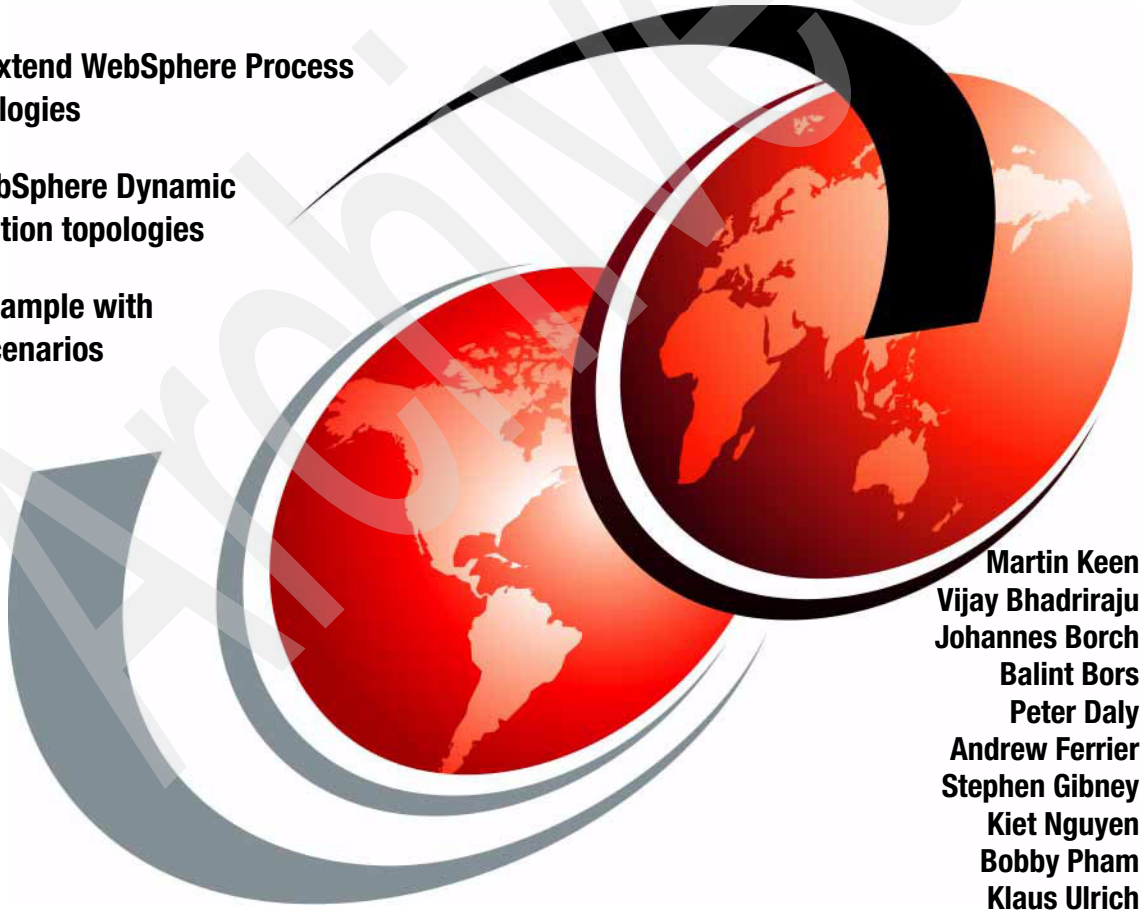
WebSphere. software

IBM

WebSphere Business Process Management V7 Production Topologies

- Build and extend WebSphere Process Server topologies
- Explore WebSphere Dynamic Process Edition topologies
- Learn by example with practical scenarios



Martin Keen
Vijay Bhadriraju
Johannes Borch
Balint Bors
Peter Daly
Andrew Ferrier
Stephen Gibney
Kiet Nguyen
Bobby Pham
Klaus Ulrich

Redbooks

ibm.com/redbooks

Archive

Archived

International Technical Support Organization

WebSphere Business Process Management V7 Production Topologies

May 2010

Archived

SG24-7854-00

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

Archived

First Edition (May 2010)

This edition applies to WebSphere Process Server V7.0.0.1, WebSphere Business Services Fabric V7.0.0.0, and WebSphere Business Monitor V7.0.0.0.

© Copyright International Business Machines Corporation 2010. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contact an IBM Software Services Sales Specialist

Start SMALL, Start BIG, ... **JUST START**

architectural knowledge, skills, research and development . . .

that's IBM Software Services for WebSphere.

Our highly skilled consultants make it easy for you to design, build, test and deploy solutions, helping you build a smarter and more efficient business. **Our worldwide network of services specialists wants you to have it all!** Implementation, migration, architecture and design services: IBM Software Services has the right fit for you. We also deliver just-in-time, customized workshops and education tailored for your business needs. You have the knowledge, now reach out to the experts who can help you extend and realize the value.

For a WebSphere services solution that fits your needs, contact an IBM Software Services Sales Specialist:
ibm.com/developerworks/websphere/services/contacts.html

Archive

Archived

Contents

Contact an IBM Software Services Sales Specialist iii

Notices xiii
Trademarks xiv

Preface xv
The team who wrote this book xv
Now you can become a published author, too! xviii
Comments welcome xix
Stay connected to IBM Redbooks xix

Part 1. Overview 1

Chapter 1. Business Process Management products overview 3
1.1 WebSphere Dynamic Process Edition 6
1.1.1 Discover, model, and simulate 7
1.1.2 Deploy, execute, and change 7
1.1.3 Monitor, analyze, predict, and act 7
1.1.4 Governance and compliance 7
1.2 WebSphere Business Monitor 8
1.3 WebSphere Business Services Fabric 8
1.4 WebSphere Process Server 9
1.5 WebSphere Business Compass 10
1.6 WebSphere Enterprise Service Bus 10

Chapter 2. Business Process Management production topologies 13
2.1 Introduction 14
2.2 Network deployment concepts 15
2.2.1 Components of a WebSphere network deployment environment . . . 15
2.2.2 Clusters 17
2.2.3 Load balancing 20
2.2.4 Failover 20
2.3 Product components 20
2.3.1 Databases and schemas 21
2.3.2 Service integration buses 22
2.3.3 Business Process Choreographer 23
2.3.4 Business Process Choreographer Explorer (BPC Explorer) 23
2.3.5 Business Rules Manager (BRM) 24
2.3.6 Common Event Infrastructure 24

© Copyright IBM Corp. 2010. All rights reserved.

2.3.7 Business Space powered by WebSphere . . . 24
2.3.8 Monitor action services . . . 24
2.3.9 Monitor scheduled services . . . 25
2.3.10 Data movement service . . . 25
2.3.11 Mobile device dashboard . . . 25
2.3.12 AlphaBlox . . . 25
2.3.13 Compass Server Application . . . 25
2.3.14 Publishing Server Application . . . 26
2.3.15 IBM HTTP Server and plug-in . . . 26
2.3.16 WebSphere proxy server . . . 26
2.4 Deployment environment patterns . . . 26
2.4.1 Single Cluster topology pattern . . . 30
2.4.2 Remote Messaging topology pattern . . . 32
2.4.3 Remote Messaging and Remote Support topology pattern . . . 33
2.4.4 Remote Messaging, Support and Web topology pattern . . . 35
2.4.5 Custom topology patterns . . . 37
2.5 Five-cluster topology pattern . . . 38
2.6 Topology choice . . . 41
2.6.1 Single cluster . . . 42
2.6.2 Remote Messaging and Remote Support . . . 43
2.6.3 Double Remote Messaging and Remote Support . . . 44
2.6.4 Remote Messaging, Support and Web . . . 46
2.6.5 Five cluster . . . 46
2.6.6 Messaging engine location and partitioning . . . 46
2.6.7 REST endpoint configuration . . . 47

Part 2. Building production topologies . . . 53

Chapter 3. Preparing your topology . . . 55
3.1 Prerequisite software installation . . . 56
3.2 Software installation . . . 57
3.2.1 Create a software repository . . . 57
3.2.2 Installing WebSphere Business Monitor . . . 61
3.2.3 Installing WebSphere Process Server . . . 70
3.2.4 Installing WebSphere Business Services Fabric . . . 70
3.2.5 Installing WebSphere Dynamic Process Edition . . . 71
3.2.6 Applying the fix packs . . . 72
3.2.7 The all-in-one install and apply fix pack method . . . 78
3.2.8 Installing WebSphere Business Compass . . . 80
3.3 Database and schema creation . . . 82
3.3.1 Using the database design tool . . . 83
3.3.2 The Common database . . . 86
3.3.3 The Business Process Choreographer database . . . 87

Archived

3.3.4 The messaging engine database 87
3.3.5 The business process reporting database 88
3.3.6 The Business Space database 88
3.3.7 The EVENT database 89
3.3.8 The MONITOR database 89
3.3.9 The business compass database 90
3.4 Creating a deployment manager profile 91
3.4.1 Creating a deployment manager profile 91
3.4.2 Configuring LDAP as the user account registry 99
3.4.3 Adding administrative groups 105
3.5 Creating custom profiles 107
3.5.1 Creating a custom profile 107
3.5.2 Federate the custom profile into the cell 110
3.6 Other software installations 110
3.6.1 Installing IBM HTTP Server 111
3.6.2 Add IBM HTTP Server to the cell 111
3.6.3 Installing and configuring WebSphere proxy server 118

Chapter 4. WebSphere Process Server: single cluster 123
4.1 Topology summary 124
4.2 Prerequisites for creating this topology 124
4.3 Configuring topology using deployment environments in Integrated Solutions Console 125
4.4 Configuring topology manually using command line 136
4.5 Post-creation configuration and verification 137
4.5.1 Restarting the deployment manager 137
4.5.2 Creating event database tables 138
4.5.3 Checking database connectivity 138
4.5.4 Change the virtual host hostname 139
4.5.5 Correct REST endpoints 140
4.5.6 Configuring CEI event persistence 140
4.5.7 Configuring CEI event logging 141
4.5.8 Verification: human task manager 142
4.5.9 Verification: business process engine 145
4.5.10 Verification: Business Space 146

Chapter 5. WebSphere Process Server: Remote Messaging and Remote Support 149
5.1 Topology summary 150
5.2 Prerequisites for creating the topology 151
5.3 Configuring the topology using the Integrated Solutions Console 151
5.3.1 Creating the deployment environment topology 152
5.3.2 Generating the deployment environment topology 163

5.4 Configuring the topology manually using the command line 166
5.4.1 Create a new deployment environment 166
5.4.2 Export existing deployment environment definitions 171
5.4.3 Import and generate to create a new deployment environment . . . 172
5.5 Post-creation configuration and verification 175
5.5.1 Configuring deferred configurations 175
5.5.2 Configuring Business Integration security 177
5.5.3 Configuring REST service endpoints 179
5.5.4 Mapping modules to the Web server 183
5.5.5 Configuring host aliases 184
5.5.6 Verifying the Failed Event Manager 185
5.5.7 Disabling CEI event persistence 185
5.5.8 (Optional) Enabling CEI event logging 186
5.5.9 Verifying the deployment environment 187
5.5.10 Verifying Business Process Choreographer 190
5.5.11 Verifying Business Space 192
5.5.12 Verifying human interaction using Business Space 194

Chapter 6. WebSphere Enterprise Service Bus: Remote Messaging and Remote Support 203
6.1 Putting WebSphere Enterprise Service Bus in context 204
6.2 Overview of topology creation 205

Chapter 7. WebSphere Process Server: Double Remote Messaging and Remote Support 207
7.1 Topology summary 208
7.2 Prerequisites for creating this topology 209
7.2.1 Existing Remote Messaging and Remote Support topology 209
7.2.2 Creating an additional BPE database 209
7.2.3 Creating an additional messaging engine database 210
7.3 Configuring the topology using deployment environments in the Integrated Solutions Console 211
7.3.1 Adding the new application and messaging clusters 212
7.3.2 Configuring Service Component Architecture 216
7.3.3 Configuring Business Process Choreographer 221
7.3.4 Setting the authentication aliases for data sources 225
7.3.5 Installing an additional BPC Explorer 227
7.3.6 Configuring the Common Event Infrastructure 231
7.3.7 Configuring the messaging target 234
7.4 Creating new clusters manually using command line 242
7.5 Post-creation configuration and verification 244
7.5.1 Distributing the messaging load among cluster members 244
7.5.2 Verifying the configuration 255

Chapter 8. WebSphere Business Monitor: Remote Messaging and Remote Support topology 259
8.1 Topology summary 260
8.2 Prerequisites for creating this topology 263
8.3 Extending the topology 264
8.3.1 Augment the deployment manager profile 264
8.3.2 Augment the nodes 264
8.4 Configuring the topology using deployment environments in the Integrated Solutions Console 265
8.5 Post-creation configuration and verification 285
8.5.1 Verify server startup without errors 286
8.5.2 Creating event database tables 286
8.5.3 Test data sources 286
8.5.4 Verify virtual host settings and host aliases 287
8.5.5 Test Alphablox admin page 288
8.5.6 Test Business Space 290
8.5.7 Test Business Rule Manager 291
8.5.8 Test the Business Process Choreographer 292
8.5.9 Test the human task manager 292
8.5.10 Test WebSphere Business Monitor 292
8.5.11 Verify the topology using a sample application 296

Chapter 9. WebSphere Business Monitor: cross cell 297
9.1 Topology summary 298
9.2 Prerequisites for creating this topology 301
9.3 Configuring the topology using deployment environments in the Integrated Solutions Console 304
9.4 Configuring topology manually using command line 313
9.5 Post-creation configuration and verification 313
9.5.1 Restarting the deployment manager 314
9.5.2 Verify the WebSphere Business Monitor configuration 315
9.5.3 Checking database connectivity 316
9.5.4 Map Web modules on the HTTP server 316
9.5.5 Update virtual host settings 319
9.5.6 Configure monitor data security 320
9.5.7 Starting the clusters and deployment environment 322
9.5.8 Setup cross-cell queue bypass 325
9.5.9 Verify the topology using a sample application 329

Chapter 10. WebSphere Business Services Fabric: Remote Messaging and zRemote Support 331
10.1 Topology summary 332
10.2 Prerequisites for creating this topology 332

10.3 Configuring topology using deployment environments in Integrated Solutions Console . . . 333
10.4 Configuring topology manually using the command line . . . 345
10.4.1 Creating the deployment environment definition . . . 346
10.4.2 Importing and generating Deployment Environment topology . . . 347
10.5 Post-creation configuration and verification . . . 347
10.5.1 Restarting the deployment manager and nodes . . . 348
10.5.2 Creating event database tables . . . 348
10.5.3 **Checking database connectivity . . . 349**
10.5.4 **Starting the clusters and deployment environment. . . . 349**
10.5.5 Installing and testing a WebSphere Business Services Fabric sample application . . . 352
10.5.6 Business Space for Fabric . . . 358

Chapter 11. WebSphere Dynamic Process Edition . . . 361
11.1 Topology summary . . . 362
11.2 Prerequisites for creating this topology . . . 363
11.3 Configuring topology using deployment environments in Integrated Solutions Console . . . 365
11.3.1 Define and configure a deployment environment . . . 366
11.3.2 Generate the deployment environment . . . 377
11.3.3 Create the Monitor cluster using Integrated Solutions Console . . . 378
11.4 Configuring the topology manually using wsadmin . . . 380
11.5 Create Monitor cluster manually using the command line . . . 382
11.6 Post-creation configuration and verification . . . 383
11.6.1 Verify server startup without errors . . . 383
11.6.2 Creating event database tables . . . 383
11.6.3 Test data sources . . . 383
11.6.4 Test message engines are running . . . 384
11.6.5 Disabling CEI event persistence . . . 385
11.6.6 Test the Alphablox admin page . . . 386
11.6.7 Test Business Space . . . 389
11.6.8 Test the Business Process Choreographer (BPC) . . . 389
11.6.9 Test the Business Rule Manager . . . 389
11.6.10 Test WebSphere Business Services Fabric . . . 390
11.6.11 Test WebSphere Business Monitor . . . 391
11.6.12 Verify the topology using a sample application . . . 392

Chapter 12. WebSphere Business Compass . . . 393
12.1 Topology summary . . . 394
12.2 Prerequisites for creating this topology . . . 395
12.2.1 Prepare your operating system . . . 395
12.2.2 Install database server . . . 396

12.2.3 Install LDAP server . . . 396
12.2.4 Define users in LDAP user registry . . . 396
12.2.5 Install WebSphere Business Compass . . . 396
12.2.6 Create and configure deployment manager . . . 397
12.2.7 Create and federate custom node . . . 397
12.2.8 Install IBM HTTP Server . . . 397
12.2.9 Create databases . . . 398
12.3 Installing WebSphere Business Compass using Integrated Solutions Console . . . 398
12.3.1 Create a cluster . . . 398
12.3.2 Install Business Space . . . 401
12.3.3 Create WebSphere Business Compass and Business Space databases . . . 402
12.3.4 Install WebSphere Business Compass applications . . . 403
12.4 Post-creation configuration and verification . . . 404
12.4.1 Start WebSphere Business Compass . . . 405
12.4.2 Verify database connectivity . . . 406
12.4.3 Install Lotus Forms Turbo . . . 407
12.4.4 Verify installation using Business Space . . . 414

Part 3. Appendixes . . . 419

Appendix A. Additional material . . . 421
Locating the Web material . . . 421
How to use the Web material . . . 421

Appendix B. Verification using the sample application . . . 423
Verification using the sample application . . . 424
Install WebSphere Process Server module . . . 424
Install the monitor model . . . 425
Emit sample events with recorded event management . . . 427
Create a test Business Space . . . 427
Emit sample events with Business Space Choreographer . . . 432
Verify cross-cell event distribution using the sample application . . . 435

Appendix C. Deployment environment definition files . . . 439
Finding the required properties . . . 440
Generating a deployment environment definition . . . 440

Abbreviations and acronyms . . . 443

Related publications . . . 445
IBM Redbooks publications . . . 445
How to get Redbooks . . . 445

Help from IBM 445

Archived

Notices

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which 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.

© Copyright IBM Corp. 2010. All rights reserved.

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:

AlphaBlox®	iSeries®	System z®
DB2®	Lotus®	Tivoli®
developerWorks®	Rational®	WebSphere®
eServer™	Redbooks®	z/OS®
FileNet®	Redpaper™	
IBM®	Redbooks (logo) ®	

The following terms are trademarks of other companies:

Adobe, the Adobe logo, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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

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

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

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

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

Archived

Preface

This IBM® Redbooks® publication describes how to build production topologies for Business Process Management (BPM) solutions. It is aimed at IT Architects and IT Specialists who want to understand and implement these topologies. Use this book to select the appropriate production topologies for a given environment, then follow the step-by-step instructions included in this book to build these topologies. You must have a high-level understanding of WebSphere Business Process Management products to get the most out of this book. This book addresses the following WebSphere® products:

- WebSphere Process Server V7
- WebSphere Business Monitor V7
- WebSphere Business Services Fabric V7
- WebSphere Enterprise Service Bus V7
- WebSphere Business Compass V7

Part 1, “Overview” on page 1, introduces the BPM products that we discuss and provides an overview of basic topology terminology. This part also provides an overview of the production topologies that we describe in this book, including a selection criteria for when to select each topology.

Part 2, “Building production topologies” on page 53, provides a series of step-by-step instructions for creating production topology environments using deployment environment patterns. We describe topologies using the Integrated Solutions Console and a command-line interface.

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.

Martin Keen is a Consulting IT Specialist at the ITSO, Raleigh Center. He writes extensively about WebSphere products and service-oriented architecture (SOA). He also teaches IBM classes worldwide about WebSphere, SOA, and enterprise service bus (ESB). Before joining the ITSO, Martin worked in the EMEA WebSphere Lab Services team in Hursley, U.K. Martin holds a bachelor’s degree in Computer Studies from Southampton Institute of Higher Education.

Vijay Bhadriraju is a Senior IT Specialist working for the IBM Application Integration Middleware organization in the Research Triangle Park Lab, USA. He

© Copyright IBM Corp. 2010. All rights reserved.

has 14 years of IT industry experience with six years spent in developing Rational® and WebSphere Studio IDEs and six years spent in Smalltalk consulting services. He currently works on enabling IBM Business Partners on IBM WebSphere Telecommunication software. He holds a Bachelor of Engineering Degree in Computer Science from the University of Pune, India.

Johannes Borch is a Client Technical Professional in the Software Group of IBM Luxembourg. He joined IBM in 2001 as a Java™ EE Developer where he led several Web development projects in the government sector on WebSphere Application Server for z/OS®. In his current role he is part of a European WebSphere Technical Expert Group with focus on WebSphere Business Monitor where he covers tasks ranging from product demonstration to implementation of high-availability environments. Johannes has a bachelor's degree in Commercial Information Technology.

Balint Bors is an IT Specialist working for IBM Software Group, Application and Integration Middleware Software division in Munich, Germany. Before he joined IBM in 2007, he was involved in research and development on .NET and Java EE platforms. In his current role at IBM he works on the resolution of WebSphere BPM and ESB issues of customers worldwide. He holds the IBM Certified Administrator for SOA Solutions certification. He has a master's degree in Applied Computer Science from the University of Freiburg, Germany.

Peter Daly is a WebSphere Consultant in the Software Group of IBM UK. He has been with IBM for over 10 years and previously worked extensively with UNIX® and other middleware systems and software. He now specializes in WebSphere Process Server and dataPower technologies and is a contributing author to other IBM Redbooks publications. Peter holds bachelor's degrees in Physics and Computer Science.

Andrew Ferrier is a Consultant within IBM Software Services for WebSphere (ISSW). He specializes in WebSphere ESB and WebSphere Process Server. He regularly consults and works with customers on both of these products, and is known both inside and outside IBM for his knowledge of WebSphere ESB in particular. He co-founded the WebSphere ESB Customer SWAT Team before joining ISSW. He has experience in documentation and technical writing, regularly contributing intellectual capital to the WebSphere community as a core part of his role. Andres contributed to the IBM Redbooks publication *Connecting Enterprise Applications to WebSphere Enterprise Service Bus*, SG24-7406, and has published on a technical blog that he co-founded called *SOA Tips 'n' Tricks*, and has recently published his first DeveloperWorks article. He also regularly speaks at internal and external customer conferences on topics such as WebSphere ESB and Process Server best practices.

Stephen Gibney works for IBM Software Group Services for WebSphere. He started his career working with solid state detectors at a UK research center in

1986, and he moved to a French research center in 1992. He made the transition to IT as a UNIX System Administrator in 1993. He returned to the UK in 1998 to work in Systems Management with the Tivoli® product set. He has been working with the WebSphere product set since 2004. Stephen works as part of a team that designs, develops, and implements WebSphere solutions. Stephen holds a degree in Mathematics and Computing.

Kiet Nguyen is an IT Solution Architect and Developer. He is currently working with IBM Software Group's WebSphere BPM Lab SWAT Team in Raleigh, North Carolina. He has extensive application integration and development experience in cross-platform/technology solutions. He supports and advocates worldwide internal and external customers in the adoption and usage of the IBM SOA BPM product suite to optimize their business processes.

Bobby Pham is a Senior Software Engineer in the Application and Integration Middleware Division of the IBM Software Group. He has over 10 years of experience implementing middleware topologies. His areas of expertise include WebSphere Integration Developer, WebSphere Application Server, and WebSphere Process Server. Before joining IBM, Bobby implemented large-scale distributed systems as a Senior Systems Programmer at North Carolina State University. He holds several WebSphere product certifications. Bobby has Bachelor of Science degrees in Electrical Engineering and Computer Science from North Carolina State University in Raleigh, NC.

Klaus Ulrich is an IT Specialist with the IBM Software Group in Boeblingen, Germany. He has 17 years of IT experience in the telecommunication and automotive industry and joined IBM in July 2006. He is a Sun Certified Java Programmer and an IBM Certified Business Process Analyst, and holds several WebSphere product certifications. Klaus has a degree in Computer Science from the University of Erlangen/Nuremberg, Germany.

Figure 1 From left to right: Vijay, Balint, Peter, Martin, Andrew, Stephen, Kiet, Klaus, Johannes, Bobby

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

- Karri Carlson-Neumann
- Charlie Redlin
- Torsten Wilms
- Ekkehard Voesch
- Matt Roberts
- Himanshu Srivastava
- Pankuj S Chachra
- Srikanth Bhattiprolu
- Kai (Mike) Zhang
- David Enyeart
- Salil Ahuja
- Ralf Schmauder
- Andreas Arning
- Richard Johnson
- Arthur Kevin McGrath
- Jonas Grundler
- Dominik Sebastian Meyer
- Andrew L. Tharp

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

Archived

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 e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/pages/IBM-Redbooks/178023492563?ref=ts

- 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

Archived

Archived

Part 1

Overview

Archived

© Copyright IBM Corp. 2010. All rights reserved.

Archived

1

Business Process Management products overview

This chapter provides a short introduction to the IBM Business Process Management (BPM) product family with emphasis on the related products discussed in this book.

Archived

© Copyright IBM Corp. 2010. All rights reserved.

The IBM BPM Suite (Figure 1-1) brings together IBM software and service offerings that allow you to realize successful BPM projects. It includes two foundational offerings, the IBM WebSphere Dynamic Process Edition and the IBM FileNet® Business Process Manager. These two offerings can be optionally augmented by extended value offerings. The service offerings provided by IBM help organizations realize BPM projects based on their business strategy.

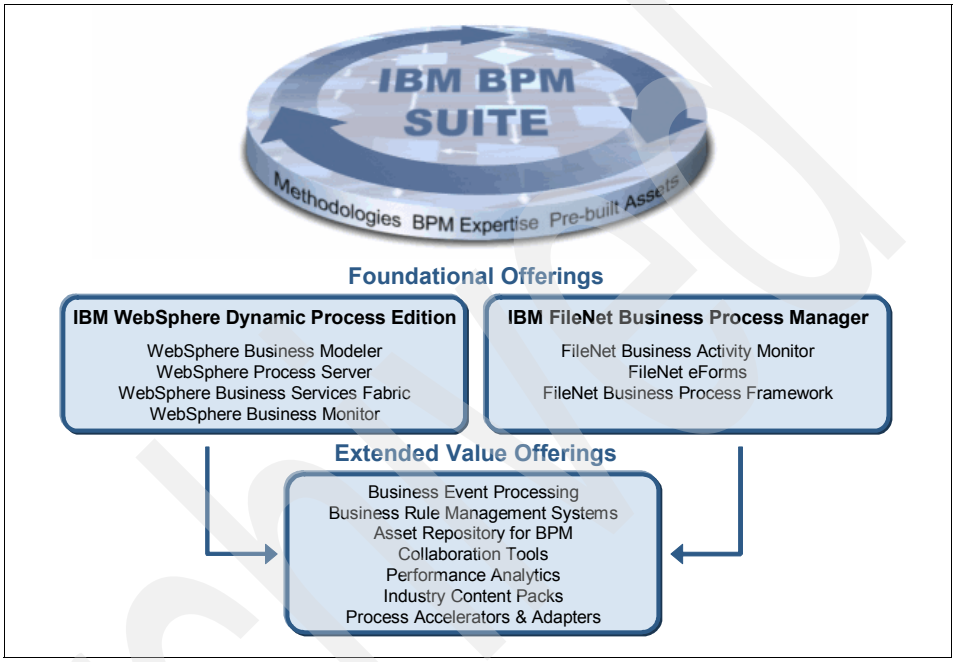


Figure 1-1 IBM BPM Suite

WebSphere Dynamic Process Edition provides end-to-end dynamic BPM life-cycle capabilities through a comprehensive set of products that enable you to model, simulate, execute, rapidly change, monitor, and continuously optimize core business processes.

IBM FileNet Business Process Manager is a complete offering for content-centric BPM. It enables Agile Enterprise Content Management (ECM) by combining Enterprise Content Management with Business Process Management. It unifies content and process capabilities, which allows the creation, management, and optimization of unstructured contents, business processes, and user interfaces.

The optional extended value offerings expand the IBM BPM Suite by leveraging additional areas of BPM such as business rules management, business event processing, performance analytics, and collaboration.

Note: Learn more about the IBM BPM Suite at:

http://www-01.ibm.com/software/info/bpm/offerings.html

Archived

1.1 WebSphere Dynamic Process Edition

WebSphere Dynamic Process Edition is one of the foundational offerings of the IBM BPM Suite. It provides end-to-end BPM capabilities for businesses to adapt and respond dynamically to changes in today's rapidly changing business environments.

WebSphere Dynamic Process Edition V7 includes the following products:

- WebSphere Dynamic Process Edition Tool and Testing Pack
  - IBM WebSphere Business Modeler Advanced 7.0
  - IBM WebSphere Business Services Fabric Tool Pack 7.0 (bundles WebSphere Integration Developer 7.0 and WebSphere Business Monitor Development Toolkit 7.0)
- WebSphere Dynamic Process Edition Server
  - IBM WebSphere Business Services Fabric Foundation Pack 7.0 (bundles WebSphere Process Server 7.0)
  - IBM WebSphere Business Monitor 7.0

The comprehensive set of role-based BPM products included in WebSphere Dynamic Process Edition allows continuous optimization and improvement of business processes throughout the complete BPM life cycle (Figure 1-2).

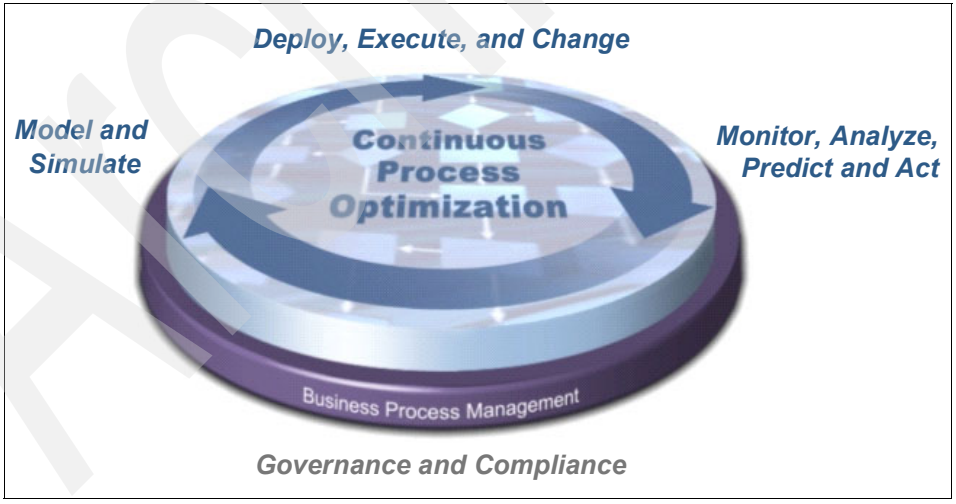


Figure 1-2 IBM BPM life cycle

Archived

1.1.1 Discover, model, and simulate

The implementation of a Business Process Management solution can start with modeling followed by simulating, analyzing, and defining business measures for business processes using WebSphere Business Modeler. While the basic edition of the product is a feature-rich, high-level business process modeling tool, the advanced edition additionally allows you to run simulations of the business processes, define business measures (key performance indicators and metrics), or develop applications from the process models.

1.1.2 Deploy, execute, and change

WebSphere Integration Developer allows developers to build service-oriented architecture (SOA)-based BPM and integration solutions. The model provided by WebSphere Business Modeler can be imported into the Eclipse-based WebSphere Integration Developer. This integration of the tools makes it possible to make incremental changes in the business model and let the implementation be adapted for rapid changes. The resulting solution is deployed to WebSphere Process Server or WebSphere Enterprise Service Bus runtime environments. See 1.4, “WebSphere Process Server” on page 9, and 1.6, “WebSphere Enterprise Service Bus” on page 10.

1.1.3 Monitor, analyze, predict, and act

WebSphere Business Monitor provides real-time end-to-end monitoring capabilities and enables users to monitor, analyze, and proactively manage their business solutions through configurable dashboards. WebSphere Business Monitor makes it possible to rapidly respond to business problems and helps to identify process improvement opportunities. See 1.3, “WebSphere Business Services Fabric” on page 8.

1.1.4 Governance and compliance

The cycle of continuous process improvement throughout the BPM life cycle is underpinned by robust governance and compliance capabilities to ensure the consistent operation of business processes as well as compliance with internal policies and external regulations. WebSphere Service Registry and Repository integrates across BPM solutions, providing a foundation of service governance and service information management. (WebSphere Dynamic Process Edition does not include WebSphere Service Registry and Repository.)

Note: Learn more about WebSphere Dynamic Process Edition at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp?topic=/com.ibm.wdpe.install.intro.doc/wdpe_intro/concept/c_introdn_wdpe.html

1.2 WebSphere Business Monitor

WebSphere Business Monitor is a comprehensive business activity monitoring tool that provides a real-time end-to-end view of business process performance. It allows for business users to predict problems before they occur. Process events and data can be collected from a wide variety of sources such as business applications running in the WebSphere Process Server runtime environment. This information is then presented in personalized dashboards for business users.

WebSphere Business Monitor V7 has added the ability to link strategic organization goals to operational metrics, has capabilities to handle in-flight processes, and improves end-to-end process monitoring by capturing events from additional IBM middleware and applications. In addition, the installation is simplified due to the new wizard-driven topology configuration capabilities.

Note: Learn more about WebSphere Business Monitor at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp?topic=/com.ibm.btools.help.monitor.intro.doc/intro/intro.html

1.3 WebSphere Business Services Fabric

WebSphere Business Services Fabric is an end-to-end platform to rapidly assemble, deliver, and govern industry-focused composite business services in SOA. A business service represents a business function whose behavior can be adapted at run time based on the current business situation and policy. These business services are assembled into modules whose life cycles are separated from the core processes, making manageability of complex BPM projects easier. WebSphere Business Services Fabric enables easy and fast governed change of business processes. It includes two software packages:

- IBM Business Services Foundation Pack
- IBM Business Services Tool Pack

The foundation pack contains a runtime business service policy enforcement and building engine that enables dynamic assembly of intelligent business services. It is based on WebSphere Process Server.

The tool pack is integrated into the WebSphere Integration Developer and includes a visual assembly environment for creating and managing intelligent business service models and policies. In the WebSphere Business Services Fabric V7 release, IBM Installation Manager now installs both the tool pack and the foundation pack.

The WebSphere Industry Content Packs V7, packaged as part of WebSphere Business Services Fabric in prior releases, is now available as a separate product.

Note: Learn more about WebSphere Business Services Fabric at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp?topic=/com.ibm.ws.fabric.intro.doc/intro/concept/c_product_overview.html

1.4 WebSphere Process Server

WebSphere Process Server is a standards-based business process integration server underpinned by the robust Java EE infrastructure of WebSphere Application Server, on which it is based. It allows deployment and execution of business processes that orchestrate services within an SOA environment. It also includes the complete functionality of WebSphere Enterprise Service Bus, enabling the implementation of SOA-based integration solutions.

WebSphere Process Server serves as the execution engine of business modules designed in WebSphere Business Modeler or WebSphere Integration Developer.

WebSphere Process Server V7 new features include enhanced support for open standards such as Java EE 5 and EJB 3.0, simplified system installation, easier cluster configuration, richer capabilities for process administrators, and single-step migration. WebSphere Process Server now allows you to change process instances in-flight to react dynamically for changing business conditions.

Note: Learn more about WebSphere Process Server at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp?topic=/com.ibm.websphere.wps.doc/doc/covw_intro_server.html

1.5 WebSphere Business Compass

WebSphere Business Compass provides a single, comprehensive authoring and reviewing environment for WebSphere Business Modeler. It allows for publishing of process models developed in WebSphere Business Modeler so that reviewers can view and comment on the models. IBM WebSphere Business Compass was known before V7 as IBM WebSphere Business Modeler Publishing Server. WebSphere Business Compass has two environments:

- WebSphere Business Modeler Review space, which allows for multiple people to view and contribute to the development of business processes through a Web-based environment
- Business Design space, which is a collaborative, Web-based environment built on the Business Space BPM framework for designing and building business strategy planning documents

Note: Learn more about WebSphere Business Compass at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp?topic=/com.ibm.btools.help.modeler.collab.publish.intro.doc/intro/introduction.html

1.6 WebSphere Enterprise Service Bus

WebSphere Enterprise Service Bus is a flexible connectivity infrastructure for integrating applications and services enabling the development of an SOA. It mediates message flow between service requesters and providers via Service Component Architecture (SCA) modules, which contain mediation flows. Mediation primitives implement the mediation logic in each flow. Such flows are designed visually using WebSphere Integration Developer.

Archived

The functionality of WebSphere Enterprise Service Bus can be found within WebSphere Process Server, and it is packaged also as a stand-alone product. Figure 1-3 illustrates how WebSphere Enterprise Service Bus fits into the WebSphere product stack and how each product buids on and extends the capabilities of other WebSphere products.

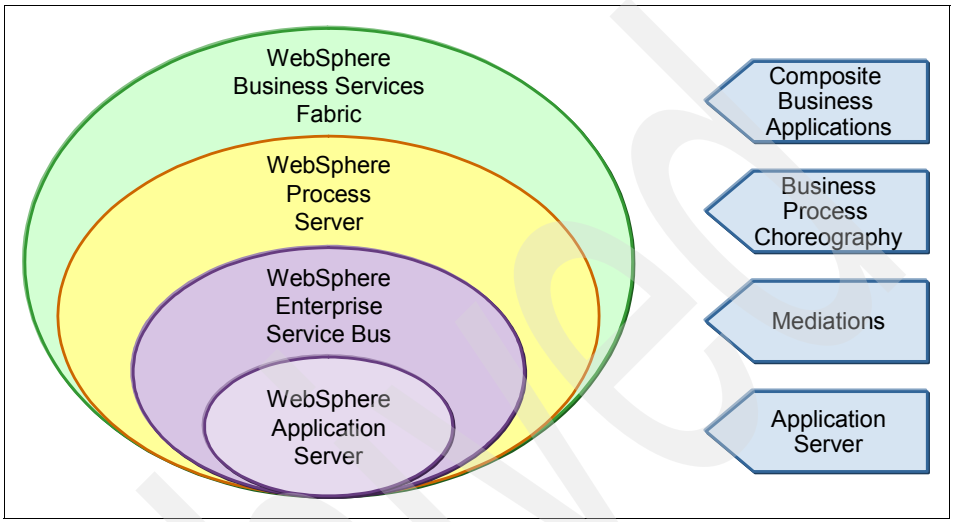


Figure 1-3 WebSphere product stack

New features in V7 include simplified system and cluster installation, enhanced policy-driven connectivity, a new Enterprise JavaBeans (EJB) export binding, additional message validation capability, a trace mediation primitive, Web service endpoint lookup from Universal Description, Discovery, and Integration (UDDI), and control of the execution order of mediation flow branches.

Note: Learn more about WebSphere Enterprise Service Bus at:

`http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp?topic=/com.ibm.websphere.wesb.doc/doc/welcome_wps_ovw.html`

Archived

2

Business Process Management production topologies

This chapter provides an introduction to the components of the Business Process Management (BPM) suite of products and to topology patterns. This chapter presents the various deployment environment and advanced topology scenarios implemented throughout this book:

- Single Cluster topology (described with WebSphere Process Server)
- Remote Messaging topology
- Remote Messaging and Remote Support (described with WebSphere Process Server and WebSphere Business Services Fabric)
- Double Remote Messaging and Remote Support (described with WebSphere Process Server)
- Remote Messaging, Support, and Web (described with WebSphere Business Monitor)
- Five-cluster topology (described with Dynamic Process Edition and WebSphere Business Monitor augmentation of WebSphere Process Server)

The chapter also includes recommendations and guidelines for how to select a production topology that best meets your requirements.

© Copyright IBM Corp. 2010. All rights reserved.

2.1 Introduction

A Business Process Management topology is the physical layout of the deployment environment required to meet your business needs for capacity, availability, and scalability. Key aspects of the Business Process Management topology design are:

- The number of physical machines (in distributed environments)
- The number of servers on those machines
- The number of clusters needed to provide your production environment with the processing capabilities required by your business.

In addition, a production deployment topology includes other supporting resources such as a user registry (for security), one or more HTTP servers (for Web content or IP spraying), necessary firewalls, load balancers, and so forth.

Carefully plan the production deployment topology, considering the:

- Number of physical machines and hardware resources that you require
- Number of clusters and cluster members required to support your business
- Number of databases required
- Authentication roles and security considerations
- Method that you will use to implement the deployment environment
- The cost of software and hardware

To make the topology design and implementation process easier, the Business Process Management suite of products includes a set of deployment environment patterns that represent common production topologies.

The deployment patterns offer a repeatable method that may be automated for creating the deployment environment that best suits your needs. The patterns also allow you to capture the configuration for later export and use on other systems. However, manual deployment (through the administrative console) or a scripted install is still possible in V7. Whether you perform a manual configuration of your topology or use the deployment topology patterns, there are a number of components to consider when creating the topology.

Note: The topologies in this chapter do not specifically consider software-licensing costs. Investigate these costs as part of any production topology decision.

Archived

2.2 Network deployment concepts

This section defines the following concepts:

- Components of a WebSphere network deployment environment
- Clusters
- Load balancing
- Failover

2.2.1 Components of a WebSphere network deployment environment

WebSphere Application Server Network Deployment solutions are described and built using the components discussed in this section.

Deployment environment

A deployment environment is a logical concept. It is a collection of configured clusters, servers, and middleware that collaborate to provide an environment to host software modules. For example, a deployment environment might include a host for message destinations, a processor or sorter of business events, and administrative programs.

Cells

A WebSphere cell is a logical concept. It is a grouping of nodes that are centrally managed and have access to shared resources. Nodes within a cell typically run one or more application servers. Each application server hosts one or more applications that are similar in terms of business requirements or non-functional requirements.

Nodes

A WebSphere node is a logical concept. A node consists of a node agent, by which the node is controlled, and the application servers that are created as members of that node.

Node agents

The WebSphere node agent is a server that enables the deployment manager for the cell to remotely manage the node, its application servers, and their applications. It is implemented via a Java Virtual Machine (JVM).

Deployment manager

A WebSphere deployment manager is an application server whose only task is the management and configuration of the cell in which it exists. The deployment

Archived

manager runs a Web-based configuration front-end known as the Integrated Solutions Console (ISC), through which you can perform management tasks.

Figure 2-1 shows a cell topology that consists of two nodes, each running a node agent. Each node contains one or more application servers. You can administer the cell using either the Integrated Solutions Console or command-line scripting (wsadmin). Both communicate with the deployment manager (*not* directly to the application servers). The deployment manager communicates with node agents, which communicate with application servers on the nodes. This allows central administration of the cell through the deployment manager, which maintains the master repository of configuration information and other artifacts for the cell.

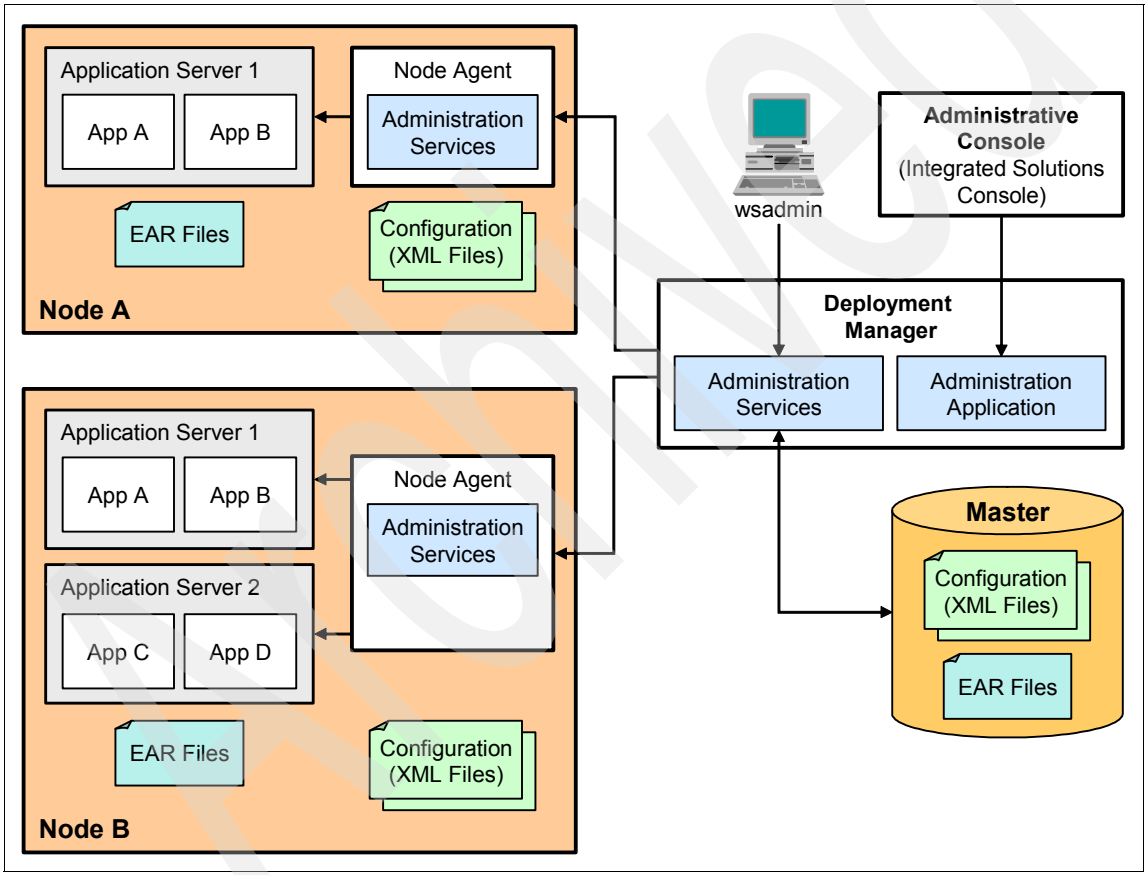


Figure 2-1 A cell topology

Clusters

A WebSphere cluster is a logical collection of application servers configured to perform the same task as a team. The member application servers can be distributed across one or more nodes in any configuration.

Application servers

A WebSphere Application Server hosts zero or more J2EE applications. An application server instance (or *profile*) can be configured as follows:

- Stand-alone application

  A stand-alone application server does not belong to a cell and runs its own Integrated Solutions Console.

- Singleton application

  A singleton application server resides on a node belonging to a cell and is managed by a deployment manager residing on a separate node. The application server is not part of a cluster.

- Member of a cluster

  An application server that is a cluster member resides on a node belonging to a cell and is managed by a deployment manager residing on a separate node. The application server is part of a cluster.

Profiles

A profile defines the runtime environment with a set of files. The profile includes all the files that the server processes in the runtime environment and that you can change. The `<app_server_root>/profiles` directory is the default directory for creating profiles. The configuration for every defined application server process is within the `profiles` directory unless you specify a new directory when you create a profile.

2.2.2 Clusters

A cluster is a grouping of one or more fundamentally identical units that perform one task. WebSphere Application Server Network Deployment application servers are clustered to allow for higher throughput, to achieve higher levels of resilience, or both. The WebSphere clusters are different from most other types of clusters. A WebSphere cluster also includes a life-cycle model.

Archived

Vertical clustering

In a vertical cluster, multiple application servers are placed onto the same node in order to better use the available resources (Figure 2-2). Such clusters can increase throughput and provide resiliency if one member of the cluster fails due to an application fault. Vertical clusters do not provide resiliency if the hardware hosting the member's' node fails.

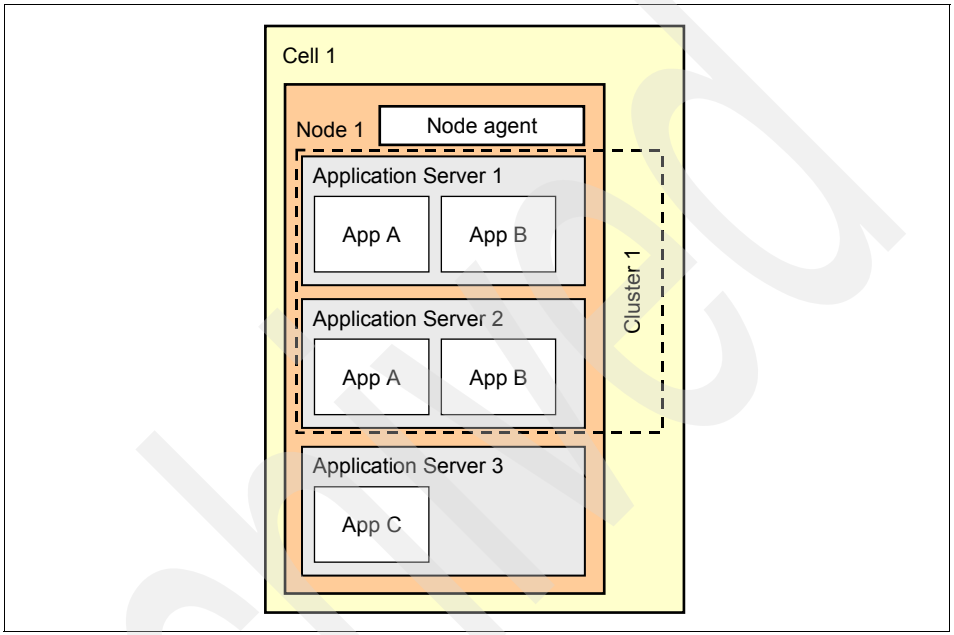


Figure 2-2 A vertically clustered WebSphere environment

Archive

Horizontal clustering

In a horizontal cluster, multiple application servers are distributed across nodes in order to use more physical resources (Figure 2-3). Such clusters can increase throughput and provide resiliency if a cluster member fails due to an application fault or if the hardware for that member's node fails.

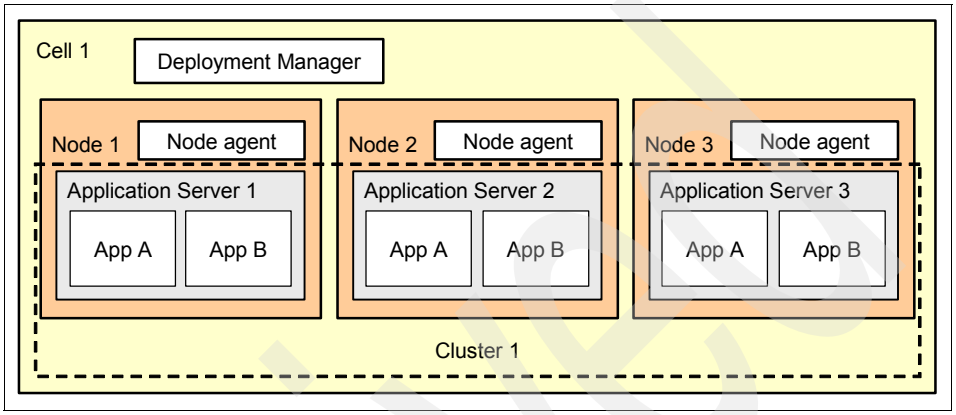


Figure 2-3 A horizontally clustered WebSphere environment

Archived

2.2.3 Load balancing

A load-balanced environment presents a collection of application servers as a single processing environment. Requests are distributed across application servers in response to the individual load and availability of each server in order to prevent an individual server from being overloaded (Figure 2-4).

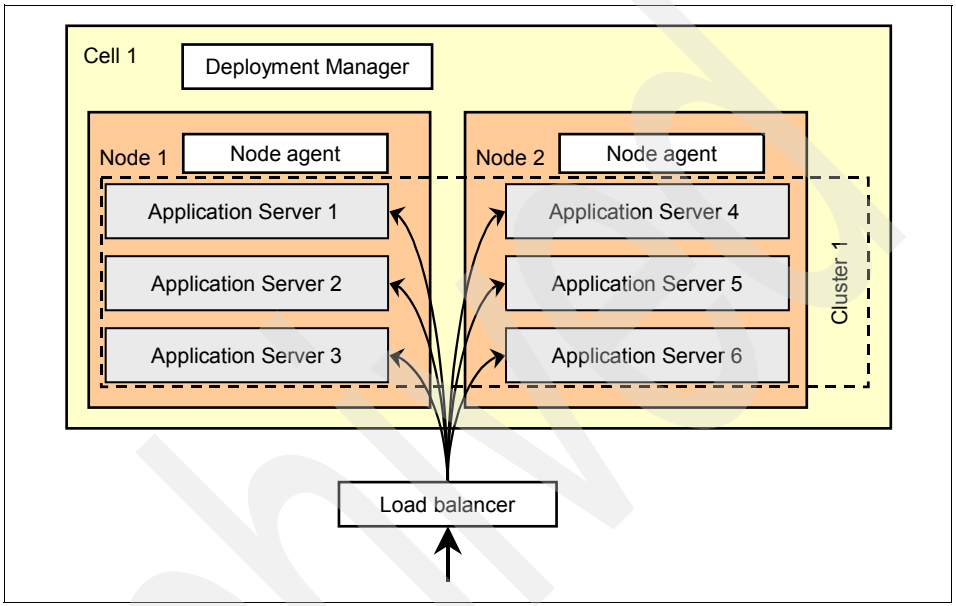


Figure 2-4 A load-balanced WebSphere environment

2.2.4 Failover

Clustering of application servers enables an environment to achieve higher throughput by distributing the load among a collection of application servers. By sharing data, a cluster of servers can all work on a single transaction should different requests arrive at different servers. However, transactions are usually passed to the same server to reduce the need for inter-server communication.

2.3 Product components

When you generate a deployment environment, a number of different components are created and used. These components are discussed in this section.

2.3.1 Databases and schemas

The Business Process Management suite of products detailed in this book are configured to use multiple databases and schemas. The tables within are used to hold, store, and track information. WebSphere Process Server, WebSphere Business Monitor, and WebSphere Business Services Fabric make use of the following databases and schemas:

- Common database (WPRCSDB)

  This database is used as a repository for various components in WebSphere Process Server, and there is one common database per cell. It must be created prior to starting the WebSphere Process Server, the WebSphere Business Services Fabric, or the WebSphere Dynamic Process Edition deployment manager. WebSphere Business Services Fabric augments this database with Fabric-specific tables.

- Business Process Choreographer database (BPEDB)

  This database is used by the Business Flow Manager and the Human Task Manager. It must be created prior to starting BPC components. There is one BPEDB database for each Business Process Choreographer deployment target.

- Business Process Reporting schema (BPCREP, BPCEventCollector, or OBSVRDB)

  This may be a dedicated database but is typically a schema within the BPEDB database if event collection is not heavily used. It is used by the BPC Explorer reporting function to store event information from the Common Event Infrastructure (CEI) bus in an event collector table.

- Business Space database (BSPACE)

  This database is used by the Business Space applications to store configuration data for the various user profiles, space definitions, and page definitions.

- Monitor database (MONITOR)

  This database is used by WebSphere Business Monitor to store metadata and configuration information, AlphaBlox® data, and schemas for each deployed monitor model. It must be created prior to starting the WebSphere Business Monitor deployment manager.

- Messaging engine database (MEDB)

  This database is used to host tables for messaging engines. There are multiple messaging engines in any BPM environment. There are messaging engines for use by the Service Component Architecture (SCA) system and application buses, the CEI bus, the business monitor bus, and the Business Process Choreographer bus. Each messaging engine needs a separate set of

Archived

tables. This is normally implemented as separate schemas within the same database. The messaging engines require one database for each messaging engine.

The WebSphere Business Services Fabric bus is no longer required in V7 and hence the associated schema is no longer required.

- Event database (EVENT)

  This database contains information regarding the event service, such as common based events. In production the persistence of events into this database is typically disabled because of performance implications.

- WebSphere Business Compass database (WBCDB)

  This database contains configuration information. In addition, it contains the published process artifacts from WebSphere Business Modeler and the business documents from WebSphere Business Compass.

2.3.2 Service integration buses

A service integration bus (SIB) supports applications using message-based and service-oriented architectures (SOAs). A bus is a group of interconnected servers and clusters that have been added as members of the bus. Applications connect to a bus at one of the messaging engines associated with its bus members. Business Process Management products make use of the following service integration buses:

- SCA system bus

  This bus is used to host queue destinations for SCA modules. The SCA run time uses these queue destinations to support SCA asynchronous interactions between components and modules.

- SCA application bus

  Besides the SCA system bus used for SCA modules, you can also create other service integration buses for supporting the service integration logic provided by the modules. The SCA application bus is used for interaction of SCA components with other asynchronous components. For example, this bus supports the asynchronous communication between WebSphere Business Integration Adapters, other SCA components, and Java Message Service (JMS) resources for modules deployed with JMS bindings.

- Business Process Choreographer bus

  This bus is used for transmitting messages internally in the Business Flow Manager.

Archived

- Common Event Infrastructure bus

  This bus is used to transmit common base events asynchronously to a CEI server.

- Monitor bus

  This bus is used in the processing of events delivered to various JMS queues associated with monitor models. A specific name for this bus is expected (MONITOR.cell.Bus). If queue bypass is configured WebSphere Business Monitor connects directly to the Monitor database. However, the bus is still required for action services queues.

2.3.3 Business Process Choreographer

Business Process Choreographer container (BPC) is an enterprise workflow engine that supports both business processes and human tasks. The core of the BPC configuration consists of the following components:

- Business Flow Manager (BFM)

  This component executes and manages business processes written in Business Process Execution Language (BPEL).

- Human Task Manager (HTM)

  This component manages human tasks. The HTM can be used to dynamically change the behavior of human tasks, such as transferring ownership of a task or assigning an expiration date at run time. It also allows for programmatic control of the human task flow.

2.3.4 Business Process Choreographer Explorer (BPC Explorer)

Business Process Choreographer Explorer is a Web application that can be installed as part of the configuration of the business process container. Before you can start using Business Process Choreographer Explorer from a Web browser, you must have installed the business process container, human task container, and the Business Process Choreographer Explorer application, and the application must be running. The event collector application must be installed and running in order to use the reporting function.

Depending on your user role, you can use Business Process Choreographer Explorer to manage business processes and human tasks, or to work with your assigned tasks. While business processes and tasks are running, WebSphere Process Server can emit events that contain information about state changes of process instances and their related activities. Using reporting, you can retrieve statistical information based on these events and create reports on processes and activities.

Archived

2.3.5 Business Rules Manager (BRM)

The business rules manager is a Web-based tool that assists the business analyst in browsing and modifying business rule values. The tool is an option of WebSphere Process Server that you can select to install at profile creation time or after installing the server.

Business rules are designed and developed in WebSphere Integration Developer using if/then rule sets and decision tables to implement their operations. Business rules can also be created in WebSphere Business Modeler. However, WebSphere Business Modeler only supports the creation of business rule tasks, which become rule sets when exported out of WebSphere Business Modeler. The rule sets and decision tables are set into templates. The templates control which aspects of a business rule you can modify and by exactly how much. They define the structure of if/then rules, condition cases, and actions for decision tables.

2.3.6 Common Event Infrastructure

A business event is a significant occurrence in a business process that warrants monitoring over time. A business process component can be configured to generate an event. These events are then used to evaluate whether an aspect of the business reaches predefined goals. WebSphere Process Server uses the CEI for basic event management services and WebSphere Business Monitor uses the CEI as a source of events that are analyzed historically. Events can be published to the CEI server for possible distribution to JMS queues and topics. If enabled for non-production environments, events might also be persisted to a database.

WebSphere Business Monitor uses the CEI as a source of events.

2.3.7 Business Space powered by WebSphere

Business Space powered by WebSphere is a browser-based graphical user interface that allows the customization of content from the Business Process Management (BPM) suite of products. It provides a common presentation layer for application users to manipulate the Web interfaces of BPM applications.

2.3.8 Monitor action services

Monitor action services is a component of WebSphere Business Monitor that invokes monitor action services when it receives defined situation events emitted

Archive

by WebSphere Business Monitor and other applications. The actions are based on user-defined action templates.

2.3.9 Monitor scheduled services

This component provides services at scheduled intervals. Many services are scheduled, such as:

- Data movement services
- Key performance indicator (KPI) history
- Key performance indicator predictions
- Archiving
- Pruning alert evaluation

2.3.10 Data movement service

The data movement service (DMS) optimizes server processing and reporting in higher-volume production environments. Operational tables are optimized for inserts and updates, and the reporting tables are optimized for dashboard queries. When enabled, the data movement service runs automatically as a scheduled service. Once enabled, the data movement service cannot be disabled. The data movement service is optional in this release.

2.3.11 Mobile device dashboard

This component allows WebSphere Business Monitor dashboards to be viewed on mobile devices. The intention is to bring WebSphere Business Monitor to a larger audience by allowing these devices to receive alerts and view KPIs previously only accessible via desktops.

2.3.12 AlphaBlox

The AlphaBlox component provides dimensional data analysis capabilities and is required where Business Space Manager is included in a WebSphere Business Monitor deployment.

2.3.13 Compass Server Application

The Compass Server Application is used by the Business Design space and widgets. The Business Design space provides a collaborative, Web-based environment for designing and building business strategy planning documents.

Archived

2.3.14 Publishing Server Application

The Publishing Server Application is used by the WebSphere Business Modeler Review space. The WebSphere Business Modeler Review space provides a way to publish business processes and related business information such as organization diagrams and dashboard and form pages to a secure Web site.

2.3.15 IBM HTTP Server and plug-in

The IBM HTTP Server (IHS) is a Web server based on Apache HTTP Server with IBM enhancements. The IHS plug-in provides (among other functionality) a load-balancing capability that can be closely integrated with the WebSphere topology in the environment.

The HTTP server's responsibility is to distribute load among cluster members. IHS is designed to handle HTTP load that comes from thousands of users, and is a key component for high-volume scenarios.

The topologies described in this book use this load-balancing functionality to route requests from Business Space widgets to the Representational State Transfer (REST) gateway services, from the BPCExplorer UI to the BPEContainer and HTMContainer REST services. Essentially, this is to decouple the Web UI interfaces from the backend data requests by acting as a HTTP sprayer.

2.3.16 WebSphere proxy server

The WebSphere proxy server is used to classify, prioritize, and route HTTP and SIP requests to servers in the enterprise and cache server content. It resides within a node. It is used in the topologies described in this book in a similar way to the IBM HTTP Server and plug-in.

2.4 Deployment environment patterns

A Business Process Management topology can be created using the IBM-supplied deployment environment patterns. The deployment environment patterns included in the administrative console represent the common topologies used. Each pattern centers around the number of clusters and cluster members and how these clusters are grouped and allocated across nodes.

Any topology contains four basic sets of functions that together form a complete production environment. Each of these functions can be separated into

individual, dedicated clusters, or they can be combined, depending on your needs. The set of functions are:

- Application deployment target

  An application deployment target is the set of servers to which you install your applications (human tasks, business processes, monitor models, mediations, and so forth).

- Supporting infrastructure

  The supporting infrastructure includes the CEI and other infrastructure services such as the *action services and monitor scheduled service* used to support your environment.

- Messaging infrastructure

  The messaging infrastructure is the set of servers used to provide asynchronous messaging support for your applications and for the internal messaging needs of the Business Process Management components (for example, the internal navigation queues used by long-running business processes).

- Web infrastructure

  The Web infrastructure provides GUI-based functionality and the GUI supporting infrastructure. For example, AlphaBlox, Business Space, REST API services, BPC Tools, and the Business Rules Manager components.

Note: IBM HTTP Server is a key component for handling HTTP traffic in a production topology. Although not explicity shown in the diagrams in this section, each production topology described here uses IBM HTTP Server to handle HTTP load.

Each of the provided deployment environment patterns creates a different number of clusters to support the required functions. The deployment environment patterns included in the BPM product suite in this book are:

- Single cluster

  In this pattern, the messaging infrastructure, the application deployment target, and the support functions are contained in a single cluster. This is the default pattern for a z/OS installation.

  Supported feature combinations for this pattern include:

  – WebSphere Enterprise Service Bus
  – WebSphere Process Server
  – WebSphere Business Services Fabric
  – WebSphere Business Monitor

Archived

- Remote Messaging

  This pattern separates the messaging infrastructure from the application deployment target and support infrastructure. In this pattern, two clusters are created:

  – One for applications and support functions
  – One for the messaging infrastructure

  Supported feature combinations for this pattern include:

  – WebSphere Enterprise Service Bus
  – WebSphere Process Server
  – WebSphere Business Services Fabric

  Note: In a production topology with a CEI component, any significant throughput of CEI events would warrant a dedicated support cluster for hosting the CEI architecture.

- Remote Messaging and Remote Support

  This pattern separates the messaging infrastructure, the support infrastructure, and the application deployment target into individual clusters. In this pattern, the following three clusters are created:

  – Applications (named AppTarget)
  – Support infrastructure (named support)
  – Messaging infrastructure (named messaging)

  This is the preferred pattern for WebSphere Process Server and WebSphere Business Services Fabric for Multiplatforms.

  Supported feature combinations for this pattern include:

  – WebSphere Enterprise Service Bus
  – WebSphere Process Server
  – WebSphere Business Services Fabric

- Remote Messaging, Support, and Web

  This pattern consists of four distinct clusters. The application deployment target cluster is for service modules, mediation modules, and monitor models. The messaging cluster is for the messaging infrastructure. The support cluster is for CEI or WebSphere Adapters, or both. The Web cluster is for all of the Web-based client applications (such as Business Space, BPC tools, REST API Services, AlphaBlox, or any home-grown Web applications).

  This pattern is only presented in the Integrated Solutions Console for environments that have WebSphere Business Monitor installed.

Archived

This is the preferred deployment environment pattern for WebSphere Business Monitor, primarily because Alphablox Web reporting is resource intensive, so it is best to use a dedicated cluster for the Web applications.

Supported feature combinations for this pattern include:

- WebSphere Business Monitor
- WebSphere Business Monitor + WebSphere Business Services Fabric
- WebSphere Business Monitor + WebSphere Process Server

Note: In a production topology, co-hosting the business monitor reporting on the same cluster as the business processing applications could have an negative effect on the business applications themselves (that is, looking at purchases can impede the actual purchase). An option to consider is to host the monitor models on a dedicated cluster, which implies an augmentation to this topology to the five cluster topology.

- Custom deployment environments

  If none of the deployment environment patterns meets your requirements, you may create a custom deployment environment.

  Note: WebSphere Business Monitor does not support custom deployment environments. Instead there is a WebSphere Business Monitor configuration wizard.

Regardless of the type of pattern that you use, generating a deployment environment on the administrative console creates an XML-based representation of your topology that can be exported, modified, imported, and re-used to create the topology on any number of systems. For example, you may want to use the same XML topology descriptor to generate both your test and your pre-production environments.

There are several methods that you can use to generate a topology:

- Install the software on the host systems using the Installation Manager. Use the Profile Management Tool (32-bit operating systems only) or the **manageprofiles** command to create the administrative architecture (deployment manager and nodes). Create the deployment environment using the Integrated Solutions Console.
- Install the software on the host systems using the Installation Manager. Use the Profile Management Tool (32-bit operating systems only) or the **manageprofiles** command to create the administrative architecture. Create the deployment environment using the **wsadmin** utility.

- Install the software on the host systems using the Installation Manager. Use the Profile Management Tool (32-bit operating systems only) or the **`manageprofiles`** command to create the administrative architecture (deployment manager and nodes), then use wsadmin or the administration console for a manual configuration of the topology (create databases, create clusters, deploy BPC, and so on). This is not discussed in this book.

Regardless of which method you use to create the deployment environment, you can still manage certain aspects of the deployment environment through the Integrated Solutions Console (for example, add more nodes to the deployment environment).

Making changes: The Deployment Environment wizard is for initial generation only. You can make changes to the underlying configuration artifacts using the Integrated Solutions Console or wsadmin. Any changes made to a specific resource *after* generation (for example, a data source) are not reflected in the deployment environment XML file that can be exported.

2.4.1 Single Cluster topology pattern

The Single Cluster topology pattern provides one cluster for all the functional components. The user applications, messaging infrastructure, CEI, and support applications are all configured in the same cluster. Typically, this topology is used for testing, proofs of concept, demonstration environments, and environments with simple failover requirements.

Archive

Figure 2-5 shows a Single Cluster topology sample configuration for WebSphere Process Server.

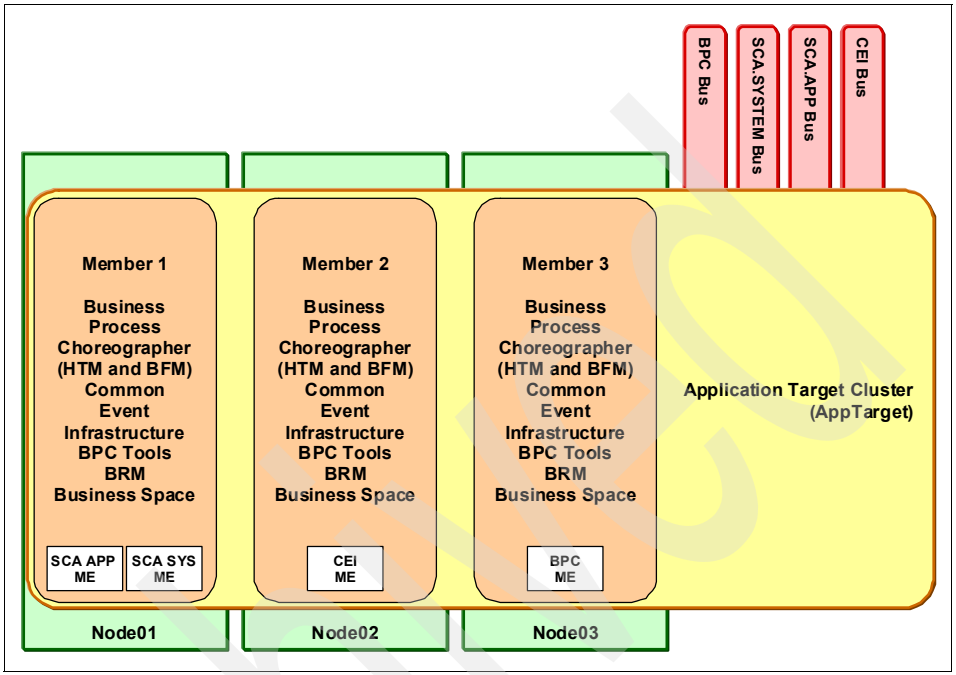


Figure 2-5 Example process of server Single Cluster topology

Note the following aspects of this example:

- All of the components are configured in a single cluster that has a default name of AppTarget.
- The AppTarget cluster is a member of all four of the required WebSphere Process Server buses:
  - SCA.SYSTEM bus
  - SCA.APPLICATION bus
  - CEI bus
  - BPC bus
- The BPC is configured in the cluster so that each cluster member has a business process container and a human task container.
- All of the supporting infrastructure applications are configured in the cluster:
  - BPC tools
  - Business Rules Manager
  - CEI
  - Business Space

- Each cluster member is an application deployment target.
- In Figure 2-5 on page 31, the messaging engines are distributed across the cluster members. Cluster member 1 has active SCA.SYSTEM and SCA.APPLICATION messaging engines. Cluster member 2 has an active CEI messaging engine. Cluster member 3 has an active BPC messaging engine. It is not the default configuration. By default, each cluster member is capable of running all four of the messaging engines, and the server that starts first will automatically run all four of the engines.

2.4.2 Remote Messaging topology pattern

The Remote Messaging topology pattern provides one cluster for the messaging infrastructure (named Messaging) and a second cluster for all of the remaining components (named AppTarget). The Remote Messaging topology is sometimes used by small and medium-sized businesses, or for isolated environments in large enterprises. Figure 2-6 shows a Remote Messaging sample topology.

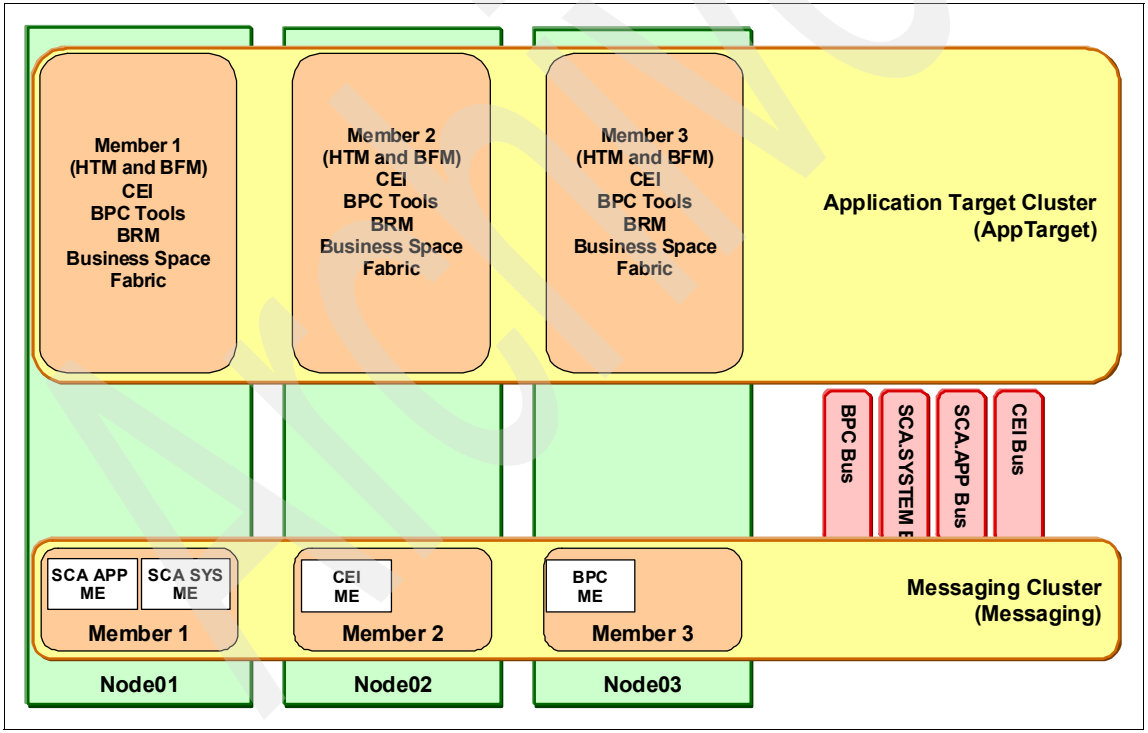


Figure 2-6 Example Fabric Remote Messaging topology

Note the following aspects of this example:

- All of the applications and supporting infrastructure components are configured in a single cluster, which has a default name of AppTarget.
- The BPC is configured in the AppTarget cluster, so each cluster member has a business process container and a human task container.
- Fabric is co-hosted with the BPC component.
- The messaging cluster is a member of all the required WebSphere Process Server buses (used by WebSphere Business Services Fabric):
  - SCA.SYSTEM bus
  - SCA.APPLICATION bus
  - CEI bus
  - BPC bus
- All of the supporting infrastructure applications are configured in the AppTarget cluster:
  - BPC Tools
  - Business Rules Manager
  - CEI
  - Business Space
- In Figure 2-6 on page 32, the messaging engines are distributed across the members of the messaging cluster. Cluster member 1 has active SCA.SYSTEM and SCA.APPLICATION messaging engines. Cluster member 2 has an active CEI messaging engine. Cluster member 3 has an active BPC messaging engine. It is not the default configuration. By default, each cluster member is capable of running all of the messaging engines, and the server that starts first will automatically run all of the engines.

Note that the behavior of the messaging engines in a Remote Messaging topology is different from the default behavior when the messaging engines are co-located with the applications. Because the messaging engines are in a remote cluster, there is no preference for the message producers and consumers to use a local messaging engine. Each member of the AppTarget cluster will connect to the appropriate bus and use the Remote Messaging engine for that bus.

2.4.3 Remote Messaging and Remote Support topology pattern

The Remote Messaging and Remote Support topology pattern is the preferred topology for WebSphere Process Server and WebSphere Business Services Fabric production environments. This is preferred because it provides for scaling by allowing cluster members to be added to any components that are the performance bottleneck.

Archived

This topology provides three separate clusters:

- Remote Messaging cluster (named messaging)
- Support infrastructure cluster (named support)
- Application deployment target cluster (named AppTarget)

Figure 2-7 shows a Remote Messaging and Remote Support sample topology.

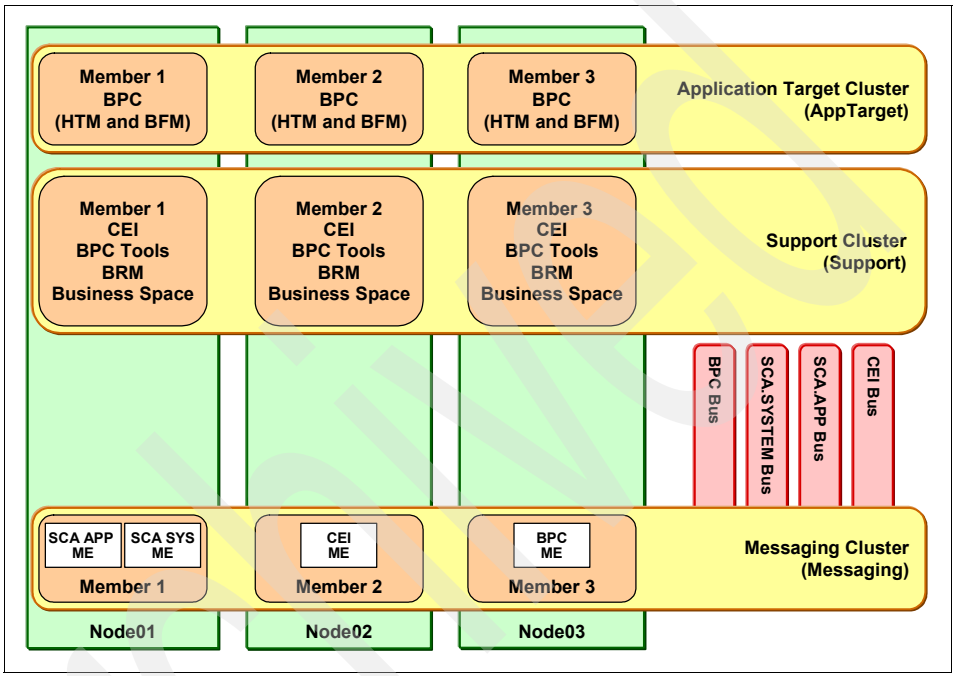


Figure 2-7 Example process server Remote Messaging and Remote Support topology

Note the following aspects of this example:

- All of the applications are deployed to the AppTarget cluster.
- The BPC is configured in the AppTarget cluster, so each cluster member has a business process container and a human task container.
- The messaging cluster is a member of all four of the required WebSphere Process Server buses:
  - SCA.SYSTEM
  - SCA.APPLICATION
  - CEI
  - BPC

Archive

- All of the supporting infrastructure applications are configured in the support cluster:
  - BPC Tools
  - Business Rules Manager
  - CEI
  - Business Space
- The messaging engines are split across the members of the messaging cluster, as shown in Figure 2-7 on page 34.
  - Cluster member 1 has active SCA.SYSTEM and SCA.APPLICATION messaging engines.
  - Cluster member 2 has an active CEI messaging engine.
  - Cluster member 3 has an active BPC messaging engine. It is not the default configuration.

  By default, each cluster member is capable of running all four of the messaging engines, and the server that starts first automatically runs all four of the engines.

2.4.4 Remote Messaging, Support and Web topology pattern

The Remote Messaging, Support and Web topology pattern is the preferred topology for WebSphere Business Monitor production environments. This topology provides four separate clusters:

- Remote Messaging cluster (named messaging)
- Support infrastructure cluster (named support)
- Application deployment target cluster (named AppTarget)
- Web component cluster (named WebApp)

Archived

Figure 2-7 on page 34 shows a Remote Messaging, Support and Web sample WebSphere Business Monitor topology.

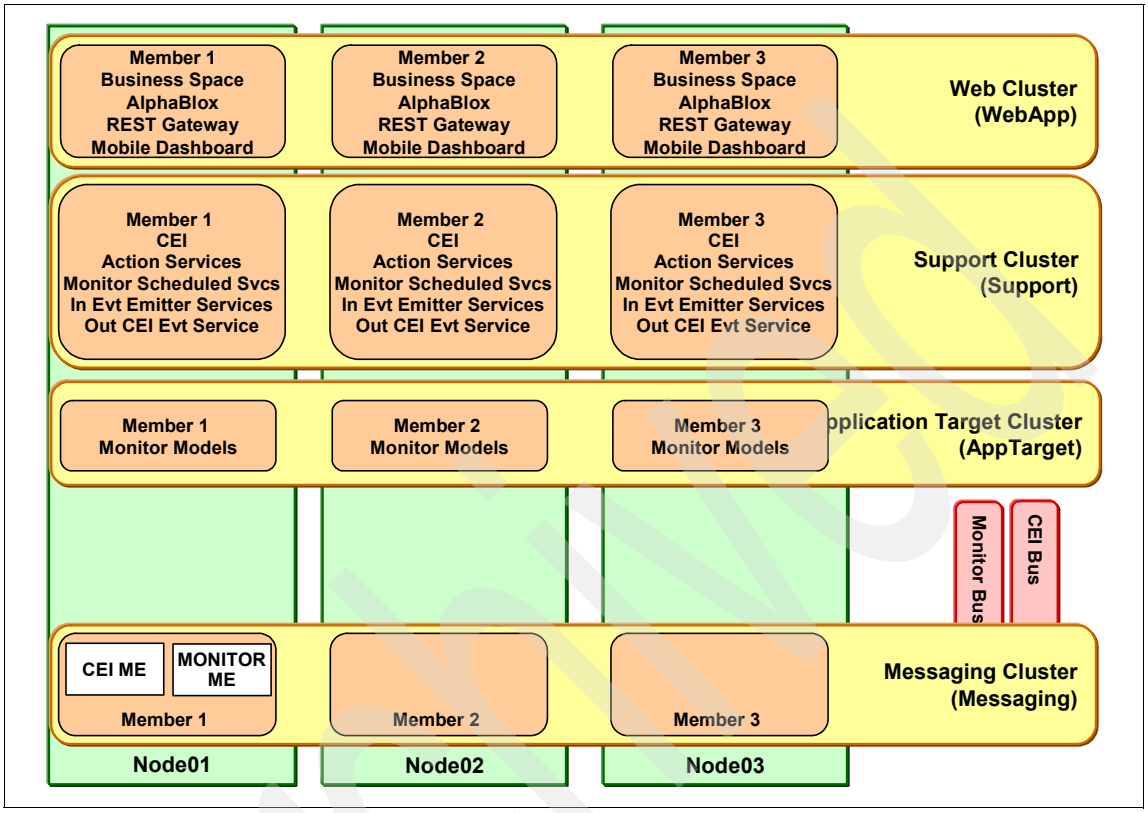


Figure 2-8 Example Monitor Remote Messaging, Support and Web topology

Note the following aspects of this example:

- The monitor models are deployed to the AppTarget cluster.
- The Messaging cluster is a member of the required WebSphere Business Monitor buses:
  - CEI
  - MONITOR
- The Web components for supporting the visualization infrastructure are deployed to the Web cluster:
  - Business Space
  - AlphaBlox
  - REST
  - Monitor widgets

- Monitor Alphablox widgets
- Monitor mobile dashboards

- The remaining supporting infrastructure applications are configured in the support cluster:
  - Action services
  - Monitor scheduled services
  - CEI
  - Inbound event emitter services
  - Outbound CEI event service
- The messaging engines are set to the default. The first member of the messaging cluster to start hosts both messaging engines. Given that queue bypass is the default in WebSphere Business Monitor V7 (and a leading practice), the monitor bus is not likely to generate heavy load. The messaging engines might need to be distributed if the source of monitor events do not have direct access to reach the monitor database (and hence queue bypass cannot be used). The choice between queue bypass and interconnecting cells is described in more detail in 9.1, "Topology summary" on page 298.
  - Cluster member 1 has active CEI and MONITOR messaging engines.
  - Cluster members 2 and 3 are alternative locations for the messaging engines.

  By default, each cluster member is capable of running the messaging engines, and the server that starts first automatically runs all the engines.

Note that the behavior of the messaging engines in a Remote Messaging, Support and Web topology is identical to the behavior discussed in the Remote Messaging and Remote Support topology description.

2.4.5 Custom topology patterns

If none of the default deployment environment patterns are suitable for your needs, it is possible to create a custom deployment environment.

Note: WebSphere Business Monitor does not support custom deployment environments. Instead, there is a WebSphere Business Monitor configuration wizard.

Also, you can use the administrative console or scripting to manually deploy the environment in any way that you choose. If you use the administrative console instead of the custom topology pattern, you will not have a re-usable XML representation of the topology.

Creating a custom topology is slightly different from the process for using the default topology patterns. There are several scenarios that are appropriate for a custom topology, for example:

- Removing the Business Rules Manager

  In most organizations, governance rules prevent business analysts from changing the parameters of business rules at run time. Thus, you may not expose any of your business rules at run time using the rule template functionality in WebSphere Integration Developer. If you cannot change rule parameters and you do not want to provide users with other functionality available in the Business Rules Manager (deleting rules, changing the order of rule execution, and so forth), you may want to create a custom deployment environment without the Business Rules Manager.

- Removing CEI support

  If you have a separate monitoring infrastructure in place, or if you are not currently taking advantage of the CEI, you may want to create a deployment environment without CEI support. Note that if you choose to remove CEI support, you will also lose the ability to use the BPC Explorer reporting function and the Common Base Event browser.

This list of possibilities is not meant to be exhaustive. There are many other possible reasons for creating custom deployment environments, including extending the Remote Messaging and Remote Support topology by adding additional clusters (to form a Double Remote Messaging and Remote Support topology).

If you choose to implement a custom topology pattern, note that it is generally unwise for you to use a custom deployment environment to move components into non-default locations. For example, you should not use a custom deployment environment to alter the Remote Messaging and Remote Support topology by placing the BPC Observer in the AppTarget cluster. The default topology patterns were designed to maximize performance. Altering their structure can have unexpected performance drawbacks.

2.5 Five-cluster topology pattern

WebSphere Dynamic Process Edition includes WebSphere Process Server, WebSphere Business Services Fabric and WebSphere Business Monitor configured in distinct combinations. The five-cluster topology pattern is an optional topology for WebSphere Dynamic Process Edition production environments. It is also an option for environments where WebSphere Process Server and WebSphere Business Monitor coexist in the same cell and monitor models must be isolated from business processes.

In a production topology, co-hosting the business monitor reporting on the same cluster as the business processing applications could impact the business applications themselves (that is, looking at purchases can impede the actual purchase). This need for isolation would suggest hosting the monitor models on a dedicated cluster.

This topology provides five separate clusters:

- BPM Web cluster
- BPM Support cluster
- BPM Application cluster
- Monitor Application cluster
- BPM Messaging cluster

Note: Create a four-cluster deployment environment and add an extra cluster manually.

Archived

Figure 2-9 shows a five-cluster sample topology. This sample incorporates WebSphere Dynamic Process Edition (WebSphere Process Server, WebSphere Business Services Fabric, and WebSphere Business Monitor) into one cell.

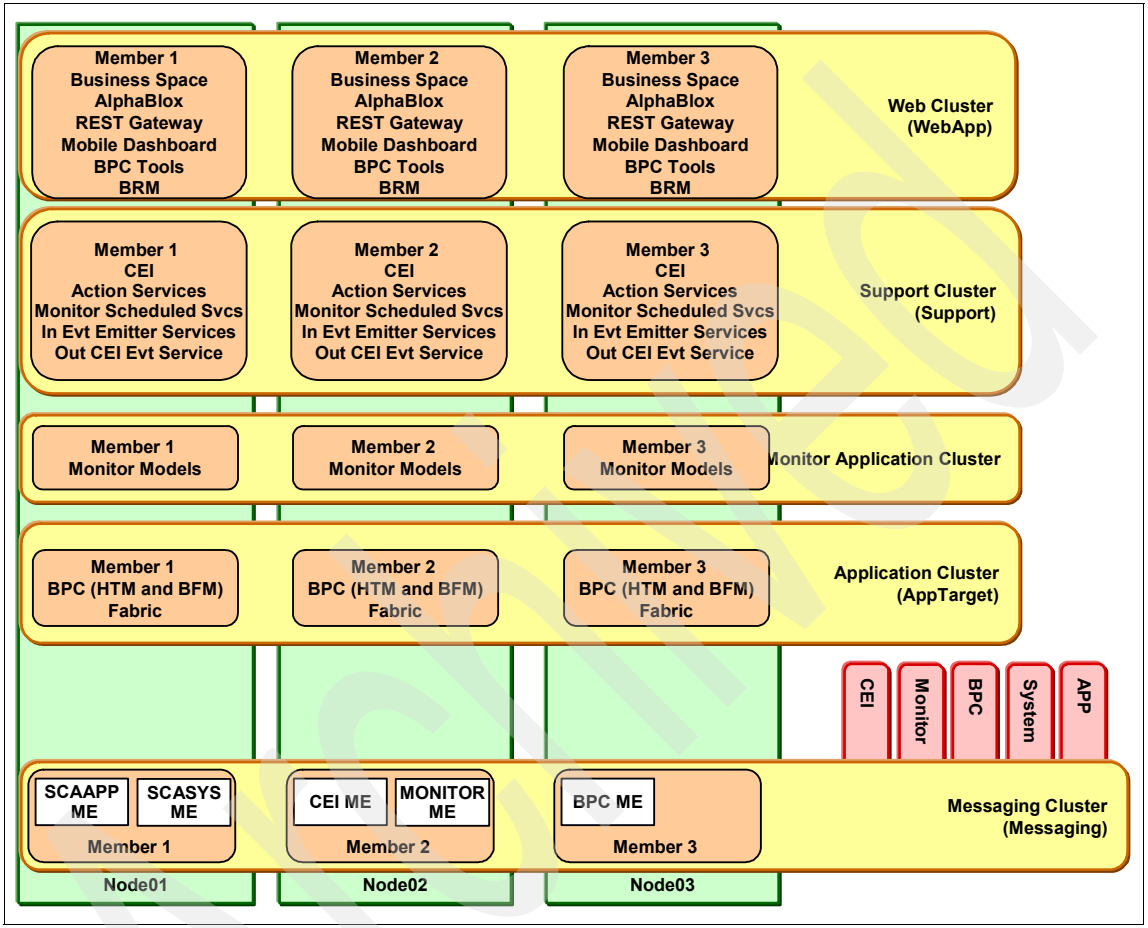


Figure 2-9 Five Cluster Dynamic Process Edition topology

Note the following aspects of this example:

- The application cluster contains:
  - BPC (HTM and BFM)
  - BPEL applications, Fabric application
- The monitor application cluster contains the monitor model (moderator/application logic)

- The support cluster contains:
  - CEI
  - Action services
  - Monitor scheduled services
  - Inbound event emitter service
  - Outbound CEI Event service
- The Web cluster contains:
  - Business Space
  - REST gateway
  - BPC tools
  - Business Rules Manager
  - Monitor widgets
  - Monitor Alphablox widgets
  - Monitor mobile dashboards
  - Fabric widgets
  - WPS widgets
  - Alphablox
- The messaging cluster will host the messaging engines for the required buses:
  - SCA.SYSTEM bus
  - SCA.APPLICATION bus
  - CEI bus
  - BPC bus
  - Monitor bus

2.6 Topology choice

Selecting an appropriate topology for your production environment depends on several factors, including, but not limited to, the following factors:

- Available hardware resources and operating system
- Hardware and software costs
- Application invocation patterns
- Types of business processes that you plan to implement (interruptible versus non-interruptible)
- How heavily you intend to use the CEI
- Individual scalability requirements
- Administrative effort involved

Archived

In general, the Remote Messaging and Remote Support topology pattern is the most suitable production topology for WebSphere Process Server and WebSphere Business Services Fabric. Remote Messaging, Support and Web is the most suitable for WebSphere Business Monitor, and a five-cluster custom topology is the choice for WebSphere Dynamic Process Edition or topologies with both WebSphere Process Server and WebSphere Business Monitor. However, the choice ultimately depends on the individual business requirements.

As you plan for your production environment, carefully consider the advantages and disadvantages of each of the common topology patterns

2.6.1 Single cluster

A Single Cluster topology is targeted for environments with limited hardware. Since all of the components are installed in the same cluster, fewer physical machines are required. However, each server instance must run the supporting applications and your integration applications. Therefore, the memory requirements for the individual Java Virtual Machines (JVMs) are much greater (which becomes a limitation in a 32-bit JVM). In addition, one or more members of the cluster must also run the messaging engines required for asynchronous interactions.

Combining all aspects of the Business Process Management environment into a single cluster has other implications aside from the increased memory requirements. Because asynchronous interactions (involving JMS and MQ/JMS bindings), human tasks, state machines, and long-running business processes can make extensive use of the messaging infrastructure, a single-cluster environment is not ideal for applications with these components. This topology is also not ideal if you intend to make extensive use of the CEI. Generating events and CEI-related messaging traffic places an additional burden on the cluster members.

From an administrative and scalability perspective, the Single Cluster topology has advantages. A single cluster where each member runs all the WebSphere Process Server or all the WebSphere Business Monitor components is easier to administer. Rather than several server instances in multiple clusters, you have a single cluster with fewer members. If the needs of your environment grow, scaling the infrastructure is a simple matter of adding additional nodes and cluster members. Therefore, the process of adding capability is easy, but all components are scaled at the same rate. For example, each additional cluster member adds CEI processing whether you need it or not. If you have the messaging engines spread across server members using policies, there may be additional administrative effort in creating and maintaining the policies.

