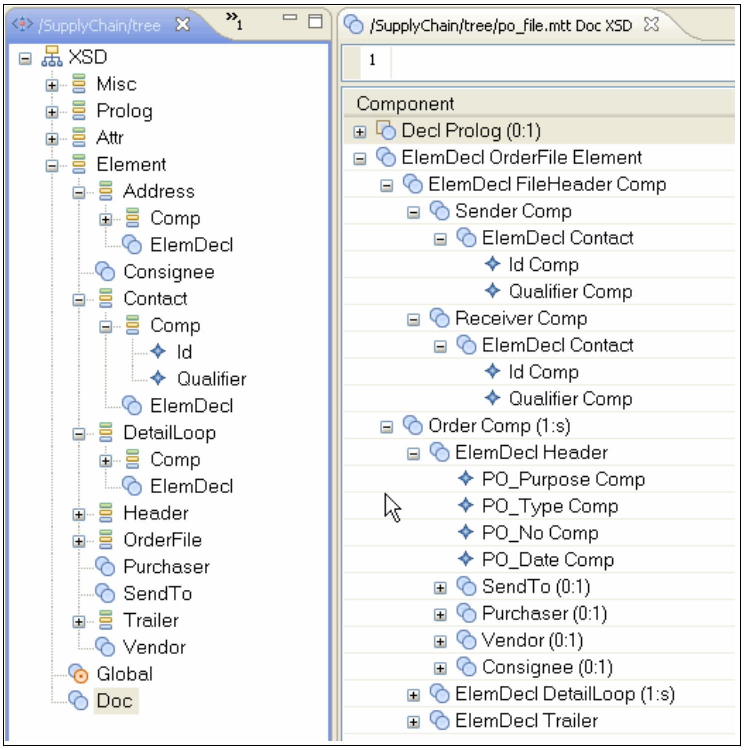


Figure 7-30 Type Tree generated by PO schema import

The map used to translate the EDI X12 850 to XML is composed of one input card and two output cards:

- The input card is defined using the trimmed type tree shown in Figure 7-29 on page 291.
- The first output card is defined using the type tree shown in Figure 7-30.
- The second output card is defined using an unbounded string element shown in Figure 7-31. This allows the map to return the xml representation of the purchase order when successful and nothing when it fails.

> **Tip:** The second output card is returned to the service, and the first output card is only used as input to the second output card.

- The input card definition for the map is shown in Figure 7-32 on page 296.
- The first output card definition for the map is shown in Figure 7-33 on page 297.
- The second output card for the map is shown in Figure 7-34 on page 298.
- The map is shown in Figure 7-35 on page 299.

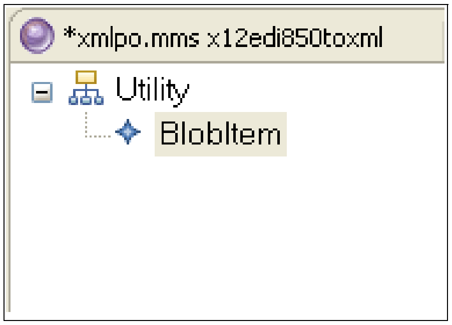


Figure 7-31   Unbounded string element

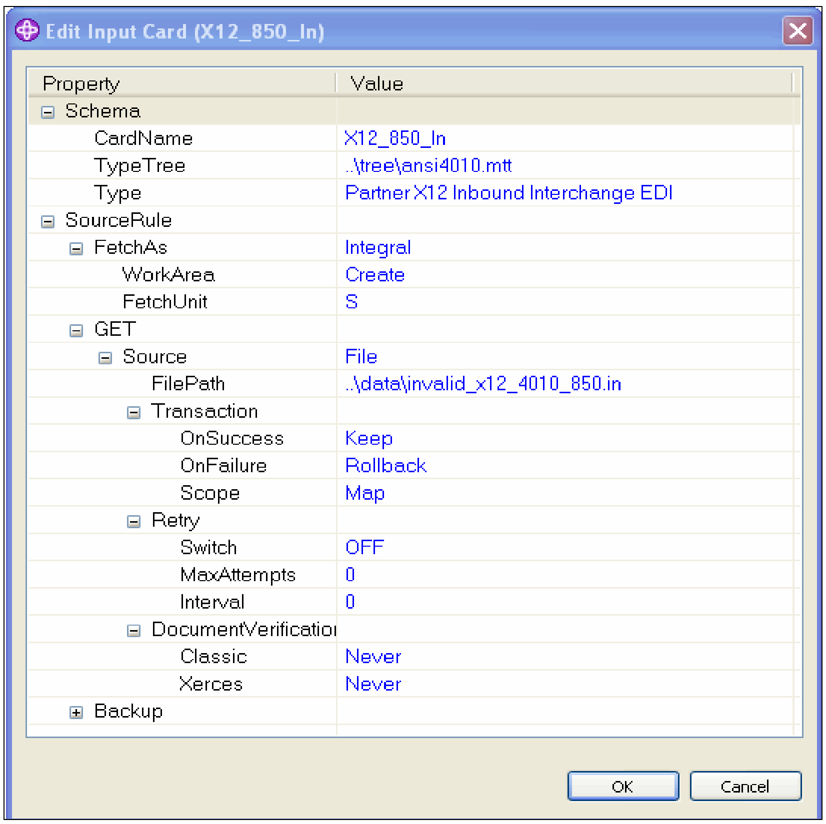


Figure 7-32   Input card definition

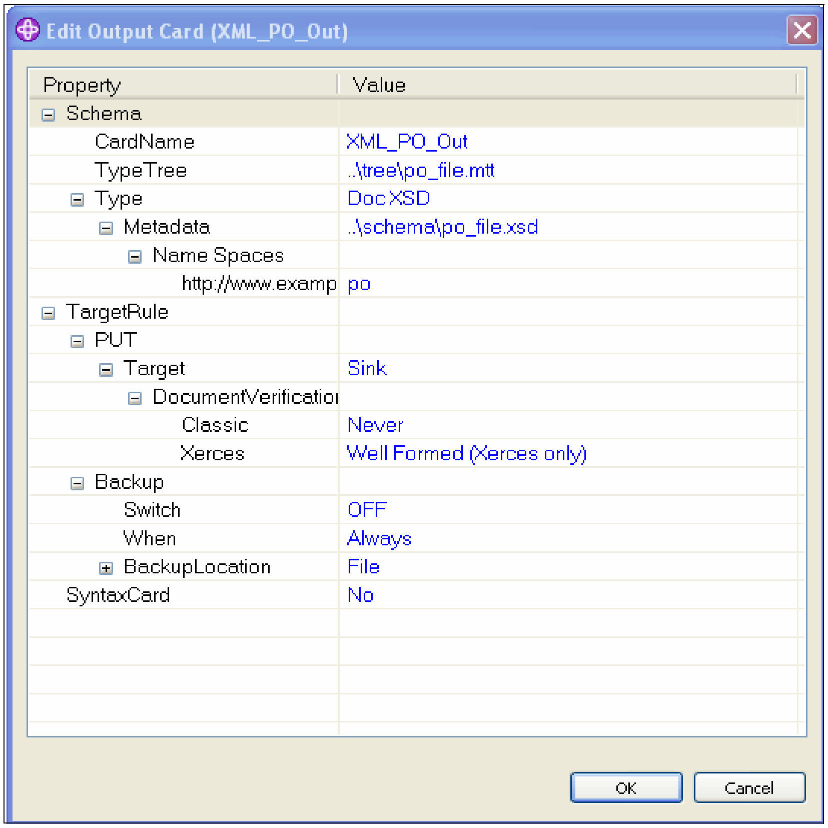


Figure 7-33 Output card 1 definition

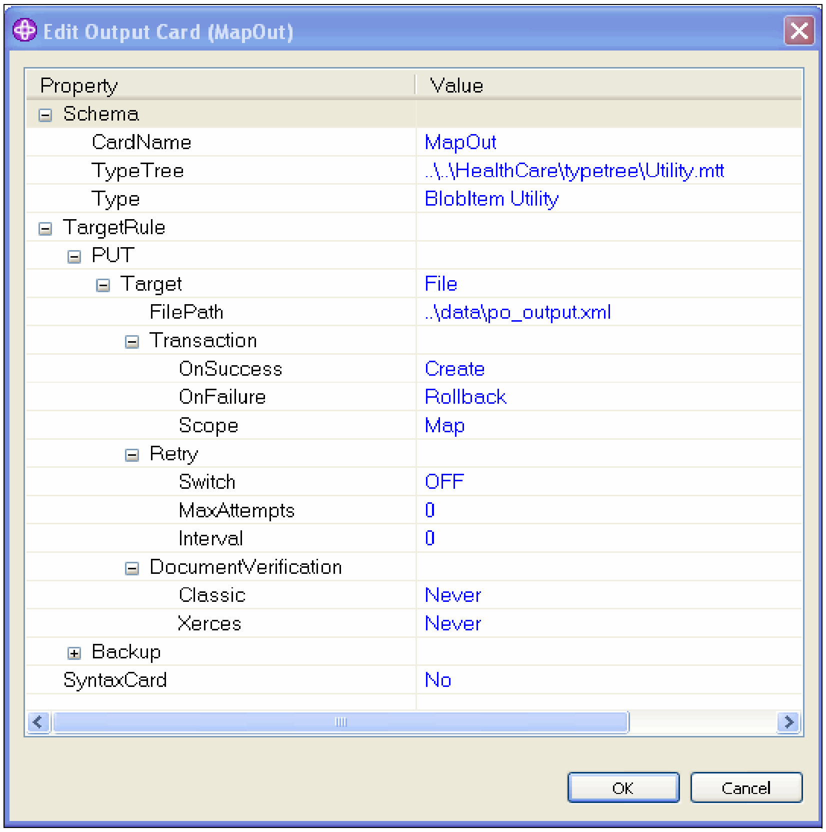


Figure 7-34 Output card 2 definition

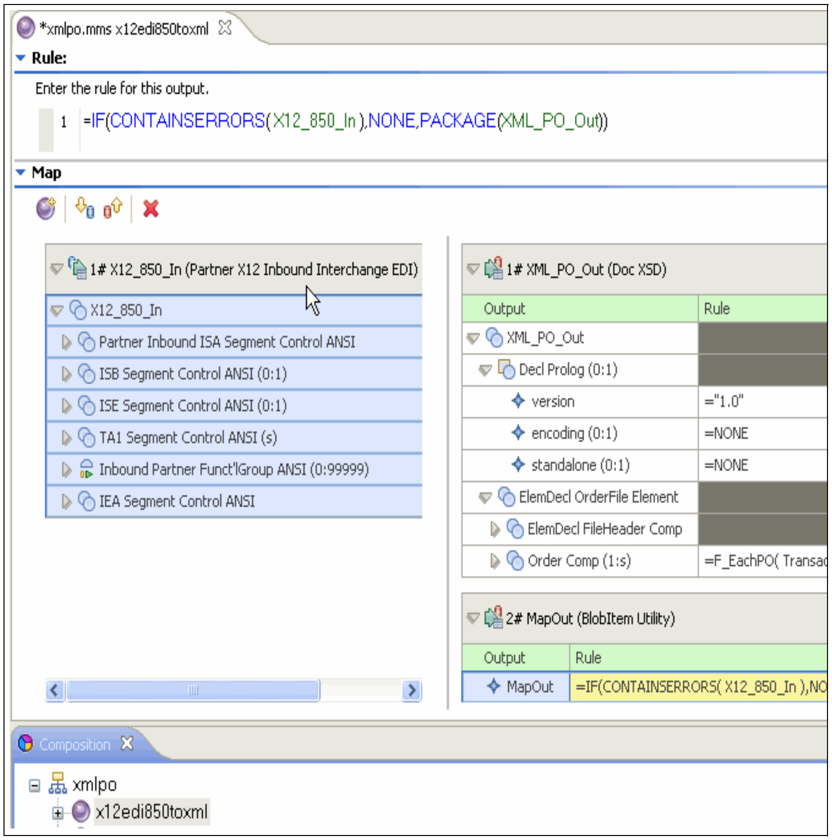


Figure 7-35 Map x12edi850toxml.mmc

After constructing the map using the map designer, you would then build and test locally. In order to execute this map from the Sterling B2B Integrator de-envelope service, the map settings would need to be modified as described in the *Pack for EDI* documentation that ships with the x12 industry pack:

`http://publib.boulder.ibm.com/infocenter/wtxdoc/v8r4m0/index.jsp?topic=%2Fcom.ibm.websphere.dtx.packediref.doc%2Ftopics%2Fg_pkediref_Introduction.htm`

If the local build and test is successful, you would test on the Sterling B2B integrator using the procedure defined in Chapter 4, “Routing and transforming messages” on page 69. After this test is successful, you would then deploy the map to Sterling B2B Integrator using the method also described in Chapter 4. After the map is successfully deployed, it can be used in a business process.

This chapter also makes use of the X12 compliance map system that ships with the industry pack. In order for the compliance map to execute properly it must be deployed to the server that hosts the Sterling B2B Integrator on the file system to a location that the Sterling B2B Integrator can access. deploying compliance check maps that ship with industry packs to be executed from Sterling B2B Integrator is described in the documentation that ships with the industry pack. For the scenario discussed in this chapter, this was done by extracting the file `SC_Artifacts1.zip`, referenced in 7.4, “Configuring the scenario” on page 265, so that the structure `C:\WTXMap` and its contents are created on the server. The compliance map is called ccx12.mmc and ships with 9 input cards and 12 output cards.