2.6.2 Remote Messaging and Remote Support

For the vast majority of customers (especially those with large computing infrastructures), the Remote Messaging and Remote Support topology is the preferred starting environment for WebSphere Process Server. The hardware requirements for distributed platforms are more intensive, but having three (or more) clusters with multiple members performing specific functions allows greater flexibility in adjusting and tuning memory and CPU usage for the JVMs.

Creating three clusters, each with its own functions and applications, creates an additional administrative burden. As you add clusters and cluster members, your performance-tuning plan and the troubleshooting burden can expand greatly. Spreading messaging engines across the members of the messaging cluster also adds to the administrative burden associated with creating and maintaining policies.

From a scalability standpoint, the Remote Messaging and Remote Support topology provides the most flexibility. Because each of the distinct functions within WebSphere Process Server is divided among the three clusters, you can pinpoint performance bottlenecks and adjust the cluster size fairly easily. If you need additional CEI processing, you can simply add a node and cluster member to the support cluster. Similarly, if you need more processing capability for your business processes or human tasks, you can add additional nodes and members to the application target cluster.

The scalability options for the Remote Messaging and Remote Support topology are as straightforward as the options for the Single Cluster topology. Because the messaging engines are subject to one of *n* policies (each messaging engine is active on only one server), adding additional members to the messaging cluster has little effect. Spreading the messaging engines across server members using policies can allow you to split the messaging burden across a the three servers (the SCA.SYSTEM and SCA.APPLICATION engines should be active on the same server, as their communications are more closely coupled). Thus, adding more than three cluster members to the messaging cluster has no effect on the processing capability of the messaging infrastructure.

The Remote Messaging and Remote Support topology provides an ideal environment for long-running business processes, state machines, human tasks, and asynchronous interactions (including JMS and MQ/JMS bindings). Since the application target cluster is only responsible for running your business integration applications, performance tuning and diagnostics are much simpler than in the previous topologies where the application target cluster had additional responsibilities.

The Remote Messaging and Remote Support topology is also ideal for environments that make extensive use of CEI for monitoring and auditing. Separating the support infrastructure into its own cluster provides you with a dedicated set of cluster members for CEI and for the supporting applications like BPC Explorer and Business Space.

2.6.3 Double Remote Messaging and Remote Support

The Double Remote Messaging and Remote Support topology builds on the Remote Messaging and Remote Support topology. This topology contains a second WebSphere Process Server application cluster, a second BPEDB, and a second messaging cluster that will join the existing bus infrastructure. This topology is used to extend the Remote Messaging and Remote Support topology for reasons of:

- Isolation

  There are a number of definitions of isolation when it comes to expanding topologies:

  - New applications might have unique maintenance and update requirements at a business level that are inconsistent with other applications. Newer applications may not be as well behaved, and their deployment may add significant risks to existing, well-established, and critical applications.
  - In addition to business, or logical, separation, different applications may be required to run on different sets of physical hardware. New clusters can be created on separate hardware, allowing for both hardware and application-level isolation.
  - Some applications may have different quality of service (QoS) requirements, including different failover and recovery capabilities. They may even have different functional requirements. For example, some applications may not require any process choreography. The cluster may not have a BPC or Human Task Manager (HTM) configured, which other applications may require.
  - Some clusters may need to serve distinct sets of clients or unrelated applications (for example, different departments in your company, or a generic application provider service. This is know as multi-tenancy).

Archive

Note: Isolation has drawbacks as well as benefits. Consider an extended topology with multiple application targets, each with its own BPEDB. With this topology, when querying the BPEDB (for example, to claim a human task), the result would only show tasks from within a single BPEDB. While this may not be an issue, based on a pure isolation application model, it is worth considering when deciding how to grow your topology.

- Growth

  With an increase in applications comes the possibility of an increase in messaging engine destinations. More applications deployed on a single application cluster increases the possibility of memory utilization issues on the application target (especially in a 32-bit environment), increases failover for messaging engines, and increases startup times for both messaging engines and application targets. Similarly, if new versions of applications are deployed, the number of destinations increases further.

  Deploying numerous applications, especially those that use large objects, causes these clusters to become constrained by memory or resources. Shared thread pools and activation specifications may not be optimally tuned for the many modules deployed on a single cluster.

In a topology where you have two application clusters and two messaging clusters, both messaging clusters are members of the SCA.SYSTEM, SCA.APPLICATION, and BPC buses. When you add additional application and messaging clusters, there are still only four service integration buses in your topology.

If you implement this topology, it is not necessary to add the second messaging cluster as a member of all four buses. Because the CEI destination is configured at the cell level, both application clusters can use the same CEI destination, CEI bus, and CEI messaging engine. If you are making extensive use of CEI when you implement this topology, you may also want to add additional nodes and cluster members to the support cluster to prevent bottlenecks.

Archive

2.6.4 Remote Messaging, Support and Web

The Remote Messaging, Support and Web topology is the recommended starting topology for WebSphere Business Monitor. WebSphere Business Monitor makes greater use of the Web UI components. This pattern uses a fourth cluster to house the following Web applications:

- BPC tools
- Business Rules Manager
- Business Space
- REST API Services[1]
- AlphaBlox

Aside from giving you the ability to precisely control the individual components deployed in your environment, the advantages of the Remote Messaging, Support and Web topology are similar to those of the Remote Messaging and Remote Support topology. The disadvantages are also similar.

2.6.5 Five cluster

The Remote Messaging, Support and Web topology is the recommended starting point for a full WebSphere Dynamic Process Edition deployment. The five-cluster topology is an option if greater isolation is required. The augmentation to a Remote Messaging, Support and Web topology is that the monitor application cluster is separated out to prevent contention with the business application target cluster.

WebSphere Dynamic Process Edition includes much of the BPM stack, and isolation of the different functions into dedicated clusters minimizes the impact of cross-cluster interference, and allows for the scalability of separate components independently.

2.6.6 Messaging engine location and partitioning

When the messaging engines and the applications are co-located in a *single cluster*, the default behavior is for message producers and consumers to always use a local active messaging engine (if one is available). For example, assume that you have two applications deployed to each cluster member needing to communicate asynchronously. Once each message producer places messages in the queues, the message consumer on the machine where the engine is local consumes all of the messages produced. Thus, the consuming application only processes messages on the server with the local messaging engine, and messages can be stranded.

[1] REST API services for human task widgets reside in the application cluster.

Archived

Note that the behavior of the messaging engines in any topology with a *Remote Messaging* cluster is different from the behavior when the messaging engines are co-located with the applications. Since the messaging engines are in a remote cluster, there is no preference for the message producers and consumers to use a local messaging engine. Each member of the AppTarget cluster will connect to the appropriate bus and use the Remote Messaging engine for that bus.

When a single bus member has more than one active messaging engine created in a cluster, its destinations are partitioned across all messaging engines in that cluster. Each messaging engine deals with a subset of the messages that the destination handles. Each server's active messaging engines contain a portion of the queues assigned to that engine. Thus, you can attain additional throughput if there are active messaging engines on each member of the messaging cluster. However, this configuration can create issues for your applications as follows:

- Lost message order: If you partition destinations when the applications and messaging engines are in separate clusters, you will no longer have the ability to maintain message order. Any time that you partition destinations you lose message order. This is true even if you attempt to enable event sequencing in WebSphere Process Server.
- Possible stranded messages: By default, you have no control over which active messaging engine your applications will use at run time. This can create situations where two applications on the same server attach to two different messaging engines. If one application produces messages for one engine and the message consumer is using a different engine, stranded messages can result.

Destination partitioning is strongly discouraged.

More information: For detailed information about workload sharing with queue destinations, refer to the WebSphere Application Server Network Deployment Information Center at the following Web site:

`http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.pmc.nd.multiplatform.doc/concepts/cjt0014_.html`

2.6.7 REST endpoint configuration

Most of the products in the WebSphere BPM product suites expose a public API through Representational State Transfer (REST) APIs. In a production topology where components are spread over several nodes and clusters for high-availability and scalability reasons, the REST APIs must also be configured for high availability and scalability. This section describes the different configuration options available to achieve this.

The REST APIs are extensively used by the Business Space widgets. The widgets use the different REST APIs to communicate with product components. For example, the WebSphere Business Monitor Instance widget uses the following WebSphere Business Monitor REST API to query Monitor Model properties:

```
https://<web server host>/rest/bpm/monitor/models/HelloWorldMM
```

The available REST services can be listed in the Integrated Solution Console by navigating to **Services** → **REST services**, as shown in Figure 2-10.

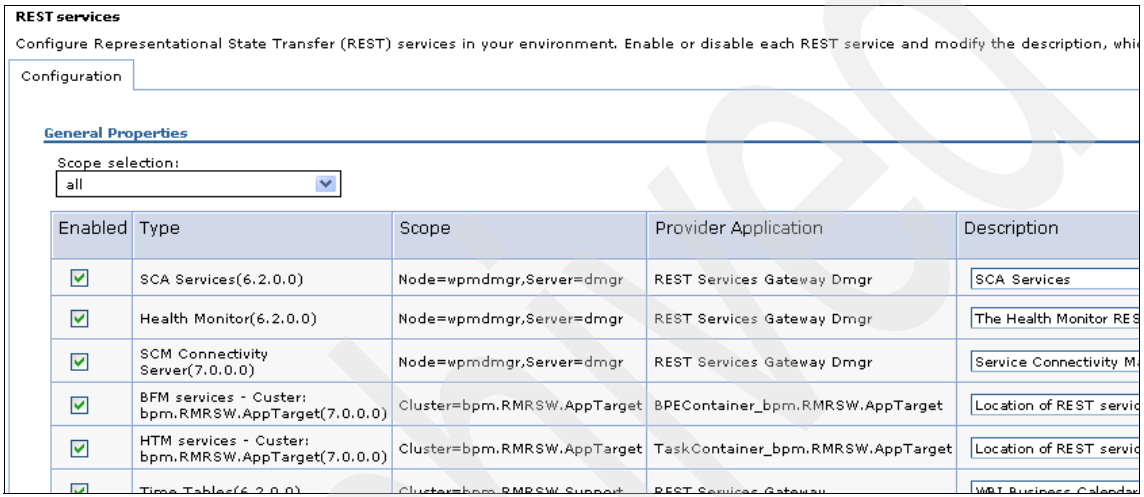


Figure 2-10 List of available REST services

All installed REST services listed here are served by service providers (except the Human Task Manager REST API and the Business Flow Manager API, which are served directly by the corresponding container). The service providers or REST service gateway can be deployed on a different cluster/server in the topology.

Archived

The available service provider and deployment location can be listed in the Integrated Solution Console by navigating to **Services** → **REST service providers**, as shown in Figure 2-11.

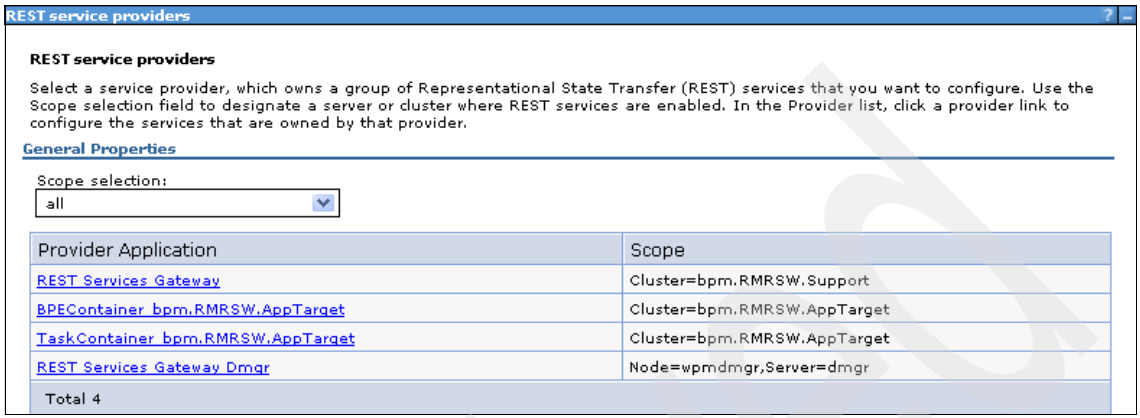


Figure 2-11 Available REST service provider

The distribution of REST service providers on a different cluster/server within the cell raises an issue related to JavaScript restrictions. For security reasons, JavaScript code running in a Web browser can only make HTTP requests to the originating server where the Web page came from. So it will not be able to call REST services deployed on the support cluster, for example, if the Business Space application is deployed on the WebApp cluster.

Archived

Two solutions to get around this restriction and to ensure REST service load balancing in a clustered environment are described below. The first one is the recommended one and uses a front-end reverse proxy HTTP server that routes the REST request to the correct cluster member. Figure 2-12 shows this configuration.

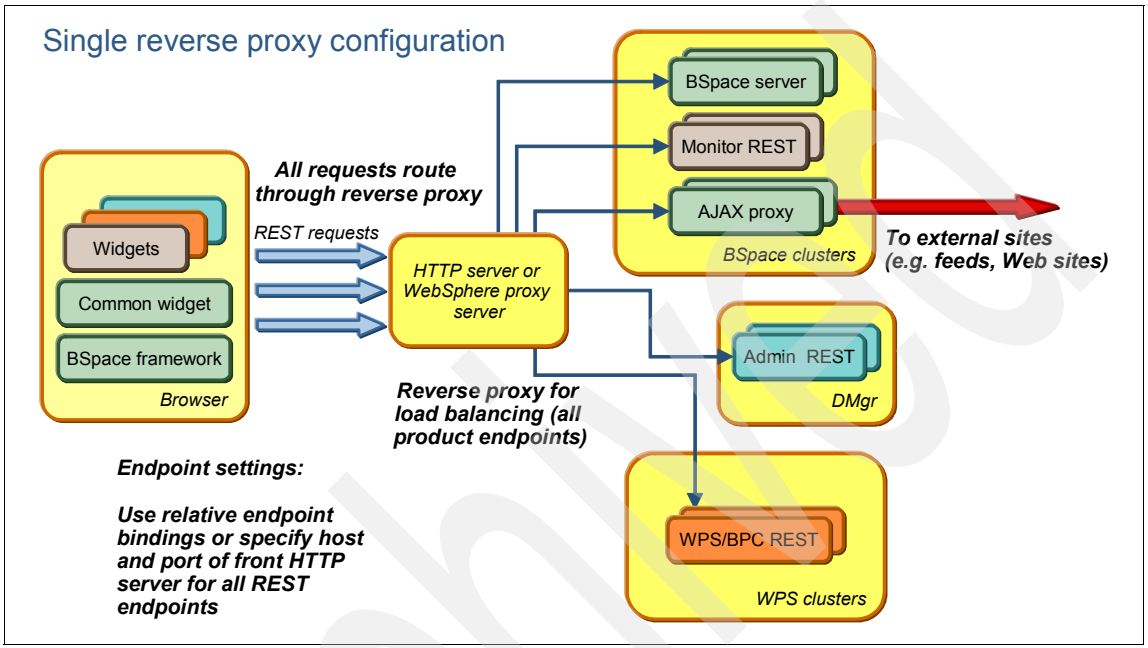


Figure 2-12 Single reverse proxy configuration

Archive

The second solution uses a AJAX gateway built into the Business Space that uses URL rewriting to bypass the JavaScript same origin restriction. Figure 2-13 shows this configuration.

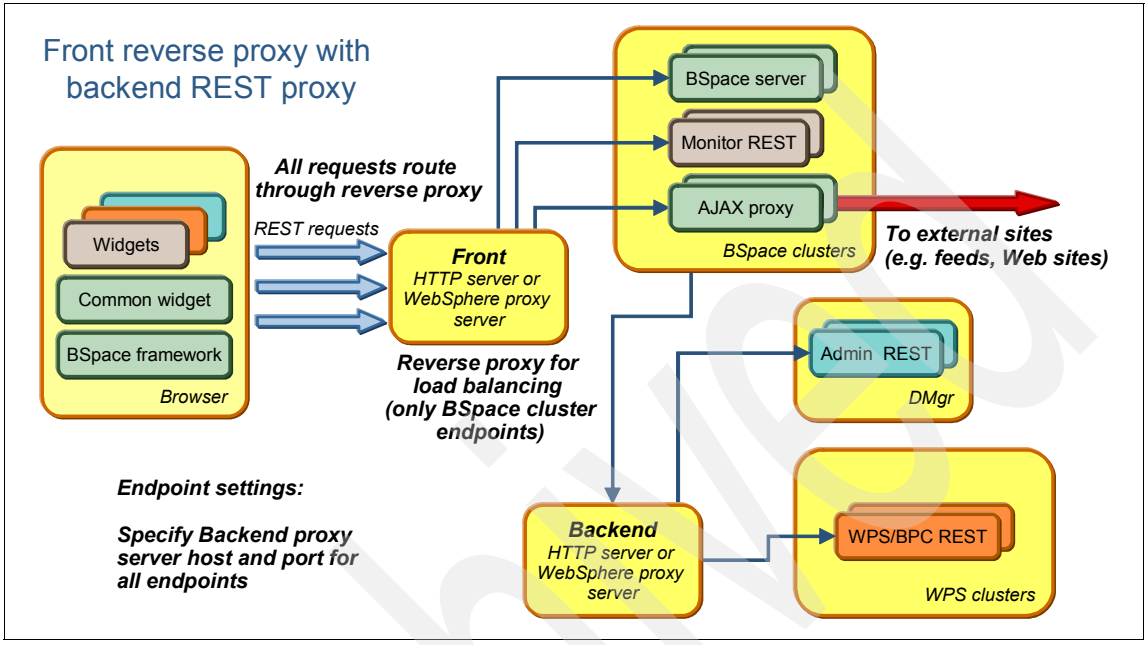


Figure 2-13 Front reverse proxy with backend REST proxy

The two solutions described above can be configured by setting one of the two service provider REST endpoints modes:

- Relative REST endpoint configuration, as shown in Figure 2-12 on page 50

  This can be set by specifying the HTTP server and port through which the user accesses the Business Space. In this configuration the Business Space run time detects that the REST service provider is co-located on the same server and widgets access the REST service provider directly. This solution is easier to set up, and it is easier to unload the Business Space run time, as REST requests are directly routed to REST providers without going through the Business Space AJAX proxy.

- Absolute REST endpoint configuration, as shown in Figure 2-13

  This can be set by specifying a different host and port than the host/port through which the user accesses Business Space. In this configuration the Business Space run time detects that the service provider is not co-located on the same server as the Business Space run time and activates URL rewriting through the AJAX proxy.

If, for example, the instance widget wants to access the monitor API described above in absolute REST endpoint mode, it emits the following request:

```
https://<httpserver>/mum/proxy/https/<proxyServer>%3A9080/rest/bpm/
monitor/models/HelloWorldMM
```

This solution has certain disadvantages because all REST requests must go through the AJAX proxy and because we require a second HTTP proxy to load balance REST requests issued by the AJAX gateway.

One exception is the Health Monitor REST service. This REST service is deployed on the deployment manger and does not need to be load balanced. This endpoint is always configured as an absolute REST endpoint in both solutions (pointing to the deployment manager host and port).

Archived

Part 2

Building production topologies

Archived

© Copyright IBM Corp. 2010. All rights reserved.

Archived

3

Preparing your topology

In this chapter we create a base infrastructure for other chapters to build on. Read this chapter in conjunction with the specific topology in which you are interested. This base infrastructure includes deployment manager and node creation with security using Lightweight Directory Access Protocol (LDAP). We include the following important features:

- Installation of the product silently and using the GUI
- Applying fix packs silently and using the GUI
- Using the database design tool (DDT) to generate the Structured Query Language (SQL)
- Creating the databases before deployment manager creation
- Deployment Manager creation
- Securing your infrastructure using LDAP
- Node creation
- Other software installs

By the end of this chapter you should have a database server with all databases available, a LDAP repository populated with users, a deployment manager with security, two nodes created (and federated), and other software required for your specific product and topology.

© Copyright IBM Corp. 2010. All rights reserved.

3.1 Prerequisite software installation

Before we begin product installation we need a database server (DB2®) and a LDAP repository. You may already have a DB2 and LDAP infrastructure that you can leverage.

Software versions used in this book

We used the following products to create the topologies in this publication. Although these topologies are implemented on Linux®, the procedures are the same for all distributed platforms.

- SUSE Linux Enterprise Server V10 SP1 64-bit
- IBM DB2 Enterprise Server Edition V9.5.5 64-bit
- IBM Tivoli Directory Server V6.2 64-bit
- WebSphere Process Server V7.0.0.1 64-bit
- WebSphere Business Monitor 7.0.0 64-bit
- WebSphere Business Services Fabric 7.0.0 64-bit
- WebSphere Dynamic Process Edition 7.0.0 64-bit
- WebSphere Business Compass 7.0.0 64-bit

Note: During our build of the topologies we required the following changes to the operating system:

- Increased the ulimit for open files (We used ulimit -n 10240.)
- Set the umask to 022
- Enabled Network Time Protocol (NTP) so that all servers were time synchronized

Installing IBM DB2 V9.5.5

We do not provide the details of installing DB2 because it is well documented in the Information Center, however, we do provide the response file that we used as part of the additional materials (the file is called `db2ese.rsp`).

Note: The installation instructions for DB2 are described in the Information Center at:

`http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.qb.server.doc/doc/c0007503.html`

Installing IBM Tivoli Directory Server V6.2

We do not provide the details of installing IBM Tivoli Directory Server because it is well documented in the Information Center, however, we do provide the LDIF

file that we used as part of the additional materials (the file is called `ldifmins.ldif`).

Note: The installation and configuration instructions of IBM Tivoli Directory Server are described in the Information Center at:

`http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.IBMDS.doc/install05.htm`

3.2 Software installation

The order of installation for the software stack in this book is:

1. Install DB2 and apply DB2 fix packs or leverage your existing infrastructure.
2. Create DB2 instances using the **`db2icrt`** command (a sample shell script is provided as part of the additional materials).
3. Install IBM Tivoli Directory Server (or use an existing LDAP) and import the LDIF file.
4. Install the chosen WebSphere Business Process Management software.
5. Apply any fix packs.

We assume that you have DB2 available with instance owners and that the LDAP is populated with users.

Installing the product is a similar process for all products. We run through a WebSphere Business Monitor installation, then highlight the differences or additional steps for other products. The installation is performed using the IBM Installation Manager both via the command line and with the GUI. In both cases we require a software repository.

We use the root user to perform the installation.

3.2.1 Create a software repository

Note: In our lab environment we use a 64-bit operating system and install the 64-bit version of the products. However, you can install either the 64-bit or 32-bit product on a 64-bit operating system.

Archived

A repository is a folder that contains the installation software and fix packs. We un-tar or un-zip the files shown in Table 3-1 and Table 3-2 on page 59 to the appropriate locations (Example 3-1 on page 59). In this book we identify the product files as `<base_dir>` and the fix pack files as `<fixpack_dir>`.

Table 3-1 Software installation repository locations

Product	Source file part number (filename)	Base location (<base_dir>)	Install location (<install_dir>)
WebSphere Process Server	CZ7GWML (WPS_v7_Linux_x86_64_Install.tar.gz)	/software/wps7000	/opt/ibm/WebSphere/ProcServer
WebSphere Business Monitor	CZ8W3ML (Monitor_V7.0_Linux_64.tar.gz)	/software/wbm700	/opt/ibm/WebSphere/MonServer
WebSphere Business Services Fabric	CZAR1ML (CZAR1ML.tar.gz)	/software/fab700	/opt/ibm/WebSphere/Fabric/FoundationPack
WebSphere Dynamic Process Edition	CZB4IML (CZB4IML_linux64.tar.gz) CZB6MML (CZB6MML_for_unixes.tar.gz)	/software/wdpe700	/opt/ibm/WebSphere/DynProcSvr7
WebSphere Business Compass	CZ8LNML (Compass_V7.0_Linux_32_1.tar.gz) CZA7ZML (Compass_V7.0_Linux_32_2.tar.gz)	/software/wbc700	/opt/ibm/WebSphere/BusComp
IBM HTTP Server	C1G36ML (C136GML.tar.gz)	/software/IHS	/opt/IBM/HTTPServer

Archive

The locations listed in Table 3-2 will be used for fix pack repositories.

Note: WebSphere Process Server has *two* repository locations because it is the only one that has a fix pack available at the time of writing.

Table 3-2 Fix packs for WebSphere Process Server

Product	Source file	Fix pack location (<fixpack_dir>)
WebSphere Process Server	wps.7001.repository.zip, sca.1.0.1.1.fp.zip, xml.1.0.0.1.fp.zip	/software/wps7001

In Example 3-1 we create a WebSphere Process Server software repository using the values in Table 3-1 on page 58 and Table 3-2.

Example 3-1 Example of creating a software repository

```
mkdir -p /software/wps7000
cd /software/wps7000
tar xzf /software/WPS_v7_Linux_x86_64_Install.tar.gz
mkdir -p /software/wps7001
cd /software/wps7001
unzip /software/wps.7001.repository.zip
unzip /software/sca.1.0.1.1.fp.zip
unzip /software/xml.1.0.0.1.fp.zip
```

Note: By default, and after creating the repositories above, the installation manager is configured to connect to the Internet. If the environment does not have an Internet connection, the installation manager hangs during startup while trying to access this site. This is discussed in the following Tech Note:

http://www-01.ibm.com/support/docview.wss?uid=swg21414213

We also include the changes here.

Archived

After creating the base repository for your product, you must edit the files `install.xml` and `post-install.xml` in the `<base_dir>/IM` folder. Remove or comment out the line referring to `public.dhe.ibm.com`, as shown in the snippet in Example 3-2 (we do not show the entire file).

Example 3-2 Comment out the internet repository

```
...
<server>
<repository location='.' temporary='true'/>
<repository location='../repository/' temporary='true'/>
<!-- repository
location='http://public.dhe.ibm.com/software/websphere/repositories/'
-->
</server>
...
```

Note: By default, the silent mode installs the Installation Manager to `/opt/IBM/IM`. The interactive procedure installs the Installation Manager to `/opt/IBM/InstallationManager`.

After you have installed the software and all patches, the WebSphere **`versionInfo.sh`** command should report the products shown in Table 3-3.

Table 3-3 Product versions installed as reported by the versionInfo.sh command

	WebSphere Business Monitor	WebSphere Business Services Fabric	WebSphere Process Server	WebSphere Dynamic Process Edition	WebSphere Business Compass
WebSphere Application Server - ND 7.0.0.7	✓	✓	✓	✓	✓
WebSphere Business Services Fabric Foundation Pack 7.0.0.0		✓		✓	
XML Feature Pack 1.0.0.1	✓	✓	✓	✓	

Archived

	WebSphere Business Monitor	WebSphere Business Services Fabric	WebSphere Process Server	WebSphere Dynamic Process Edition	WebSphere Business Compass
SCA Feature Pack 1.0.1.1	✓	✓	✓	✓	
WebSphere Process Server 7.0.0.1	✓	✓	✓	✓	
WebSphere Dynamic Process Edition 7.0.0.0				✓	
WebSphere Business Monitor 7.0.0.0	✓			✓	
WebSphere Business Modeler Publishing Server 7.0.0.0					✓
Lotus® Forms Turbo 3.5 (includes Embeddable WebSphere Application Server 6.1.0.23)					✓

3.2.2 Installing WebSphere Business Monitor

At the time of writing there are no fix packs available for WebSphere Business Monitor so we simply install the base product. We show how to install the product using the command line or the GUI. The command line is recommended for consistency and repeatability. It can also be scripted for unattended installation.

Archive

Installation using the command line

In this section we provide the steps to install using the command line.

Note: The instructions to install WebSphere Business Monitor silently are described in this Information Center article:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/topic/com.ibm.btools.help.monitor.inst.doc/inst/inst_silent.html

1. In <base_dir>/responsefiles/WBM copy the template response template_response.xml file to responsefile.xml.
2. Edit the newly created responsefile.xml and comment out the Internet repository (public.dhe.ibm.com) and set the following properties to false, as shown in Example 3-3 (we do not show the full file):
   - offering.service.repositories.areUsed
   - com.ibm.cic.common.core.preferences.searchForUpdates

Example 3-3 Changes required for the response file

```
<server>
  <repository location='../../IM/' temporary='true'/>
  <repository location="../../WAS_SYNC/" />
  <repository location="../../repository/" />
  <repository location="../../repository/" />
  <!-- repository
location="http://public.dhe.ibm.com/software/websphere/repositories/
" -->
</server>
…
<preference value="false"
name="com.ibm.cic.common.core.preferences.searchForUpdates" />
…
<preference value="false"
name="offering.service.repositories.areUsed" />
…
```

Archived

3. (Optional step) if you want to install separate instances of products then the Installation Manager must describe these installations using distinct package names. You must edit your response file to include the following changes:
   - Look for the profile stanza with attribute `id='IBM WebSphere Application Server - ND'` and change this to (for example) `'IBM WebSphere Application Server - ND_WBM'`.
   - Look for the import stanza and change the profile attribute to match the amended ID value.
   - Look for the offering stanzas and change the profile attribute to match the amended ID value.

   Example 3-4 shows the lines after they have been changed. The changes are highlighted.

Example 3-4 Making the package name specific to this installation

```
…
<profile installLocation='/opt/ibm/WebSphere/MonServer' id='IBM
WebSphere Application Server - ND_WBM'>
…
</profile>
…
<import profile="IBM WebSphere Application Server - ND_WBM"
type="WAS" />
…
<install>
        <offering profile="IBM WebSphere Application Server -
ND_WBM" id="com.ibm.websphere.XML.v10" />
        <offering profile="IBM WebSphere Application Server -
ND_WBM" id="com.ibm.ws.WBM" />
</install>
…
```

Archived

4. Edit the `run_templates` script to point to your new response file. Be careful to make the change to *both* lines (since the first one is just an echo statement), as shown in Example 3-5.

Example 3-5 Changing run_templates to use your new response file

```
echo "${IM_IMAGE}"/install  --launcher.ini
"${IM_IMAGE}"/silent-install.ini -input "${PROGDIR}"/responsefile.xml
-log "${WAS_LOCATION}"/wps/silent_install.log

"${IM_IMAGE}"/install  --launcher.ini "${IM_IMAGE}"/silent-install.ini
-input "${PROGDIR}"/responsefile.xml -log
"${WAS_LOCATION}"/wps/silent_install.log
```

5. (Optional) If you want to change the default installation directory, change the variable WAS_LOCATION in the `run_templates` file as appropriate.
6. Execute `run_templates`.

The installation takes a little bit of time. After the installation is complete you can verify the installation using:

`<install_dir>/bin/versionInfo.sh`

Change <*install_dir*> to suit your product installation.

For the Installation Manager you can verify with:

```
cd /opt/IBM/IM/eclipse
./IBMIM --launcher.ini silent-install.ini -version
```

Installation using the GUI

At the time of writing there are no fix packs available for WebSphere Business Monitor so we simply install the base product.

The following section describes how to install WebSphere Business Monitor binaries using the IBM Installation Manager. It is a two-step process. First we install WebSphere Application Server Network Deployment and the IBM Installation Manager, then we install the product itself.

1. Run the `<base_dir>/launchpad.sh` shell script.
2. Verify that installation directory points to your `<install_dir>`.

Archived

3. Install the base WebSphere Application Server Network Deployment binaries:
   a. Click the **Install WebSphere Application Server** link (Figure 3-1).
   b. Wait until the base WebSphere Application Server binaries have been installed. This takes a bit of time. Monitor the progress bar on the bottom of the window.

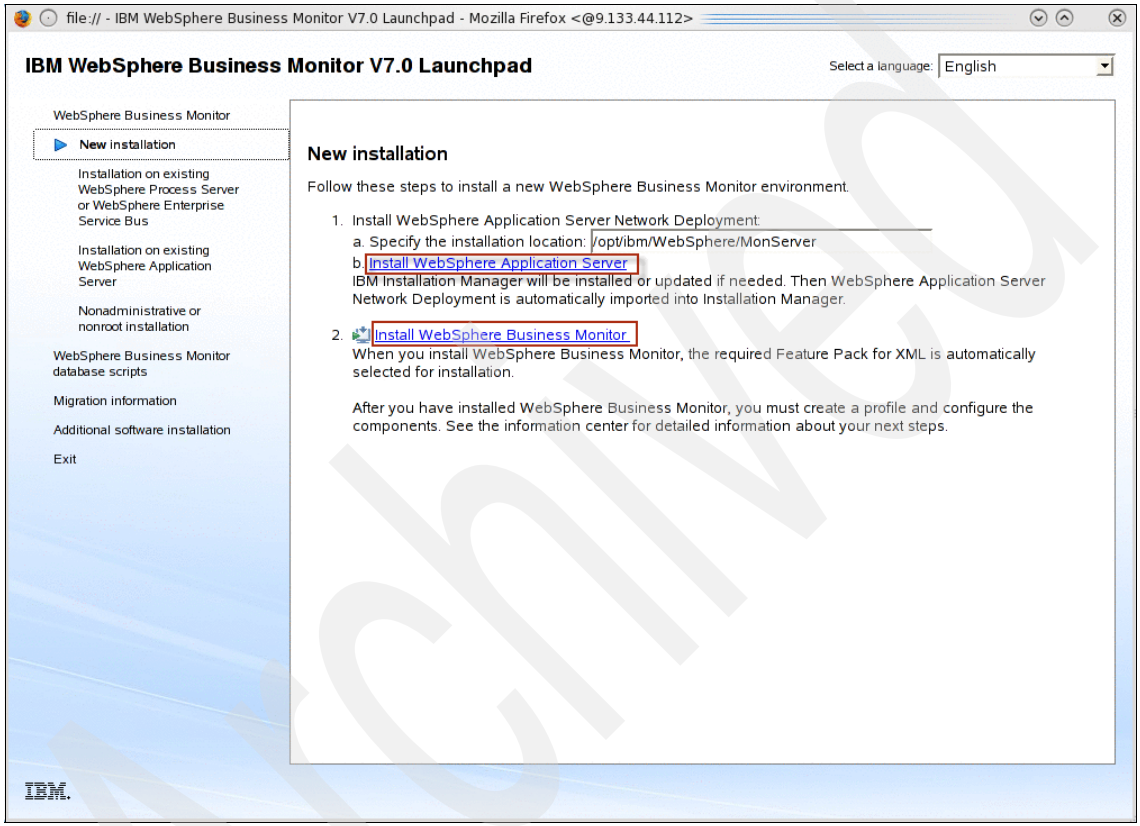


Figure 3-1 WebSphere Business Monitor launchpad

Archive

4. Once finished, install the WebSphere Business Monitor binaries by clicking the **Install WebSphere Business Monitor** link (Figure 3-2). The IBM Installation Manager is launched.

   a. If not already checked, select the **Version 7.0.0** sub-package of the IBM WebSphere Business Monitor package and the **Version 1.0.0.0** sub-package of the IBM WebSphere Application Server Feature Pack for XML package (Figure 3-2) and click **Next**.

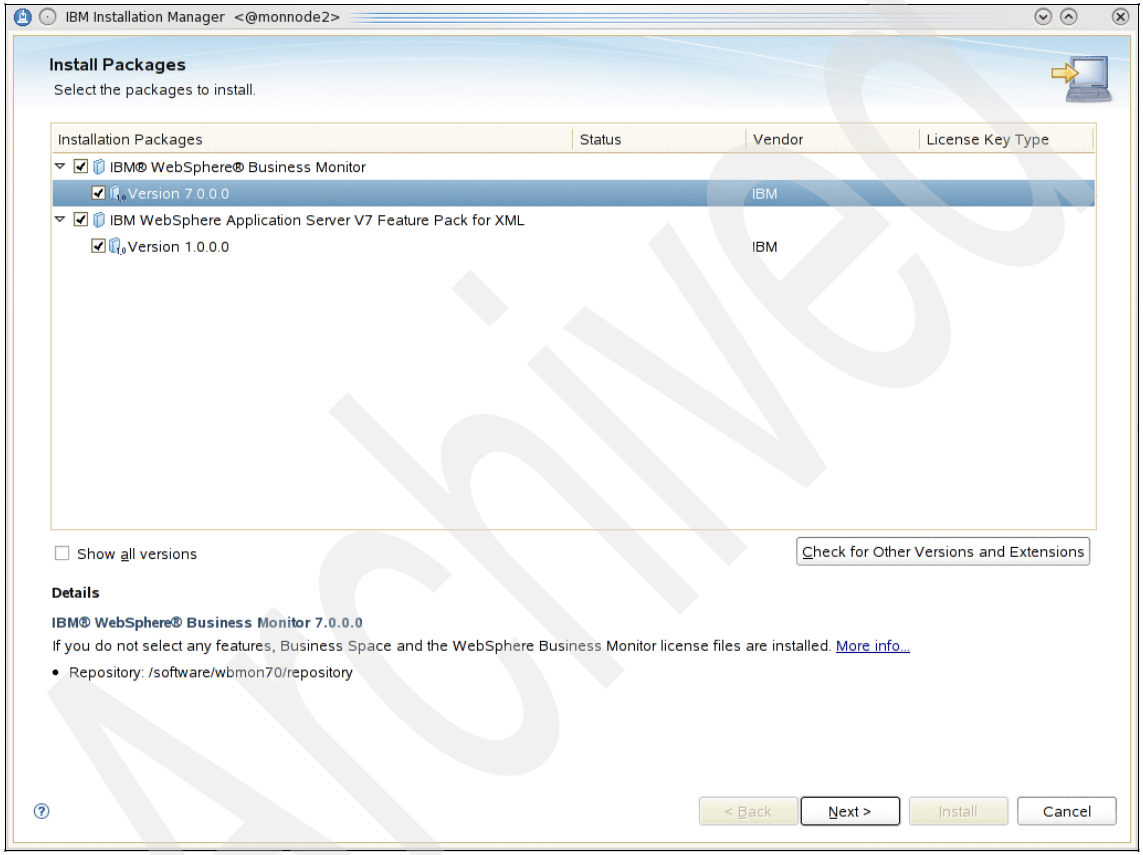


Figure 3-2 IBM Installation Manager package selection

   b. Accept the license agreement by clicking **I accept the terms in the license agreement** and click **Next**.

Archived

c. Verify the information on the Location page. **Use existing package group** should be selected and the installation directory should correspond to what you have chosen on the first page of the Launchpad window (Figure 3-3). Also verify that you have enough disk space on the destination drive. Once done click **Next**.

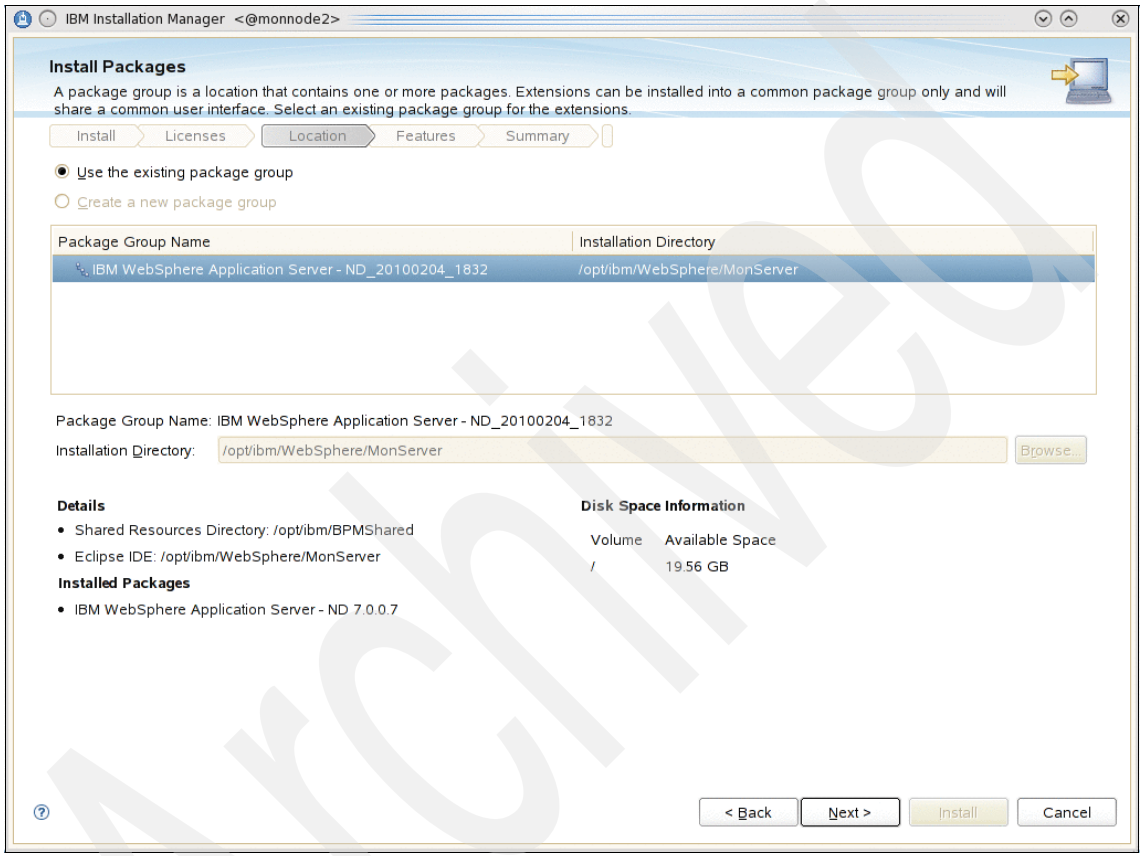


Figure 3-3 IBM Installation manager Location page

Archived

d. On the next Feature page, choose the feature that you want to install. Click **WebSphere Business Monitor Server**, **Alphablox**, and **IBM WebSphere Application Server V7 Feature Pack for XML**. Do not select any of the development profiles (Figure 3-4). Click **Next**.

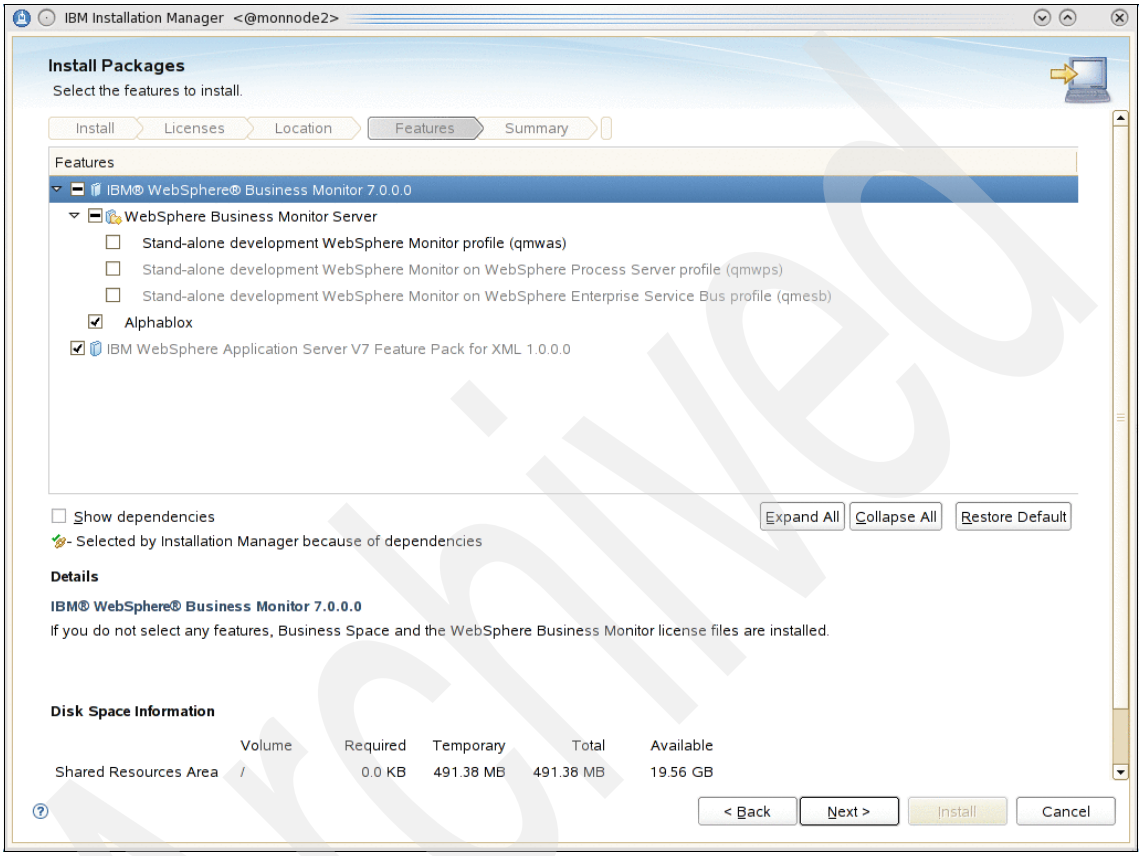


Figure 3-4 IBM Installation Manager Features page

e. Verify your installation options on the Summary page and click **Install**.

Archived

f. Verify that the installation was successful. The page should resemble Figure 3-5. Click **None** on the Which Package do you want to start? panel and click **Finish** to close the IBM Installation Manager. If the installation was not successful consult the log file by clicking **View Log File** to determine the cause of the failure.

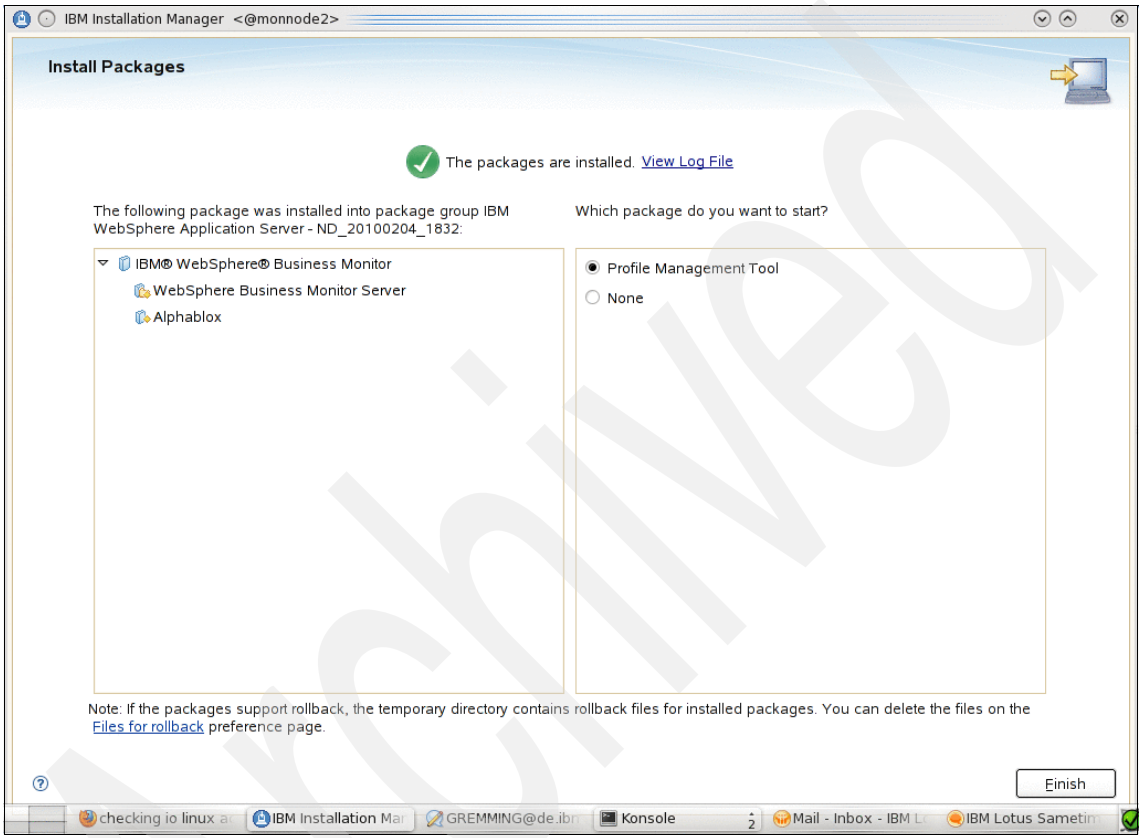


Figure 3-5 IBM Installation manager installation summary page

5. From the IBM WebSphere Business Monitor V7.0 Launchpad window, click **Exit**.

The IBM WebSphere Business Monitor launchpad has installed WebSphere Application Server Network Deployment, IBM Installation Manager, and WebSphere Business Monitor. There are no fix packs for WebSphere Business Monitor, so this completes the software installation. After the installation you can verify the installation using:

`<install_dir>/bin/versionInfo.sh`

Archived

Change <*install_dir*> to suit your product installation.

For the Installation Manager you can verify with:

```
cd /opt/IBM/InstallationManager/eclipse
./IBMIM --launcher.ini silent-install.ini -version
```

3.2.3 Installing WebSphere Process Server

Note: The instructions to install WebSphere Process Server V7.0.0 are described on the following Web page:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/topic/com.ibm.websphere.wps.doc/doc/tins_installingwpsim.html

Installation of the base product is identical to that in 3.2.2, “Installing WebSphere Business Monitor” on page 61. The response file template for WebSphere Process Server can be found in <base_dir>/responsefiles/WBI.

You must now apply the patches to the WebSphere Process Server product.

3.2.4 Installing WebSphere Business Services Fabric

Note: The instructions to install WebSphere Business Services Fabric by using the installation manager in the silent installation mode from a command line are described in the Information Center article on the following Web page:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/topic/com.ibm.ws.fabric.install.doc/fpi/concept/c_installsilent.html

Installation is identical to that in 3.2.2, “Installing WebSphere Business Monitor” on page 61. The response file template for WebSphere Business Services Fabric can be found in <base_dir>/responsefiles/WBI.

However, the following additional changes must also be made:

1. Edit the run_templates file and change the value of WAS_LOCATION to `/opt/ibm/WebSphere/Fabric/FoundationPack` or your preferred location.
2. Make the following additional changes to the `responsefile.xml`:
   - For the profile stanza with attribute `id='IBM WebSphere Application Server - ND'` change the attribute values for `installLocation` and `eclipseLocation` to `/opt/ibm/WebSphere/Fabric/FoundationPack` or your preferred location. It must match the value of WAS_LOCATION in the run_templates script.
   - In the install stanza add the following line:

     ```
     <offering profile="IBM WebSphere Application Server - ND"
     id="com.ibm.wbsf.fp" />
     ```

You can now proceed with the installation and then apply the patches to the WebSphere Process Server product.

We have provided these files, with the changes implemented, in Appendix A, “Additional material” on page 421.

Note: It is necessary to install the WebSphere Process Server 7.0.0.1 fix pack for the proper functioning of Fabric V7.0. Due to a problem when starting WebSphere Business Services Fabric development or test profiles we see transaction time-out errors and WTRN0006W and WTRN0124I messages in the system log. More specific details about the requirement of the fix pack can be found at:

`http://www-01.ibm.com/support/docview.wss?uid=swg21418029`

3.2.5 Installing WebSphere Dynamic Process Edition

Note: The instructions to install WebSphere Dynamic Process Edition by using the Installation Manager in the silent installation mode from a command line are described in the Information Center article at the following Web page:

`http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/topic/com.ibm.wdpe.install.server.doc/WDPEserver/task/t_silent.html`

Installation is identical to that in 3.2.2, “Installing WebSphere Business Monitor” on page 61. The response file template for WebSphere Dynamic Process Edition can be found in `<base_dir>/responsefiles/WDPE`, `<base_dir>/responsefiles/WBI`, and `<base_dir>/responsefiles/WBM`. You must edit all of these files.

You must now apply the patches to the WebSphere Process Server product.

3.2.6 Applying the fix packs

We explicitly list the installation steps. When we patch the product with the installation manager, the original repository *must* also be available. The steps below are for WebSphere Process Server because fix packs are currently available for this product.

Installing V7.0.0 Fix Pack 1 (V7.0.0.1) using the command line

Note: The instructions to silently install WebSphere Process Server and WebSphere Enterprise Service Bus V7.0.0 Fix Pack 1 (V7.0.0.1) are described at this URL:

`http://www-01.ibm.com/support/docview.wss?rs=2307&uid=swg27017658`

To install:

1. Use the template response file (`WPS_Patch.xml`) in Appendix A, “Additional material” on page 421, and copy it to `<fixpack_dir>/responsefile.xml`.
2. Edit the response file to point to your repositories. Example 3-6 shows our specific folders.

   Example 3-6 The repository configuration elements

   ```
   <repository location='/software/wps7000/repository'/>
   <repository location='/software/wps7000/WAS_SYNC'/>
   <repository location='/software/wps7001'/>
   <repository location='/software/wps7001/xml.1.0.0.1.fp'/>
   <repository location='/software/wps7001/sca.1.0.1.1.fp'/>
   ```

3. Ensure that the preference values for the following variables are set to false:
   - com.ibm.cic.common.core.preferences.searchForUpdates
   - offering.service.repositories.areUsed
4. Save your changes and run the Installation Manager:

   ```
   cd /opt/IBM/IM/eclipse
   ./IBMIM --launcher.ini silent-install.ini -input
   <fixpack_dir>/responsefile.xml -log <fixpack_dir>/install.log
   ```

At the end of the installation you can check whether the patches were successful with:

```
<install_dir>/bin/versionInfo.sh
```

Archived

Installing V7.0.0 Fix Pack 1 (V7.0.0.1) using the GUI

Note: The instructions to install WebSphere Process Server and WebSphere Enterprise Service Bus V7.0.0 Fix Pack 1 (V7.0.0.1) are described on the following Web page:

http://www-01.ibm.com/support/docview.wss?rs=2307&uid=swg27017658

To install:

1. Start the Installation Manager:

   ```
   /opt/IBM/InstallationManager/eclipse/IBMIM
   ```

2. From the Installation Manager window click **File** → **Preferences**.
3. In the Preferences windows select the WebSphere repository **public.dhe.ibm.com/software/webSphere/repositories** and click **Remove Repository**.
4. Un-check the **Search service repositories during installation and updates** preference.

Archived

5. To keep imported packages synchronized, add the original WebSphere product and the fix pack repositories. Click **Add Repository** to add the following repositories:

   – <base_dir>/repository
   – <base_dir>/WAS_SYNC
   – <fixpack_dir>/
   – <fixpack_dir>/xml.1.0.0.1.fp
   – <fixpack_dir>/sca.1.0.1.1.fp

   Figure 3-6 shows the result.

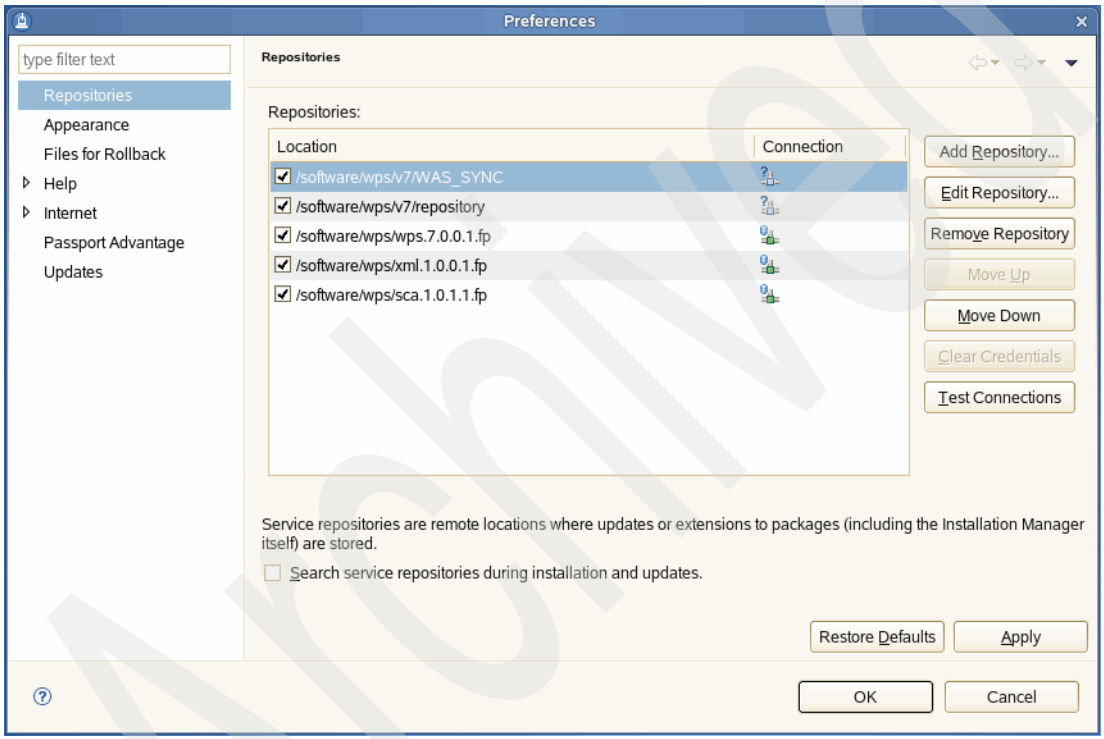


Figure 3-6 Repository preferences

6. Click **OK**.
7. From the Installation Manager window, click **Update**.

Archived

8. From the Update Packages window, the list of available package group names is shown in Figure 3-7. The installation manager can manage several package groups. Select the appropriate IBM WebSphere Application Server - ND_ and click **Next**.

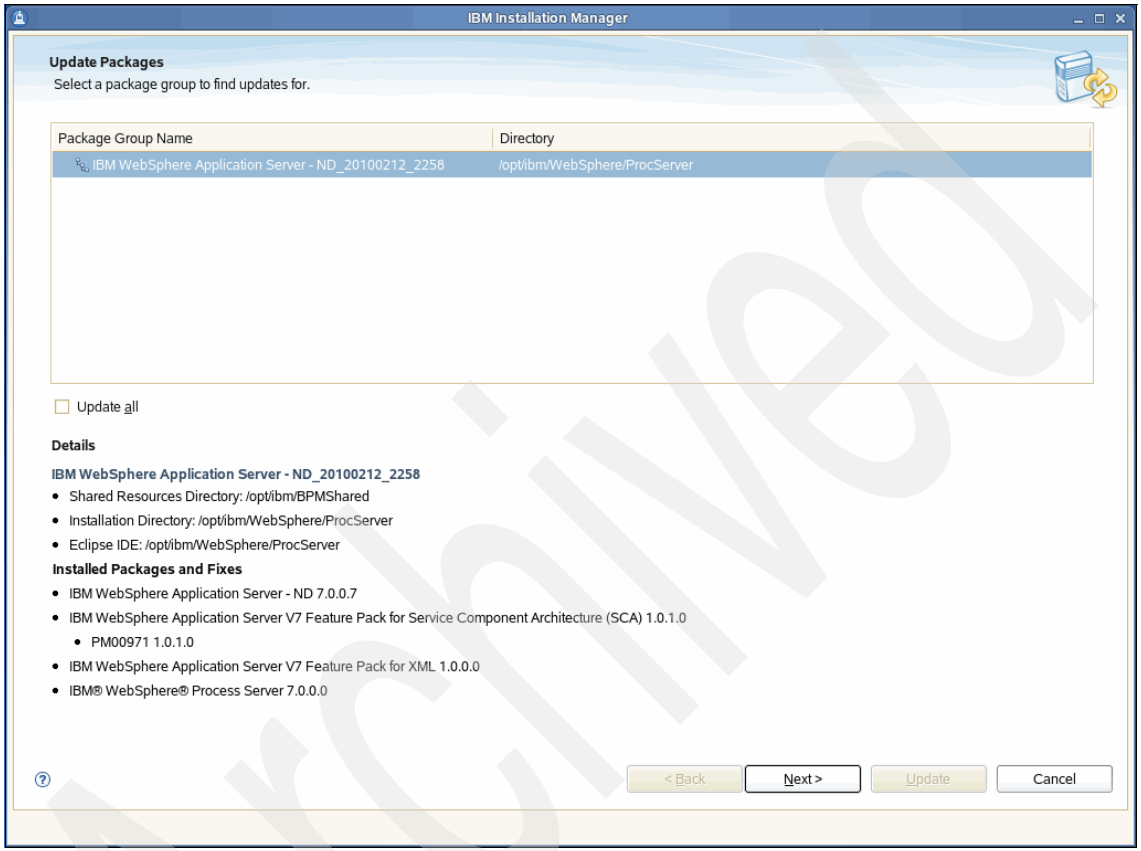


Figure 3-7 Package group name

Archived

9. From the Update window (Figure 3-8) you can select the updates or fixes to install. Click **Next**.

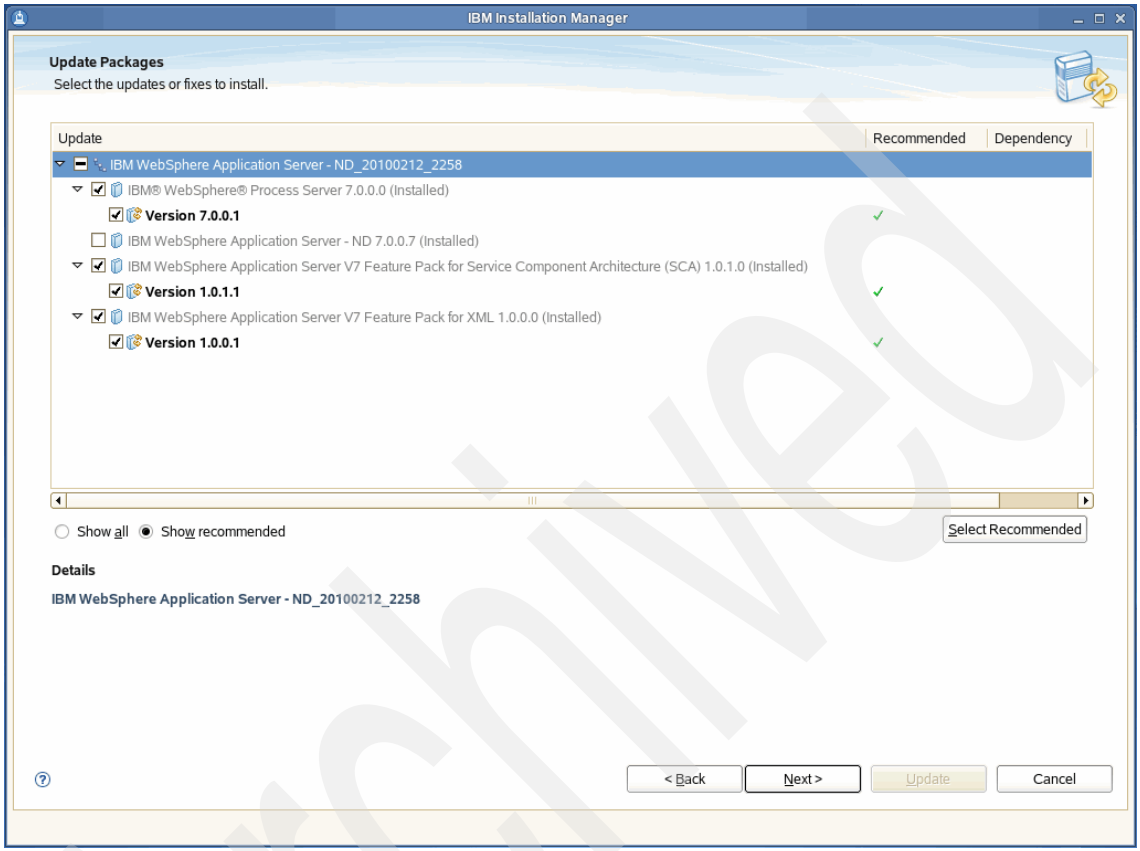


Figure 3-8 Update packages

10. From the Update Packages Licenses window, select **I accept the terms in the license agreements**. Click **Next**.

Archived

11. From the Update Packages Features window (Figure 3-9), you can choose to create standalone development profiles. In this environment, you will create these profiles after the installation of the software product. Click **Next**.

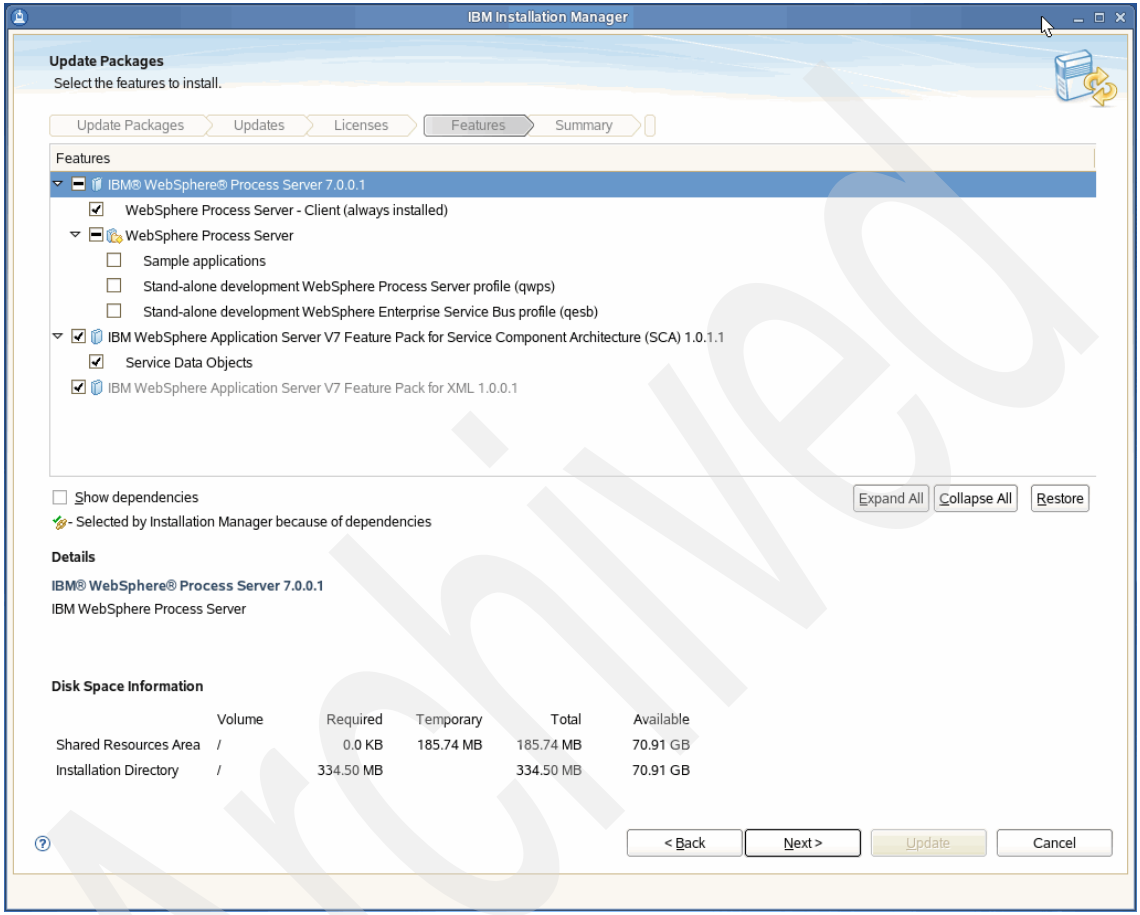


Figure 3-9 Update package features

12. In the Update Packages Summary window, click **Update**.
13. After the successful update of WebSphere Process Server, click **Finish**.
14. From the Installation Manager window, click **File** → **Exit**.

Archived

3.2.7 The all-in-one install and apply fix pack method

It is possible to install and patch a product in a single step. We describe how to do that in this section. We begin with the assumption that no products have been installed, including the installation manager.

Install and patch using the command line

To do this:

1. Ensure that you have created the <base_dir> and <fixpack_dir> repositories.
2. From Appendix A, “Additional material” on page 421, copy the template response file (WPS_BaseAndPatch.xml) to <base_dir>/responsefiles/WBI/responsefile.xml.
3. Edit this response file to point to your repositories. Example 3-7 shows our specific folders.

Example 3-7 The repository configuration elements

```
<repository location='/software/wps7000'/IM>
<repository location='/software/wps7001'/>
<repository location='/software/wps7001/xml.1.0.0.1.fp'/>
<repository location='/software/wps7001/sca.1.0.1.1.fp'/>
<repository location='/software/wps7000/repository'/>
<repository location='/software/wps7000/WAS_SYNC'/>
```

Note: You can add a temporary=’true’ parameter to each of the lines in the example and the repository will not be added to the installation manager.

4. Ensure that the preference values for the following variables are set to false:
   - com.ibm.cic.common.core.preferences.searchForUpdates
   - offering.service.repositories.areUsed

Archived

5. Edit the `run_templates` script to point to your new response file. Be careful be to make the change to *both* lines (since the first one is just an echo statement), as shown in Example 3-8.

Example 3-8 Changing run_templates to use your new response file

```
echo "${IM_IMAGE}"/install  --launcher.ini
"${IM_IMAGE}"/silent-install.ini -input "${PROGDIR}"/responsefile.xml
-log "${WAS_LOCATION}"/wps/silent_install.log

"${IM_IMAGE}"/install  --launcher.ini "${IM_IMAGE}"/silent-install.ini
-input "${PROGDIR}"/responsefile.xml -log
"${WAS_LOCATION}"/wps/silent_install.log
```

6. (Optional) If you want to change the default installation directory, change the variable WAS_LOCATION in the `run_templates` file as appropriate.
7. Execute `run_templates`.

At the end of the installation you can check whether the patches were successful with:

```
<install_dir>/bin/versionInfo.sh
```

Install and patch using the GUI

To do this:

1. In `<base_dir>/IM` make sure that you have made the changes shown in 3.2.1, “Create a software repository” on page 57.
2. From the `<base_dir>/IM` folder run the `launchpad.sh` script.
3. On the opening page, click **New installation**.
4. Check the install location and click the **Install WebSphere Application Server** link.
5. A pop-up window appears reminding you to close any existing running versions of the Installation Manager. Click **OK.**
6. The installation of WebSphere Application Server and the Installation Manager take several minutes to complete. A pop-up window is displayed at the end. Click **OK**.
7. When the install is finished click **Exit**.

You can verify that the installation manager is installed by opening a shell and using the following commands:

```
cd /opt/IBM/InstallationManager/eclipse
./IBMIM --launcher.ini silent-install.ini -version
```

Archived

We can now run the installation manager as a standalone application:

```
/opt/IBM/InstallationManager/eclipse/IBMIM
```

The welcome page is displayed. We nowmust change the list of repositories:

1. Click **File** → **Preferences** and remove the `public.dhe.ibm.com` repository by clicking the repository name and clicking **Remove Repository**.
2. Click **Add Repository** and enter `<base_dir>/repository` as the location. Click **OK** and it appears in the list of repositories.
3. Continue to add the following repositories:
   - `<base_dir>/WAS_SYNC`
   - `<fixpack_dir>`
   - `<fixpack_dir>/xml.1.0.0.1.fp`
   - `<fixpack_dir>/sca.1.0.1.1.fp`
4. Uncheck the Search service repositories during installation and updates option, then click **OK**. This returns youto the main Installation Manager page
5. Click **Install**.
6. Select all packages and click **Next**.
7. Click **I accept the terms in the license agreements** and click **Next**.
8. On the next page, select **Use the existing package group** and click **Next**.
9. On the Install Packages page click **Next**.
10. On the final page click **Install**.
11. The installation takes some time to complete. When it is complete click **Finish** and **File** → **Exit** to close the installation manager.

Use the following command to check the installation.

```
<install_dir>/bin/versionInfo.sh
```

3.2.8 Installing WebSphere Business Compass

Note: The instructions to install WebSphere Business Compass by using the Installation Manager in the silent installation mode from a command line are described in the Information Center article on the following Web page:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/topic/com.ibm.btools.help.modeler.collab.publish.inst.doc/topics/t_installing_silently_overview11.html

Archived

Installation is identical to that in 3.2.2, “Installing WebSphere Business Monitor” on page 61. The response file template for WebSphere Business Compass can be found in `<base_dir>/responsefiles/WBC`.

Note: The Lotus Forms Turbo software as shipped is packaged with its own integrated DB2 database and WebSphere Application Server Express run time. In order to not interfere with WebSphere Business Compass components, install the software on a separate Microsoft® Windows® server and integrate it into WebSphere Business Compass. Refer to Chapter 12, “WebSphere Business Compass” on page 393, for instructions about how to install Lotus Forms Turbo.

Warning: The Lotus Forms Turbo software shipped on the original media is not compatible with WebSphere Business Compass. You must download the software from IBM Fix Central:

`http://www.ibm.com/support/fixcentral/`

Navigate to **Lotus** → **Lotus Forms Turbo** → **3.5.1.0** → **Windows**. Choose **Browse** for fixes and download the `LotusFormsTurbo-3.5.1-Fixpack1-Win32` fix pack.

Chapter 12, “WebSphere Business Compass” on page 393, more fully discusses the details of installing WebSphere Business Compass.

Archive

3.3 Database and schema creation

In DB2 the (UNIX) login user is the same as the instance owner. The instance owner manages a number of databases. Each database can have different schemas (collections of tables), as shown in Table 3-4, where the MEDB database has four schemas. All these databases are normally managed by a single instance owner. In Table 3-4, where no explicit schema is listed the default is to use the instance owner as the schema name.

Table 3-4 Databases used and schema names

Database name	Schema name	Comments
WPRCSDB	COMMONDB	The common database.
BPEDB	BPC	The Business Process Choreographer database.
OBSVRDB	BPCREP	This database used by the Business Process Choreographer tools (BPC Observer and BPC Explorer).
MEDB	SCASYS	The Service Component Architecture (SCA) system messaging data store.
MEDB	SCAAPP	The SCA application messaging data store.
MEDB	CEIME	The Common Event Infrastructure (CEI) messaging data store.
MEDB	BPCME	The Business Process Choreographer messaging data store.
MEDB	MONME	The WebSphere Business Monitor messaging data store.
EVENT		The event database for CEI events. The default schema name is the same as the instance owner.
BSPCDB	IBMBUSSP	The database for Business Space powered by WebSphere and WebSphere Business Compass.
MONDB	MONITOR	The database used by WebSphere Business Monitor.
WBCDB		The database used by WebSphere Business Compass. The default schema name is the same as the instance owner

Archived

3.3.1 Using the database design tool

Note: WebSphere Business Compass does not use the database design tool (DDT).

This section describes the database configuration for the WebSphere process management software using the new database design tool. The DDT can generate most of the WebSphere Process Management component-specific database scripts, except for the Common Event Infrastructure database. The output of this tool is a collection of SQL and other scripts to create the databases, schemas, and so on (if required), and a configuration file. This configuration file can be used to:

- Provide database information during profile creation.
- Specify database configuration properties during deployment environment configuration.

The output from the DDT is verbose, but the input is straightforward. However:

- By default, all tables are created in the common database (using separate schema names). This is an anti-pattern and should *not* be used in production topologies because different databases have different performance requirements.
- We do not take the default names for schemas. In real production topologies it is likely that the database team will have naming standards for database names, schema names, and so forth.
- We create all the databases *before* we generate any deployment environments. Consequently, we must disable the flag to generate tables on first access to the database. This option is part of the configuration for the EVENT, Business Process Choreographer, business reporting, and messaging engine databases.
- The database design configuration file is a simple text file and you can inspect the various settings within it. Note that the file name *must* end in `.dbDesign` or the DDT tool will not accept it as a valid configuration file.

Archive

Note: The DDT does not perform the following tasks:

- It does not generate database scripts for the Common Event Infrastructure. The Common Event Infrastructure database scripts are created *after* the generation of the deployment environment.
- It does not generate a **`create database`** command for all database components (for example, the messaging engine database).
- It does not generate a master script to execute all database scripts.

In Appendix A, “Additional material” on page 421, we include the DDT configuration files used for the topologies in this book. You must edit this file and change the details of user name, DB2 server name, port, password, and so on, or you can create a new one by interactively running the design tool.

Once you create the configuration file or use an existing one, you can generate the SQL automatically using the -g option as follows (change <*install_dir*> to your product location):

```
cd <install_dir>/util/dbUtils
./DbDesignGenerator.sh -g wps.RMRS.dbDesign
```

Example 3-9 shows typical output.

Example 3-9 Running the DDT tool to generate SQL

```
[info] running DbDesignGenerator in generating mode...
[info] generating database scripts from wps.RMRS.dbDesign
[info] The script(s) have been generated in
/opt/ibm/WebSphere/ProcServer/util/dbUtils/WBI_CEI_ME_DB2-distributed-S
ibME
[warning] database scripts generation failed for [WBI_CEI_EVENT] due to
DDL provider is not available. You will not be able to generate SQL
scripts for the component : CEI
[info] The script(s) have been generated in
/opt/ibm/WebSphere/ProcServer/util/dbUtils/WBI_BPC_ME_DB2-distributed-S
ibME
[info] The script(s) have been generated in
/opt/ibm/WebSphere/ProcServer/util/dbUtils/WBI_BPC_DB2-distributed-BPC
[info] The script(s) have been generated in
/opt/ibm/WebSphere/ProcServer/util/dbUtils/WBI_SCA_APP_ME_DB2-distribut
ed-SibME
[info] The script(s) have been generated in
/opt/ibm/WebSphere/ProcServer/util/dbUtils/WBI_BSPACE_DB2-distributed-B
Space
```

```
[info] The script(s) have been generated in
/opt/ibm/WebSphere/ProcServer/util/dbUtils/WBI_BPCEventCollector_DB2-di
stributed-BPCReporting
[info] The script(s) have been generated in
/opt/ibm/WebSphere/ProcServer/util/dbUtils/WBI_SCA_SYS_ME_DB2-distribut
ed-SibME
[info] The script(s) have been generated in
/opt/ibm/WebSphere/ProcServer/util/dbUtils/WBI_CommonDB_DB2-distributed
-CommonDB
[info] thanks, quitting now ...
```

The output of the DDT generator run is a collection of directories with SQL and other scripts to create the databases and schemas. You must to hand over this entire collection of scripts for your database team to implement. For completeness, we list the steps and assume that during the DDT run we accepted the default folder names for all the SQL scripts.

We assume that the files have been copied to the home folder of the instance owner (`~/dbscripts`) on the DB2 server. In Table 3-5 we list the instance owners that we used in *each* topology. For example, to build a Remote Messaging and Remote Support environment for WebSphere Process Server we used the instance owner wps1inst for all the databases required by that topology.

Table 3-5 Instances owners used in our environments

Topology	Instance owner
WebSphere Process Server Remote Messaging and Remote Support	wps1inst
WebSphere Process Server single cluster	wps2inst
WebSphere Business Monitor cross cell	wbminst
WebSphere Business Monitor	moninst
WebSphere Business Services Fabric	fabinst
WebSphere Process Server double Remote Messaging and Remote Support	drmsinst
WebSphere Dynamic Process Edition Remote Messaging and Remote Support	dpeinst
WebSphere Business Compass	cominst

Archived

In Table 3-6 we list the databases needed by each product.

Table 3-6 Databases used by product

	WebSphere Business Monitor	WebSphere Business Services Fabric	WebSphere Process Server	WebSphere Dynamic Process Edition	WebSphere Business Compass
Common database		✓	✓	✓	
Business Process Choreographer database		✓	✓	✓	
Messaging Engine satabase	✓	✓	✓	✓	
Business Process Reporting database		✓	✓	✓	
Business Space database	✓	✓	✓	✓	✓
Event database	✓	✓	✓	✓	
Monitor database	✓			✓	
Compass database					✓

For WebSphere Business Services Fabric the databases required are the underlying databases used by WebSphere Process Server. For WebSphere Dynamic Process Edition the databases required are the underlying databases used by WebSphere Process Server and WebSphere Business Monitor.

3.3.2 The Common database

Create the common database as the appropriate instance owner on the DB2 server and verify:

```
cd ~/dbscripts/WBI_CommonDB_DB2-distributed-CommonDB
./configCommonDB.sh createDB
```

Enter your instance owner password when prompted.

Note: If you want to create the common database on the command line without being prompted for a user name use:

```
/createDBTables.sh wps1inst WPRCSDB COMMONDB createDB
```

The parameters are instance owner, database name, schema name, and the string ‘createDB’.

```
db2 “connect to WPRCSDB”
db2 “list tables for schema COMMONDB”
db2 “list tables”
db2 connect reset
```

The output of the “**list tables** …’ commands should be approximately 40 tables in the COMMONDB schema and for tables in the default schema.

3.3.3 The Business Process Choreographer database

Create the Business Process Choreographer database as the appropriate instance owner on the DB2 server and verify:

```
cd ~/dbscripts/WBI_BPC_DB2-distributed-BPC
db2 "CREATE DATABASE BPEDB USING CODESET UTF-8 TERRITORY EN-US PAGESIZE
8 K"
db2 "connect to BPEDB"
db2 "CREATE SCHEMA BPC"
db2 -tf createTablespace.sql
db2 -tf createSchema.sql
db2 "list tables for schema BPC"
db2 connect reset
```

The output of the “**list tables** …’ command should be approximately 173 tables.

3.3.4 The messaging engine database

Create the messaging engine database as the appropriate instance owner on the DB2 server and verify:

```
cd ~/dbscripts
db2 "CREATE DATABASE MEDB USING CODESET UTF-8 TERRITORY EN-US PAGESIZE
8 K"
db2 "connect to MEDB"
db2 -tf WBI_BPC_ME_DB2-distributed-SibME/DB2-distributed-SibME.sql
```

Archived

```
db2 "list tables for schema BPCME"

db2 -tf WBI_CEI_ME_DB2-distributed-SibME/DB2-distributed-SibME.sql
db2 "list tables for schema CEIME"

db2 -tf WBI_SCA_APP_ME_DB2-distributed-SibME/DB2-distributed-SibME.sql
db2 "list tables for schema SCAAPP"

db2 -tf WBI_SCA_SYS_ME_DB2-distributed-SibME/DB2-distributed-SibME.sql
db2 "list tables for schema SCASYS"
db2 connect reset
```

The output of the "**list tables** …' commands should be nine tables per schema.

3.3.5 The business process reporting database

Create the business process reporting database as the appropriate instance owner on the DB2 server and verify:

```
cd ~/dbscripts/WBI_BPCEventCollector_DB2-distributed-BPCReporting
db2 "CREATE DATABASE OBSVRDB USING CODESET UTF-8 TERRITORY EN-US"
db2 "connect to OBSVRDB"
db2 "CREATE SCHEMA BPCREP"
db2 -tf createTablespace_Observer.sql
db2 -tf createSchema_Observer.sql
db2 "list tables for schema BPCREP"
db2 connect reset
```

The output of the "**list tables** …' command should be eight tables.

3.3.6 The Business Space database

Note: The configBusinessSpaceDB.sh script used below prompts you for a user name. You can prevent this by editing this file and adding the user name to the line starting:

```
USER_NAME=
```

Create the Business Space database as the appropriate instance owner on the DB2 server and verify:

```
cd ~/dbscripts/WBI_BSPACE_DB2-distributed-BSpace
./configBusinessSpaceDB.sh createDB
```

Archived

Note: An alternative to this command is to run:

```
./createDBTables.sh  wps1inst BSPCDB createDB
```

The parameters are instance owner, database name, and the string 'createDB'.

Both of the above commands **configBusinessSpaceDB.sh** and **createDBTables.sh** prompt for the instance owner password:

```
db2 "connect to BSPCDB"
db2 "list tables for schema IBMBUSSP"
db2 connect reset
```

The output of the "**list tables** …' command should be 29 tables.

3.3.7 The EVENT database

Note: The Common Event Infrastructure tables is *only* created after the generation of a specific *deployment environment*.

The generated Common Event Infrastructure scripts are created in the following location:

```
<install_dir>/profiles/<dmgr_name>/databases/event/<topology>/dbscripts
/db2
```

Where the values highlighted are specific to your deployment environment. Copy these scripts over to the DB2 server (into a folder called ~/dbscripts/event) as the instance owner and run the following on the DB2 server:

```
cd ~/dbscripts/event
echo "wpsinst:itso4you" | ./dbConfigureCr.sh 2 | tee output.log
```

The echo parameters are the instance owner and the password. Change these to your values.

3.3.8 The MONITOR database

Note: This database is only required by WebSphere Business Monitor.

Archived

Perform the following actions on the DB2 server:

```
cd ~/dbscripts/Monitor_DB_DB2-distributed-Monitor
db2 -tf createDatabase.sql
db2 “connect to MONDB”
db2 “CREATE SCHEMA MONITOR”
db2 -tf createTables.sql
db2 “connect to MONDB”
db2 -tf dim_time_entries_db2.sql
db2 "list tables for schema MONITOR"
db2 connect reset
```

The output of the “**`list tables`** …’ command should be 80 tables.

Note: In a standalone WebSphere Business Monitor installation the messaging engine database does not yet exist. You must create it before running the commands below.

```
cd ~/dbscripts/WBI_Monitor_ME_DB2-distributed-SibME
db2 “connect to MEDB”
db2 -tf DB2-distributed-SibME.sql
db2 “list tables for schema MONME”
db2 connect reset
```

Note: If WebSphere Business Monitor is installed in its own cell then we also need a CEI messaging schema in the messaging engine database and we must create the Business Space database. These scripts are generated as part of the DDT tool.

3.3.9 The business compass database

Note: This database is only required by WebSphere Business Compass.

Warning: WebSphere Business Compass also needs a Business Space database. However, this database can only be created once Business Space has been installed. You cannot use the DDT tool for WebSphere Business Compass.

This product does not use the DDT tool, but the ddl script to run is created as part of the deployment manager profile creation. Once you have created your deployment manager you must copy the `createWPBSDB.ddl` file to the DB2 server

as the instance owner. You can then perform the following tasks on the DB2 server:

1. Edit the `createWPBSDB.ddl` file and change the value for $DBPWD$ to your password (our complete line is shown below). Connect to WBCDB user cominst using '`itso4you`'.
2. Now you can proceed with the database installation:

```
db2 -tf createWPBSDB.ddl
db2 "connect to WBCDB"
db2 "list tables"
db2 connect reset
```

3.4 Creating a deployment manager profile

In this section we create a deployment manager and configure it to use LDAP. We also show how to add administrative groups. In production topologies, use LDAP as the user registry because it provides a consistent repository for users and it is required to deal with several of the advanced functions of the human task manager.

3.4.1 Creating a deployment manager profile

In this section we discuss how to create a deployment manager using the command line or the GUI.

Using the command line

The following steps describe the procedure to create a WebSphere Business Monitor deployment manager profile using the **`manage profiles`** command:

1. Log in to the deployment manager machine.
2. Create a `dmgr.resp` response file in the `<base_dir>` directory with content similar to that shown in Example 3-10. Change the values to suit your product installation (sample files are given in the additional materials).

Example 3-10 Creating a WebSphere Business Monitor deployment manager profile

```
create
templatePath=/opt/ibm/WebSphere/MonServer/profileTemplates/wbmonitor/dm
gr
profilePath=/opt/ibm/WebSphere/MonServer/profiles/mondmgr
profileName=mondmgr
cellName=moncell
```

```
nodeName=mondmgrnode
enableAdminSecurity=true
adminUserName=monadmin
adminPassword=itso4you
wbmDBType=DB2_Universal
wbmDBDelayConfig=true
wbmDBSchemaName=MONITOR
wbmDBName=MONDB
wbmDBUserId=moninst
wbmDBPassword=itso4you
wbmDBJDBCClasspath=/opt/ibm/WebSphere/MonServer/universalDriver.wbm/lib
wbmDBHostName=itsodb2.itso.ral.ibm.com
wbmDBServerPort=50015
wbmDBDriverType=4
configureBSpace=true
startingPort=39443
```

3. Run the **manage profiles** command from your product install location:

```
<install_dir>/bin/manageprofiles.sh -response <base_dir>/dmgr.resp
```

At the end of the installation the following message is displayed:

```
INSTCONFSUCCESS: Success: Profile mondmgr now exists. Please consult
/opt/ibm/WebSphere/MonServer/profiles/mondmgr/logs/AboutThisProfile.txt
for more information about this profile.
```

Note: The template file above is for WebSphere Business Monitor. You must change several of the options (for example, templatePath) and remove several of them to suit other products. We include all deployment manager profile options for each product in Appendix A, “Additional material” on page 421.

Important: For topologies that use the DDT configuration file (for example, WebSphere Process Server) you can use the wbidbDesign=/tmp/wps.RMRS.dbDesign option, but the path *must* be an absolute path to the file.

Archived

Using the GUI

Note: On distributed systems, using the 64-bit version of the software, the GUI based profile management tool is unavailable. The example below uses the 32-bit software. On System z®, the 64-bit pmt tool exists but the 32-bit tool does not.

To use the GUI:

1. Log in to the deployment manager machine and start the Profile Management Tool:

   ```
   <install_dir>/bin/ProfileManagement/pmt.sh
   ```

2. A welcome page displays. Click **Launch Profile Management Tool**.
3. On the Profiles window, click **Create**.

Archived

4. The Environment Selection window displays a list of available environments to create. Navigate to **WebSphere Process Server → Process server deployment manager**, as shown in Figure 3-10. Click **Next**.

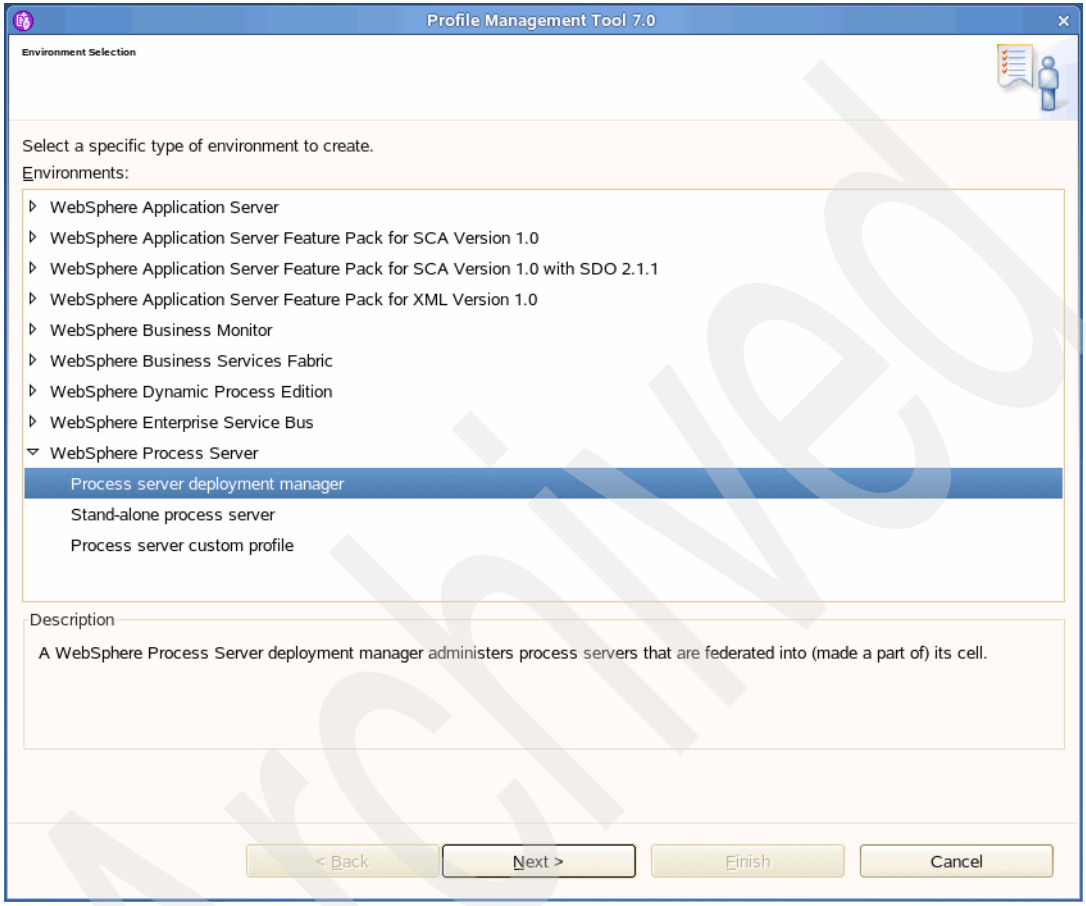


Figure 3-10 Choosing the type of deployment manager

Archived

5. From the Profile Creation Options window, select **Advanced profile creation**, as shown in Figure 3-11. Click **Next**.

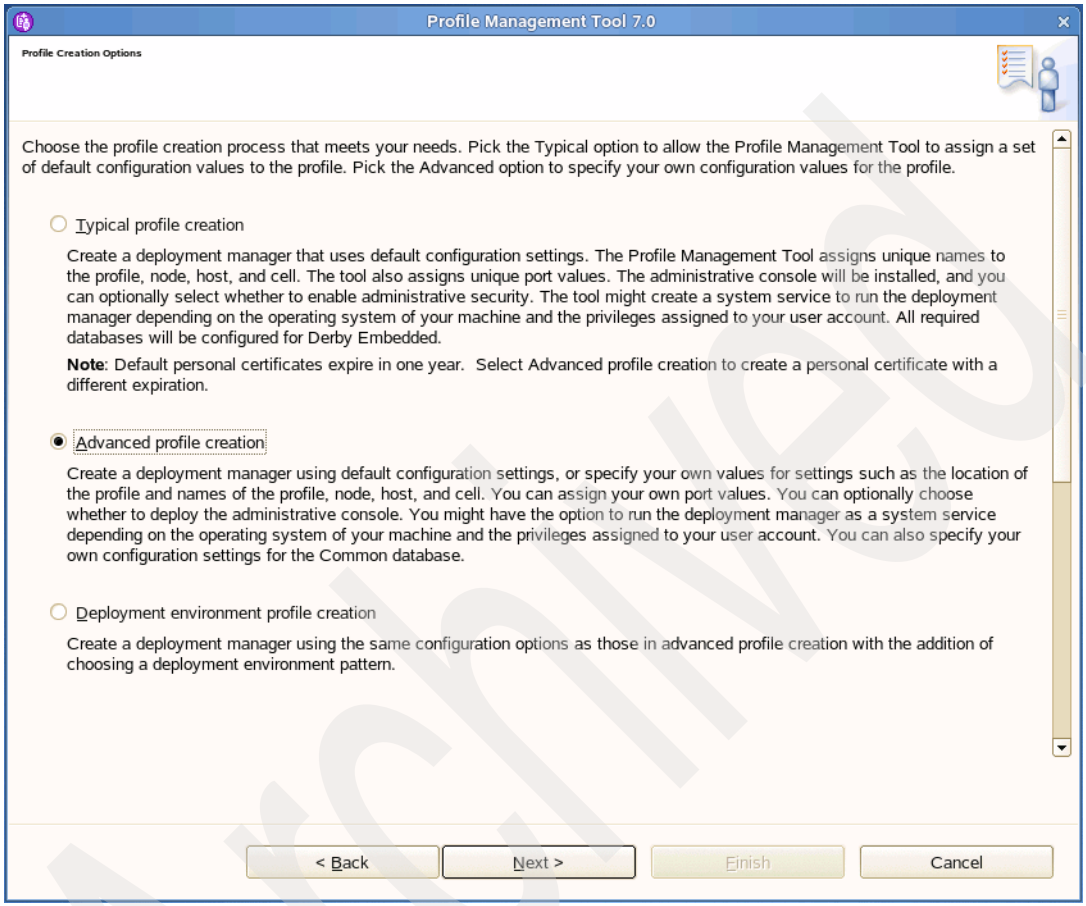


Figure 3-11 Profile Creation Options

6. From the Optional Application Deployment window, accept the default option to **Deploy the administrative console**. Click **Next**.
7. From the Profile Name and Location window, enter the values for the profile name and profile directory as follows, then click **Next**:
   - Profile name: `wps1dmgr`
   - Profile directory: `/opt/ibm/WebSphere/ProcServer/profiles/wps1dmgr`

Archived

8. From the Node, Host, and Cell Names window, enter the values for the node name, host name, and cell name as follows, then click **Next**:
   - Node name: `wps1dmgr`
   - Host name: `itsodmgr`
   - Cell name: `wpscell01`
9. From the Administrative Security window, select the **Enable administrative security** check box. Enter the values for the user name and password, then click **Next**.
10. From the Security Certificate (Part 1) window, accept the default to **Create a new default personal certificate** and to **Create a new root signing certificate**, then click **Next**.
11. From the Security Certificate (Part 2) window, accept the default certificate information. The default key store password is WebAS. Click **Next**.

Archived

12. From the Port Values Assignment, enter the port values for the deployment manager. Make sure that these ports are available and do not conflict with any process on this machine. Figure 3-12 shows the default ports. Click **Next**.

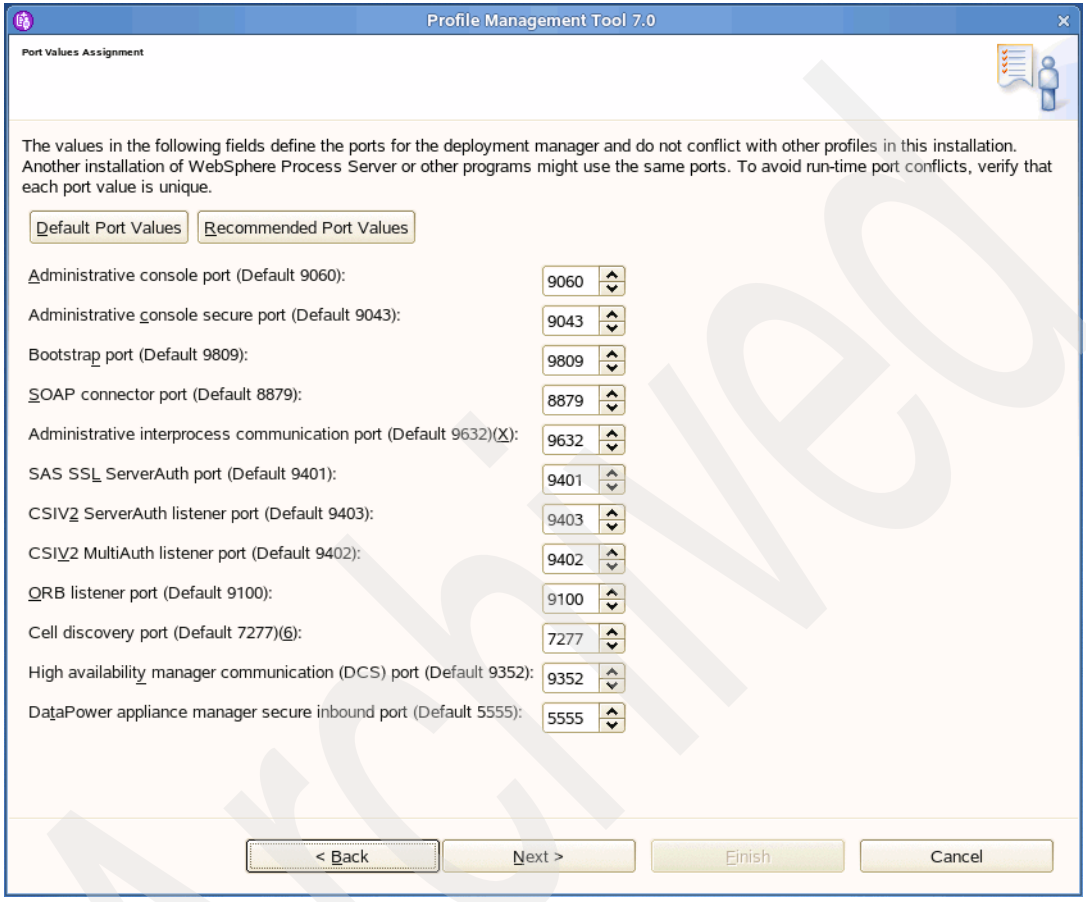


Figure 3-12 Default port values for the deployment manager

Note: The default SOAP connector port (8879) is used by the custom node to federate to the deployment manager.

13. From the Service Definition window, accept the default to not run the process server process as a service. Click **Next**.

Archived

14. From the Database Design window, check **Use a database design file for database configuration** and **Delay execution of database scripts**. Enter the fully qualified path name of the database design file. The database design file must have the .dbDesign extension, as shown in Figure 3-13. Click **Next**.

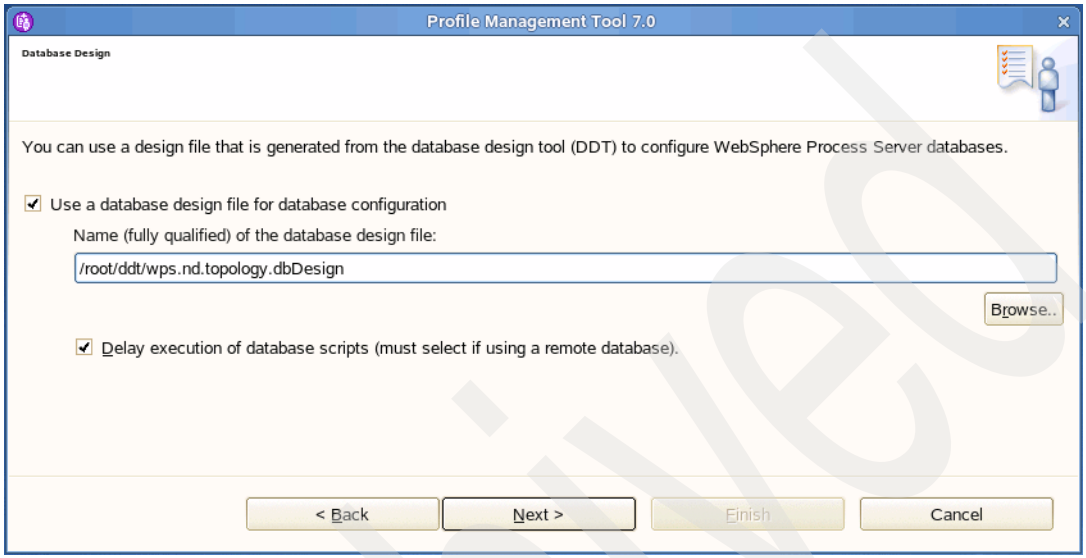


Figure 3-13 Database design

15. From the Profile Creation Summary window, verify the information. Click **Create**.
16. From the Profile Creation Complete window, the `Profile Management Tool created the profile successfully` message is displayed. Un-check the **Launch the First Steps console** box. Click **Finish**.
17. From the Profiles window, the created deployment manager profile is listed. Click **File** → **Exit**.

Archived

3.4.2 Configuring LDAP as the user account registry

The steps in this section describe the procedure to configure LDAP as the user registry. Table 3-7 lists the users expected within LDAP.

Table 3-7 LDAP users by product

	WebSphere Business Monitor	WebSphere Business Services Fabric	WebSphere Process Server	WebSphere Dynamic Process Edition	WebSphere Business Compass
Primary administrative user for WebSphere Process Server	wbmadmin	wpsadmin	wpsadmin	dpeadmin	
User for the business flow and human task monitor role		bpcmon	bpcmon	bpcmon	
JMS API Authentication		jmsapi	jmsapi	jmsapi	
Authentication for HTM Escalation		escalation	escalation	escalation	
Authentication for the business flow manager and human task manager administrative jobs		cleanup	cleanup	cleanup	
User for connection to CEI Messaging Bus	ceijms	ceijms	ceijms	ceijms	
User for connection to SCA Messaging Buses		sca	sca	sca	
User for connection to BPC Messaging Bus		bpcjms	bpcjms	bpcjms	
Alphablox	abx				

Archive

The steps are:

1. Start the deployment manager if it is not already started:

   `<install_dir>/profiles/<dmgr_name>/bin/startManager.sh`

2. Log in to the Integrated Solutions Console.
3. Navigate to **Security** → **Global security**.
4. Make sure that **Enable administrative security** is selecte,d as shown in Figure 3-14

Figure 3-14 Enable administrative security

5. In the User account repository panel (Figure 3-15), from the Available realm definitions list, the default realm is Federated repositories. Click **Configure**.

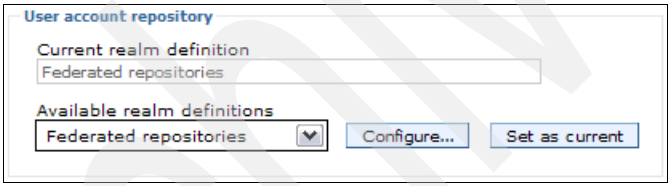


Figure 3-15 User account repository

6. From the Federated repositories window, under Related Items, click **Manage Repositories**, as shown in Figure 3-16.

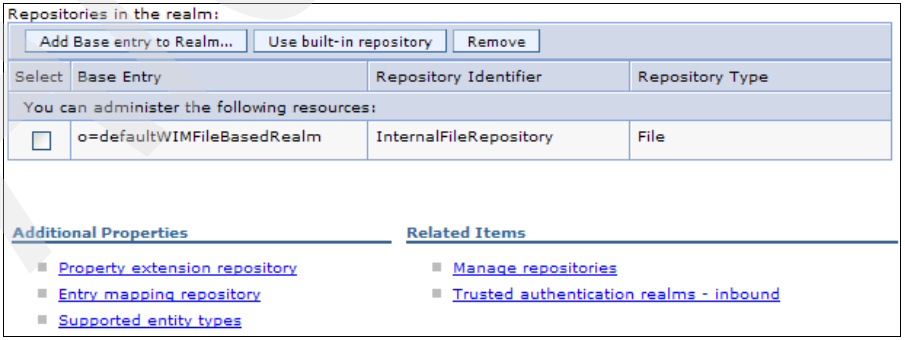


Figure 3-16 Repositories in the realm

Archived

7. On the Manager repositories panel click **Add**, as shown in Figure 3-17.

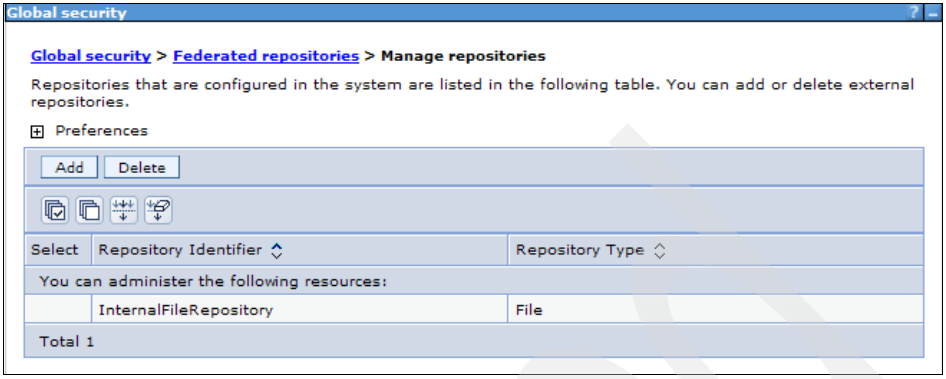


Figure 3-17 Manage repositories

Archived

8. On the New Repository panel (Figure 3-18), set the following attributes:
   a. Set Repository identifier to `ITSO LDAP`.
   b. For Directory type select **IBM Tivoli Directory Server**.
   c. Set Primary host name to `itsoldap.itso.ral.ibm.com`.
   d. Ensure that Port is set to 389.
   e. Set Bind distinguished name to `cn=wpsadmin,ou=IBM,o=ITSO`.
   f. Set Bind password to `itso4you`.
   g. Set Login properties to `uid;cn`.
   h. Click **Apply**.
   i. Click **Save**.

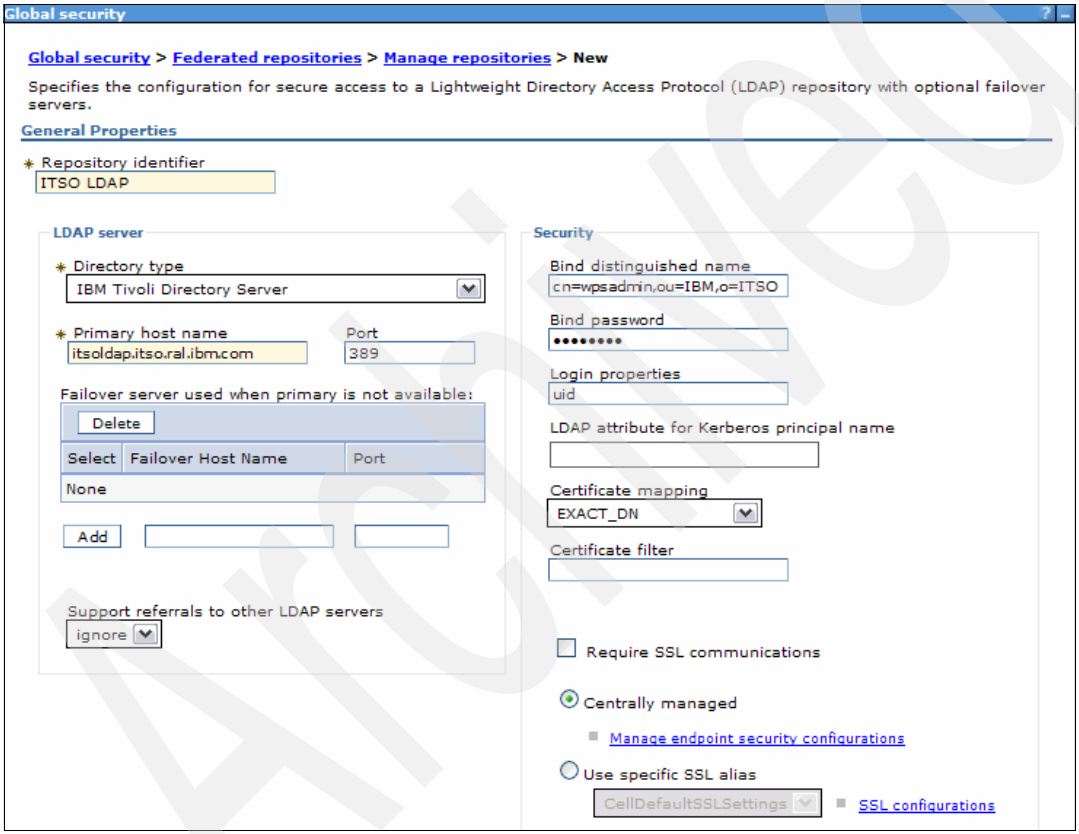


Figure 3-18 Creating a new LDAP repository

Archived

There are now two repositories, as shown in Figure 3-19.

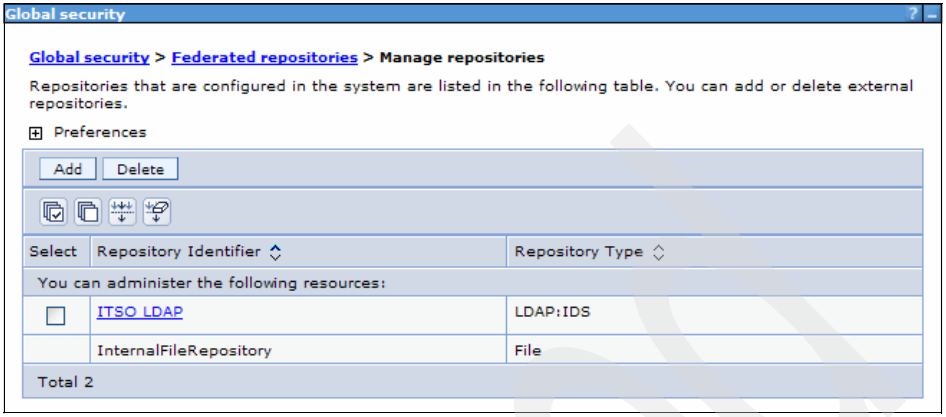


Figure 3-19 Manage repositories

9. Click the **Federated Repositories** link at the top of the page.
10. From the Federated Repositories panel, in the Repositories in the realm section, click **Add Base entry to Realm**, as shown in Figure 3-20.

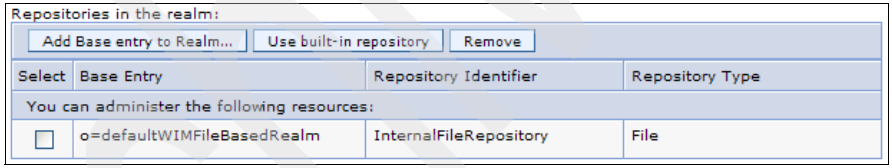


Figure 3-20 Repositories in the realm

Archive

11. From the Repository reference panel (Figure 3-21), enter the following attributes:

    a. For the “Distinguished name of a base entry that uniquely identifies this set of entries in the realm” feild enter `ou=IBM,o=ITSO`

    b. For the “Distinguished name of a base entry in this repository” field enter `ou=IBM,o=ITSO`

    c. Click **OK**.

    d. Click **Save**.

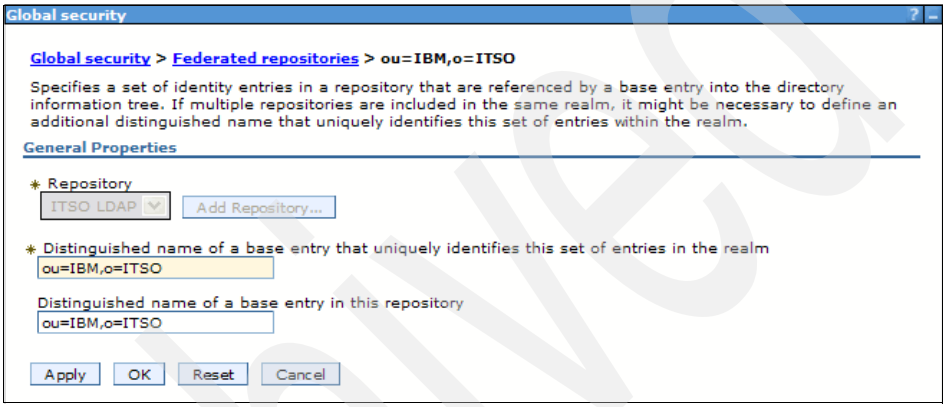


Figure 3-21 ITSO LDAP repository

The Federated Repositories panel displays two repositories, as shown in Figure 3-22.

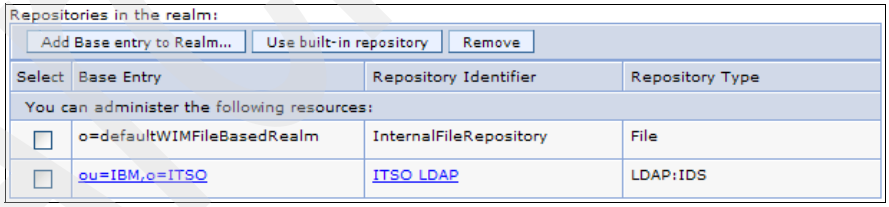


Figure 3-22 Federated repositories with InternalFileRepository and LDAP

12. From the Federated Repositories panel, in the Repositories in the realm, remove the InternalFileRepository repository as follows:

    a. Select the **InternalFileRepository** check box.
    b. Click **Remove**.
    c. Click **Save**.

Archived

The Federated Repositories panel only shows the ITSO LDAP repository, as shown in Figure 3-23.

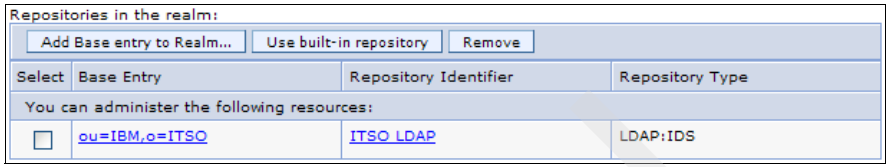


Figure 3-23 Federated repositories with only LDAP

13. From the Federated Repositories panel, enter the following attributes:
    a. Set the realm name to `itsoldap`.
    b. Set the primary administrative user name to `wpsadmin`.
    c. Click **Server identity that is stored in the repository**.
    d. Set Server user ID or administrative user to `cn=wpsadmin,ou=IBM,o=ITSO`
    e. Set the password to `itso4you`.
    f. Click **OK**.
    g. Click **Save**.
14. Log out from the Integrated Solutions Console.
15. Restart the deployment manager.

3.4.3 Adding administrative groups

To do this:

1. Log on to the Integrated Solutions Console.
2. Navigate to **Users and Groups** → **Administrative group roles** and click **Add**.
3. On the Roles select panel select appropriate roles for your topology.
4. Click **Search**. The available groups are displayed from the LDAP repository.
5. Select the group for this role (for example, wasadmins for WebSphere Process Server) and click the arrow to add to “Mapped to role.”

Archived

6. Click **OK**. The panel is shown in Figure 3-24.
7. Save your configuration.

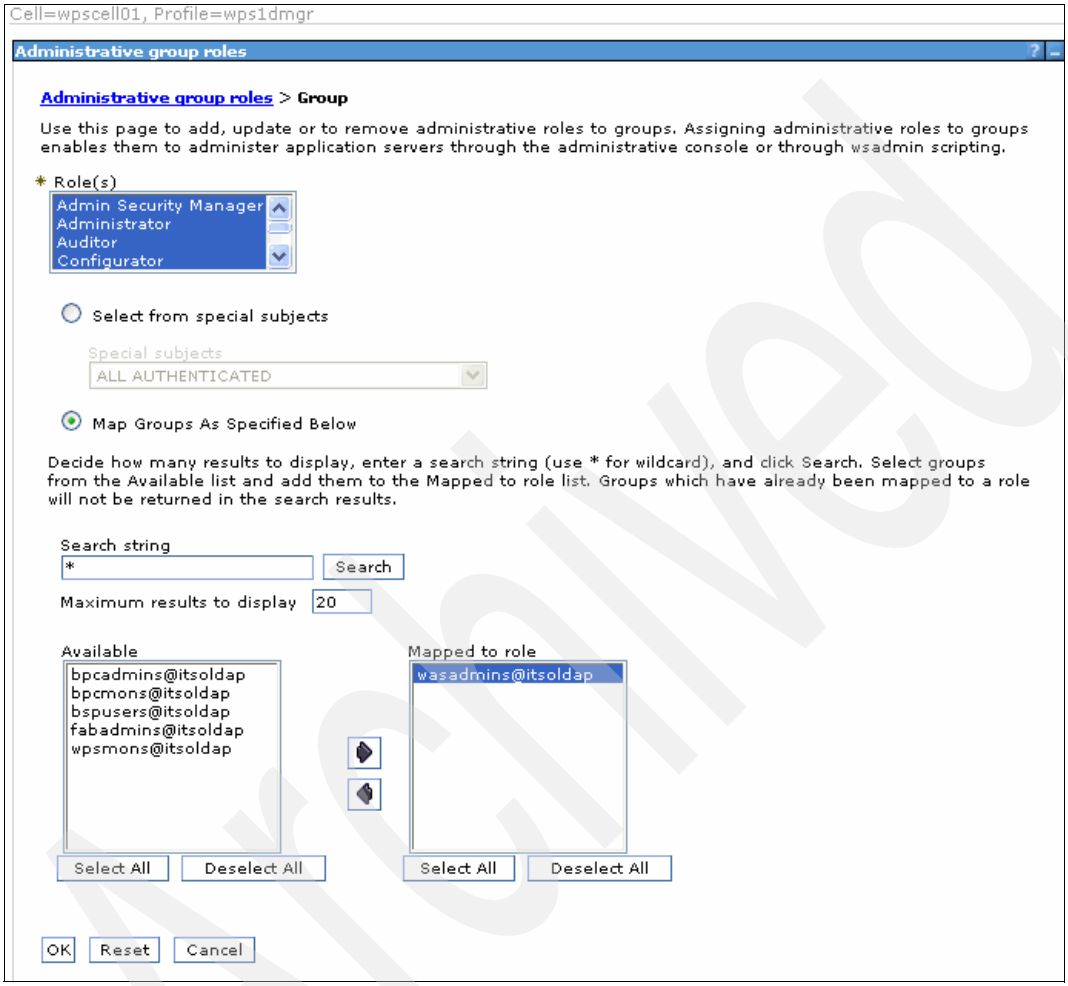


Figure 3-24 Adding administrative groups

Archived

You should now be able to logi n to the Integrated Solutions Console using any of the users defined in the wasadmins group, as shown in Figure 3-25.

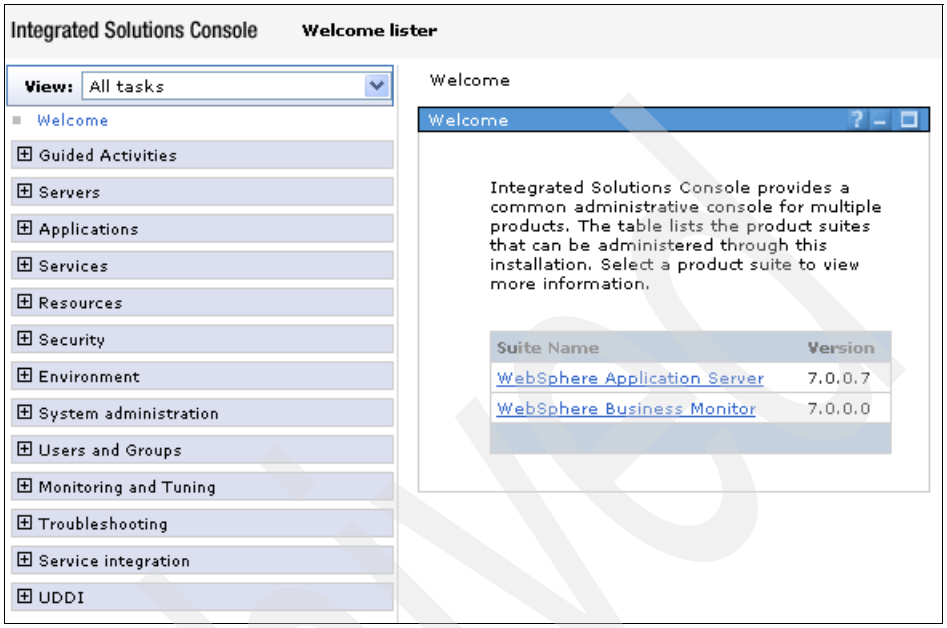


Figure 3-25 Log in as a member of the wasadmins group

3.5 Creating custom profiles

We now create our nodes and federate them into the cell.

3.5.1 Creating a custom profile

We discuss how to create a custom profile using the command line or using the GUI.

Archive

Using the command line

Note: The template file below is for WebSphere Business Monitor. You must change some of the options (for example, templatePath) to suit other products. We include all deployment manager profile options for each product as part of the additional materials.

To use the command line:

1. Create a response file `node.resp` in the `<base_dir>` directory with content similar to that shown in Example 3-11. Change the values to suit your product installation (sample files are given in the additional materials).

   Example 3-11 monnode1 profile response file

   ```
   create
   templatePath=/opt/ibm/WebSphere/MonServer/profileTemplates/wbmonitor
   /managed
   profileName=monnode1
   nodeName=monnode1
   federateLater
   wbmDBType=DB2_Universal
   ```

2. Run the **manage profiles** command from your product install location:

   ```
   <install_dir>/bin/manageprofiles.sh -response <base_dir>/node.resp
   ```

At the end of the installation the following message is displayed:

```
INSTCONFSUCCESS: Success: Profile monnode1 now exists. Please consult
/opt/ibm/WebSphere/MonServer/profiles/monnode1/logs/AboutThisProfile.tx
t for more information about this profile.
```

For the second node use the same response file but change the values for:

- profileName
- nodeName
- hostName
- cellName
- profilePath

Using the GUI

Note: On distributed systems using the 64-bit version of the software, the GUI-based profile management tool is unavailable. The example below uses the 32-bit software. On System z, the 64-bit pmt tool exists but the 32-bit tool does not.

Archive

To use the GUI:

1. Log in to your node and launch the profile management tool:

   ```
   <install_dir>/bin/ /pmt.sh
   ```

2. On the welcome page click the **Launch Profile Management Tool**.
3. On the Profile page click **Create**, and on the Environment Selection panel choose the appropriate custom profile and click **Next**. An example is shown in Figure 3-26.

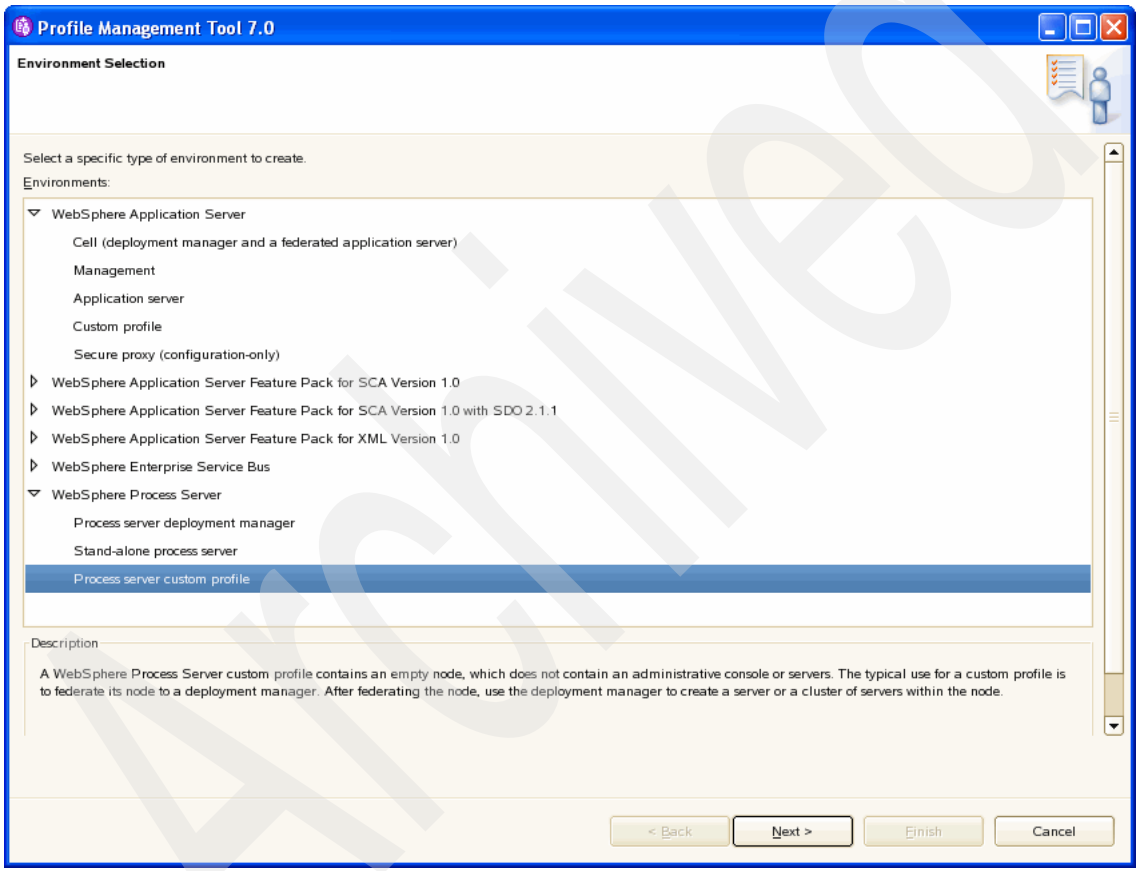


Figure 3-26 Selecting a WebSphere Process Server custom profile

4. On the Profile Creation Options panel, select **Advanced Profile Creation** and click **Next**. We use this option because we will be using DB2 rather than Derby.

Archived

5. On the Profile Name and Location panel, enter your values for Profile name and Profile directory (for example, we used `wps1node1` and `/opt/ibm/WebSphere/ProcServer/profiles/wps1node1`) and click **Next**.
6. On the Node and Host Names panel, enter your values for Node name and Host name or accept the defaults and click **Next**.
7. On the Federation panel, tick the **Federate this node later** box and click **Next**.
8. On the Security Certificate (part 1) panel just click **Next**.
9. On the Security Certificate (part 2) panel just click **Next**.
10. On the Database Configuration panel, select **DB2 Universal** from the drop-down list and click **Next**.
11. On the Profile Creation Summary panel, click **Create**. Your profile will be created.
12. From the Profiles window, the created custom profile is listed. Click **File** → **Exit** to exit the tool.

3.5.2 Federate the custom profile into the cell

We can federate the node into the cell using the **addNode.sh** command. We need the SOAP_CONNECTOR_ADDRESS of the deployment manager, which can be found in the `portdef.props` file of the deployment manager profile in `<install_dir>/profiles/<dmgr_name>/properties`, where *<dmgr_name>* is the name of your deployment manager profile. Note that since security is turned on we must supply a suitable user name and password:

```
<install_dir>/profiles/monnode1Prof/bin/addNode.sh itsodmgr 39446
-username wpsadmin -password itso4you
```

There are a number of lines of output but, eventually, the following message is displayed.

```
ADMU0003I: Node wps1node1 has been successfully federated.
```

3.6 Other software installations

This section addresses additional software that may need to be installed.

Archived

3.6.1 Installing IBM HTTP Server

To install:

1. Log in to the system where you want to install IBM HTTP Server.
2. Create a file called `ihs.resp` in the `<base_dir>/IHS/IHS` (or use the one provided in Appendix A, “Additional material” on page 421) with the contents shown in Example 3-12 adapted to your location values.
3. Run the installation silently:

```
cd /software/IHS/IHS
./install -options ihs.resp -silent
```

Example 3-12 The IBM HTTP Server response file

```
-OPT silentInstallLicenseAcceptance="true"
-OPT allowNonRootSilentInstall=false
-OPT disableOSPrereqChecking="true"
-OPT installLocation="/opt/IBM/HTTPServer"
-OPT httpPort="80"
-OPT adminPort="8008"
-OPT createAdminAuth="true"
-OPT adminAuthUser="ihsadmin"
-OPT adminAuthPassword="itso4you"
-OPT adminAuthPasswordConfirm="itso4you"
-OPT runSetupAdmin="true"
-OPT createAdminUserGroup=true
-OPT setupAdminUser="ihsadmin"
-OPT setupAdminGroup="ihsadmins"
-OPT installPlugin="true"
-OPT webserverDefinition="httpserver"
-OPT washostname="webserver"
```

3.6.2 Add IBM HTTP Server to the cell

This section explains how to add the IBM HTTP Server to the deployment manager. Make sure that the deployment manager is running before starting this section.

Archived

Using the Integrated Solutions Console

To do this:

1. Launch the Integrated Solutions Console.
2. Add an unmanaged node to the WebSphere cell:
   a. Click **System Administration** → **Nodes**.
   b. Click **Add Node**.
   c. On next panel click **Unmanaged node** and then **Next**.
   d. Enter `IHSNode` as the name and `webserver.itso.ral.ibm.com` as the host name. Select **Linux** from the Platform Type drop-down menu and click **OK,** as in Figure 3-27.

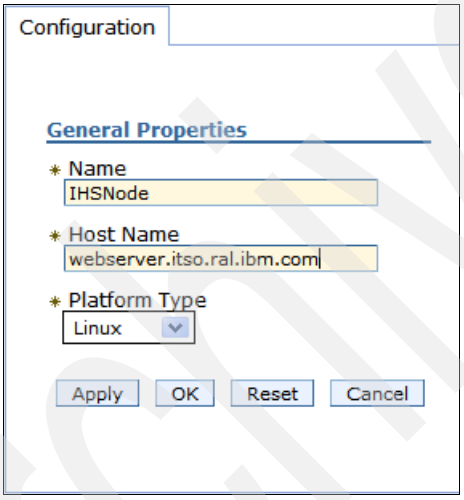


Figure 3-27 New unmanaged node

Archived

3. Add IHS to the cell:
   a. Click **Server** → **Server Types** → **Web servers**.
   b. On top of the empty table click **New**.
   c. In the next dialog choose the newly created node **IHSNode** in the Choose Node selection list, type `httpserver` in the Server Name field, and choose **IBM HTTP Server** in Type selection list, as in Figure 3-28, and click **Next**.

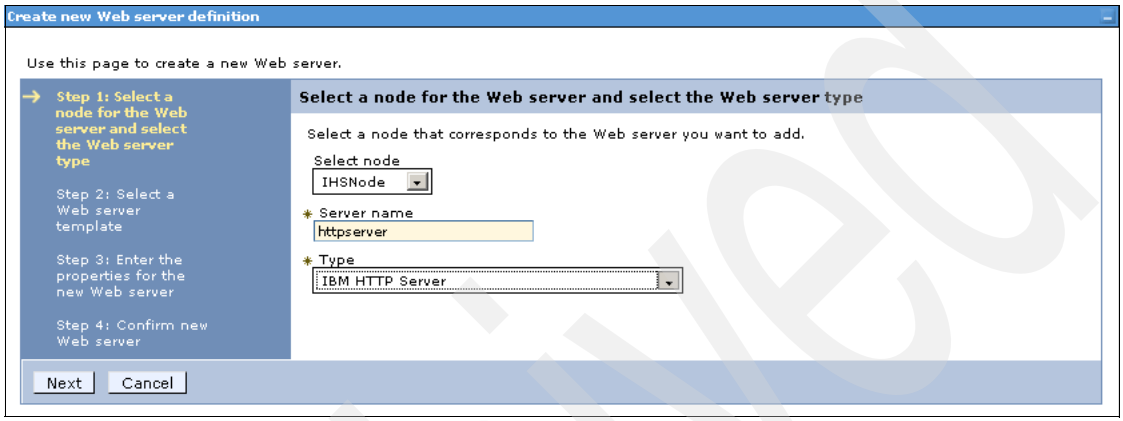


Figure 3-28 Define new http server

   d. Click the radio button next to IHS template on next page and click **Next**.

Archived

e. Enter the following values, as in Figure 3-29, and click **Next**:

- Port: `80`
- Web server installation location: `/opt/IBM/HTTPServer`
- Plug-in installation location: `/opt/IBM/HTTPServer/Plugins`
- Application mapping to the Web server: **All**
- Administration Server Port: `8008`
- Username: `ihsadmin`
- Password and Confirm password: `itso4you`

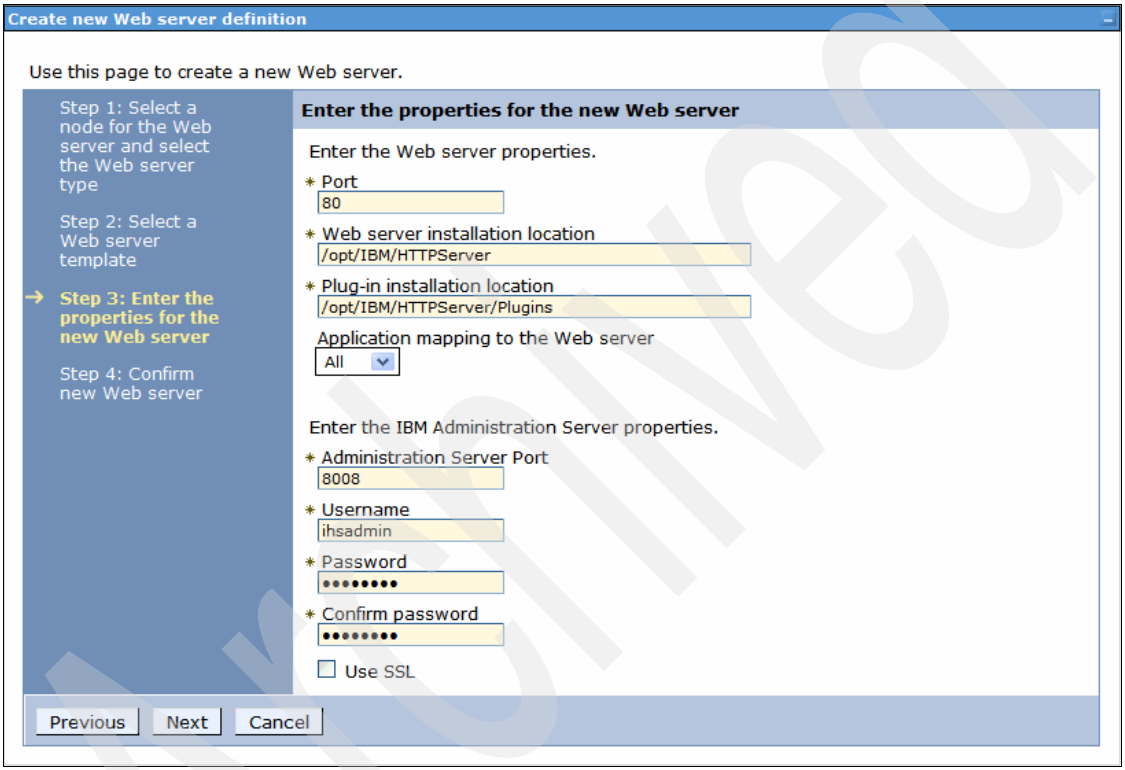


Figure 3-29 IBM HTTP Server details

f. Review details in summary and click **Finish**.

4. Save your configuration.
5. Verify that the IHS Administration server is started by executing the command:

```
/opt/IBM/HTTPServer/bin/adminctl start
```

Archived

6. Go back to the Integrated Solutions Console, click check box next to new IHS server, and click **Start,** as in Figure 3-30.

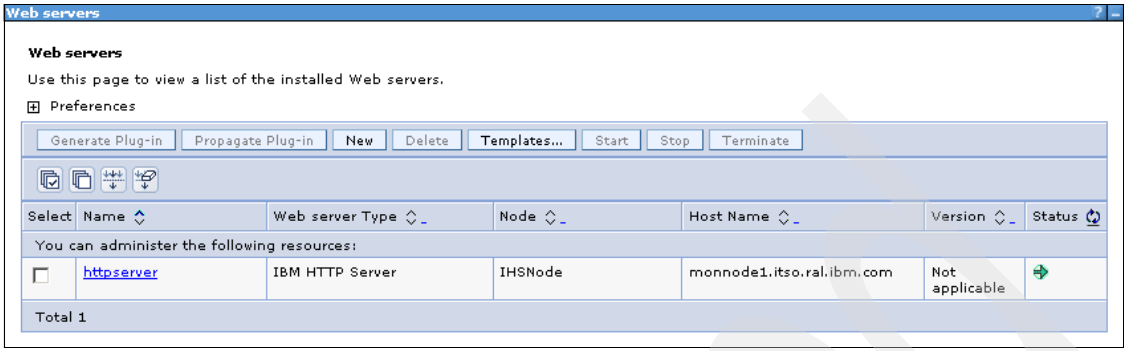


Figure 3-30 List of IHS server and run status

7. Verify that the Status icon changes from a red cross to a green circle.
8. Check the tickbox next to the new Web server and click **Generate Plug-in**.
9. Check the tickbox next to the new Web server and click **Propagate Plug-in** to copy the plug-in XML file to the IHS server.

Using the command line

During the installation of the IBM HTTP Server Plug-in, a *configureWebserver* script is produced by the install process on the Web server machine. In this environment, this file is located in the path:

```
/opt/IBM/HTTPServer/Plugins/bin/configurethttpserver1.sh
```

To configure an IBM HTTP Server called httpserver to the deployment environment, perform the following steps:

1. From the deployment manager, copy this script to the `bin` directory of the deployment manager profile. In our environment, copy this script from the IHS server to the following directory:

   ```
   /opt/ibm/WebSphere/ProcServer/profiles/wps1dmgr/bin
   ```

2. Run the configureWebserver script to configure the Web server:

   ```
   cd /opt/ibm/WebSphere/ProcServer/profiles/wps1dmgr/bin
   ./configurehttpserver1.sh -profileName wps1dmgr -user wpsadmin
   -password itso4you -ihsAdminPassword itso4you
   ```

Note: The configureWebserver script automatically maps all Web application modules to the Web server. Otherwise, you must manually map these Web modules.

On the Web server machine, start the IBM HTTP Server administrative server:

```
/opt/IBM/HTTPServer/bin/adminctl start
```

Verify that the new configured Web server in the deployment environment:

1. Log in to the Integrated Solutions Console as an administrative user.
2. Click **Servers** → **Server Types** → **Web servers**.
3. From the Web servers pane, select the Web server and click **Start**. The arrow should turn green.

Enable SSL on HTTP Server

By default SSL is not enabled on IBM HTTP Server. Enabling of SSL can be done through the Integrated Solutions Console as follows:

1. In the Integrated Solutions Console navigate to **Servers** → **Server Types** → **Web servers** to display the list of configured HTTP servers.
2. Click **httpserver** to display the IBM HTTP server configuration page.
3. Click **Configuration Settings** → **Web Server Virtual Hosts** to display the virtual host configuration page.
4. On the Web server virtual hosts page click **New**.
5. On the first page of the New Web server virtual host wizard click **Security enabled virtual host**, then click **Next**, as shown in Figure 3-31.

Figure 3-31 Create a new security-enabled virtual host

Archived

6. On the second page enter in the Key store file name field enter `httpserver`, in the Target key store directory field enter `${WEB_INSTALL_ROOT}/conf`, in the Key store password and Verify key store password fields type `itso4you`, and in the Certificate alias field enter `selfSigned`, as shown in Figure 3-32. Click **Next**.

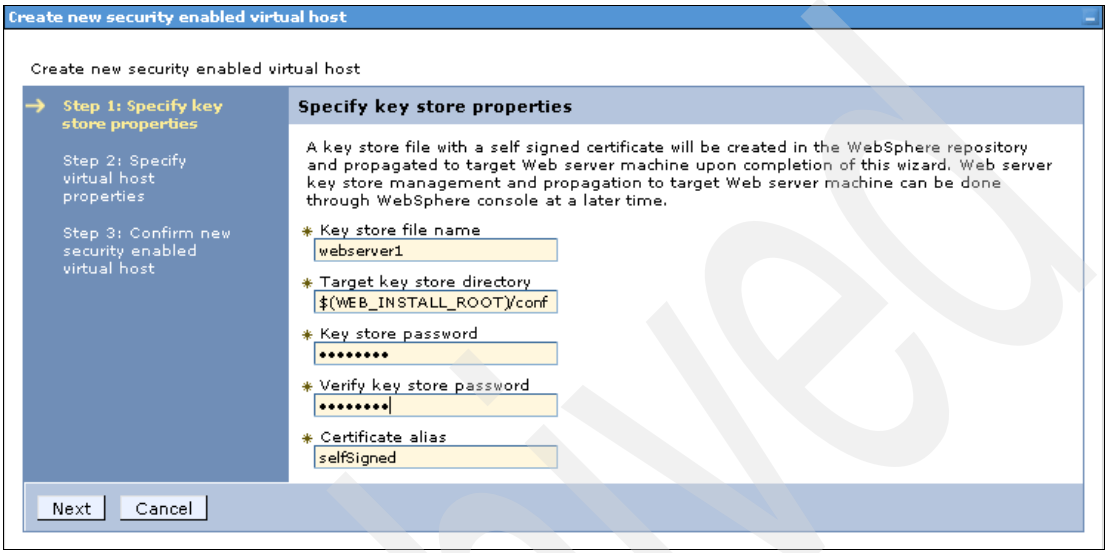


Figure 3-32 Enter SSL keystore properties

7. On the third page enter the IP address of the HTTP server in the IP Address field, leave the default SSL port as 443, and click **Next,** as shown in Figure 3-33.

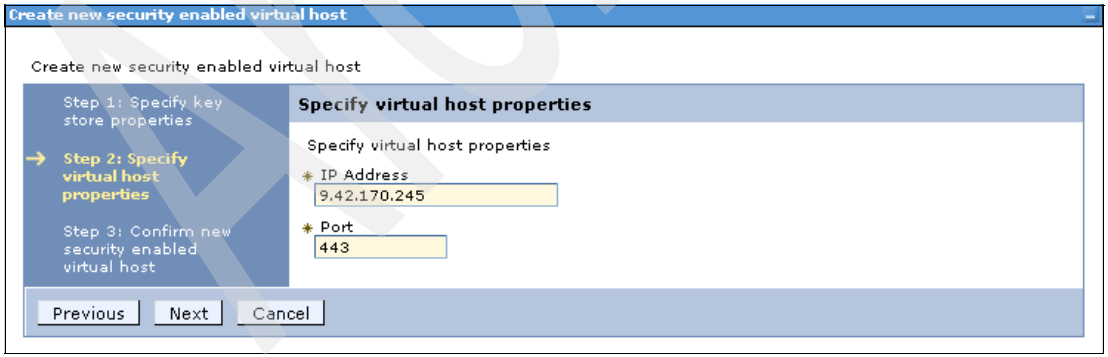


Figure 3-33 Configure HTTP server IP address

8. On the last page review the summary information and click **Finish**.

Archived

Stop and start the HTTP server to activate the secured virtual host by clicking **Stop**, then **Start**.

3.6.3 Installing and configuring WebSphere proxy server

The WebSphere proxy server is used to load balance the Representational State Transfer (REST) API calls between components. Make sure that the deployment manager is running before starting this section.

Note: This section does not cover how to make the proxy servers highly available. High availability of proxy servers could be achieved by using the IBM Edge components or an external IP sprayer appliance. In our test systems we added a dedicated host alias in `/etc/hosts` for the proxy servers with name proxyserver that points to node two. Using a dedicated host alias for the proxy server allows us to add an IP sprayer without changing the configuration afterwards.

Follow these steps:

1. In the Integrated Solution Console, navigate to **Servers** → **Server Types** → **WebSphere proxy server**.
2. In the Proxy Server table, click **New**.
3. On the first page of the wizard select node 2, enter `RESTProxy` as the server name, and click **Next**, as shown in Figure 3-34.

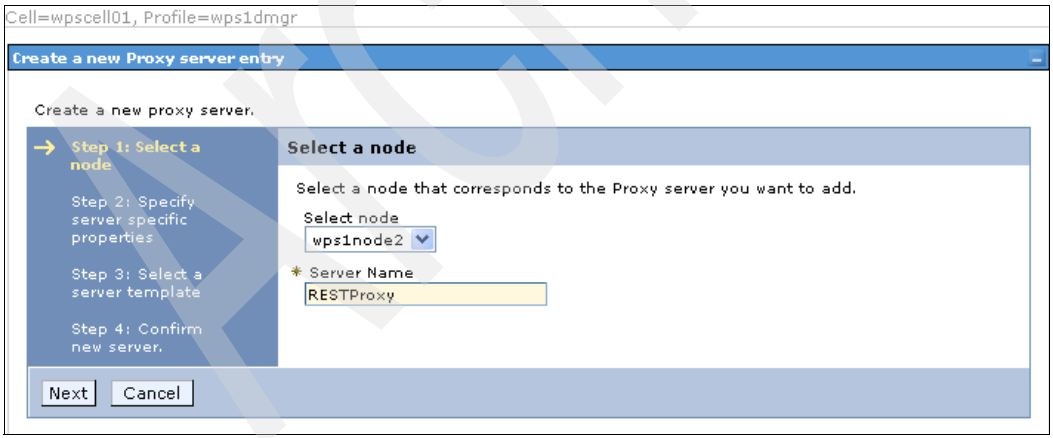


Figure 3-34 Select a node on which to install the proxy

4. On the next panel accept the defaults and click **Next**, as shown in Figure 3-35.

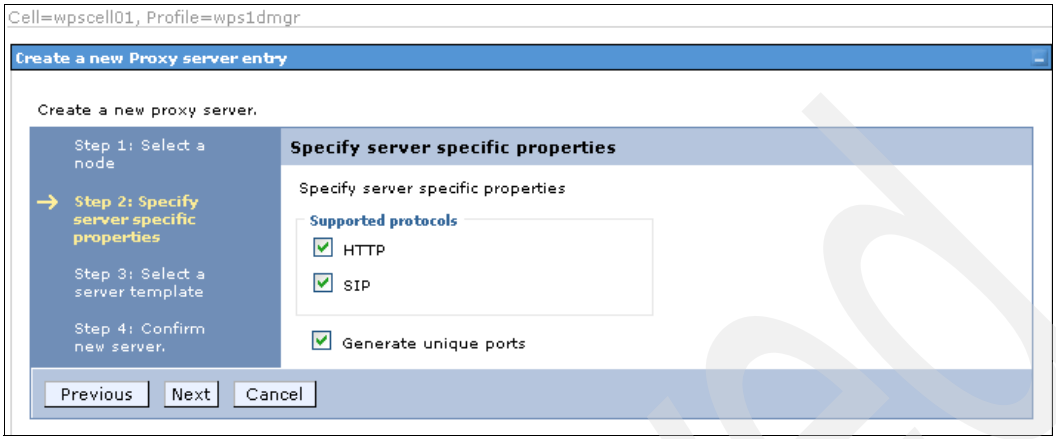


Figure 3-35 Specify the proxy server properties

5. On the next panel accept the defaults and click **Next**, as shown in Figure 3-36.

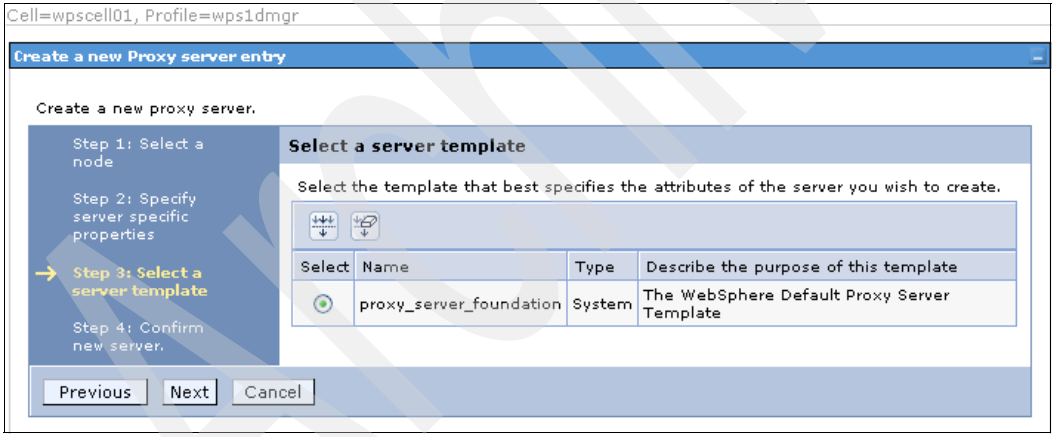


Figure 3-36 Select a proxy server template

6. On the summary panel click **Finish**, as shown in Figure 3-37.

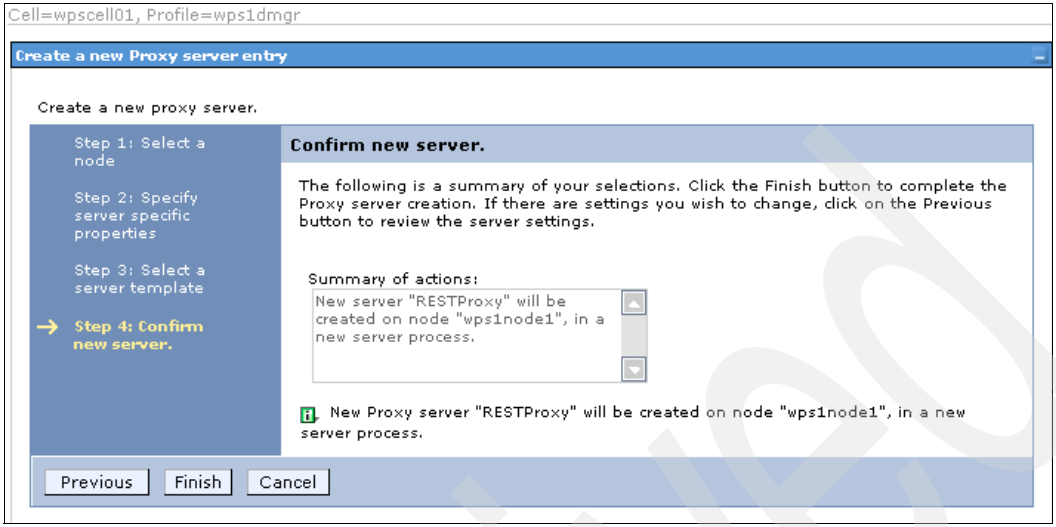


Figure 3-37 The proxy summary panel

7. Save and synchronize your changes.

By default the WebSphere Proxy Server is bound to port 80. If this port is in use (for example, by IBM HTTP Server) we can change the default ports to 8080 for HTTP traffic and 8043 for HTTPS traffic as follows:

1. Click **Server** → **Server Types** → **WebSphere proxy servers** in the Integrated Solutions Console.
2. Click the WebSphere proxy server **RESTProxy** link in the table.
3. Under the Communications section click the **Ports** link.
4. Find the entries for PROXY_HTTP_ADDRESS and PROXY_HTTPS_ADDRESS:
   - For the PROXY_HTTP_ADDRESS change the port to 8080.
   - For the PROXY_HTTPS_ADDRESS change the port to 8043.

   Figure 3-38 shows these changes.

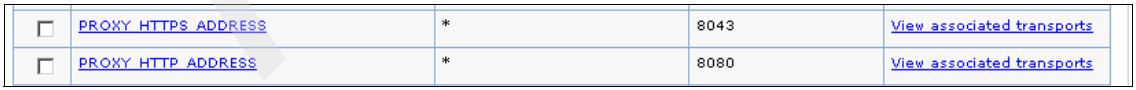


Figure 3-38 HTTPS and HTTP port settings

5. Save the changes and synchronize your changes with the nodes.

Archived

Finally, we can start the proxy server itself.

1. Navigate to **Servers** → **Server Types** → **WebSphere proxy servers**.
2. Select the check box in front of the RESTProxy cluster and click **Start**. Wait until the proxy server has started and the red cross has turned green.

Archived

Archived

4

WebSphere Process Server: single cluster

This chapter provides full instructions for creating a Single Cluster topology for WebSphere Process Server V7.0. In this topology, all of the functional components (user applications, messaging infrastructure, Common Event Infrastructure (CEI), and support applications) run in the same cluster.

Archived

© Copyright IBM Corp. 2010. All rights reserved.

4.1 Topology summary

Figure 4-1 shows a diagram of the topology we will be creating in this chapter. In this topology, all of the functional components (user applications, messaging infrastructure, CEI, and support applications) run in the same cluster, and so there is only one cluster member on each node. The messaging engines run on one cluster member at a time. The passive messaging engines are shown shaded.

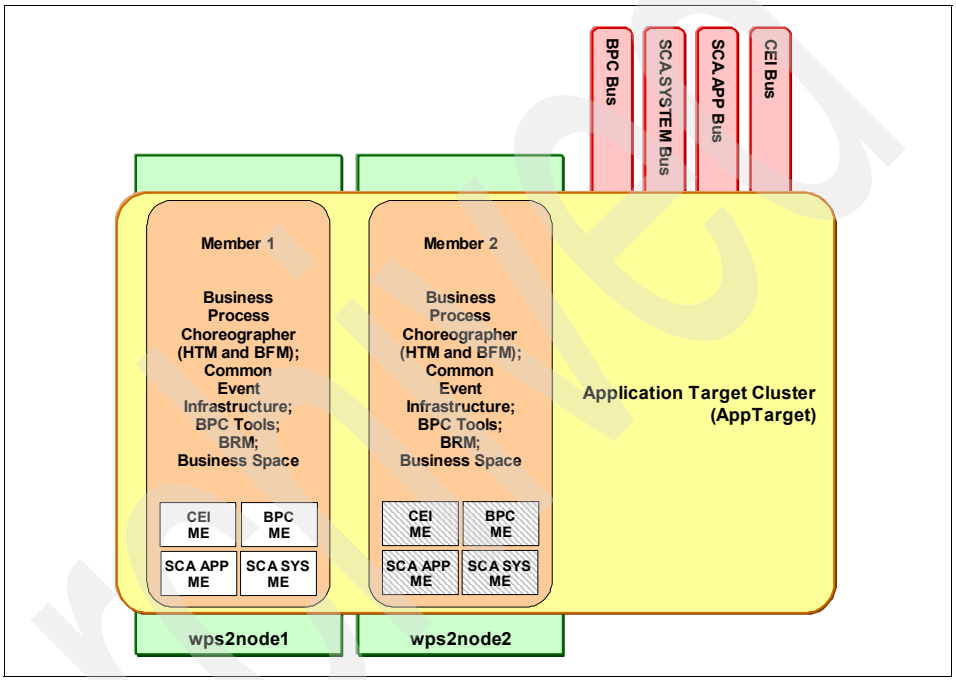


Figure 4-1 Single Cluster topology

4.2 Prerequisites for creating this topology

Before you start creating this topology, you must have completed various prerequisite steps described in Chapter 3, “Preparing your topology” on page 55. These are:

- Install DB2 (or an alternative database) as per “Software versions used in this book” on page 56.
- Install Tivoli Directory Server (or an alternative user registry) as per “Software versions used in this book” on page 56.

- Define the appropriate users in your user registry as per 3.4.2, “Configuring LDAP as the user account registry” on page 99.
- Install WebSphere Process Server as per 3.2, “Software installation” on page 57.
- Install IBM HTTP Server (or an alternative HTTP server) as per 3.6.1, “Installing IBM HTTP Server” on page 111.
- Create the required databases in DB2 as per 3.3, “Database and schema creation” on page 82, which are:
  - Common database (WPRCSDB)
  - Business Process Choreographer database (BPEDB)
  - Business process reporting schema (OBSVRDB)
  - Messaging engine database (MEDB)
  - Event database (EVENT)

  Note: Tables for the event database are not created now, but after the deployment environment has been created, so do not execute the steps from 3.3.7, “The EVENT database” on page 89, yet.

- Create a Deployment Manager profile, set up the user registry integration, add IBM HTTP Server to the cell, and modify certain configuration settings, as per 3.4, “Creating a deployment manager profile” on page 91. Note that you do not need to install WebSphere Proxy Server for this topology (as shown in 3.6.3, “Installing and configuring WebSphere proxy server” on page 118).
- Create and federate two custom nodes as per 3.5, “Creating custom profiles” on page 107.

4.3 Configuring topology using deployment environments in Integrated Solutions Console

This section provides instructions to configure and deploy a Single Cluster topology. The topology includes a database server (DB2), an LDAP server, and two nodes to provide the clustering required. Once you have completed these instructions, you must complete the steps in 4.5, “Post-creation configuration and verification” on page 137.

Creating a deployment environment

The following procedure creates a Single Cluster topology using the Integrated Solutions Console (ISC). Before you begin, ensure that the deployment manager and nodes are running and that their logs are error free.

You will first define, then generate, a Single Cluster topology. These are logically separate steps. You can define a topology using the Integrated Solutions Console, then generate it later if you want. In this section we do both. Topology definitions can be exported from the administrative console. In 4.4, “Configuring topology manually using command line” on page 136, we discuss how this can be used to silently generate the deployment environment instead.

1. Log in to the Integrated Solutions Console as the primary administrative user that you defined when you created the profile. Here we use `wpsadmin`.
2. Navigate to **Servers Deployment Environments**. Click **New**. The system displays the first page of the Deployment Environment Configuration wizard with **Create new deployment environment** selected. Enter the name of your deployment environment (we use `WPSSingleCluster`), as shown in Figure 4-2, then click **Next**.

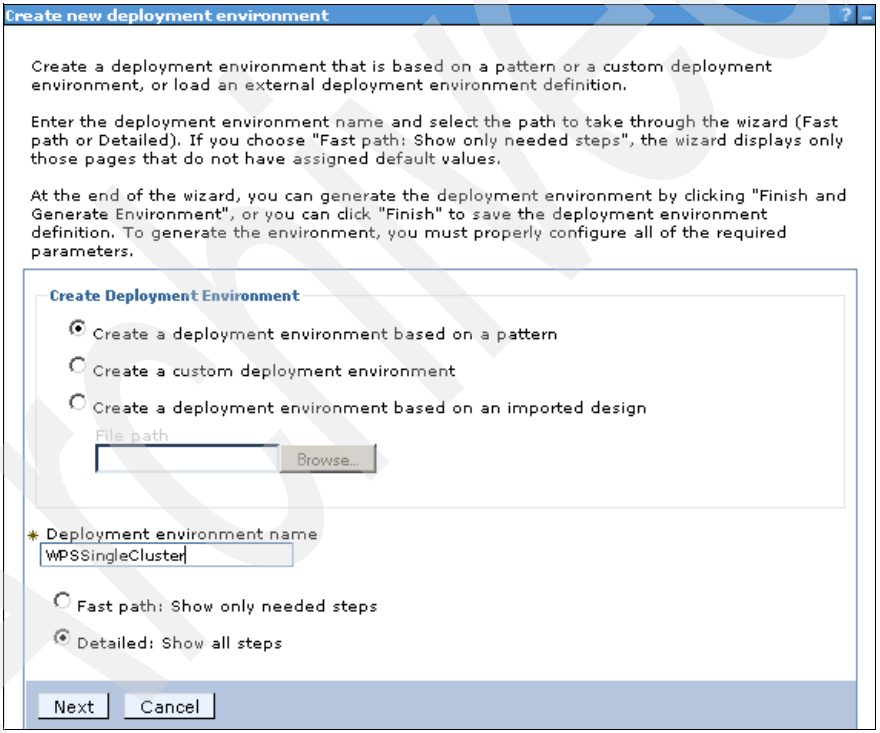


Figure 4-2 Create new deployment environment window: name selection

Archive

3. Ensure that **WPS** is selected as the deployment environment type on the Create new deployment environment wizard, as shown in Figure 4-3, then click **Next**.

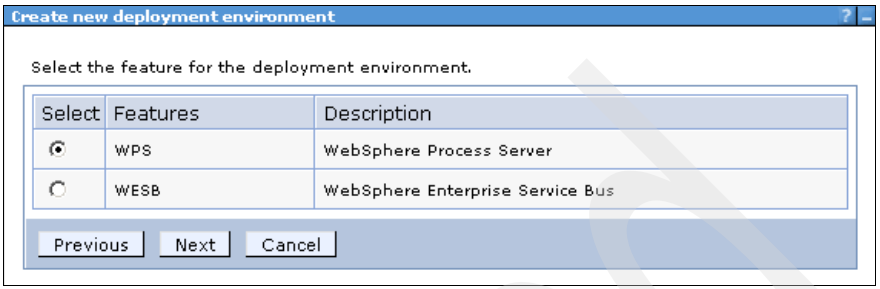


Figure 4-3 Create new deployment environment window: type selection

4. Change the deployment environment pattern to **Single Cluster** (as shown in Figure 4-4), as this is the type of deployment environment that we create in this chapter, and click **Next**.

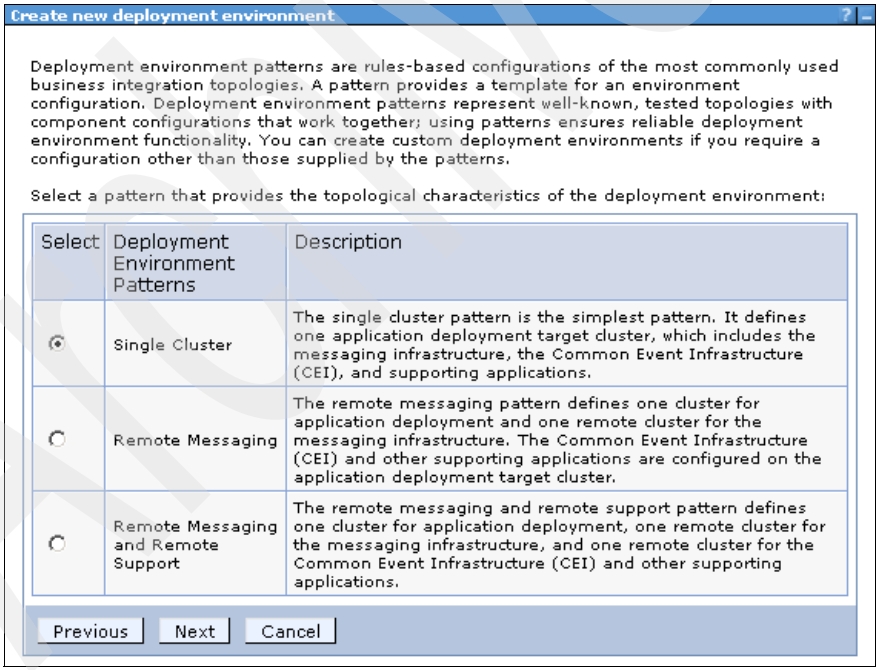


Figure 4-4 Deployment environment pattern type window

5. You now run through a series of steps in the wizard, specifying how the deployment environment should be created. For step 1, you must specify

which nodes are to be used to create the environment. Select both **wpsnode1** and **wpsnode2**, as these are the nodes that you created earlier, as shown in Figure 4-5. The single cluster spans the nodes selected.

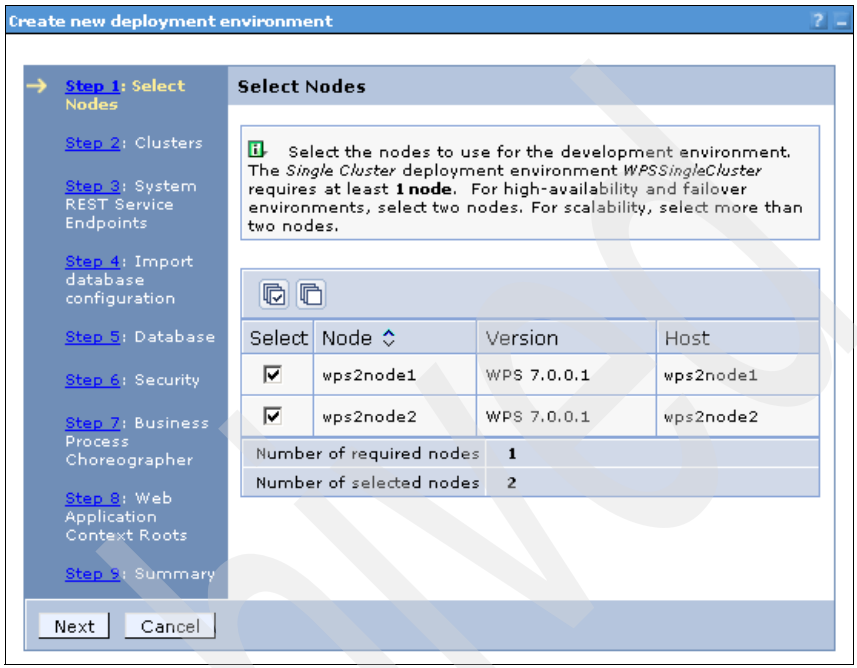


Figure 4-5 Step 1 of the Create new deployment environment wizard

Archive

6. On Step 2 of the wizard, you can select how many cluster members to create on each node for each cluster that will be created. In this case, since we are creating a Single Cluster topology, only one cluster (named Application Deployment Target) is listed. You can leave the values at the default to specify one cluster member per node (as shown in Figure 4-6).

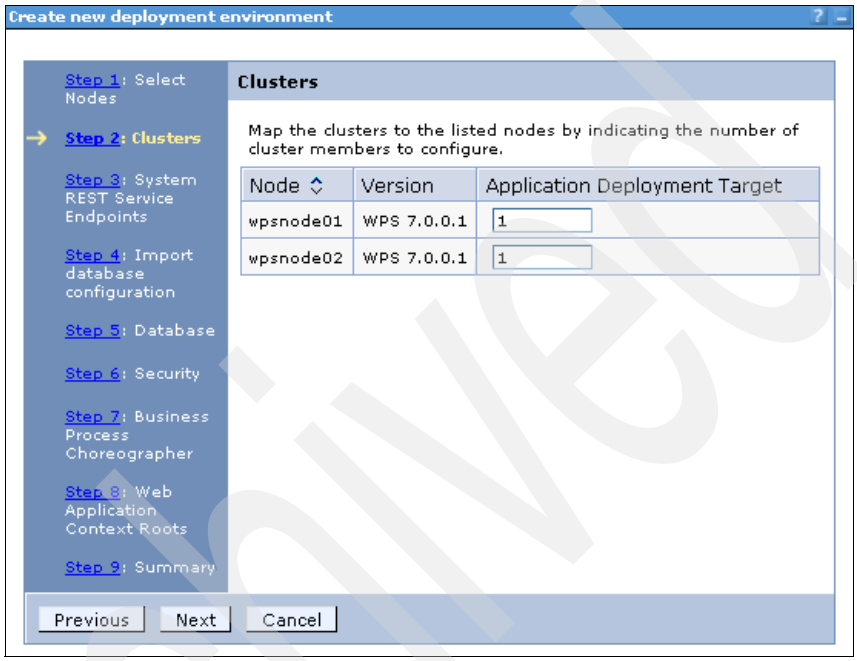


Figure 4-6 Step 2 of the Create new deployment environment wizard: clusters

Archived

7. For step 3 of the wizard, enter the details for your REST endpoints. Because we are developing a production topology, we installed IBM HTTP Server to provide load-balancing across our cluster. On this page, we enter the HTTP Server hostname and port where this is installed (Figure 4-7).

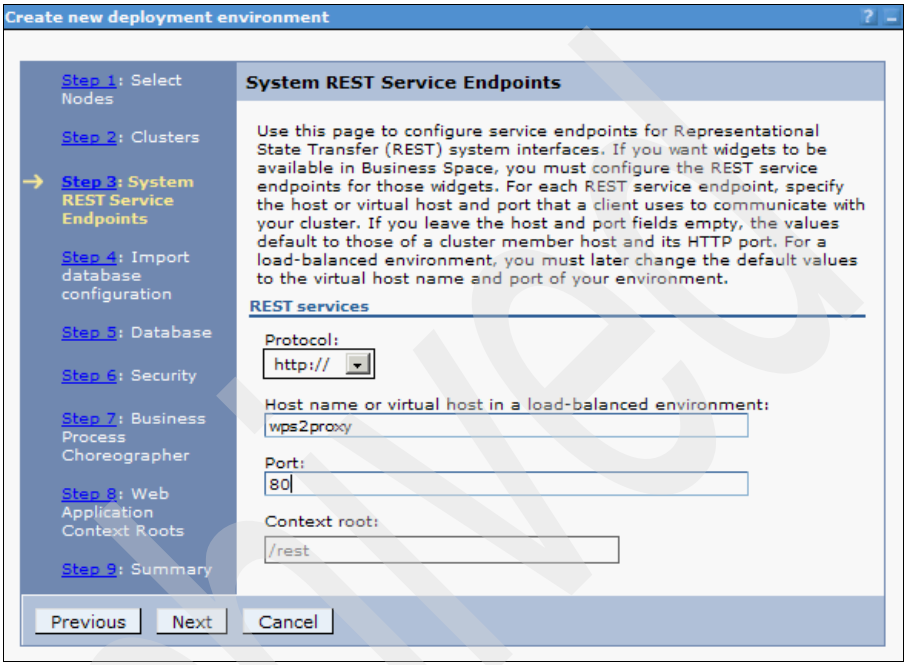


Figure 4-7 Step 3 of Create new deployment environment wizard: REST endpoints

Archived

8. For Step 4 of the wizard, select the database design file that was output by the database design tool that you used in 3.3, “Database and schema creation” on page 82. This automatically populates the database parameters during the next step of the wizard, saving you time. See Figure 4-8 for an example.

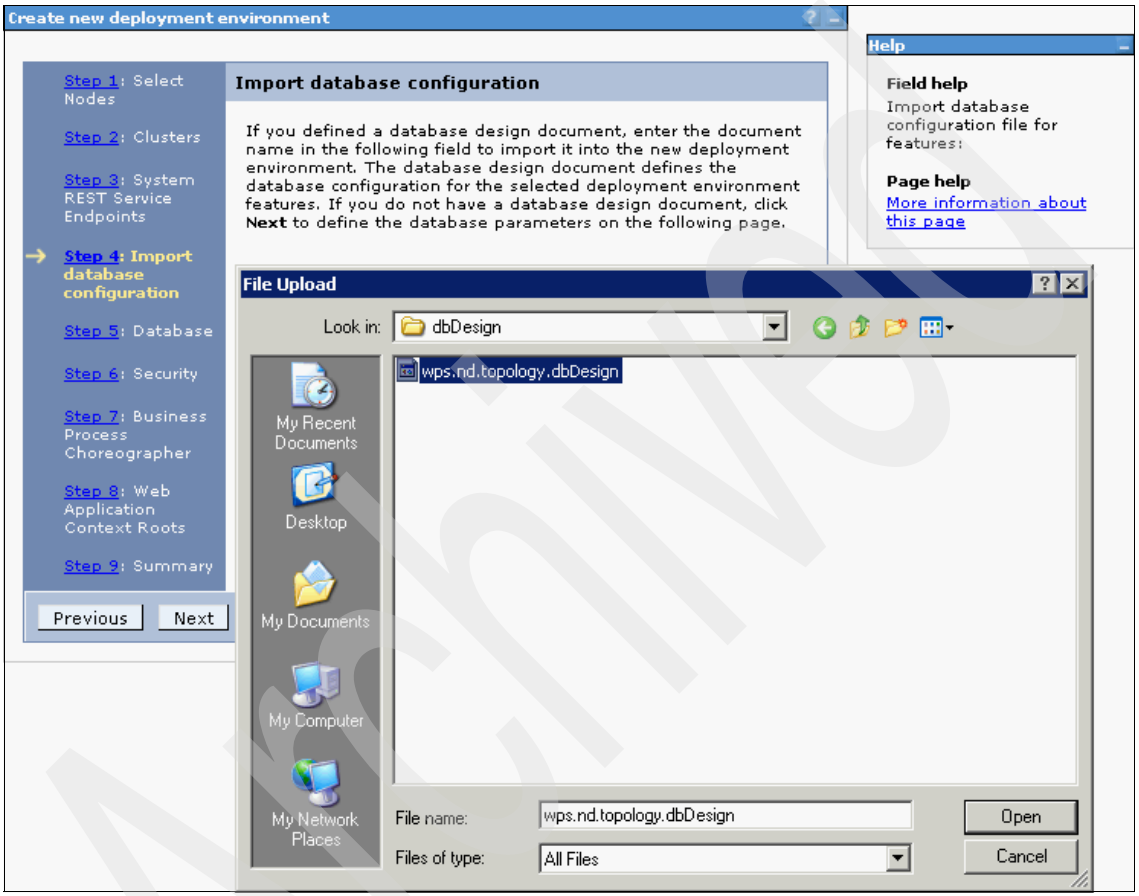


Figure 4-8 Step 4 of the Create new deployment environment wizard: dbDesign selection

Archive

9. For Step 5 of the wizard, you may want to review the database settings that have been selected, as shown in Figure 4-9. For each line double-check that the Create Tables option is *not* selected, as you have already created the database tables. However, you should not have to change anything else, as the database names, schema names, and so on, should be consistent with the earlier selections in the database design tool described in Chapter 3, “Preparing your topology” on page 55.

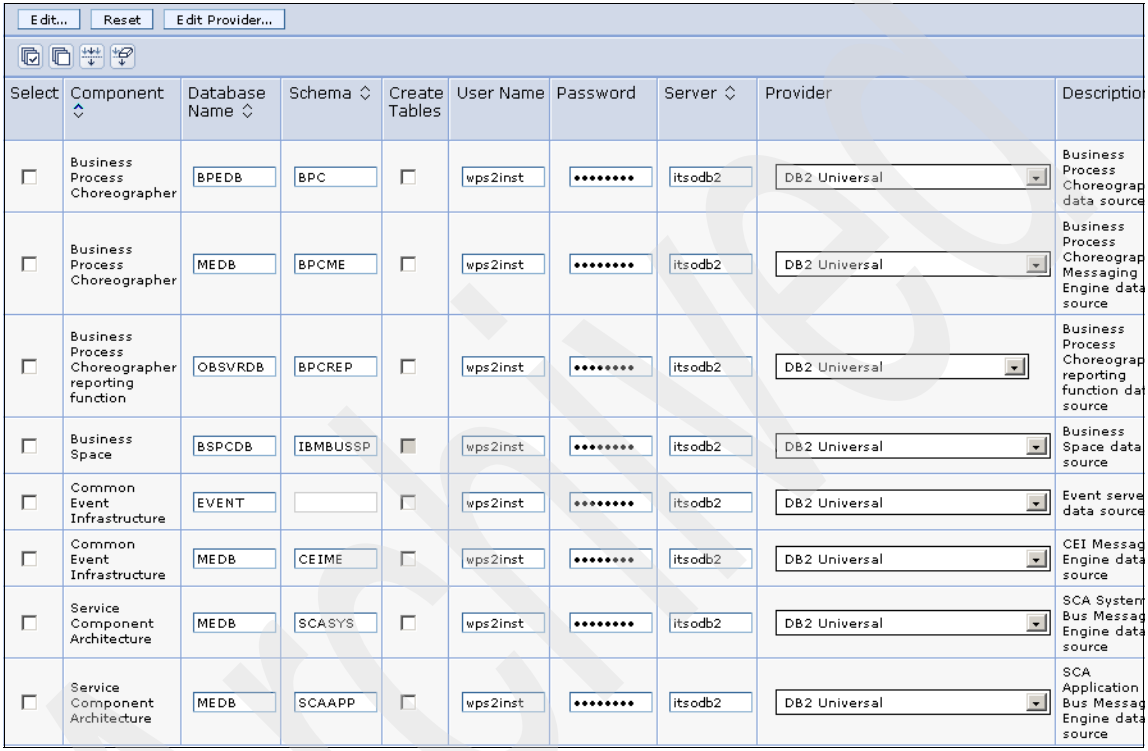


Figure 4-9 Step 5 of the Create new deployment environment wizard: database parameters

10. On step 6 of the wizard, set the username and passwords to values that you have defined in user registry, as shown in Figure 4-10.

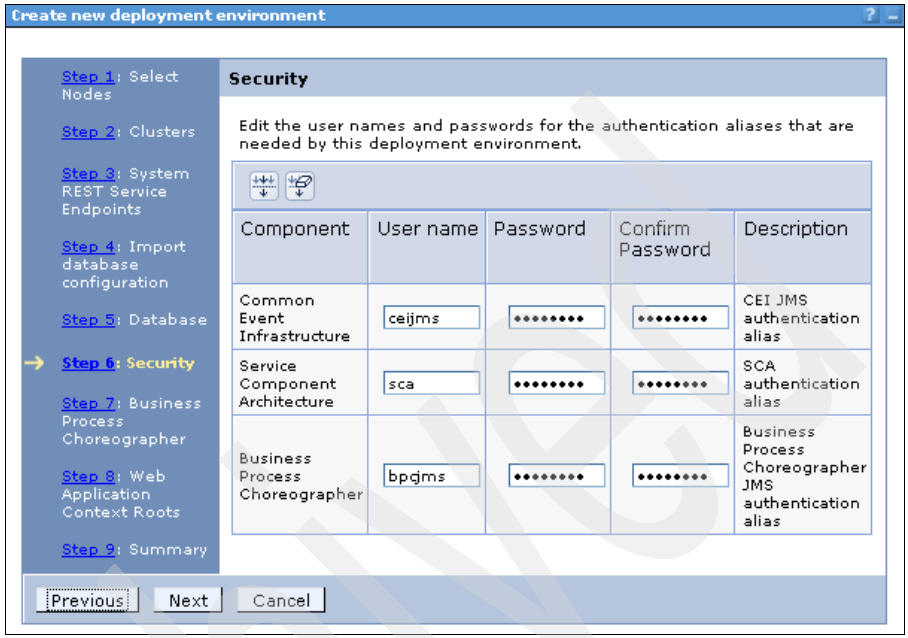


Figure 4-10 Step 6 of Create new deployment environment wizard: SIBus security roles

Archive

11. For step 7 of the wizard, set the usernames and groups (with applicable passwords as defined in the user registry) as shown in Figure 4-11. Here we have not defined an e-mail service for the human task manager. If you need one for your topology, define it now.

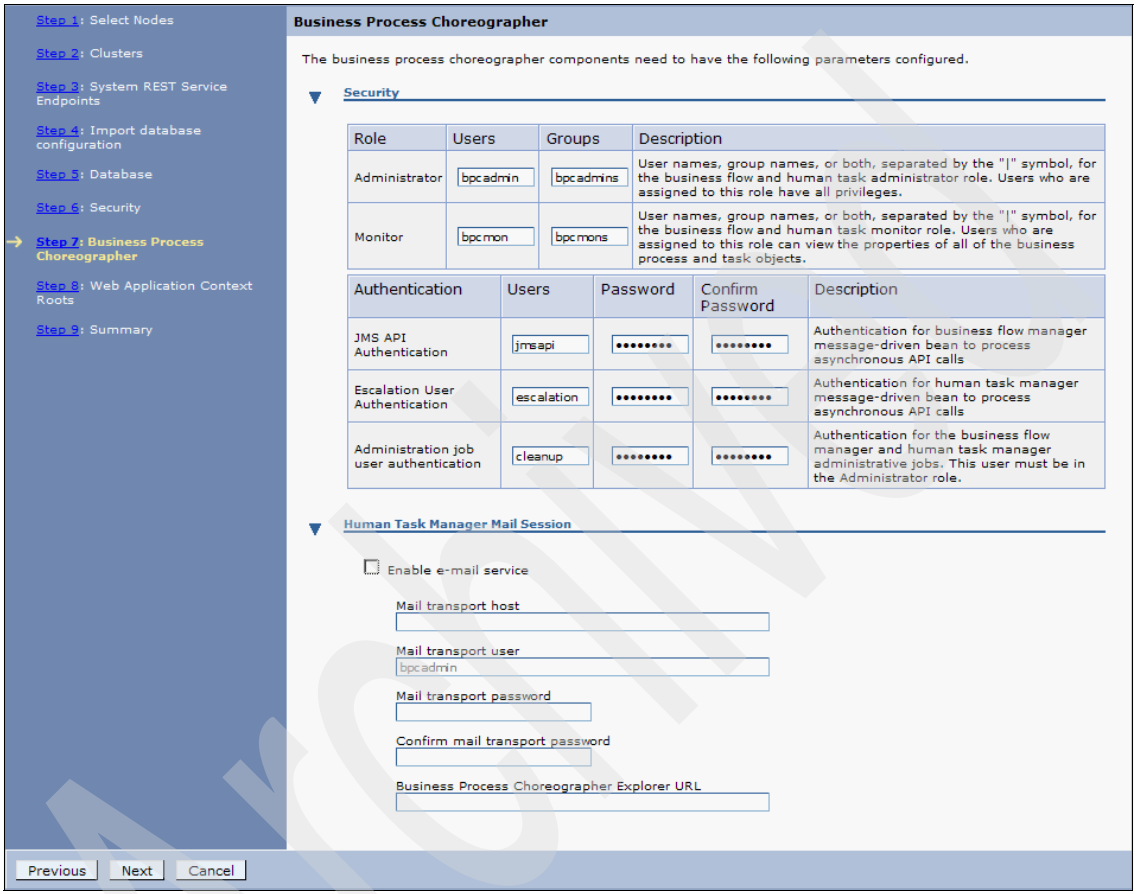


Figure 4-11 Step 7 of the Create new deployment environment wizard: security settings

12. On step 8 of the wizard, you can modify the context roots for various WebSphere Process Server components. In the environment that we have created for this book, we used the default values (Figure 4-12).

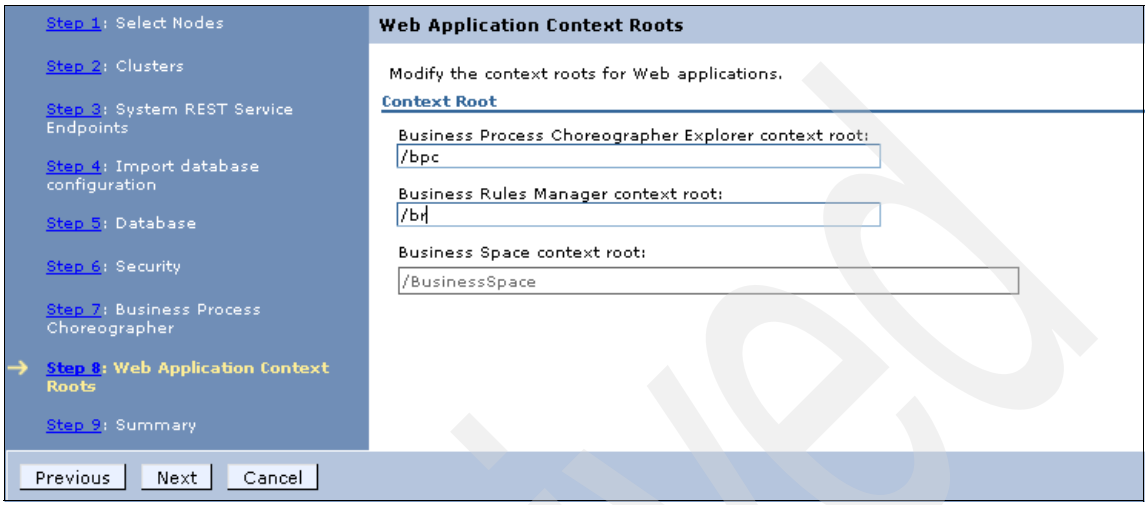


Figure 4-12 Step 8 of the Create new deployment environment wizard: context root

13. For step 9 of the wizard, you can review a summary of the entire topology before it is created. Select **Finish and Generate Environment**.

> **Note:** If you want, you can select **Finish** here instead to save the deployment environment definition without generating it, giving you a chance to review and edit it first, or export a topology definition file for use in command-line generation (see 4.4, “Configuring topology manually using command line” on page 136, for more information about this).

14. Messages appear as the environment is generated. Once finished, click **Save changes**. If any errors appear, review and correct them before you continue.

You have now completed definition and generation of your deployment environment.

> **Warning:** Do not restart your deployment manager yet. You must ensure that the generation of the deployment environment is complete first. Carry out the steps in 4.5.1, “Restarting the deployment manager” on page 137, to ensure that this is done successfully.

Archived

4.4 Configuring topology manually using command line

In order to configure a topology using the command line, you will need a topology definition file. The typical way to create this is to define a topology using an Integrated Solutions Console and then export it (this is typically *not* done on the WebSphere cell on which you want to generate the topology). To do this:

1. Define a topology as in 4.3, “Configuring topology using deployment environments in Integrated Solutions Console” on page 125, for example. Though typically you would only save the environment definition by selecting **Finish** at the end. You would not actually generate it.
2. Export the topology by opening the Integrated Solutions Console, navigating to **Servers** → **Deployment Environments**, selecting the relevant topology definition, and clicking **Export**.

> **Note:** To provide an alternative approach if you do not have access to an administrative console, we supply a template topology definition file that you can fill out. See Appendix C, “Deployment environment definition files” on page 439, for more information and certain caveats about using this approach.
>
> You can also generate a new topology definition from scratch using **wsadmin** commands. We do not cover that approach in this book. You can find more information in the InfoCenter here:
>
> http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp?topic=/com.ibm.websphere.wps.doc/doc/tcnfg_create_depenv_cmd.html

Whichever of these methods you choose, you can then generate the topology using two **wsadmin** commands, as shown in “Importing and generating a deployment environment topology” on page 136—one to import the topology definition file and another to generate it.

Once you have completed these instructions, you must complete the steps in 4.5, “Post-creation configuration and verification” on page 137.

Importing and generating a deployment environment topology

Once you have a file representing a deployment environment topology, you can import it and generate the environment using two **wsadmin** commands. Here we show how to do this interactively. You could also build this into a wsadmin script, with appropriate error checking.