The compliance map is called as an external map, instead of from the Sterling B2B Integrator map repository. This means that it is executed from its location on the file system. The business process passes the EDI X12 850 data to the first input card. The other 8 input cards are used to grab the configuration data needed to run the compliance check.

Figure 7-36 shows the compliance map. An example of the 997 generated is shown in Example 7-2.

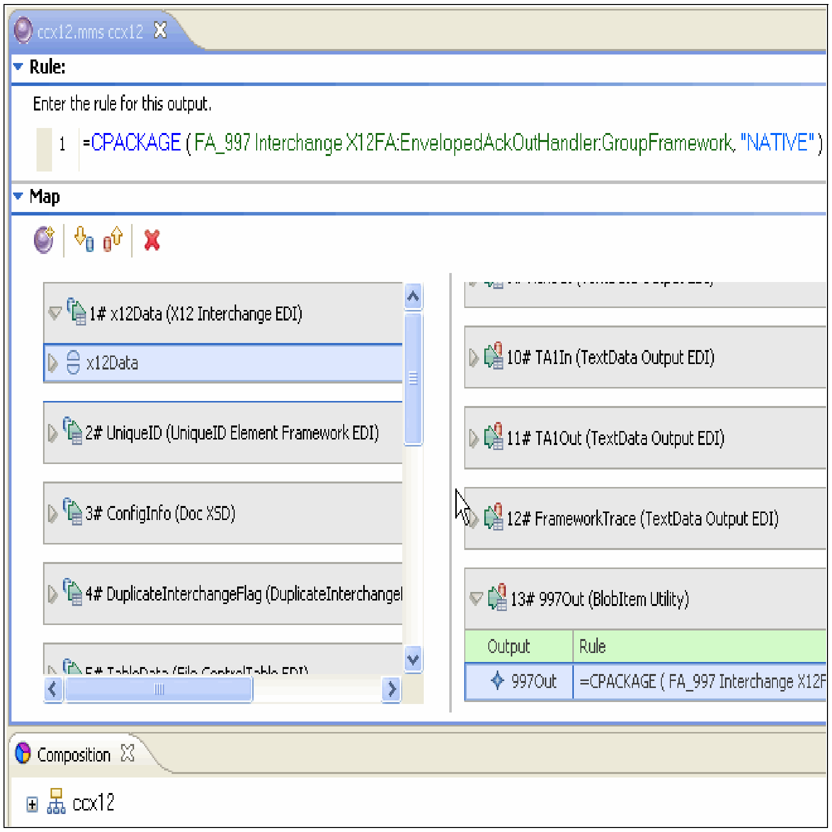


Figure 7-36 Compliance Map ccx12.mmc

Example 7-2 997 generated by compliance

```
ISA*00*          *00*          *ZZ*MYCOMPANY *ZZ*YOUR COMPANY
*111118*1205*U*00401*000000001*0*T*:
GS*FA*MYCOMPANY*YOUR COMPANY*20111118*120523*1*X*004010
ST*997*0001
AK1*MY*35
AK2*225*0001*005010
AK5*R*5
AK9*R*1*1*0
SE*6*0001
GE*1*1
IEA*1*000000001
```

7.4.3 Configuring IBM Sterling B2B Integrator

After the configuration is complete for WebSphere Transformation Extender, we must now configure Sterling B2B Integrator to accept the inbound message and call the components deployed in the previous section.

The process flow in Sterling B2B Integrator is to collect the message from the WebSphere MQ queue, and then write that file to a the file system to preserve it. Then, another service in Sterling B2B Integrator, operating on a schedule, collects the file and initiates the de-enveloping of the message. As part of the de-enveloping, we do a compliance check of the EDI 850 file using WebSphere Transformation Extender, and then transform the message into a format for our backend application to process, again using WebSphere Transformation Extender. During the transformation, we also prepare an acknowledgement, which is an EDI 997, and then place that acknowledgement onto another WebSphere MQ queue for the appliance to return to the trading partner.

To implement this portion of the scenario, we need to create the following items:

- A Business Process to collect the message from the WebSphere MQ queue and place it on the file system.
- File System Adapter configuration and schedule to collect the file and initiate the de-enveloping.
- Envelopes for the inbound EDI 850 file:
  - ISA-IEA
  - GS-GE
  - ST-SE
- Envelopes for the outbound acknowledgement (EDI 997):
  - ISA-IEA
  - GS-GE
  - ST-SE
- A Business Process to send the acknowledgement (EDI 997) to WebSphere MQ queue for the return to the trading partner.

Business Process to collect the message from WebSphere MQ queue

The first step is to configure a business process in Sterling B2B Integrator to collect the message from the WebSphere MQ queue, and then to write this file out to a location on the file system to preserve it for the next step.

Example 7-3 is the BPML that we can use for the business process in this scenario.

Example 7-3 BPML for the ITSOSupplychain_GetMQ_FSExtract business process

```
<process name = "ITSOSupplychain_GetMQ_FSExtract">
  <rule name="DocumentExists">
    <condition>string-length(PrimaryDocument/@SCIObjectID) &gt; 0</condition>
  </rule>

  <sequence>
    <sequence name="Sequence Start">
      <operation name="WebSphereMQ Suite Open Session">
        <participant name="WSMQ_OpenSession"/>
        <output message="WSMQOpenSessionInputMessage">
          <assign to="." from="*"></assign>
          <assign to="wsmq_channel">B2B.SUPPLYCHAIN.DP</assign>
```

```
    <assign to="wsmq_debug">No</assign>
    <assign to="wsmq_errorOnMQException">Yes</assign>
    <assign to="wsmq_hostname">9.42.170.226</assign>
    <assign to="wsmq_port">1414</assign>
    <assign to="wsmq_qmanager">B2BO2</assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

<operation name="WebSphereMQ Suite Open Queue">
  <participant name="WSMQ_OpenQueue"/>
  <output message="WSMQOpenQueueInputMessage">
    <assign to="." from="*"></assign>
    <assign to="wsmq_errorOnMQException">Yes</assign>
    <assign to="wsmq_MQOO_type">GET</assign>
    <assign to="wsmq_qname">SUPPLY.B2B.REQUEST</assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

<operation name="WebSphereMQ Suite Get Message">
  <participant name="WSMQ_GetMessage"/>
  <output message="WSMQGetMessageInputMessage">
    <assign to="." from="*"></assign>
    <assign to="wsmq_docTracking">Yes</assign>
    <assign to="wsmq_errorOnMQException">Yes</assign>
    <assign to="wsmq_qname">SUPPLY.B2B.REQUEST</assign>
    <assign to="wsmq_type">GETONE</assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

<operation name="WebSphereMQ Suite Commit">
  <participant name="WSMQ_Commit"/>
  <output message="WSMQCommitInputMessage">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

<operation name="WebSphereMQ Suite Close Queue">
  <participant name="WSMQ_CloseQueue"/>
  <output message="WSMQCloseQueueInputMessage">
    <assign to="." from="*"></assign>
    <assign to="wsmq_qname">SUPPLY.B2B.REQUEST</assign>
  </output>
  <input message="inmsg">
```

```
          <assign to="." from="*"></assign>
        </input>
      </operation>

      <operation name="WebSphereMQ Suite Close Session">
        <participant name="WSMQ_CloseSession"/>
        <output message="WSMQCloseSessionInputMessage">
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>

    </sequence>
    <choice>
      <select>
        <case ref="DocumentExists" activity="Sequence Start"/>
        <case ref="DocumentExists" negative="true" activity=""/>
      </select>

      <sequence name="Sequence Start">
        <operation name="File System Adapter">
          <participant name="ITSORetail_MQ_FSA"/>
          <output message="FileSystemInputMessage">
            <assign to="." from="*"></assign>
            <assign to="Action">FS_EXTRACT</assign>
            <assign to="appendOnExtract">false</assign>
            <assign to="concatenateFiles">false</assign>
          </output>
          <input message="inmsg">
            <assign to="." from="*"></assign>
          </input>
        </operation>

      </sequence>
      <sequence>
        <assign to="false" from="true"></assign>
      </sequence>
    </choice>
  </sequence>
</process>
```

Note that this business process is using a File System Adapter to extract the file to the file system, which need to be created. It also uses a Rule to check for the existence of a document, which corresponds to the message received from the WebSphere MQ queue. If that document does not exist, the process simply ends without attempting to write the file out.

File System Adapter configuration and schedule

In order to collect the file being written out in the previous step, and to begin the de-enveloping process, we must create a File System Adapter configuration that is set to:

- Collect the file.
- Run on a schedule.
- Initiate (bootstrap) the proper business process.

Figure 7-37 is the File System Adapter used in this scenario.

ITSOSupplychain_Collect_FSA

Service Settings	
Service Type	File System Adapter
Description	v1
System Name	ITSOSupplychain_Collect_FSA
Group Name	None
Collection folder	C:\ITSOSupplychain\mq_extract
Filename filter	None provided
Collect files from sub folders within and including the collection folder?	No
Use the absolute file path name for the document name?	No
Start a business process once files are collected?	Yes
Business Process	EDIInboundBootstrap
Document Storage Type	System Default
Obscure File Contents?	No
User Parameter1	None provided
User Parameter2	None provided
User Parameter3	None provided
User Parameter4	None provided
User Parameter5	None provided
Schedule	Every 0 hours and 1 minutes Excluded Times None Excluded Dates None Run As User: admin
Extraction folder	C:\ITSOSupplychain\mq_extract
Unobscure File Contents?	No
Filenaming convention	Use the original filename as the extracted filename

Figure 7-37 ITSOSupplychain_Collect_FSA configuration

Note that we are going to “bootstrap” the process EDIInboundBootstrap, which initiates the de-enveloping of the message. It uses the EDIDeenvelope service to call the X12DeenvelopeUnified business process to de-envelope the message, create the acknowledgement, and then write the transformed message out to the file system for the ‘backend’ application.

This File System Adapter is set to run on a 1-minute schedule, which ensures that any files placed in the collection folder are collected and processed almost immediately.

Envelopes for the inbound EDI 850 file

For the message being collected, we need to configure standard X12 inbound envelopes for the EDI 850 type.

We need to create an envelope at each of the three standard levels:

- ISA-IEA
- GS-GE
- ST-SE

When creating the envelopes, there are several items we need to take into account:

- In order to use WebSphere Transformation Extender to check the compliance of the message, we need to specify “Yes” to “Use WTX compliance checking” in the ISA-IEA envelope.
- We need to ensure that we select “Yes” to “Generate an Acknowledgement when this group is received” in the GS-GE level envelope. In addition, we must also specify the format of the acknowledgement to create, in this case, an EDI 997. We also need to set “Translate and/or Compliance Check Document” to “Yes”.
- In the ST-SE level, we must specify the map to be used to transform the data. In this case, we select a WebSphere Transformation Extender map, named ‘x12edi850toxml’, that was checked into Sterling B2B Integrator. We also set “Translate and/or Compliance Check Document” to “Yes”.
- Also in the ST-SE level, be sure to select the option “Extract to a file system directory” for the “Extraction Options,” and specify the location on the file system where the backend application can collect the file.

Envelopes for the outbound acknowledgement (EDI 997)

Similar to the previous situation, we must also create envelopes at each standard level for the EDI 997 that is prepared as part of the de-enveloping of the message.

The following three envelopes must be created:

- ISA-ISE
- GS-GE
- ST-SE

For each of these envelopes, we need to specify that each is for Outbound X12. In the ISA-IEA level, we specify the business process to run to output this acknowledgement to the WebSphere MQ queue for return to the trading partner. The business process itself is described in the next section.

Business Process to send acknowledgement to WebSphere MQ queue

Finally, we must create the business process to be called by the X12EnvelopeUnified process that is initiated when the acknowledgement is created and enveloped. This business process is used to send the acknowledgement, in this case, an EDI 997, back by the WebSphere MQ queue to the appliance, for eventual transmission back to the trading partner.

Example 7-4 is the BPML of the this process. Note that this process is quite simple, taking the EDI 997 message and putting it to the WebSphere MQ queue.

Example 7-4 BPML of the ITSOSupplyChain_Put_MQ process

```
<process name = "ITSOSupplychain_Put_MQ">
  <sequence name="Sequence Start">
    <operation name="WebSphereMQ Suite Open Session">
      <participant name="WSMQ_OpenSession"/>
      <output message="WSMQOpenSessionInputMessage">
        <assign to="." from="*"></assign>
        <assign to="wsmq_channel">B2B.SUPPLYCHAIN.DP</assign>
        <assign to="wsmq_debug">No</assign>
        <assign to="wsmq_errorOnMQException">Yes</assign>
        <assign to="wsmq_hostname">9.42.170.226</assign>
        <assign to="wsmq_port">1414</assign>
        <assign to="wsmq_qmanager">B2B02</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="WebSphereMQ Suite Open Queue">
      <participant name="WSMQ_OpenQueue"/>
      <output message="WSMQOpenQueueInputMessage">
        <assign to="." from="*"></assign>
        <assign to="wsmq_errorOnMQException">Yes</assign>
        <assign to="wsmq_MQOO_type">PUT</assign>
        <assign to="wsmq_qname">SUPPLY.B2B.RESPONSE</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="WebSphereMQ Suite Put Message">
      <participant name="WSMQ_PutMessage"/>
      <output message="WSMQPutMessageInputMessage">
        <assign to="." from="*"></assign>
        <assign to="wsmq_docTracking">Yes</assign>
        <assign to="wsmq_errorOnMQException">Yes</assign>
        <assign to="wsmq_qname">SUPPLY.B2B.RESPONSE</assign>
        <assign to="wsmq_send_retryCount">10</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
```

```
    <operation name="WebSphereMQ Suite Commit">
      <participant name="WSMQ_Commit"/>
      <output message="WSMQCommitInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="WebSphereMQ Suite Close Queue">
      <participant name="WSMQ_CloseQueue"/>
      <output message="WSMQCloseQueueInputMessage">
        <assign to="." from="*"></assign>
        <assign to="wsmq_qname">SUPPLY.B2B.RESPONSE</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="WebSphereMQ Suite Close Session">
      <participant name="WSMQ_CloseSession"/>
      <output message="WSMQCloseSessionInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

  </sequence>
</process>
```

This process is set to handle only a single message, and would be called to output this acknowledgement each time one is created by the X12EnvelopeUnified process.

7.5 Testing the scenario

In this section, we are testing the end-to-end flow through our configuration of the four IBM products used in this scenario (DataPower XB62, WebSphere MQ, Sterling Integrator, and WTX). As depicted in the scenario overview in Figure 7-1 on page 264, we consume an AS2 message containing an EDI-X12 purchase order (850) from the external trading partner and return an EDI-X12 functional acknowledgment (997) that is wrapped in an AS2 message envelope.

In the Partner's B2B Gateway, we are using a simulated backend, which communicates over an http connection. For the purpose of this scenario, we start the transfer from the external partner’s backend and send the 850 over the partner’s http integration point using an HTTP Utility. You can use any utility you are comfortable with, in this test we are using a tool called NetTool.

7.5.1 Inbound flow

Follow these steps:

1. Locate the `x12_4010_850.in` file the location where you unzipped `SCS_DataPowerArtifacts.zip` to, use your HTTP utility to post it to the partner's http Front Side Handler, in this example we used the IP of our XB62 and a port of 31002. Figure 7-38 depicts what the post looks like when using the NetTool utility.

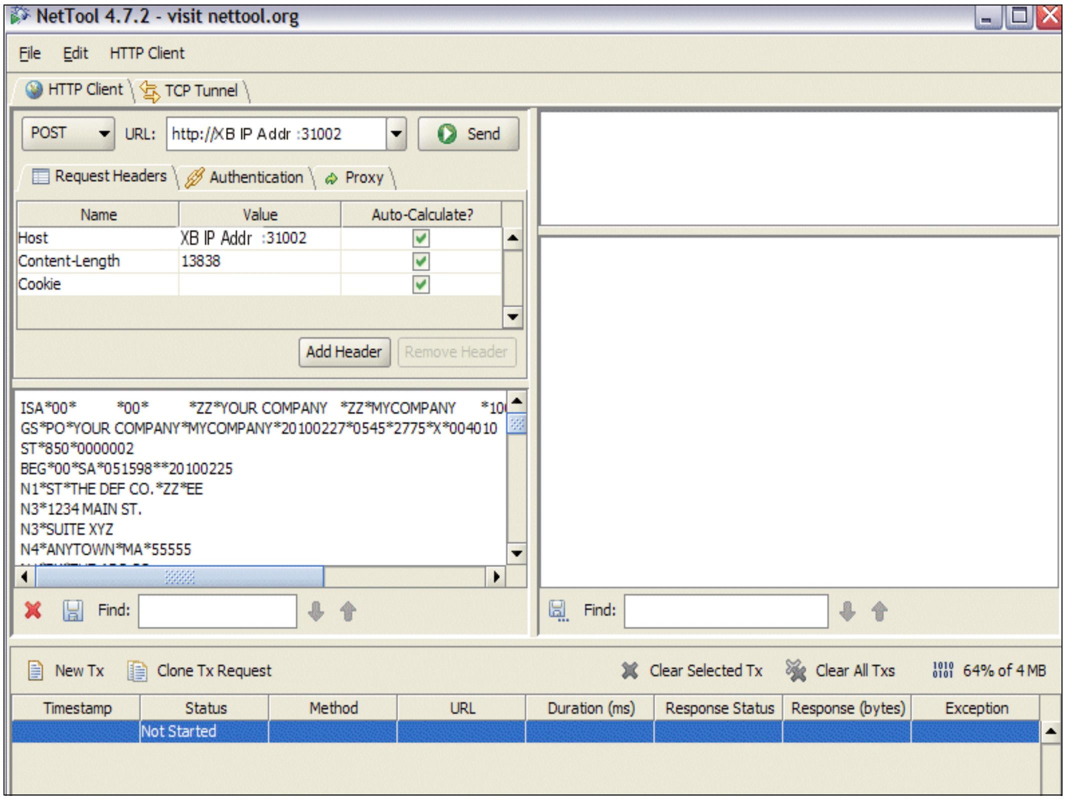


Figure 7-38 NetTool post of sample EDI PO

2. The Partner's B2B gateway parses the RAW EDI 850 file, recognizes that it is X12, and extracts the Sender and Receiver information from the file. Figure 7-39 shows where the sender (ZZYOUR COMPANY) and receiver (ZZMYCOMPANY) information is located in the EDI 850 file.

```
ISA*00*          *00*          *ZZ*YOUR COMPANY   *ZZ*MYCOMPANY
*100227*0545*U*00401*000008160*0*P*>
GS*PO*YOUR COMPANY*MYCOMPANY*20100227*0545*2775*X*004010
ST*850*0000002
BEG*00*SA*051598**20100225
N1*ST*THE DEF CO.*ZZ*EE
N3*1234 MAIN ST.
N3*SUITE XYZ
N4*ANYTOWN*MA*55555
```

Figure 7-39 EDI 850 from Partner

3. Based on the destination configuration in the mycompany external profile in the Partner's B2B Gateway, the file is packaged in AS2 and sent to your B2B Gateway.
4. Your B2B Gateway receives the AS2 message at the AS2 Front-side Handler and extracts the Sender and Receiver information from the AS2 headers. Figure 7-40 shows where the AS2-From (ZZYOUR COMPANY) and AS2-To: (ZZMYCOMPANY) information is located.

```
POST / HTTP/1.1
Host: 127.0.0.1:31020
Cookie:
Via: 1.1 AwAAADEAAAA-
X-CLIENT-IP: 9.76.194.232
Date: Fri, 16 Dec 2011 07:28:40 GMT
AS2-From: "ZZYOUR COMPANY"
AS2-To: ZZMYCOMPANY
AS2-Version: 1.1
Message-ID: <a9bdb845-c147-41c3-ab0f-a8415f4d9222@9.70.153.61>
Subject: ZZMYCOMPANY;ZZYOUR COMPANY
Disposition-Notification-To: ignored@example.com
Disposition-Notification-Options: signed-receipt-protocol=optional,
pkcs7-signature; signed-receipt-micalg=optional, sha1,md5
Recipient-Address: as2://127.0.0.1:31020
Content-Type: application/pkcs7-mime; smime-type=enveloped-data;
name="smime.p7m"
Content-Disposition: attachment; filename="smime.p7m"
Connection: Keep-Alive
Content-Length: 14221

0,7‰*†H†÷
  ,7z0,7v
```

Figure 7-40 Sample AS2 header

5. Your B2B Gateway Service unpackages the AS2 Envelope, looks up the partner information, and verifies that the partners exist and are allowed to trade EDI documents. It looks at the Destination that is configured for the receiving profile and delivers the EDI 850 file to the backside. In this example, our backside is a WebSphere MQ queue that is being monitored by Sterling Integrator.
6. After the EDI 850 is successfully delivered to MQ, your B2B Gateway sends an AS2 MDN to the Partner informing them that the transaction was successfully received.

Inbound flow: This completes the Inbound flow from a DataPower perspective. Behind the scenes, Sterling Integrator is picking up the file and running it through a business process flow that uses WTX to validate the EDI, generate a 997, and transform it to an XML file. Sterling Integrator will then pass the EDI 997 into an MQ queue that the XB62 is monitoring.

7.5.2 Outbound flow

The EDI 997 is retrieved from the MQ Queue and parsed for sender and receiver ID information:

1. Your B2B gateway retrieves the EDI 997 from the MQ queue and parses the file, recognizes that it is X12, and extracts the Sender and Receiver information from the file. The EDI 997 will have a sender of ZZMYCOMPANY and a receiver of ZZYOUR COMPANY (Figure 7-39).
2. Based on the destination configuration in the partner's external profile in the Partner's B2B Gateway, the file is packaged in AS2 and sent to the partner's B2B Gateway.
3. The partner's B2B Gateway receives the AS2 message at the AS2 Front-side Handler and extracts the Sender and Receiver information from the AS2 headers.
4. The partner's B2B Gateway Service unpackages the AS2 Envelope, looks up the partner information, and verifies that the profiles exist and are allowed to trade EDI documents. It looks at the Destination that is configured for the receiving profile and delivers the EDI 997 file to the back-side. In this example, our back-side for the partner is an HTTP location that throws away the file.
5. After the EDI 997 is successfully delivered to the http location, the partner's B2B Gateway sends an AS2 MDN to your B2B Gateway informing you that the transaction was successfully received.

Outbound flow: This completes the outbound flow portion of the scenario.

7.5.3 Viewing the test results

Follow these steps to see the test results:

1. Next we view the transaction in the XB62's B2B Transaction Viewer; Click on Control Panel and then on B2B Transaction Viewer.
2. In the B2B Viewer, click **Show AS Only**. If the configuration is done correctly, you should see results similar to Figure 7-41. This figure splits the viewer horizontally for the purpose of this book. Note that transaction 130 was received at your ID (ZZMYCOMPANY), this is the EDI 850, transaction 131 is sent by your ID (ZZMYCOMPANY) and is the outbound EDI 997. The result code is success, meaning that both the inbound and outbound flows were successfully completed.

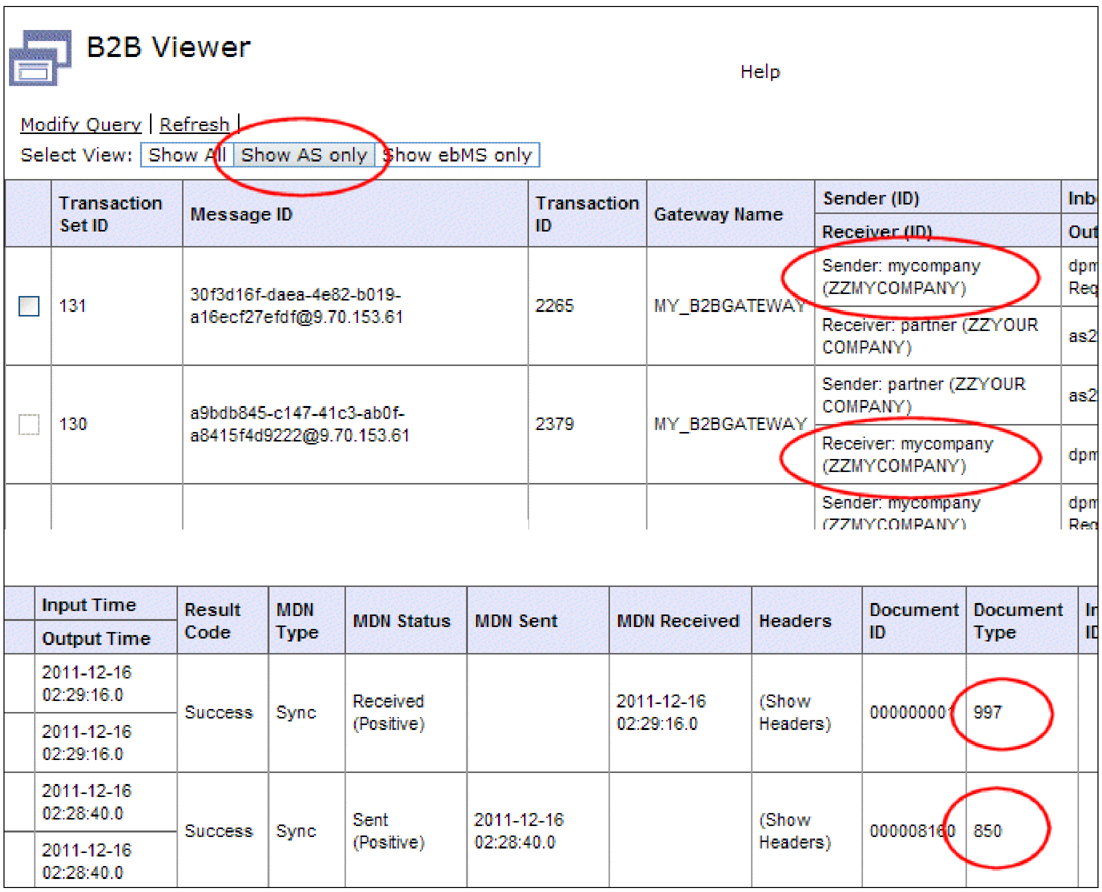


Figure 7-41 B2B Viewer - split view

7.6 Conclusion

Now that you have completed this chapter, you should have a good understanding of how IBM WebSphere DataPower B2B Appliances complement the Sterling Integrator product. You can see how they sit at the edge of the network to provide B2B security and governance acting as a secure gateway protecting downstream systems sitting in the protected network zone.

We only demonstrated the B2B capabilities of the device in this example. However, the WebSphere DataPower B2B Appliance XB62 is a functional super-set of the Security and Integration appliances that came before it. Many of the other services in the device can be used in conjunction with the B2B services to provide very robust and reliable secure options for connecting to your external trading partners. This chapter also provided you with a good understanding of how the Sterling Integrator product can process and transform data utilizing its ability to integrate with WTX for EDI validation and transformation.