First, you would import the deployment environment definition:

```
AdminTask.importDeploymentEnvDef ([ '-filePath', fileLocation,
'-topologyName', topologyName])
```

Assuming that this completes successfully, you must ensure that this configuration change is saved with `AdminConfig.save()`.

Then generate the deployment environment itself:

```
AdminTask.generateDeploymentEnv (['-topologyName', topologyName])
```

Again, assuming that this completes successfully, save the change with `AdminConfig.save()`. Do not save the changes if they do not complete successfully, or you will most likely save a broken deployment environment.

Warning: Do not restart your deployment manager yet. Ensure that the generation of the deployment environment is complete first. Carry out the steps in 4.5.1, “Restarting the deployment manager” on page 137, to ensure that this is done successfully.

4.5 Post-creation configuration and verification

These steps must be followed after creating a deployment environment via the Integrated Solutions Console (4.3, “Configuring topology using deployment environments in Integrated Solutions Console” on page 125) or manually using the command line (4.4, “Configuring topology manually using command line” on page 136).

4.5.1 Restarting the deployment manager

Restart the deployment manager. First, however, it is *very important* to ensure that the deployment environment generation has finished successfully or you may encounter problems. This is an asynchronous process, so a successful response from the Integrated Solutions Console or wsadmin stating that the deployment environment has been created is *not* sufficient.

To do this, monitor the logs on each of your node agents (*not* the deployment manager). Toward the end of the deployment environment being created, you will see messages of the form:

```
[24/02/10 13:51:17:234 GMT] 00000050 AppBinaryProc I   ADMA7021I:
Distribution of application
HTM_PredefinedTaskMsg_V700_WPSSingleCluster.AppTarget completed
successfully.
```

There will be many of these messages in the logs of both node agents as the standard applications are deployed to both nodes in your deployment environment. Wait for these messages to stop appearing and for the logs to quiesce, as this tells you that the deployment environment generation is complete. Then restart your deployment manager.

4.5.2 Creating event database tables

Create and populate the event database tables. Refer to 3.3.7, “The EVENT database” on page 89, for information about how to do this.

4.5.3 Checking database connectivity

Before you start the environment check database connectivity to ensure that the deployment environment has been generated correctly:

1. Log in to the administrative console and navigate to **Resources JDBC JDBC Providers** → **Data Sources**.
2. Make sure that **All scopes** is selected.
3. Select all of the data sources and click **Test connection** to ensure that they have connectivity. You will see a message that the connection was successful for each.

Archived

Command-line equivalent

You can use a script such as that shown in Example 4-1 to check database connectivity using wsadmin instead. Change the cell name at the top of the script to the cell name that you are using.

Example 4-1 Checking database connectivity using wsadmin

```
cellName = "wpscell02"

dataSources = AdminConfig.list('DataSource', AdminConfig.getid(
'/Cell:' + cellName + '/')).splitlines()

for dataSource in dataSources:
   if not dataSource.startswith('DefaultEJB'):
      print dataSource
      AdminControl.testConnection(dataSource)
```

Execute the script above with a command like:

```
./wsadmin.sh -lang jython -user wpsadmin -password itso4you -f
script.py
```

You will see output from the script that looks like:

```
WBI_DataSource(cells/wpscell01|resources.xml#DataSource_1266883043700)
event(cells/wpscell01/clusters/WPSSingleCluster.AppTarget|resources.xml
#DataSource_1267018736416)
WASX7467I: Connection to provided datasource on node wps1node1 process
nodeagent was successful.
```

This output should be repeated for each data source.

4.5.4 Change the virtual host hostname

To improve the security of your configuration, you can restrict the virtual host hostname used for port 9080 (the default Web port) for your Single Cluster so that it only accepts incoming requests from the Web server. To do this:

1. Open the Integrated Solutions Console and log in.
2. Navigate to **Environment** → **Virtual hosts**.
3. Click **default_host**.
4. Click **host aliases**.

Archived

5. Add and delete entries as appropriate so that port 9080 is only associated with the hostname for the Web server (wps2proxy in our case). There may be entries associated with the hostname *. These match any host, so remove them.
6. Save your changes.

4.5.5 Correct REST endpoints

At the time of writing, the endpoints for our BPEContainer and TaskContainer Representational State Transfer (REST) services were incorrectly configured. However, if you try to change them using the Integrated Solutions Console, the changes do not persist. You can use the wsadmin script supplied with the book called modify_restURLs.jython to make this change instead. Just run it against your Deployment Manager. Modify the three variables at the top of the script to match your cell name and Web server URL.

4.5.6 Configuring CEI event persistence

For a production system, typically it is best to disable persistence of CEI events to a database, as this can have a significant performance overhead. To do this:

1. Open the Integrated Solutions Console and navigate to **Service Integration → Common Event Infrastructure → Event Service → Event Services**.

   Note: If this menu item does not appear, you may not have restarted your deployment manager since generating your deployment topology, in which case, restart it now.

2. Click **Default Common Event Infrastructure event server**. Ensure that the Enable event data source option is not selected.
3. Click **OK**. Save and synchronize your changes.

Archived

Command-line equivalent

Alternatively, you may use the **wsadmin** commands shown in Example 4-2 with wsadmin to accomplish the same steps. Make sure to replace the cell name at the top of the script with the cell name that you are using.

Example 4-2 wsadmin script to set CEI event persistence

```
cellName = "wpscell02"

eventServerProfile = AdminConfig.list('EventServerProfile', AdminConfig.getid(
'/Cell:' + cellName + '/')).split("\n")[0]
AdminConfig.modify(eventServerProfile, '[[enableDataStore false]]')
AdminConfig.save()
```

Execute the script above with a command like:

```
./wsadmin.sh -lang jython -user wpsadmin -password itso4you -f
script.py
```

4.5.7 Configuring CEI event logging

You may want to enable CEI event logging to the BPC Observer database (also called the BPC Event Collector or BPC Reporting database) if this functionality is a production requirement:

1. In the Integrated Solutions Console, navigate to **Servers Clusters** → **WebSphere application server clusters**. Click your cluster (in this chapter, this is called **WPSSingleCluster.AppTarget**).
2. On the right-hand side, under Business Integration, expand the **Business Process Choreographer** section, then click **Business Process Choreographer Containers**.
3. Scroll down the page and expand the **State Observers** section. Click **Common Event Infrastructure Logging** for either the Business Flow Manager check box or the Human Task Manager check box, or both, depending on your requirements (whether you want these types of CEI events to be logged to the BPC Observer database).
4. Click **OK**. Save and synchronize your changes.

wsadmin equivalent

There is a jython script supplied with this book called cei-event-logging.jython that does the equivalent of the steps above. You must modify the variables at the top of the script, then run it against your deployment manager.

Archived

4.5.8 Verification: human task manager

The following steps verify that the Human Task Manager component is working correctly.

1. Start your cluster if it is not yet started.
2. Open the Business Process Choreographer (BPC) Explorer on the Web server for your topology (that is, `http://yourwebserver/bpc`). You may need to log in. Use the username and password that you configured for this (in our case, `bpcadmin/itso4you`).
3. Navigate to **Task Templates** → **My Task Templates**. You will see preconfigured sample human tasks. Click the box next to **Widget Message**, and click **Create Instance** (Figure 4-13).

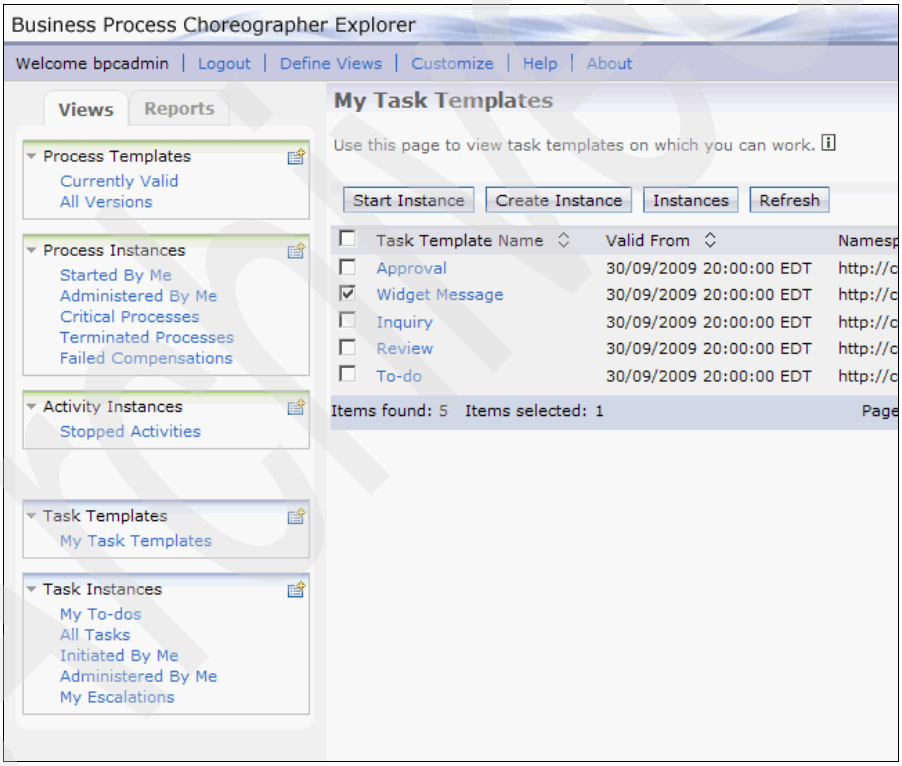


Figure 4-13 BPC Explorer: creating a widget message, step 1

4. Enter values for the subject and message fields of the human task, as shown in Figure 4-14, then click **Submit**.

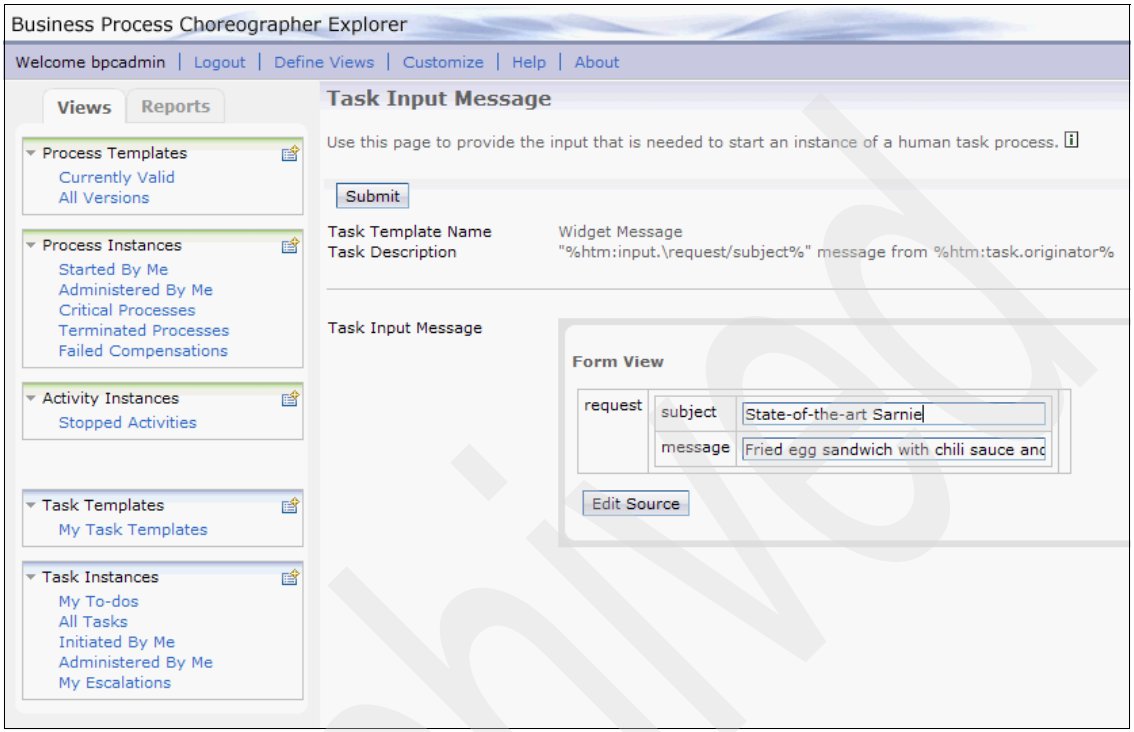


Figure 4-14 BPC Explorer: creating a Widget Message, step 2

5. Navigate to **Task Instances** → **Initiated By Me**. You should see the one instance of the task that you just created. Click the task name itself, **Widget Message**.

Archived

6. Select the **Custom Properties** tab and click the empty space next to the htm_assigned_to field (this is an editable property, although it does not appear so at first). Enter a valid user to own the task, as shown in Figure 4-15 (`bpcadmin` is sensible here). Click **Save**, then **Start**, then **Submit**. The task now has an owner.

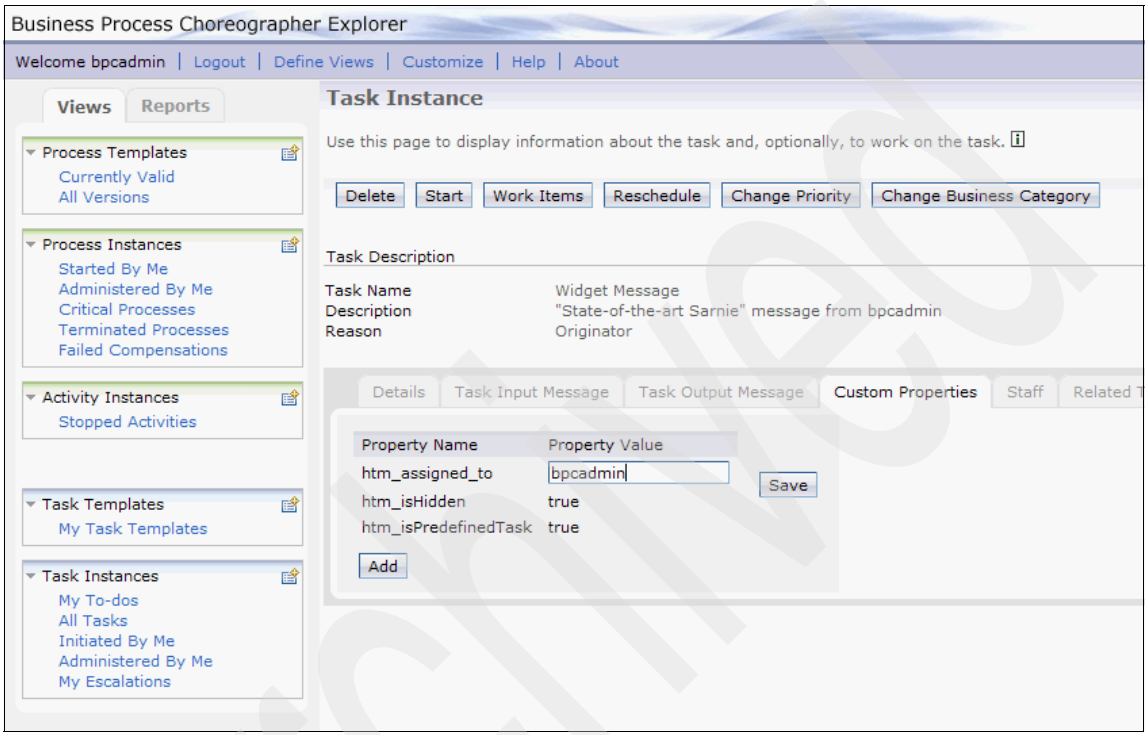


Figure 4-15 BPC Explorer: creating a widget message, step 3

Archive

7. Click the check box next to the widget message task and click **Work on**. In the response comments field, enter text, as shown in Figure 4-16, then click **Complete**.

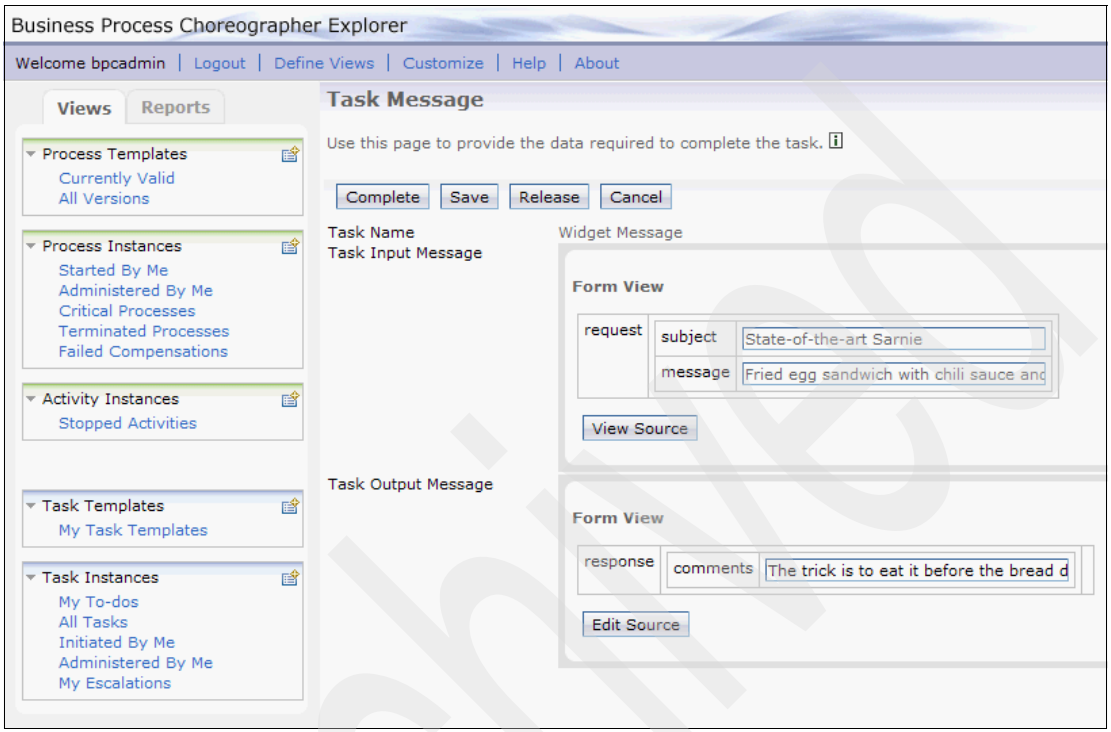


Figure 4-16 BPC Explorer: creating a widget message, step 4

8. The human task is now finished and you have verified that the human task manager is working correctly. If you want to delete it, click **Task Instances** → **Administered By Me**, select the task, and select **Delete**.

4.5.9 Verification: business process engine

To verify that the Business Process Choreographer is working correctly, execute the Hello World sample on the official Business Process Management Samples and Tutorials site at:

`http://publib.boulder.ibm.com/bpcsamp/`

Navigate to **Getting started** → **Hello World** for instructions. Under the **Build it yourself** link you can find instructions about how to build the application and deploy it to WebSphere Process Server. To save you time, and in case you do not have access to WebSphere Integration Developer, we have pre-built the

application and supplied it with this book as `HelloWorld.ear`, so you can skip the actual building of the application (although you still must install it). Under the link **Run the Sample**, you can find instructions about how to execute the sample using the BPC Explorer. This verifies that the Business Process Choreographer is working correctly.

4.5.10 Verification: Business Space

In this section we log into Business Space and navigate to the system health widget to ensure that the Business Space infrastructure is working correctly.

Note: These instructions assume that you have set up your Web server and that it is configured to accept HTTPS connections. We do not include instructions on that in this book. For more information about the IBM HTTP Server see *IBM HTTP Server (powered by Apache): An Integrated Solution for IBM eServer iSeries Servers*, SG24-6716.

1. Open a Web browser, enter the Business Space URL `http://`*`yourwebserver`*`/BusinessSpace`, and log in using wpsadmin.
2. Click **Go to Spaces**, then click **Problem Determination**.

Archived

3. You will see a page similar to that shown in Figure 4-17, showing the various deployment environments, clusters, and node agents that you have, and that they are running. Explore the other tabs and check other components in your environment if you want.

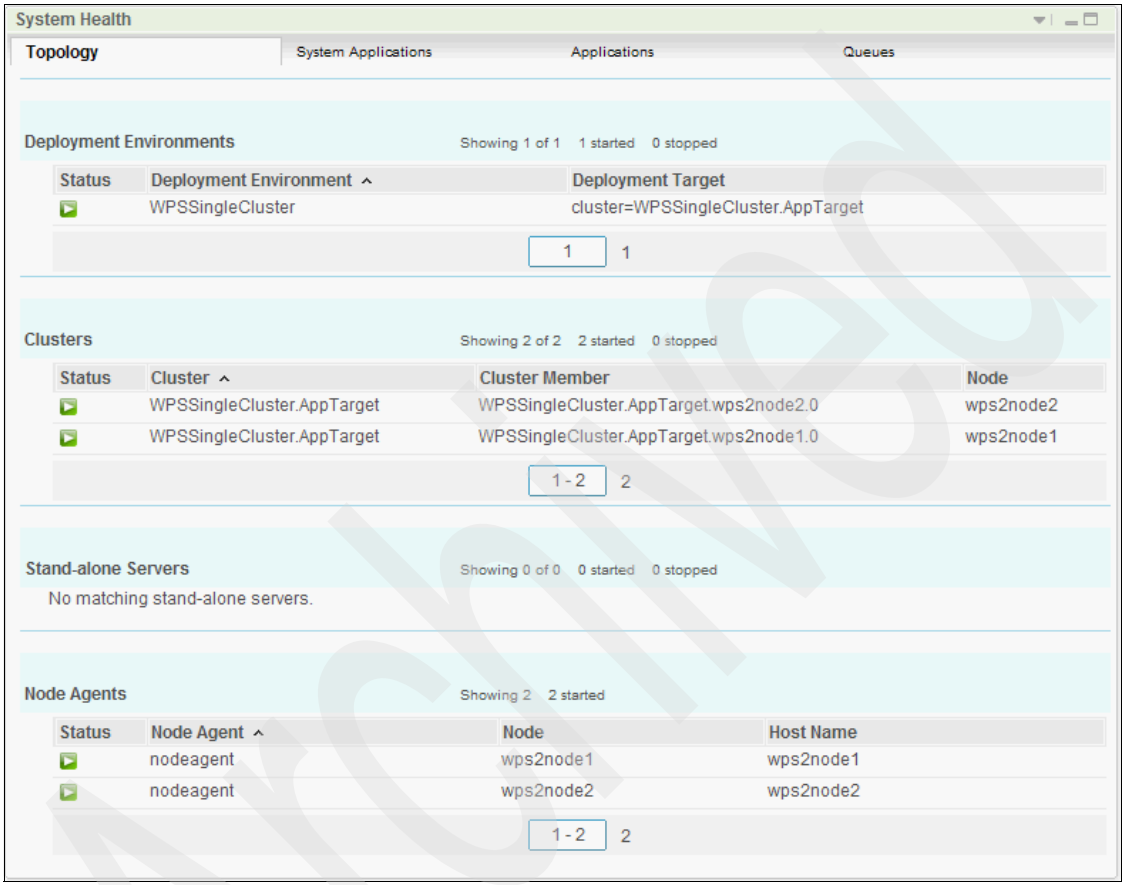


Figure 4-17 Business Space System Health widget

Archived

Archived

5

WebSphere Process Server: Remote Messaging and Remote Support

This chapter provides instructions for creating a Remote Messaging and Remote Support topology, a deployment environment pattern for WebSphere Process Server V7.0.

© Copyright IBM Corp. 2010. All rights reserved.

5.1 Topology summary

In this environment the Remote Messaging and Remote Support topology is deployed to two nodes, as shown in Figure 5-1.

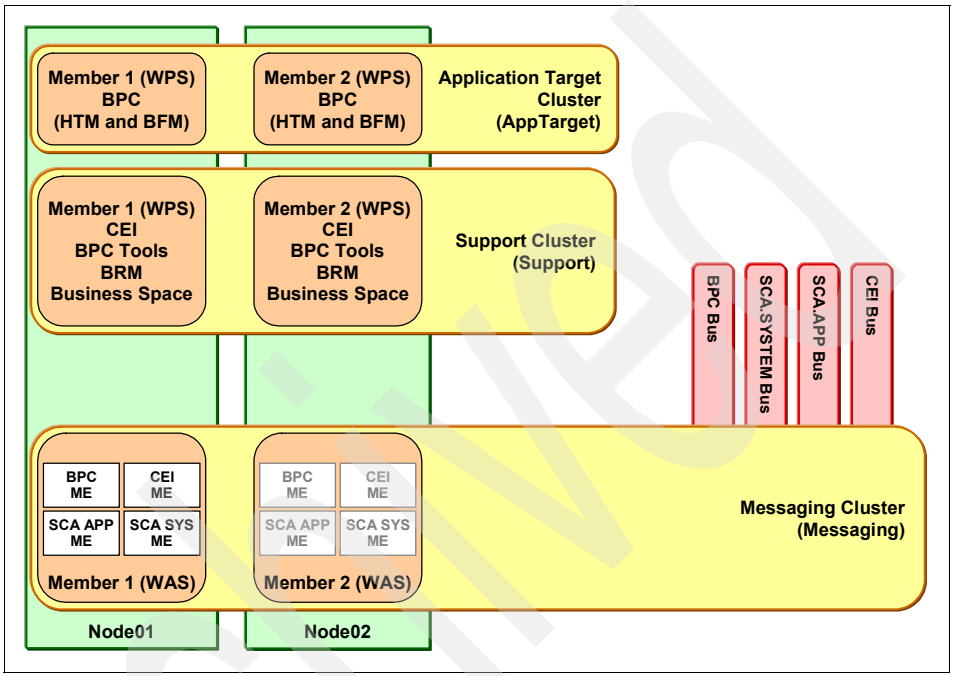


Figure 5-1 Remote Messaging and Remote Support topology

This deployment environment pattern consists of three clusters, as follows:

- Remote Messaging cluster
  - Service Component Architecture (SCA) application bus members
  - SCA system bus members
  - Business Process Choreographer (BPC) bus members
  - Common Event Infrastructure (CEI) bus members
- Remote Support cluster
  - CEI server application
  - Business rules manager
  - Business Process Choreographer components
- Application deployment cluster
  - Application deployment target
  - Business Process Choreographer container

Archived

The application deployment target is used to support SCA applications and Business Process Choreographer components. The messaging engines run on one cluster member at a time.

5.2 Prerequisites for creating the topology

Before you begin the creation of this topology, you must perform the following steps:

- Install and configure a database server using IBM DB2 as per “Software versions used in this book” on page 56.
- Install and configure an LDAP user registry using IBM Tivoli Directory Server as per “Software versions used in this book” on page 56.
- Populate the user registry with the appropriate users as per 3.4.2, “Configuring LDAP as the user account registry” on page 99.
- Install a Web server using IBM HTTP Server as per 3.6.1, “Installing IBM HTTP Server” on page 111.
- Install WebSphere Process Server for the deployment manager and all the participating nodes as per 3.2, “Software installation” on page 57.
- Create the required databases as per 3.3, “Database and schema creation” on page 82.
- Create the deployment manager profile, secure the deployment manager with the LDAP user registry, configure the Web server to the cell, and update the deployment manager configurations. This is described in 3.4, “Creating a deployment manager profile” on page 91.
- Create and federate two managed nodes to the deployment manager as per 3.5, “Creating custom profiles” on page 107.

5.3 Configuring the topology using the Integrated Solutions Console

This section provides instructions to create a Remote Messaging and Remote Support deployment environment pattern by using the Deployment Environment Configuration wizard from the Integrated Solutions Console (ISC).

Archived

5.3.1 Creating the deployment environment topology

Ensure that the deployment manager and all nodes are running. In addition, you must have the database design file that was generated by the database design tool (DDT), to be imported by the Deployment Environment Configuration wizard.

1. Log in to the Integrated Solutions Console as an administrative user.
2. Click **Servers** → **Deployment Environments**.
3. From the Deployment Environments panel, click **New**.
4. From the Create Deployment Environment panel (Figure 5-2), accept the default to **Create a deployment environment based on a pattern**. To view all configuration steps in the wizard, select **Detailed: Show all steps**.

   The deployment environment name allows you to uniquely identify the deployment environment. For this example, in the Deployment environment name field, enter `RMRS`. Click **Next**.

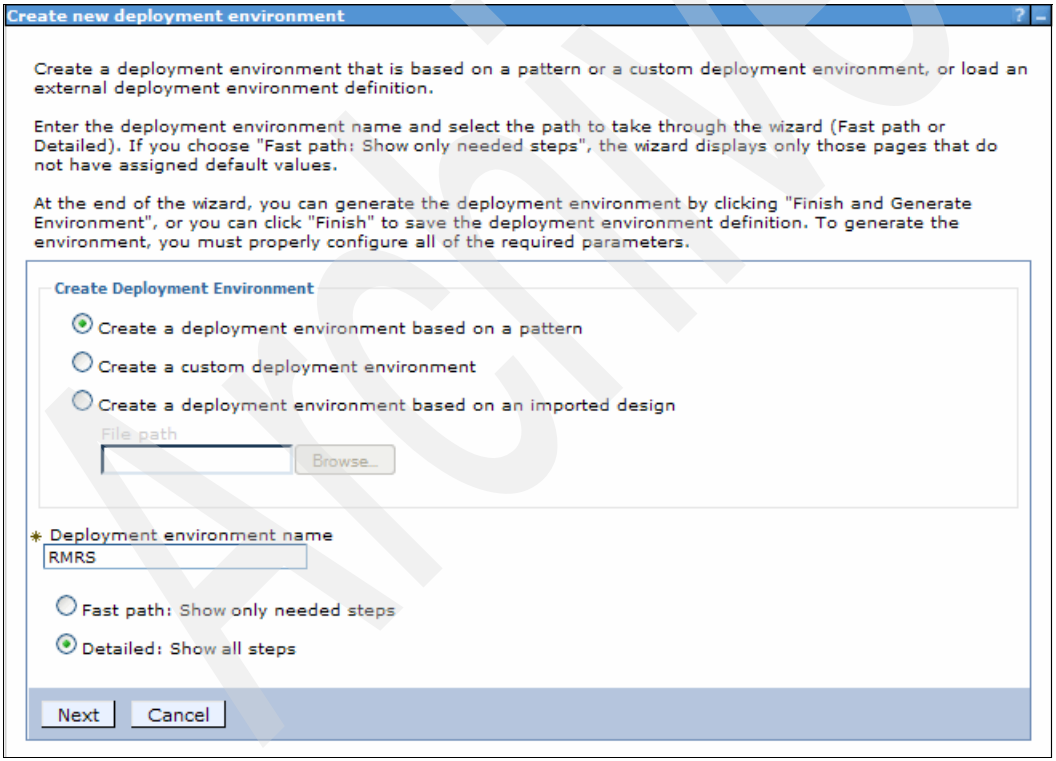


Figure 5-2 Creating a new deployment environment

Archived

5. From the Feature for the Deployment Environment panel (Figure 5-3), select **WPS**. Click **Next**.

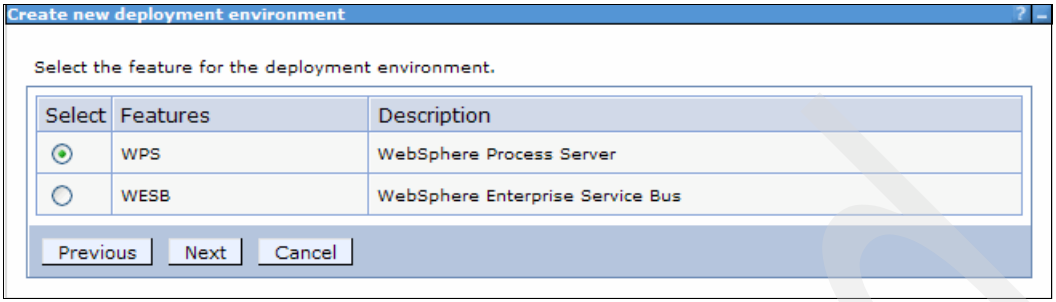


Figure 5-3 Feature for the deployment environment

6. From the Deployment Environment Patterns panel (Figure 5-4), accept the default **Remote Messaging and Remote Support** pattern. Click **Next**.

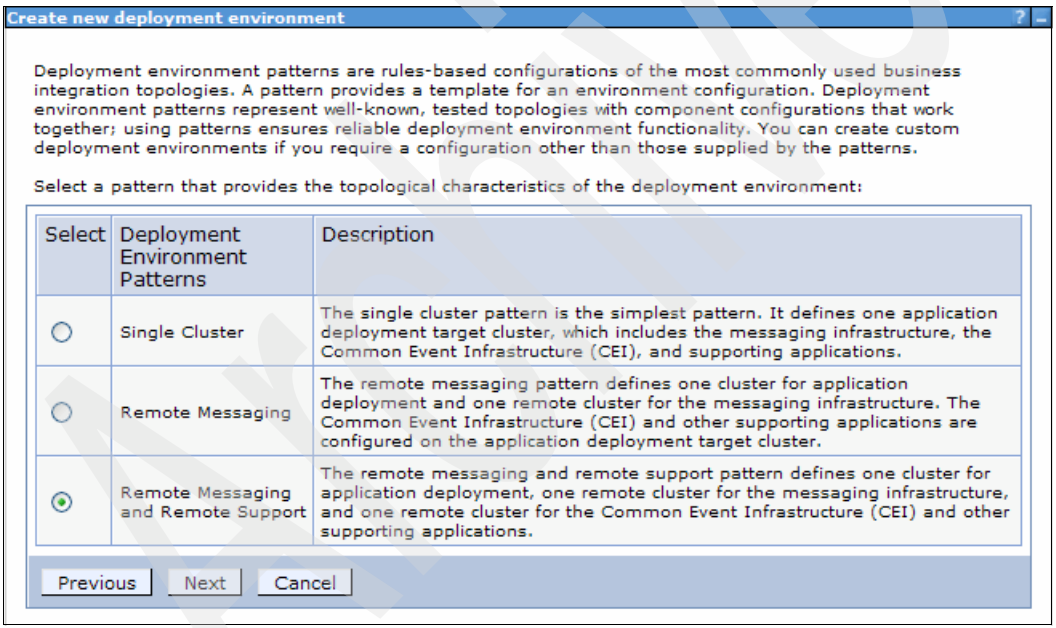


Figure 5-4 Deployment environment patterns

7. In this environment, the topology is created on two managed nodes. From the Select Nodes panel (Figure 5-5), check the boxes next to the available WebSphere Process Server nodes. Click **Next**.

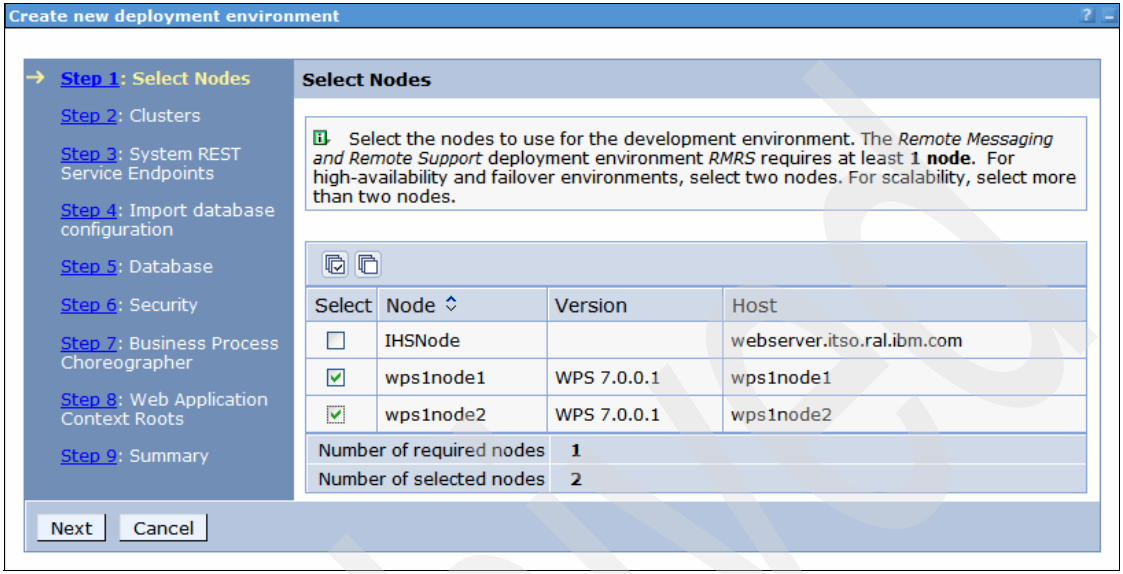


Figure 5-5 Select nodes

Archive

8. From the Clusters panel (Figure 5-6), each node is assigned as a member of all three clusters, application deployment target, messaging infrastructure, and supporting infrastructure. Accept the defaults. Click **Next**.

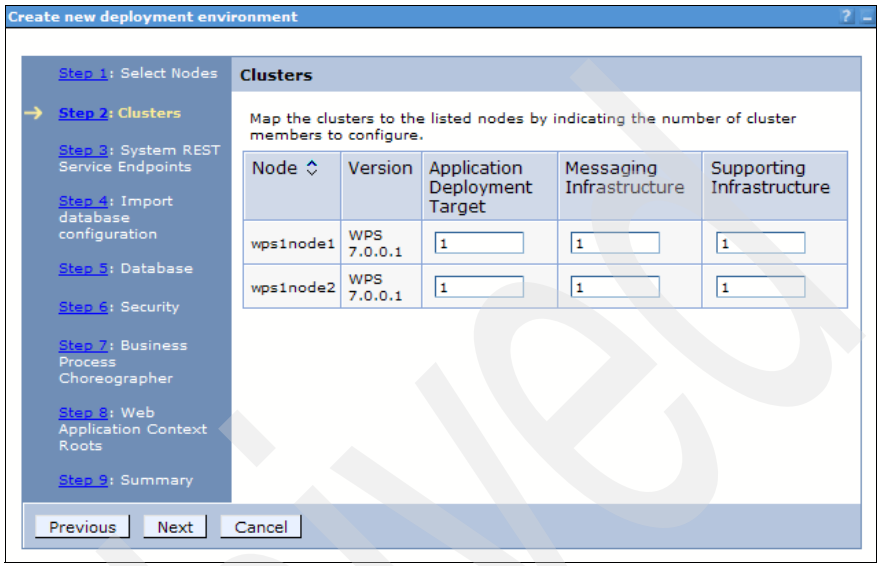


Figure 5-6 Clusters

Archived

9. From the System REST Service Endpoints panel (Figure 5-7), you can configure the REST application program interfaces (APIs). In order for the Business Space widgets to be available, you must configure the REST service endpoints for those widgets as follows:

   a. From the Protocol list, select either **http://** or **https://**.

   b. In the Host name or virtual host in a load-balanced environment field, enter the name of a Web server, a proxy server, or a proxy server cluster that has been configured in the above-mentioned prerequisite section.

   c. In the Port field, enter the port number that a client uses to communicate with the Web or proxy server. Click **Next**.

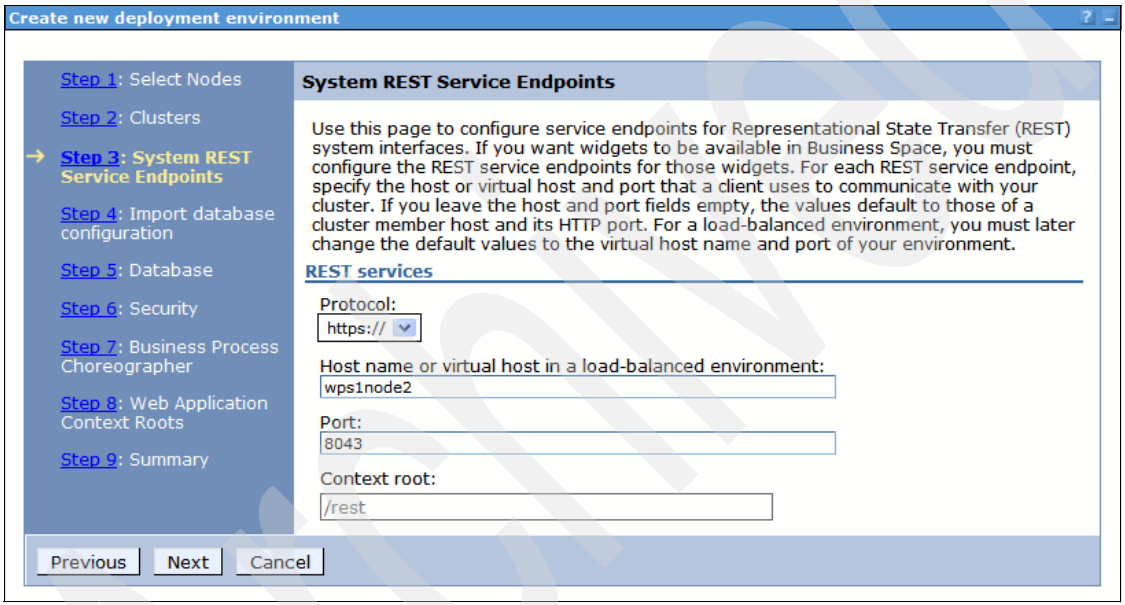


Figure 5-7 System REST service endpoints

10. In the Import database configuration panel (Figure 5-8), enter the fully qualified path name to the database design file that was created by the database design tool. This file must have the .dbDesign extension. Click **Next**.

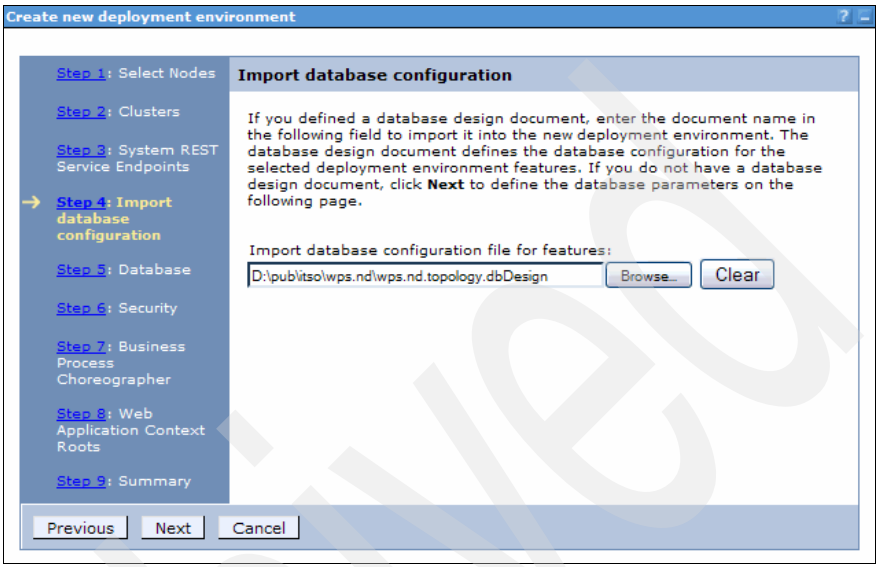


Figure 5-8 Import database configuration

Archived

11. From the Database panel (Figure 5-9), all data source components are populated with the data source configurations imported from the database design file.

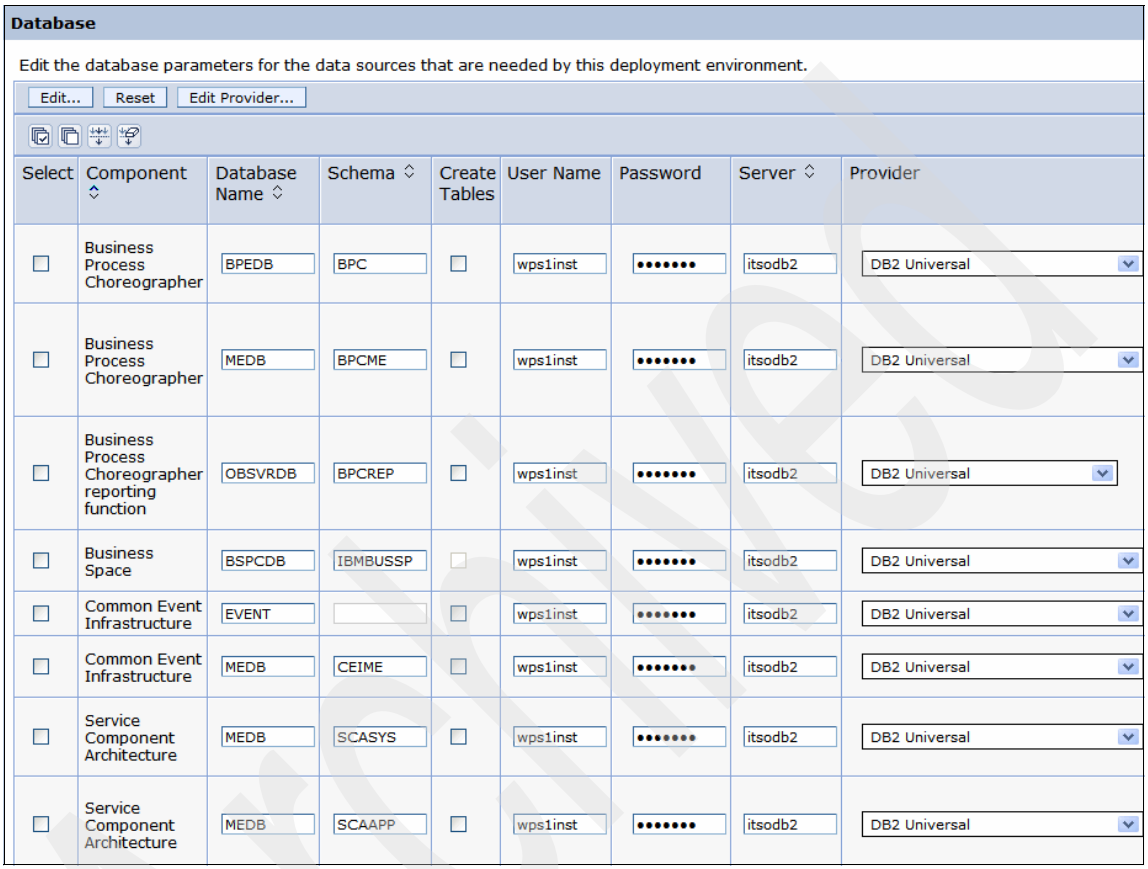


Figure 5-9 Database

In this environment, the createTables property from the database design file is set to false because the database tables are expected to have already been created. Therefore, the Create Tables column is cleared for all data sources, indicating that the database tables do not have to be created when they are accessed the first time. Click **Next**.

Note: The imported database design file does not modify the existing WebSphere Common database data source.

12. On the Security panel (Figure 5-10), enter the user names and passwords for each of the components:

   – Common Event Infrastructure
   – Service Component Architecture
   – Business Process Choreographer

   In this environment, the following user names are used:

   – `ceijms` - CEI JMS user
   – `sca` - SCA user
   – `bpcjms` - BPC JMS user

   Click **Next**.

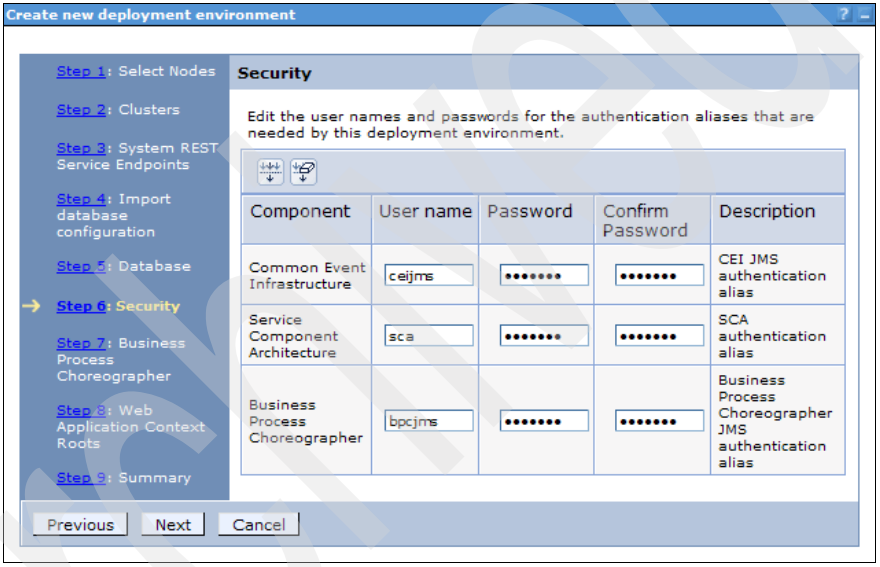


Figure 5-10 Security

Archived

13. On the Business Process Choreographer panel (Figure 5-11), enter the administrator and monitor roles for the business flow and human task role, and the user authentication information for the JMS API, escalation, and administration job user.

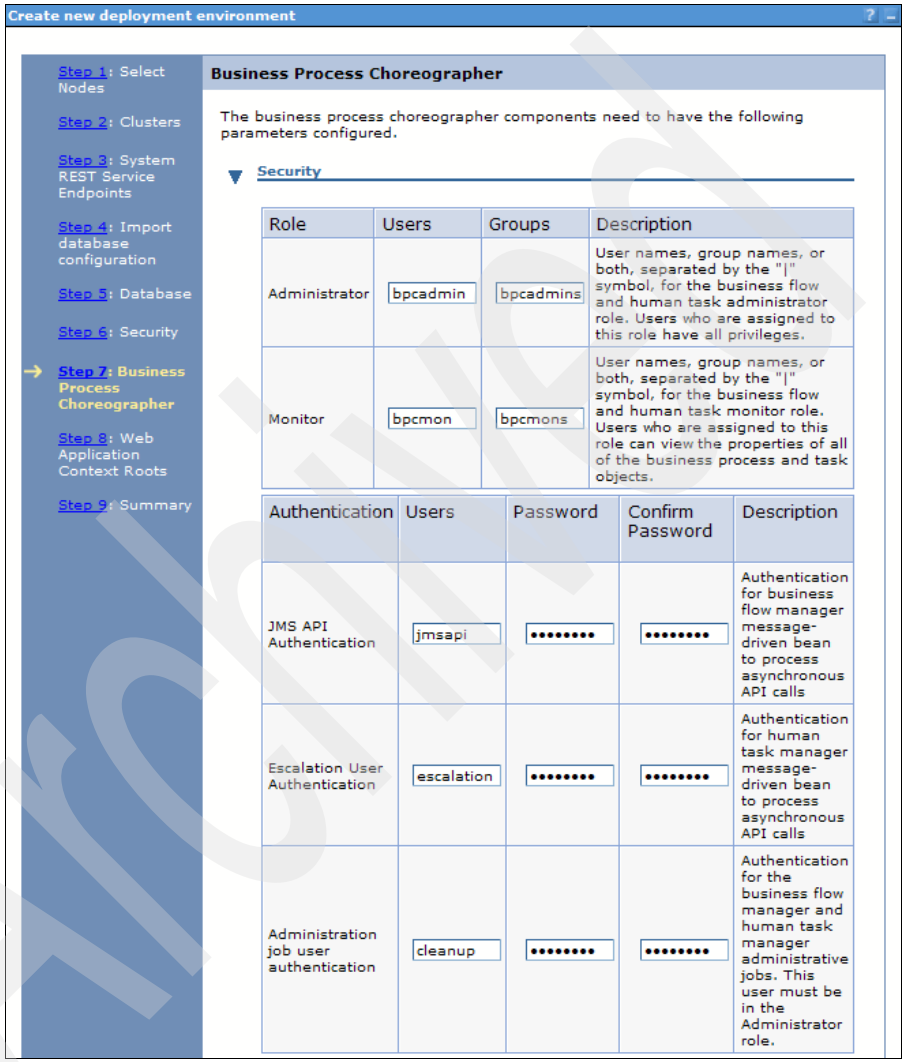


Figure 5-11 Business Process Choreographer security roles and authentications

Archived

In this environment, the following users are used:

- bpcadmin: business flow and human task administrator
- bpcmon: business flow and human task monitor
- jmsapi: business flow manager message-driven bean
- escalation: human task manager message-driven bean
- cleanup: business flow and human task manager administrative jobs

And the following groups are used:

- bpcadmins: business flow and human task administrator group
- bpcmons: business flow and human task monitor group

Click **Next**.

14. On the Human Task Manager Mail Session panel (Figure 5-12), clear the **Enable e-mail service** check box to disable e-mail escalation for human tasks. Note that if you want to enable features such as escalation e-mails, leave this check box enabled. Click **Next**.

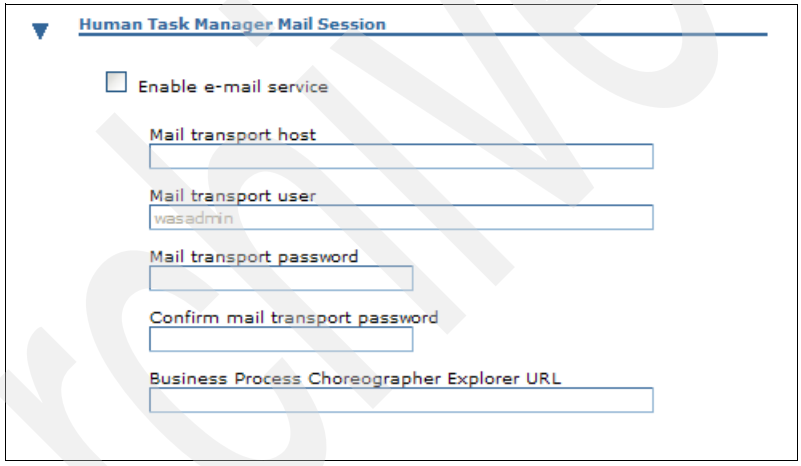


Figure 5-12 Human Task Manager Mail Session

Archived

15. On the Web Application Context Roots panel (Figure 5-13), accept the default context root for Business Process Choreographer Explorer, Business Rules Manager, and Business Space. Click **Next**.

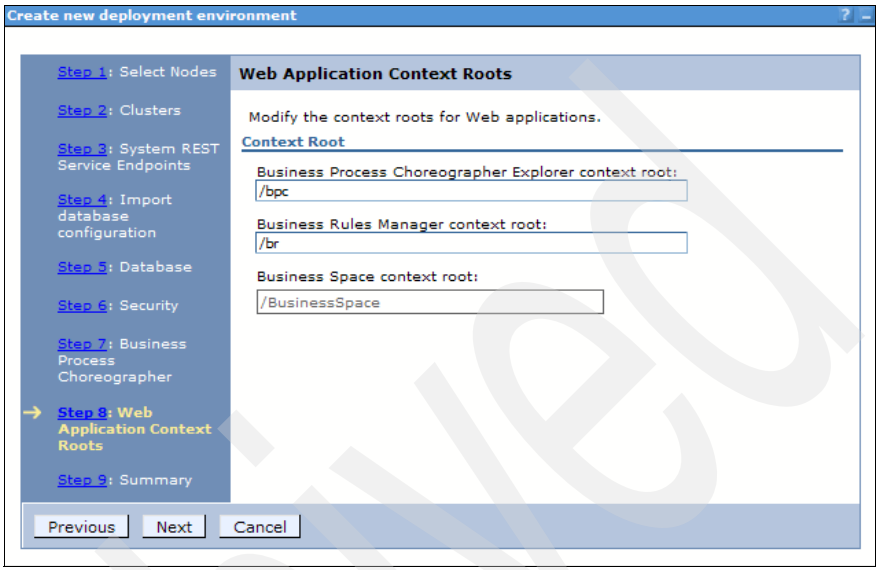


Figure 5-13 Web Application Context Roots

16. The Summary panel shows an overview of the new deployment environment. On this panel you have the choice to save the deployment environment definition or to both save the definition and generate the deployment environment. It is a leading practice to save the deployment environment configuration without generating it so you may examine and verify or modify the topology prior to generating the environment. Click **Finish**, as shown in Figure 5-14.

Figure 5-14 Save the deployment environment definitions

17. From the Deployment Environments panel, click **Save** to save the changes to the master configuration.

Archived

5.3.2 Generating the deployment environment topology

To generate the deployment environment from a deployment environment configuration:

1. Log in to the Integrated Solutions Console as an administrative user.
2. Click **Servers** → **Deployment Environments**. On the Deployment Environments panel, the deployment environment in the *not configured* state is as shown in Figure 5-15.

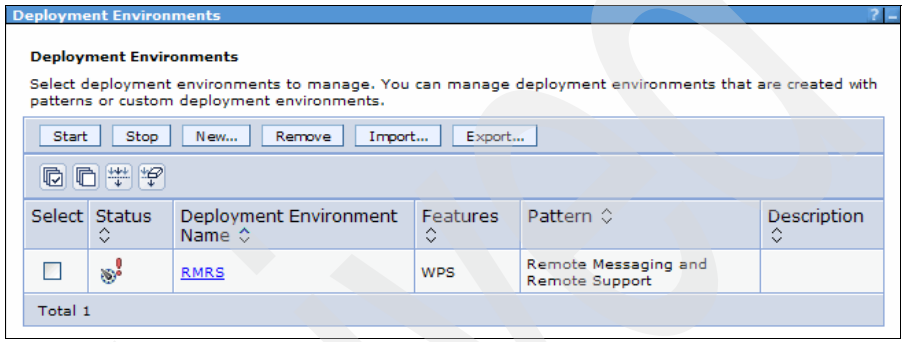


Figure 5-15 Deployment environments: unconfigured

3. Click the name of deployment environment, **RMRS**.

Archived

4. On the Deployment Environment Configuration panel (Figure 5-16), you may examine and modify the deployment topology, data sources, and authentication aliases. To complete this deployment environment, click **Generate Environment**.

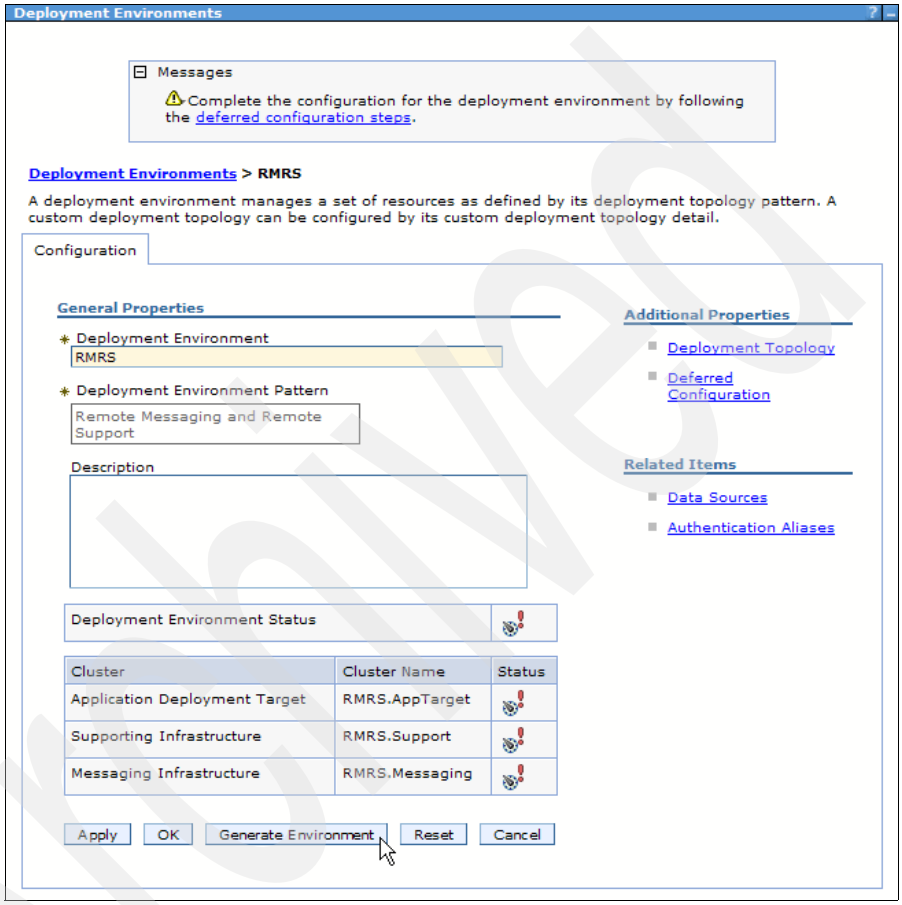


Figure 5-16 Generate the deployment environment

Archived

5. The Configuration Status panel displays the configuration progress. This step takes time to complete. Upon a successful deployment, click **Save Changes**, as shown in Figure 5-17.

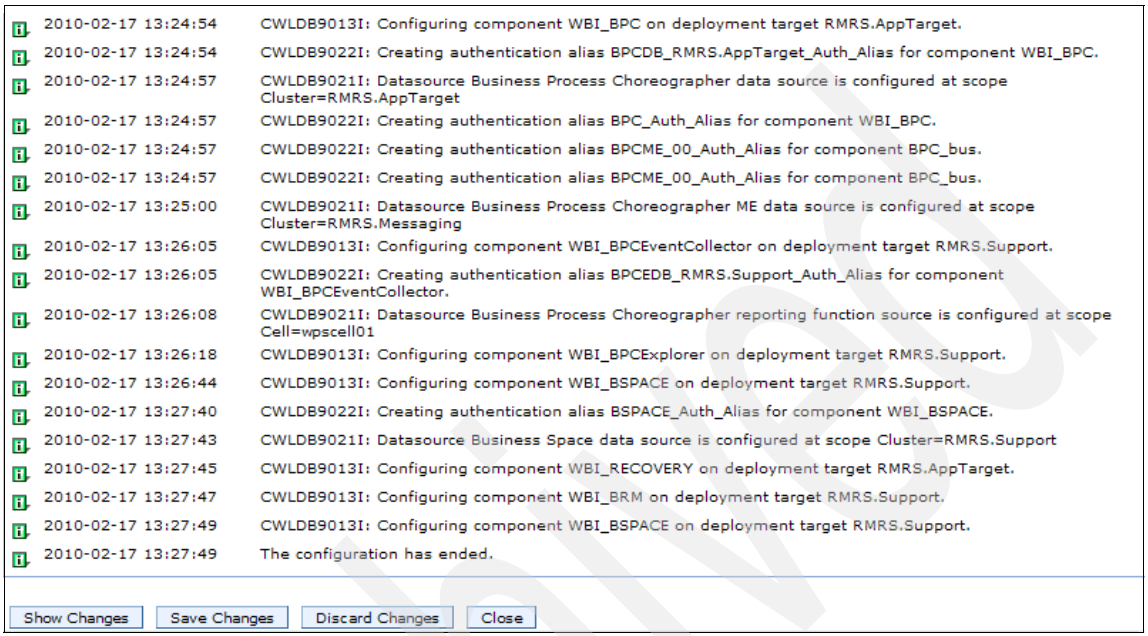


Figure 5-17 Deployment environment configuration status

6. From the Deployment Environments panel, click **Save** to save the changes to the master configuration.

> **Note:** The generation of the deployment environment, the saving of the configuration to the master repository, and the synchronization with all the nodes take time to complete. Do not restart the deployment manager or any of the node agents. To ensure that the generation of the deployment environment has successfully completed, refer to 5.5.9, “Verifying the deployment environment” on page 187.

On the Deployment Environments panel, the deployment environment is in the stop state, as shown in Figure 5-18.

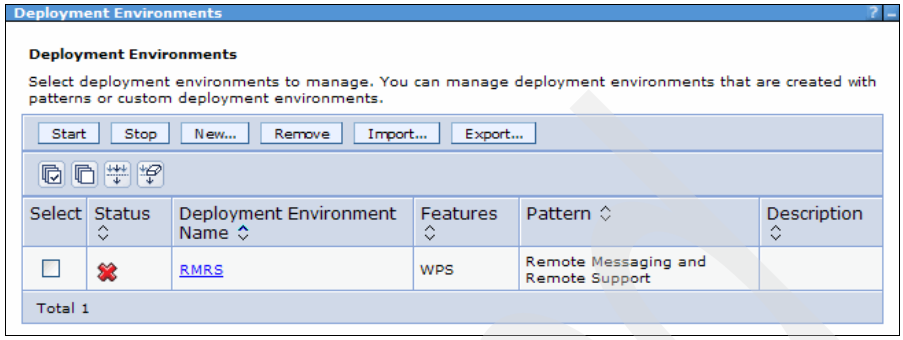


Figure 5-18 Deployment environments: stopped

5.4 Configuring the topology manually using the command line

This section describes the creation of a Remote Messaging and Remote Support (RMRS) deployment environment by using the **wsadmin** command.

Note: The generation of a deployment environment may take a long time. To prevent incomplete configuration due to timeout, set the value of com.ibm.SOAP.requestTimeout to at least 1800 in:

`<dmgr_profile_path>/properties/soap.client.props`

5.4.1 Create a new deployment environment

From a deployment manager, you can create a new deployment environment by using the **createDeploymentEnvDef**, **addNodeToDeploymentEnvDef**, and **generateDeploymentEnv** commands.

Note: By default, the **createDeploymentEnvDef** command:

- Assigns the SCA_Auth_Alias authentication alias to the Business Integration Security authentication aliases.
- Does not set the administrator group role (adminGroups) and the monitor group role (monitorGroups) for the business flow manager and human task manager.
- Does not set the administration job user (cleanupUser) and password (cleanupPassword) for the business flow manager and human task manager cleanup services and the process instance migration tool.
- Does not set the Representational State Transfer (REST) service endpoints. The REST service endpoints are assigned the default values for the host name (localhost) and the port value (9080).

This book provides a wsadmin Jython script, UpdateWPITopology.py, to configure a complete business integration security environment using specific roles and authentication users, and REST service endpoints using a Web server or a proxy server for load-balancing HTTP requests.

The general steps to create a new deployment environment are:

1. Start the deployment manager and all nodes.
2. Log in to the deployment manager.
3. Create a new deployment environment by using the **createDeploymentEnvDef** and *addNodeToDeploymentEnvDef* commands. Save the deployment environment definitions. Do not generate it.
4. Update the deployment environment definitions by using the provided wsadmin Jython script, `UpdateWBITopology.py`.

   Note: The wsadmin Jython script, `UpdateWBITopology.py`, must be run before the deployment environment is configured with the **generateDeploymentEnv** command.

5. Generate the new deployment environment by using the **generateDeploymentEnv** command.

In addition, you must have the database design file that was generated by the database design tool (DDT), to be loaded by the **createDeploymentEnvDef** command.

Creating the deployment environment definitions

Example 5-1 shows the wsadmin Jython script, `createEnvironment.py`, used to create a new deployment environment pattern.

Example 5-1 createEnvironment.py: script to create new deployment environment pattern

```
import sys
import java

topologyName='RMRS'
dbDesignFile='/tmp/wps.RMRS.dbDesign'

AdminTask.createDeploymentEnvDef ([ '-topologyName',  topologyName,
'-topologyPattern', 'RemoteMessagingAndSupport', '-topologyRuntime',
'WPS', '-dbDesign', dbDesignFile ])

AdminTask.addNodeToDeploymentEnvDef ([ '-topologyName',  topologyName,
'-topologyRole', 'ADT', '-nodeRuntime', 'WPS', '-nodeName',
'wps1node1', '-serverCount', '1' ])
AdminTask.addNodeToDeploymentEnvDef ([ '-topologyName',  topologyName,
'-topologyRole', 'Messaging', '-nodeRuntime', 'WPS', '-nodeName',
'wps1node1', '-serverCount', '1' ])
AdminTask.addNodeToDeploymentEnvDef ([ '-topologyName',  topologyName,
'-topologyRole', 'Support', '-nodeRuntime', 'WPS', '-nodeName',
'wps1node1', '-serverCount', '1' ])

AdminTask.addNodeToDeploymentEnvDef ([ '-topologyName',  topologyName,
'-topologyRole', 'ADT,Messaging,Support', '-nodeRuntime', 'WPS',
'-nodeName', 'wps1node2', '-serverCount', '1' ])

AdminTask.validateDeploymentEnvDef ([ '-topologyName', topologyName ])

AdminTask.showDeploymentEnvStatus ([ '-topologyName', topologyName ])
AdminConfig.save()
```

In the `createEnvironment.py` script the **`wsadmin`** commands perform the following tasks:

1. Create a new deployment environment definition with an existing database design file using the **`createDeploymentEnvDef`** command:

   ```
   AdminTask.createDeploymentEnvDef ([ '-topologyName',  topologyName,
   '-topologyPattern', 'RemoteMessagingAndSupport', '-topologyRuntime',
   'WPS', '-dbDesign', dbDesignFile ])
   ```

2. Add a node to the application deployment target role by using the **addNodetoDeploymentEnvDef** command:

```
AdminTask.addNodeToDeploymentEnvDef ([ '-topologyName',
topologyName, '-topologyRole', 'ADT', '-nodeRuntime', 'WPS',
'-nodeName', 'wps1node1', '-serverCount', '1' ])
```

3. Add a node to the messaging infrastructure role:

```
AdminTask.addNodeToDeploymentEnvDef ([ '-topologyName',
topologyName, '-topologyRole', 'Messaging', '-nodeRuntime', 'WPS',
'-nodeName', 'wps1node1', '-serverCount', '1' ])
```

4. Add a node to the support infrastructure role:

```
AdminTask.addNodeToDeploymentEnvDef ([ '-topologyName',
topologyName, '-topologyRole', 'Support', '-nodeRuntime', 'WPS',
'-nodeName', 'wps1node1', '-serverCount', '1' ])
```

5. Add a node to all roles at the same time:

```
AdminTask.addNodeToDeploymentEnvDef ([ '-topologyName',
topologyName, '-topologyRole', 'ADT,Messaging,Support',
'-nodeRuntime', 'WPS', '-nodeName', 'wps1node2', '-serverCount', '1'
])
```

6. Validate the deployment environment definitions by using the **validateDeploymentEnvDef** command:

```
AdminTask.validateDeploymentEnvDef ([ '-topologyName', topologyName
])
```

7. Display the current status of the deployment environment by using the **showDeploymentEnvStatus** command:

```
AdminTask.showDeploymentEnvStatus ([ '-topologyName', topologyName
])
AdminConfig.save()
```

   At this point, the status of the deployment environment is complete. This state is also known as *not configured*. It indicates that the configuration is completed but that it has not been generated.

> **Note:** The complete list of deployment environment states is described in the Information Center article at:
>
> http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/topic/com.ibm.websphere.wps.doc/doc/tadm_dispdepenvdefstatcmd.html
>
> It is a leading practice to save only the deployment environment configuration without generating it so you may examine and verify or modify the topology prior to generating the environment.

Archived

You can run this wsadmin Jython script, `createEnvironment.py`, as follows:

```
cd <dmgr_profile>/bin
./wsadmin.sh -lang jython -f createEnvironment.py -connType SOAP -host
<dmgr_host> -port <dmgr_SOAP_port> -user <dmgr_admin_user> -password
<dmgr_admin_user_password>
```

Updating the deployment environment definitions

After creating the deployment environment pattern, the business integration security roles and authentication users, and REST service endpoints are set to default values. Update these deployment environment definitions by using the wsadmin Jython script, `UpdateWBITopology.py`, included with this publication.

1. Edit the profile script file `setup.py` and modify the following environment variables:
   - PARAMETERS_FILE: a property file containing name and value pairs of the deployment environment parameters to be updated. For this example, the parameter file is called `wbitopology.properties`.
   - CELLNAME: the name of your cell.
   - TOPOLOGY_NAME: the name of your deployment environment.
2. Edit the parameter file, `wbitopology.properties`, and modify the deployment environment parameters that you want to update.
3. Run the wsadmin Jython script, `UpdateWBITopology.py`, as follows:

   ```
   cd <dmgr_profile>/bin
   ./wsadmin.sh -lang jython -profile setup.py -f UpdateWBITopology.py
   -connType NONE
   ```

   Note: The **wsadmin** command-line options -profile and -profileName are not the same. The -profile option specifies a profile script to run before other scripts. The -profileName option indicates the profile from which the wsadmin tool will run.

Generating the deployment environment

After creating and updating the deployment environment definitions, you can generate a new deployment environment by using the **generateDeploymentEnv** command.

Archive

Example 5-2 shows the wsadmin Jython script, `generateEnvironment.py`, used to generate a new deployment environment. If the script detects any error conditions, it discards the changes and does not save the failed configuration to the master repository.

Example 5-2 generateEnvironment.py: script to generate a new deployment environment

```
import sys
import java

topologyName='RMRS'

try:
  AdminTask.generateDeploymentEnv ([ '-topologyName', topologyName ])
except:
  print "Caught Exception ", sys.exc_info()
  sys.exit(1)
#endTry

AdminConfig.save()
```

You can run this wsadmin Jython script, `generateEnvironment.py`, as follows:

```
cd <dmgr_profile>/bin
./wsadmin.sh -lang jython -f generateEnvironment.py -connType SOAP
-host <dmgr_host> -port <dmgr_port> -user <dmgr_admin_user> -password
<dmgr_admin_password>
```

Note: The generation of the deployment environment, the saving of the configuration to the master repository, and the synchronization with all the nodes takes time to complete. Do not restart the deployment manager or any of the node agents. To ensure that the generation of the deployment environment has successfully completed, refer to 5.5.9, “Verifying the deployment environment” on page 187.

5.4.2 Export existing deployment environment definitions

From a deployment manager, you can export existing deployment environment definitions by using the **`exportDeploymentEnvDef`** command. Then this deployment environment definition file can be used to import into another deployment manager for the purpose of creating a new deployment environment or replicating a working deployment environment.

Archived

Example 5-3 shows the wsadmin Jython script, `exportEnvironment.py`, used to export an existing deployment environment.

Example 5-3 exportEnvironment.py: script to export an existing deployment environment

```
import sys
import java

topologyName='RMRS'
topologyDefinitionFile='/tmp/RMRS.xml'

AdminTask.exportDeploymentEnvDef ([ '-topologyName', topologyName,
'-filePath', topologyDefinitionFile ])
```

The **exportDeploymentEnvDef** command is used to export the deployment environment definitions of an existing deployment environment, RMRS. and store the result in a file, `/tmp/RMRS.xml`.

You can run this Jython script, `exportEnvironment.py`, as follows:

```
cd /opt/ibm/WebSphere/ProcServer/profiles/wps1dmgr/bin
./wsadmin.sh -lang jython -f exportEnvironment.py -connType SOAP -host
itsodmgr -port 8879 -user wpsadmin -password itso4you
```

5.4.3 Import and generate to create a new deployment environment

From a deployment manager, you can import existing deployment environment definitions to create or replicate a deployment environment by using the **importDeploymentEnvDef** and **generateDeploymentEnv** commands.

Note: The generation of a deployment environment may take a long time. To prevent incomplete configuration due to timeout, set the value of com.ibm.SOAP.requestTimeout to at least 1800 in:

`<dmgr_profile_path>/properties/soap.client.props`

Archived

The deployment manager from which you import the deployment environment definitions must meet the following conditions:

- Does not have a deployment environment with the same name as the deployment environment definitions

  Note: Deleting the deployment environment does not remove the configuration of servers, nodes, and clusters of the deployment environment.

- Supports all functions that are defined in the deployment environment definitions

You may want to modify certain parameters in this existing deployment environment design file as necessary for this new deployment manager, such as the topology name, node name, or the database server name and port.

In this example, the deployment environment design file was edited to change the topology name to GoldTopology by replacing all string instances from RMRS to GoldTopology.

Example 5-4 shows the wsadmin Jython script, `importEnvironment.py`, used to import a deployment environment definition file and to create a new deployment environment from this imported configuration.

Example 5-4 importEnvironment.py: script to import and generate new deployment environment

```
import sys
import java

topologyName='GoldTopology'
topologyDefinitionFile='/tmp/RMRS.xml'

AdminTask.importDeploymentEnvDef ([ '-topologyName', topologyName,
'-filePath', topologyDefinitionFile ])
AdminTask.validateDeploymentEnvDef ([ '-topologyName', topologyName ])
AdminTask.showDeploymentEnvStatus ([ '-topologyName', topologyName ])

try:
  AdminTask.generateDeploymentEnv ([ '-topologyName', topologyName ])
except:
```

Archived

```
  print "Caught Exception ", sys.exc_info()
  sys.exit(1)
#endTry

AdminConfig.save()
```

In the importEnvironment.py script, the **wsadmin** commands perform the following tasks:

1. Import the deployment environment definitions by using the **importDeploymentEnvDef** command:

   ```
   AdminTask.importDeploymentEnvDef ([ '-topologyName', topologyName,
   '-filePath', topologyDefinitionFile ])
   ```

2. Validate the deployment environment definitions by using the **validateDeploymentEnvDef** command:

   ```
   AdminTask.validateDeploymentEnvDef ([ '-topologyName', topologyName
   ])
   ```

3. Display the current status of the deployment environment by using the **showDeploymentEnvStatus** command:

   ```
   AdminTask.showDeploymentEnvStatus ([ '-topologyName', topologyName
   ])
   ```

4. Generate the new deployment environment by using the **generateDeploymentEnvDef** command. If the script detects any error conditions, it discards the changes and does not save the failed configuration to the master repository:

   ```
   AdminTask.generateDeploymentEnv ([ '-topologyName', topologyName ])
   ```

5. Save the configuration to the master repository:

   ```
   AdminConfig.save()
   ```

You can run this wsadmin Jython script, `importEnvironment.py`, as follows:

```
cd <dmgr_profile>/bin
./wsadmin.sh -lang jython -f importEnvironment.py -connType SOAP -host
<dmgr_host> -port <dmgr_SOAP_port> -user <dmgr_admin_user> -password
<dmgr_admin_password>
```

Note: The generation of the deployment environment, the saving of the configuration to the master repository, and the synchronization with all the nodes takes time to complete. Do not restart the deployment manager or any of the node agents. To ensure that the generation of the deployment environment has successfully completed, refer to 5.5.9, “Verifying the deployment environment” on page 187.

5.5 Post-creation configuration and verification

After completing the generation of the deployment environment, verify the following items prior to starting all the clusters in the deployment environment:

- Configuring deferred configurations
- Configuring Business Integration Security
- Configuring REST service endpoints
- Configuring host aliases
- Configuring a Web server

5.5.1 Configuring deferred configurations

The Deferred Configuration page provide instructions to locate and run scripts to create databases and database tables. Prior to the creation of the deployment environment, most of the required databases have been created. To view this Deferred Configuration page, perform the following steps:

1. Log in to the Integrated Solutions Console as an administrative user.
2. Click **Servers** → **Deployment Environments**.
3. From the Deployment Environments panel (Figure 5-19), click the name of the deployment environment.

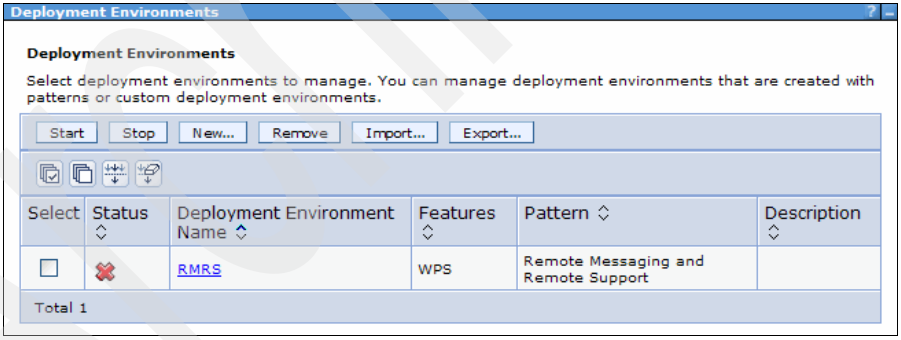


Figure 5-19 Deployment environments

4. In the Configuration tab, under Additional Properties, click **Deferred Configurations**.
5. From the list of instructions, all databases have already been created except for the common event infrastructure database.

Archived

Creating the event database

The common event infrastructure database is only created after the generation of the deployment environment. The generated common event infrastructure database script directory is created in the deployment manager profile in the following directory path:

`/opt/ibm/WebSphere/ProcServer/profiles/wps1dmgr/`*`databases/event/`*`RMRS.Support`*`/dbscripts/db2`*

1. Copy the generated common event infrastructure database directory to the database server.
2. Log in to the database server as the database instance owner.
3. Create the common event infrastructure database and tables, as shown in Example 5-5.

Example 5-5 Creating the event database from the database server

```
cd databases/event/RMRS.Support/dbscripts/db2
./cr_event_db2.sh 2>&1 | tee create_event_database.log
Do you have a client or server connection to the database on the
machine where this script is being run?
Type '1' for server or '2' for client.
1
Enter DB2 user id:
wps1inst
Password: itso4you
...
```

Checking database connections

After creating all databases, to ensure that all data sources have database connections, perform the following steps:

1. Log in to the Integrated Solutions Console as an administrative user.
2. Click **Resources** → **JDBC** → **Data sources**.
3. Set the scope to **All scopes**.
4. Select all data sources and click **Test connection**.

For each data source connection, there must be a successful message.

Archived

(Optional) Confirming the deferred configurations

You may want to return to the deployment environment deferred configuration page and clear the lists as follows:

1. Log in to the Integrated Solutions Console as an administrative user.
2. Click **Servers** → **Deployment Environments**.
3. Click the name of the deployment environment.
4. In the Configuration tab, under Additional Properties, click **Deferred Configurations**.
5. Click **Configuration Done**.
6. Click **OK** and **Save** to save the updates.

5.5.2 Configuring Business Integration security

In a high-security configuration, it is necessary to assign the BPC security roles and authentication users to different users.

Note: If you used the `createDeploymentEnvDef` command to create the deployment environment, all Business Process Choreographer (BPC) related security roles and authentication users are assigned to the `SCA_Auth_Alias` authentication alias.

This book provides a wsadmin Jython script, `UpdateWPITopology.py`, to configure a complete business integration security environment using specific roles and authentication users. Refer to 5.4.1, “Create a new deployment environment” on page 166, for further details.

To configure complete security for Business Process Choreographer, perform the following steps:

1. Log in to the Integrated Solutions Console as an administrative user.
2. Click **Security** → **Business Integration Security**.

Archived

3. From the Authentication Alias pane (Figure 5-20), you can modify the following authentication aliases:
   - Business Process Choreographer (BPC) Java Message Service (JMS) user
   - Common Event Infrastructure (CEI) JMS user
   - Service Common Architecture (SCA) user

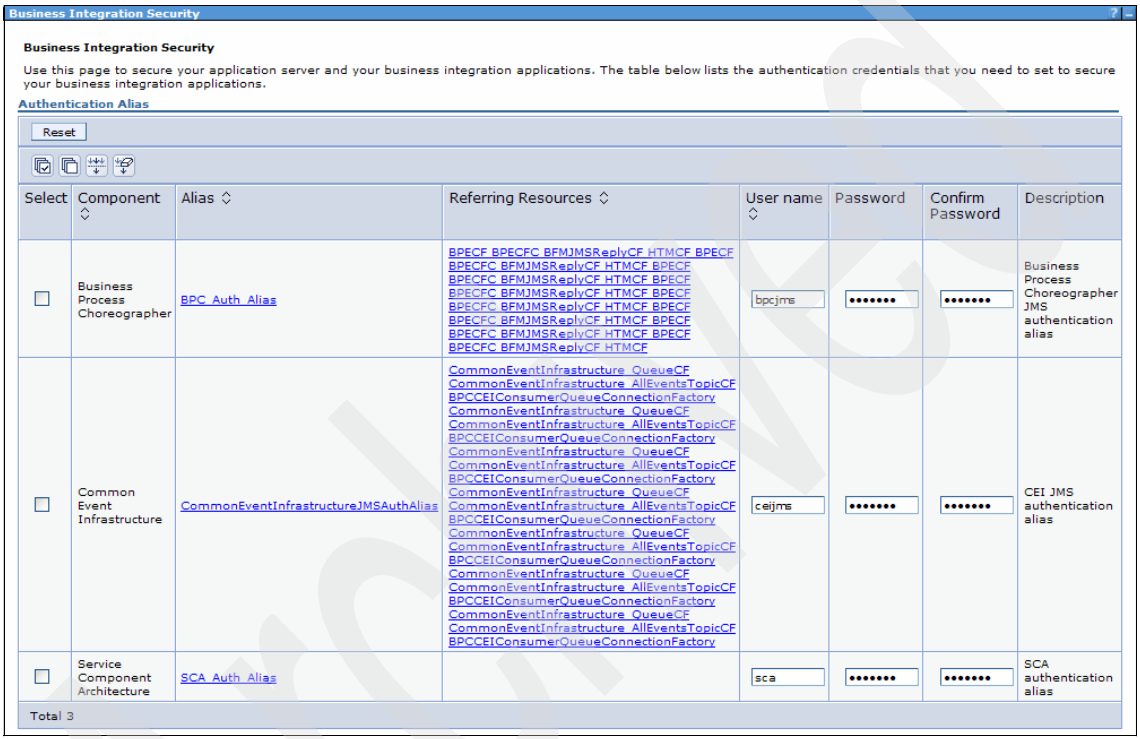


Figure 5-20 Business integration security: authentication alias

Around

4. From the Business Process Choreographer Security pane (Figure 5-21), you can modify the following roles and authentications:
   - Administrator user and group roles for both the business flow manager and the human task manager.
   - Monitor user and group roles for both the business flow manager and the human task manager.
   - JMS API user for the business flow manager.
   - Escalation user for the human task manager.
   - Administration job user.

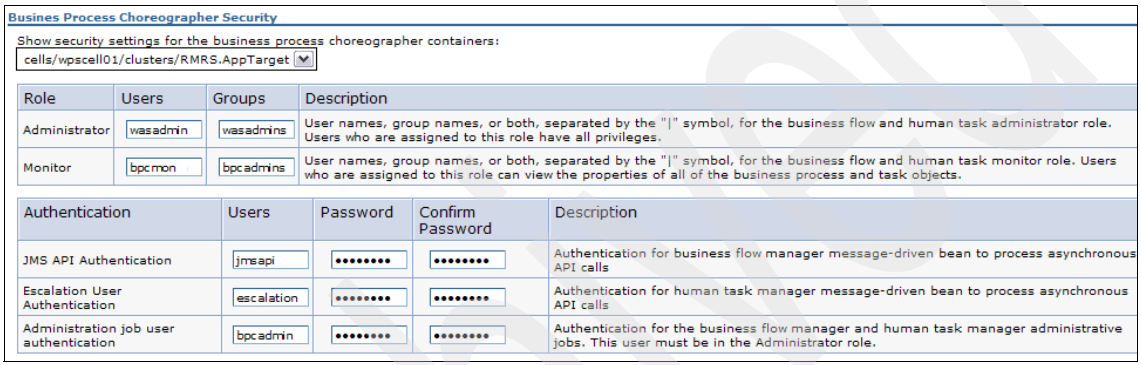


Figure 5-21 Business integration security: Business Process Choreographer Security

5. Click **OK** and **Save** to save any changes.

5.5.3 Configuring REST service endpoints

After the generation of the deployment environment, you must verify that the REST service provider endpoints are synchronized with the mashups endpoints for the business flow manager and the human task manager.

> **Note:** If you used the **`createDeploymentEnvDef`** command to create the deployment environment, the REST service endpoint is defaulted to the local host name.
>
> This book provides a wsadmin Jython script, `UpdateWPITopology.py`, to configure the REST service endpoints using a Web server or a proxy server for load-balancing HTTP requests. Refer to 5.4.1, “Create a new deployment environment” on page 166, for further details.

To configure the REST service endpoints for the business flow manager and the human task manager, perform the following steps:

1. Log in to the Integrated Solutions Console as an administrative user.
2. Click **Services** → **REST services** → **REST service providers**.
3. Click the name of the Business Process Choreographer container of the application deployment target for your deployment environment (for example, BPEContainer_RMRS.AppTarget).
4. On the Configuration tab, modify the following attributes:
   - Protocol: Enter `http` or `https`.
   - Host name or virtual host in a load-balanced environment: Enter the host name of the Web server.
   - Port: Enter the port number on which the Web server is listening.

   Click **OK** and **Save** to save any changes.

   The REST service endpoint for the Business Flow Manager (Figure 5-22) is `http://<hostname>:<port>/rest/bpm/bfm/`.

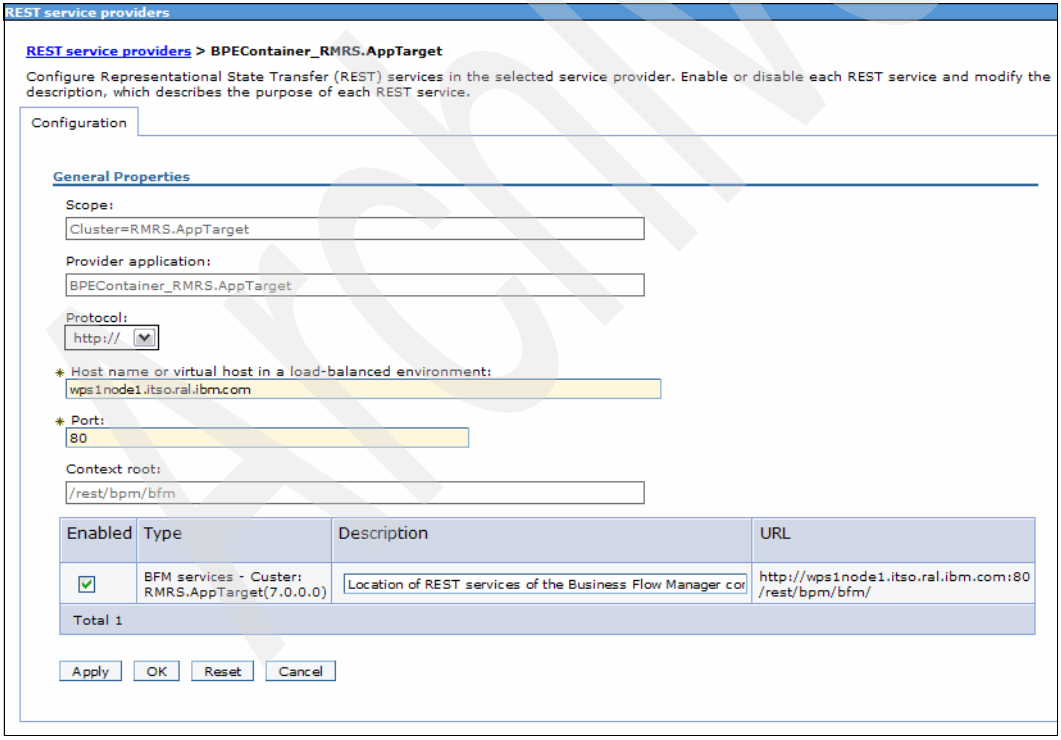


Figure 5-22 REST service endpoint for the business flow manager

5. Click **REST service providers**.
6. Click the name of the human task container of the application deployment target for your deployment environment (for example, TaskContainer_RMRS.AppTarget).
7. Repeat step 4. The REST service endpoint for the human task manager (Figure 5-23) is:

   `http://<hostname>:<port>/rest/bpm/htm/`

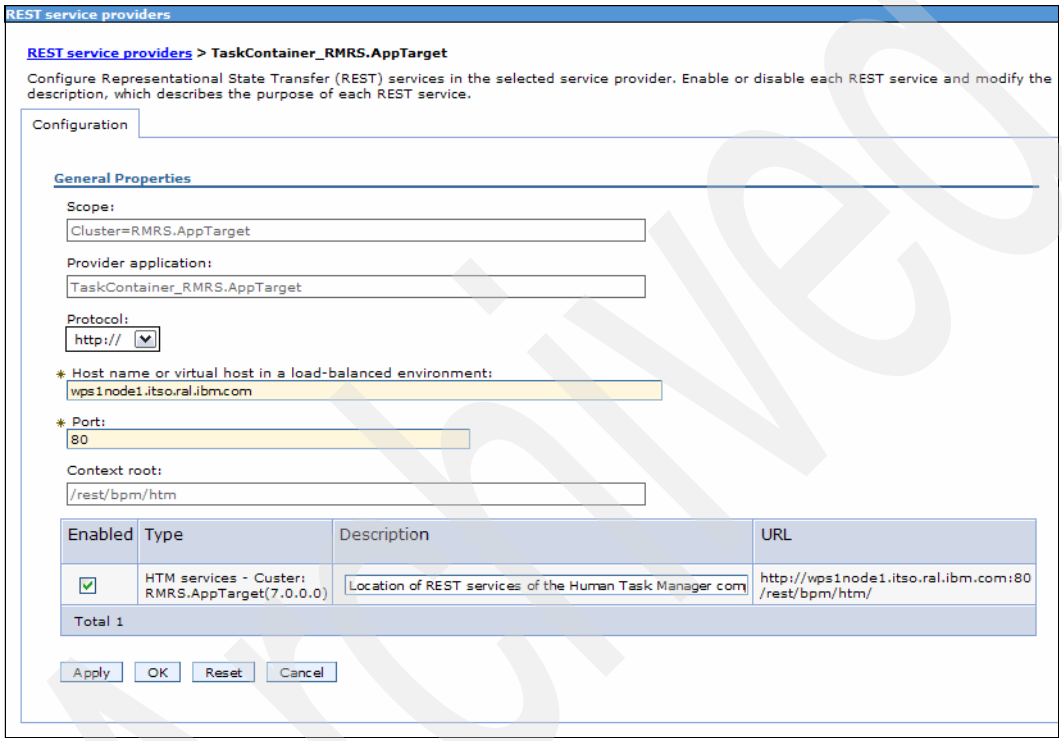


Figure 5-23 REST service endpoint for the human task manager

After verifying the REST service locations for the business flow manager and the human task manager from the REST service providers, confirm that the mashup endpoints are also synchronized with these locations.