It is important to note that all of the products used in this scenario are very flexible and can be used in many different ways to accomplish similar if not the same functions demonstrated in this example. The scenario chosen in this example is only one way to integrate these products together. Its intent is meant to illustrate the value of using these products together to provide a more complete and efficient solution for handling your supply chain connectivity needs.

8

IBM Sterling B2B Cloud Service scenarios

This chapter shows two trading scenarios using IBM Sterling File Transfer Service and IBM Sterling B2B Collaboration Network to connect to trading partners. We also describe IBM Sterling Web Forms to enable Non-EDI trading partners.

This chapter includes the following sections:

8.1 File-based B2B cloud scenario

We start this chapter with a file-based B2B cloud scenario.

8.1.1 Business value

ITSORetailCloud is a company which uses FTP and FTP/S with 500 customers for point-to-point file exchange. There is no special handling required with these files; the backend systems consume or generate the payloads that are exchanged.

The company has a corporate-wide strategy to standardize in all areas possible to reduce the complexity of their environment, reduce redundancy and to eliminate one-off solutions. They have decided to standardize the file transfer protocol on FTP/S.

Through surveying their customer community, they identified the following facts:

- 100 of the 500 customers online today need to be migrated to FTP/S and 50 of them do not migrate.
- There are 1,000 additional customers and supply chain partners with which they want to establish point-to-point connectivity. Also, 500 of the 1,000 customers are using FTP, 300 customers are using FTP/S, and 200 customers are using S/FTP.
- They have no expertise in S/FTP and do not want to hire, contract, or establish the skills in-house.
- Their IT staff do not have the time to handle the ongoing support needs of the additional 1,000 customers.
- There is not sufficient file transfer visibility in their existing environment to support all the protocols.

Their IT decides to recommend that the file transfer expansion project be outsourced due to compliance issues, staffing requirements, partner onboarding, community management and visibility. The business areas agree to fund the project. They are looking for the following business benefits:

- Reduced IT infrastructure and operational costs
- Reduced B2B file transfer complexity
- A highly scalable platform to grow their business
- Centralized monitoring and management
- Skilled personnel to manage their B2B file transfer environment

8.1.2 Presenting the scenario

In this scenario, we onboard ITSORetailCloud to use IBM Sterling File Transfer Service. We also onboard all of its existing and additional trading partners to use IBM Sterling File Transfer Service.

We establish one discreet FTP/S connection to bring ITSORetailCloud up to IBM Sterling File Transfer Service. Each of its trading partners also have one discreet connection to IBM Sterling File Transfer Service using FTP, FTP/S, or S/FTP.

Figure 8-1 shows the connectivity choice of this scenario in a high-level perspective.

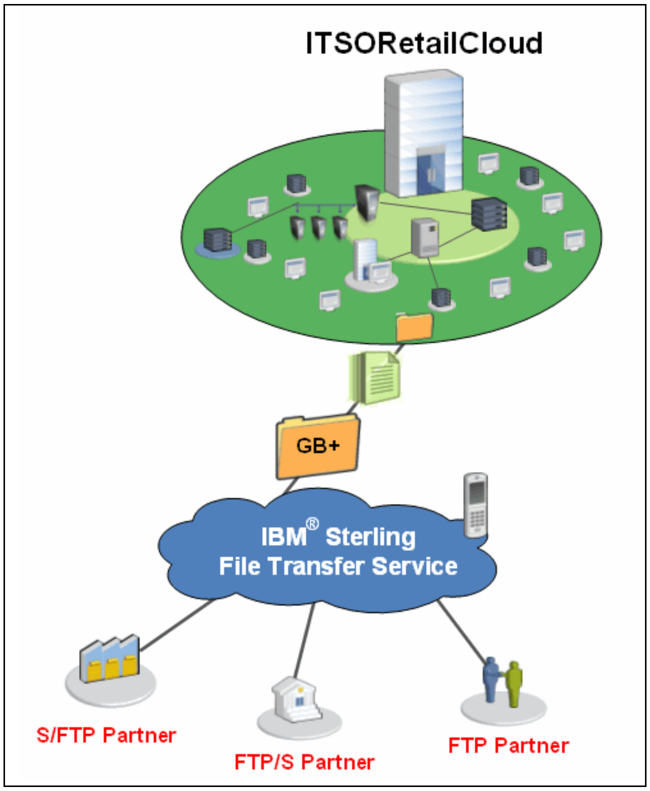


Figure 8-1 Scenario overview

8.1.3 Configuring the scenario

After depicting at a high level the connectivity choice of the scenario that we implement, we explain in detail how to onboard the customer and all of its trading partners.

Onboarding is the name of the process during which IBM sets up customer and its trading partners to access File Transfer Service. The customer and its trading partners receive help from IBM in getting started with File Transfer Service. The following list is the general sequence of steps in the onboarding process:

1. IBM sends a Welcome Letter and questionnaire to the customer's contact person for File Transfer Service.
2. The customer returns the completed questionnaire to IBM.
3. IBM grants the customer access to File Transfer Service.
4. The customer tests the connection to File Transfer Service, working with IBM as necessary.
5. IBM onboards the customer's trading partners. For each trading partner, here are the general sequence of steps in the onboarding process:
   a. IBM sends an invitation to contact the trading partner to get set up with File Transfer Service. The trading partner might need to provide communications information for setting up the File Transfer Service connection, if the customer has already not given this information to IBM.

   b. The trading partner tests the connection to File Transfer Service.
   c. After the trading partner is successfully onboarded, IBM sends the trading partner information about how to contact IBM customer support for assistance in the future.
6. After the trading partners are onboarded, IBM sends the customer information about how to contact IBM customer support, which has a world-wide staff providing support 24 hours per day, 365 days per year.

Onboarding the FTP/S customer

The customer uses their FTP client software that supports SSL encryption to access File Transfer Service. They complete an FTP/S service questionnaire that collects the following information:

- Company name and contact information
- FTP/S service details such as FTP transfer mode and type of data to be transferred
- Access method such as by Internet or VPN connection

After IBM configures the customer's account using information collected on the questionnaire, IBM provides the customer with a URL and a user ID and password to connect to the File Transfer Service. The customer works together with IBM to test the connection to File Transfer Service and waits for the IBM specialists to onboard the customer's trading partners and implement the routing logic of the files to be transferred.

Onboarding the FTP partner

Each of the customer's FTP partners uses their FTP client or server software to access File Transfer Service. They complete an FTP service questionnaire that collects the following information:

- Company name and contact information
- FTP service details, such as FTP transfer mode and type of data to be transferred
- Access method, such as by Internet or VPN connection
- Delivery method, which defines who initiates the FTP connection

After IBM configures the trading partner's account using information collected on the questionnaire, IBM provides the trading partner with a URL, as well as a user ID and password, to connect to the File Transfer Service. The trading partner then works together with IBM to test the connection to File Transfer Service.

Onboarding the FTP/S partner

Each of the customer's FTP/S partners uses their FTP client software that supports SSL encryption to access File Transfer Service. They complete a FTP/S service questionnaire that collects the following information:

- Company name and contact information
- FTP/S service details, such as FTP transfer mode and type of data to be transferred
- Access method, such as by Internet or VPN connection

After IBM configures the trading partner's account using information collected on the questionnaire, IBM provides the trading partner with a URL, as well as a user ID and password, to connect to the File Transfer Service. The trading partner then works together with IBM to test the connection to File Transfer Service.

Onboarding the S/FTP partner

Each of the customer's S/FTP partner uses their S/FTP client software to access File Transfer Service and they complete an S/FTP service questionnaire which collects the following information:

- Company name and contact information
- S/FTP service details such as S/FTP client software, authentication method and type of data to be transferred
- Access method such as by Internet or VPN connection

After IBM configures the trading partner's account using information collected on the questionnaire, IBM provides the trading partner with a URL and a user ID and password to connect to the File Transfer Service. The trading partner then works together with IBM to test the connection to File Transfer Service.

8.1.4 Testing the scenario

For our testing, the customer uses FileZilla to connect to File Transfer Service over FTP/S protocol to send data to trading partners. All the trading partners also use FileZilla to connect to File Transfer Service over FTP, FTP/S, or S/FTP protocol to receive data from the customer.

Sending data to the FTP partner

Follow these steps to send a file to an FTP partner:

1. The customer connects to File Transfer Service using FileZilla. See Figure 8-2.

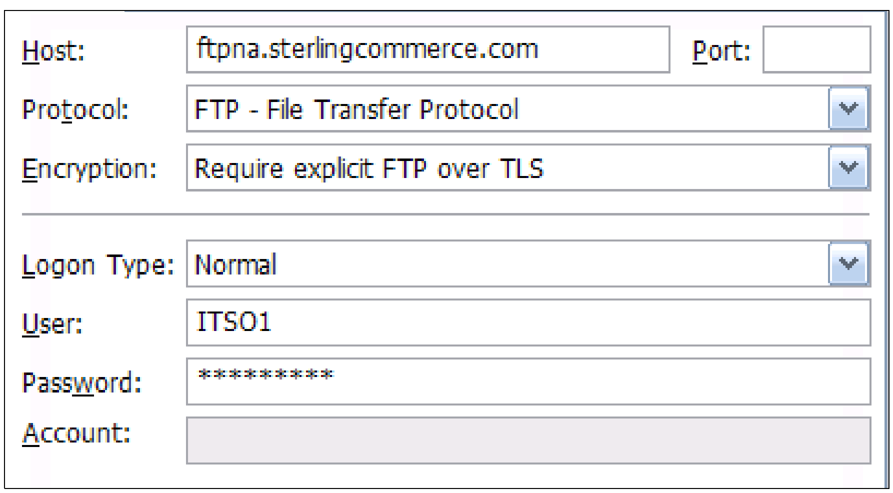


Figure 8-2 Connection parameters

2. The Customer drags the file to the COMMIT folder to upload the file (Figure 8-3). In our scenario, File Transfer Service is configured to route files based on their names.

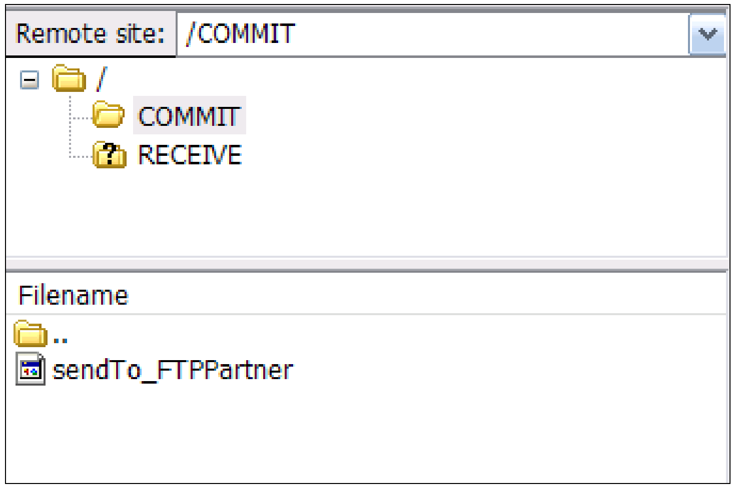


Figure 8-3 Uploading file using FileZilla

3. The FTP partner connects to File Transfer Service using FileZilla. See Figure 8-4.

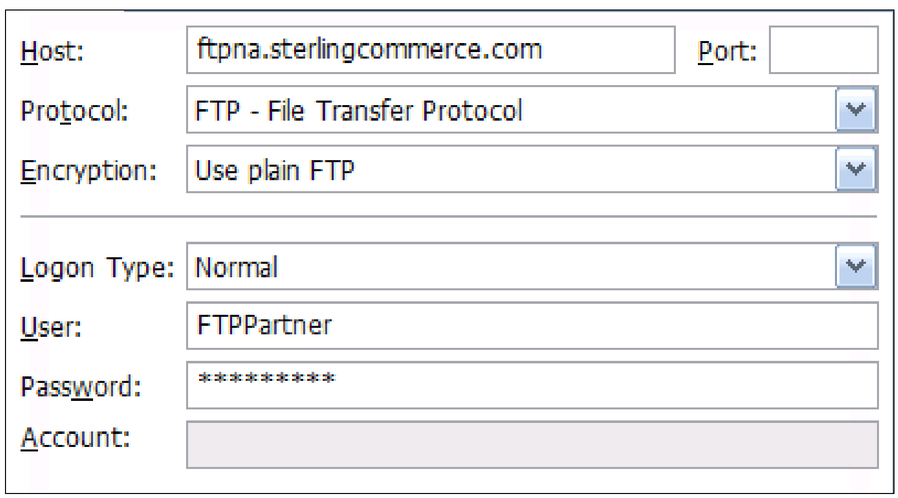


Figure 8-4 Uploading file using FileZilla

4. The FTP partner drags the file from the RECEIVE folder to download the file. See Figure 8-5.

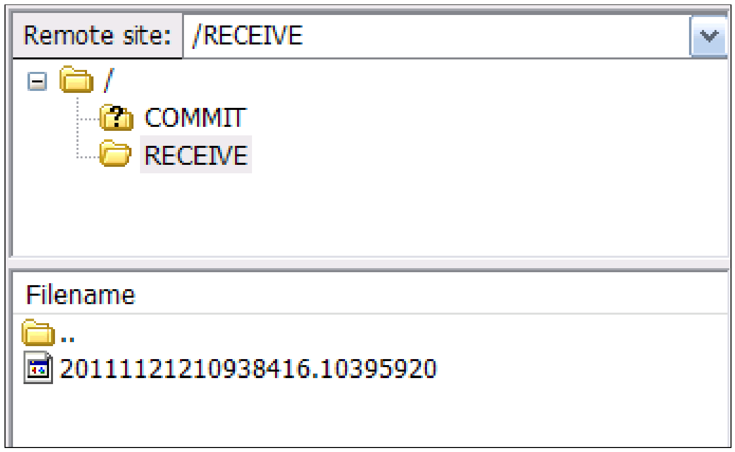


Figure 8-5 Downloading file using FileZilla

Sending data to the FTP/S partner

Follow these steps to send a file to an FTP/S partner:

1. The customer connects to File Transfer Service using FileZilla and drags the file to the COMMIT folder to upload the file. See Figure 8-6.