8. In the left navigation area, click **Resources** → **Resource Environment** → **Resource Environment Providers**.
9. On the Resource environment providers panel, click **Mashups_Endpoints**.
10. On the Configuration tab, under Additional Properties, click **Custom properties**.

Archived

11. On the Custom properties panel, look for the resource of the business flow manager. At the bottom of each page, click the right arrow to page 6 of 11. Its resource name is `{com.ibm.bpm}BFM.url`. Its value must be the same as the one that you set from the REST service providers page, as shown in Figure 5-24.

☐	{com.ibm.bpm}BFM.name	
☐	{com.ibm.bpm}BFM.description	Location of REST services of the Business Flow Manager component of WebSphere Process Server
☐	{com.ibm.bpm}BFM.type	{com.ibm.bpm}BFM
☐	{com.ibm.bpm}BFM.version	7.0.0.0
☐	{com.ibm.bpm}BFM.url	http://proxyserver1:81/rest/bpm/bfm/

Figure 5-24 Mashups endpoint for the Business Flow Manager

12. If not, click **{com.ibm.bpm}BFM.url**.
13. Enter the value for the correct REST endpoint service for the business flow manager:

    `http://<hostname>:<port>/rest/bpm/bfm/`

14. Click **OK** and **Save** to save any changes.
15. On the Custom properties panel, look for the resource of the human task manager. At the bottom of the page, click the right arrow to page 7 of 11. Its resource name is `{com.ibm.bpm}HTM.url`. Its value must be the same as the one that you set from the REST service providers page, as shown in Figure 5-25.

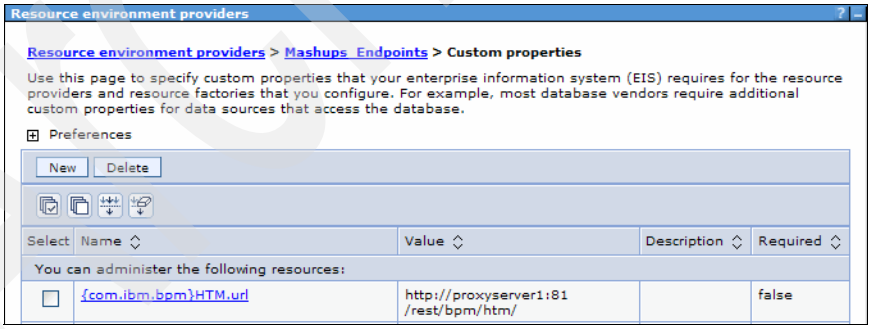


Figure 5-25 Mashups endpoint for the human task manager

16. If not, click **{com.ibm.bpm}HTM.url**.
17. Enter the value for the correct REST endpoint service for the human task manager:

    `http://<hostname>:<port>/rest/bpm/htm/.`

Archived

18. Click **OK** and **Save** to save any changes.

Note: If you modify the REST services gateway service provider endpoints, you must verify that the mashup endpoints are also synchronized with these locations.

5.5.4 Mapping modules to the Web server

During the installation of the IBM HTTP Server Plug-in, a `configureWebserver` script is produced by the install process on the Web server machine. The `configureWebserver` script automatically maps all Web application modules to the Web server. Therefore, run this script after the generation of the deployment environment. Otherwise, you must manually map these modules as follows:

1. Log in to the Integrated Solutions Console as an administrative user.
2. Click **Applications** → **Application Types** → **WebSphere enterprise applications**.
3. On the Enterprise Applications panel, click the name of the application.
4. On the Configuration tab, under Modules, click **Manage Modules**.
5. On the Manage Modules panel, perform the following steps:
   a. Select the check boxes next to the name of the modules.
   b. In the Clusters and servers section, select the application deployment target and the Web server (if any). Use the Ctrl key to select multiple targets. For example, to have a Web server serve your application, press the Ctrl key and then select the application server cluster and the Web server together.
   c. Click **Apply**.
   d. Click **OK** and **Save** to save any changes.

For each enterprise application that you want the Web server to serve, map all of its Web modules. Table 5-1 is a list of WebSphere Process Server enterprise applications. Each enterprise application may have one or more Web modules.

Table 5-1 WebSphere Process Server enterprise applications

Components	Enterprise applications
Business Space	BSpaceEAR BSpaceWebformsEnabler BusinessSpaceHelpEAR mm.was

Components	Enterprise applications
REST services	REST Services Gateway
Business flow manager	BPEContainer BPCExplorer
Human task manager	TaskContainer HTM_PredefinedTaskMsg_V700 HTM_PredefinedTasks_V700
Business rules	BusinessRulesManager
Widgets	BPMAdministrationWidgets HumanTaskManagementWidgets BusinessRules wesbWidget widgets_fabric
Remote Artifact Loader (RAL)	RemoteAL61

5.5.5 Configuring host aliases

To ensure that the deployment target clusters are visible to the deployment manager, verify that the host name alias, `default_host`, contains the host name and port number of every cluster member. If you are using a Web server or a proxy server, also include its host name and port number.

1. Log in to the Integrated Solutions Console as an administrative user.
2. Click **Servers** → **Server Types** → **WebSphere application servers**.
3. For every cluster member, click the name of the application server.
   a. Under Communications, expand **Ports**.
   b. For the port name, *WC_defaulthost*, remember its port number.
4. From the left navigation area of the Integrated Solutions Console, click **Environment** → **Virtual hosts**.
5. Click the **default_host** name.
6. Under Additional Properties, click **Host Aliases**.
7. Delete all entries where the host name is an asterisk (*).
8. If the host name and port number for the cluster members is not displayed on the list, click **New** to add the missing entry to the list.
9. If you add a new entry, click **Save** and **Synchronize**.

Archived

10. After mapping virtual hosts, regenerate the plug-in configuration file. From the left navigation area, click **Servers** → **Server Types** → **Web servers**.
11. From the Web servers panel, select the check box next to the name of the Web server for which you want to generate a plug-in. Click **Generate Plug-in**.
12. After the Plug-in configuration file generation is completed, select the check box next to the name of the Web server. Click **Propagate Plug-in**.

5.5.6 Verifying the Failed Event Manager

The WebSphere Process Server Recovery service captures data about failed events. The failed event manager allows you to view, modify, resubmit, or delete the failed event. To verify that the failed event manager is enabled, perform the following steps:

1. Log in to the Integrated Solutions Console as an administrative user.
2. Click **Integration Applications** → **Failed Event Manager**.
3. On the Failed Event Manager panel, under About your failed event manager, the following message is displayed

```
The Recovery sub-system is enabled.
Total failed events 0
```

5.5.7 Disabling CEI event persistence

In a production system, you may want to disable the persistence of CEI events to a database to improve performance:

1. Log in to the Integrated Solutions Console as an administrative user.
2. Click **Service integration** → **Common Event Infrastructure** → **Event service**.
3. Under Additional Properties, click **Event services**.
4. Click **Default Common Event Infrastructure event server**.
5. Uncheck the **Enable event data store** box.
6. Click **OK** and **Save** to save the configurations.

After disabling the event data store for event services, from the left navigation area, click **Integration Applications** → **Common Base Event Browser**. The Common Base Event client displays the following message

```
CWLCB0020E: The Common Event Infrastructure is unavailable.
```

This message is the result of disabling CEI event persistence.

5.5.8 (Optional) Enabling CEI event logging

To enable common event infrastructure logging to the Business Process Choreographer reporting function, perform the following steps:

1. Log in to the Integrated Solutions Console as an administrative user.
2. Click **Servers** → **Clusters** → **WebSphere application server clusters**.
3. Click the name of the application deployment target cluster (for example, RMRS.AppTarget).
4. On the Configuration tab, in the Business Integration section, expand **Business Process Choreographer** and click **Business Process Choreographer Containers**.
5. On the Configuration tab, scroll down to the middle of the page and expand the **State Observers** section.
6. In the State Observers section, for the business flow manager or the human task manager (Figure 5-26), click the **Common Event Infrastructure Logging** entry, as required by your applications.

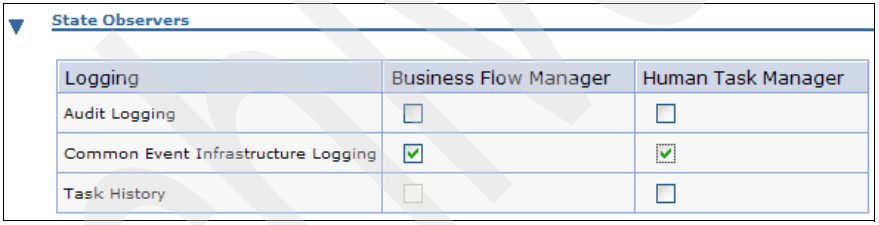


Figure 5-26 Common event infrastructure logging

7. Click **OK** and **Save** to save the changes.

Archived

5.5.9 Verifying the deployment environment

To ensure that the generation of the deployment environment has successfully completed, check the following items:

- On each node, monitor the `SystemOut.log` file of each node agent until there are no more output messages. You want to see success messages, as shown in Example 5-6.

Example 5-6 Node agent messages for generation of deployment environment

```
...
ADMA7021I: Distribution of application BPEContainer_RMRS.AppTarget
completed successfully.
ADMA7021I: Distribution of application BPCExplorer_RMRS.Support
completed successfully.
ADMA7021I: Distribution of application REST Services Gateway
completed successfully.
ADMA7021I: Distribution of application BSpaceEAR_RMRS.Support
completed successfully.
ADMA7021I: Distribution of application TaskContainer_RMRS.AppTarget
completed successfully.
...
ADMS0003I: The configuration synchronization completed successfully.
```

- On each node, the **serverStatus** command displays the new servers for each cluster, as shown in Example 5-7.

Example 5-7 Checking server status on a node

```
cd /opt/ibm/WebSphere/ProcServer/profiles/wps1node1/bin
./serverStatus.sh -all -user wpsadmin -password itso4you
ADMU0116I: Tool information is being logged in file
/opt/ibm/WebSphere/ProcServer/profiles/wps1node1/logs/serverStatus.l
og
ADMU0128I: Starting tool with the wps1node1 profile
ADMU0503I: Retrieving server status for all servers
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: nodeagent
ADMU0506I: Server name: RMRS.Support.wps1node1.0
ADMU0506I: Server name: RMRS.AppTarget.wps1node1.0
ADMU0506I: Server name: RMRS.Messaging.wps1node1.0
ADMU0508I: The Node Agent "nodeagent" is STARTED
ADMU0509I: The Application Server "RMRS.Support.wps1node1.0" cannot
be reached.
           It appears to be stopped.
```

Archived

```
ADMU0509I: The Application Server "RMRS.AppTarget.wps1node1.0"
cannot be
           reached. It appears to be stopped.
ADMU0509I: The Application Server "RMRS.Messaging.wps1node1.0"
cannot be
           reached. It appears to be stopped.
```

- On the deployment manager, restart the deployment manager, as shown in Example 5-8.

Example 5-8 Restarting the deployment manager

```
cd /opt/ibm/WebSphere/ProcServer/profiles/wps1dmgr/bin
./stopManager.sh -user wpsadmin -password itso4you
./startManager.sh
```

After restarting the deployment manager, perform the following steps:

1. Log in to the Integrated Solutions Console as an administrative user.
2. From the left navigation area, click **Servers** → **Clusters** → **WebSphere application server clusters**.
3. For each cluster, start the cluster in the order of messaging infrastructure, supporting infrastructure, and application deployment Target as follows:
   a. Select **RMRS.Messaging** and click **Start**. Wait for the messaging cluster to start, indicated by a green arrow.
   b. Select **RMRS.Support** and click **Start**. Wait for the support cluster to start, indicated by a green arrow.
   c. Select **RMRS.AppTarget** and click **Start**. Wait for the application deployment target cluster to start, indicated by a green arrow.

Archived

Figure 5-27 shows the started status for all clusters.

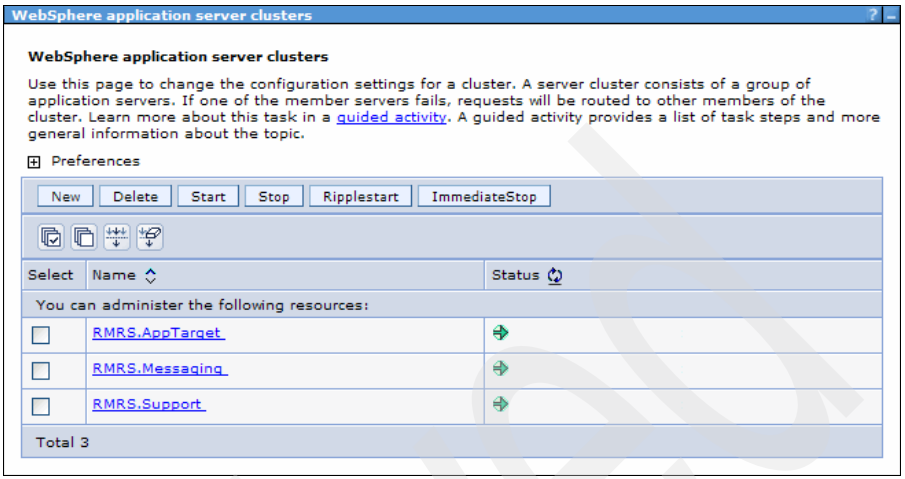


Figure 5-27 WebSphere application server clusters started

4. From the left navigation area, click **Service integration** → **Buses**.
5. For each bus, SCA system, SCA application, CEI, and BPC, perform the following steps:
   a. Click the name of the bus.
   b. Click **Local Topology**.
   c. Expand the bus topology to the messaging engine. The messaging engine is started, indicated by a green arrow, as shown in Figure 5-28.

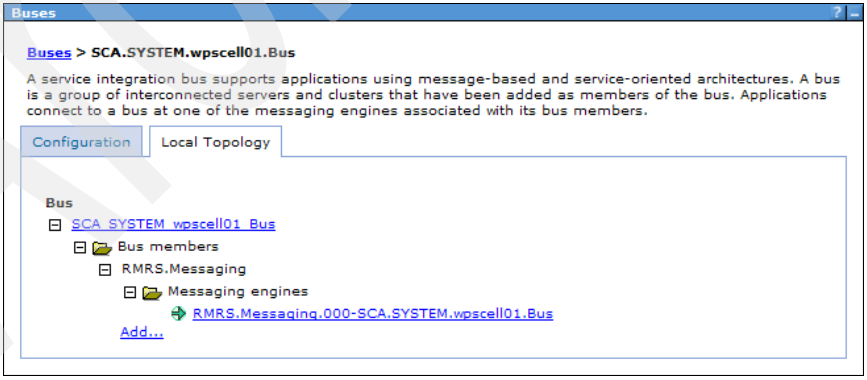


Figure 5-28 Messaging engine for the SCA system bus

On each node, for each member of a cluster, check the `SystemOut.log` and `SystemErr.log` files. Look for the line:

```
Server ... is open for e-business.
```

5.5.10 Verifying Business Process Choreographer

To verify that Business Process Choreographer is working correctly, install and run the Business Process Choreographer installation verification application (BPCIVT) provided with the product.

Installing the BPC installation verification application

In a deployment environment, you can install one instance of the Business Process Choreographer installation verification application. For example, if there are two Business Process Choreographer clusters in the deployment environment, you can install the `bpcivt.ear` application only to one of the clusters.

1. Log in to the Integrated Solutions Console as an administrative user.
2. Click **Applications** → **New Application**.
3. On the Install a New Application panel, click **New Enterprise Application**.
4. Enter the full path of the BPCIVT EAR file, located at `/opt/ibm/WebSphere/ProcServer/installableApps/bpcivt.ear`.
5. On the Preparing for the application installation panel, select **Fast Path - Prompt only when additional information is required**. Click **Next**.
6. On the Select installation options panel, accept the defaults. Click **Next**.
7. On the Map modules to servers panel, perform the following steps:
   a. Select the check box next to the BPCIVTApp module.
   b. In the Clusters and servers section, select **the application deployment target** and **the web server** (if any). Use the Ctrl key to select multiple targets. For example, to have a Web server serve your application, press the Ctrl key and then select the application server cluster and the Web server together.
   c. Click **Apply**.

Archives

Figure 5-29 shows the BPCIVTApp module mapped to the application deployment target and the Web server.

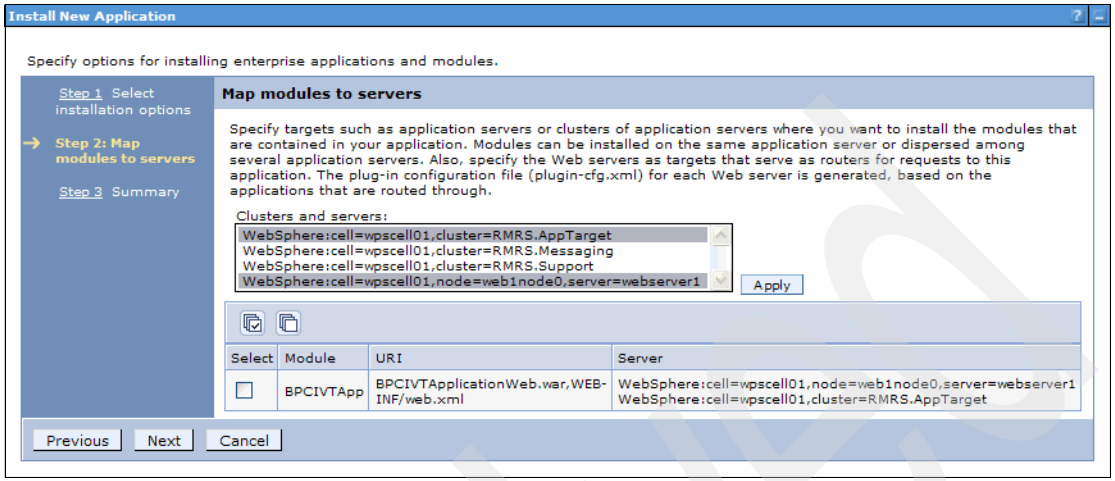


Figure 5-29 BPCIVTApp module mapped to servers

8. In the Summary panel, click **Finish**.
9. Click **OK** and **Save** to save to the master configuration.

> **Note:** After the BPCIVTApp enterprise application is installed, it is in the state *stopped*. But any process and task templates that it contains are in the state *started*. No process or task instances can be created until the application is started.

10. From the left navigation area, click **Applications** → **Application Types** → **WebSphere enterprise applications**.
11. From the Enterprise Application panel, select the **BPCIVTApp** application. Click **Start**.

Running the BPC installation verification application

To run the Business Process Choreographer installation verification application, using a Web browser, open the following page:

```
http://<application_server_hostname>:<port>/bpcivt
```

From the resulting success message we see that the basic functionality of the Business Process Choreographer configuration works, as shown in Example 5-9.

Example 5-9 BPCIVT success message

```
Looking up the HumanTaskManager API EJB...
Querying the originating task template...
Creating the originating task...
Creating the input message for the originating task...
Starting the originating task...
Querying the participating task...
Participating task not yet created, waiting 1s...
Participating task not yet created, waiting 1s...
Participating task not yet created, waiting 1s...
Participating task not yet created, waiting 1s...
Participating task not yet created, waiting 1s...
Participating task not yet created, waiting 1s...
Claiming the participating task...
Checking the input data...
Completing the participating task...
Checking the output data...
Output message of starter oTask not yet available, waiting 1s...
Deleting the originating task ...
Passed
Done.
```

5.5.11 Verifying Business Space

After the installation and configuration of Business Space powered by WebSphere for WebSphere Process Server, security is enabled for Business Space. By default, Business Space is configured to require both authentication and authorization of access.

Note: Although Business Space can be accessed by either HTTP or HTTPS, user authentication is always redirected to an HTTPS login page. Therefore, if a Web server (for example, IBM HTTP Server) is used to access Business Space URLs, the Web server must be configured to support HTTPS

1. Start the Business Space. By using a Web browser, open the following page:

   `http://<hostname>:<port>/BusinessSpace`

2. Log in to the Business Space.
3. On the Welcome to Business Space page, click **Manage Spaces**.

4. On On Space Manager window, click **Problem Determination**.
5. From the Problem Determination page, click **System Health**.
6. On the Topology tab, you can view the status of your deployment environment, as shown in Figure 5-30 on page 193. The system health widget examines the health of the overall system, including system applications, system data sources, system messaging engines, and queues.

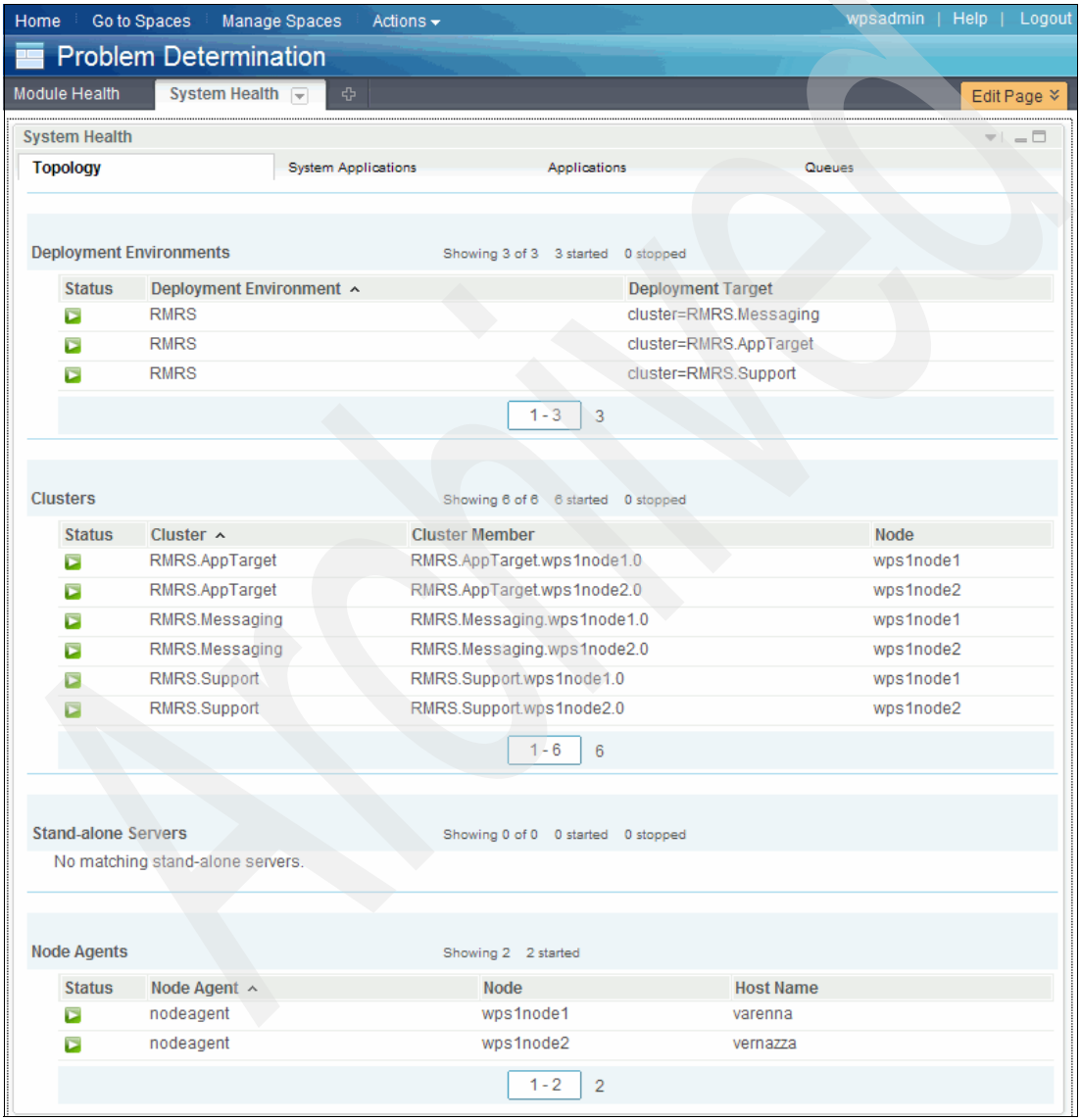


Figure 5-30 Business Space: system health widget

5.5.12 Verifying human interaction using Business Space

The following sample application, from the IBM WebSphere Business Process Management Samples and Tutorials, is a long-running business process requiring a human to enter a string into a data field. After the data is provided, you browse the output message when the process is finished.

The human interaction and Business Space sample application is modified to enable the HTML Dojo forms, which are generated from WebSphere Integration Developer, to work with human task management widgets in Business Space, where Business Space is installed on a different server instance from the Business Process Choreographer.

In this deployment environment pattern, Remote Messaging and Remote Support, the Business Process Choreographer is configured on the application deployment target, and Business Space is configured on the support target.

For this sample application, the user interfaces for the human task activity are HTML Dojo forms enabled for Business Space. The generated HTML files are deployed to a separate Web project (`AskQuestion`) from the business integration project (`HumanInteractionApp`). The separation of the user interface Web project and the business integration project allows the Web module containing the HTML files to be deployed to the support target, where Business Space is running.

Refer to 5.5.10, “Verifying Business Process Choreographer” on page 190, to install the `HumanInteractionApp` and `AskQuestion` enterprise applications, included with this publication.

1. Install the `HumanInteractionApp` enterprise application and map its modules to the application deployment target and the Web server.
2. Install the `AskQuestion` enterprise application and map its modules to the application deployment target, the support target, and the Web server.
3. Start both the `HumanInteractionApp` and the `AskQuestion` enterprise application.

The following steps describe how you can run this sample by using Business Space for human workflow:

1. Start the Business Space. By using a Web browser, open the following page:

   `http://<hostname>:<port>/BusinessSpace`
2. Log in to the Business Space.

3. If you have not worked with human tasks in this Business Space, create a new Business Space. From the Welcome to Business Space page, click **Manage Spaces**, as shown in Figure 5-31.

Figure 5-31 Manage Spaces

4. From the Space Manager window, click **Create Space**, as shown in Figure 5-32.

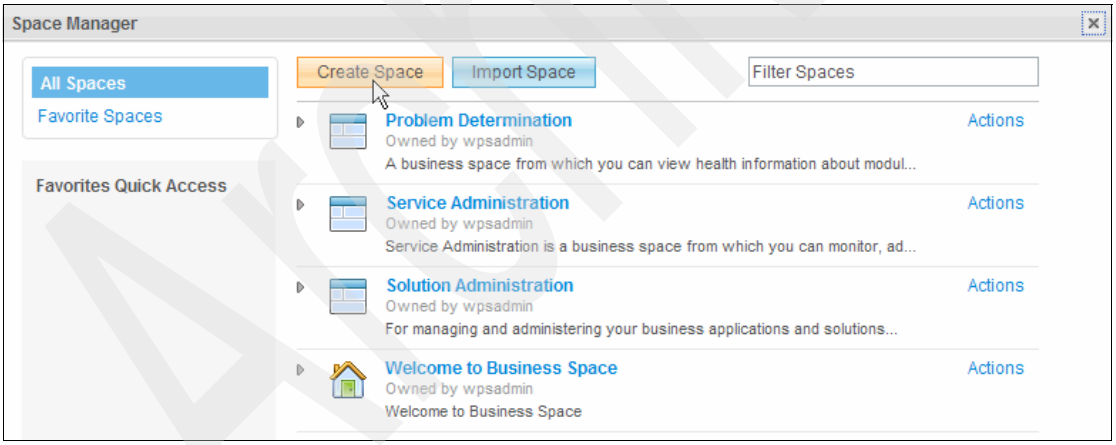


Figure 5-32 Create Space

Archived

5. From the Create Space window (Figure 5-33), enter a name in the Space Name field to identify your Business Space page. Select **Create a new space using a template**. Click the down arrow and select **Advanced Managing of Human Tasks and Workflows**. Click **Save**.

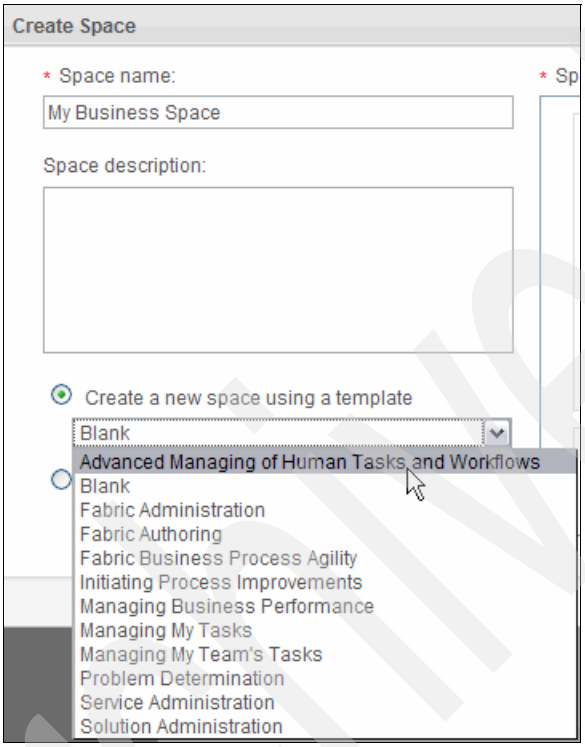


Figure 5-33 Create a new space using a template

6. From the Space Manager window, click the name of your new Business Space.
7. From your Business Space, click the **Manage Human Tasks** tab.

Archived

8. From the Manage Human Tasks page, in the Task Definitions List widget, select the check box next to Start the HumanInteractionProcess business process. Click the icon in the upper right corner to create an instance from this task definition, as shown in Figure 5-34.

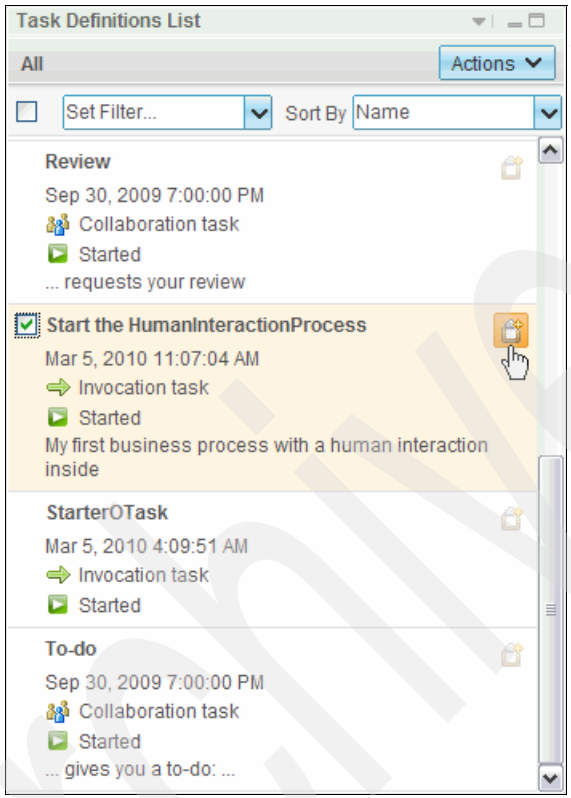


Figure 5-34 Task Definitions List

Archived

9. In the Task Information widget, the HTML Dojo form for entering the input data is displayed. Enter a string value in the input field. Click **Submit**, as shown in Figure 5-35.

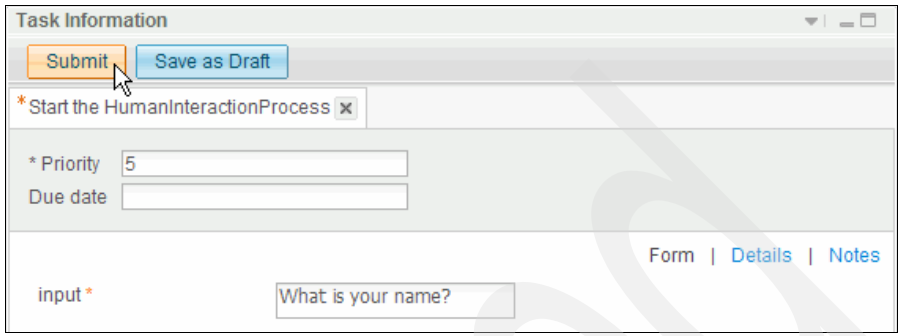


Figure 5-35 Task Information

The HumanInteractionProcess business process is started and the AskQuestion human task activity is available.

10. To start working with human tasks, click **Work on Tasks**.

11. From the Task List widget, select the check box next to the AskQuestion task and click the icon in the upper right corner to accept and edit this task, as shown in Figure 5-36.

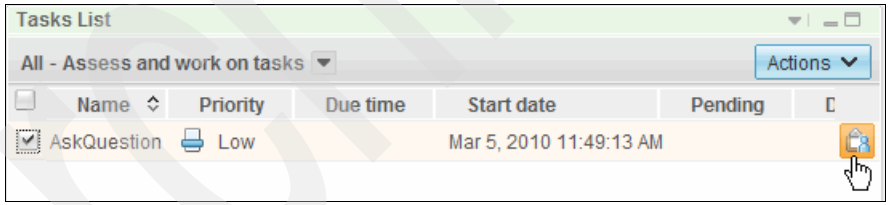


Figure 5-36 Tasks List

Archived

12. From the task information widget, the HTML Dojo form for processing the task is displayed. The output field (`question`) is populated with the input data at the start of the business process. This input data is passed from the business process instance to the task instance. Enter a string value in the answer field. Click **Submit**, as shown in Figure 5-37.

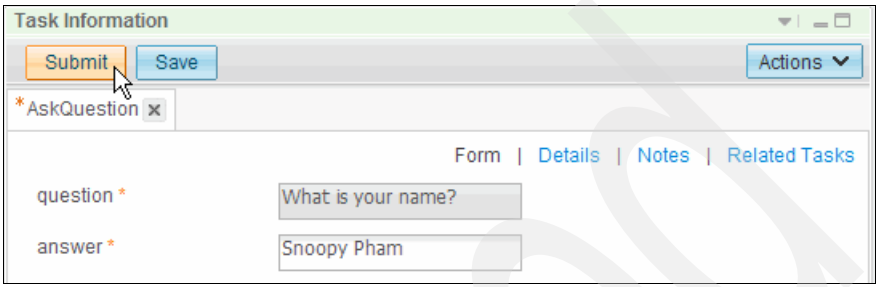


Figure 5-37 Task Information

Figure 5-38 shows the business process in the human workflow diagram widget, since the task was claimed by a privileged user who can view the process template details.

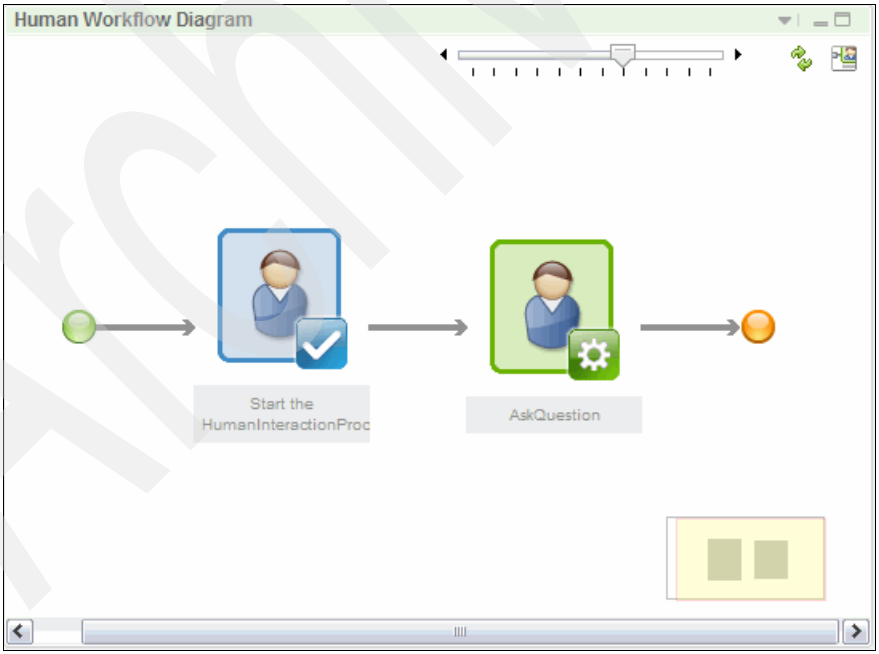


Figure 5-38 Human Workflow Diagram

Archived

Note: A regular user can still complete the human task, but she is not able to view the diagram.

The task is now completed and is removed from the task list widget. Since the human interaction process is a long-running business process, you can still view the output message.

13. Click **Create Tasks**.
14. From the tasks list widget, select the check box next to the most recent process instance with the name Start the HumanInteractionProcess. Click the icon in the upper right corner to open the task as shown in Figure 5-39.

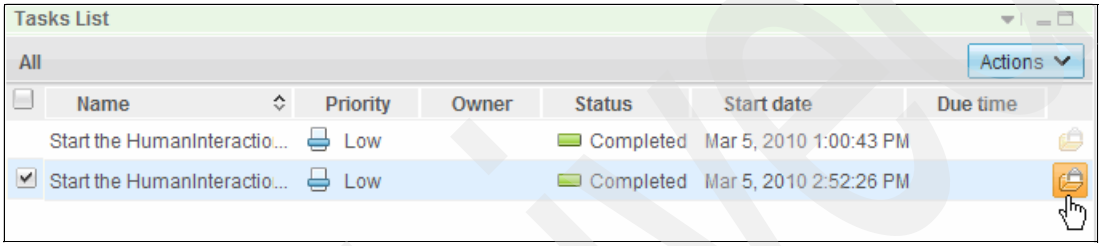


Figure 5-39 Tasks List: Completed

In the task information widget, both the input and output of the HumanInteractionProcess business process are displayed, as shown in Figure 5-40.

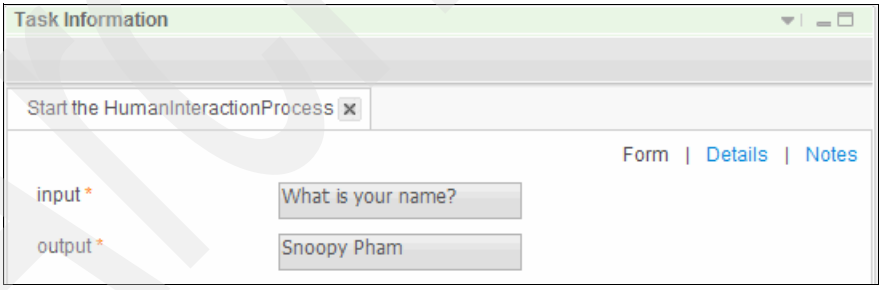


Figure 5-40 Task Information: Completed

Archive

The workflow is complete (Figure 5-41).

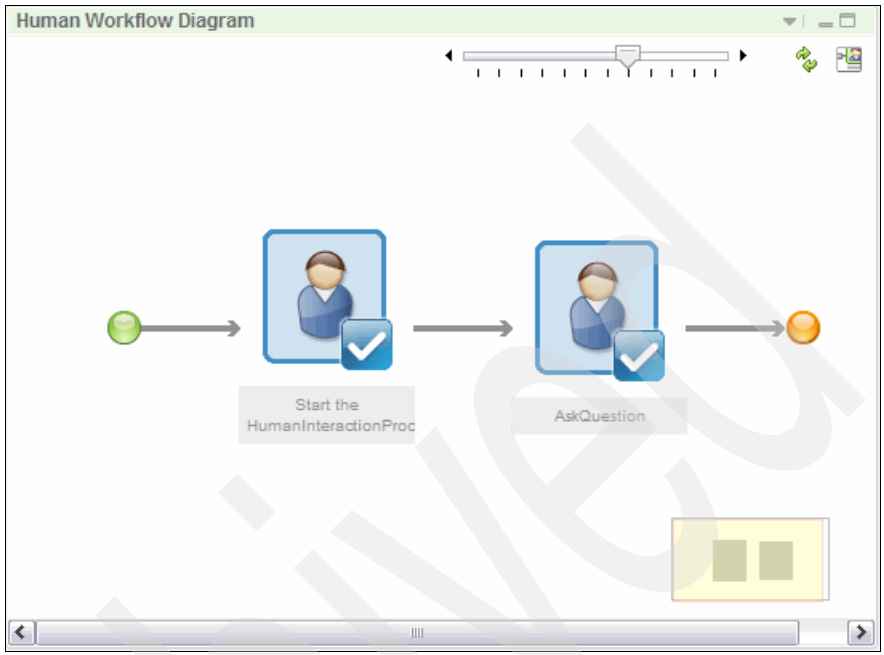


Figure 5-41 Human Workflow Diagram: Completed

Archived

Archived

6

WebSphere Enterprise Service Bus: Remote Messaging and Remote Support

This chapter briefly discusses the WebSphere Enterprise Service Bus product, its relationship to WebSphere Process Server, and how a WebSphere Enterprise Service Bus Remote Messaging and Remote Support topology could be created based on the pattern described in this book for creating a WebSphere Process Server Remote Messaging and Remote Support topology.

© Copyright IBM Corp. 2010. All rights reserved.

6.1 Putting WebSphere Enterprise Service Bus in context

WebSphere Enterprise Service Bus is part of the WebSphere SOA family of products. It provides service-oriented architecture (SOA) connectivity services—the ability to connect disparate systems via mediation modules. These connect systems by providing services and invoking other services. Mediation modules contain mediation flows, which define how these services are connected. The mediation flows are constructed of mediation primitives, which are re-usable chunks of functionality, typically supplied with WebSphere Enterprise Service Bus.

Structurally and functionally, WebSphere Enterprise Service Bus has a subset of the functionality available in WebSphere Process Server. In other words, WebSphere Process Server includes all the functionality that is available in WebSphere Enterprise Service Bus and more.

Figure 6-1indicates the components that are found in the respective layers of the *stack* of software.

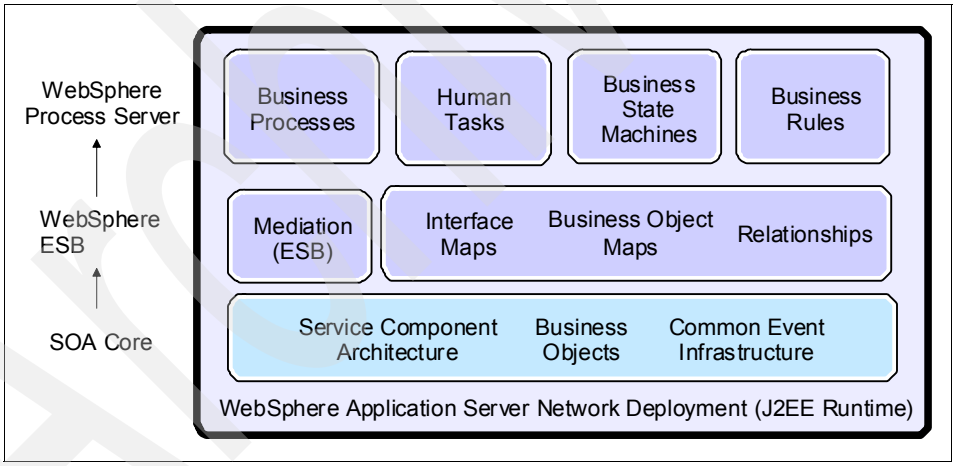


Figure 6-1 Stack of products including WebSphere Enterprise Service Bus and WebSphere Process Server

Archive

Therefore, significant components that WebSphere Enterprise Service Bus does not include, that WebSphere Process Server does, are:

- *Business processes* provided by the Business Process Choreographer component of WebSphere Process Server
- *Human tasks* provided by the human task manager component of WebSphere Process Server
- *Business state machines* provided by the Business Process Choreographer component of WebSphere Process Server
- *Business rules* provided by the business rules engine of WebSphere Process Server

There are no components that are unique to WebSphere Enterprise Service Bus that are not also found in WebSphere Process Server.

6.2 Overview of topology creation

WebSphere Enterprise Service Bus is a functionality subset of WebSphere Process Server, and so creating a WebSphere Enterprise Service Bus Production Topology is essentially equivalent to creating one for WebSphere Process Server, except that some steps or items are omitted.

In Table 6-1 we summarize how to create an Remote Messaging and Remote Support topology for WebSphere Enterprise Service Bus. Because creating a topology for WebSphere Enterprise Service Bus is so similar to creating one for WebSphere Process Server, you can use Chapter 6, “WebSphere Enterprise Service Bus: Remote Messaging and Remote Support” on page 203, as the basis for this. Table 6-1 lists the steps that are different for WebSphere Enterprise Service Bus, as well as where in the WebSphere Process Server chapter they are referenced.

Table 6-1 Differences between WebSphere Process Server and WebSphere ESB

Section	Difference
3.2.3, “Installing WebSphere Process Server” on page 70	Rather than installing the WebSphere Process Server product, install the WebSphere Enterprise Service Bus product. The install technology is identical.

Archived

Section	Difference
5.2, “Prerequisites for creating the topology” on page 151	When creating the DDL to define databases with using the database design tool, pick a WebSphere Enterprise Service Bus topology rather than a WebSphere Process Server one. The tool then does not generate DDL to define the BPC (BPEDB) or BPC Event Collector (OBSVRDB) databases.
3.4, “Creating a deployment manager profile” on page 91, and 3.5, “Creating custom profiles” on page 107	Instead of creating a WebSphere Process Server deployment manager and custom node types, you create WebSphere Enterprise Service Bus deployment manager and custom node types.
5.4.1, “Create a new deployment environment” on page 166	There is one panel that does not appear in the Create deployment environment wizard: BPC configuration (step 7).

Archive

7

WebSphere Process Server: Double Remote Messaging and Remote Support

This chapter provides the steps to create a Double Remote Messaging and Remote Support topology. In this topology we have:

- A standard Remote Messaging and Remote Support topology
- A second application target cluster with a new Business Process Choreographer (BPC) configuration
- A second messaging cluster
- A shared support cluster and Common Event Infrastructure (CEI)
- Additional databases for the new clusters
- Reconfigured messaging connections
- New core group policies for effective resource utilization

© Copyright IBM Corp. 2010. All rights reserved.

7.1 Topology summary

Figure 7-1 shows the Double Remote Messaging and Remote Support topology, which we are going to build. We extend the standard Remote Messaging and Remote Support topology with an additional application cluster (RMRS.AppTarget2) and an additional messaging cluster (RMRS.Messaging2). The new messaging cluster members hold new messaging engines for SCA.SYSTEM, SCA.APPLICATION, and BPC buses. The active messaging engines are distributed among cluster members for better resource utilization. Their counterparts are waiting in standby mode in the other cluster members of the same messaging cluster.

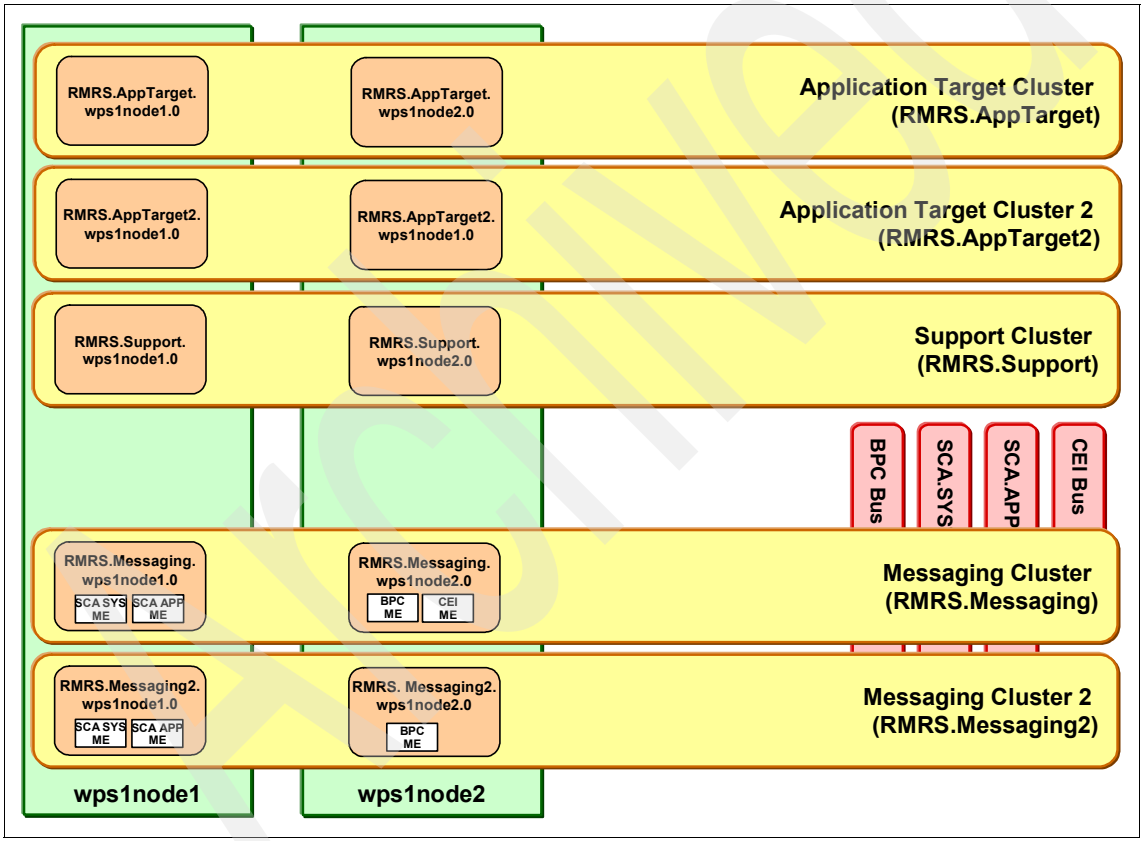


Figure 7-1 Double Remote Messaging and Remote Support topology

7.2 Prerequisites for creating this topology

This chapter explains the steps required to extend a standard Remote Messaging and Remote Support topology to Double Remote Messaging and Remote Support. For this reason, all the prerequisites described for the Remote Messaging and Remote Support topology apply here as well. This section provides an overview of the Remote Messaging and Remote Support environment, which is the starting point to build Double Remote Messaging and Remote Support. In this section, we also detail the required steps to configure additional databases for the extended environment.

7.2.1 Existing Remote Messaging and Remote Support topology

To configure a Double Remote Messaging and Remote Support topology we expand the Remote Messaging and Remote Support topology. Using the steps described in Chapter 5, "WebSphere Process Server: Remote Messaging and Remote Support" on page 149, we start with the following environment:

- Three machines hosting the deployment environment:
  - Deployment manager (wps1dmgr)
  - Node 1 (wps1node1)
  - Node 2 (wps1node2)
- Three clusters defined according to the Remote Messaging and Remote Support topology:
  - Application target (RMRS.AppTarget)
  - Messaging target (RMRS.Messaging)
  - Support target (RMRS.Support)
- The node wps1node1 containing the following cluster members:
  - An application target cluster member: RMRS.AppTarget.wps1node1.0
  - A support target cluster member: RMRS.Support.wps1node1.0
  - A messaging target cluster member: RMRS.Messaging.wps1node1.0
- The node wps1node2 containing the following cluster members:
  - An application target cluster member: RMRS.AppTarget.wps1node2.0
  - A support target cluster member: RMRS.Support.wps1node2.0
  - A messaging target cluster member: RMRS.Messaging.wps1node2.0

7.2.2 Creating an additional BPE database

The Business Process Choreographer is always configured to one specific deployment target, which is the application target server or cluster. We already

set up this cluster in our Remote Messaging and Remote Support environment. In the Double Remote Messaging and Remote Support topology we extend the topology with an additional application cluster.

Two BPC configurations cannot share the same Business Process Engine (BPE) database with the same schema. Either you must create a separate BPE database or add an additional schema to the existing database for the new cluster to hold a new set of BPE tables. When in doubt, consult your database administrator (DBA) to learn which way is appropriate in your environment.

When creating the second database (BPEDB2), we can reuse the scripts generated for the Remote Messaging and Remote Support topology:

1. Make sure that the following sql files generated by the database design tool (3.3 “Database and schema creation” on page 82) are copied to the DB2 database machine locally:
   - WBI_BPC_DB2-distributed-BPC/createTablespace.sql
   - WBI_BPC_DB2-distributed-BPC/createSchema.sql
2. Create a new database BPEDB2 and connect to it:

```
db2 "CREATE DATABASE BPEDB2 USING CODESET UTF-8 TERRITORY EN-US
PAGESIZE 8 K"
db2 "connect to BPEDB2"
```

3. Create the BPC schema and tables:

```
db2 "CREATE SCHEMA BPC"
db2 -tf WBI_BPC_DB2-distributed-BPC/createTablespace.sql
db2 -tf WBI_BPC_DB2-distributed-BPC/createSchema.sql
```

4. Verify that the tables are successfully generated (173 tables should be listed):

```
db2 "list tables for schema BPC"
```

5. Close the connection to the database:

```
db2 connect reset
```

7.2.3 Creating an additional messaging engine database

In the Double Remote Messaging and Remote Support topology, we extend the Remote Messaging and Remote Support with an additional Messaging cluster as well. Similarly to the BPEDB, we can decide whether we add a new schema to the existing messaging engine database (MEDB) or create a completely new MEDB for the second set of messaging engines. When in doubt, consult your DBA to see which way would be appropriate in your environment.

For creating the second database (MEDB2), we can reuse the scripts generated for the Remote Messaging and Remote Support topology:

1. Make sure that the following sql files generated by the database design tool (3.3 “Database and schema creation” on page 82) are copied to the DB2 database machine locally:
   - WBI_SCA_APP_ME_DB2-distributed-SibME/DB2-distributed-SibME.sql
   - WBI_SCA_SYS_ME_DB2-distributed-SibME/DB2-distributed-SibME.sql
   - WBI_BPC_ME_DB2-distributed-SibME/DB2-distributed-SibME.sql
2. Create a new database MEDB2 and connect to it:

```
db2 "CREATE DATABASE MEDB2 USING CODESET UTF-8 TERRITORY EN-US
PAGESIZE 8 K"
db2 "connect to MEDB2"
```

3. Create the BPC schema and tables:

```
db2 -tf
WBI_SCA_APP_ME_DB2-distributed-SibME/DB2-distributed-SibME.sql
db2 -tf
WBI_SCA_SYS_ME_DB2-distributed-SibME/DB2-distributed-SibME.sql
db2 -tf WBI_BPC_ME_DB2-distributed-SibME/DB2-distributed-SibME.sql
```

   Note: Ignore the following or similar warning messages:

```
SQL0598W  Existing index "SCAAPP.SIB0O2PKIX" is used as the index
for the primary key or a unique key.  SQLSTATE=01550.
```

4. Verify that the tables are successfully generated (nine tables each):

```
db2 "list tables for schema SCAAPP"
db2 "list tables for schema SCASYS"
db2 "list tables for schema BPCME"
```

5. Close the connection to the database:

```
db2 connect reset
```

7.3 Configuring the topology using deployment environments in the Integrated Solutions Console

The most common Double Remote Messaging and Remote Support pattern is when the support cluster is reused by the new application target cluster. It means that the two application clusters share a central Common Event Infrastructure. The decision to add an additional support cluster depends on your business requirements. For example, if a high amount of events are emitted or isolation of

the events arriving from different application clusters is an important aspect, you may consider creating an additional support cluster. However, since the most common case is using a shared support cluster, we focus here on building a Double Remote Messaging and Remote Support pattern that is based on Remote Messaging and Remote Support and contains only an additional application and a corresponding messaging cluster.

Note: This developerWorks® article describes common and discouraged patterns when creating additional clusters:

http://www.ibm.com/developerworks/websphere/library/techarticles/0901_herness/0901_herness.html

7.3.1 Adding the new application and messaging clusters

To do this:

1. To create a new cluster, log in to the Integrated Solutions Console, navigate to **Servers** → **Clusters,** and choose **WebSphere application server clusters** (Figure 7-2).

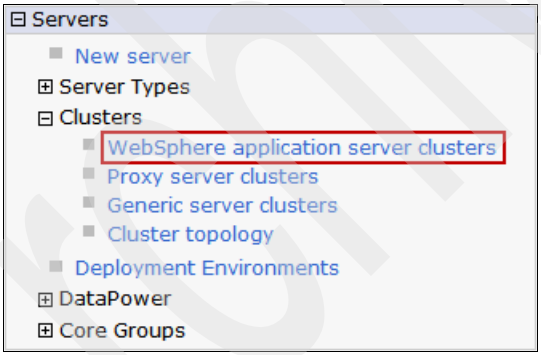


Figure 7-2 Navigate to WebSphere application server clusters

2. In the WebSphere application server clusters window click **New**.

Archived

3. Enter `RMRS.AppTarget2` in the cluster name field (Figure 7-3). Leave the other settings as the defaults and click **Next**.

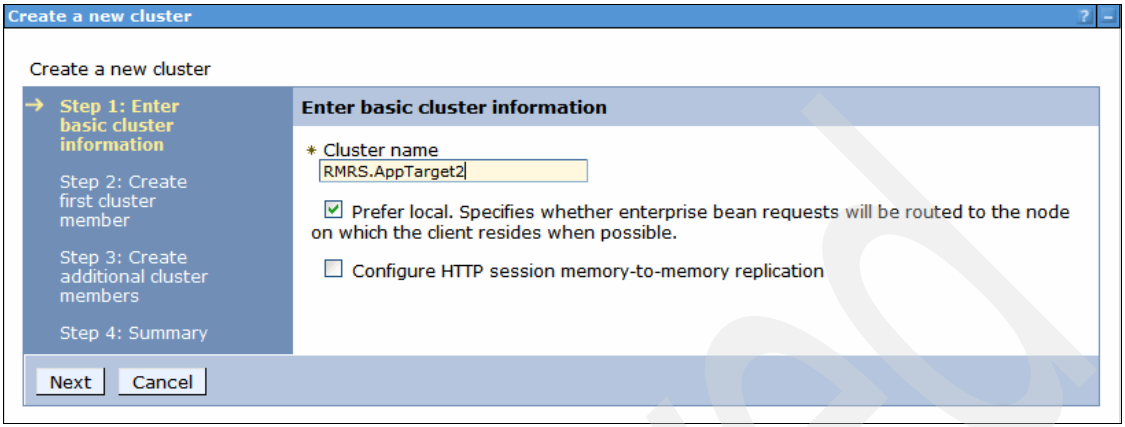


Figure 7-3 Enter cluster name

Archived

4. In the Create first cluster member window, enter the member name `RMRS.AppTarget2.wps1node1.0` and leave the other fields as the defaults. This configures the first cluster member to be generated using the defaultProcessServer template on the first node, as shown in Figure 7-4. Click **Next** to continue.

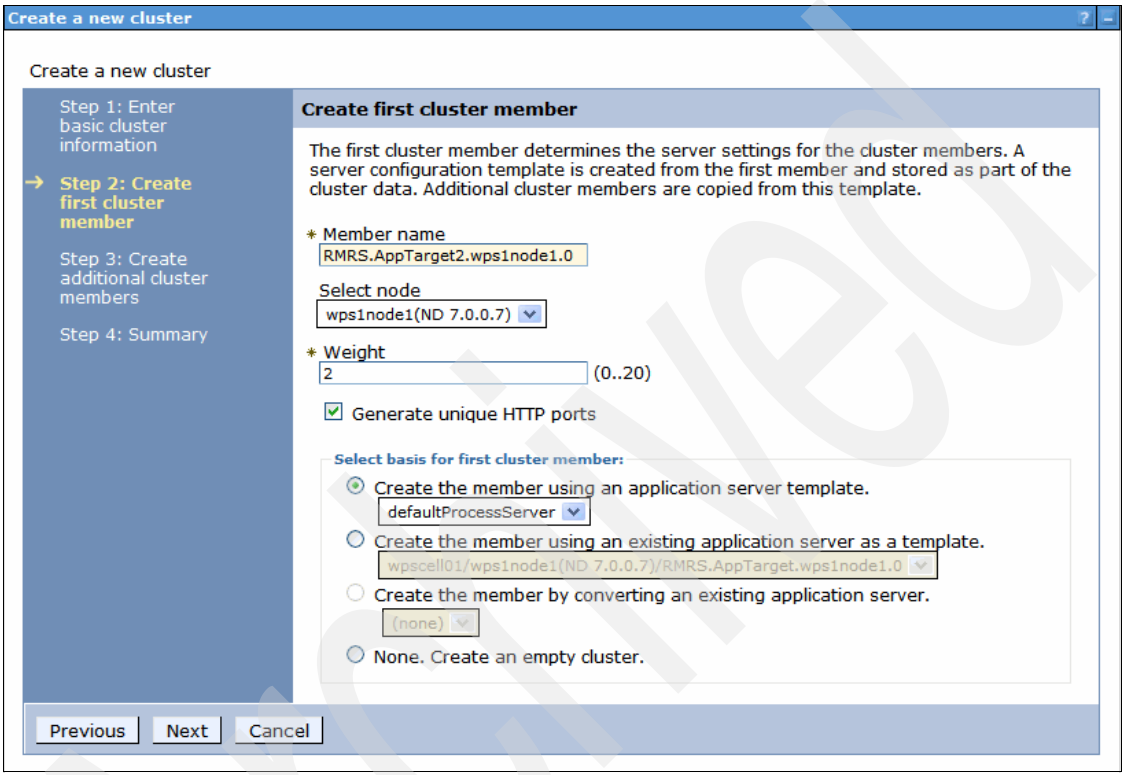


Figure 7-4 Create first cluster member

Archived

5. Enter `RMRS.AppTarget2.wps1node2.0` as the additional cluster member name in accordance with the naming conventions applied for the Remote Messaging and Remote Support topology. Select the **wps1node2** node from the drop-down menu (Figure 7-5).

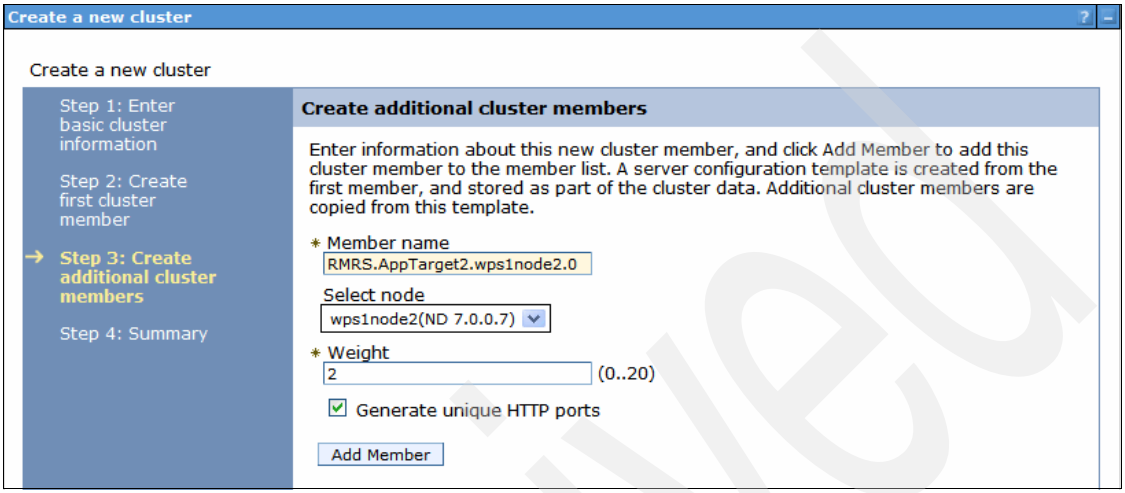


Figure 7-5 Additional cluster members

6. Click **Add Member**, which adds this additional member to the members list at the bottom of the window (Figure 7-6), then click **Next**.

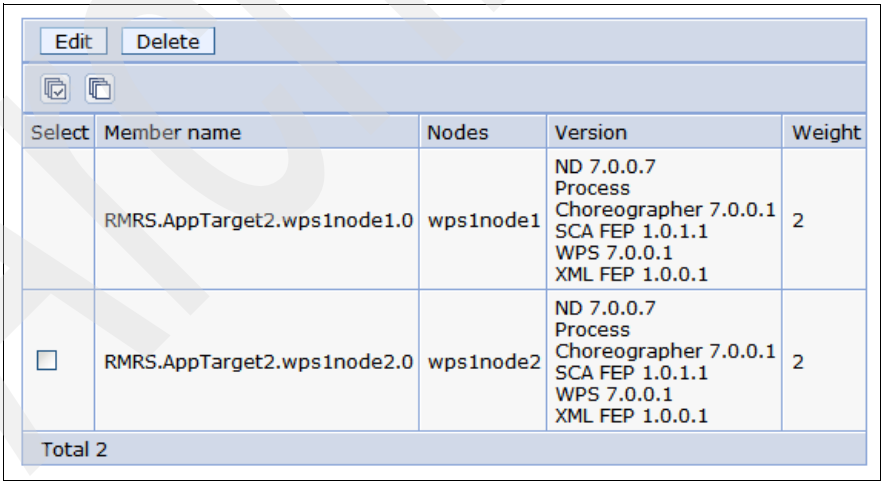


Figure 7-6 Cluster member list

7. Review the Summary window and verify your settings, then click **Finish**.

Archived

8. Repeat this process for the new messaging cluster RMRS.Messaging2 with cluster members:
   - RMRS.Messaging2.wps1node1.0
   - RMRS.Messaging2.wps1node2.0
9. Save your configuration.

7.3.2 Configuring Service Component Architecture

The next step is to configure Service Component Architecture (SCA) support.

Configuring SCA for the second messaging cluster

To do this:

1. Log in to the Integrated Solutions Console, expand **Servers** → **Clusters**, and click **WebSphere application server clusters**
2. Click the **RMRS.Messaging2** link (Figure 7-7).

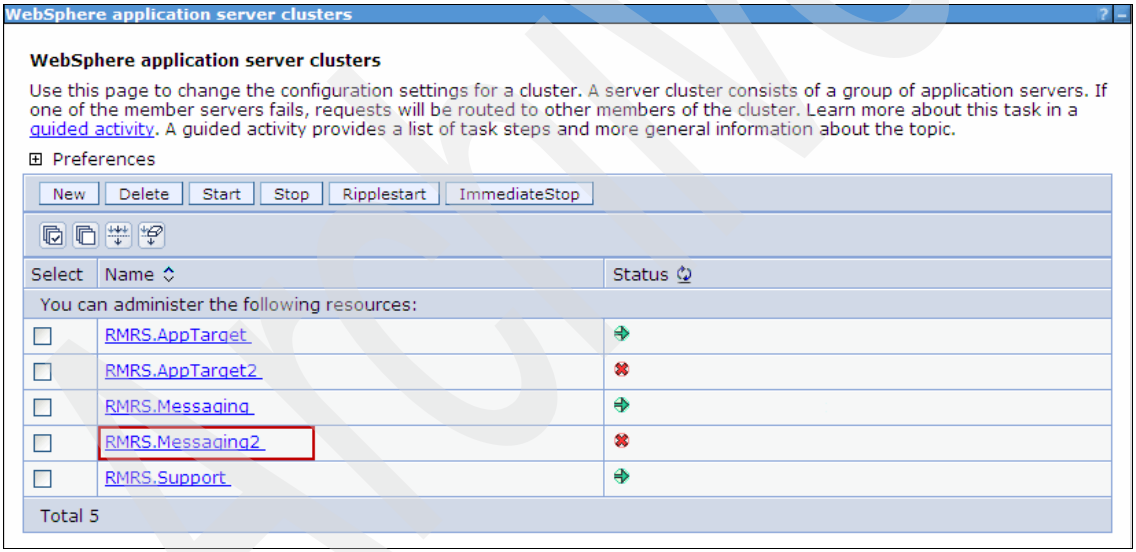


Figure 7-7 Clusters

Archived

3. On the Configuration page, under the Business Integration heading, click the **Service Component Architecture** link (Figure 7-8).

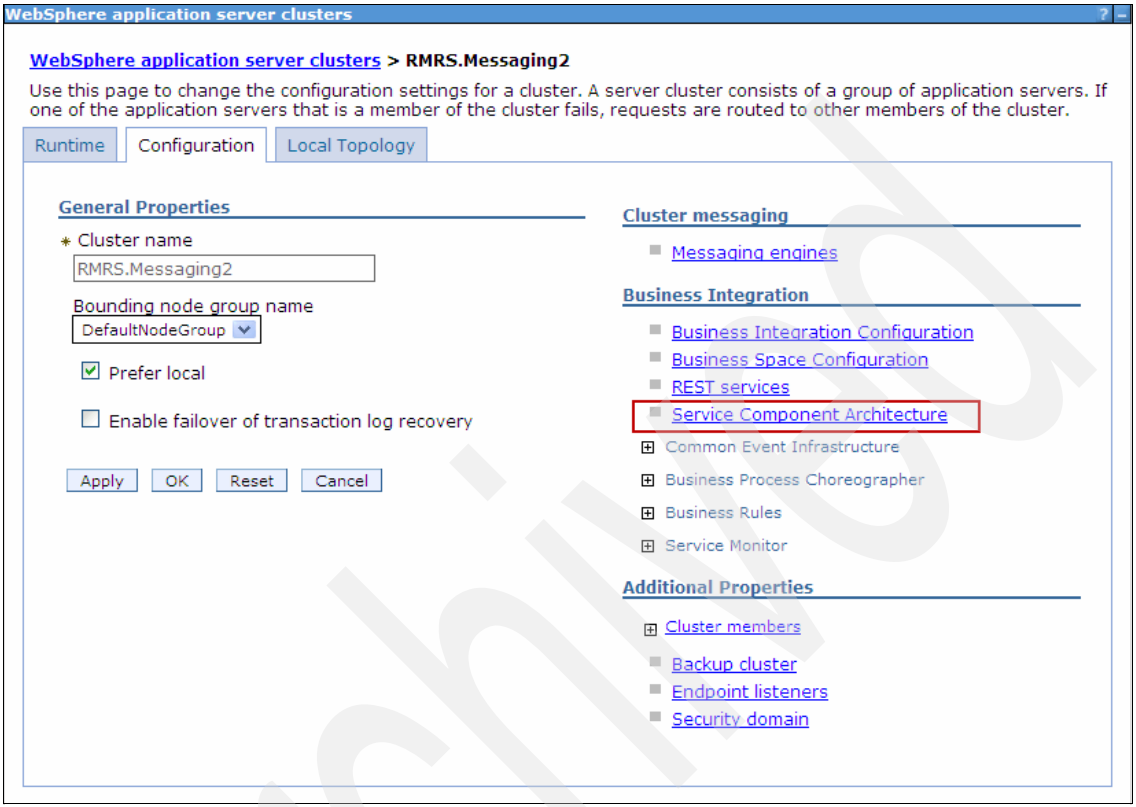


Figure 7-8 Service Component Architecture configuration

Archive

4. Select the **Support the Service Component Architecture components** check box and click the **Local** radio button for Bus Member Location (Figure 7-9).

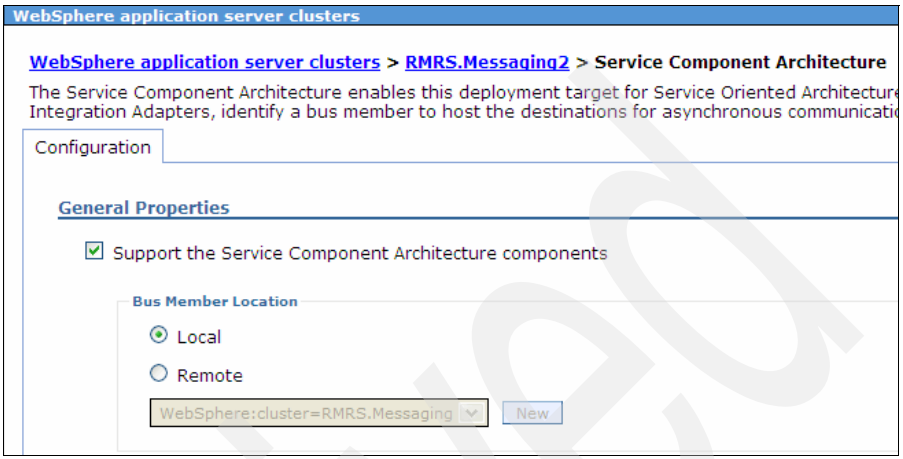


Figure 7-9 SCA properties

Archived

5. At the bottom of the page, fill out the database-related fields (Figure 7-10).
   - For both the system bus member and the application bus member apply the following settings:
     - Database Name: `MEDB2`
     - Uncheck the Create Tables check box, as we have already generated the tables by using the output of the database design tool.
     - Fill out the User name, Password, and Server name fields according to your database settings.
     - Leave the Provider drop-down menu at DB2 Universal.
   - For the System Bus Member section apply the following setting: Schema: `SCASYS`
   - For the Application Bus Member section apply the following setting: Schema: `SCAAPP`

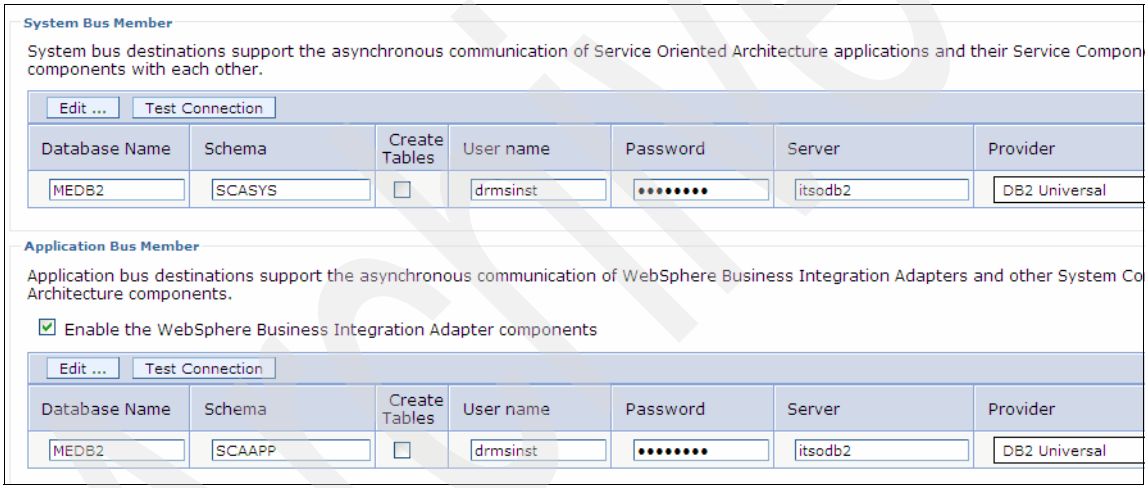


Figure 7-10 System and Application Bus Member properties

6. Click **OK** to finish the configuration.

Archived

7. Repeat this process for the application cluster RMRS.AppTarget2. Ensure that the bus member location is set to Remote on the RMRS.Messaging2 cluster, as shown in Figure 7-11.

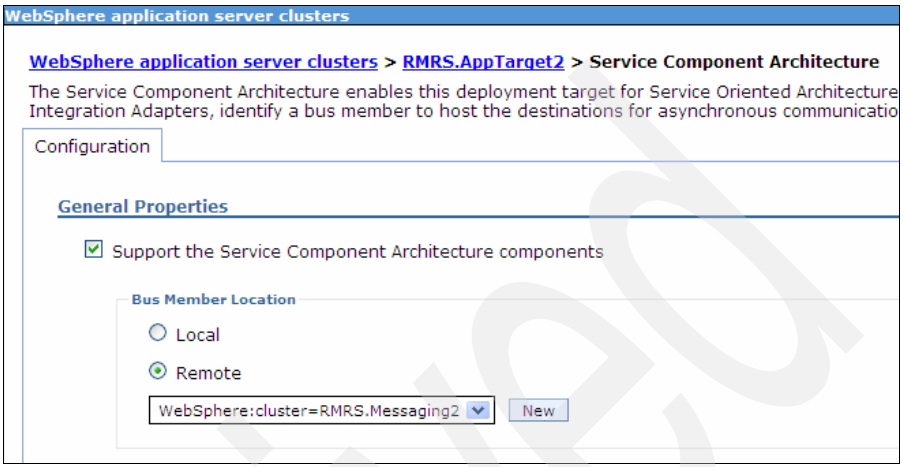


Figure 7-11 Setting SCA support and remote bus member location

8. Save your configuration.

Archived

7.3.3 Configuring Business Process Choreographer

In this section we configure BPC support for the second application cluster. This allows us to run applications containing business processes and human tasks on the new application cluster.

1. Log in to the Integrated Solutions Console, navigate to **Servers** → **Clusters**, and click **WebSphere application server clusters**.
2. Click the **RMRS.AppTarget2** link (Figure 7-12).

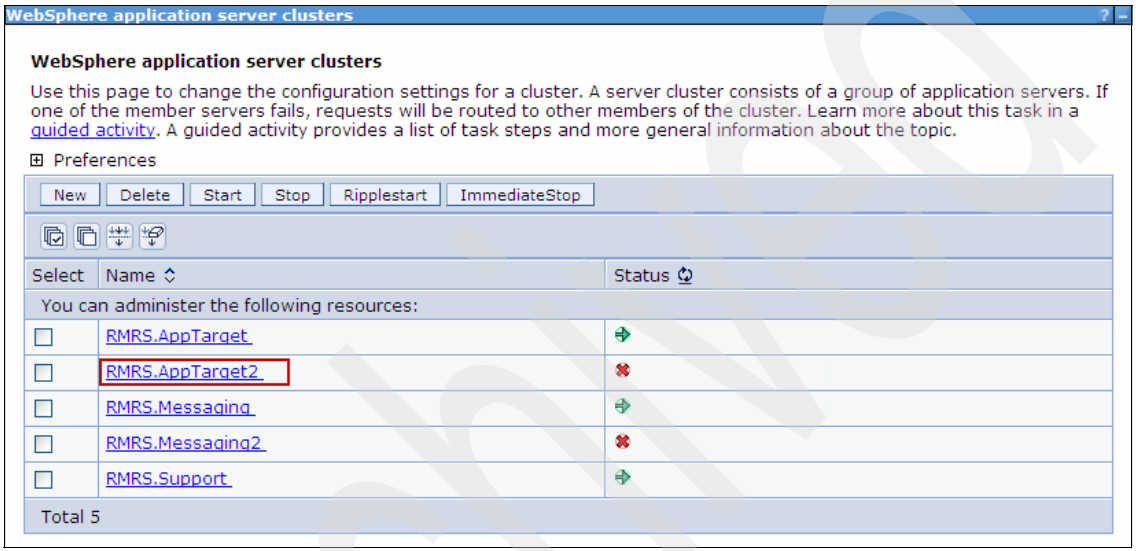


Figure 7-12 Clusters

Archived

3. On the Configuration page, under the Business Integration heading, expand **Business Process Choreographer** and click the **Business Process Choreographer Containers** link (Figure 7-13).

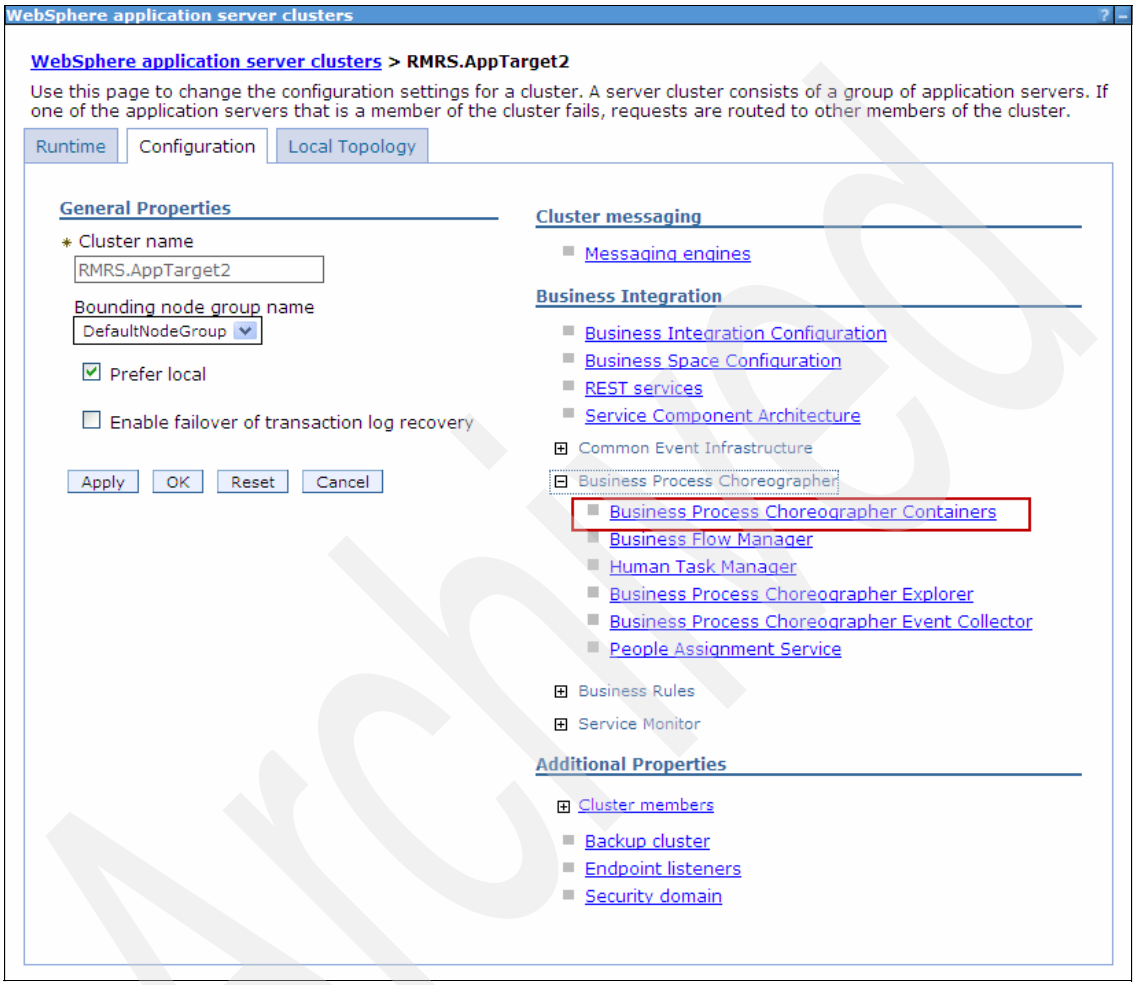


Figure 7-13 Business Process Choreographer Containers

Archived

4. In the Data Source section, type `BPEDB2` for the database name and `BPC` for the schema name. Uncheck the Create Tables check box. Fill out the User Name, Password, and Server fields according to your database settings. Leave the Provider drop-down menu selection as DB2 Universal (Figure 7-14).

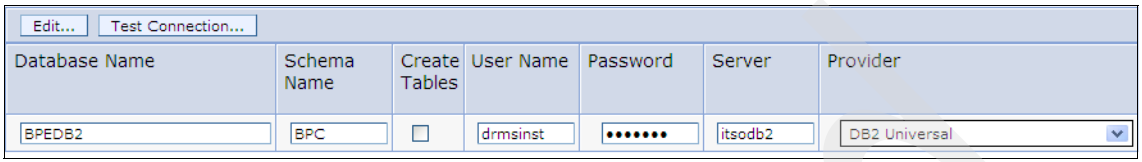


Figure 7-14 Data Source properties

5. We do not configure the human task email service. That can be done later if required. Uncheck the **Enable e-mail service** check box in the Human Task Manager Mail Session section.
6. In the Security section, enter the users and passwords that you configured earlier (Figure 7-15).

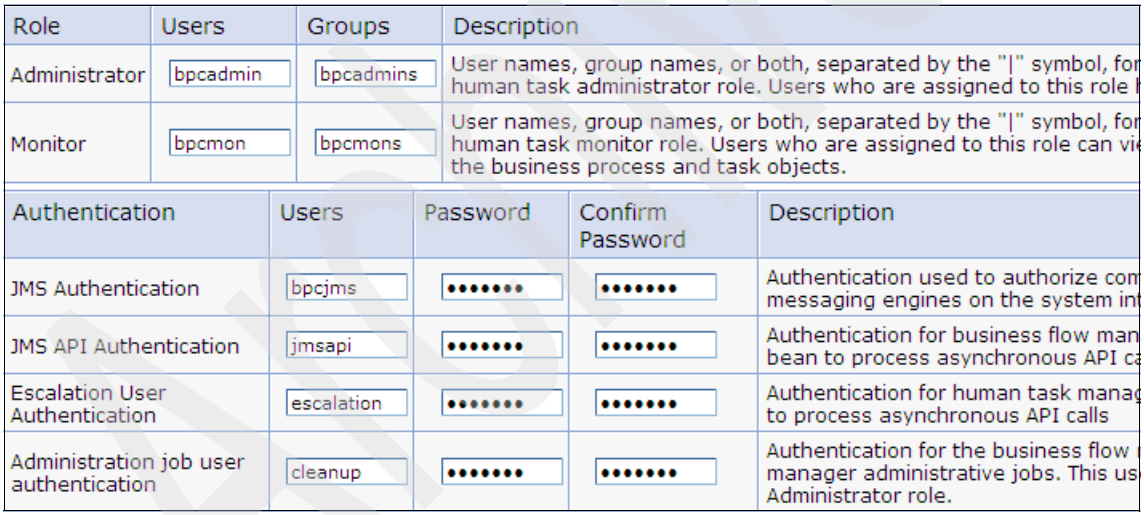


Figure 7-15 Security properties

Archive

7. The State Observer section allows the configuration of monitoring for business processes and human tasks. We enable the emission of CEI events for the business flow manager (BFM) and human task manager (HTM). Select the check boxes for CEI Logging for BFM and HTM (Figure 7-16).

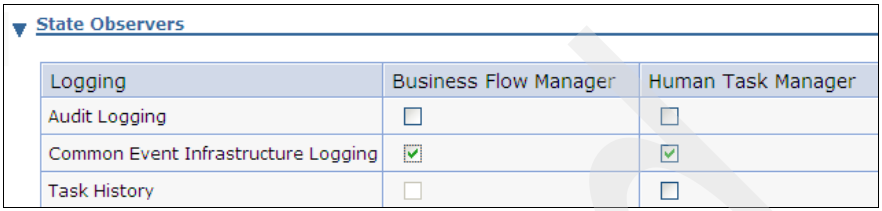


Figure 7-16 CEI Logging

8. In the SCA Bindings section leave the default values.
9. In the Bus section do the following:
   - Clear the **Use the default configuration** check box.
   - Click the **Remote** radio button.
   - If you cannot see the second messaging cluster in the drop-down list, click **New** and add it to the list.
   - Select the **WebSphere:cluster=RMRS.Messaging2** as the remote location (Figure 7-17).

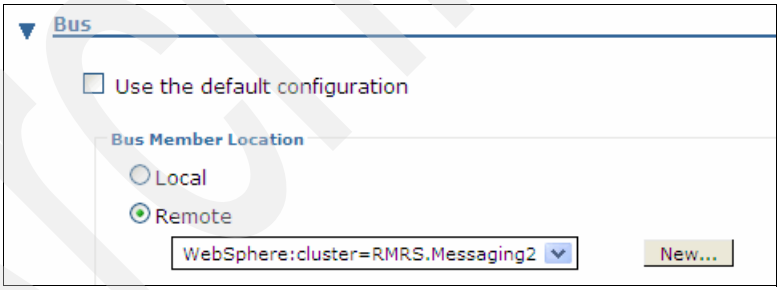


Figure 7-17 Bus member location properties

Archived

- Enter `MEDB2` as the database name.
- Enter `BPCME` as the schema name.
- Clear the Create Tables check box.
- Enter the user name, password, and server name according to your database settings.
- Leave DB2 Universal in the Provider drop-down list (Figure 7-18).

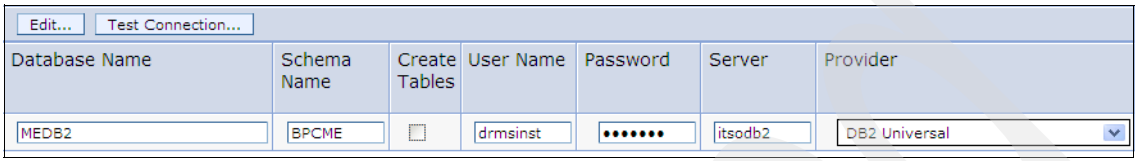


Figure 7-18 Database properties

10. Click **OK** to deploy the BPC.
11. When the configuration completes, you will see the following message:

```
CWWBZ0252I: The Business Process Choreographer component is
configured on the RMRS.AppTarget2 deployment target.
```

12. Save your configuration.

7.3.4 Setting the authentication aliases for data sources

To do this:

1. From the Integrated Solutions Console, navigate to **Resources** → **JDBC** and click **Data Sources** (Figure 7-19).

Figure 7-19 Data sources

2. Select the **RMRS.Messaging2** scope from the drop-down list to display the related data sources (Figure 7-20).

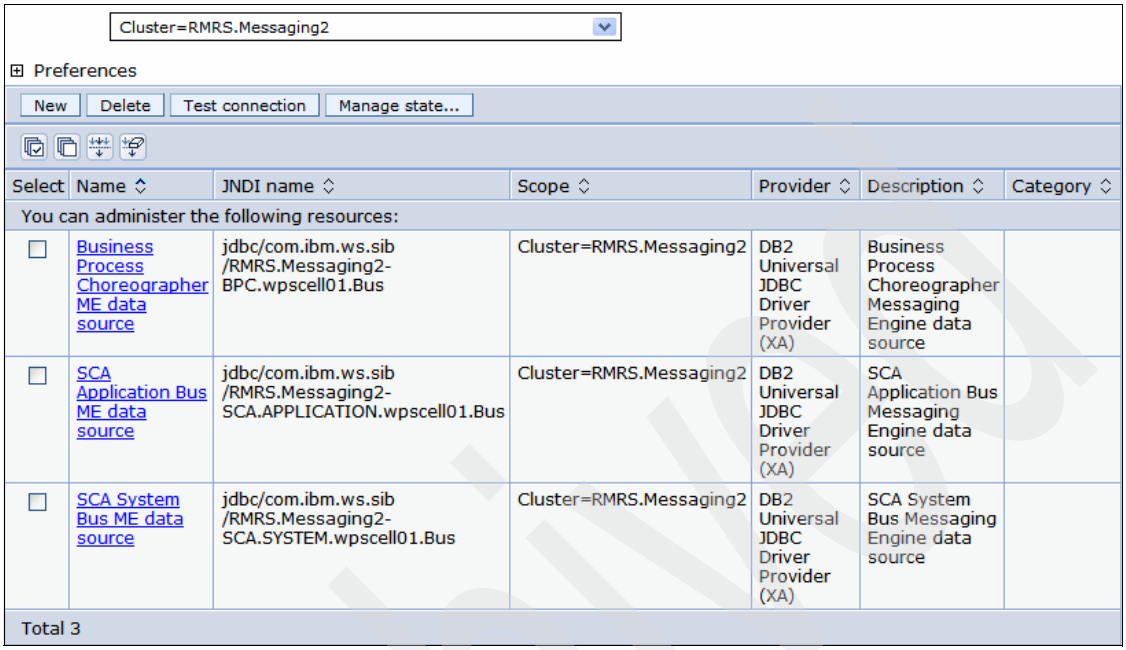


Figure 7-20 Data sources in scope of RMRS.Messaging2

Archive

3. Click the **Business Process Choreographer ME data source** link. On the Configuration page, scroll down to the Security settings section. Here set the following values in the drop-down lists:
   - Authentication alias for XA recovery: **BPCME_01_Auth_Alias**
   - Component-managed authentication alias: **BPCME_01_Auth_Alias** (Figure 7-21).

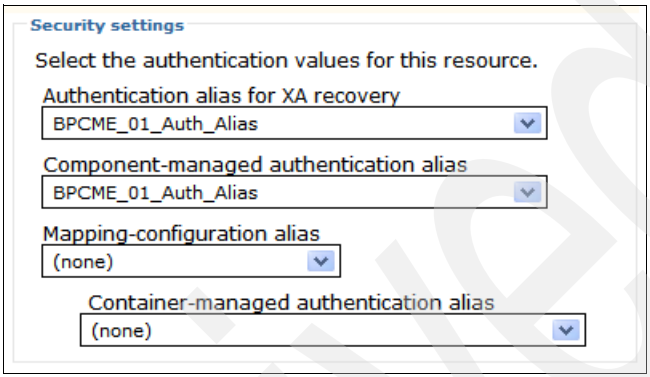


Figure 7-21 Security settings

4. Click **OK** and **Save** the changes.
5. Repeat steps 3–4 for each new data source:
   - SCA Application Bus ME data source
     - Authentication alias for XA recovery: **SCAAPPME01_Auth_Alias**
     - Component-managed authentication alias: **SCAAPPME01_Auth_Alias**
   - SCA System Bus ME data source
     - Authentication alias for XA recovery: **SCASYSME01_Auth_Alias**
     - Component-managed authentication alias: **SCASYSME01_Auth_Alias**
6. Ensure that the nodes are synchronized after the changes.
7. Verify your settings by selecting the check boxes next to the three data sources and clicking **Test connection**.

7.3.5 Installing an additional BPC Explorer

With BPC Explorer we can manage one BPC configuration. It is bound to one deployment target. As in a Double Remote Messaging and Remote Support

Archived

topology, we have another deployment cluster (RMRS.AppTarget2). We must install a separate BPC Explorer to manage the applications deployed there:

1. From the Integrated Solutions Console navigate to **Servers** → **Clusters** → **WebSphere application server clusters** and click **RMRS.Support**.
2. On the Configuration page, under the Business Integration heading, click **Business Process Choreographer** → **Business Process Choreographer Explorer** (Figure 7-22).