Figure 8-6 Uploading file using FileZilla

2. The FTP/S partner connects to File Transfer Service using FileZilla. See Figure 8-7.

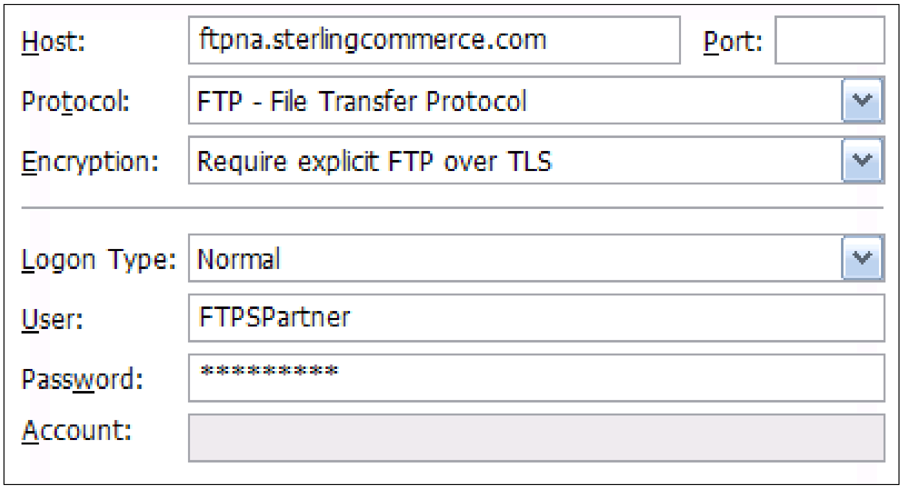


Figure 8-7 Connection parameters

3. The FTP/S partner drags the file from the RECEIVE folder to download the file. See Figure 8-8.

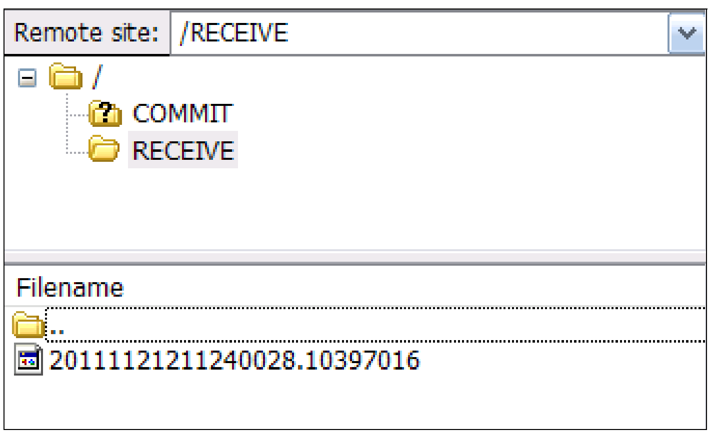


Figure 8-8 Downloading file using FileZilla

Sending data to the S/FTP partner

Use the following steps to send a file to an S/FTP partner:

1. The customer connects to File Transfer Service using FileZilla and drags the file to the COMMIT folder to upload the file. See Figure 8-9.

Figure 8-9 Uploading file using FileZilla

2. The S/FTP partner connects to File Transfer Service using FileZilla. See Figure 8-10.

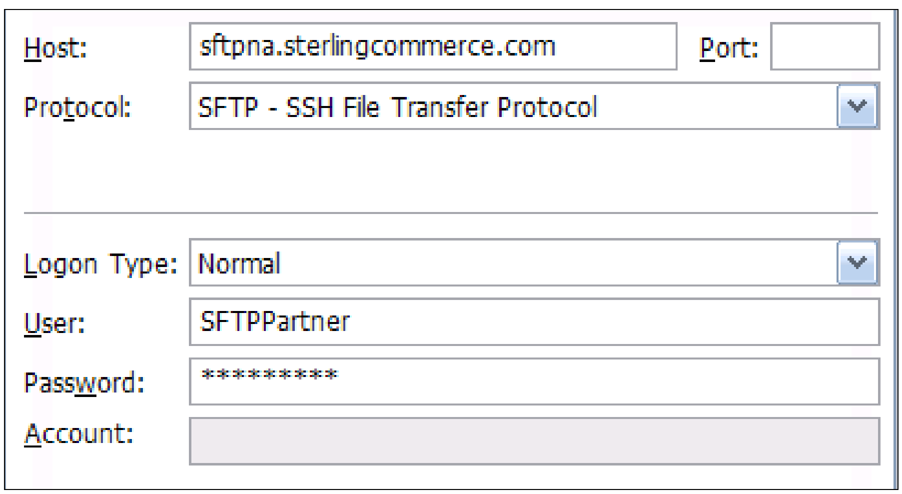


Figure 8-10 Connection parameters

3. The S/FTP partner drags the file from the RECEIVE folder to download the file. See Figure 8-11.

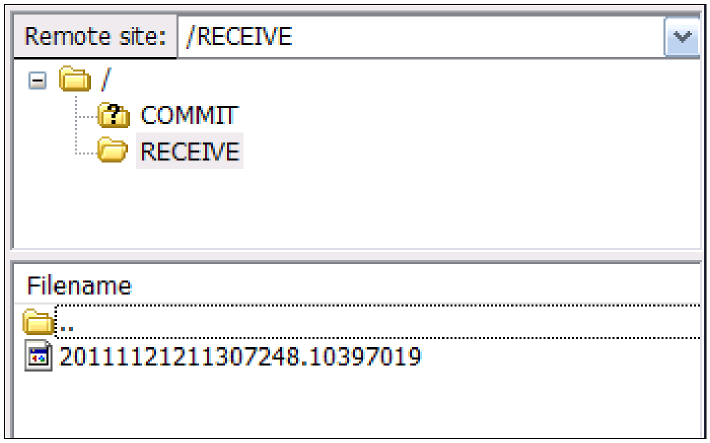


Figure 8-11 Downloading file using FileZilla

InFlight document tracking

IBM Sterling InFlight Data Management is a visibility tool presenting a web interface for tracking documents and generating reports. It tracks data associated with several services in IBM Sterling B2B Collaboration Network.

In our scenario, the customer uses Inflight to track all of the documents sent to trading partners (Figure 8-12) and the overall status of each document (Figure 8-13).

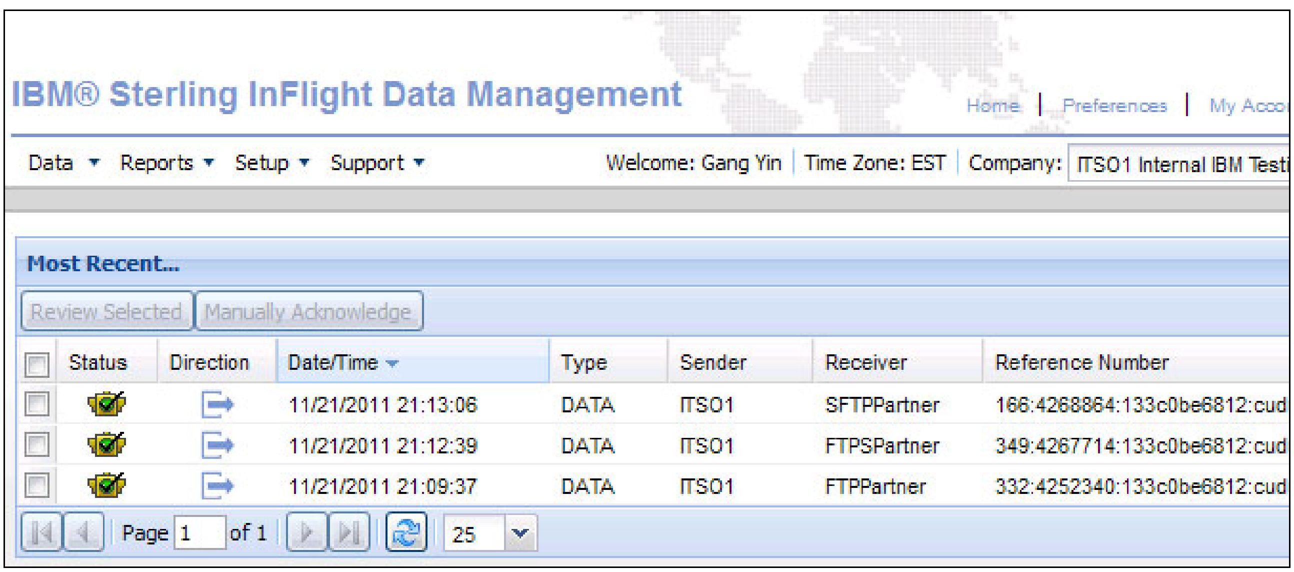


Figure 8-12 Most recent documents

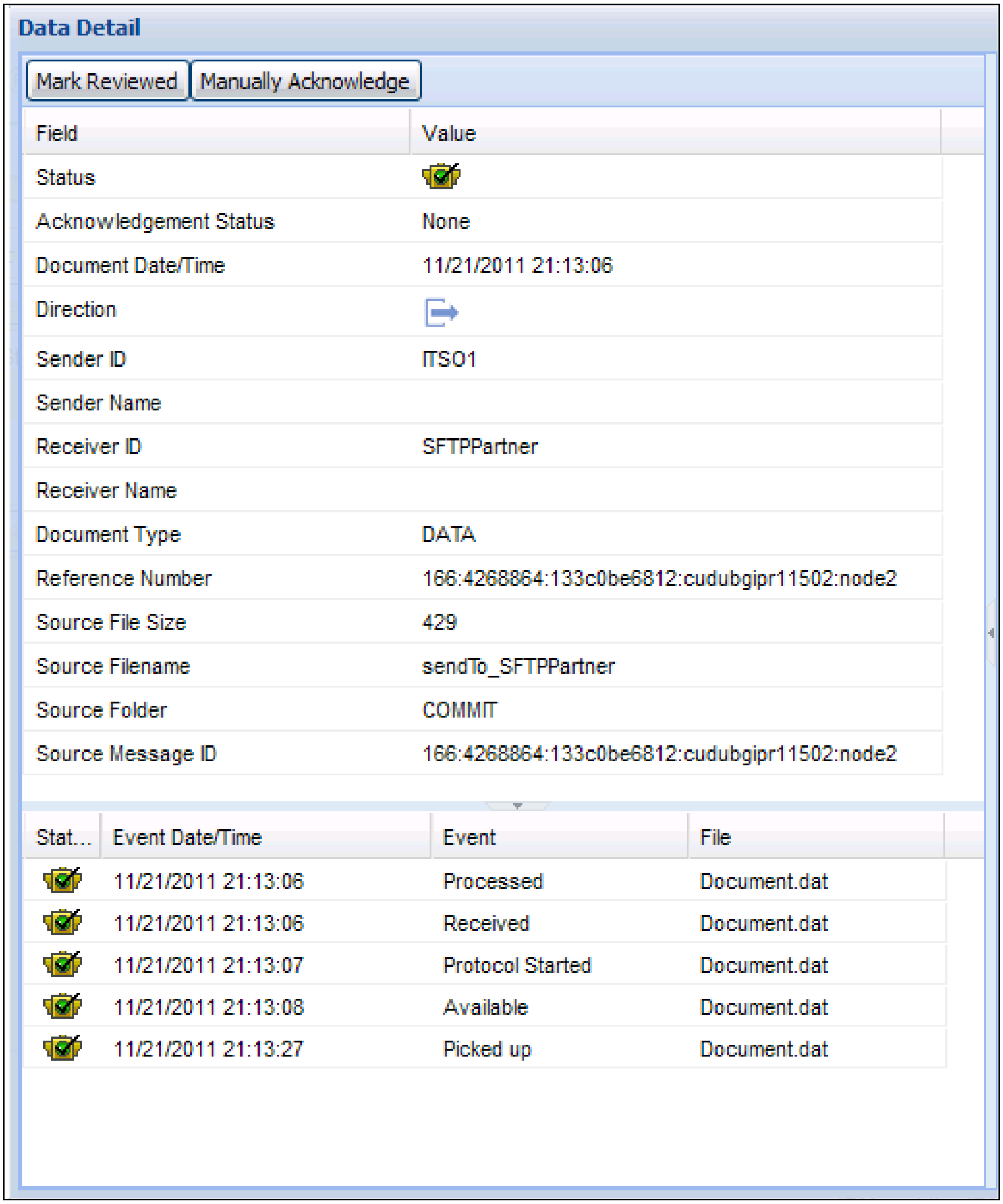


Figure 8-13 Document details

8.1.5 Conclusion

This scenario demonstrates how to onboard and use IBM Sterling File Transfer Service to transfer Non-EDI files with trading partners. The onboard process is quick and simple. All the customer needs to do is to provide connection information and configure its client software to connect to the cloud service. IBM takes care of the configuration for both the customer and its trading partners. The file to be transferred can be larger than one gigabyte. A wide range of transfer protocols is supported, including FTP(S), SFTP, OFTP2, AS2, and Connect:Direct. And as a centralized visibility tool, IBM Sterling InFlight Data Management helps the customer to keep track of their files.

8.2 EDI-based B2B cloud scenario

The second scenario that we cover in this chapter is the EDI-based B2B cloud scenario.

8.2.1 Business value

ITSOSupplyChainCloud is a company in which 50% of its suppliers do not have the capability to handle EDI transactions. They are still using phone, fax, e-mail, and the postal service to receive purchase orders (POs), and to send advance ship notices (ASNs) and invoices.

The company conducted a study and revealed that the average cost of processing a paper document is $10, regardless of volume per partner. In order to maximize ROI of their investment, they must either automate more documents or more partners. They have done all they can to compel suppliers to use EDI, but these remaining suppliers have their own business reasons for not operating their own EDI systems.

Through surveying their Non-EDI suppliers, they find that half of them are willing to use a web-enabled method, if sponsored by their customer, to receive POs and send ASNs and invoices.

The business areas agree to fund this initiative to deploy IBM Sterling Web Forms. They are looking for the following business benefits:

- Faster turnaround on important business documents (POs, order acknowledgements, ASNs, invoices, and so on.)
- More accurate data sent and received, no re-keying, fewer errors, less rework
- Reduced labor, transaction, and operations costs
- Improved efficiencies throughout their supply chain
- Improved internal business processes

8.2.2 Presenting the scenario

In this scenario, we onboard ITSOSupplyChainCloud to use IBM Sterling B2B Collaboration Network, and its EDI trading partners are onboarded on the other side. We also onboard Non-EDI trading partners of ITSOSupplyChainCloud to use IBM Sterling Web Forms.

We establish one discreet AS2 connection to bring ITSOSupplyChainCloud up to IBM Sterling B2B Collaboration Network. Each of its EDI trading partners also have one discreet AS2 connection to IBM Sterling B2B Collaboration Network. Each of its Non-EDI trading partners uses a web browser to access IBM Sterling Web Forms to exchange business documents electronically over the Internet with the customer.

Figure 8-14 shows the connectivity choice of this scenario in a high-level perspective.

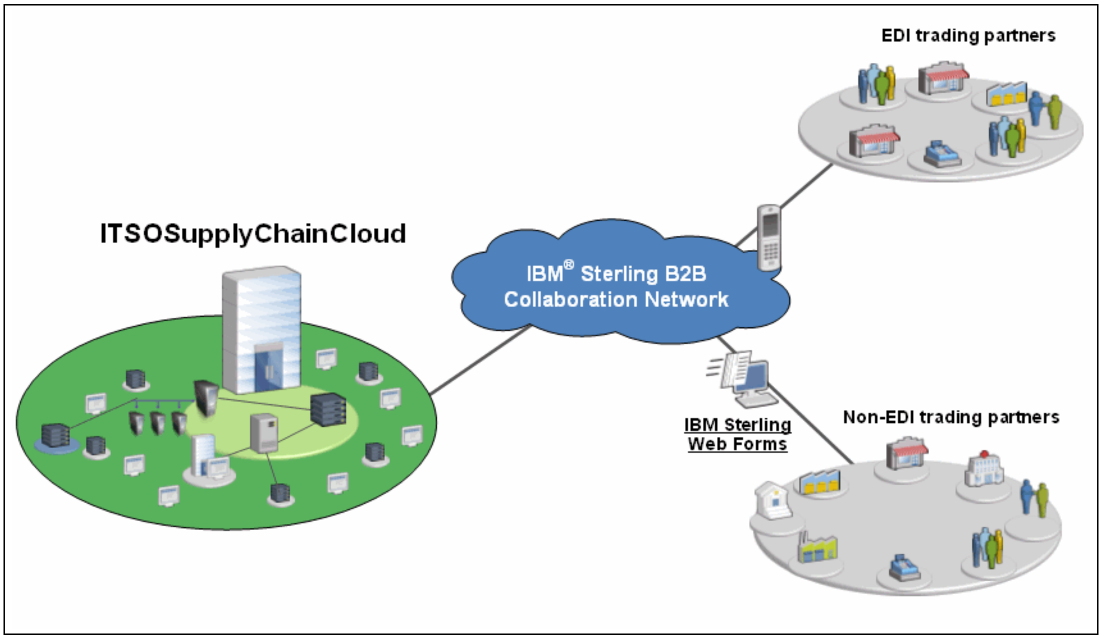


Figure 8-14 Scenario overview

8.2.3 Configuring the scenario

After depicting at a high level the connectivity choice of the scenario, we explain in detail the onboard process of both Sterling B2B Collaboration Network and Web Forms.

Onboarding of IBM Sterling B2B Collaboration Network

Onboarding the customer and EDI trading partners in this scenario to use Sterling B2B Collaboration Network is almost the same as the onboarding process of IBM Sterling File Transfer Service in the first scenario of the chapter. However, the two cloud services use separate connections.

The customer and EDI trading partners use their AS2 client software to connect to Sterling B2B Collaboration Network. They complete an AS2 service questionnaire that collects the following information:

- Company name and contact information
- AS2 service details such as AS2 client software, AS2 identifier, AS2 certificate and AS2 endpoint address
- Access method such as by Internet or VPN connection

After IBM configures their accounts using information collected on the questionnaire, IBM provides them with an AS2 identifier, an AS2 certificate and an AS2 endpoint address of Sterling B2B Collaboration Network that they enter into their AS2 client software to connect to the cloud service. IBM also provides them with accounts of the visibility tool, which is called IBM Sterling Document Tracking. They then work together with IBM to test the connection.

Onboarding of IBM Sterling Web Forms

Sterling Web Forms uses automated online registration to enable faster partner onboarding. The customer sponsors a Sterling Web Forms community and sends a community password to its Non-EDI trading partners so that they can register to join the community. See Figure 8-15.

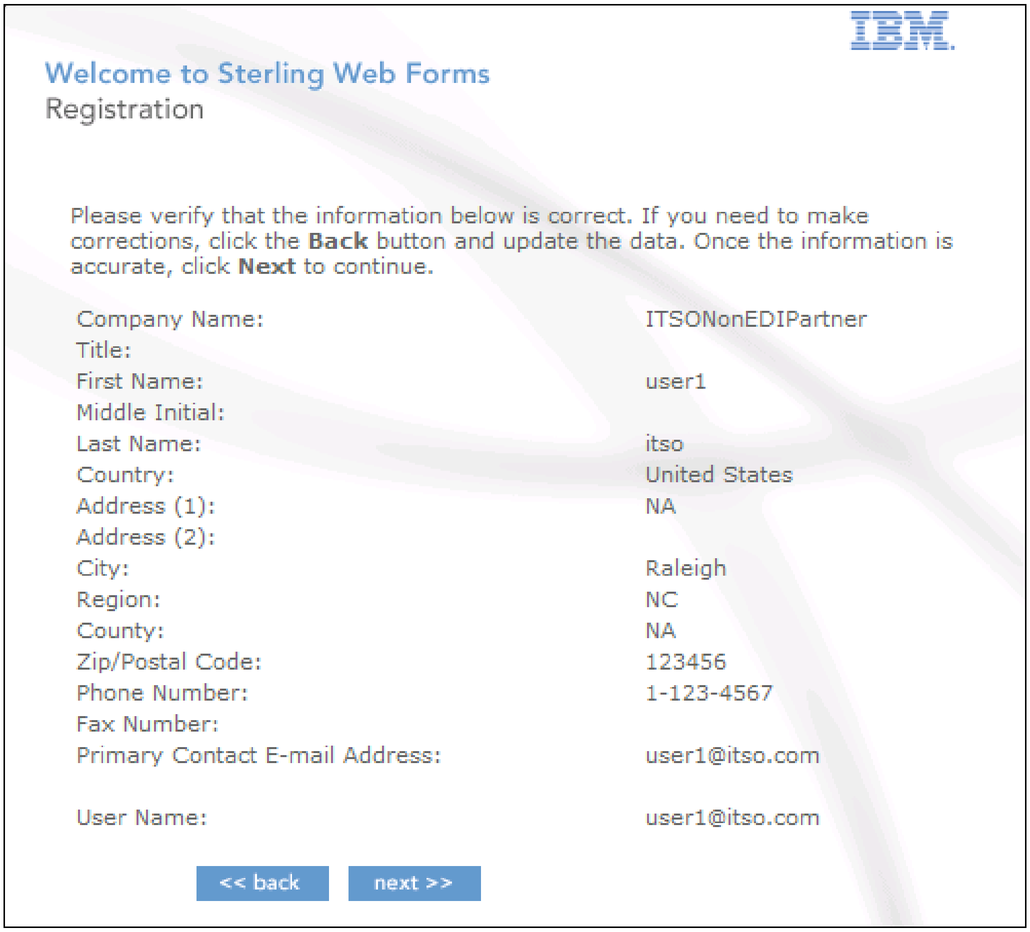


Figure 8-15 Web Forms registration

8.2.4 Testing the scenario

For our testing, the customer and EDI partners use IBM Sterling B2B Integrator to connect to Sterling B2B Collaboration Network over AS2 protocol to send and receive EDI document. Non-EDI partners use a web browser to access IBM Sterling Web Forms site to receive electronic document translated from EDI.

Sending data to an EDI partner

Follow these steps:

1. The customer puts the EDI document into the outbound folder on its file system. The Sterling B2B Integrator automatically picks up the document and sends it to Sterling B2B Collaboration Network. See Figure 8-16.

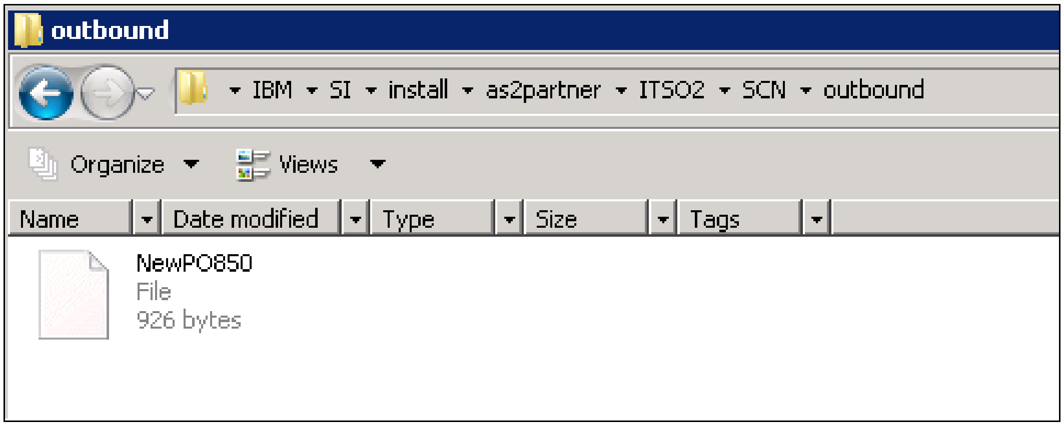
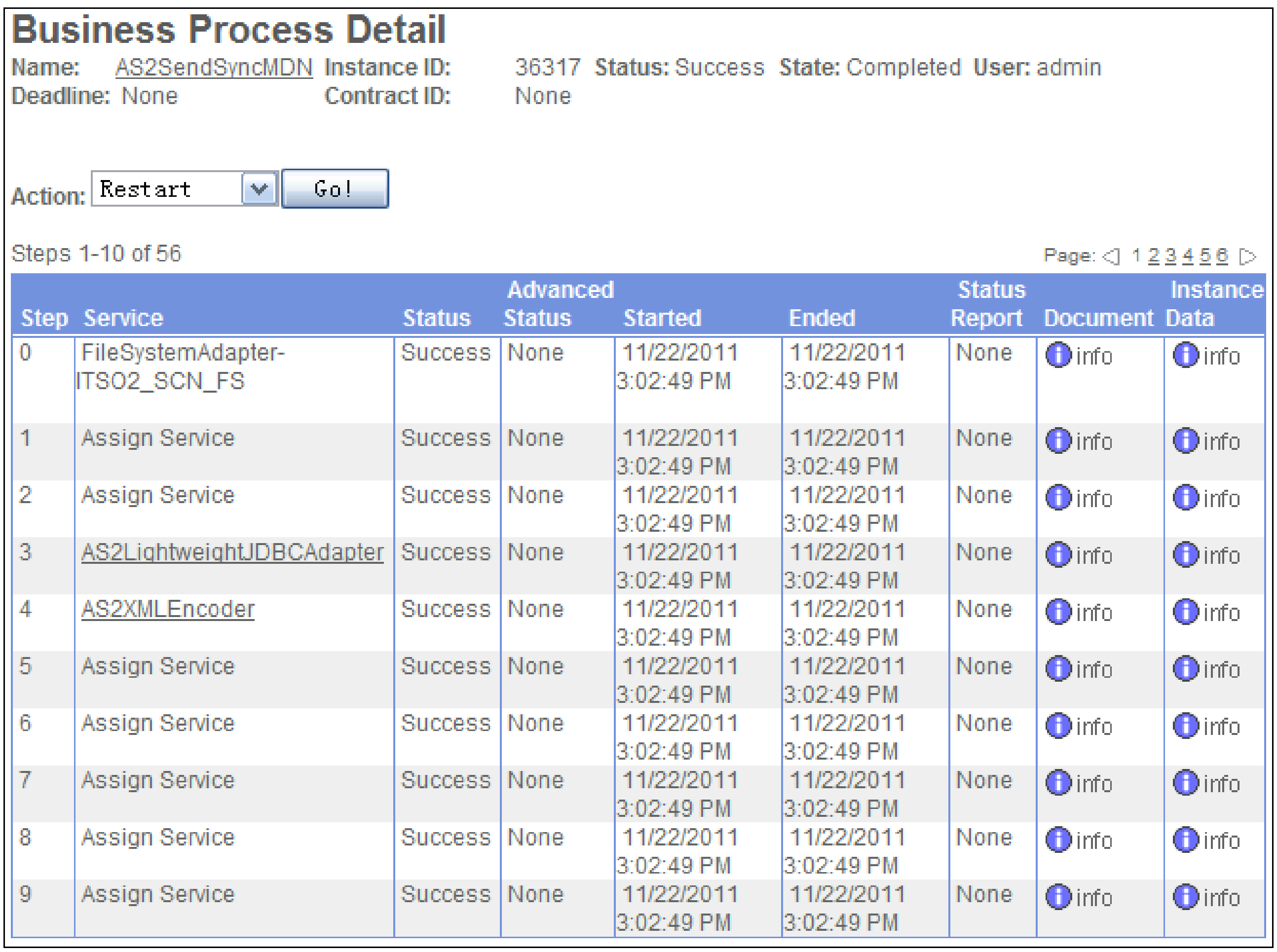


Figure 8-16 Outbound folder of customer

2. The customer uses the dashboard of IBM Sterling B2B Integrator to track the business process that sends the EDI document to Sterling B2B Collaboration Network (Figure 8-17) and Message Disposition Notification (MDN) sent back from the Sterling B2B Collaboration Network (Figure 8-18).