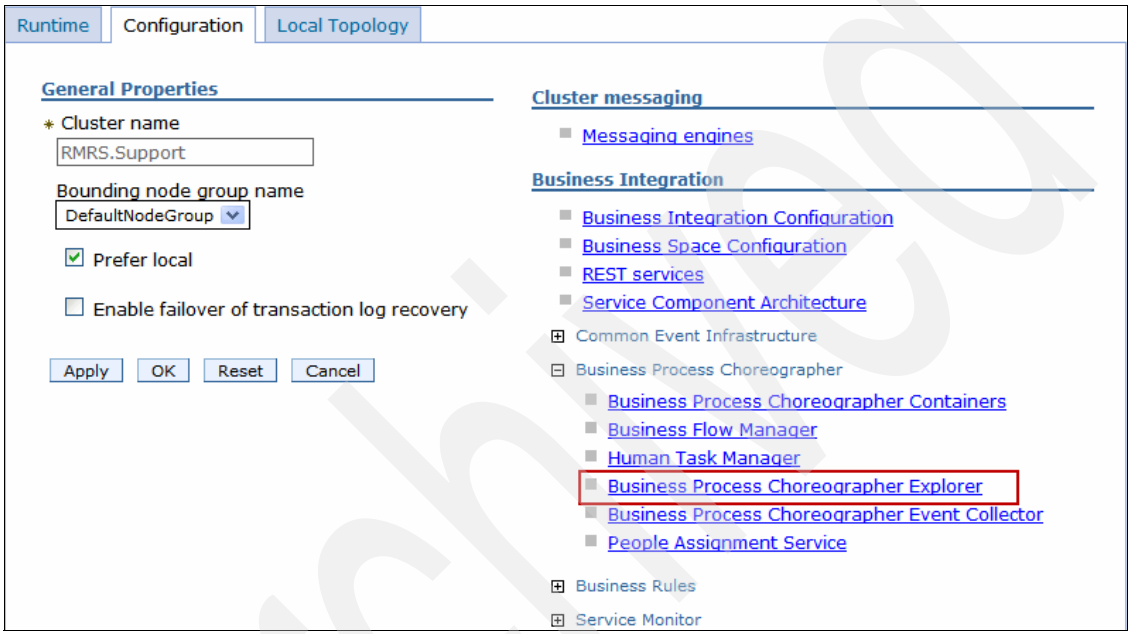


Figure 7-22 BPC Explorer

3. Note that there is a BPC Explorer installed already. Click **Add** (Figure 7-23).

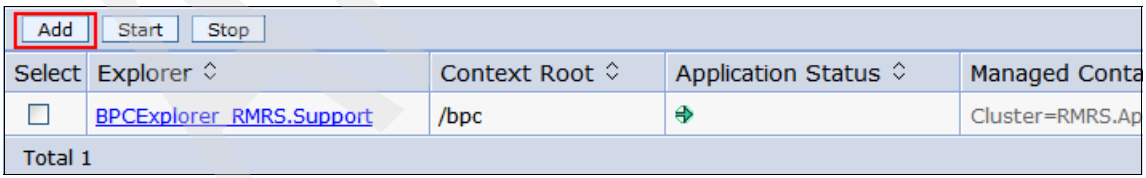


Figure 7-23 Add new BPC Explorer

4. On the Configuration page, you can to set the properties for the new BPC Explorer. Apply the proxy server address to achieve load balancing for Representational State Transfer (REST) API calls. Setting up a proxy server is described in 3.6.3 “Installing and configuring WebSphere proxy server” on

page 118. In our environment we installed a proxy server on wps1node2, listening on port 8043. The BPC Explorer settings are:

- Context root: `/bpc2`
- Managed Business Process Choreographer container: **Cluster=RMRS.AppTarget2**
- Business Flow Manager REST API URL: `https://wps1node2:8043/rest/bpm/bfm`
- Human Task Manager REST API URL: `https://wps1node2:8043/rest/bpm/htm`
- Select the **Enable reporting function** check box (Figure 7-24).

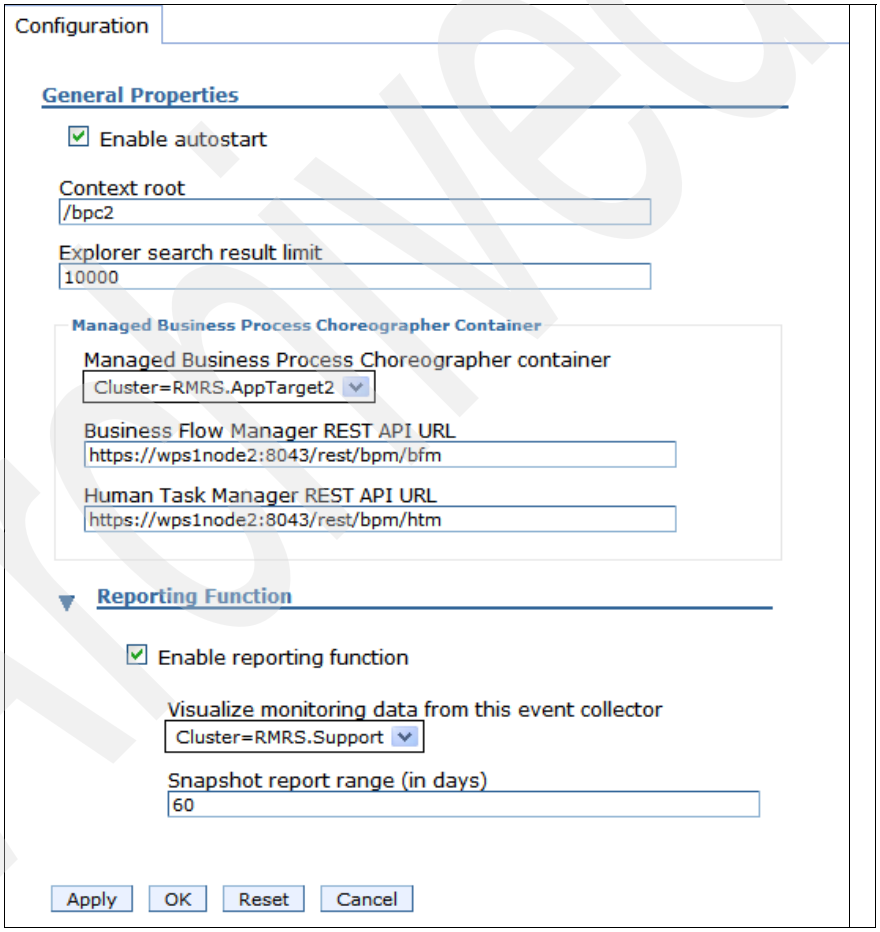


Figure 7-24 BPC Explorer properties

Archived

5. Click **OK** to deploy the BPC Explorer. The following message should be displayed:

   ```
   ADMA5013I: Application BPCExplorer_RMRS.Support_bpc2 installed
   successfully.
   ```

6. Click **Save Changes**.

After installing the BPC Explorer, it must be mapped to the Web server in order to route the requests correctly. The following steps achieve this:

1. From the Integrated Solutions Console, navigate to **Applications** → **Application Types** → **WebSphere enterprise applications**.
2. Click the **BPCExplorer_RMRS.Support_bpc2** link.
3. On the Configuration panel, in the Modules section, click **Manage Modules**.
4. Select both the support cluster and the Web server from the list, and select the check boxes next to the **bpcobserverejb** and **BPCExplorer** modules (Figure 7-25). Click **Apply**, then **OK** to finish the configuration.

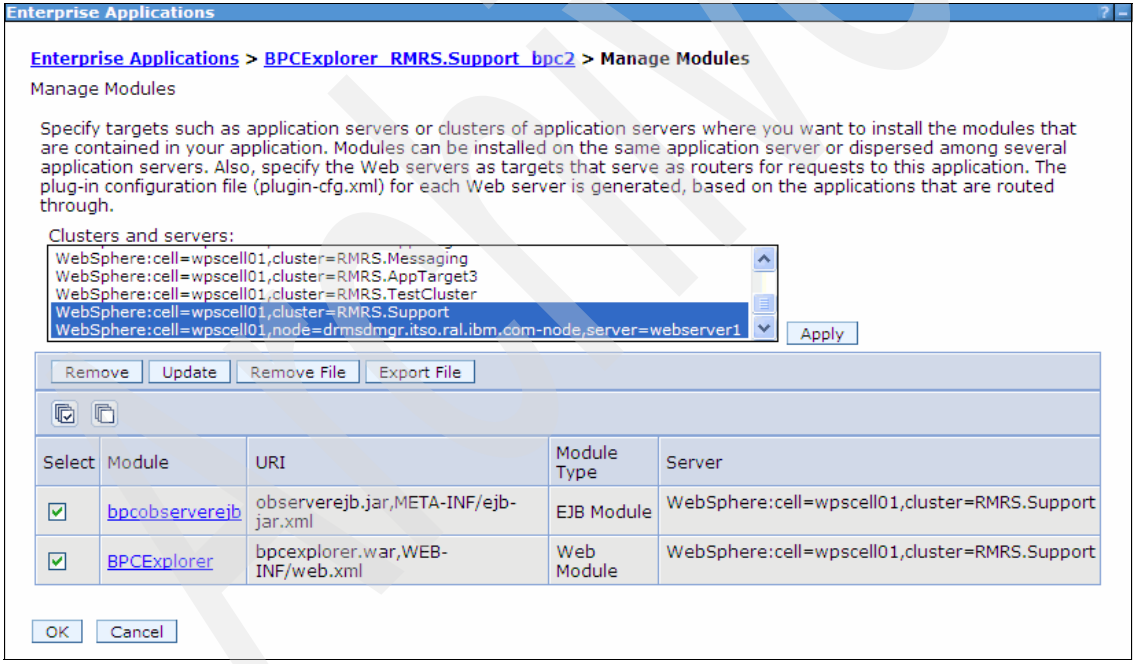


Figure 7-25 Manage Modules

5. Save your changes and synchronize the nodes.

Finally, start and test the application:

1. From the Integrated Solutions Console navigate to **Servers** → **Clusters** → **WebSphere application server clusters** and click **RMRS.Support.**
2. On the Configuration page, under the Business Integration heading, click **Business Process Choreographer** → **Business Process Choreographer Explorer**.
3. On the list of the installed BPC Explorers, select the check box next to **BPCExplorer_RMRS.Support_bpc2** and click **Start**. Verify that the new BPC Explorer started correctly (Figure 7-26).

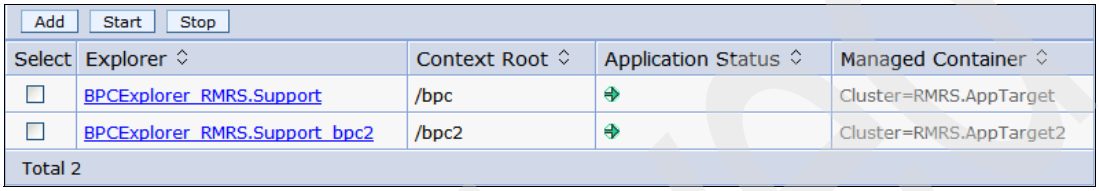


Figure 7-26 Started BPC Explorer

4. In the Integrated Solutions Console, click **Servers** → **Server Types** → **Web servers**, select the check box next to your Web server, and click **Generate Plug-in**, then **Propagate Plug-in**.
5. Direct your Web browser to the BPC application on the Web server, `http://wps1node1/bpc2` in our environment.
6. Verify that the BPC Explorer can be started and that you can log in to the Web interface as an appropriate user.

7.3.6 Configuring the Common Event Infrastructure

In this section we describe how to configure the new application cluster to reuse the existing CEI functionality. In our topology both application clusters will use a single support cluster with a shared CEI.

Archived

Configuring CEI destination for the new application cluster

To do this:

1. Log in and expand **Servers** → **Clusters** in the Integrated Solutions Console, then click **WebSphere application server clusters**.
2. Click the **RMRS.AppTarget2** link (Figure 7-27).

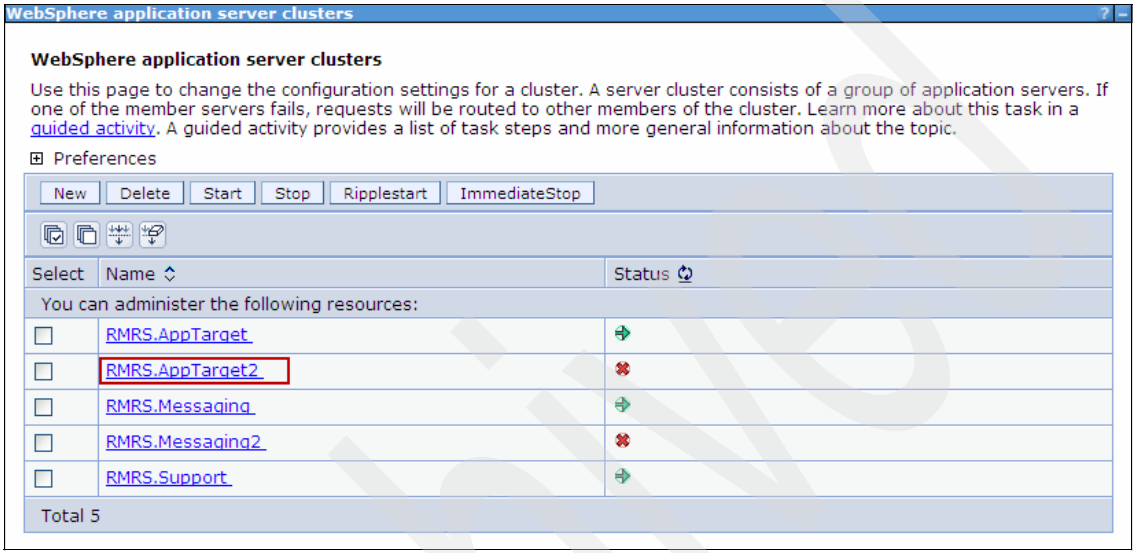


Figure 7-27 Clusters

Archive

3. On the Configuration page, in the Business Integration section, expand **Common Event Infrastructure**. Click the **Common Event Infrastructure Destination** link (Figure 7-28).

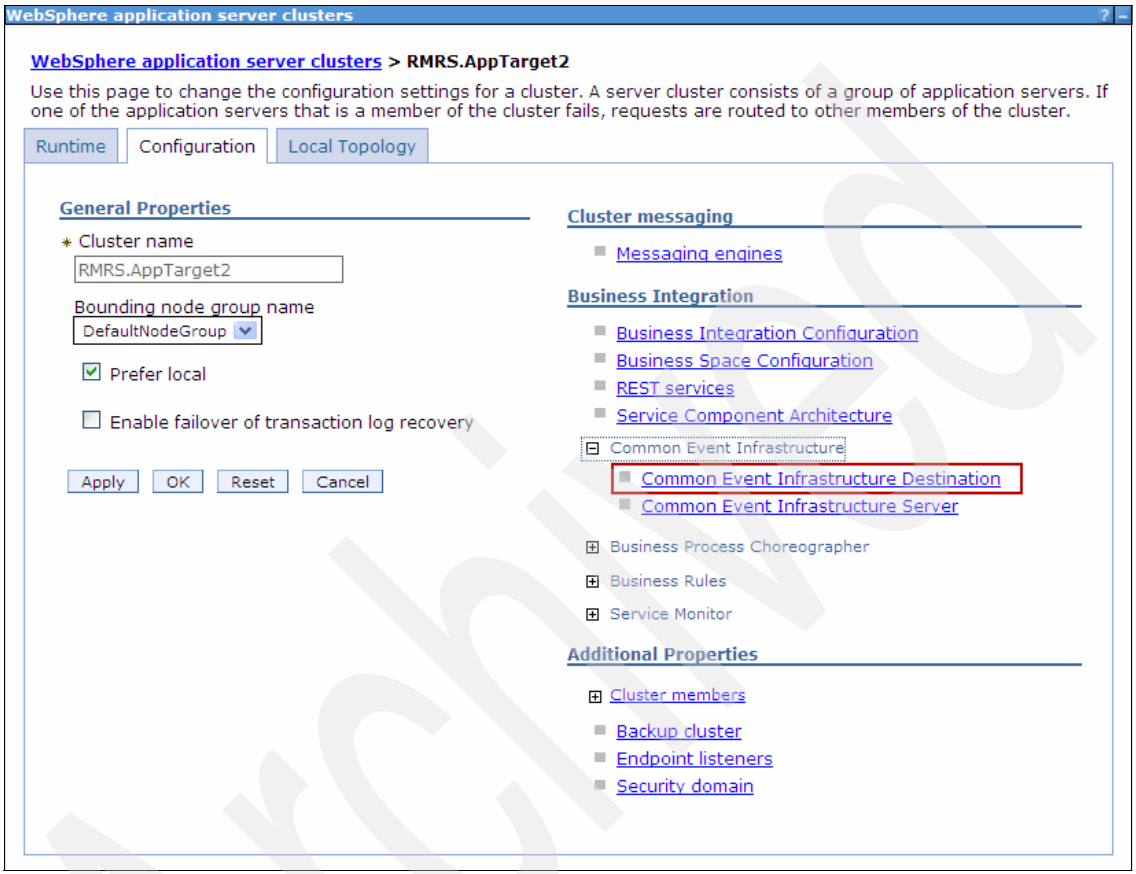


Figure 7-28 CEI settings

Archived

4. Select the CEI destination defined at cell scope by clicking the radio button next to the drop-down list that shows cell/clusters/RMRS.Support/com/ibm/events/configuration/emitter/Default (Figure 7-29).

Figure 7-29 CEI destination

5. Click **OK** to finish and save your configuration.

7.3.7 Configuring the messaging target

This section describes the following:

- The cross-connection pattern
- Related administrative objects and settings
- Configuring application target properties
- Automated configuration of target properties

The cross-connection pattern

The Remote Messaging and Remote Support topology has a messaging cluster with two cluster members. By default each messaging engine (ME) becomes active on the first available cluster member.

For the Double Remote Messaging and Remote Support topology we created a new messaging cluster for the new application cluster. Each additional ME of the new messaging cluster will be a member of the already existing Service Integration Buses. This allows applications to connect to the MEs in both messaging cluster.

Archived

Messaging engines hold destinations for SCA applications. To access destinations, the application must connect to the ME in the appropriate messaging cluster. At run time the Work Load Manager (WLM) decides which ME will be used by the application. In the case of a single cluster and a single active ME, the destination is immediately located. If we have two MEs on the same bus, the WLM may direct the connection to the ME that does not hold the destination.

This causes runtime creation of remote queue points, which are used to manage the flow of messages between the messaging engine where the destination is located and the messaging engine to which the application is attached. In this way the application that has to connect to the destination through another ME suffers from unnecessary performance overhead and other implications.

A simplified diagram on Figure 7-30 shows this behavior, when an application wants to connect to a destination in the first messaging cluster (dashed black arrow), but the WLM connect it to the ME in the second messaging cluster, which in turn creates a remote queue point and forwards the message to the ME in the first messaging cluster (solid red arrows).

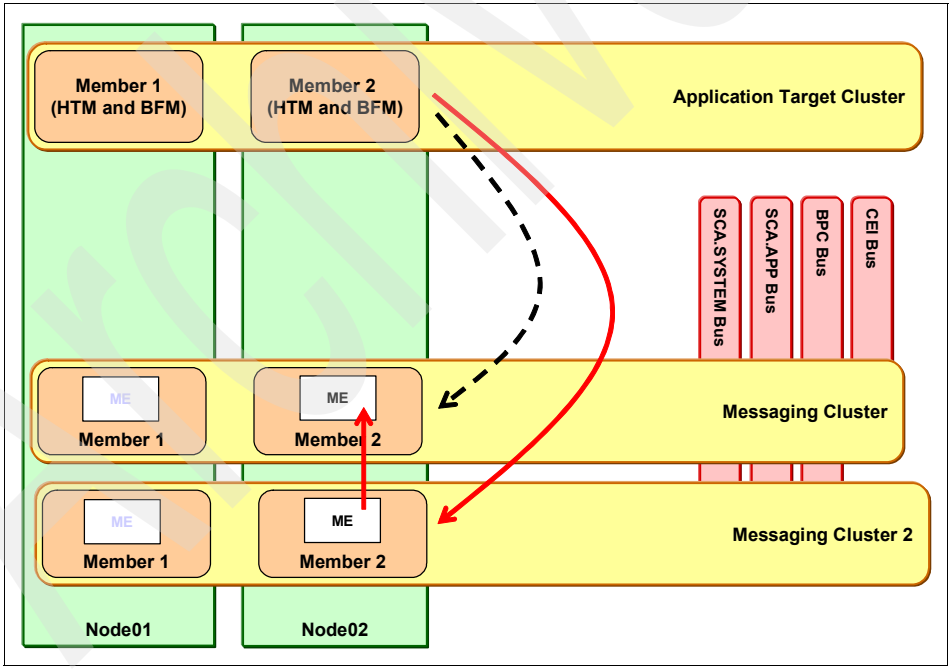


Figure 7-30 Message flow

Related administrative objects and settings

The connection from an SCA module to the SCA.SYSTEM and SCA.APPLICATION buses is defined by the configuration of the following administrative objects:

- Java Message Service (JMS) connection factory
- JMS activation specification
- J2C resource adapter activation specification

These administrative objects are generated when we deploy an application.

There are three properties of these objects that must be configured in order to ensure that the application connects to a certain messaging engine:

- Target type

  The target type can be bus member, a custom messaging engine group, or a single messaging engine.

- Target name

  The target name identifies a group of messaging engines.

- Target significance

  The target significance sets the way a messaging engine is chosen:

  – Required: only allows connections to the MEs defined in the target group
  – Preferred: allows connections to MEs outside the target group

Configuring application target properties

In order to demonstrate target property settings, we install and configure two sample modules from the Business Process Management Samples at Tutorials - Version 7.0 sample library:

`http://publib.boulder.ibm.com/bpcsamp/gettingStarted/jmsInvocation.html`

This simple scenario implements an asynchronous invocation of another module via JMS from a Business Process Execution Language (BPEL) process.

1. Download the two EAR files from the sample library:

   – JMSInvokerApp

     `http://publib.boulder.ibm.com/bpcsamp/gettingStarted/jmsInvocation/download/JMSInvokerApp.ear`

   – MPGConverterApp

     `http://publib.boulder.ibm.com/bpcsamp/gettingStarted/jmsInvocation/download/MPGConverterApp.ear`

Archived

2. Install the MPGConverterApp and then the JMSInvokerApp applications. In the Integrated Solutions Console, click **Applications** → **WebSphere enterprise applications** (Figure 7-31).

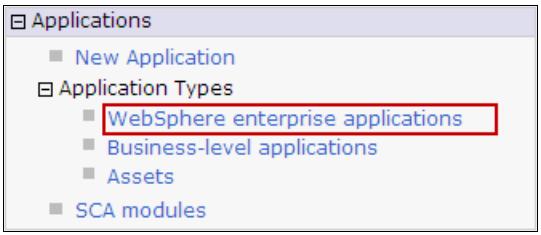


Figure 7-31 Enterprise applications

3. Click **Install** to install the new application. Choose **MPGConverterApp.ear** from your local system and click **Next** (Figure 7-32).

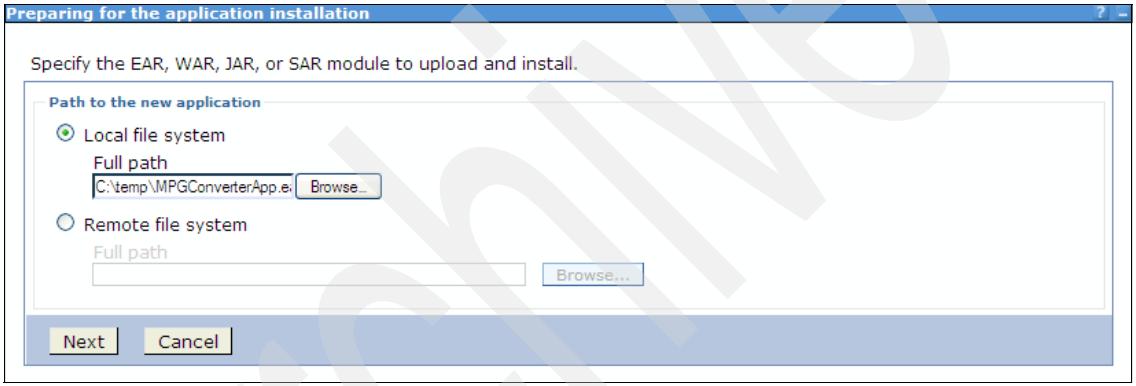


Figure 7-32 Select MPGConverterApp

4. Leave the default fast path installation and click **Next** twice in the wizard.

Archived

5. Select **WebSphere:cell=wpscell01, cluster=RMRS.AppTarget** from the list and click the check box next to the application. Click **Apply** (Figure 7-33).

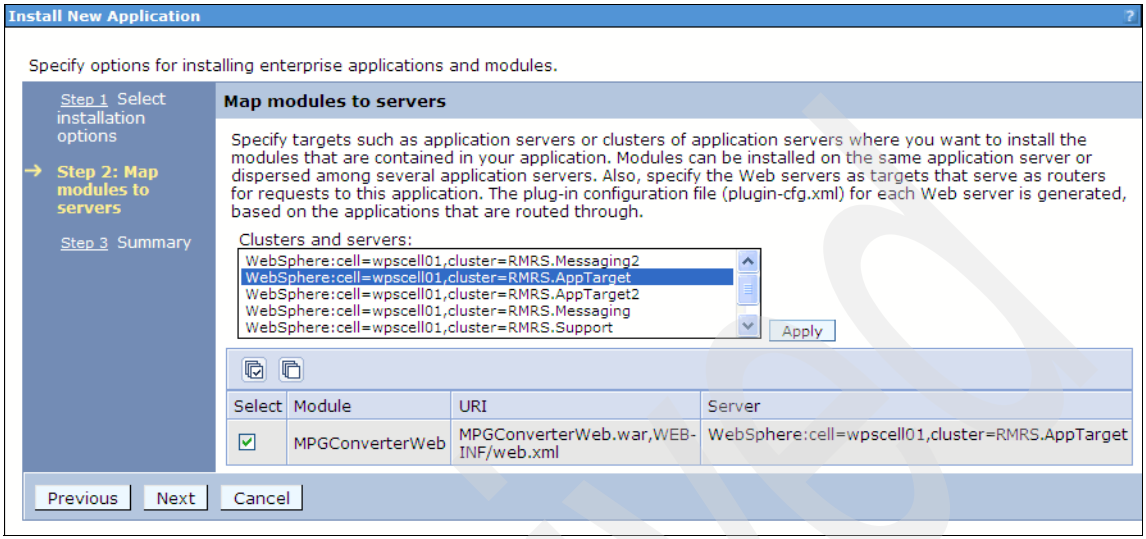


Figure 7-33 Map modules

6. Click **Next**, then **Finish** at step 3.
7. After the application has installed successfully, save the configuration.
8. Repeat steps 2–8 and install JMSInvokerApp as well.
9. Expand the **JMS** settings in the Resources section in the Integrated Solutions Console, then click **Connection factories** (Figure 7-34).

Figure 7-34 Connection factories

10. The generated connection factories are displayed. Click **JMSInvoker.MPGConverterProcessExport_CF** to configure it.

11. Set the following fields in the Connection section (leave the other fields as the defaults):
   - Target: `RMRS.Messaging.000-SCA.APPLICATION.wpscell01.Bus`
   - Target type: Choose **Messaging engine name** from the drop-down list.
   - Target significance: Choose **Required** from the drop-down list (Figure 7-35).

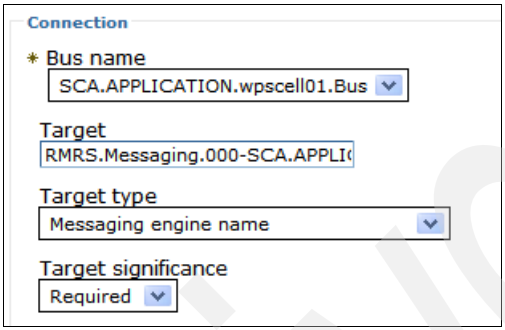


Figure 7-35 Connection properties

12. Click **OK** and save the changes.
13. Repeat the same steps (4–6) to configure all the other connection factories used by your application and also the ones used internally by BFM and HTM. Set the properties for each connection factory noting the following:
   - Set up the applications so that they use the ME on the related messaging cluster. This means that applications installed onto the first application cluster use the first messaging cluster, and applications installed onto the second application cluster should use the MEs on the second messaging cluster.
   - In the Target field, enter the messaging engine name that the actual application should use. This ME should be a member of the bus defined in the Bus name drop-down list.
   - Alternatively, you might choose to set Target type of Bus member (and set the messaging cluster name as the target name) instead of using a Target type of Messaging engine name.
   - You can display the names of the available messaging engine by going to **Service Integration** → **Buses** → **<bus name>** → **Messaging engines**. Make sure that you correctly type the names, otherwise the application fails to connect to the messaging engine.

Archived

14. To configure the activation specifications, expand the **JMS** settings in the Resources section in Integrated Solutions Console, then click **Activation specifications**.
15. Note the generated activation specifications for the two modules installed. Click **JMSInvoker.MPGConverterProcessExport_AS** from the list of activation specifications.
16. Apply the following fields in the Destination section (leave the other fields as the defaults):
    - Target: `RMRS.Messaging.000-SCA.APPLICATION.wpscell01.Bus`
    - Target type: Choose **Messaging engine name** the drop-down list.
    - Target significance: Choose **Required** from the drop-down list.
17. Click **OK** and save the changes.
18. Repeat the same steps (9–11) for all other activation specifications. Always set the desired messaging engine for the actual application.

Automated configuration of target properties

The configuration of connection factories and activation specifications for every installed application can be a very time-consuming task. However, there is a freely available script that can help you to configure the target settings of the already-installed applications in a faster, more efficient way.

This script also updates the administrative object templates to ensure that future application deployments will be configured correctly. Still verify whether the connection factories and activation specifications are correctly configured after installing new applications.

The script assumes that all resources are defined at cluster scope, and calculates the associated ME cluster name from the defined SCA remote destination location of the application cluster. It changes the following settings of all existing connection factory and activation specification objects and templates:

- Target: the associated ME cluster name calculated from the remote SCA destination defined for the application cluster. In our environment, RMRS.Messaging or RMRS.Messaging2.
- Target type: Bus member
- Target significance: Required

These settings enable our application to connect to the correct messaging engine and overcome the cross-connection pattern.

To configure target significance by scripting:

1. Download the `configure_MultipleGold_Messaging.jacl` script file to your deployment manager. The script is downloadable from the following URL:

   http://www-01.ibm.com/support/docview.wss?rs=2307&uid=swg27013612&S_TACT=105AGX10&S_CMP=ART

2. In order to review the intended changes run the script with the VALIDATE parameter:

   ```
   ./wsadmin.sh -username wpsadmin -password itos4you -lang jacl -f
   configure_MultipleGold_Messaging.jacl VALIDATE RMRS.AppTarget
   Required
   ```

3. Run the script with the parameter CORRECT so that it updates all values according to the parameters given:

   ```
   ./wsadmin.sh -username wpsadmin -password itos4you -lang jacl -f
   configure_MultipleGold_Messaging.jacl CORRECT RMRS.AppTarget
   Required
   ```

4. Verify that the script finished successfully. You will see the following message:

   ```
   -----------------------------------------------------
   Saving any configuration changes that were made.
   -----------------------------------------------------
   ```

5. Apply the same steps for the second application cluster as well:

   ```
   ./wsadmin.sh -username wpsadmin -password itos4you -lang jacl -f
   configure_MultipleGold_Messaging.jacl VALIDATE RMRS.AppTarget2
   Required

   ./wsadmin.sh -username wpsadmin -password itos4you -lang jacl -f
   configure_MultipleGold_Messaging.jacl CORRECT RMRS.AppTarget2
   Required
   ```

Archived

6. Verify the changes in the Integrated Solution Console. You will see the changed connection factory and activation specification settings (Figure 7-36).

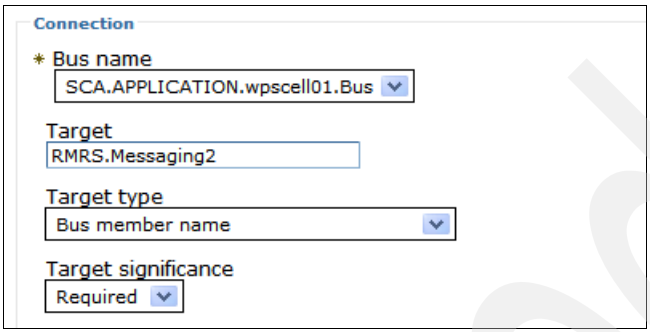


Figure 7-36 Connection factory properties

7. Synchronize your nodes and restart both application clusters to finish the configuration.

Note: You can read more about the configure_MultipleGold_Messaging.jacl script in the following developerWorks article:

http://www.ibm.com/developerworks/websphere/library/techarticles/0811_roberts/0811_roberts.html

7.4 Creating new clusters manually using command line

You can use the wsadmin tool and admin tasks to add new clusters and cluster members to the Remote Messaging and Remote Support topology. We provide two Jython scripts with which you can achieve this:

- createNewAppCluster.py (Example 7-1)

Example 7-1 Creating a new application cluster

```
AdminTask.createCluster('[-clusterConfig [-clusterName RMRS.AppTarget2
-preferLocal true]]')

AdminTask.createClusterMember('[-clusterName RMRS.AppTarget2
-memberConfig [-memberNode wps1node1 -memberName
RMRS.AppTarget2.wps1node1.0 -memberWeight 2 -genUniquePorts true
-replicatorEntry false] -firstMember [-templateName
```

```
defaultProcessServer -nodeGroup DefaultNodeGroup -coreGroup
DefaultCoreGroup]]')

AdminTask.createClusterMember('[-clusterName RMRS.AppTarget2
-memberConfig [-memberNode wps1node2 -memberName
RMRS.AppTarget2.wps1node2.0 -memberWeight 2 -genUniquePorts true
-replicatorEntry false]]')

AdminConfig.save()
```

- createNewMessCluster.py (Example 7-2)

Example 7-2 Creating a new messaging cluster

```
AdminTask.createCluster('[-clusterConfig [-clusterName RMRS.Messaging2
-preferLocal true]]')

AdminTask.createClusterMember('[-clusterName RMRS.Messaging2
-memberConfig [-memberNode wps1node1 -memberName
RMRS.Messaging2.wps1node1.0 -memberWeight 2 -genUniquePorts true
-replicatorEntry false] -firstMember [-templateName
defaultProcessServer -nodeGroup DefaultNodeGroup -coreGroup
DefaultCoreGroup]]')

AdminTask.createClusterMember('[-clusterName RMRS.Messaging2
-memberConfig [-memberNode wps1node2 -memberName
RMRS.Messaging2.wps1node2.0 -memberWeight 2 -genUniquePorts true
-replicatorEntry false]]')

AdminConfig.save()
```

To execute each script use the wsadmin tool (Example 7-3).

Example 7-3 Execute cluster creation scripts

```
./wsadmin.sh -user wpsadmin -password itso4you -f
createNewAppCluster.py
./wsadmin.sh -user wpsadmin -password itso4you -f
createNewMessCluster.py
```

Archived

7.5 Post-creation configuration and verification

This section describes how to distribute the active messaging engines between cluster members. Finally, we verify that the Double Remote Messaging and Remote Support topology functions correctly.

7.5.1 Distributing the messaging load among cluster members

In this section we describe how to configure your Double Remote Messaging and Remote Support topology so that the active MEs are distributed between cluster members, as shown in the topology diagram in Figure 7-1 on page 208. This distributes the messaging load across the cluster.

Background

A high-availability group (HAGroup) provides the mechanism for building a highly available component and enables the component to run in one of several processes. A HAGroup cannot extend beyond the boundaries of a core group, and is associated with a specific component. The members of the group are the set of processes where it is possible to run that component.

An administrator cannot directly configure or define a high-availability group and its associated set of members. Instead, high-availability groups are created dynamically at the request of the components that must provide a highly available function.

Every HAGroup has a policy associated with it. This policy is used to determine which members of a HAGroup are active at a given time. The policies that the HAGroups use are stored as part of the core group configuration. The policy is used to determine which members of a HAGroup are active at a given point in time.

A match criteria is a set of one or more name-value pairs in a policy definition. You use the match criteria to make a policy bind to a particular messaging engine or a set of messaging engines. You can specify multiple match criteria. The more that match, the stronger the match becomes. The most general policy is the default included with the product, the *Default SIBus Policy*. This policy has a single match criterion, type=WSAF_SIB, which matches any messaging engine that does not have a stronger match to another policy.

Note: For a more detailed description of high-availability concepts refer to the *WebSphere Application Server V7: Concepts, Planning and Design*, SG24-7708:

http://www.redbooks.ibm.com/abstracts/sg247708.html

Creating new policy

The appropriate distribution of MEs always depends on your actual environment. In our example, using the topology of our Double Remote Messaging and Remote Support environment, we want to ensure that the SCA system and application MEs are started on the first member, and the BPC and CEI MEs start on the second member of the cluster. This same setting should apply to both messaging clusters (however, the CEI ME is only available on the first messaging cluster). Figure 7-1 on page 208 shows the final distribution of active MEs.

For each messaging engine you are going to pin to a particular cluster member, define a policy. In our environment, this means four policies on the first messaging cluster and three additional policies for the second messaging cluster.

The configuration steps are the same for all MEs. We describe how to configure the SCA.SYSTEM ME in order to activate it on the first cluster member of the first messaging cluster.

In the first step we add a new policy to the DefaultCoreGroup. We set the One of N policy, which means that only one member of the group is active. We select the Preferred servers only option, which ensures that the active server is always selected from the preferred servers list, which we define in a later step.

Archived

We also set the failback property, which ensures that if the preferred server fails, another ME is activated, but after the preferred server restarts, the ME fails back to the preferred server once again:

1. Expand **Servers** → **Core Groups** and click **Core group settings** (Figure 7-37).

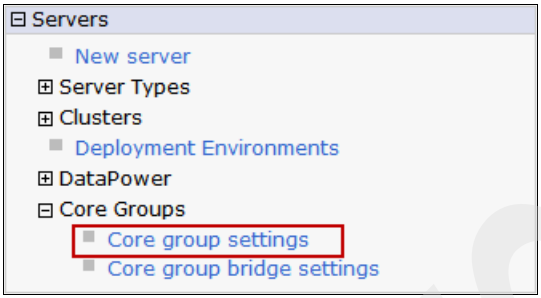


Figure 7-37 Core group settings

Archived

2. Choose the **DefaultCoreGroup** and click **Policies** on the Configuration page (Figure 7-38).

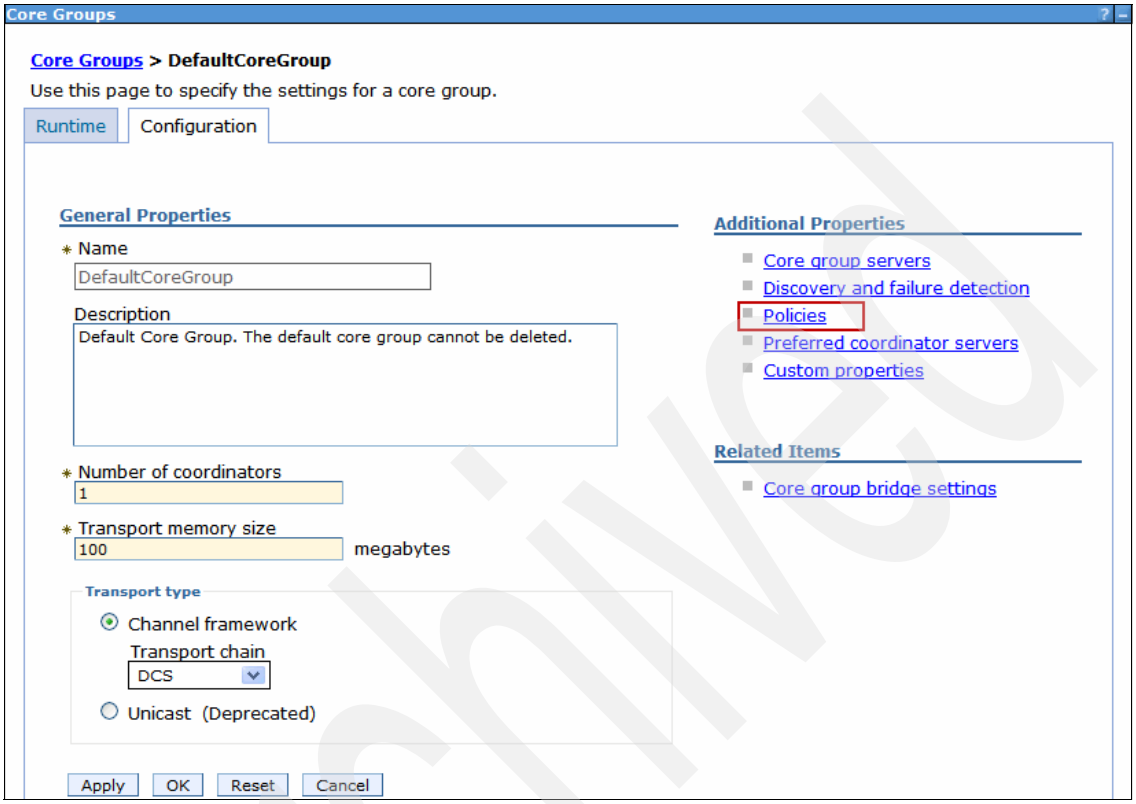


Figure 7-38 DefaultCoreGroup policies settings

3. Click **New** to add a new policy to the list (Figure 7-39).

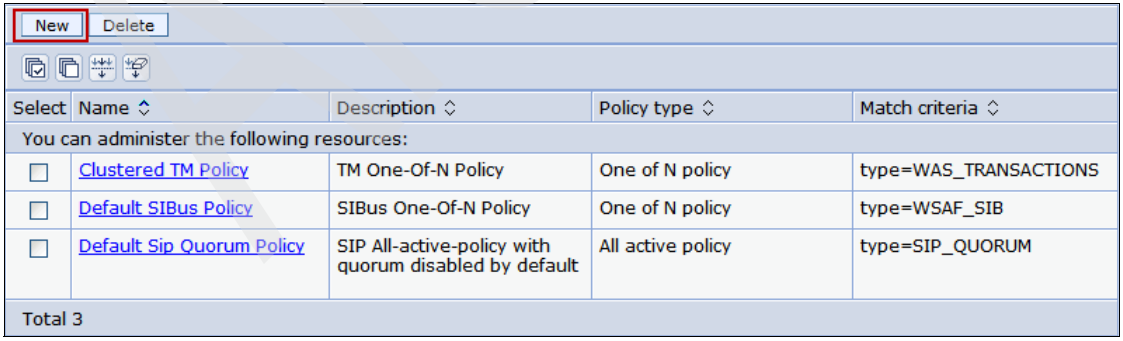


Figure 7-39 Adding new policy

Archived

4. Choose **One of N policy** from the drop-down list and click **Next**.
5. Set the following property values for the policy (leave the others as the defaults) and click **OK**:
   - Name: `SCA_SYSTEM_ME on Member1`
   - Select the **Failback** check box.
   - Select the **Preferred servers only** check box (Figure 7-40).

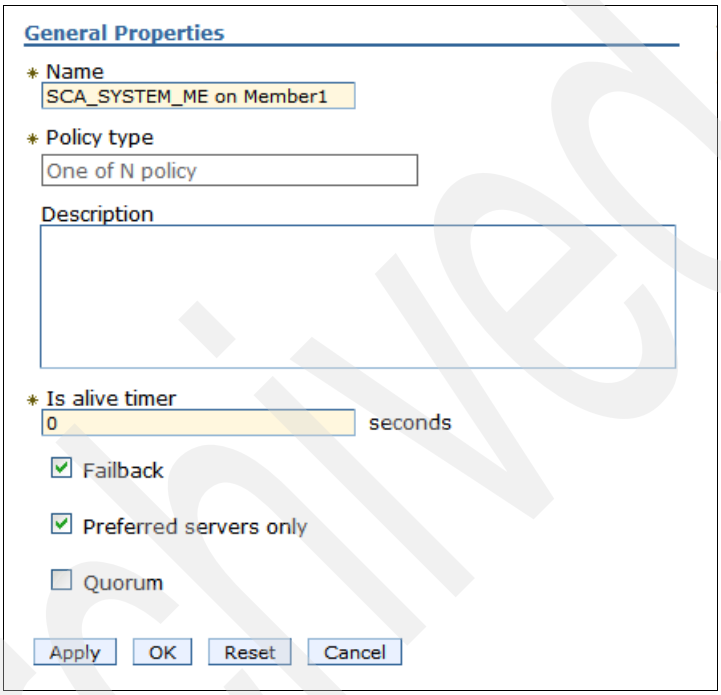


Figure 7-40 Properties of the new policy

6. Ignore the warning about missing match criteria and **Save** the changes.

Archived

Adding new match criteria

You must configure the match criteria of the policy to match the properties of the HAGroup (Table 7-1).

Table 7-1 HAGroup properties

HAGroup property name	Value	MEs that the policy matches
type	WSAF_SIB	Any ME
WSAF_SIB_MESSAGING_ENGINE	The name of the ME	A particular ME
WSAF_SIB_BUS	The name of the bus	All MEs in a bus
IBM_hc	The name of the cluster	All MEs in a cluster

You can specify more match criteria, which makes the match stronger. The policy with the stronger match is applied always. You must define more than one match criteria, otherwise there is a conflict with the *Default SIBus Policy*, which already contains one match. In our environment we configure four match criteria:

1. Select your newly created policy **SCA_SYSTEM_ME on Member1**.
2. Click **Match criteria** under the Additional Properties section (Figure 7-41).

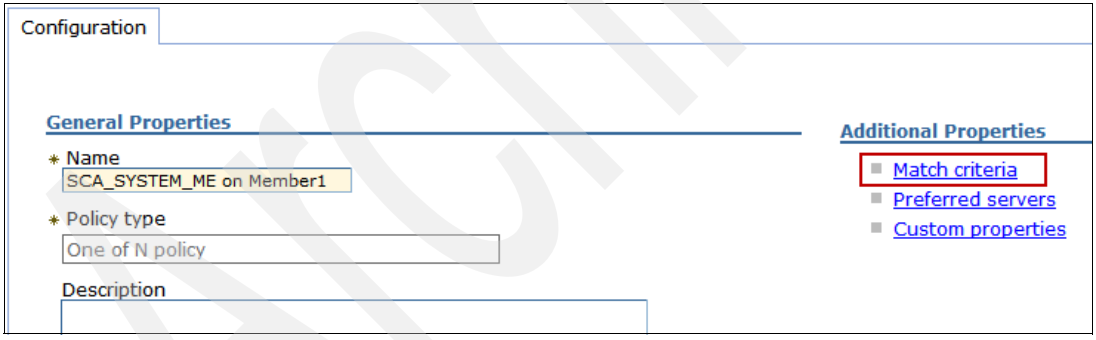


Figure 7-41 Match criteria

3. Click **New** to create a new match criteria.

Archived

4. On the Configuration page, in the General Properties section, fill out the name and value fields as shown in Figure 7-42. Click **OK** when ready.
   - Name: `type`
   - Value: `WSAF_SIB`

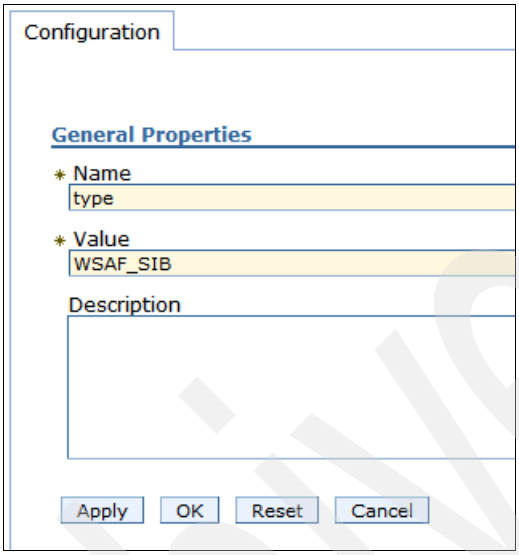


Figure 7-42 Match criteria configuration

5. Repeat the previous step with the following name-value pairs also:
   - Name: `WSAF_SIB_MESSAGING_ENGINE`
   - Value (the name of the messaging engine): `RMRS.Messaging.000-SCA.SYSTEM.wpscell01.Bus`
   - Name: `WSAF_SIB_BUS`
   - Value (the name of the bus): `SCA.SYSTEM.wpscell01.Bus`
   - Name: `IBM_hc`
   - Value (the name of the cluster): `RMRS.Messaging`

Archived

6. After reviewing the new match criteria (Figure 7-43), click **Save** to store the changes.

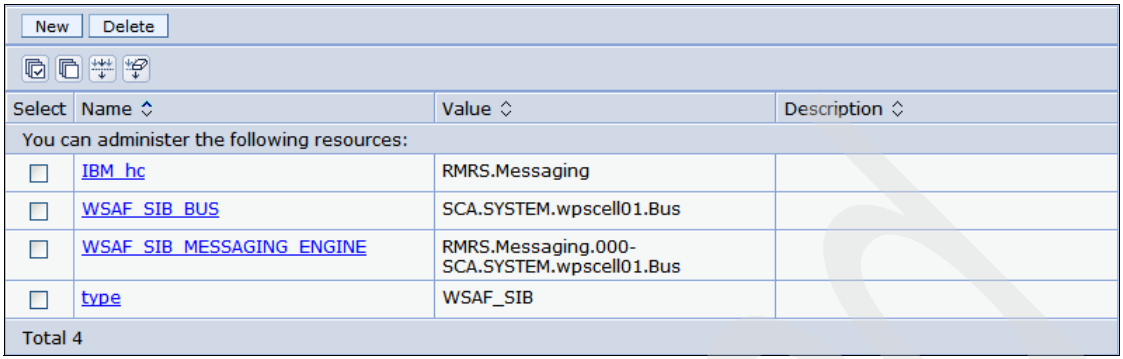


Figure 7-43 New match criteria

Archived

Defining preferred servers

We must define the list of preferred servers. Their order in the list decides which server is more preferable. In this way, we force the messaging engine to start always on the first server in the list. If it is unavailable, then the ME starts on the next server. We have already set the preferred servers only option, so the ME is only able to run on the listed servers. Perform the following steps to define the preferred servers:

1. In the Integrated Solutions Console, click **Servers** → **Core group settings**. Click **DefaultCoreGroup**. Under the Additional Properties heading, click **Policies**. From the list of policies, choose **SCA_SYSTEM_ME on Member1**. On the configuration page, click **Preferred servers** from the Additional Properties heading (Figure 7-44).

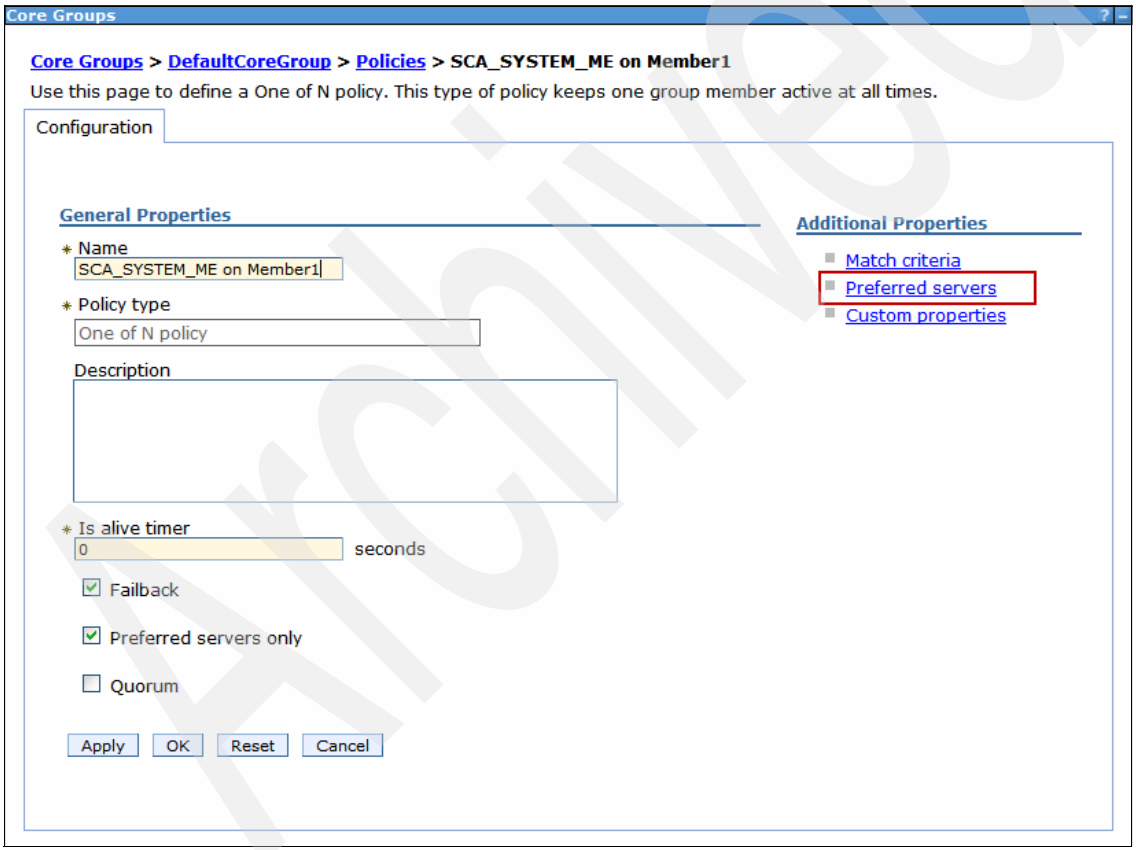


Figure 7-44 Choose Preferred servers

Archive

2. Add from the list of Core group servers to the Preferred servers the two members of the messaging cluster in this order (Figure 7-45):

   – wps1node1/RMRS.Messaging.wps1node1.0
   – wps1node2/RMRS.Messaging.wps1node2.0

   The order defines the priority with which the application servers host the active messaging engines.

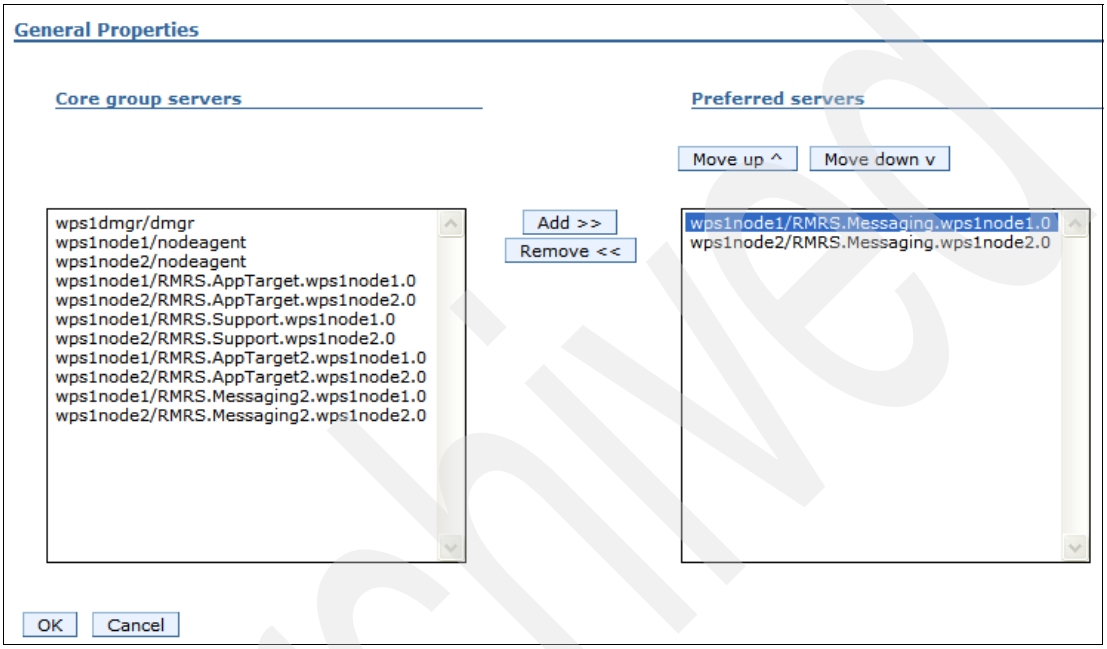


Figure 7-45 Preferred servers

3. Click **OK** and **Save** the changes.

To configure the distribution of MEs in your complete topology, follow the steps described in this section and create new policies for all MEs on both messaging clusters. Choose the appropriate preferred server order to control on which server each ME should be activated preferably.

Archived

Verify the list of defined policies at **Core group settings** → **Default Core Group** → **Policies** if you defined all seven policies correctly for all MEs (Figure 7-46).

Select	Name	Description	Policy type	Match criteria
You can administer the following resources:				
☐	SCA_SYSTEM2_ME on Member1		One of N policy	type=WSAF_SIB, WSAF_SIB_MESSAGING_ENGINE=RMRS.Messaging2.000-SCA.SYSTEM.wpscell01.Bus, WSAF_SIB_BUS=SCA.SYSTEM.wpscell01.Bus, IBM_hc=RMRS.Messaging2
☐	CEI_ME on Member2		One of N policy	type=WSAF_SIB, WSAF_SIB_MESSAGING_ENGINE=RMRS.Messaging.000-CEI.wpscell01.BUS, WSAF_SIB_BUS=CEI.wpscell01.BUS, IBM_hc=RMRS.Messaging
☐	SCA_APPLICATION_ME on Member1		One of N policy	WSAF_SIB_BUS=SCA.APPLICATION.wpscell01.Bus, type=WSAF_SIB, WSAF_SIB_MESSAGING_ENGINE=RMRS.Messaging.000-SCA.APPLICATION.wpscell01.Bus, IBM_hc=RMRS.Messaging
☐	SCA_APPLICATION2_ME on Member1		One of N policy	type=WSAF_SIB, WSAF_SIB_MESSAGING_ENGINE=RMRS.Messaging2.000-SCA.APPLICATION.wpscell01.Bus, WSAF_SIB_BUS=SCA.APPLICATION.wpscell01.Bus, IBM_hc=RMRS.Messaging2
☐	SCA_SYSTEM_ME on Member1		One of N policy	type=WSAF_SIB, WSAF_SIB_MESSAGING_ENGINE=RMRS.Messaging.000-SCA.SYSTEM.wpscell01.Bus, WSAF_SIB_BUS=SCA.SYSTEM.wpscell01.Bus, IBM_hc=RMRS.Messaging
☐	BPC_ME on Member2		One of N policy	type=WSAF_SIB, WSAF_SIB_MESSAGING_ENGINE=RMRS.Messaging.000-BPC.wpscell01.Bus, WSAF_SIB_BUS=BPC.wpscell01.Bus, IBM_hc=RMRS.Messaging
☐	BPC2_ME on Member2		One of N policy	type=WSAF_SIB, WSAF_SIB_MESSAGING_ENGINE=RMRS.Messaging2.000-BPC.wpscell01.Bus, WSAF_SIB_BUS=BPC.wpscell01.Bus, IBM_hc=RMRS.Messaging2

Figure 7-46 Policies

Archive

7.5.2 Verifying the configuration

The steps described in this section verify the correct configuration of the Double Remote Messaging and Remote Support topology.

1. Start all of your clusters and verify that both new clusters (RMRS.AppTarget2 and RMRS.Messaging2) start successfully (Figure 7-47).

Figure 7-47 All clusters are started

2. Review the logs on both nodes and verify that the MEs are started on the appropriate cluster member and that their counterpart is in the joined state. If you pinned the messaging engines to certain cluster members, check whether they are active on the correct cluster member.
   - Verify the MEs on the first messaging cluster (RMRS.Messaging):
     - On node wps1node1, review the following log file:

       `/opt/ibm/WebSphere/ProcServer/profiles/wps1node1/logs/RMRS.Messaging.wps1node1.0/SystemOut.log`

     - You will see the message shown in Example 7-4. (Always consider the last state of the ME in the log.)

Example 7-4 Verifying the state of the MEs on wps1node1

```
CWSID0016I: Messaging engine
RMRS.Messaging.000-SCA.APPLICATION.wpscell01.Bus is in state
Started.
CWSID0016I: Messaging engine
RMRS.Messaging.000-SCA.SYSTEM.wpscell01.Bus is in state
Started.
```

```
CWSID0016I: Messaging engine
RMRS.Messaging.000-BPC.wpscell01.Bus is in state Joined.
CWSID0016I: Messaging engine
RMRS.Messaging.000-CEI.wpscell01.BUS is in state Joined.
```

- On node wps1node2, review the following log file:

  `/opt/ibm/WebSphere/ProcServer/profiles/wps1node2/logs/RMRS.Messaging.wps1node2.0/SystemOut.log`

  You will see the message shown in Example 7-5. (Always consider the last state of the ME in the log.)

Example 7-5 Verifying the state of the MEs on wps1node2

```
CWSID0016I: Messaging engine
RMRS.Messaging.000-BPC.wpscell01.Bus is in state Started.
CWSID0016I: Messaging engine
RMRS.Messaging.000-CEI.wpscell01.BUS is in state Started.
CWSID0016I: Messaging engine
RMRS.Messaging.000-SCA.SYSTEM.wpscell01.Bus is in state
Joined.
CWSID0016I: Messaging engine
RMRS.Messaging.000-SCA.APPLICATION.wpscell01.Bus is in state
Joined.
```

– Verify the MEs on the second messaging cluster (RMRS.Messaging2):

- On node wps1node2, review the following log file:

  `/opt/ibm/WebSphere/ProcServer/profiles/wps1node1/logs/RMRS.Messaging2.wps1node2.0/SystemOut.log`

  You will see the message shown in Example 7-6. (Always consider the last state of the ME in the log.)

Example 7-6 Verifying the state of the MEs on wps1node1

```
CWSID0016I: Messaging engine
RMRS.Messaging2.000-SCA.APPLICATION.wpscell01.Bus is in state
Started.
CWSID0016I: Messaging engine
RMRS.Messaging2.000-SCA.SYSTEM.wpscell01.Bus is in state
Started.
CWSID0016I: Messaging engine
RMRS.Messaging2.000-BPC.wpscell01.Bus is in state Joined.
```

Archived

- On node wps1node2, review the following log file:

  ```
  /opt/ibm/WebSphere/ProcServer/profiles/wps1node2/logs/RMRS.Messaging2.wps1node2.0/SystemOut.log
  ```

  You will see the message shown in Example 7-7. (Always consider the last state of the ME in the log.)

  Example 7-7 Verifying the state of the MEs on wps1node2

  ```
  CWSID0016I: Messaging engine
  RMRS.Messaging2.000-SCA.APPLICATION.wpscell01.Bus is in state
  Joined.
  CWSID0016I: Messaging engine
  RMRS.Messaging2.000-SCA.SYSTEM.wpscell01.Bus is in state
  Joined.
  CWSID0016I: Messaging engine
  RMRS.Messaging2.000-BPC.wpscell01.Bus is in state Started.
  ```

3. Validate the Business Process Choreographer configuration on the new application cluster (RMRS.AppTarget2):

   a. Install the BPC installation verification application from the following path:

      ```
      /opt/ibm/WebSphere/ProcServer/installableApps/bpcivt.ear
      ```

      In the installation wizard, map it to the RMRS.AppTarget2 cluster and to the Web server.

      Note: You can install only one instance of the BPC installation verification application, so it may need to be uninstalled from the original application target cluster to proceed.

   b. When the installation is finished, select the **BPCIVTApp** from the enterprise applications list and click **Start**.

   c. Once the application is started, connect to it from your Web browser. In our environment the URL is:

      ```
      http://wps1node1/bpcivt
      ```

   d. If the BPC configuration was successful, you will see output similar to Example 7-8.

      Example 7-8 BPCIVT output

      ```
      Looking up the HumanTaskManager API EJB...
      Querying the originating task template...
      Creating the originating task...
      Creating the input message for the originating task...
      ```

Archived

```
Starting the originating task...
Querying the participating task...
Participating task not yet created, waiting 1s...
Participating task not yet created, waiting 1s...
Claiming the participating task...
Checking the input data...
Completing the participating task...
Checking the output data...
Output message of starter oTask not yet available, waiting 1s...
Output message of starter oTask not yet available, waiting 1s...
Output message of starter oTask not yet available, waiting 1s...
Deleting the originating task ...
Passed
Done.
```

Archived

8

WebSphere Business Monitor: Remote Messaging and Remote Support topology

This chapter provides full instructions for augmenting a Remote Message and Remote Support (RMRS) topology for WebSphere Process Server V7.0 with WebSphere Business Monitor V7.0. To create this topology, we extend an existing WebSphere Process Server RMRS topology with a fourth cluster to run the Web components of WebSphere Business Monitor. This topology has very good scalability for a solution where Business Process Management (BPM) monitoring is to be included with an existing WebSphere Process Server infrastructure.

It is a building block to extend on as your requirements grow. It is also robust enough to be used for production environments. In this scenario we build this topology within a single cell with two nodes. The new Web cluster for WebSphere Business Monitor components will be managed by the same two nodes.

© Copyright IBM Corp. 2010. All rights reserved.

8.1 Topology summary

Figure 8-1 shows a diagram of the starting Remote Messaging and Remote Support topology that we augment in this chapter. The creation of this topology is described in detail in Chapter 5, “WebSphere Process Server: Remote Messaging and Remote Support” on page 149.

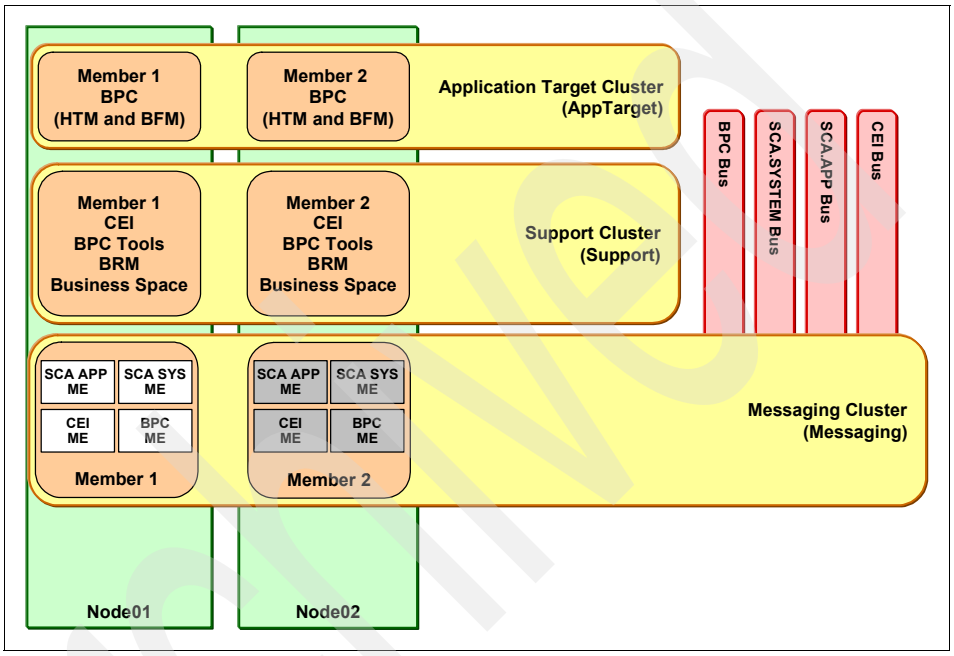


Figure 8-1 Initial Remote Messaging and Remote Support topology

Archived

Figure 8-2 shows a diagram of the Remote Messaging and Remote Support topology extended with WebSphere Business Monitor. This is referred to as Remote Messaging, Support and Web (RMSW).

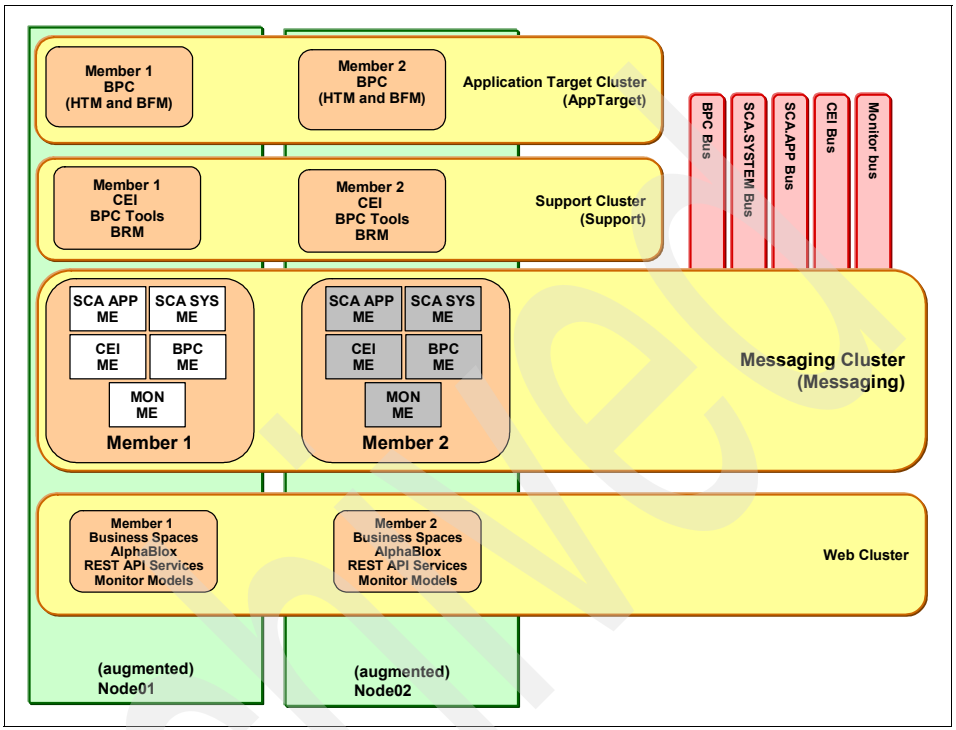


Figure 8-2 RMRS extended with Web cluster for WebSphere Business Monitor

In this topology you extend the existing WebSphere Process Server three-cluster pattern with one more cluster to run WebSphere Business Monitor components and Business Space. The usage of the clusters is as follows:

- A WebSphere Process Server Application Cluster to support WebSphere Process Server applications and mediations. It also runs the business flow manager (BFM) and human task manager (HTM). A WebSphere Process Server support cluster to run the Common Event Infrastructure (CEI), the business rules manager, the Business Process Choreographer (BPC) Explorer, and the Business Process Choreographer reporting function, the action services, the data movement services, and the emitter services.

Archive

- A messaging cluster to support the messaging engine infrastructure that serves both WebSphere Process Server and WebSphere Business Monitor messages.
- A WebSphere Business Monitor Web cluster to run all the services necessary to support WebSphere Business Monitor functionalities:
  - Business Space dashboards
  - Representational State Transfer (REST) service
  - Alphablox dimensional service
  - And so forth

Here is a list of the clusters and the applications that will be deployed to them.

- wbm.Web
  - AlphabloxPlatform
  - BSpaceEAR_wbm.Web
  - BSpaceWebformsEnabler_wbm.Web
  - BusinessSpaceHelpEAR_wbm.Web
  - IBM_WBM_MOBILE_DASHBOARD
  - WBMDashboardWeb_wbm.Web
  - mm.was_wbm.Web
  - AppScheduler
  - ApplicationStudio
  - HelloWorldMMApplication
- wps.AppTarget
  - BPEContainer_wps.AppTarget
  - HelloWorldModuleApp
  - RemoteAL61
  - TaskContainer_wps.AppTarget
  - persistentLkMgr
  - sca.sib.mediation
  - wpsFEMgr_7.0.0
- wps.Support
  - BPCECollector_wps.Support
  - BPCExplorer_wps.Support
  - BPMAdministrationWidgets_wps.Support
  - BusinessRulesManager_wps.Support
  - BusinessRules_wps.Support
  - HumanTaskManagementWidgets_wps.Support
  - persistentLkMgr
  - sca.sib.mediation
  - wesbWidget_wps.Support
  - IBM_WBM_ACTIONSERVICES

Archived

  - IBM_WBM_DATA_SERVICES
  - IBM_WBM_EMITTER_SERVICES
- wps.Messaging
  - AppScheduler
  - persistentLkMgr
  - sca.sib.mediation

You can create this topology by using:

- The Integrated Solution Console (ISC) and template guided activities
- WebSphere Administration (wsadmin) scripting.

Guidelines on this topology

Choose this topology if:

- You are extending an existing WebSphere Process Server to include WebSphere Business Monitor without a lot of changes to the existing environment.
- You want to administer both your WebSphere Process Server and WebSphere Business Monitor environments from the same deployment manager.
- You want to use a single messaging engine infrastructure to get events from WebSphere Process Server to Monitor.
- You are building a medium-sized environment for production that is scalable.

8.2 Prerequisites for creating this topology

This chapter assumes that you have already created a topology as described in Chapter 5, “WebSphere Process Server: Remote Messaging and Remote Support” on page 149. In addition, the following tasks must be completed:

- Install WebSphere Business Monitor binaries on all three WebSphere Process Server nodes: the deployment manager and the two custom nodes, as described in 3.2, “Software installation” on page 57.
- Create the monitor database and a new messaging schema to the MEDB (MONME), as described in 3.3.8, “The MONITOR database” on page 89.

Archived

8.3 Extending the topology

This section provides instructions for extending an existing WebSphere Process Server topology with WebSphere Business Monitor capabilities. Once you have installed the monitor binaries on all the three nodes and augmented each existing WebSphere Process Server profile with a WebSphere Business Monitor profile increment your topology will have the added capability but you still administer everything from the same deployment manager.

8.3.1 Augment the deployment manager profile

We must augment the deployment manager profile for it to recognize the new capability of WebSphere Business Monitor from the nodes that it administers. Refer to the `wpm_dmgr_augm.resp` file in Appendix A, "Additional material" on page 421.

Note: References to specific locations, nodes, names, and so on, are for WebSphere Process Server, as this is an existing environment that you are extending. The template is augmenting the cell to give it WebSphere Business Monitor capabilities.

Always check your specific environment template and response file values.

In this scenario, the augmentation template to be used is *wbmonitor/dmgr*.

1. Ensure that your deployment manager is stopped:

   ```
   ./stopManager.sh
   ```

2. Start the deployment manager augmentation by issuing the commands:

   ```
   cd /opt/ibm/WebSphere/ProcServer/bin
   ./manageprofiles.sh -response /tmp/wpm_dmgr_augm.resp.resp
   ```

   The response should say:

   ```
   INSTCONFSUCCESS: Profile augmentation succeeded.
   ```

3. Restart the deployment manager:

   ```
   ./startManager.sh
   ```

8.3.2 Augment the nodes

Augment each regular WebSphere Process Server node with the capabilities of WebSphere Business Monitor. Refer to the `wpm_node_augment.resp` file in Appendix A, "Additional material" on page 421.

Retrieve the sample template `wpm_node_augment.resp` from Appendix A, “Additional material” on page 421. Copy it into a temporary directory and modify the template with values for each of the nodes in this environment.

1. Stop the node agent:

   ```
   ./stopNode.sh -username XXX -password itso4you
   ```

2. Issue the command to augment WebSphere Process Server with WebSphere Business Monitor:

   ```
   cd /opt/ibm/WebSphere/ProcServer/bin
   ./manageprofiles.sh -response /tmp/wpm_node_augment.resp
   INSTCONFSUCCESS: Profile augmentation succeeded.
   ```

3. Restart the node agent:

   ```
   ./startNode.sh
   ```

4. As a quick verification, you can log into the Integrated Solutions Console and see the new WebSphere Business Monitor V7.0.0.0 in the Node Version information column, as shown in Figure 8-3.

wpmdmgr	itsodmgr	ND 7.0.0.7 Process Choreographer 7.0.0.1 SCA FEP 1.0.1.1 WBM 7.0.0.0 WPS 7.0.0.1 XML FEP 1.0.0.1
wps2node1	wpmnode1	ND 7.0.0.7 Process Choreographer 7.0.0.1 SCA FEP 1.0.1.1 WBM 7.0.0.0 WPS 7.0.0.1 XML FEP 1.0.0.1
wps2node2	wpmnode2	ND 7.0.0.7 Process Choreographer 7.0.0.1 SCA FEP 1.0.1.1 WBM 7.0.0.0 WPS 7.0.0.1 XML FEP 1.0.0.1

Figure 8-3 Dmgr and node version information with monitor capability

8.4 Configuring the topology using deployment environments in the Integrated Solutions Console

This section provides instructions for extending a WebSphere Process Server Remote Messaging and Remote Support topology with a WebSphere Business Monitor Web cluster for it to deliver the monitoring capability. The principal components in the topology include a database server (DB2), an Lightweight

Directory Access Protocol (LDAP) server, a Web server, and two nodes to support the clustering required. Once you have completed these instructions, you must complete the steps in 8.5, “Post-creation configuration and verification” on page 285, to ensure that the environment that you built is ready to run solutions end to end.

Create the clusters to support WebSphere Business Monitor

To create a cluster to support the WebSphere Business Monitor functionalities:

1. Log in to Integrated Solutions Console.
2. Change the ME data source for WebSphere Business Monitor since we are using a separate database for all messaging in this scenario. This is done because in the response file there is only one parameter provided and MONDB is specified as the database name.
   a. Click **Resources** → **Datasources** → **Monitor_ME_Database** and change the database name, as shown in Figure 8-4.

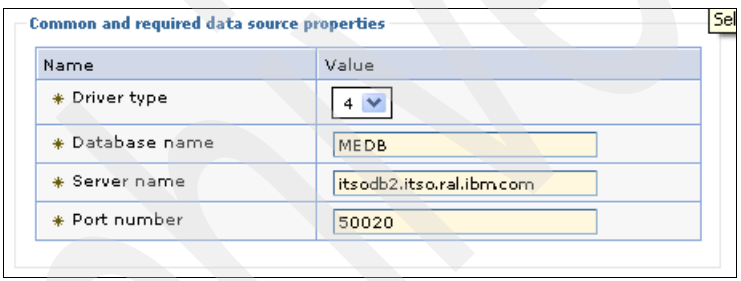


Figure 8-4 Modify ME database name for Monitor

   b. Test the connection.
   c. Save your change.

Archived

3. Click **Servers** → **Clusters** → **WebSphere application server clusters** → **New**:

   a. Enter the cluster name, as shown in Figure 8-5, and click **Next**.

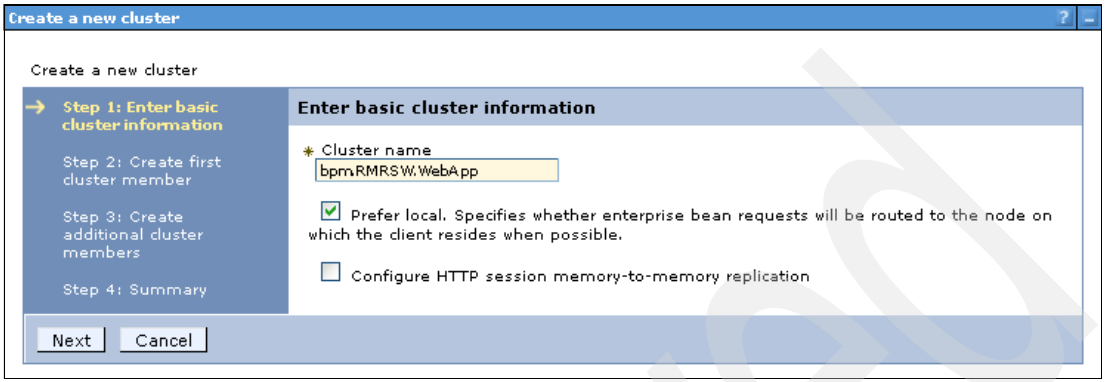


Figure 8-5 Creating a new cluster

Archived

b. Enter the member name, select the template of type **defaultProcessServer_defaultWBM**, and the node that it supports, as shown in Figure 8-6. Click **Next**.

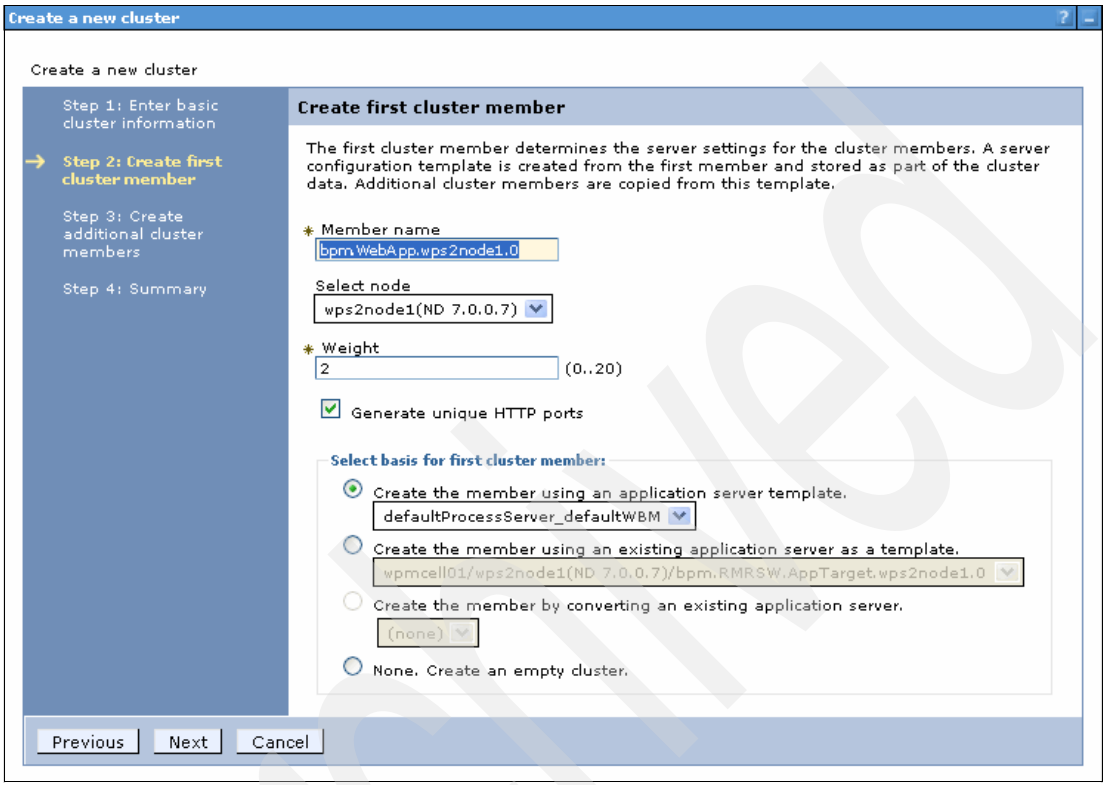


Figure 8-6 Enter member name

Archived

c. Add another member to support the second node. You should now have what is shown in Figure 8-7.

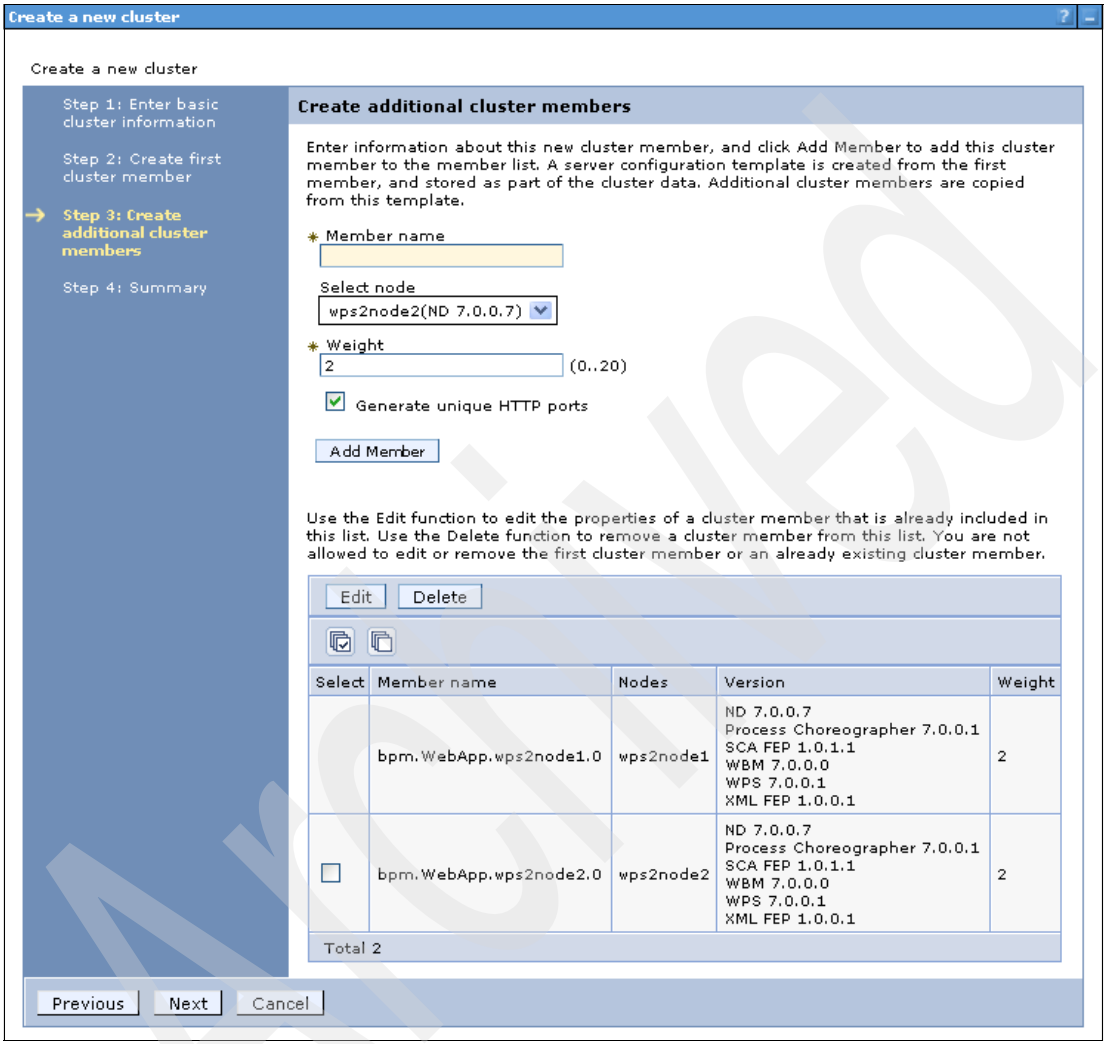


Figure 8-7 Members supporting the nodes

d. Verify your selection and click **Finish** to create the cluster.

Archived

You now have the Web cluster needed to support WebSphere Business Monitor applications, as shown in Figure 8-8.

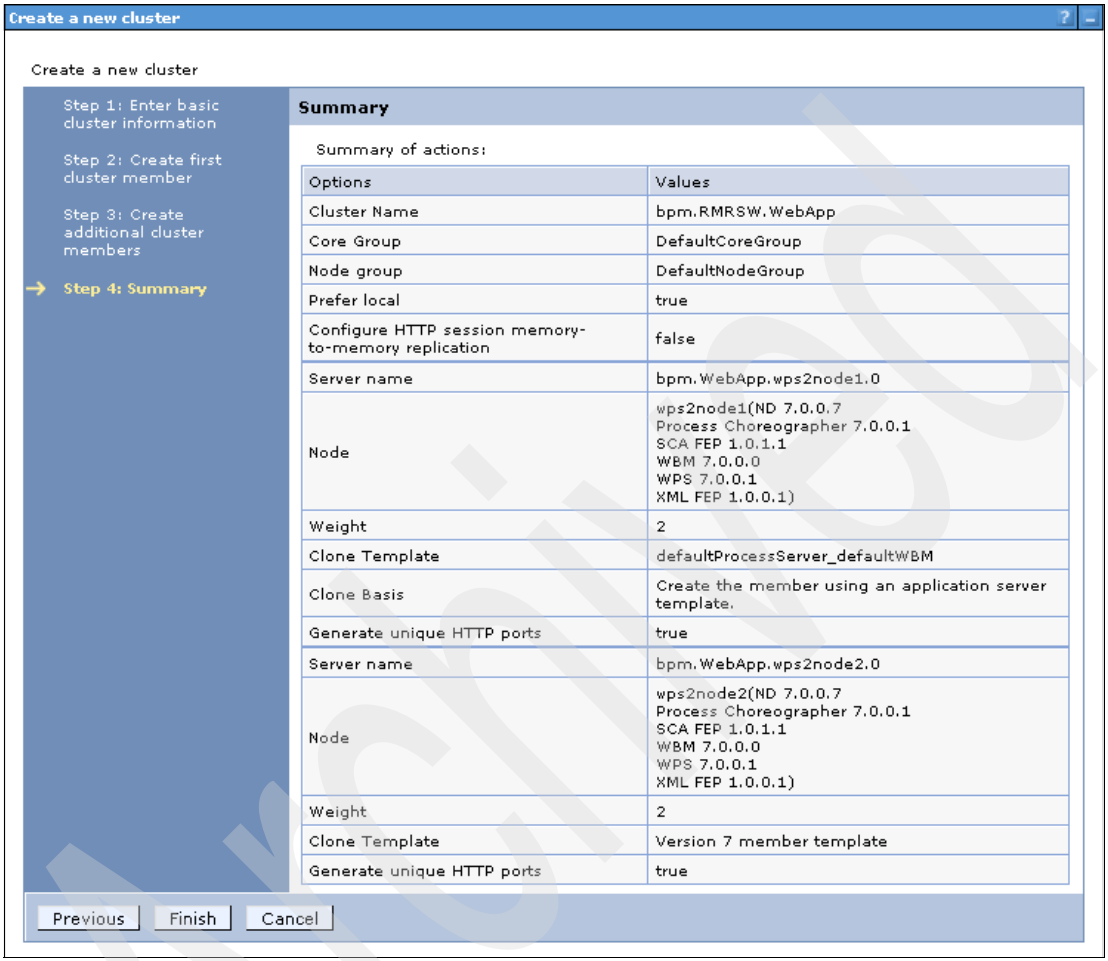


Figure 8-8 Create new Web cluster

Archive

4. Since Business Space will be hosted on the new Web cluster we must create a data source for it. To do this:
   a. Click **Resources** → **JDBC** → **Data sources** → **New**.
   b. In step 1, enter the data source information as shown in Figure 8-9.

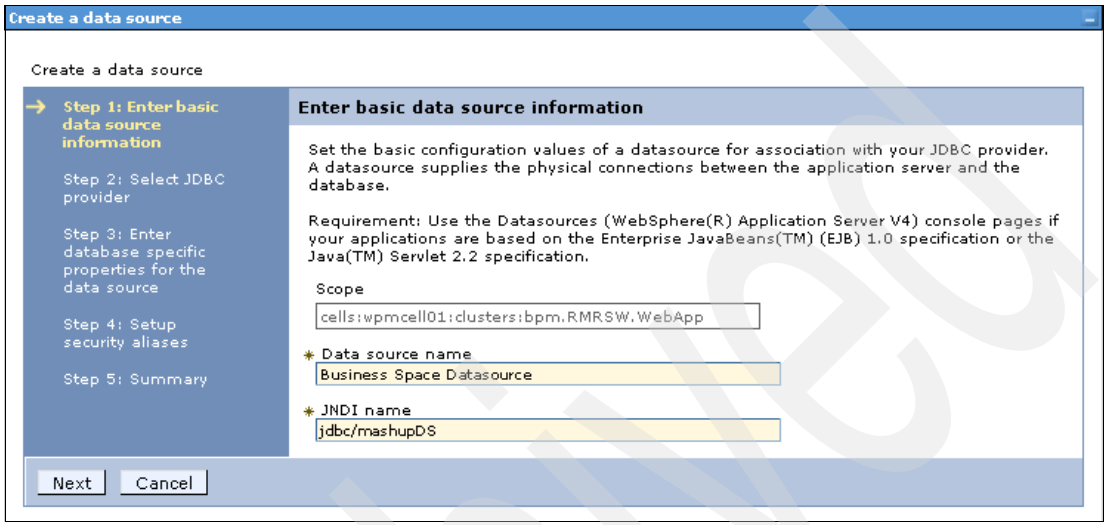


Figure 8-9 Creating a new data source

Archived

c. In step 2.1, enter the JDBC provider information, as shown in Figure 8-10.