Business Process Detail

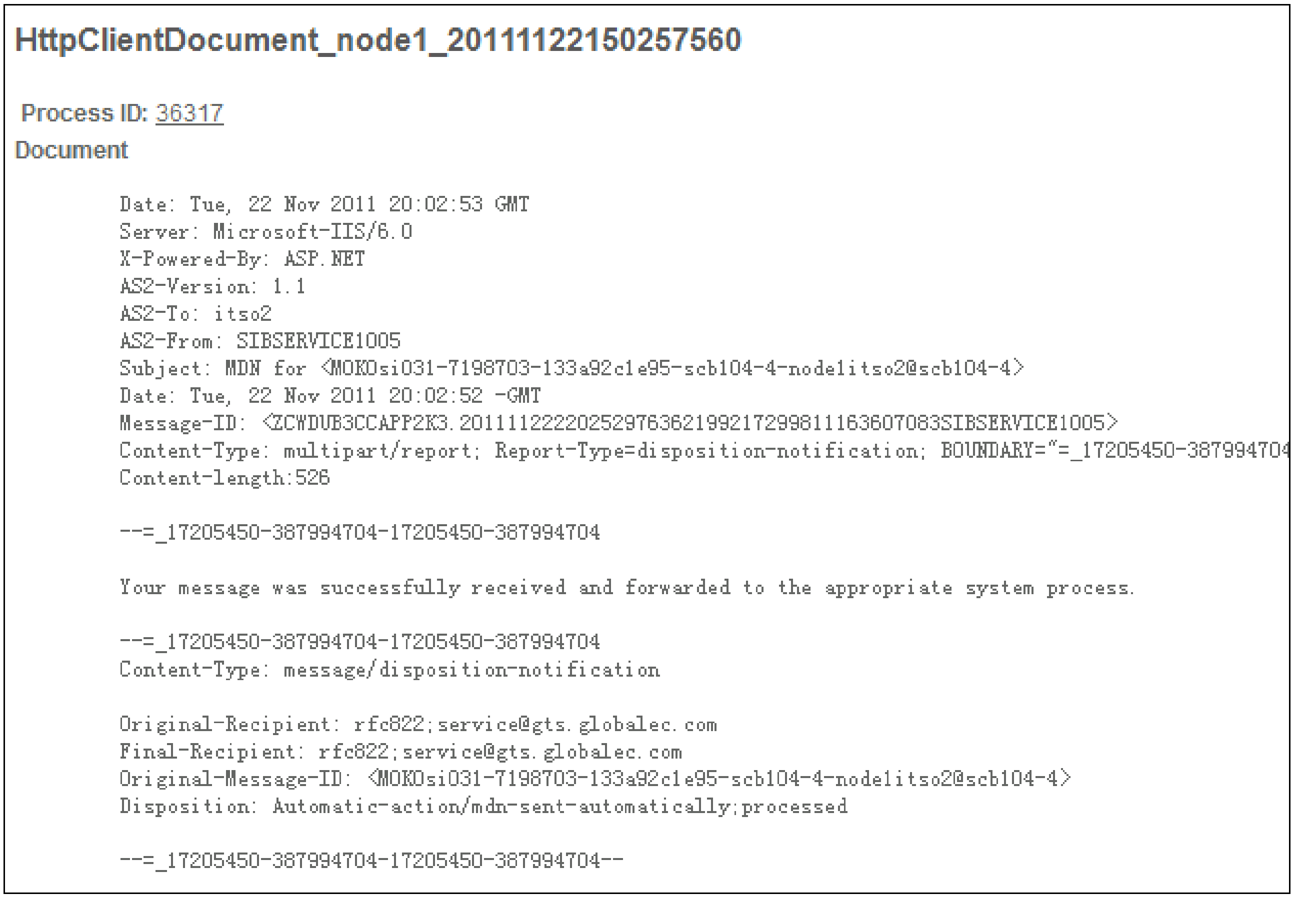
Name: AS2SendSyncMDN **Instance ID:** 36317 **Status:** Success **State:** Completed **User:** admin
Deadline: None **Contract ID:** None

Action: Restart Go!

Steps 1-10 of 56 Page: 1 2 3 4 5 6

Step	Service	Status	Advanced Status	Started	Ended	Status Report	Document	Instance Data
0	FileSystemAdapter-ITSO2_SCN_FS	Success	None	11/22/2011 3:02:49 PM	11/22/2011 3:02:49 PM	None	info	info
1	Assign Service	Success	None	11/22/2011 3:02:49 PM	11/22/2011 3:02:49 PM	None	info	info
2	Assign Service	Success	None	11/22/2011 3:02:49 PM	11/22/2011 3:02:49 PM	None	info	info
3	AS2LightweightJDBCAdapter	Success	None	11/22/2011 3:02:49 PM	11/22/2011 3:02:49 PM	None	info	info
4	AS2XMLEncoder	Success	None	11/22/2011 3:02:49 PM	11/22/2011 3:02:49 PM	None	info	info
5	Assign Service	Success	None	11/22/2011 3:02:49 PM	11/22/2011 3:02:49 PM	None	info	info
6	Assign Service	Success	None	11/22/2011 3:02:49 PM	11/22/2011 3:02:49 PM	None	info	info
7	Assign Service	Success	None	11/22/2011 3:02:49 PM	11/22/2011 3:02:49 PM	None	info	info
8	Assign Service	Success	None	11/22/2011 3:02:49 PM	11/22/2011 3:02:49 PM	None	info	info
9	Assign Service	Success	None	11/22/2011 3:02:49 PM	11/22/2011 3:02:49 PM	None	info	info

Figure 8-17 AS2SendSyncMDN business process



HttpClientDocument_node1_20111122150257560

Process ID: 36317

Document

```
Date: Tue, 22 Nov 2011 20:02:53 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
AS2-Version: 1.1
AS2-To: itso2
AS2-From: SIBSERVICE1005
Subject: MDN for <MOKOsi031-7198703-133a92c1e95-scb104-4-node1itso2@scb104-4>
Date: Tue, 22 Nov 2011 20:02:52 -GMT
Message-ID: <ZCWDUB3CCAPP2K3.20111122202529763621992172998111636O7083SIBSERVICE1005>
Content-Type: multipart/report; Report-Type=disposition-notification; BOUNDARY="=_17205450-387994704
Content-length:526

--=_17205450-387994704-17205450-387994704

Your message was successfully received and forwarded to the appropriate system process.

--=_17205450-387994704-17205450-387994704
Content-Type: message/disposition-notification

Original-Recipient: rfc822;service@gts.globalec.com
Final-Recipient: rfc822;service@gts.globalec.com
Original-Message-ID: <MOKOsi031-7198703-133a92c1e95-scb104-4-node1itso2@scb104-4>
Disposition: Automatic-action/mdn-sent-automatically;processed

--=_17205450-387994704-17205450-387994704--
```

Figure 8-18 MDN receipt

3. The EDI partner receives the document with Sterling B2B Integrator and it is extracted into the inbound folder on EDI partner's file system. See Figure 8-19.

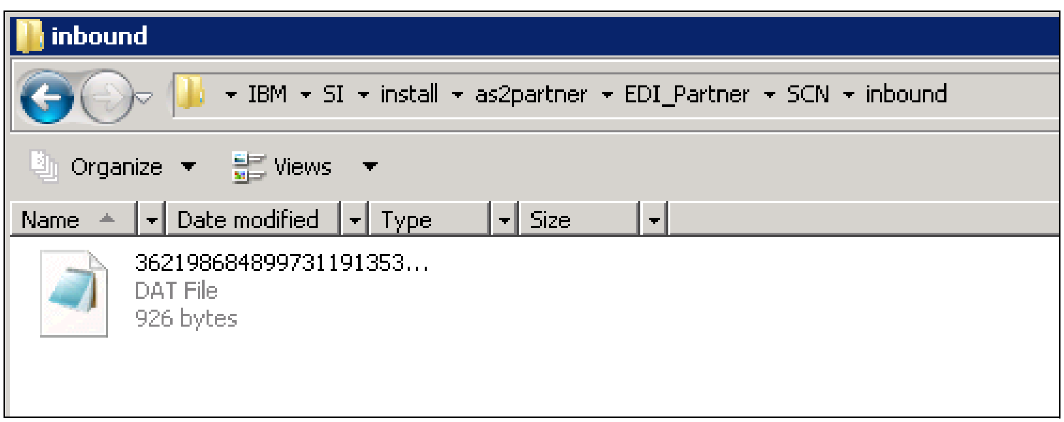


Figure 8-19 nbound folder of EDI partner

4. The customer uses IBM Sterling Document Tracking to track the EDI document in the outbox of its mailslot (Figure 8-20). The EDI partner also uses IBM Sterling Document Tracking to track the EDI document in the inbox of its mailslot (Figure 8-21).

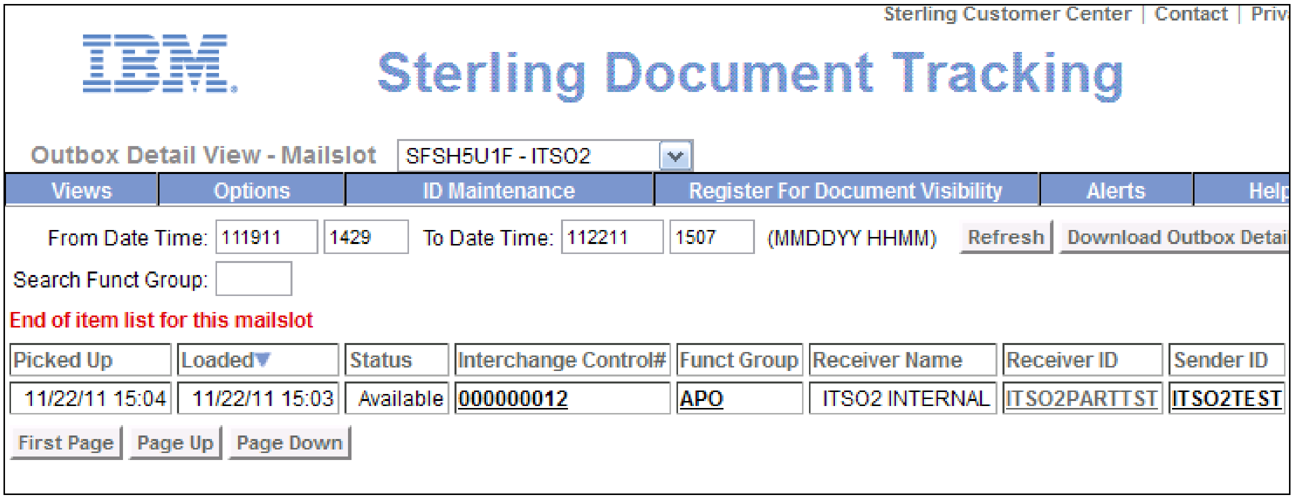


Figure 8-20 Outbox view of customer's mailslot

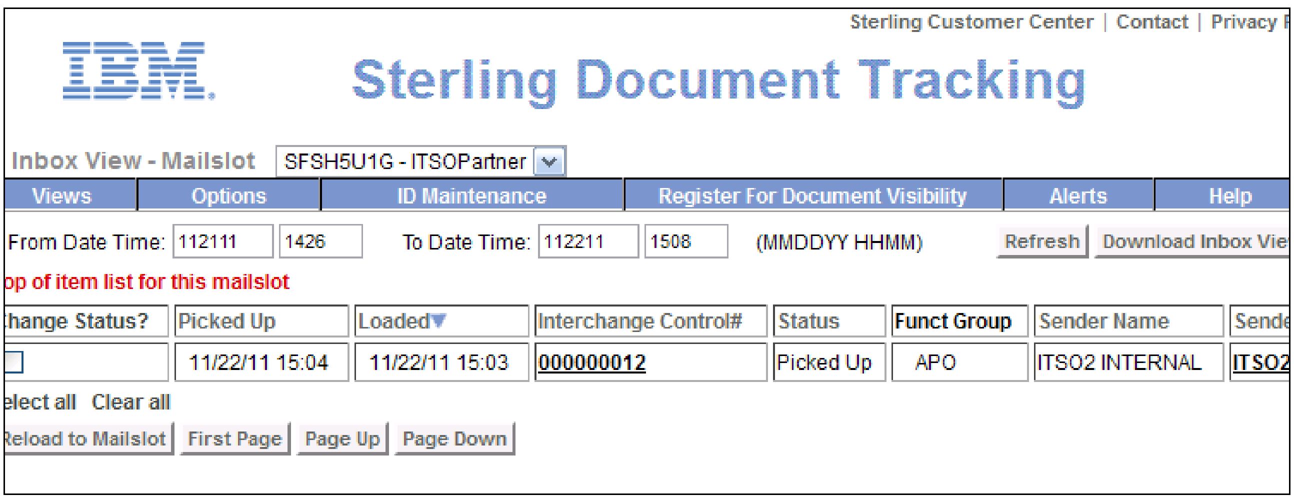


Figure 8-21 Inbox view of partner's mailslot

Sending data to a Non-EDI partner

1. The customer puts the EDI document into the outbound folder on its file system. The Sterling B2B Integrator automatically picks up the document and sends it to the Sterling B2B Collaboration Network. The EDI document is shown in Example 8-1.