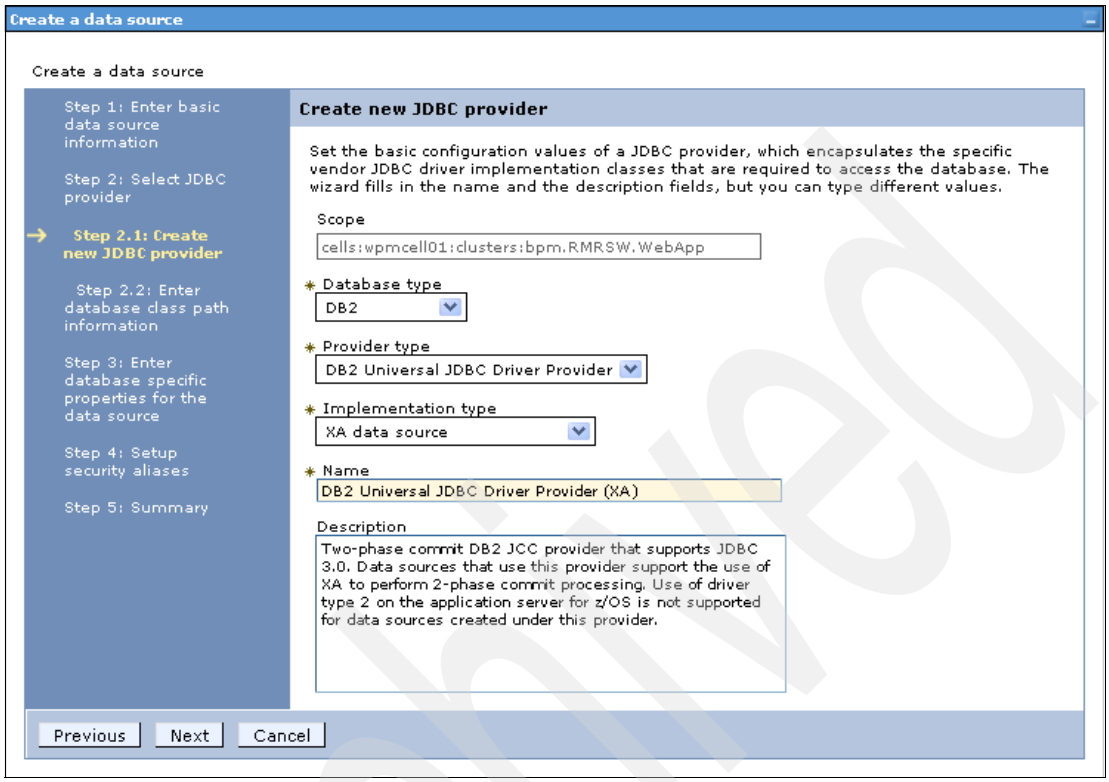


Figure 8-10 Database JDBC provider

Archived

d. In Step 2.2, enter the database class path information, as shown in Figure 8-11.

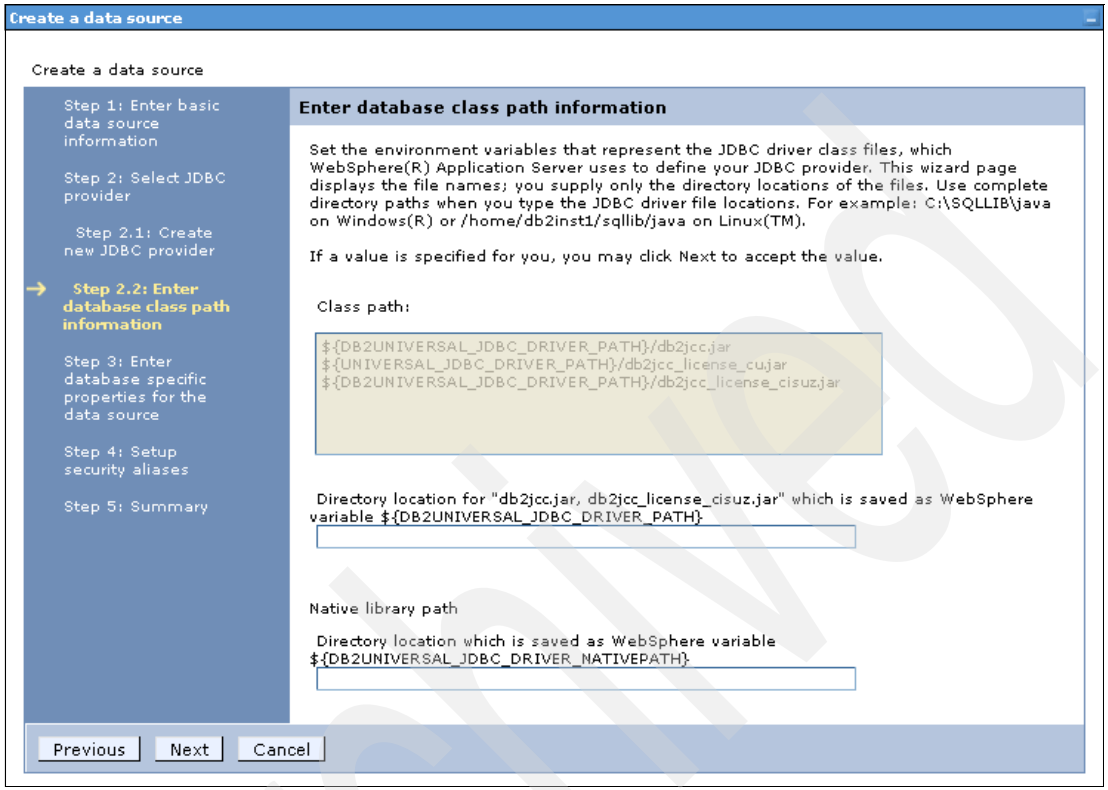


Figure 8-11 Database classpath information

Archive

e. In step 3, enter the database properties as shown in Figure 8-12.

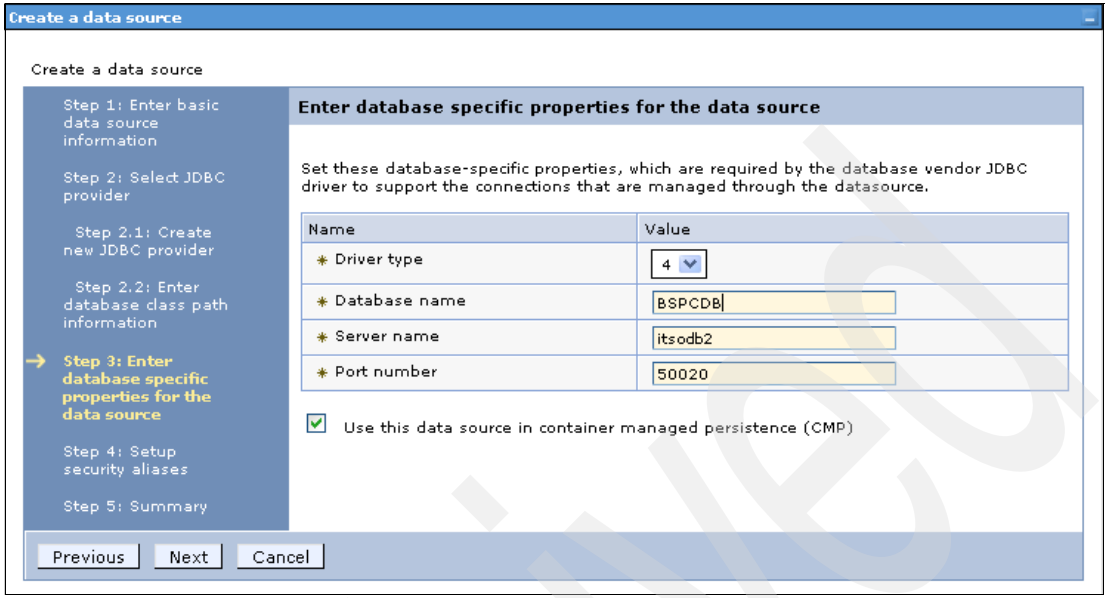


Figure 8-12 Database properties

Archived

f. In step 4, enter database security aliases as shown in Figure 8-13.

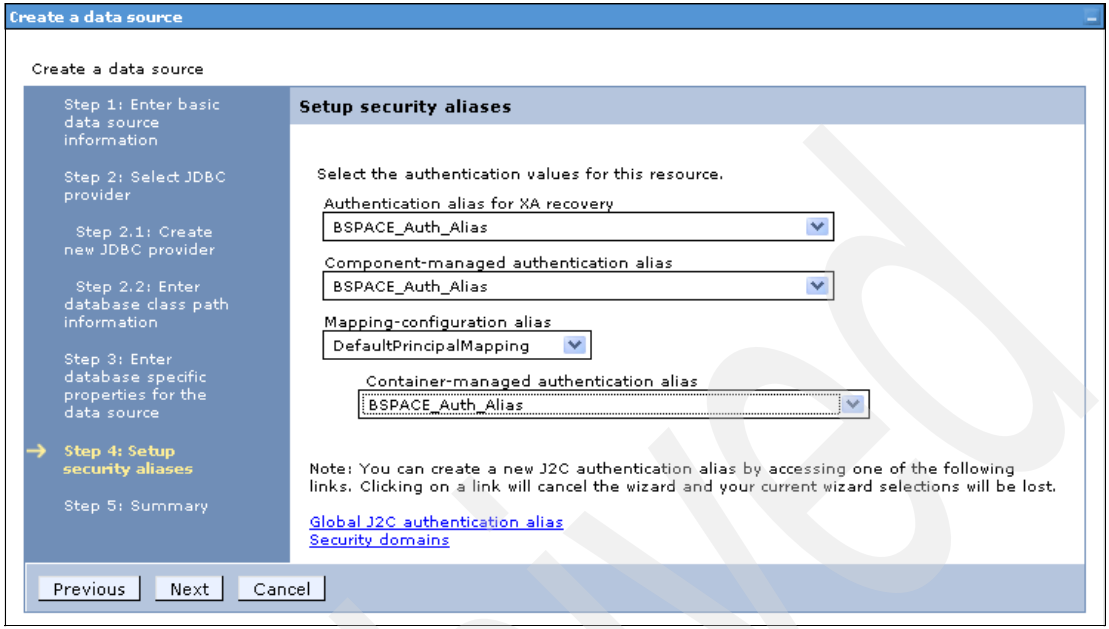


Figure 8-13 Database security aliases

g. Click **Apply** and **Save** your changes.

h. Test the connection for the new data source.

Archived

At this point you should have the data sources to support the new Web cluster created and the status of the four clusters should be as shown in Figure 8-14.

Figure 8-14 Four clusters in environment

5. Now that we have the new Web cluster, we redeploy the Business Space application to its target. To do so you must stop and uninstall these four enterprise applications supporting Business Space functionality:

   – BSpaceEAR_bpm.RMRSW.Support
   – BSpaceWebformsEnabler_bpm.RMRSW.Support
   – BusinessSpaceHelpEAR_bpm.RMRSW.Support
   – mm.was_bpm.RMRSW.Support

   Then re-deploy them to the new Web cluster by using the WebSphere Business Monitor configuration wizard later. To stop and uninstall an application:

   a. Click **Applications** → **Application Types** → **WebSphere enterprise applications**.

   b. Select the application.

   c. Click **Stop**.

d. Select the application and click **Uninstall**, as shown in Figure 8-15.

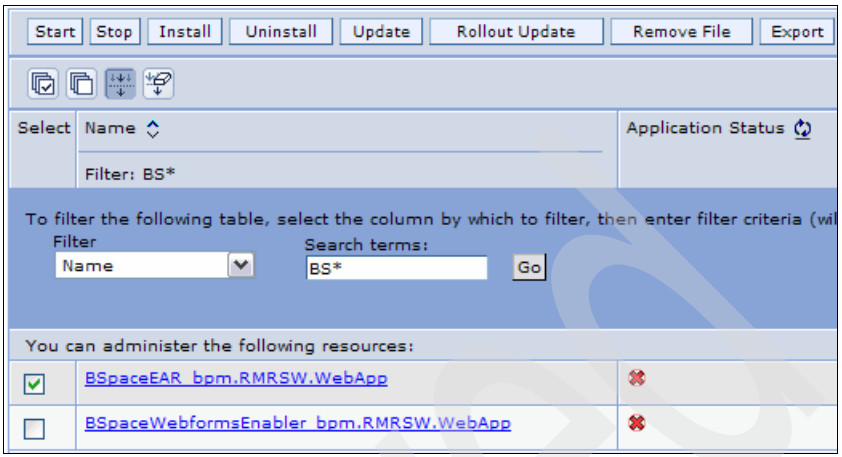


Figure 8-15 Uninstalling an application

e. Repeat steps b through d for the other three applications.

f. Save and synchronize your changes.

Archive

6. Redistribute all WebSphere Business Monitor components to the appropriate clusters using the configuration wizard:
   a. Click **Servers** → **WebSphere Business Monitor configuration**. You will see page showing the initial state of all applicable components, as shown in Figure 8-16.

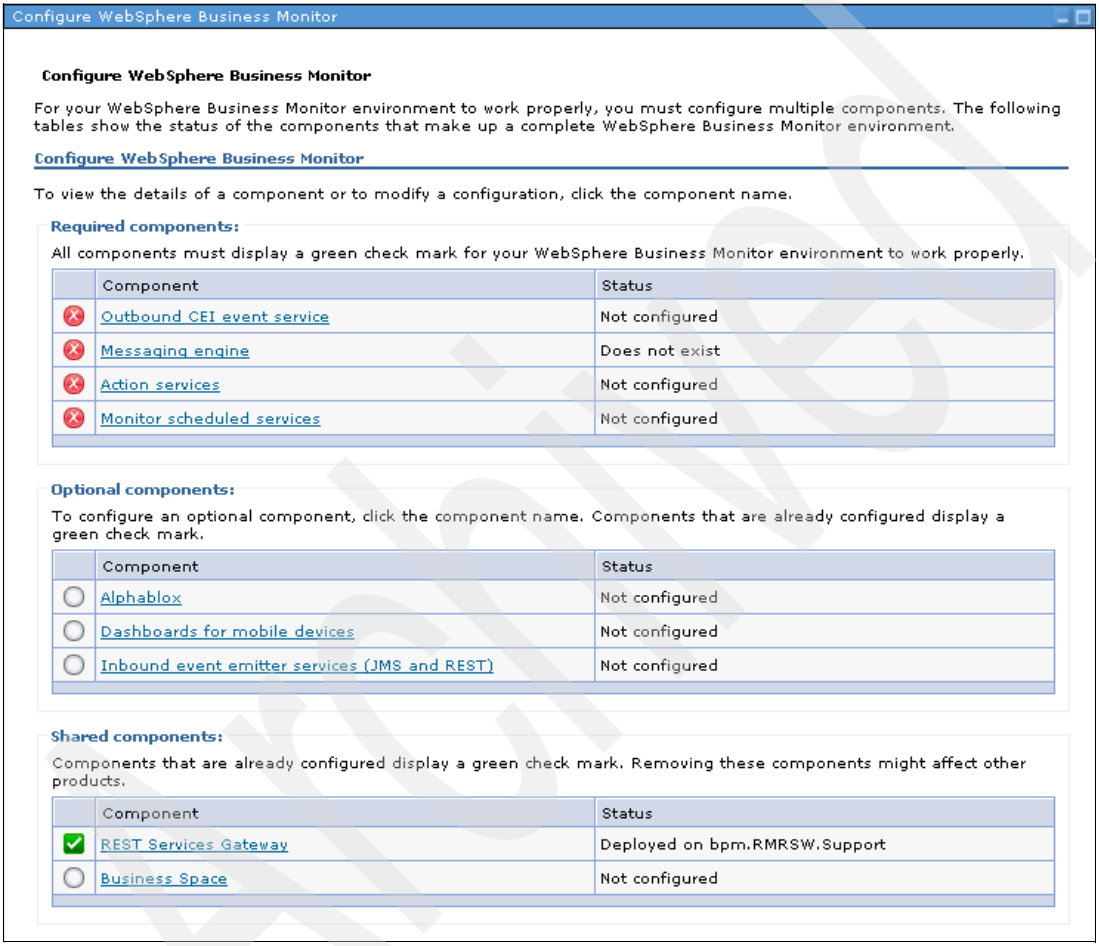


Figure 8-16 Initial state of all Monitor components

Deploy or configure the components in the order shown in Figure 8-16.

b. Click the **Outbound CEI event service** link, map it to run on the supportcluster, and click **Configure the Event Emitter Factory**, as shown in Figure 8-17.

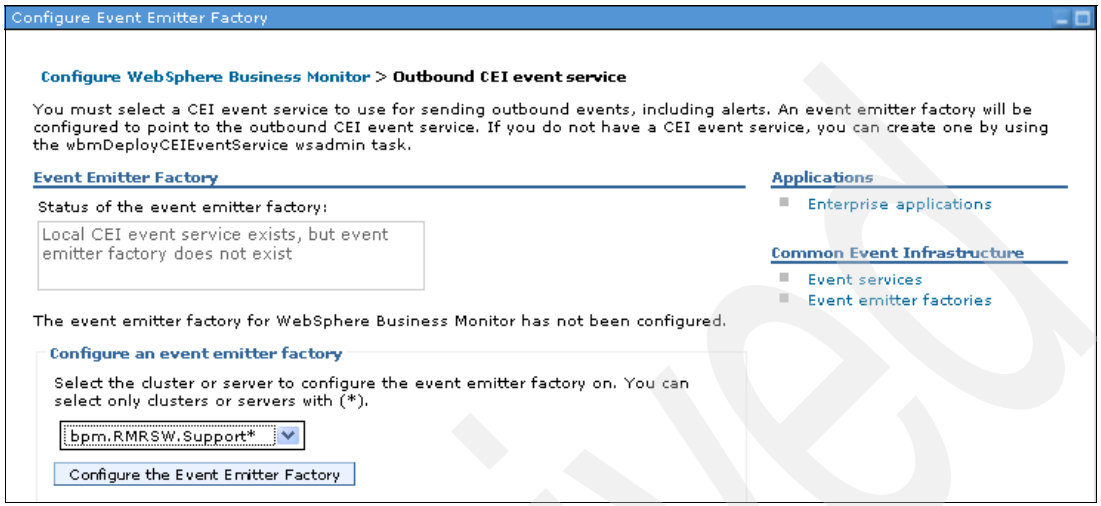


Figure 8-17 Configuring Outbound Event Emitter Factory

c. Click the **Messaging Engine** link to configure, as shown in Figure 8-18.

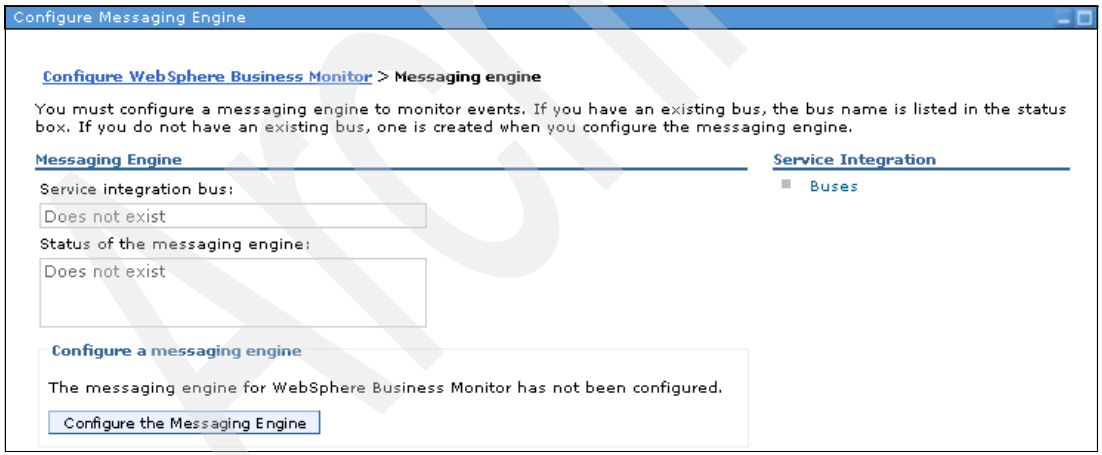


Figure 8-18 Configuring messaging engine

Archived

d. Select the messaging cluster to run the message engine, as shown in Figure 8-19.

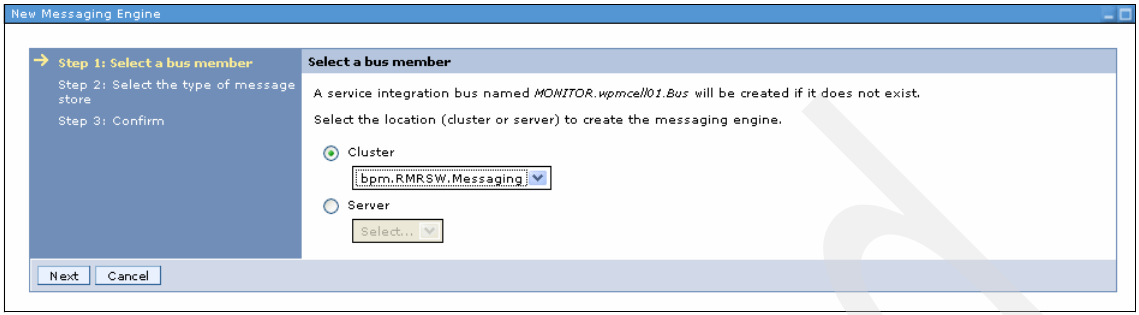


Figure 8-19 Configuring bus member for ME

e. Select **Data store** for the message engine. It is the only type supported in a clustered environment. See Figure 8-20.

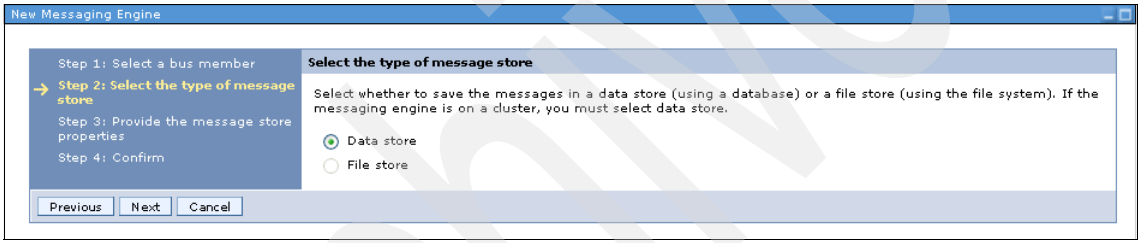


Figure 8-20 Data store for message engine

f. Enter database properties for the data store, as shown in Figure 8-21.

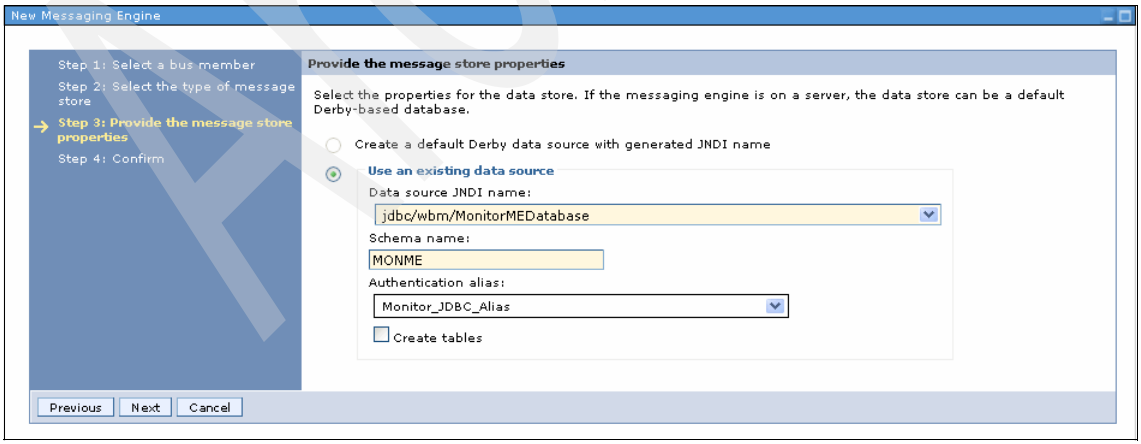


Figure 8-21 Data store properties

Archived

g. Click **Finish**.

h. Click the **Action Services** link, select the support target cluster, and click **Deploy Action Services**, as shown in Figure 8-22.

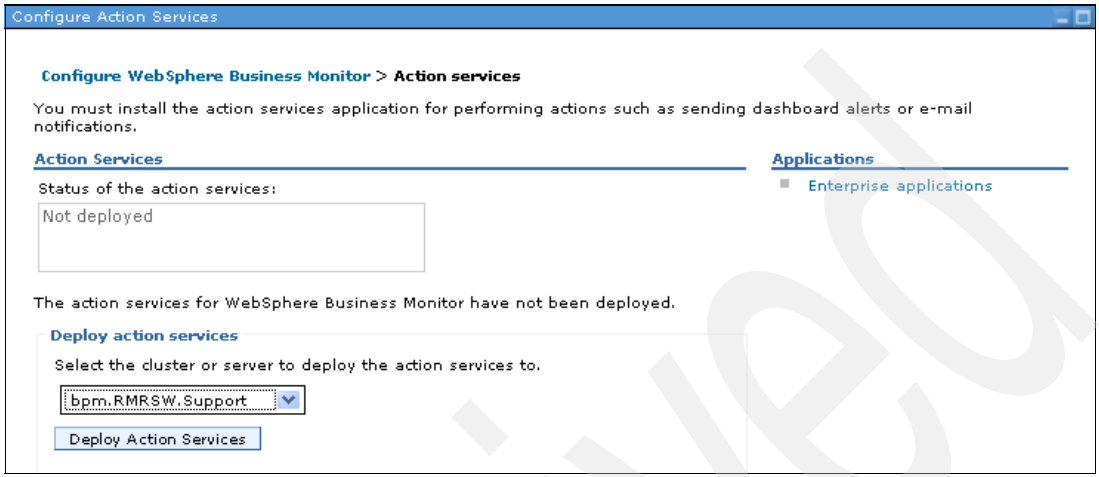


Figure 8-22 Deploying action services

i. Click the **Monitor Scheduled Services** link, select the **target** cluster, and click **Deploy Monitor Scheduled Services**, as shown Figure 8-23.

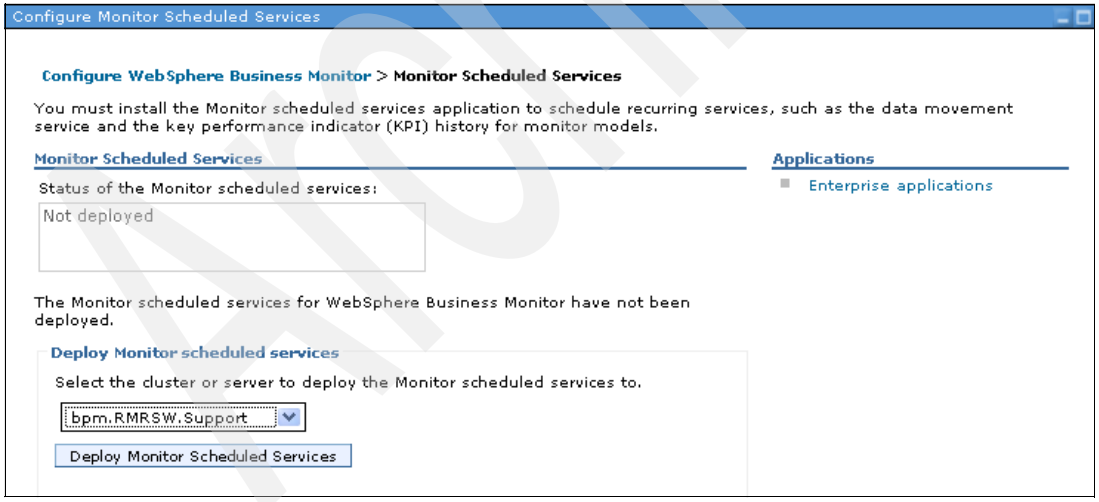


Figure 8-23 Deploy Monitor Scheduled Services

Archived

j. Click the **Alphablox service** link, enter the WebSphereadministrator user ID and password, select the **target** webapp cluster, and click **Deploy Alpahblox Service**, as shown in Figure 8-24.

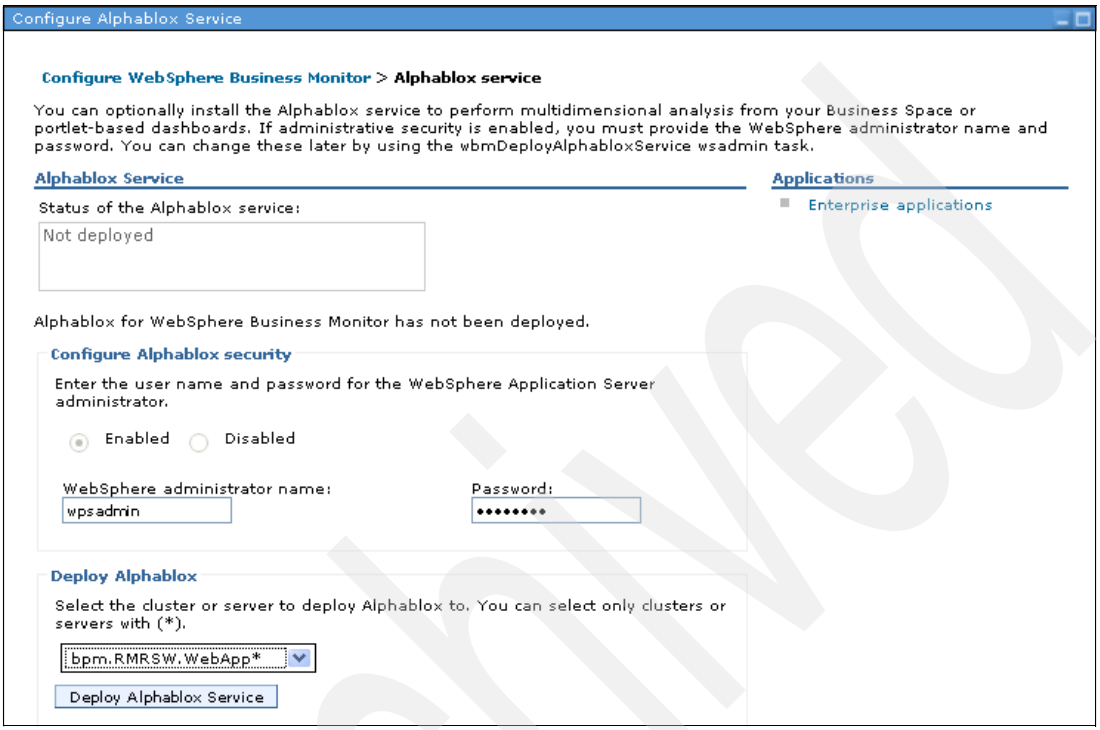


Figure 8-24 Deploy Alphablox

Archived

k. Click the **Dashboards for mobile devices** link, select the **target** webapp cluster, and click **Deploy the Dashboards for Mobile Devices**, as shown in Figure 8-25.

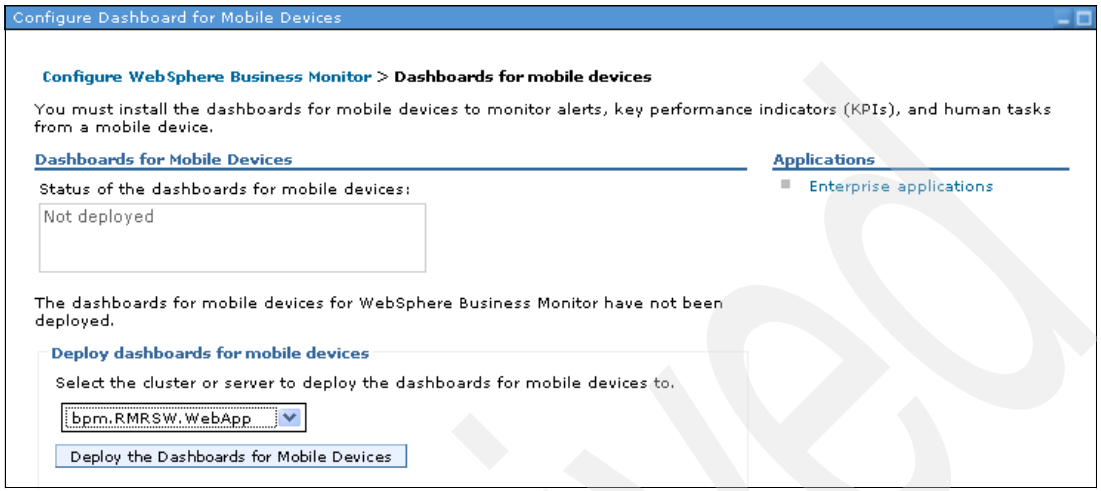


Figure 8-25 Deploy dashboards for mobile devices

l. Click the **Inbound Event Emitter Services (JMS and REST)** link, select the **target** webapp cluster, and click **Deploy Event Emitter Services**, as shown in Figure 8-26.

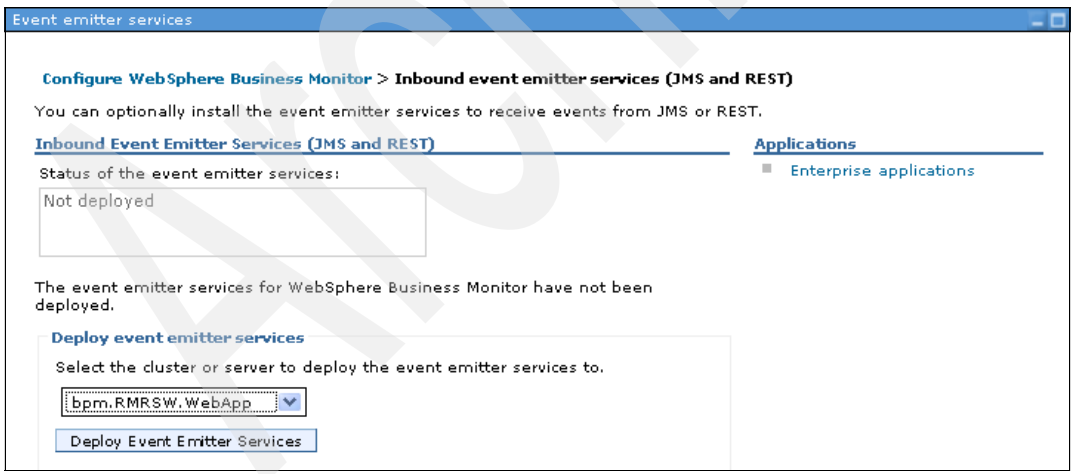


Figure 8-26 Deploy inbound event emitter service

m. Click **WebSphere application server clusters** → **bpm.RMRSW.WebApp** → **Business Space Configuration**.

Archived

n. Select **Install Business service**. Click **OK**. See Figure 8-27.

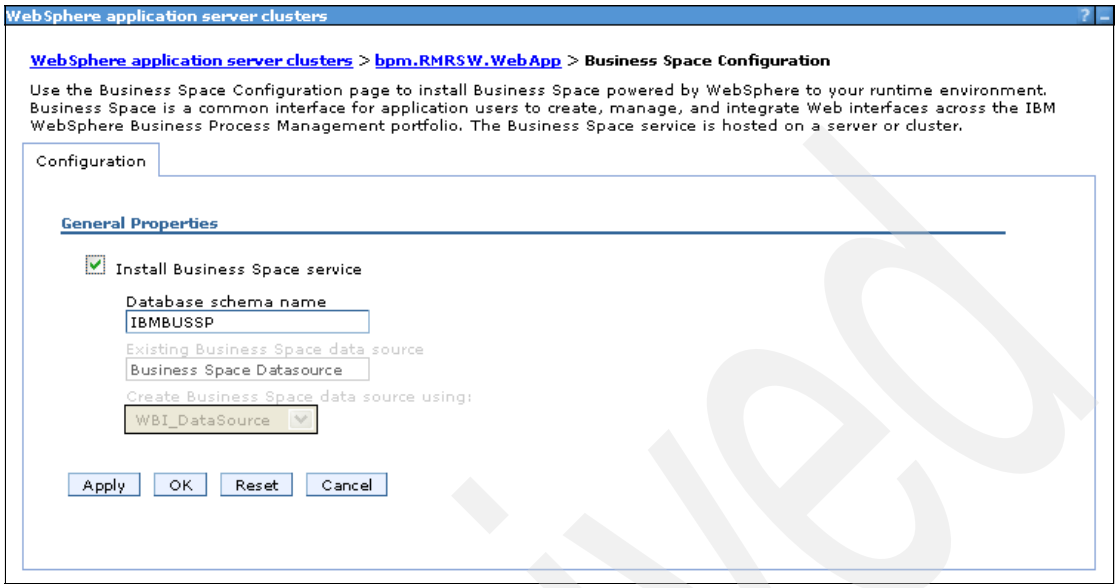
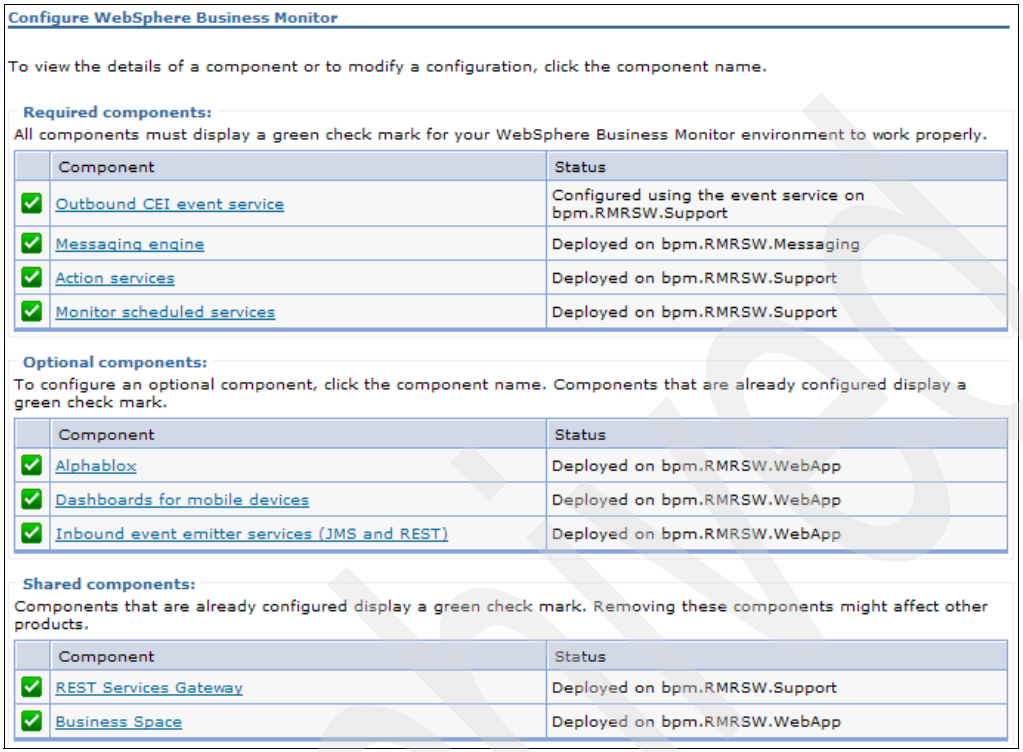


Figure 8-27 Deploy Business Space

Archived

The resulting deploy status of all the applicable Monitor components and their targets are displayed, as shown Figure 8-28.


Configure WebSphere Business Monitor

To view the details of a component or to modify a configuration, click the component name.

Required components:

All components must display a green check mark for your WebSphere Business Monitor environment to work properly.

Component	Status
Outbound CEI event service	Configured using the event service on bpm.RMRSW.Support
Messaging engine	Deployed on bpm.RMRSW.Messaging
Action services	Deployed on bpm.RMRSW.Support
Monitor scheduled services	Deployed on bpm.RMRSW.Support

Optional components:

To configure an optional component, click the component name. Components that are already configured display a green check mark.

Component	Status
Alphablox	Deployed on bpm.RMRSW.WebApp
Dashboards for mobile devices	Deployed on bpm.RMRSW.WebApp
Inbound event emitter services (JMS and REST)	Deployed on bpm.RMRSW.WebApp

Shared components:

Components that are already configured display a green check mark. Removing these components might affect other products.

Component	Status
REST Services Gateway	Deployed on bpm.RMRSW.Support
Business Space	Deployed on bpm.RMRSW.WebApp

Figure 8-28 Final status of all configured components supporting monitor

Changing REST services endpoints for BPEContainer and TaskContainer

This must be done when we use a Web server to route requests to these runtime containers. To change the REST endpoints, put the HTTP server host and port in the URL of the endpoint, for example, `https://wpmweb.itso.ral.ibm.com:443`. Detailed instructions are described in 5.5.3, “Configuring REST service endpoints” on page 179.

8.5 Post-creation configuration and verification

This section describes applicable tests to verify that the environment that you built is working correctly.

8.5.1 Verify server startup without errors

To do this:

1. Restart your deployment manager and all your node agents and check the log files for any errors.
2. Restart your clusters and check the logs for any errors.

8.5.2 Creating event database tables

Create and populate the event database tables. Refer to 3.3.7, “The EVENT database” on page 89.

8.5.3 Test data sources

Verify that you can access all the defined data sources without any errors. Log in to your deployment admin console, go to **Resources** → **JDBC** → **Data sources**, and select all data sources. Click **Test connection** to test your data sources. You should receive a success message for all data sources.

Archived

8.5.4 Verify virtual host settings and host aliases

Ensure that you have entries in your virtual hosts alias section for http and https ports of the application clusters now that you are routing all the requests through the Web server:

1. Go to the **Environment** → **Virtual hosts** → **default_host** link.
2. Go to **Host Aliases** and verify you have entries for all the ports that your webappcluster members use to present Web applications, as shown in Figure 8-29.
3. Restart the webapp cluster if you make any changes.

Host Name	Port
You can monitor the following resources:	
*	9081
*	80
*	9444
*	5060
*	5061
*	443
*	9080
*	9443
wpmnode1.itso.ral.ibm.com	9080
wpmnode1.itso.ral.ibm.com	80
wpmnode1.itso.ral.ibm.com	9443
wpmnode1.itso.ral.ibm.com	5060
wpmnode1.itso.ral.ibm.com	5061
wpmnode1.itso.ral.ibm.com	443
wpmnode2.itso.ral.ibm.com	9080
wpmnode2.itso.ral.ibm.com	80
wpmnode2.itso.ral.ibm.com	9443
wpmnode2.itso.ral.ibm.com	5060
wpmnode2.itso.ral.ibm.com	5061
wpmnode2.itso.ral.ibm.com	443
Total 20	

Figure 8-29 Default host ports

4. Navigate to **Servers** → **Server Types** → **Web Servers** and regenerate and republish the HTTP server plug-in. Select the check box to the left of the HTTPS server **webeserver1** and click **Generate Plug-in**, then click **Propagate Plug-in**.
5. Restart the HTTP server for changes to take effect by clicking **Stop**, then **Start**.

Archive

8.5.5 Test Alphablox admin page

To have access to all Alphablox components, WebSPhere security must be enabled. Browse to the following URL:

`http://<http_server>/AlphabloxAdmin`

The Alphablox admin page is displayed after logging in, as shown in Figure 8-30.

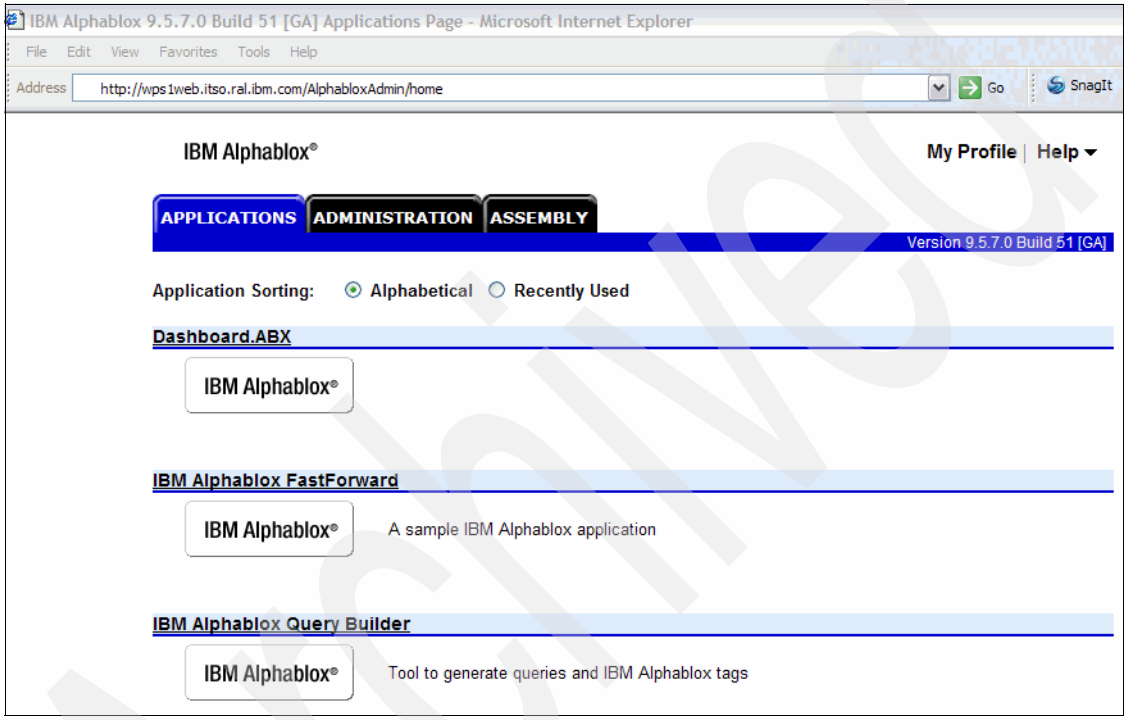


Figure 8-30 Alphablox login page

Archive

If you do not see the Alphablox admin page, verify that you mapped the applications shown in Figure 8-31 to your http server and your WebApp cluster and redistributed the IBM HTTP Server plug-in file.

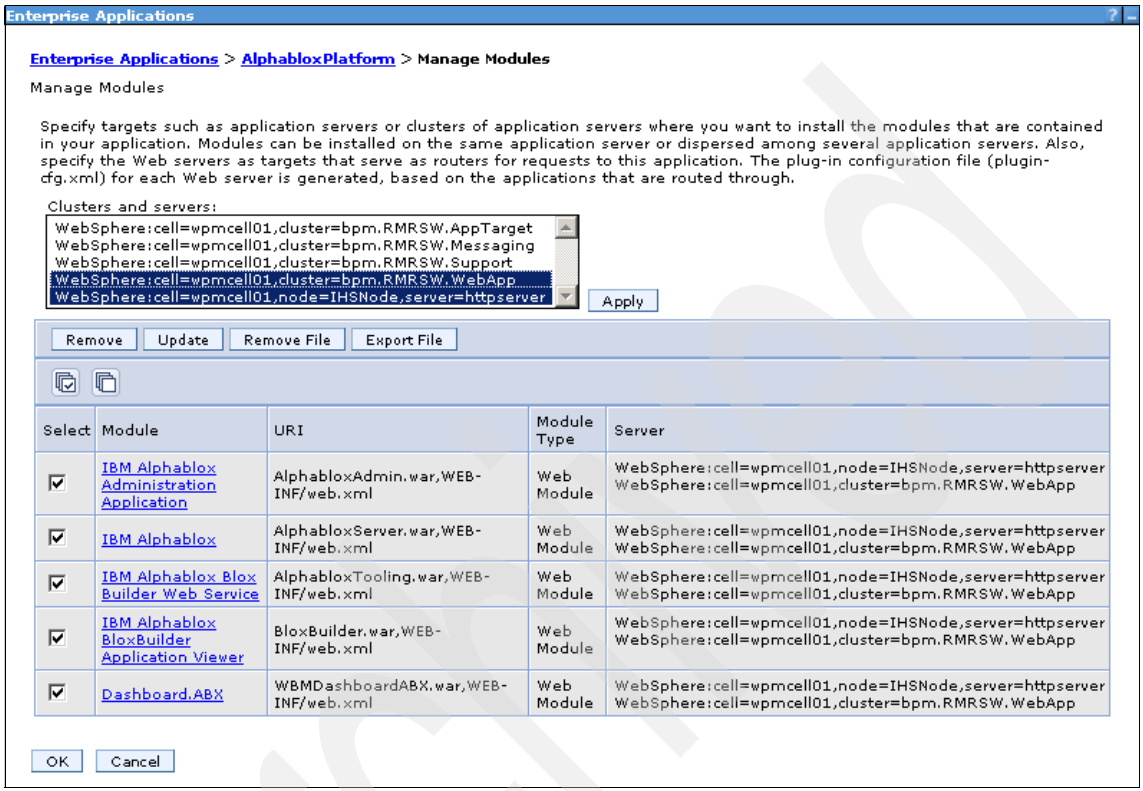


Figure 8-31 Map Alphablox modules to http server and wbm.Web cluster

8.5.6 Test Business Space

Open the URL `http://<http_server>/BusinessSpace` in a Web browser. You will see the Business Space welcome page after logging in, as shown in Figure 8-32.

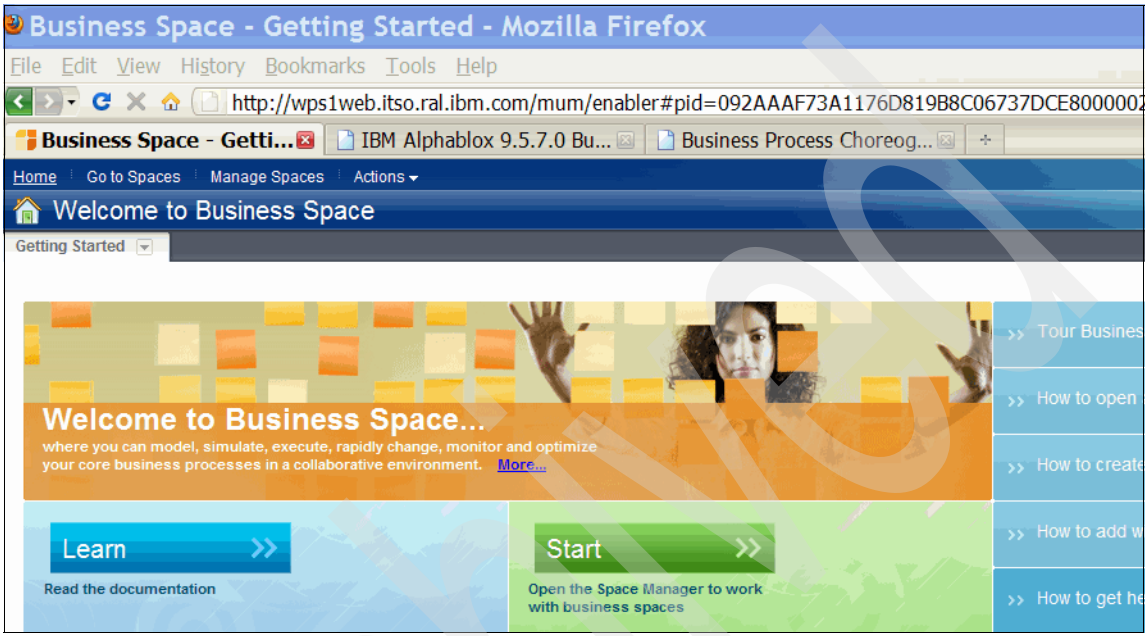


Figure 8-32 Business Space Getting Started page

If you do not see the Business Space welcome page, verify the following settings. Check whether the following applications are mapped to the http server and the WebApp cluster:

- BSpaceEAR_wbm.Web
- BSpaceWebformsEnabler_wbm.Web
- BusinessSpaceHelpEAR_wbm.Web
- mm.was_wbm.Web

Archive

8.5.7 Test Business Rule Manager

Open the URL `http://<http_server>/br` in your browser. You should see the Business Rules Manager page displayed after logging in, as shown in Figure 8-33.

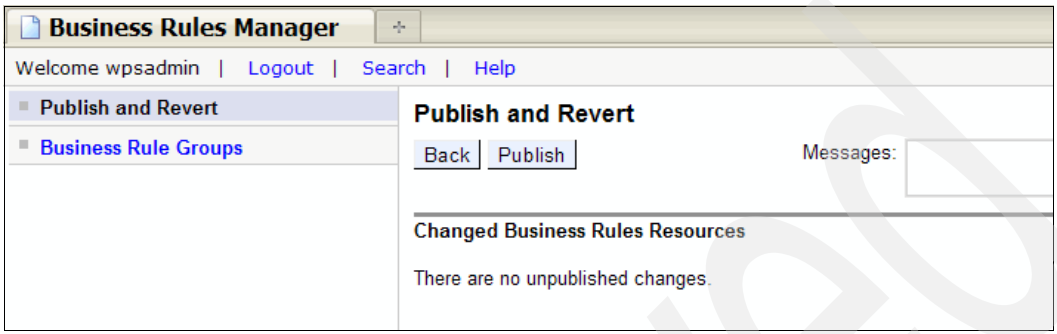


Figure 8-33 Business rules home page

If you do not see the page, verify that you mapped the application BusinessRulesManager_bpm.WebApp to your http server and your WebApp cluster.

Archived

8.5.8 Test the Business Process Choreographer

Open the URL `http://<http_server>/bpc` in your browser. You should see the BPC Explorer view, as shown in Figure 8-34.

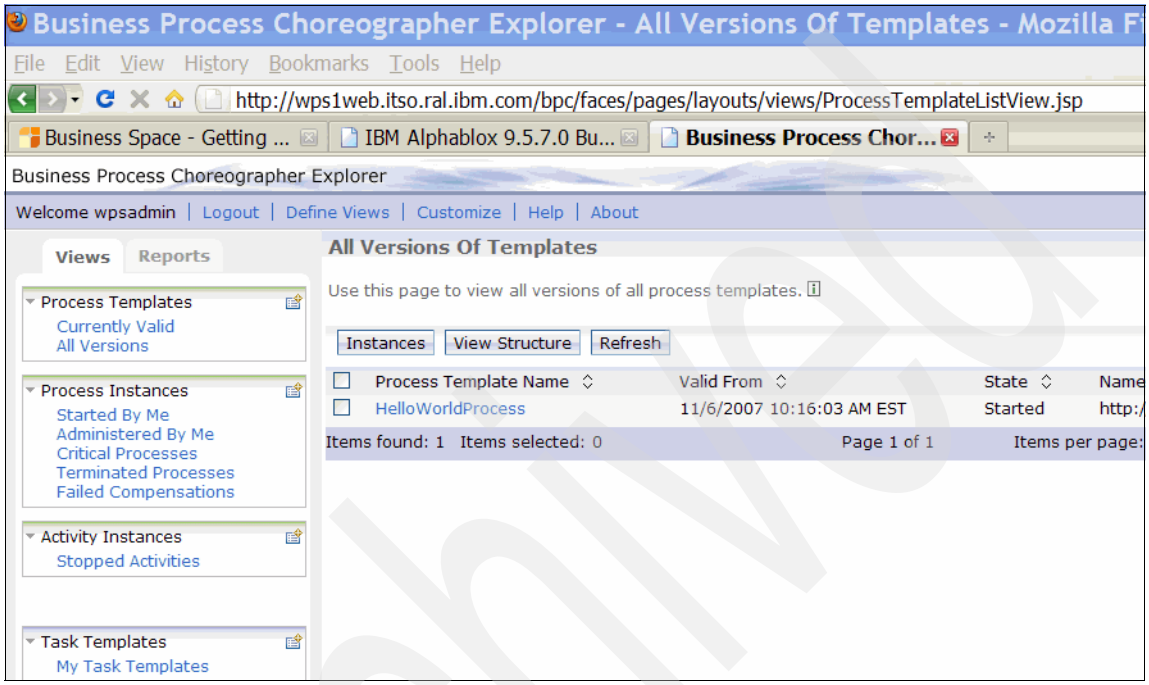


Figure 8-34 BPC Explorer page

If you do not see the BPC Explorer page, verify that you mapped the application BPCExplorer_wps.Support to your http server and your wps.Support cluster.

8.5.9 Test the human task manager

A verification test is explained in 4.5.8, “Verification: human task manager” on page 142.

8.5.10 Test WebSphere Business Monitor

This section describes how to test WebSphere Business Monitor functionality.

Prerequisites

To emit events from any Business Process Execution Language (BPEL) process, we must enable the Common Event Infrastructure logging in the Business Flow Manager:

1. Go to **Servers** → **Clusters** → **WebSphere application server clusters** and select the cluster in which the BPEL applications are running. In our environment, this is the wps.AppTarget.
2. Open the section named **Business Process Choreographer** under Business Integration and click **Business Flow Manager**. Activate the **Enable Common Event Infrastructure logging** check box under State Observers, as shown in Figure 8-35.

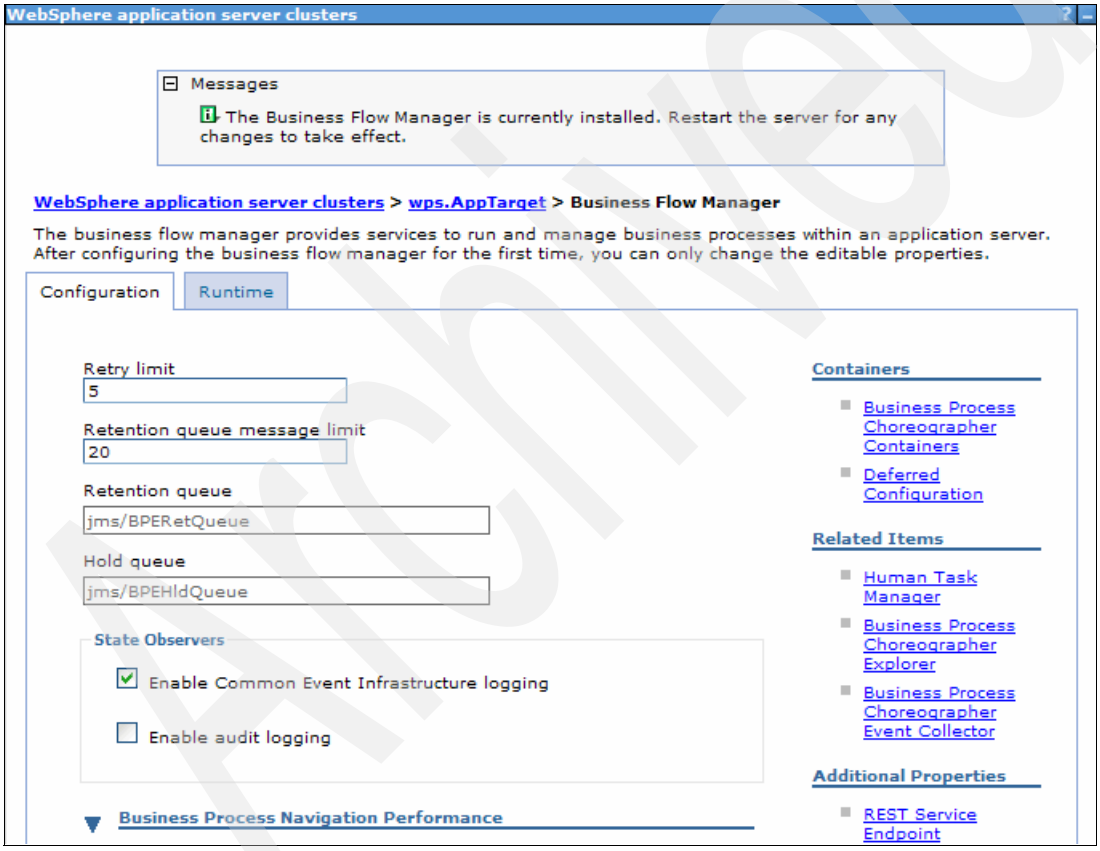


Figure 8-35 Enable Common Event Infrastructure logging for the Business Flow Manager

Repeat the above step for the Human Task Manager. Go back to your AppTarget cluster and select the **Human Task Manager** link in the Business Process Choreographer section under Business Integration.

Chck the **Enable Common Event Infrastructure logging** check box, as shown in Figure 8-36.

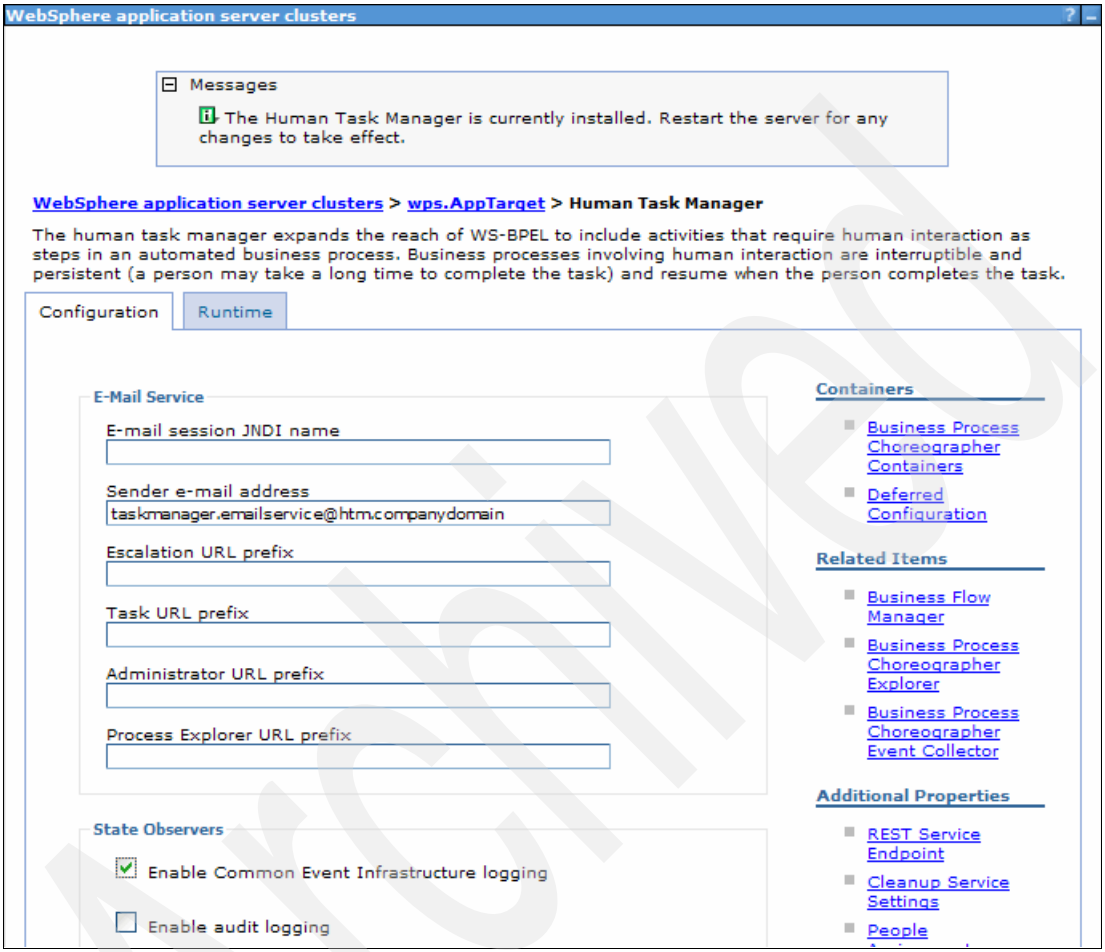


Figure 8-36 Enable Common Event Infrastructure logging for the human task manager

Archived

Troubleshooting: queue bypass transport is unavailable

When installing a monitor model, there may be an issue in step 13 (Select Monitor model CEI options) showing the error message as captured, as in Figure 8-37.

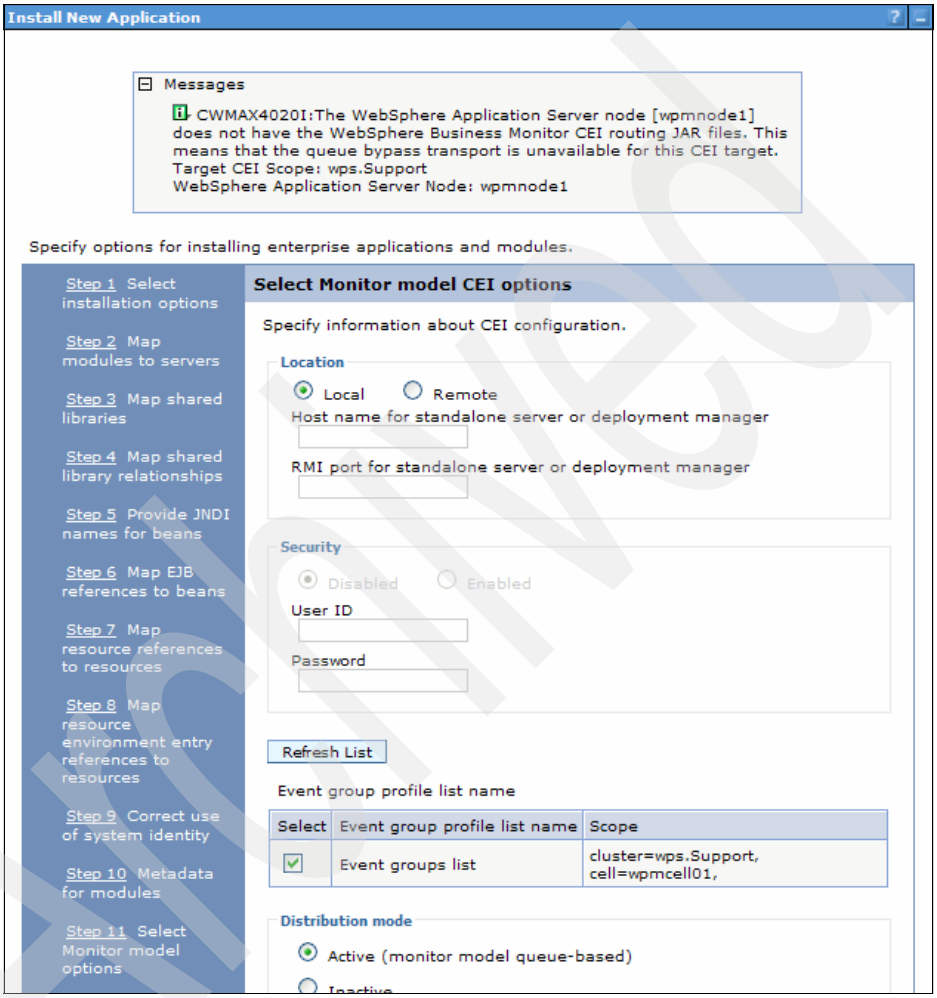


Figure 8-37 The monitor model installation issue

This error is not displayed if you have all members in your support cluster[1] up and running.

[1] The CEI server is running in the wps.Support cluster, and in Monitor V7.0.0 in step 13, the installer checks for the needed plug-ins on all defined members of this cluster. The check fails if all members are not running.

8.5.11 Verify the topology using a sample application

As a final test deploy the HelloWorld sample application provided in Appendix A, “Additional material” on page 421, to validate all components.

Appendix B, “Verification using the sample application” on page 423, describes the steps to install and test the HelloWorld application.

Archived

9

WebSphere Business Monitor: cross cell

This chapter describes how to set up a WebSphere Business Monitor topology based on the Remote Messaging, Support and Web topology pattern. WebSphere Process Server is deployed in a separate remote cell hosting the Common Event Infrastructure service (CEI).

This chapter discusses the following topics:

- The topology used in this chapter and the rationale for chosing this topology
- The software prerequisites and installation parameters used to build this topology
- Detailed step-by-step instructions about how to set up the topology using the Integrated Solutions Console in silent mode
- How to validate the topology by deploying a sample application

© Copyright IBM Corp. 2010. All rights reserved.

9.1 Topology summary

This section outlines the steps to install WebSphere Business Monitor using a Remote Messaging, Support and Web topology pattern. The installation steps were performed on three SUSE Linux Enterprise Server 10 64-bit servers. Figure 9-1 depicts the topology and shows the distribution of all the main WebSphere Business Monitor components.

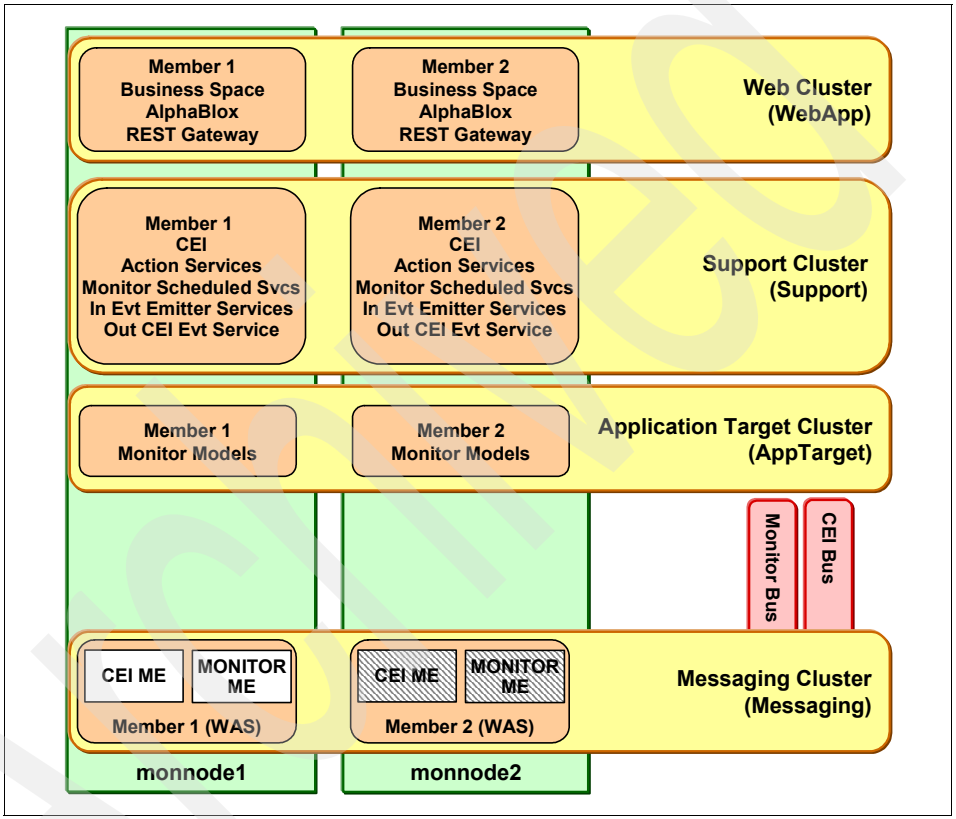


Figure 9-1 WebSphere Business Monitor Remote Messaging, Support and Web topology

Archive

In this topology WebSphere Business Monitor is installed in its own WebSphere Application Server cell. Common Business Events (CBEs) issued by a remote WebSphere Process Server instance using the CEI service are delivered to the monitor model application via the WebSphere Business Monitor database. This event delivery mode is called queue-bypass (Figure 9-2).

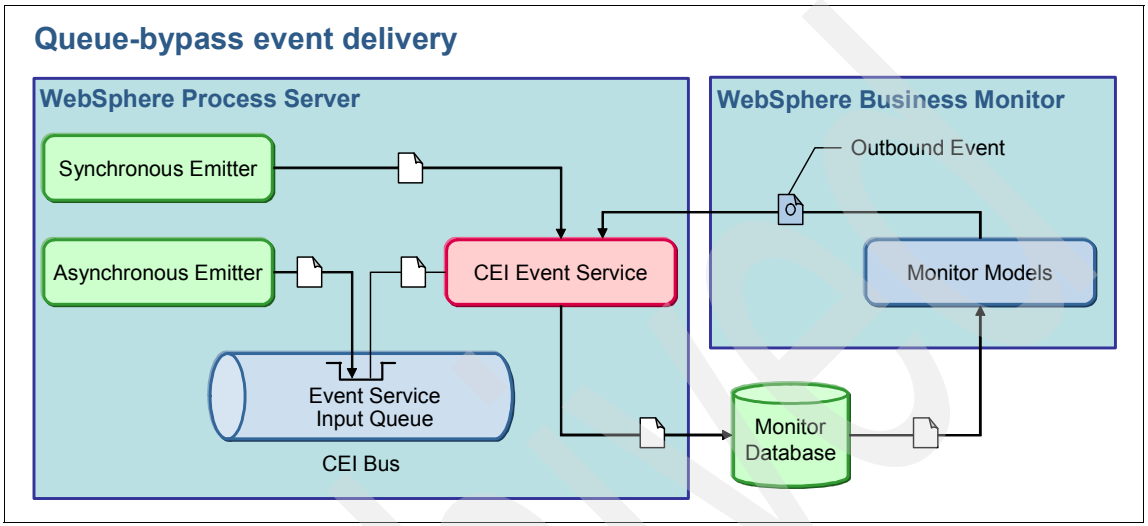


Figure 9-2 Queue-bypass event delivery

Archived

An alternative option is to use a cross-cell bus link between the monitor bus in WebSphere Process Server cell and the monitor bus in WebSphere Business Monitor (Figure 9-3).

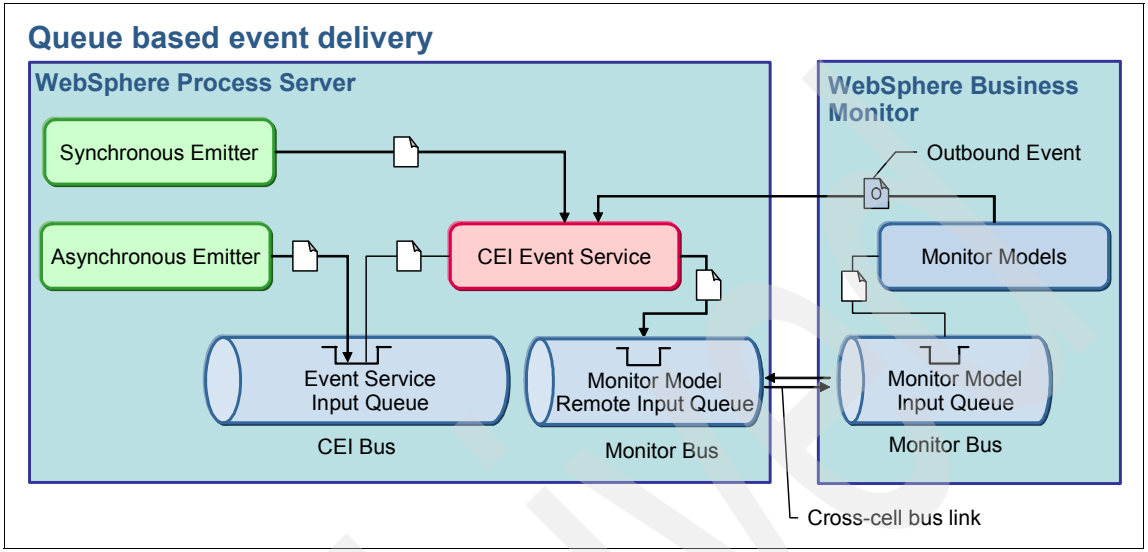


Figure 9-3 Queue based event delivery

This option has not been chosen for three reasons:

- The configuration of a cross-cell bus link delivery mode is more complex to configure and needs more maintenance effort.
- The queue-bypass delivery mode offers a higher event throughput.
- The queue-bypass delivery mode is more robust because fewer components are involved. Only the WebSphere Business Monitor database is used.

Guidelines to choose this topology

This section provides guidelines about when to choose the Remote Messaging, Support and Web topology. Choose this topology if:

- You already have an existing WebSphere Process Server installation in production and do not want to add WebSphere Business Monitor components to it.
- You plan to have different migration time frames for WebSphere Business Monitor and WebSphere Process Server or you must integrate a previous WebSphere Process Server version with WebSphere Business Monitor 7.0.
- Your WebSphere Process Server cell already has a large number of clusters.

9.2 Prerequisites for creating this topology

Before you start creating this topology, you must have completed various prerequisite steps described in Chapter 3, “Preparing your topology” on page 55. These are:

Prepare your operating system

It is crucial to verify that your hardware and operating system conforms to the requirements. Detailed instructions can be found in the InfoCenter. For hardware and software requirements consult the following section in the InfoCenter:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/topic/com.ibm.btools.help.monitor.inst.doc/plan/prep_os_lin.html

Information about your operating-system-specific instructions can be found in the following section of the InfoCenter:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/topic/com.ibm.btools.help.monitor.inst.doc/plan/prep_os.html

Note: Also verify that operating system kernel parameters must be adapted. In our example on SUSE Enterprise Linux 10, we had to increase the number of allowed open files (ulimit).

The WebSphere Business Monitor installation described in this chapter uses fully qualified host names. Table 9-1 shows the host names used in this topology. The following host names must be defined on all servers that are part of this topology. In our cases we added hostnames to the `/etc/hosts` file on each server.

Table 9-1 Host names

Host name	Description
itsodb2.itso.ral.ibm.com	This server hosts the DB2 instance.
itsoldap.itso.ral.ibm.com	This server hosts the LDAP Tivoli Directory Server.
itsodmgr.itso.ral.ibm.com	This server runs the WebSphere Application Server Deployment Manager.
monnode1.itso.ral.ibm.com	First WebSphere Business Monitor node.
monnode1.itso.ral.ibm.com	Second WebSphere Business Monitor node.
monweb.itso.ral.ibm.com	IBM HTTP server for browser access to Business Space

Archived

Install database server

The topology described in this chapter uses DB2 Enterprise Edition 9.5 FixPack 5. Refer to “Software versions used in this book” on page 56 for more information.

Install LDAP server

Tivoli Directory Server 6.2 is used as an Lightweight Directory Access Protocol (LDAP) server in this topology as per “Software versions used in this book” on page 56.

Define users in LDAP user registry

This topology uses a federated user repository with users defined on the LDAP server. Users used in this topology and instructions for how to setup WebSphere Application Server security are described in 3.4.2, “Configuring LDAP as the user account registry” on page 99.

Install WebSphere Business Monitor

Two possibilities exist for installing the WebSphere Business Monitor binaries:

- The graphical installation method using the IBM Installation manager
- The silent installation using IBM Installation Manager response files

Both methods are described in detail in 3.2, “Software installation” on page 57.

The response file `WBM_Base.xml` used to install the WebSphere Business Monitor binaries can be found in Appendix A, “Additional material” on page 421.

Created databases

WebSphere Business Monitor requires four schemas:

- One for the WebSphere Business Monitor tables
- One for the Business Space tables
- One for the CEI Messaging Engine tables
- One for the WebSphere Business Monitor Messaging Engine tables

Archived

It is preferable to spread the schemas across separate databases to ease the tuning of the databases independently. Here are some considerations:

- The WebSphere Business Monitor requires special database grants because tables and schemas are created during monitor model deployment.
- The WebSphere Business Monitor database should be optimized for large data handling.
- The messaging engine database is optimized for a high transaction volume and low data volumes.

The topology described in this chapter uses three databases:

- The Monitor database (MONDB)
- The Business Space database (BSPCDB)
- The Messaging engine database (MEDB)

The SQL generation has been performed using the DDT tool. The generated database design file `wbm.RMRSW.dbDesign` can be found Appendix A, "Additional material" on page 421. A detailed description on how to use the tool can be found in 3.3, "Database and schema creation" on page 82.

Create and configure the deployment manager

Create a deployment manager profile, set up the user registry integration, add IBM HTTP Server to the cell, and modify certain configuration settings, as described in 3.4, "Creating a deployment manager profile" on page 91.

Table 9-2 shows the key properties and values used for the WebSphere Business Monitor topology setup.

Table 9-2 Topology key properties

Topology component name	Value
WebSphere Business Monitor cell name	moncell
monnode1 WebSphere Business Monitor node name 1	monnode1
WebSphere Business Monitor node name 1	monnode2
WebSphere Business Monitor deployment manager node name	mondmgrnode
Integrated Solutions Console	39443
Dmgr bootstrap port	39445
Dmgr SOAP Connector admin port	39446

The response file `wbm_dmgr_create.resp` used to set up the deployment manager can be found in Appendix A, “Additional material” on page 421.

Create and federate custom nodes

Create and federate two custom nodes (monnode1 and monnode2), as per 3.5, “Creating custom profiles” on page 107.

The response file `wbm_node_create.resp` used to set up the custom nodes can be found in Appendix A, “Additional material” on page 421.

Install IBM HTTP Server

In this topology an IBM HTTP server is used for load balancing and failover of the WebSphere Business Monitor WebApp cluster. Refer to 3.6.1, “Installing IBM HTTP Server” on page 111, for instructions about how to set up and configure the HTTP server.

Create WebSphere Proxy Server

You do not need to install WebSphere Proxy Server for this topology because the WebSphere Business Monitor Representational State Transfer (REST) gateway and the Business Space are co-located on the same cluster. Refer to 2.6.7, “REST endpoint configuration” on page 47, for information about how to configure the WebSphere Business Monitor REST endpoints co-located on the same cluster.

9.3 Configuring the topology using deployment environments in the Integrated Solutions Console

This section explains the use of the Deployment Environment wizard that is part of the Integrated Solutions Console to set up the Remote Messaging, Support and Web topology, which creates WebSphere Business Monitor components.

Archived

You need a local copy of the `dbDesign` file because the browser does not look at remote file systems.

1. Open the Integrated Solutions Console and navigate to **Servers** → **Deployment Environments**, as in Figure 9-4.

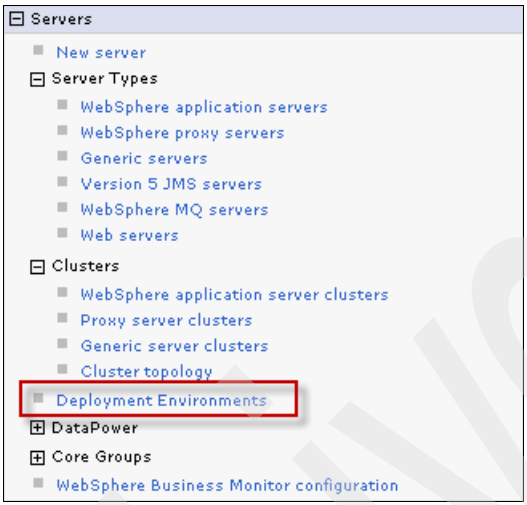


Figure 9-4 Launch Deployment Environments in Integrated Solutions Console

2. A table view of the predefined deployment environment (DE) is displayed. Click **New** to start the Deployment Environment creation wizard, as in Figure 9-5.

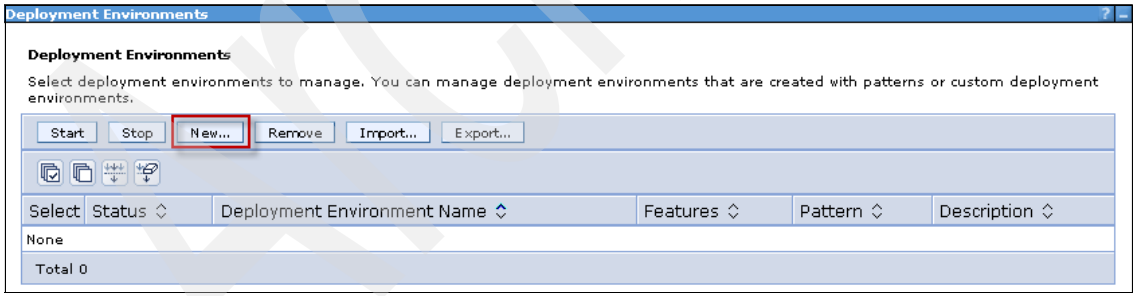


Figure 9-5 Create new deployment environment

Archived

3. On the next page click **Create a deployment environment based on pattern**, enter `WBM.RMRSW` for the deployment environment name, and click **Detailed: Show all steps**, as in Figure 9-6, then click **Next**.

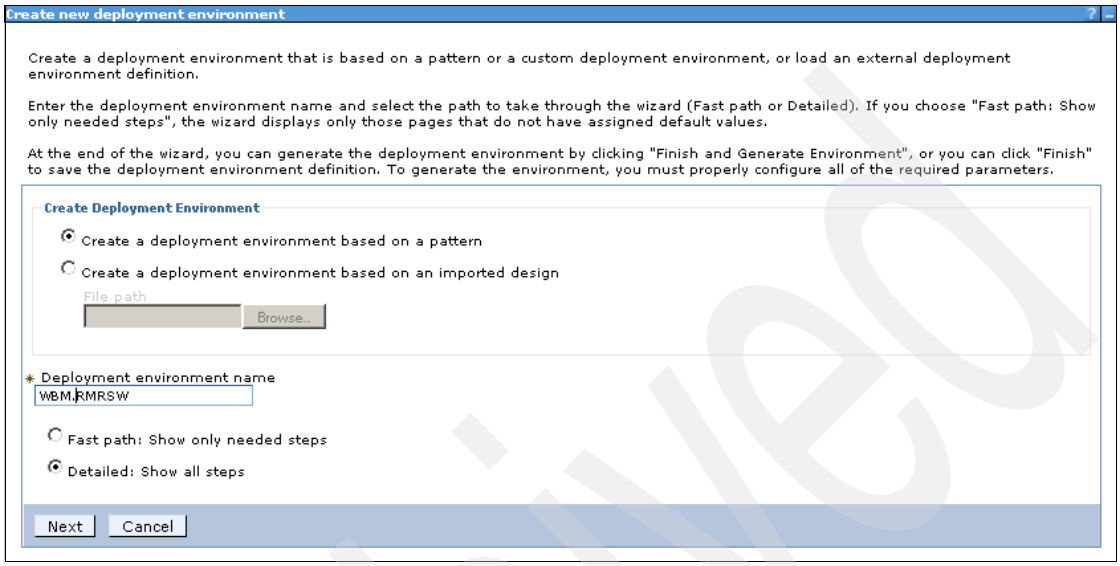


Figure 9-6 Create new Deployment Environment

4. On the next page verify that **WBM** is selected and click **Next**, as in Figure 9-7. WBM is the only feature available, as we only installed the WebSphere Business Monitor binaries.

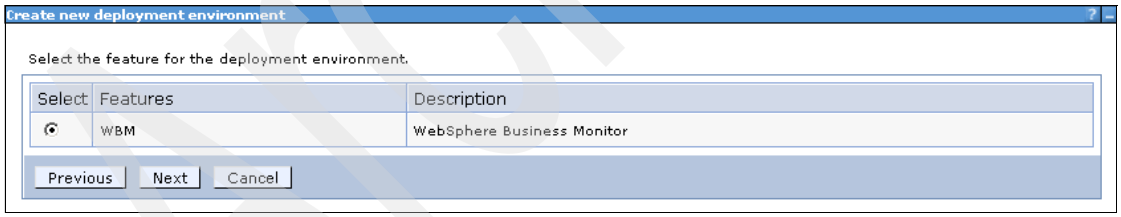


Figure 9-7 DE feature selection

Archive

5. On the next page choose the topology. WebSphere Business Monitor only supports Single Cluster and Remote Messaging, Support and Web topology. Click **Remote Messaging, Remote Support, and Web**, then click **Next**, as in Figure 9-8.

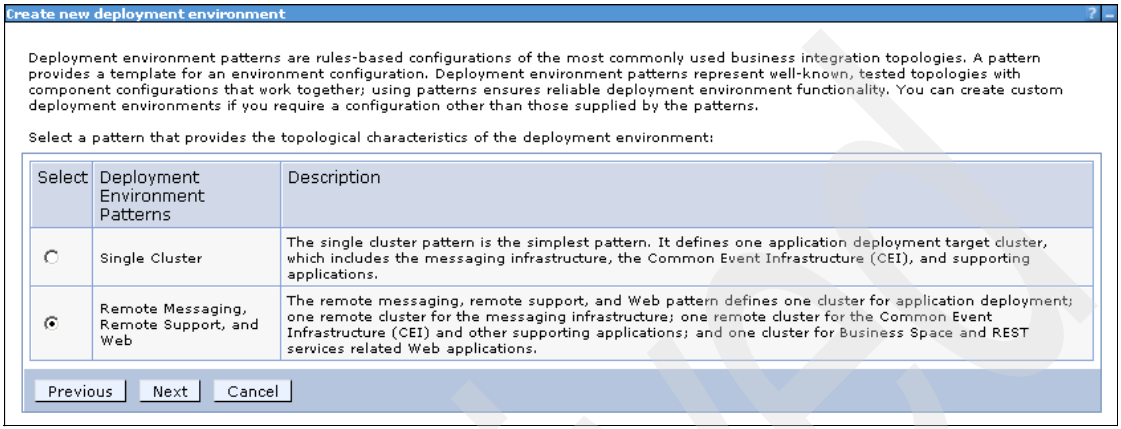


Figure 9-8 DE topology selection

6. The topology selection wizard opens. In the first step of the wizard choose the nodes to be part of the deployment environment. Click the check boxes next to **monnode1** and **monnode2** and click **Next**, as in Figure 9-9.

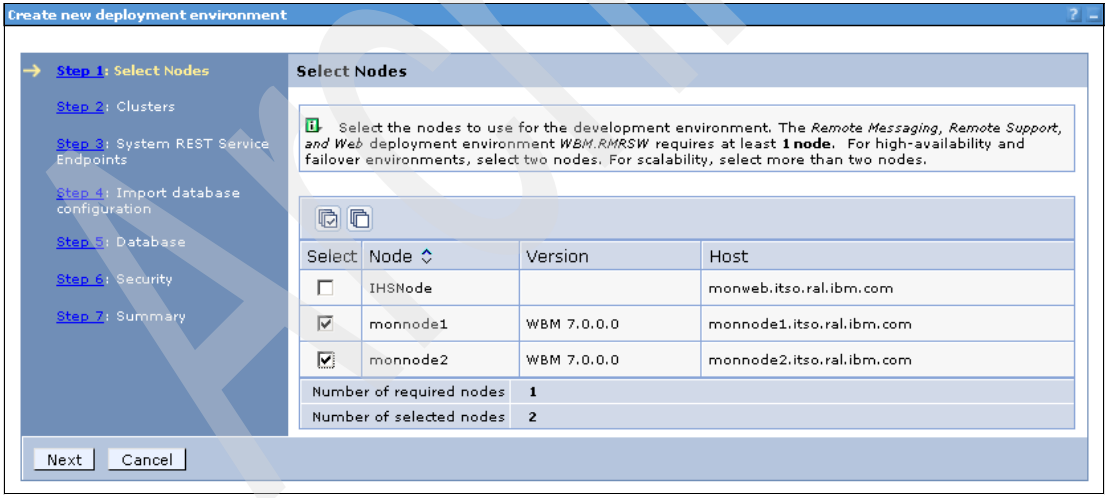


Figure 9-9 Select nodes for DE

Archive

7. Decide on the number of clusters per node. Leave the defaults (one cluster everywhere) to create a four-cluster topology on the two nodes, as in Figure 9-10. Click **Next** when finished.

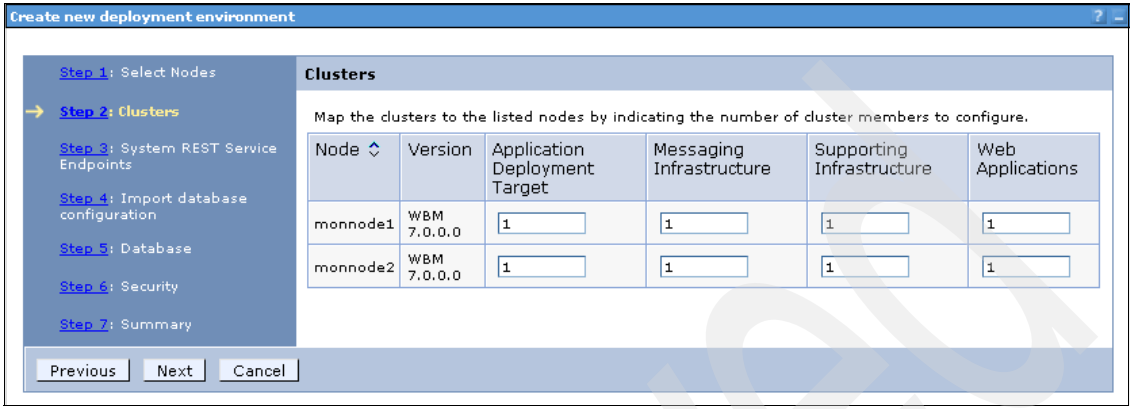


Figure 9-10 Decide on number of cluster member

8. Specify the REST endpoint. The topology in this chapter uses an HTTP server in front of the application server to load balance the REST calls issued by Business Space widgets and does not use the Business Space AJAX proxy. For this configuration specify the HTTP server address in this panel. Check that the Protocol selection box is `https://`, enter `rmonweb.itso.ral.ibm.com` for the host name and enter `80` for the port, as in Figure 9-11, then click **Next**.

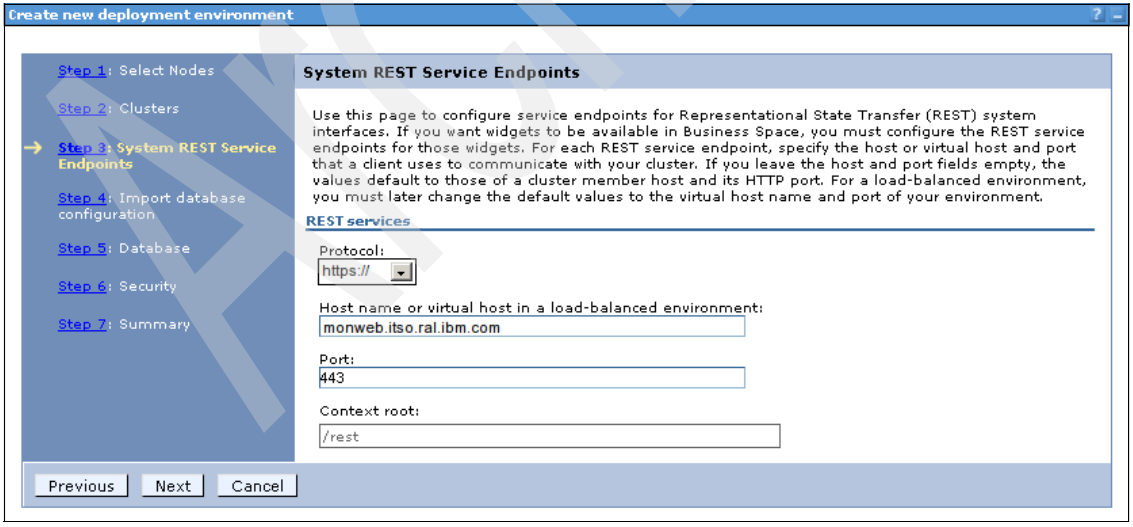


Figure 9-11 Rest endpoint configuration

9. In the next step provide the database design file generated in 3.3, “Database and schema creation” on page 82. Click **Browse** and browse to your DDT file called `wbm.RMRSW.dbDesign`, then click **Next**, as in Figure 9-12.