Example 8-1 EDI purchase order

```
ISA*00*          *00*          *ZZ*ITSO2TEST      *ZZ*ITSO2PART2TST
*111121*1145*U*00401*000000012*0*T*>~
GS*PO*ITSO2TEST*ITSO2PART2TST*111121*1145*9*X*004010~
ST*850*000000015~
BEG*00*NE*PO0212112250**111121**AE~
CSH*O~
DTM*001*120120***20~
N1*ST*ITSO Non EDI Partner*9*1234567890123~
N3*101 Buy Street*Suite 800~
N4*Buy City*TX*75757*USA~
N1*BT*ITSO Non EDI Partner*9*0987654321098~
N3*PO Box 1000~
N4*Bill City*TX*76767*USA~
PO1*1*50*EA*2.35*SR*UP*100020003001~
CTP******DIS*.85~
PID*F****Pens Ballpoint Fine10PK~
PO1*2*50*EA*2.35*SR*UP*100020003002~
CTP******DIS*.85~
PID*F****Pens Ballpoint Med 10PK~
PO1*3*200*EA*3.5*SR*UP*100020004101~
CTP******DIS*.55~
PID*F****Crayons 48/Box~
PO1*4*100*EA*4.5*SR*UP*100020004102~
CTP******DIS*.55~
PID*F****Crayons 64/Box~
PO1*5*80*EA*5*SR*UP*100040005001~
CTP******DIS*.7~
PID*F****Notebook 5-Subj Spiral~
CTT*5*480~
SE*17*000000015~
GE*1*9~
IEA*1*000000012~
```

2. The Non-EDI partner accesses the IBM Sterling Web Forms website to receive Web Form translated from the EDI document. See Figure 8-22.

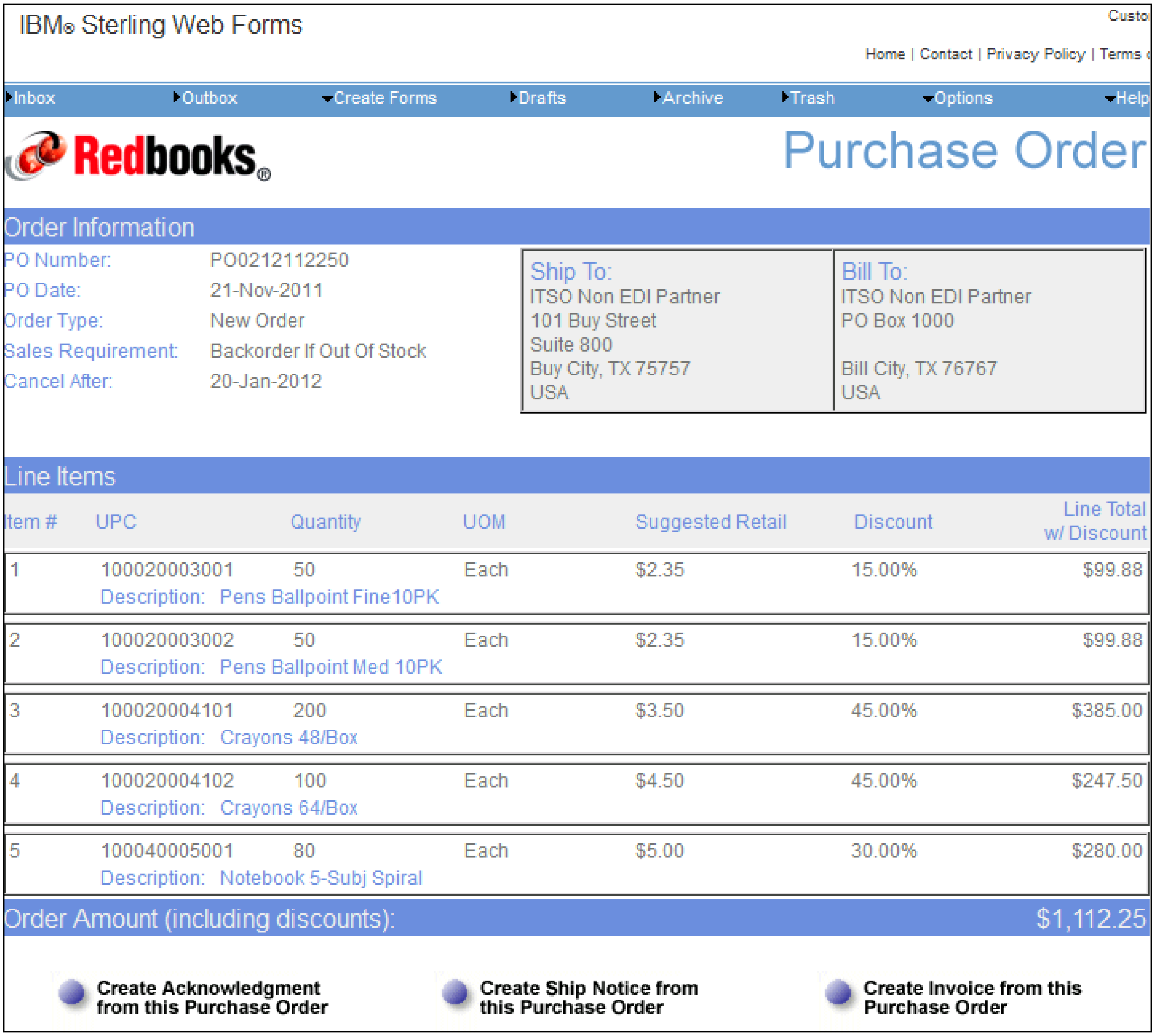


Figure 8-22 IBM Sterling Web Forms

8.2.5 Conclusion

This scenario demonstrates how to onboard and use IBM Sterling B2B Collaboration Network and IBM Sterling Web Forms to transfer EDI files with trading partners, regardless of whether a trading partner has EDI capability.

For customers who wants to automate its EDI transactions with Non-EDI partners, community services hosted in Sterling B2B Collaboration Network such as IBM Sterling Web Forms and IBM Sterling Fax Conversion Services are provided. The customer sponsors a community and invites its Non-EDI trading partners to register to join the community. Then the customer connects to the Sterling B2B Collaboration Network to exchange EDI documents with its Non-EDI partners and use IBM Sterling Document Tracking to keep track of its documents. In this scenario, Non-EDI partners use IBM Sterling Web Forms to receive and send electronic documents in HTML format. Translation between EDI documents and Web Forms is automatically done behind the scene.

A

Additional material

This book refers to additional material that can be downloaded from the Internet as described in the following sections.

Locating the Web material

The Web material associated with this book is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser at:

ftp://www.redbooks.ibm.com/redbooks/SG247992

Alternatively, you can go to the IBM Redbooks website at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the IBM Redbooks form number, SG247992.

Using the Web material

The additional Web material that accompanies this book includes the following files:

File name	*Description*
SCS_DataPowerArtifacts.zip	Contains DataPower Import file to fully configure the simulated partner, the X.509 certificates used in the profiles, and the EDI test file.
SCS_WTXArtifacts1.zip	Contains the compliance artifacts needed to execute the X12 compliance maps. When extracted, the zip will create the directory structure, and put all configuration files and map files in the proper location. The zip should be extracted at the root directory of the server that will host the Sterling B2B Integrator.
SCS_WTXArtifacts2.zip	Contains the Project Interchange needed to create the Extender Project needed to build the maps needed for this scenario. From within WebSphere Transformation Extender Design Studio, import this zip as a Project Interchange.
Chapter5_WMB_HealthCareESB.zip	Contains the WebSphere MQ Project Interchange that we created for Chapter 5, “Health Insurance scenario” on page 103.
Chapter6_WMB_Finance.zip	Contains the WebSphere MQ Project Interchange that we created for Chapter 6, “Financial Services scenario” on page 177.

System requirements for downloading the Web material

The Web material requires the following system configuration:

Hard disk space: 10 MB minimum
Operating System: Windows/Linux

Downloading and extracting the Web material

Create a subdirectory (folder) on your workstation, and extract the contents of the Web material .zip file into this folder.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this book. Note that some publications referenced in this list might be available in softcopy only.

- *IBM WebSphere DataPower B2B Appliance XB60 Revealed*, SG24-7745

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- ANSI ASC X12 EDI standard:

  http://www.x12.org/x12org/about/X12History.cfm
- FAQ site on different options in implementing an ESB solution:

  http://www-01.ibm.com/software/integration/wsesb/v6/faqs.html#provide
- IBM Sterling Connect:Direct File Agent Configuration Guide:

  http://www.sterlingcommerce.com/documentation/home/MFT/ConnectDirect/File%20Agent/CD%20FileAgent14/CDFAConfigGuide.pdf
- IBM Sterling Connect:Direct Process Language Reference Guide:

  http://www.sterlingcommerce.com/documentation/home/MFT/ConnectDirect/File%20Agent/CD%20FileAgent14/CDFAConfigGuide.pdf
- IBM DeveloperWorks articles on message based integration choices:

  https://www.ibm.com/developerworks/wikis/display/esbpatterns/Message-based+Integration
  http://www.ibm.com/developerworks/library/ws-enterpriseconnectivitypatterns/index.html
- “What’s new in Version 8.0” section in the WebSphere Message Broker V8.0 Information Center:

  http://publib.boulder.ibm.com/infocenter/wmbhelp/v8r0m0/topic/com.ibm.etools.mft.doc/bb23110_.htm
- “Which XML parser should you use” in the WebSphere Message Broker Information Center:

  http://publib.boulder.ibm.com/infocenter/wmbhelp/v8r0m0/index.jsp?topic=/com.ibm.etools.mft.doc/ad70530_.htm

- IBM Developer Works articles on WebSphere Transformation Extender:

  http://www-01.ibm.com/software/integration/wdatastagetx/
- WebSphere MQ Information Center, WebSphere MQ clusters:

  http://publib.boulder.ibm.com/infocenter/wmqv6/v6r0/index.jsp?topic=%2Fcom.ibm.mq.csqzah.doc%2Fqc10120_.htm
- IBM Sterling Connect:Direct Installation Guide:

  http://www.sterlingcommerce.com/documentation/home/MFT/ConnectDirect/Windows/CD%20Windows46/CDWindowsGettingStarted.pdf
- IBM Sterling Connect:Direct File Agent Configuration Guide:

  http://www.sterlingcommerce.com/documentation/home/MFT/ConnectDirect/File%20Agent/CD%20FileAgent14/CDFAConfigGuide.pdf
- Sterling Secure Proxy 3.4 documentation:

  http://www.sterlingcommerce.com/documentation/home/MFT/SSP/SSP.html

Help from IBM

IBM Support and downloads:

ibm.com/support

IBM Global Services:

ibm.com/services

End-to-end Integration with IBM Sterling B2B Integration and Managed File Transfer Solutions

(0.5" spine)
0.475"<->0.873"
250 <-> 459 pages

End-to-end Integration with IBM Sterling B2B Integration and Managed File Transfer Solutions

Implement an end-to-end integration with IBM Sterling and WebSphere Portfolios

Learn how to design a B2B solution for small and large partners

Experiment with real life scenarios

Across numerous vertical industries, enterprises are challenged to improve processing efficiency as transactions flow from their business communities to their internal systems and vice versa, simplify management and expansion of the external communities, accommodate customer and supplier preferences, govern the flow of information, enforce policy and standards, and protect sensitive information. Throughout this process, external partners must be on-boarded and off-boarded, information must flow across multiple communications infrastructures, and data must be mapped and transformed for consumption across multiple applications.

Some transactions require synchronous or real-time processing while others are of a more periodic nature. For some classes of customer or supplier, the enterprise might prefer a locally-managed, on-premise solution. For some types of communities (often small businesses), an as-a-Service solution might be the best option. Many large enterprises combine the on-premise and as-a-Service approach to serve different categories of business partners (customers or suppliers).

This IBM Redbooks publication focuses on solutions for end-to-end integration in complex value chains and presents several end-to-end common integration scenarios with IBM Sterling and IBM WebSphere portfolios.

We believe that this publication will be a reference for IT Specialists and IT Architects implementing an integration solution architecture involving IBM Sterling and IBM WebSphere portfolios.

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

SG24-7992-00 ISBN 0738436925