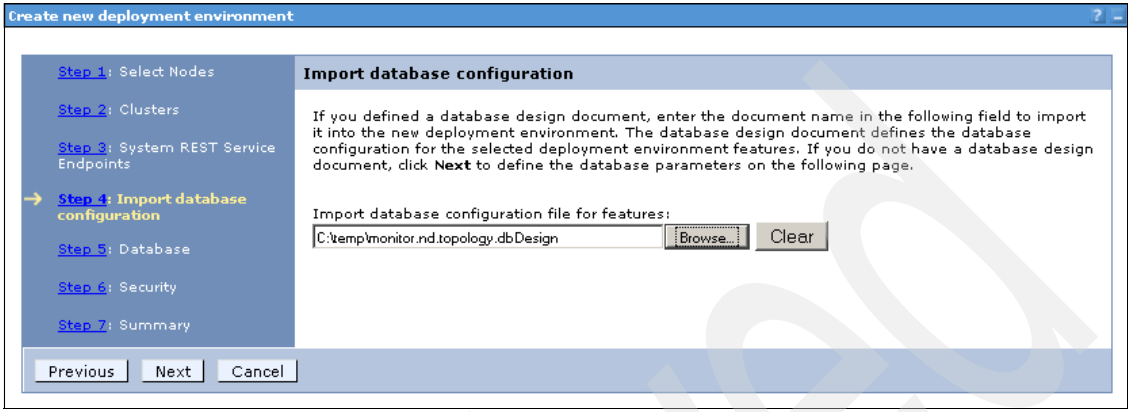


Figure 9-12 Provide database design tool file to DE wizard

10. As we provided the DDT file, the database parameters are already filled in and do not need to be changed. Review the parameter for each database, specifically check if the Create tables check boxes are unchecked because we have already created these. Then click **Next**, as in Figure 9-13.

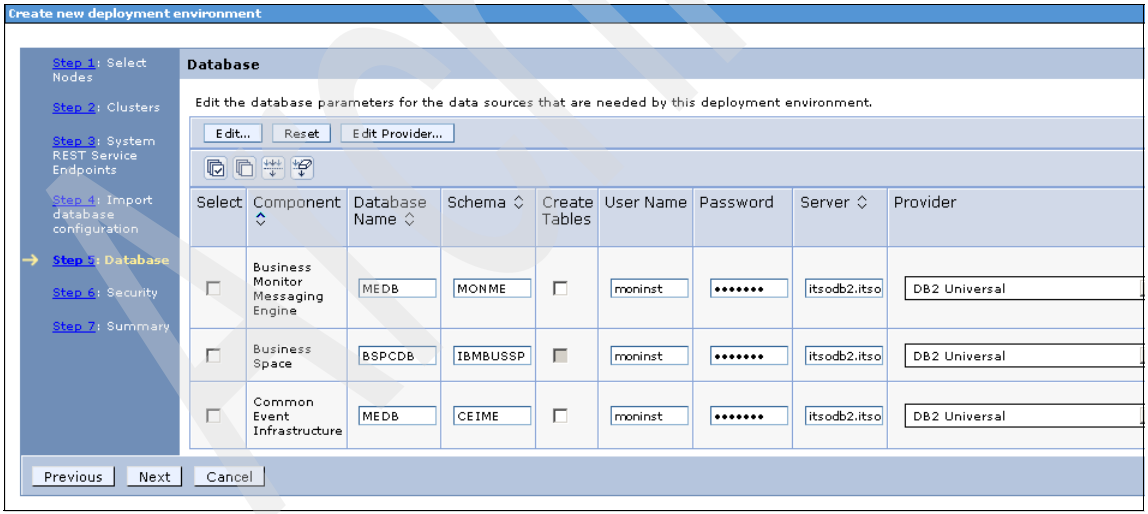


Figure 9-13 Database properties in DE wizard

Note: The WebSphere Business Monitor database MONDB is not part of the list. The related datasources have already been created during Deployment Manager profile creation. Remember that the database connection parameters were provided in the response file.

11. In the next step configure the security alias. The common event infrastructure alias is used to connect to the CEI bus and the WBM_Alphablox alias is used by Alphablox to connect to the WebSphere Business Monitor deployment manager. Leave the defaults as in Figure 9-14 and click **Next**.

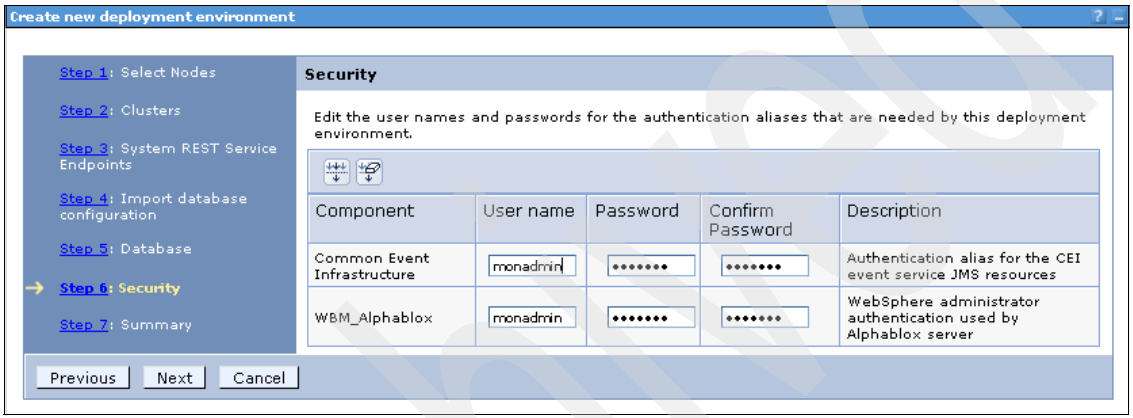


Figure 9-14 Configure security aliases

12. Review the options on the summary page and click **Finish**. Do *not* click Finish and Generate Environment.

13. Save changes and synchronize with nodes.

    The DE has been created, as shown in Figure 9-15.

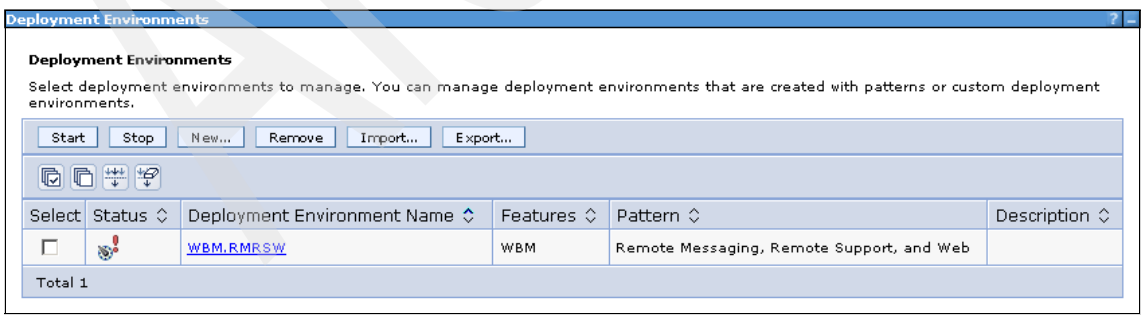


Figure 9-15 New deployment environment

14. Optionally, we could export the deployment environment into a XML file at this step by clicking **Export**. Exported deployment environment files can be used to set up similar environments or to automate the deployment environment creation, as explained in 9.4, “Configuring topology manually using command line” on page 313.
15. Generate the deployment environment. Click the **WBM.RMRSW** link in the DE table.
16. On the bottom of the page click **Generate Environment**, as in Figure 9-16.

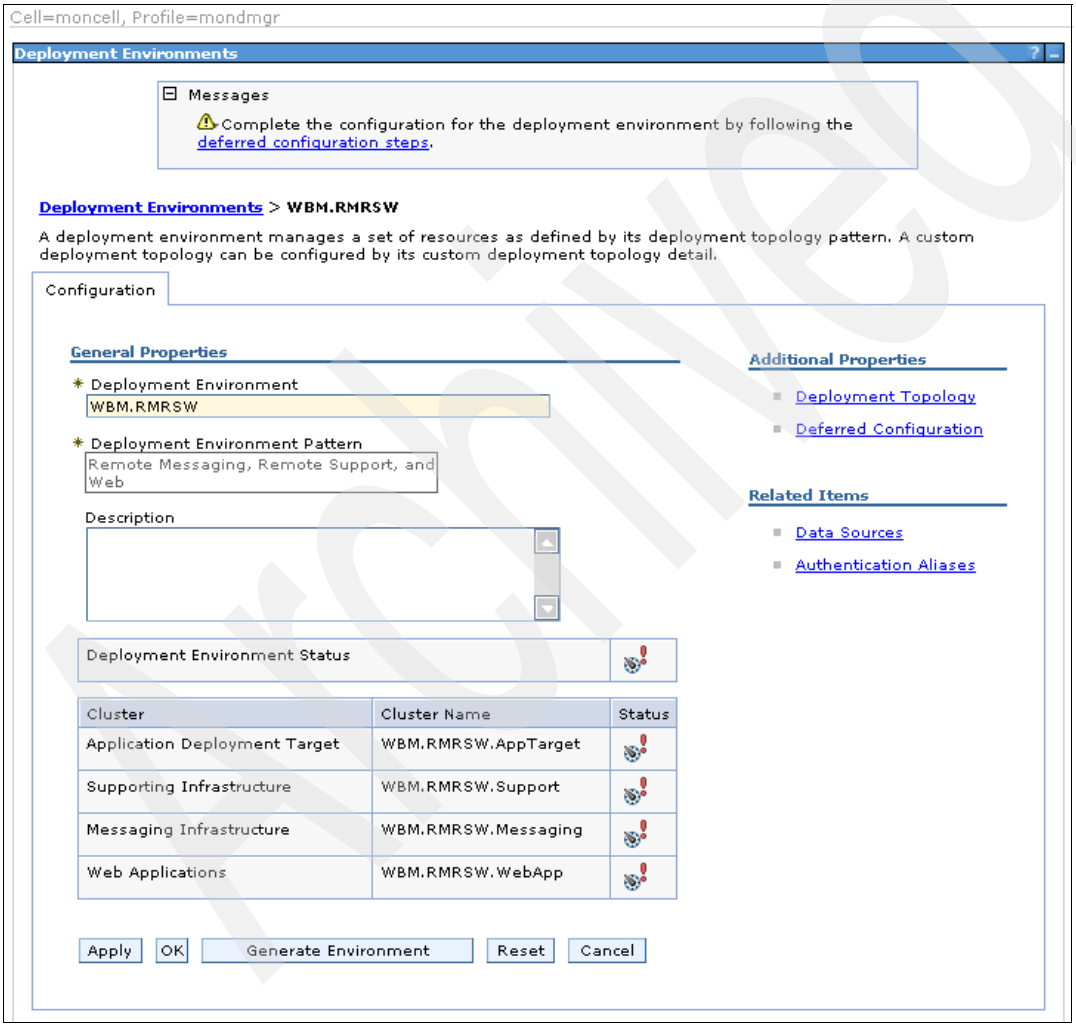


Figure 9-16 Generate deployment environment

17. A progress report is shown on the page. Wait until the `The configuration has ended` message appears on the page, then click **Save Changes**, as in Figure 9-17.

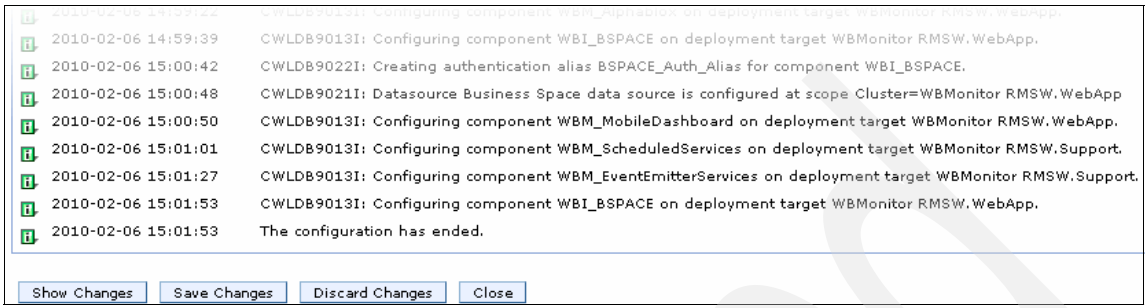


Figure 9-17 Configuration has ended

18. The deployment environment has been successfully created. Click **Save changes** and synchronize changes with the nodes.

> **Warning:** Do not restart your deployment manager yet. You must first ensure that the generation of the deployment environment is complete. Carry out the steps in 9.5.1, “Restarting the deployment manager” on page 314, to ensure that this is done successfully.

19. The deployment environment has been created successfully. Continue with the steps described in 9.5, “Post-creation configuration and verification” on page 313, to finalize the environment and to verify that all component are installed properly.

Archive

9.4 Configuring topology manually using command line

To configure a topology using the command line, you need a topology definition file. The typical way to create this is to define a topology using the Integrated Solutions Console and then export it. (This is typically *not* done on the WebSphere cell on which you want to generate the topology.) To do this:

1. Define a topology as you did in 9.3, "Configuring the topology using deployment environments in the Integrated Solutions Console" on page 304, for example. However, only save the environment definition by selecting **Finish** at the end. Do not actually generate it.
2. Export the topology by opening the Integrated Solutions Console, navigating to **Servers** → **Deployment Environments**, selecting the relevant topology definition, and clicking **Export**.

Importing and generating a deployment environment topology

Once you have a file representing a deployment environment topology, you can import it and generate the environment using one simple wsadmin script.

You can find an example script called `createDE.py` in Appendix A, "Additional material" on page 421. You could execute this script using a command such as:

```
/opt/ibm/WebSphere/ProcServer/profiles/DMgrName/bin/wsadmin.sh -lang
jython -conntype SOAP -host yourdmgr -port yourport -f createDE.py
topologyName /path/to/topology.definition.xml -user yourusername
```

The script defines the topology within the WebSphere administrative environment, then creates it.

Warning: Do not restart your deployment manager yet. You must ensure that the generation of the deployment environment is complete first. Carry out the steps in 9.5.1, "Restarting the deployment manager" on page 314, to ensure that this is done successfully.

9.5 Post-creation configuration and verification

This section discusses configuration and verification steps.

9.5.1 Restarting the deployment manager

You must restart the deployment manager. However, it is *very important* to ensure that the deployment environment generation has finished successfully before doing this or you may encounter problems. This is an asynchronous process, so a successful response from the Integrated Solutions Console or wsadmin stating that the deployment environment has been created is not sufficient.

To do this, monitor the logs on each of your node agents (*not* the deployment manager). As the deployment environment is created, you should see messages of this form:

```
[2/11/10 10:31:57:639 EST] 0000003e FileRepositor A   ADMR0015I: User
itsoldap/bpcadmin created document
BusinessSpace/help/eclipse/plugins/org.eclipse.help.nl1/plugin_ar.
properties
```

There will many of these messages in both node agents logs as the standard applications are deployed to both nodes in your deployment environment. Wait for these messages to stop appearing and for the logs to quiesce, as this tells you that the deployment environment generation is complete.

Archived

9.5.2 Verify the WebSphere Business Monitor configuration

Use the WebSphere Business Monitor configuration page to check whether all components are successfully deployed:

1. In the Integrated Solutions Console navigate to **Servers** → **WebSphere Business Monitor configuration**.
2. On the WebSphere Business Monitor configuration page verify that all components have a green icon, as shown in Figure 9-18.

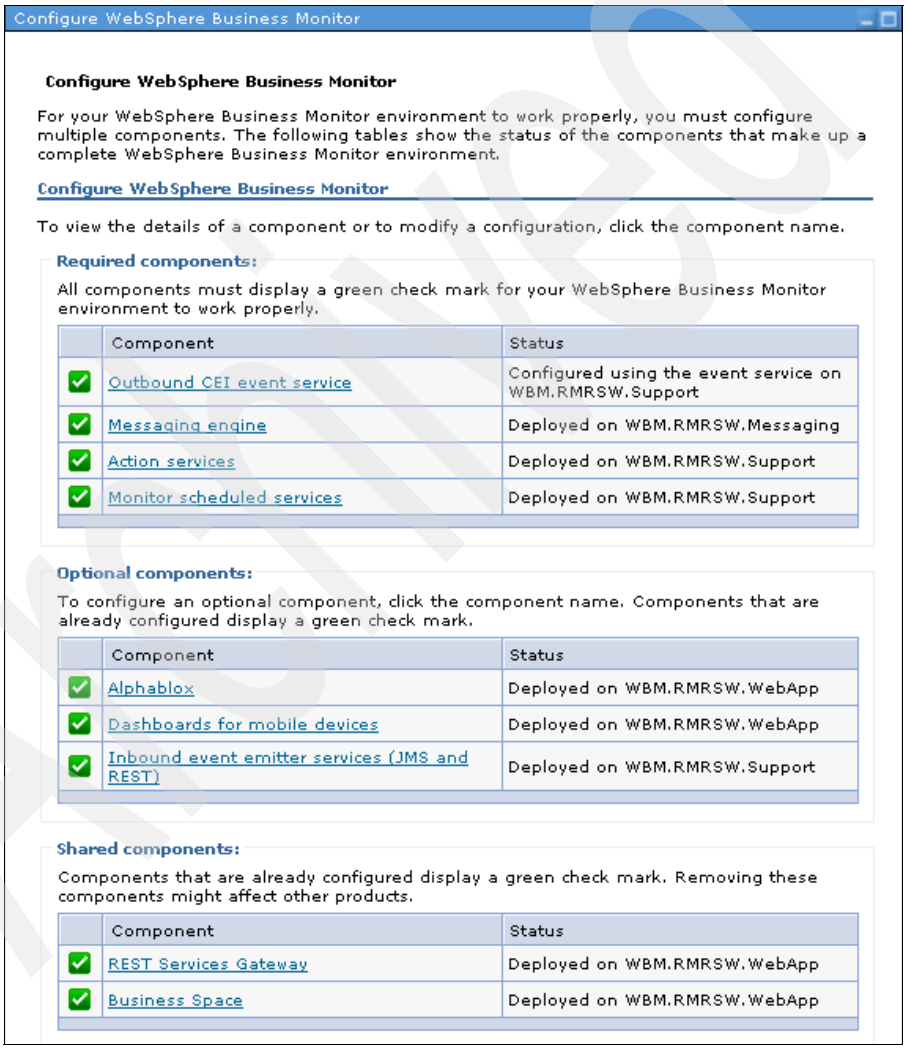


Figure 9-18 WebSphere Business Monitor configuration page

Archived

9.5.3 Checking database connectivity

Before you start the environment check the database connectivity to ensure that the deployment environment has been generated correctly. Follow these steps to verify the database connectivity:

1. Log in to the administrative console and navigate to **Resources** → **JDBC** → **JDBC Providers** → **Data Sources**.
2. Make sure that **All scopes** is selected.
3. Select all of the data sources and click **Test connection** to make sure that they have connectivity. You should see a message that the connection was successful for each.

Command-line equivalent

You can use a script such as that shown in Example 9-1 to check database connectivity using wsadmin instead. You must change the cell name to the cell name that you are using.

Example 9-1 Checking database connectivity using wsadmin

```
cellName = "moncell"

dataSources = AdminConfig.list('DataSource', AdminConfig.getid(
'/Cell:' + cellName + '/')).split("\n")

for dataSource in dataSources:
   if not dataSource.startswith('DefaultEJB'):
      print dataSource
      AdminControl.testConnection(dataSource)
```

9.5.4 Map Web modules on the HTTP server

All applications deployed on the WebApp cluster must be mapped to the HTTP Server. Perform the steps below for each application in Table 9-3.

Table 9-3 Map Web modules to HTTP server

Enterprise application	Web module
AlphabloxPlatform	IBM Alphablox
AlphabloxPlatform	Dashboard.ABX
BSpaceEAR_WBM.RMRSW.WebApp	BSpace

Archived

Enterprise application	Web module
BSpaceWebformsEnabler_WBM.RMRSW.WebApp	BSpaceWebformsProxy
BusinessSpaceHelpEAR_WBM.RMRSW.WebApp	BusinessSpaceHelp.war
IBM_WBM_EMITTER_SERVICES	EventEmitterREST
IBM_WBM_MOBILE_DASHBOARD	MobileDashboard
REST Services Gateway	REST Services Gateway
WBMDashboardWeb_WBM.RMRSW.WebApp	WBMDashboard
mm.was_WBM.RMRSW.WebApp	mm.runtime.was
mm.was_WBM.RMRSW.WebApp	mm.widget.facade.was

The steps are:

1. In the Integrated Solutions Console navigate to **Applications** → **Application Types** → **WebSphere enterprise applications**.

Archived

2. Click each enterprise application listed in Table 9-3 on page 316 and then click **Modules** → **Manage Modules** to list the modules contained in this enterprise application, as shown in Figure 9-19.

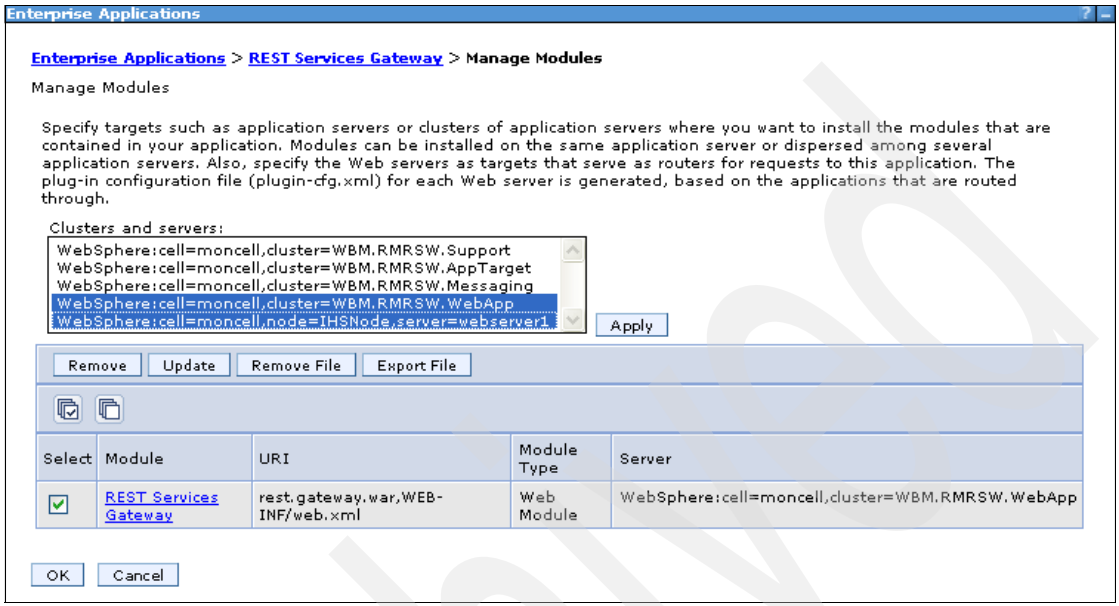


Figure 9-19 Map Web modules to http server

3. Click the **Select** check box left of the Web module that you want to map. Select the **WBM.RMRSW.WebApp** cluster and the **webserver1** HTTP server in the list and click **Apply**, then **OK.**
4. Repeat steps 2 and 3 for each enterprise archive in Table 9-3 on page 316.
5. Once finished, save and synchronize the changes.

Archive

9.5.5 Update virtual host settings

The WebSphere Application Server virtual host settings must be adapted. Web browser access is only allowed via the http server or to the WebApp cluster. The WebApp cluster listens on port 9084 for HTTP and 9447 for HTTPS in our case. Perform the steps below to adapt the virtual host settings:

1. In the Integrated Solutions Console navigate to **Environment** → **Virtual hosts** → **default_host** → **Host aliases**, as shown in Figure 9-20.

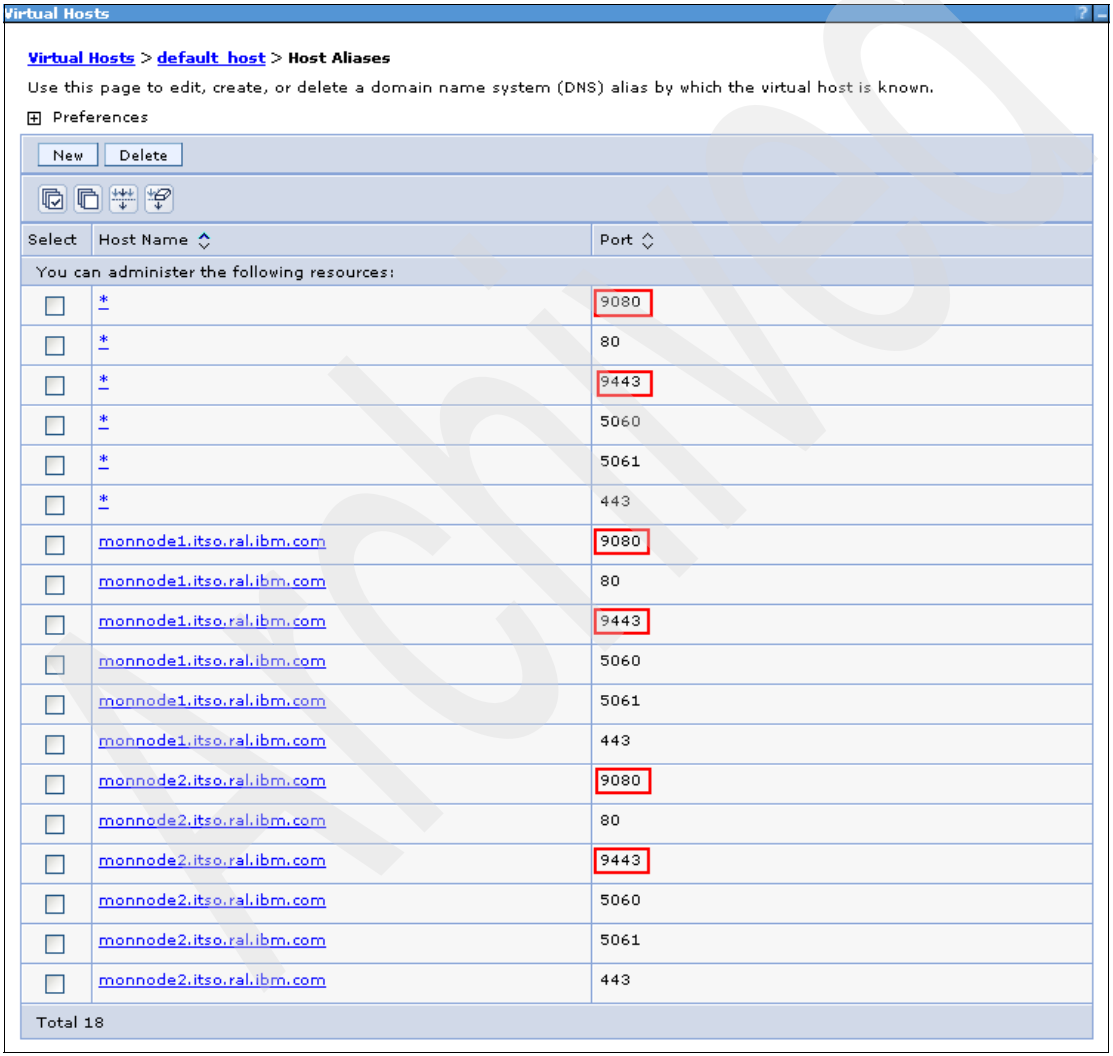


Figure 9-20 Host aliases

2. Change all entries with port 9080 and 9084 and ports 9442 to 9447 by clicking each host name in the table.
3. Save and synchronize the changes.
4. Navigate to **Servers** → **Server Types** → **Web Servers** and regenerate and republish the HTTP server plug-in by selecting the check box to the left of the HTTPS server **webeserver1** and clicking **Generate Plug-in**, then **Propagate Plug-in**.
5. Restart the HTTP server for the changes to take effect by clicking **Stop**, then **Start**.

9.5.6 Configure monitor data security

Assign at least one user to the root group in the WebSphere Business Monitor data security panel:

1. Using the WebSphere Business Monitor Integrated Solutions Console navigate to **Security** → **Monitor Data Security**.
2. Click the default resource group called **root**.
3. On the next page assign the KPI-Administrator role to the administrator monadmin by clicking radio box next to **KPI-Administrator**, then click **Users**.

Archived

4. On the Select users for KPI-Administrator page click **Search** to list all users defined in the LDAP server, click the user **cn=monadmin,ou=IBM,o=ITSO**, and click the button labeled **>** to move the user to the list of selected users. Click **OK** and on the next page again click **OK** to save settings, as shown in Figure 9-21.

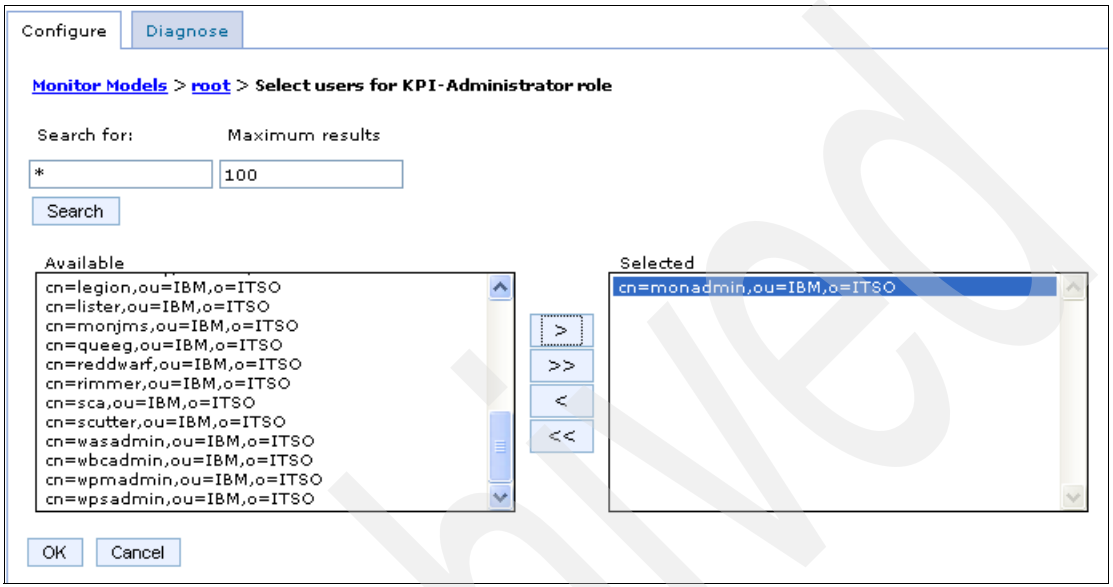


Figure 9-21 Select users for KPI-Administrator role

Archived

9.5.7 Starting the clusters and deployment environment

Now start the deployment environment from the Integrated Solutions Console. Navigate to **Servers** → **Deployment Environments**, click the check box next to WBM.RMRSW, and click **Start**, as shown in Figure 9-22.

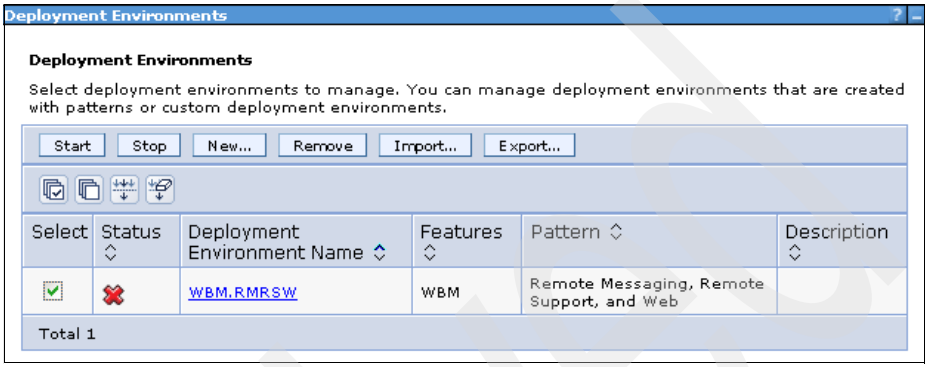


Figure 9-22 Start deployment environment

To verify the successful creation of the deployment environment each of the clusters must be started. Starting the clusters starts all the application servers and enterprise applications deployed on the servers defined in the cluster.

When the clusters are started you must check the `SystemOut.log` file for each of the servers to verify that there are no errors. The `SystemOut.log` file is in the `logs` folder of each profile. For example, in our environment the `SystemOut.log` file for the messaging cluster on monnode1 is in the `/opt/ibm/WebSphere/MonServer/profiles/monnode1/logs/WBM.RMRSW.Messaging.monnode1.0/` folder. Example 9-2 shows an example of the message written to the log during startup of the messaging cluster showing that all the messaging engine artifacts were started successfully.

Example 9-2 Messaging cluster startup log verification

```
CWSID0016I: Messaging engine
WBM.RMRSW.Messaging.000-MONITOR.moncell.Bus is in state Started.
CWSID0016I: Messaging engine WBM.RMRSW.Messaging.000-CEI.moncell.BUS is
in state Started.
WSVR0001I: Server WBM.RMRSW.Messaging.monnode2.0 open for e-business
```

Example 9-3 shows an example of the message written to the log during startup of the support cluster showing all the applications artifacts that were started successfully.

Example 9-3 Support cluster startup log verification

```
WSVR0221I: Application started: IBM_WBM_DATA_SERVICES
WSVR0221I: Application started: IBM_WBM_EMITTER_SERVICES
WSVR0221I: Application started: IBM_WBM_ACTIONSERVICES
WSVR0221I: Application started: EventService
WSVR0001I: Server WBM.RMRSW.Support.monnode2.0 open for e-business
```

Example 9-4 shows an example of the messages put out during startup of the WebApp cluster showing all the applications artifacts that were started successfully.

Example 9-4 WebApp startup log verification

```
WSVR0221I: Application started: WBMDashboardWeb_WBM.RMRSW.WebApp
WSVR0221I: Application started: ApplicationStudio
WSVR0221I: Application started: REST Services Gateway
WSVR0221I: Application started: AlphabloxPlatform
WSVR0221I: Application started: mm.was_WBM.RMRSW.WebApp
WSVR0221I: Application started: BSpaceWebformsEnabler_WBM.RMRSW.WebApp
WSVR0221I: Application started: BSpaceEAR_WBM.RMRSW.WebApp
WSVR0221I: Application started: BusinessSpaceHelpEAR_WBM.RMRSW.WebApp
WSVR0221I: Application started: IBM_WBM_MOBILE_DASHBOARD
WSVR0001I: Server WBM.RMRSW.WebApp.monnode2.0 open for e-business
```

Archived

In the Integrated Solutions Console navigate to **Servers** → **Clusters** → **WebSphere application server clusters** to confirm as the clusters are started, as shown in Figure 9-23.

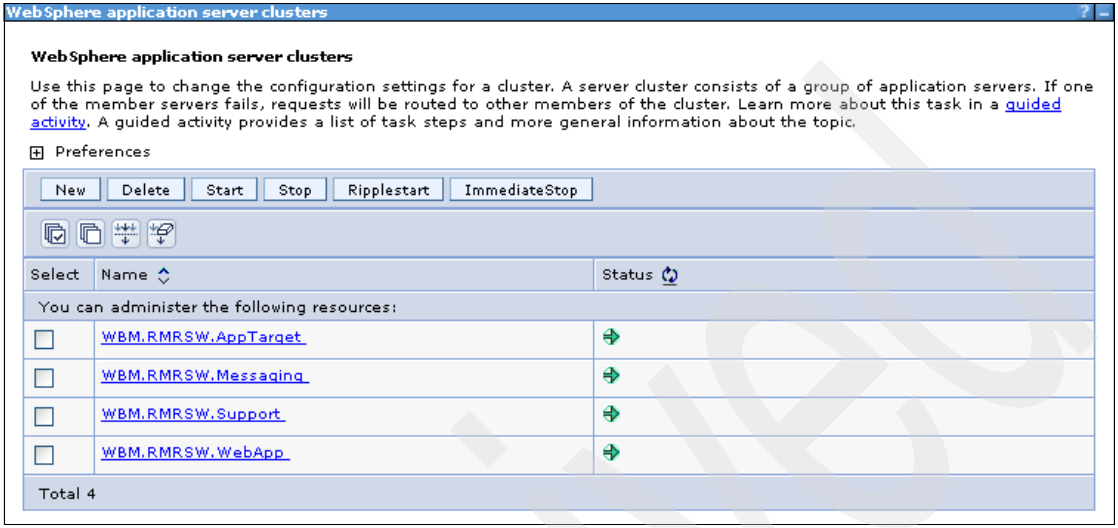


Figure 9-23 Verify cluster startup

The successful start of the clusters automatically starts the deployment environment. In the Integrated Solutions Console verify that the deployment environment is successfully started by clicking **Servers** → **Deployment Environments**, as shown in Figure 9-24.

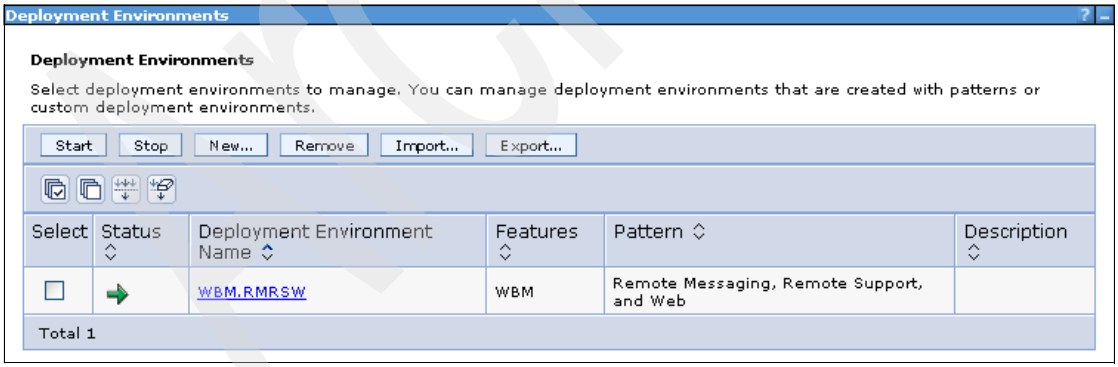


Figure 9-24 Verify startup of deployment environment

9.5.8 Setup cross-cell queue bypass

In order to consume events from a remote WebSphere Process Server cell using queue bypass event delivery mode, we must set up a data source on the remote cell that points to the WebSphere Business Monitor database. For this setup we re-use the WebSphere Process Server Remote Messaging and Remote Support topology described in Chapter 5, "WebSphere Process Server: Remote Messaging and Remote Support" on page 149. Additionally, as WebSphere security is enabled on both sides we also must exchange the LTPS token and SSL certificate. The routing datasource creation can be achieved by executing the following steps:

1. Navigate to the `/opt/ibm/WebSphere/ProcServer/util/WBM` directory on the WebSphere Process Server Deployment Manager node and make a copy of the configureQueueBypass.py script to myConfigureQueueBypass.py.
2. Edit the file and change the parameters as shown in Table 9-4. A sample file called `myConfigureQueueBypass.py` can be found in Appendix A, "Additional material" on page 421.

Table 9-4 Parameter used by configureQueueBypass.py script

Parameter	Value
cellName	moncell
dbType	db2
jdbcProvider	DB2 Universal JDBC Driver Provider (XA)
driverType	4
dbName	MONDB
dbServerName	itsodb2
dbServerPort	500015
securityParam	-user moninst -password itso4you

3. Navigate to the `/opt/ibm/WebSphere/ProcServer/bin` directory and execute the following script:

   ```
   ./wsadmin.sh -lang jython -f ../util/WBM/myConfigureQueueBypass.py
   ```

   Provide the WebSphere Application Server Administrator user name and password when prompted.
4. Log in to the Integrated Solutions Console, navigate to **Resources** → **JDBC** → **Data sources**, and verify that the data source Monitor_monCell_Routing_Database has been created. Click the **Select**

Archive

check box to the left of the data source and then click **Test connection** to verify that the connection can be established with the WebSphere Business Monitor database.

Now we must exchange the LTPA token and SSL certificates. Perform the following steps:

1. First we must export the LTPA token from the WebSphere Process Server deployment manager:
   a. Log in to WebSphere Process Server Integrated Solutions Console using the LDAP user `wpsadmin` and password `itso4you`.
   b. Navigate to **Security** → **Global Security** → **Authentication** in the WebSphere Process Server Integrated Solutions Console and click **LTPA,** as in Figure 9-26 on page 327.

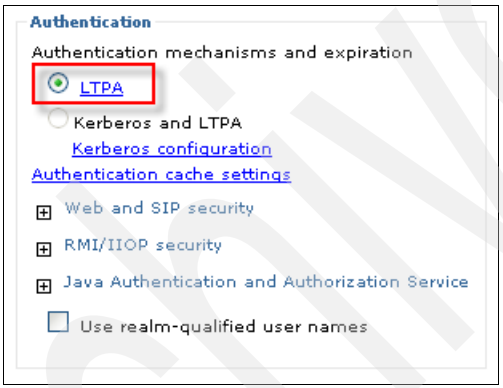


Figure 9-25 Export LTPA keys from WebSphere Business Monitor

   c. In the Cross-cell single sign-on section enter `itso4you` in the Password and Confirm password fields and enter `/tmp/ltpa.key` in the Fully qualified key file name field, then click **Export keys**.
   d. Copy the export key file to WebSphere Business Monitor server's `/tmp` directory.

Archived

2. Now we must import the LTPA token and WebSphere Process Server SSL signer certificate of the deployment manager SOAP connector into the WebSphere Business Monitor server:

   a. Log in to the WebSphere Business Monitor Integrated Solutions Console using the LDAP user `monadmin` and password `itso4you`.

   b. Navigate to **Security** → **Global Security** → **Authentication** in the WebSphere Business Compass Integrated Solutions Console and click **LTPA**, as in Figure 9-26.

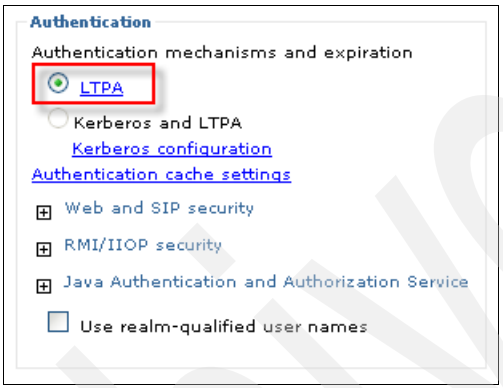


Figure 9-26 Export LTPA keys from WebSphere Business Monitor

   c. In the Cross-cell single sign-on section enter `itso4you` in the Password and Confirm password fields and enter `/tmp/ltpa.key` in the Fully qualified key file name field (or the location where you copied the LTPA keys from WebSphere Process Server), then click **Import keys**.

   d. Navigate to **SSL certificate and key management** → **Key stores and certificates** → **NodeDefaultTrustStore** → **Signer certificates**.

Archived

e. Click **Retrieve from port** and enter the WebSphere Process Server server connection parameter, as in Figure 9-27.

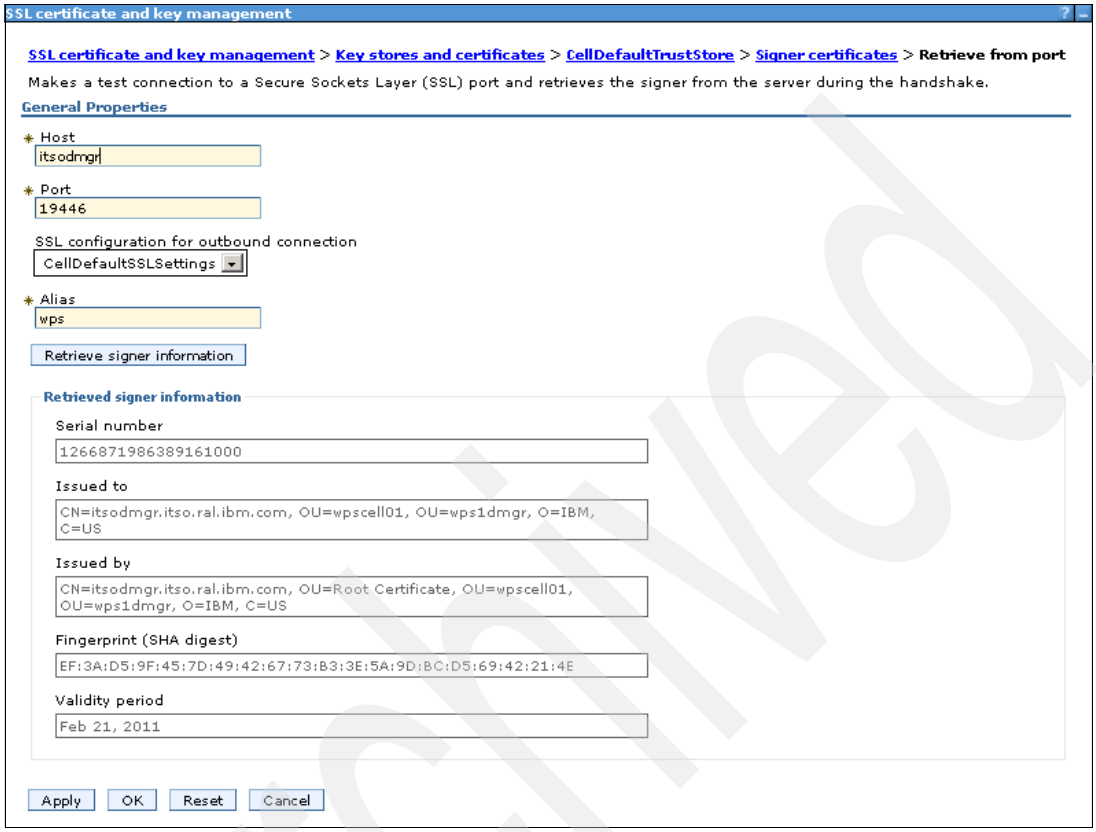


Figure 9-27 Import signer certificate from deployment manager SOAP connector port

f. Click **Retrieve Signer information**. Review the SSL information and click **OK** to import the certificate into the trust store.

g. Save and synchronize changes.

h. Stop and restart all servers, node agents, and deployment managers.

Archived

Note: If you have a different security realm in both cells you also must add the WebSphere Process Server realm name to the trusted authentication realm, or if CSIv2 security settings have been changed, verify the settings below:

1. In the Integrated Solutions Console navigate to **Security** → **Global Security**. Under RMI/IIOP security, click **CSIv2 outbound communications**.
2. Click **Trusted authentication realms** → **outbound**.
3. Select **Trust realms as indicated below**. Click **Add External Realm** and add the realm of the remote cell. Click **Apply**.

Also verify the identity assertion settings in both cells.

In the WebSphere Process Server cell verify the following settings:

1. On the navigation panel, click **Security** → **Global security**.
2. Under RMI/IIOP security, click **CSIv2 inbound communications**.
3. Select the **Use identity assertion** check box and ensure that Trusted identities is set to * (asterisk).

On the WebSphere Business Monitor cell verify the following settings:

1. Under RMI/IIOP security, click **CSIv2 outbound communications**.
2. Select the **Use identity assertion** check box and ensure that Use server trusted identity is selected.

9.5.9 Verify the topology using a sample application

As a last step deploy the HelloWorld sample application provided with the additional material of this book to validate all components. Appendix B, “Verification using the sample application” on page 423, describes the steps to install and test the HelloWorld application.

Archive

Archived

10

WebSphere Business Services Fabric: Remote Messaging and zRemote Support

This chapter provides step-by-step instructions for configuring WebSphere Business Services Fabric V7.0 using a Remote Messaging and Remote Support deployment environment pattern.

© Copyright IBM Corp. 2010. All rights reserved.

10.1 Topology summary

The WebSphere Business Services Fabric Remote Messaging and Remote Support topology described in this chapter is depicted in Figure 10-1. The Remote Messaging and Remote Support deployment pattern depicts the deployment of various WebSphere Process Server components in separate clusters. The clusters and components include one cluster for application deployment, one cluster for messaging, and one cluster for supporting applications. The Remote Messaging and Remote Support topology is the recommended pattern for WebSphere Business Services Fabric production deployment.

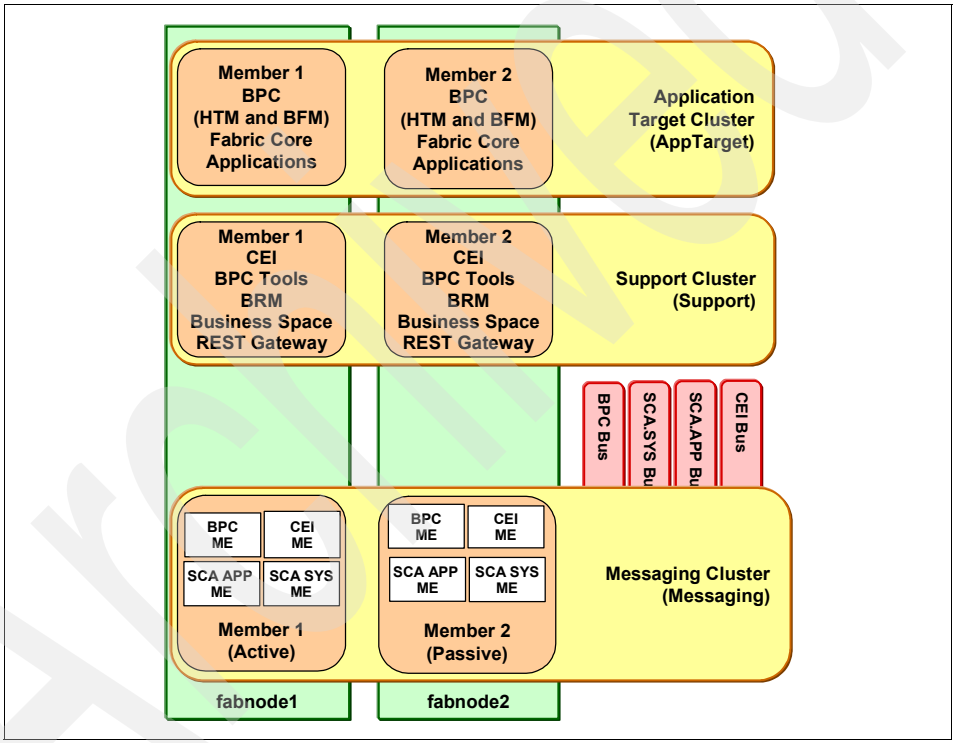


Figure 10-1 WebSphere Business Services Fabric Remote Messaging Remote Support topology

10.2 Prerequisites for creating this topology

The prerequisite software for building the topology for WebSphere Business Services Fabric V7.0 is covered in this section. Before you start creating this

topology, you must complete the prerequisite steps described in Chapter 3, “Preparing your topology” on page 55. These are:

- Install DB2 (or an alternative database) as per “Software versions used in this book” on page 56.
- Install Tivoli Directory Server (or an alternative user registry) as per “Software versions used in this book” on page 56.
- Define the appropriate administrative groups and users in your user registry as per 3.4.2, “Configuring LDAP as the user account registry” on page 99.
- Install the product binaries for WebSphere Business Services Fabric V7.0 as per 3.2.4, “Installing WebSphere Business Services Fabric” on page 70.
- Install IBM HTTP Server (or an alternative HTTP server) as per 3.6.1, “Installing IBM HTTP Server” on page 111.
- Create the required databases in DB2 as per 3.3, “Database and schema creation” on page 82.

  Note: Tables for the event database are not created now, but after the deployment environment has been created, so do not execute the steps in 3.3.7, “The EVENT database” on page 89 yet.

- Create a WebSphere Business Services Fabric Deployment Manager profile, set up the user registry integration, create the nodes, add the IBM HTTP Server to the cell, and modify configuration settings as per 3.4, “Creating a deployment manager profile” on page 91. Note that you do not need to install WebSphere Proxy Server for this topology (as shown in 3.6.3, “Installing and configuring WebSphere proxy server” on page 118).
- Create and federate two custom WebSphere Business Services Fabric nodes as per 3.5, “Creating custom profiles” on page 107.

10.3 Configuring topology using deployment environments in Integrated Solutions Console

The following procedure creates a Remote Messaging and Remote Support topology using the Integrated Solutions Console (ISC). Before you begin, ensure that the prerequisites in 10.2, “Prerequisites for creating this topology” on page 332, are completed, the deployment manager and nodes are running, and their logs are error free.

The Remote Messaging and Remote Support topology is the most commonly recommended deployment topology for production environments. This topology

creates and configures messaging, support, and application deployment in three distinct clusters.

Creating the deployment environment topology

Creating the deployment environment includes creation of the WebSphere Business Services Fabric deployment environment definition and then generating the environment using the definition. You can create the deployment environment using the deployment environments configuration wizard from the Integrated Solutions Console or by using wsadmin scripting (covered later in this chapter).

The deployment environment configuration wizard in the Integrated Solutions Console presents you with a series of panels through which you configure the components that make up your topology to be generated.

Archived

The following steps describe the deployment environment creation using the deployment environment configuration wizard:

1. Launch the wizard from the Deployment manager Integrated Solutions Console, as shown in Figure 10-2, by clicking **Servers** → **Deployment Environment**. In the new Create Deployment Environment section perform the following steps:

   a. Click the **Create a deployment environment based on a pattern** radio button.

   b. Enter a deployment environment name. WBSFRMRS is the name of the topology used here. Click **Next**.

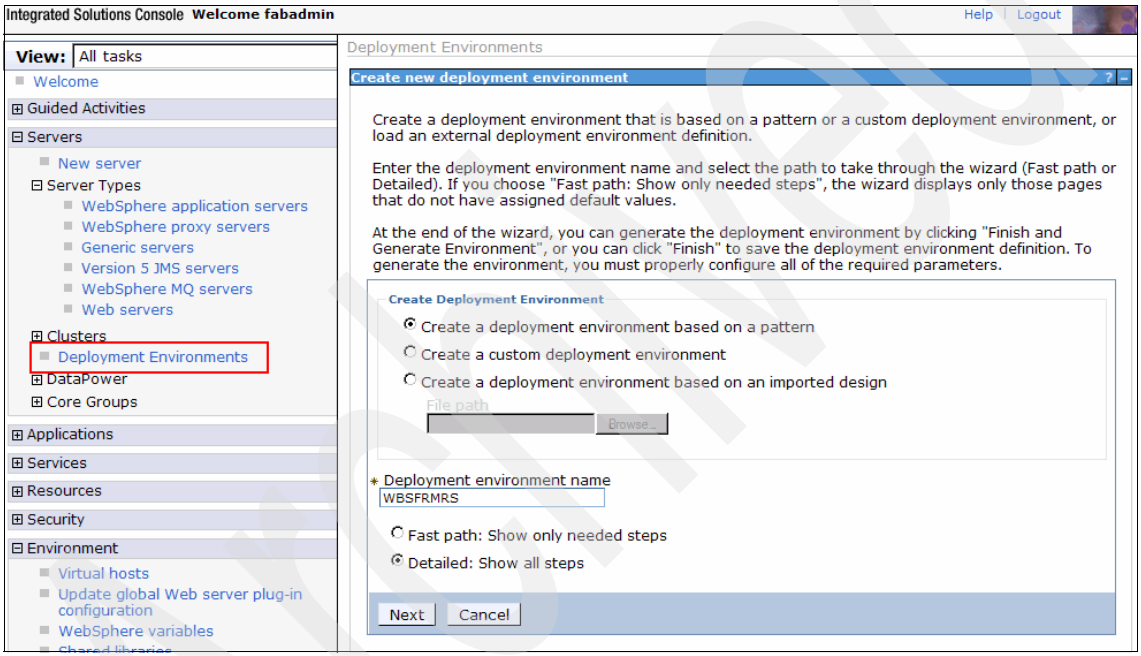


Figure 10-2 Deployment environment creation wizard

2. On the Feature Selection panel click **WBSF**, then click **Next,** as shown in Figure 10-3.

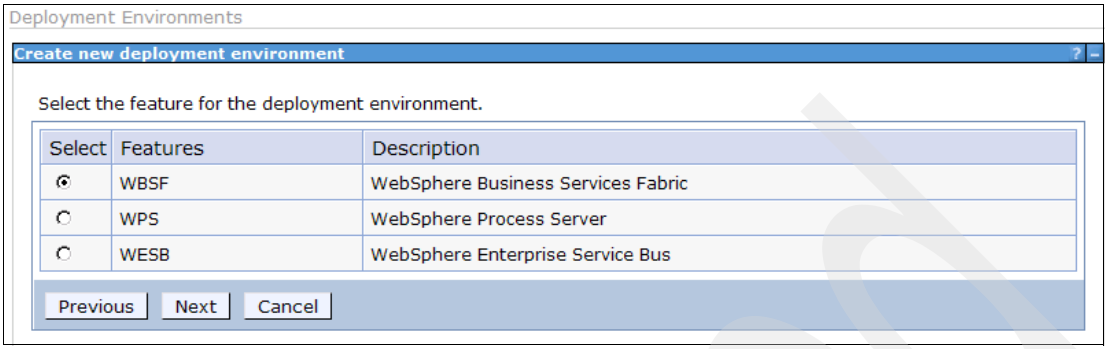


Figure 10-3 Feature selection panel in deployment environment creation wizard

3. Click the **Remote Messaging and Remote Support** radio button deployment pattern, as shown in Figure 10-4, then click **Next**.

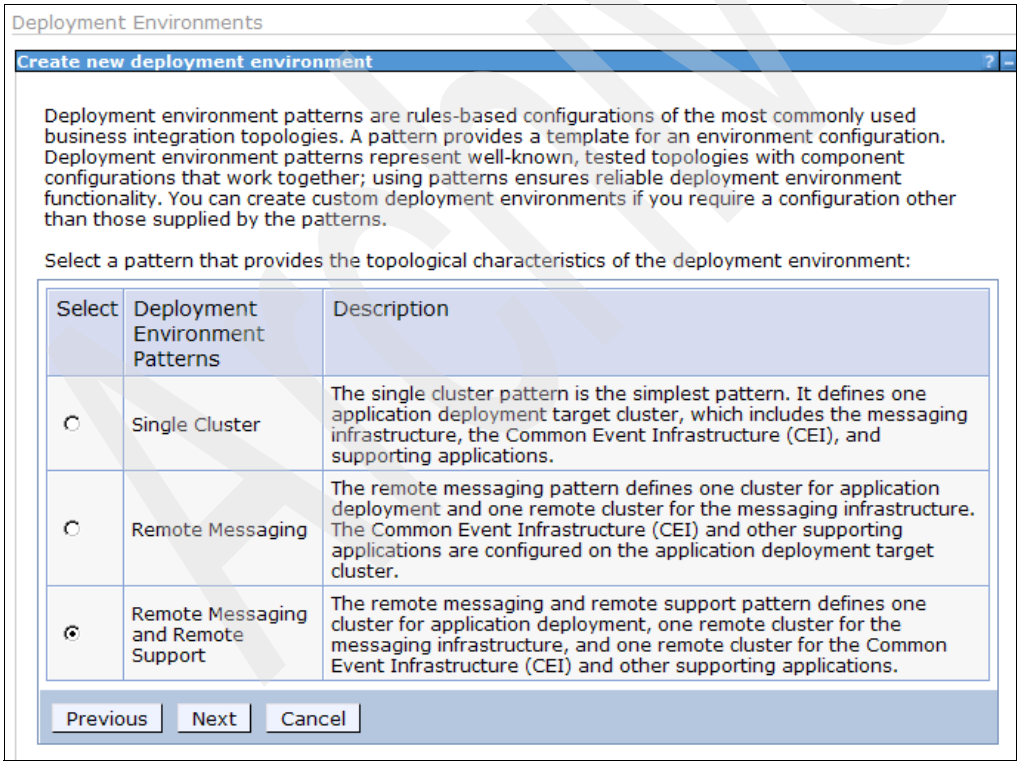


Figure 10-4 Deployment environment pattern selection panel

4. Click the WebSphere Business Services Fabric nodes for this topology and click **Next**. In our topology we use fabnode1 and fabnode2, as shown in Figure 10-5.

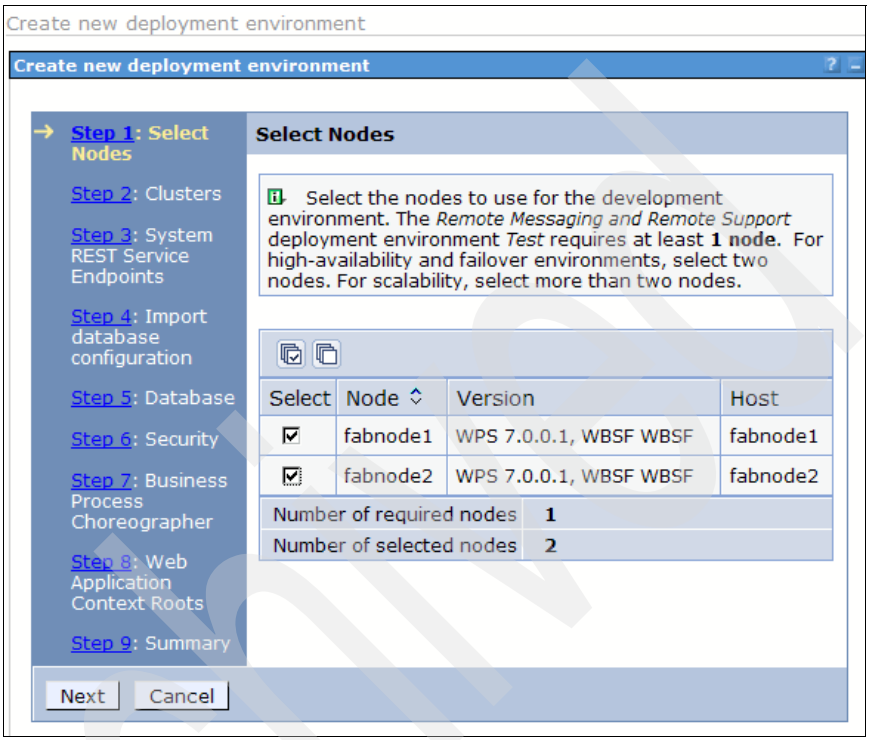


Figure 10-5 Node selection panel for the deployment environment

Archive

5. The Clusters panel shows the clusters that will be created for the application deployment target, messaging infrastructure, and support infrastructure components of the topology. Take the default values to build our topology, as shown in Figure 10-6.

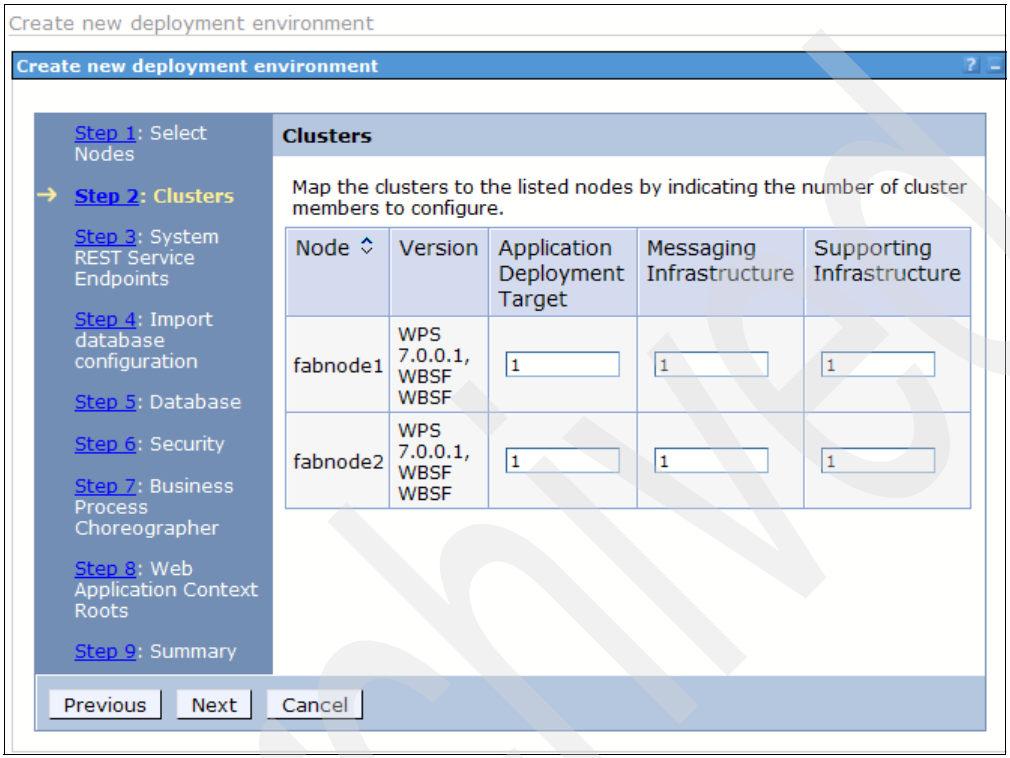


Figure 10-6 Cluster configuration for Remote Messaging and Remote Support topology pattern

Archive

6. The Representational State Transfer (REST) service endpoints must be configured for the Business Space. Typically, this endpoint is the listening host and port of an IBM HTTP Web Server. Because we are developing a production topology, we installed IBM HTTP Server to provide load-balancing of REST requests. For this topology the HTTP Server hostname and port values are as shown in Figure 10-7. Enter these values, then click **Next**.

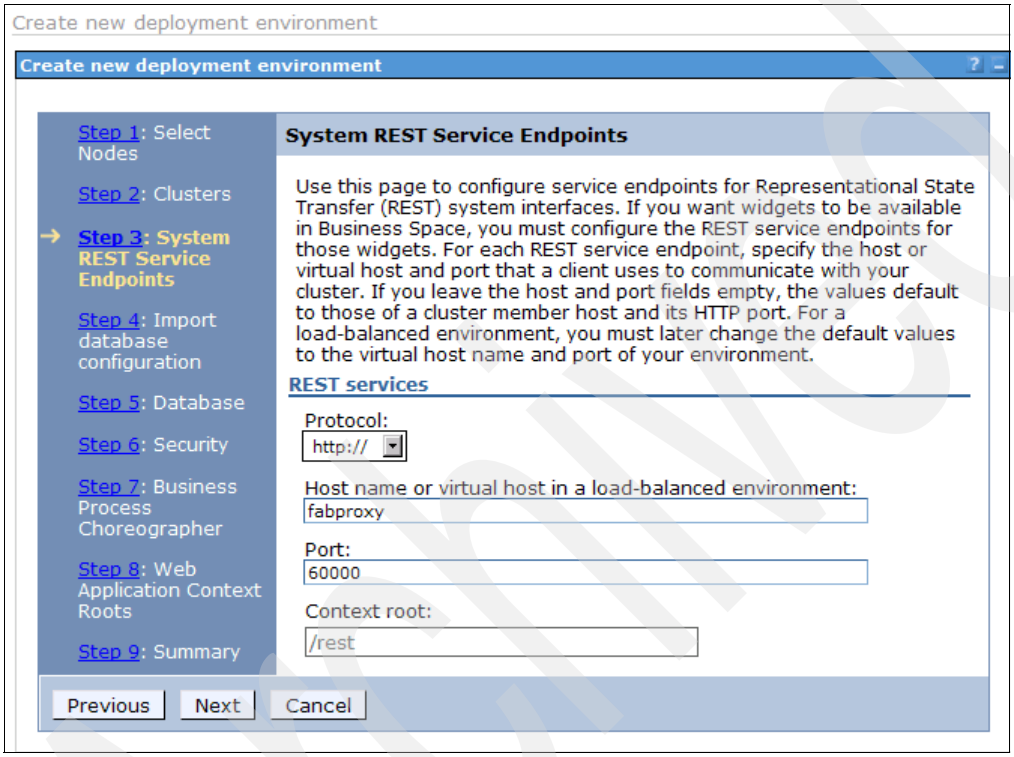


Figure 10-7 REST Service Endpoints panel

7. The deployment environment uses an existing database design to configure the data sources for the components installed in the application deployment target, messaging infrastructure, and support infrastructure clusters. Refer to 3.3, “Database and schema creation” on page 82, to find the location and name of database design created for this topology and browse to it, as shown in Figure 10-8, then click **Next**.

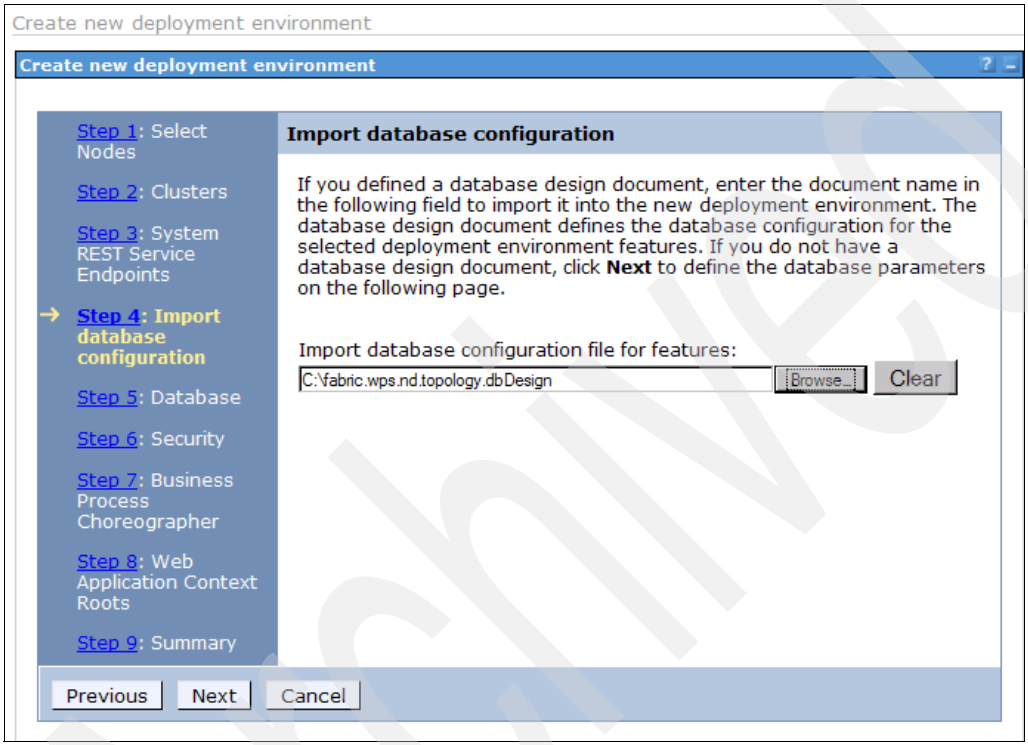


Figure 10-8 Deployment Environment: database design selection panel

Archive

8. The Import database configuration panel reads the database design selected in step 7 on page 340 and gives the user an option to make changes to the design before incorporating it in the deployment environment. In our environment, as the databases and tables are already created, the Create Tables column for all the components is deselected, as shown inFigure 10-9.

> **Note:** The Create Tables column for all the components shows as selected if you have selected false in the dbDesign tool.

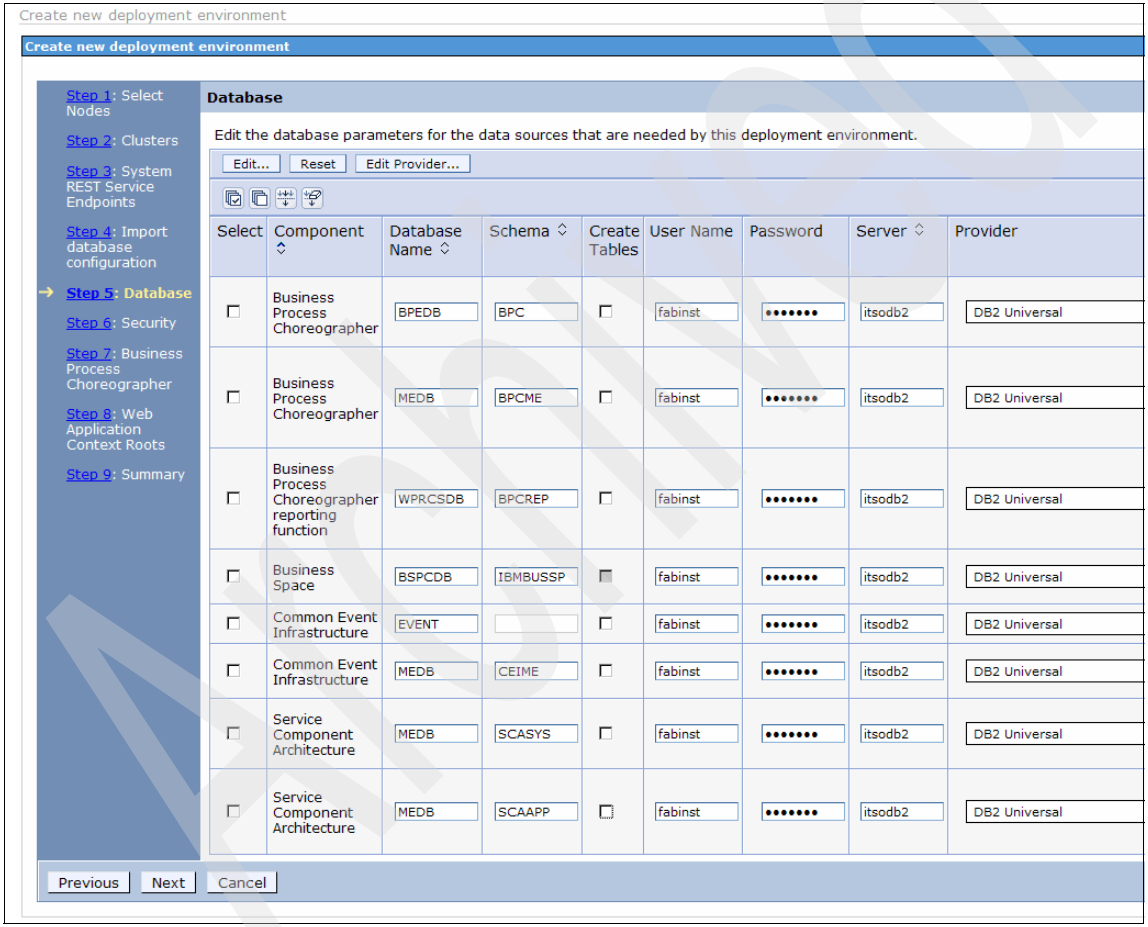


Figure 10-9 Deployment environment: Database panel

9. The Security panel presents the user with the authentication alias settings required for the Common Event Infrastructure service (CEI), Service Component Architecture (SCA), Business Process Choreographer (BPC),

and messaging components, as shown in Figure 10-10. Enter the user name and password for each component.

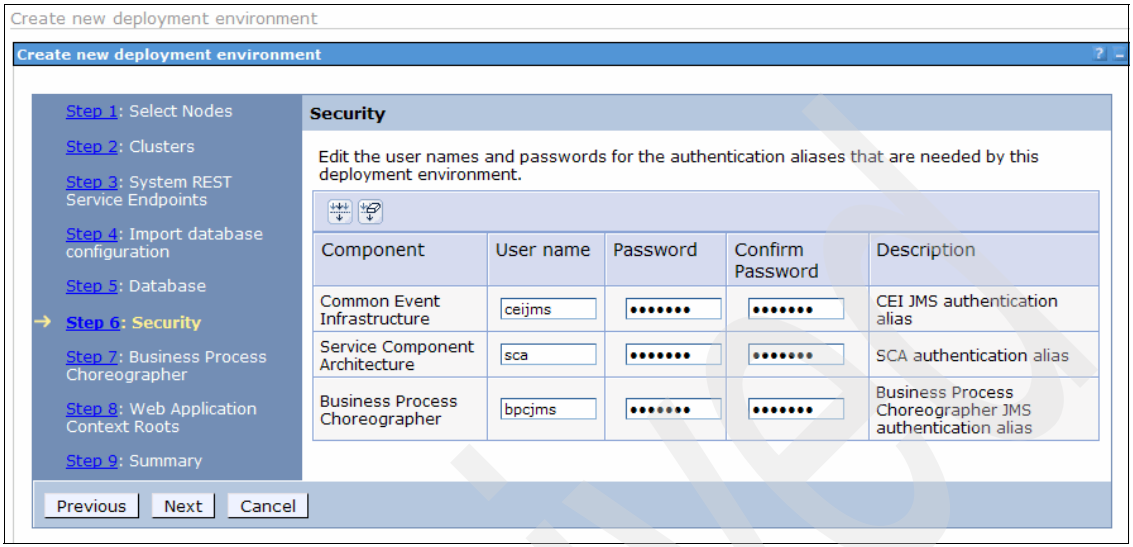


Figure 10-10 Deployment environment: Security panel

Archive

10. The Business Process Choreographer panel provides the necessary security roles and authentication settings for the Business Process Choreographer components required, as shown in Figure 10-11. Enter these values.

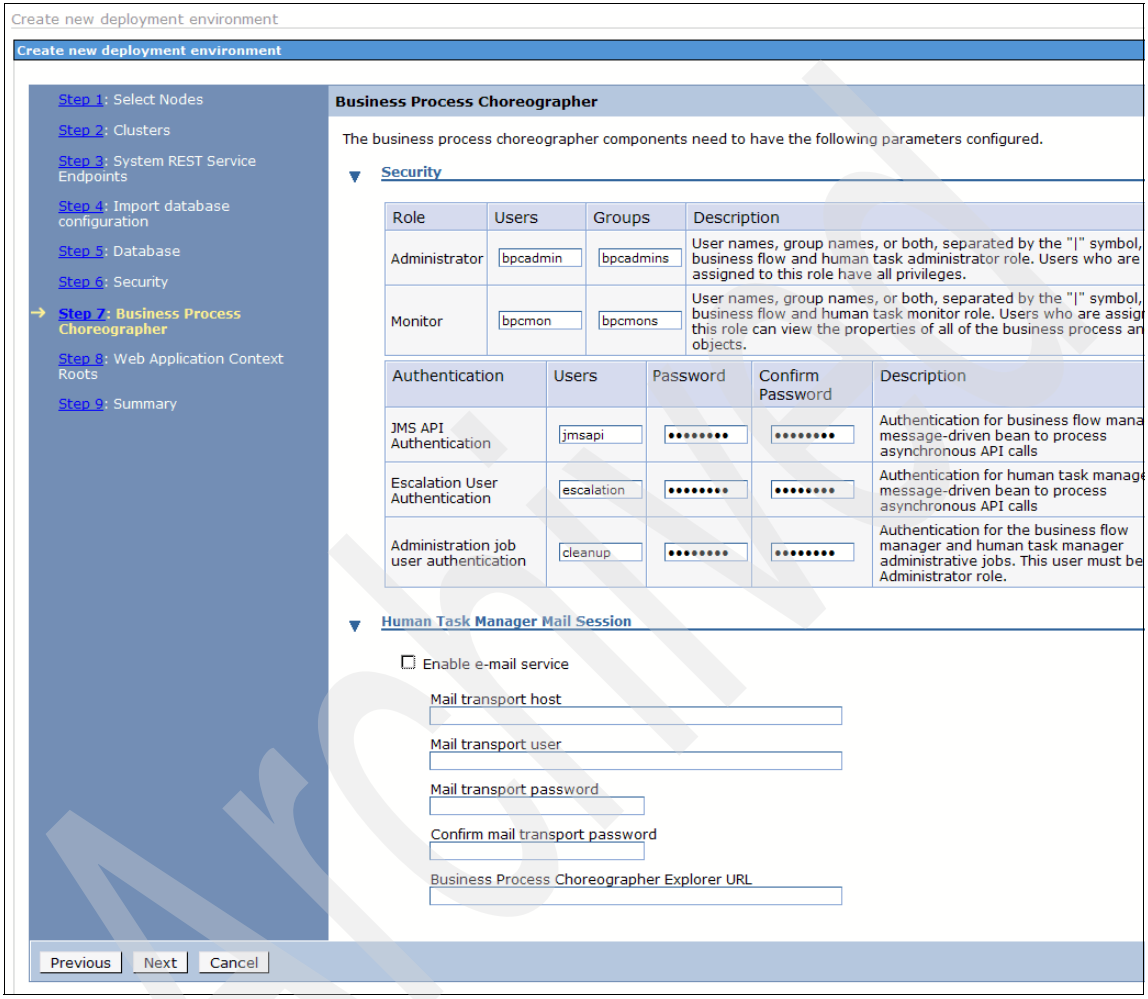


Figure 10-11 Deployment environment: Business Process Choreographer panel

11. The Web Application Context Root panel exposes the context root setting for the various user interface components, as shown in Figure 10-12. Accept the defaults.

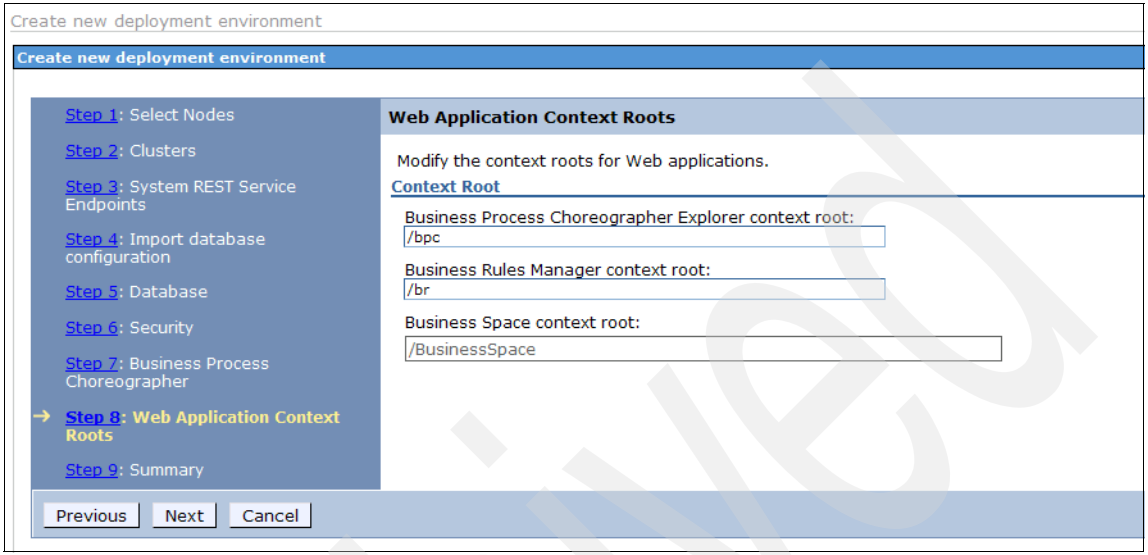


Figure 10-12 Deployment environment: Web Application Context Roots panel

12. Review the settings on the Summary panel that lists settings from the previous panels and click **Finish** or **Finish and Generate**. Clicking Finish just creates the deployment environment definition, whereas Finish and Generate also generates the topology.

13. Check for any error messages output during the environment creation. If no error messages exist click **Save Changes** → **Save** to save the master configuration.

Archived

The generated deployment environment is listed under the Deployment Environments section, as shown in Figure 10-13. This completes the steps for creating the deployment environment using the Integrated Solutions Console.

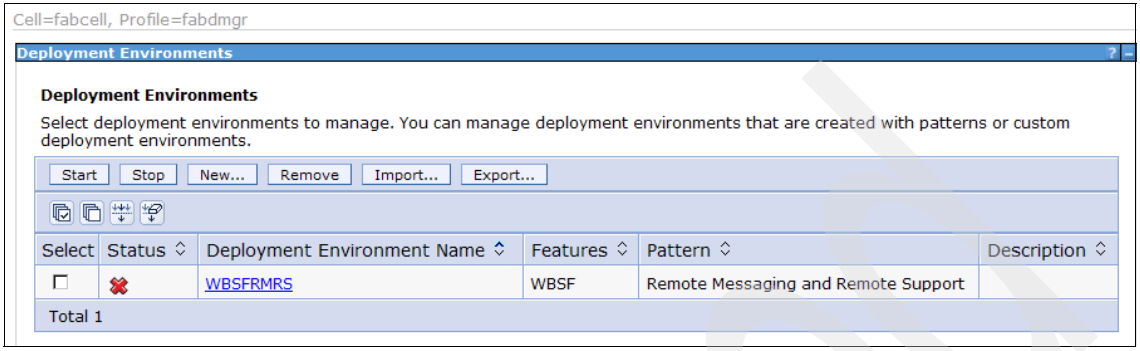


Figure 10-13 Deployment Environments panel

Note: The generation of the deployment environment takes time. Wait until it is completely generated before performing any steps in 10.5, “Post-creation configuration and verification” on page 347.

10.4 Configuring topology manually using the command line

This section describes the steps to configure a WebSphere Business Services Fabric Remote Messaging and Remote Support topology through the command line.

Archive

10.4.1 Creating the deployment environment definition

To configure a topology using the command line, a topology definition file is needed. To create this file, define a topology using the Integrated Solutions Console and then export it into an `xml` file. To create a topology definition file follow these steps:

1. Define a topology as described in "Creating the deployment environment topology" on page 334 and only save the environment definition by selecting **Finish** at the end.
2. Export the topology by opening the Integrated Solutions Console, navigating to **Servers** → **Deployment Environments**, selecting the relevant topology definition, clicking **Export**, and saving the resulting `xml` file.

Note: To provide an alternative approach if you do not have access to an administrative console at all, a template topology definition file named `WBSFRMRS.xml` for the WebSphere Business Services Fabric Remote Messaging and Remote Support topology is provided with this book that you can reuse by changing the relevant information for your environment. See Appendix C, "Deployment environment definition files" on page 439, for more information and caveats when using this approach.

You can also generate a new topology definition from scratch using **`wsadmin`** commands that are not covered in this book. More information about this approach is available at the InfoCenter at:

`http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp?topic=/com.ibm.websphere.wps.doc/doc/tcnfg_create_depenv_cmd.html`

After generating the deployment environment `xml` file, you can generate the topology using two **`wsadmin`** commands, as shown in 10.4.2, "Importing and generating Deployment Environment topology" on page 347.

After generating the topology, you can verify the deployment environment created by following the steps in 10.5, "Post-creation configuration and verification" on page 347.

10.4.2 Importing and generating Deployment Environment topology

After creating the deployment environment topology `xml` in 10.4.1, “Creating the deployment environment definition” on page 346, you can import it and generate the environment using the wsadmin script. Here we show how to do this interactively. You could also build this into a wsadmin script, with appropriate error checking. The steps are:

1. First import the deployment environment definition:

   ```
   AdminTask.importDeploymentEnvDef ([ '-filePath', fileLocation,
   '-topologyName', topologyName])
   ```

2. Assuming that this completes successfully, ensure that this configuration change is saved with:

   ```
   AdminConfig.save()
   ```

3. Generate the deployment environment itself:

   ```
   AdminTask.generateDeploymentEnv (['-topologyName', topologyName])
   ```

4. Again, assuming that this completes successfully, save the change with `AdminConfig.save()`. Do not save the changes if they do not complete successfully or you will most likely save a broken deployment environment.

Note: The generation of the deployment environment takes time. Wait until it is completely generated. Perform the steps in 10.5, “Post-creation configuration and verification” on page 347, to verify that it was created successfully.

10.5 Post-creation configuration and verification

This section describes the first steps to be performed to verify the deployment environment created using 10.3, “Configuring topology using deployment environments in Integrated Solutions Console” on page 333, or 10.4, “Configuring topology manually using the command line” on page 345.

Archived

10.5.1 Restarting the deployment manager and nodes

Before restarting the WebSphere Business Services Fabric Deployment Manager and nodes ensure that the deployment environment is completely created by watching the log files in the /opt/ibm/WebSphere/Fabric/FoundationPack/profiles/fabnode1/logs/ folder for the new deployment artifacts created by the deployment environment on fabnode1 and fabnode2 until you see messages shown in Example 10-1.

Example 10-1 End of SystemOut.log file of messaging cluster

```
00000004d FileRepositor A   ADMR0015I: User
itsoldap.itso.ral.ibm.com/server:fabcell_fabnode1_nodeagent created
document
BusinessSpace/help/eclipse/plugins/org.eclipse.help.nl1/plugin_fr.prope
rties.
00000004d FileRepositor A   ADMR0015I: User
itsoldap.itso.ral.ibm.com/server:fabcell_fabnode1_nodeagent created
document
BusinessSpace/help/eclipse/plugins/org.eclipse.help.nl1/plugin_it.prope
rties.
```

Restart the Deployment Manager and all nodes using the stop and start commands shown in Example 10-2.

Example 10-2 Commands to stop deployment manager and nodes

```
/opt/ibm/WebSphere/Fabric/FoundationPack/profiles/fabdmgr/bin/stopManag
er.sh -username <username> -password <pwssword>
/opt/ibm/WebSphere/Fabric/FoundationPack/profiles/fabdmgr/bin/startMana
ger.sh
/opt/ibm/WebSphere/Fabric/FoundationPack/profiles/fabnode1/bin/stopNode
.sh
/opt/ibm/WebSphere/Fabric/FoundationPack/profiles/fabnode1/bin/startNod
e.sh
/opt/ibm/WebSphere/Fabric/FoundationPack/profiles/fabnode2/bin/stopNode
.sh
/opt/ibm/WebSphere/Fabric/FoundationPack/profiles/fabnode2/bin/startNod
e.sh
```

10.5.2 Creating event database tables

Create and populate the event database tables. Refer to 3.3.7, “The EVENT database” on page 89, for information about how to do this.

10.5.3 Checking database connectivity

Before you start the environment check the database connectivity to ensure that the deployment environment has been generated correctly. Follow these steps to verify the database connectivity:

1. Log in to the administrative console and navigate to **Resources** → **JDBC** → **JDBC Providers** → **Data Sources**.
2. Make sure that **All scopes** is selected.
3. Select all of the data sources and click **Test connection** to make sure that they have connectivity. For each data source verify that the `connection is successful` message is displayed.

Command-line equivalent

You can use a script such as that shown in Example 10-3 to check database connectivity using wsadmin instead. You must change the cell name to the cell name that you are using.

Example 10-3 Checking database connectivity using wsadmin

```
cellName = "fabcell"

dataSources = AdminConfig.list('DataSource', AdminConfig.getid(
'/Cell:' + cellName + '/')).split("\n")

for dataSource in dataSources:
   if not dataSource.startswith('DefaultEJB'):
      print dataSource
      AdminControl.testConnection(dataSource)
```

10.5.4 Starting the clusters and deployment environment

To verify the successful creation of the deployment environment each of the clusters must be started. Starting the clusters starts all the application servers and enterprise applications deployed on the servers defined in the cluster.

Archived

When the clusters are started you must check the SystemOut.log file for each of the servers to verify there are no errors. The SystemOut.log file is in the logs folder of each profile. For example, in our environment the SystemOut.log file for the messaging cluster is in the /opt/ibm/WebSphere/Fabric/FoundationPack/profiles/fabnode1/logs/WBSFRMRS.Messaging.fabnode1.0/ folder. Example 10-4 shows an example of messages put out during startup of the messaging cluster showing that the messaging engine artifacts were started successfully.

Example 10-4 Messaging cluster startup log verification

```
BPC.fabcell.Bus is in state Started.
WBSFRMRS.Messaging.000-SCA.APPLICATION.fabcell.Bus is in state Started.
WBSFRMRS.Messaging.000-SCA.SYSTEM.fabcell.Bus is in state Started.
CEI.fabcell.BUS is in state Started.
```

Example 10-5 shows an example of messages put out during startup of the support cluster showing that the support application artifacts were started successfully.

Example 10-5 Support cluster startup log verification

```
WSVR0221I: Application started: BusinessRulesManager_WBSFRMRS.Support
WSVR0221I: Application started: BSpaceWebformsEnabler_WBSFRMRS.Support
WSVR0221I: Application started: wesbWidget_WBSFRMRS.Support
WSVR0221I: Application started:
BPMAdministrationWidgets_WBSFRMRS.Support
WSVR0221I: Application started: BusinessRules_WBSFRMRS.Support
WSVR0221I: Application started:
HumanTaskManagementWidgets_WBSFRMRS.Support
WSVR0221I: Application started: widgets_fabric_WBSFRMRS.Support
WSVR0221I: Application started: BSpaceEAR_WBSFRMRS.Support
WSVR0221I: Application started: BusinessSpaceHelpEAR_WBSFRMRS.Support
WSVR0221I: Application started: BPCECollector_WBSFRMRS.Support
WSVR0221I: Application started: BPCExplorer_WBSFRMRS.Support
WSVR0221I: Application started: mm.was_WBSFRMRS.Support
WSVR0001I: Server WBSFRMRS.Support.fabnode1.0 open for e-business
```

Archived

Example 10-6 shows an example of the messages put out during startup of the App Target cluster showing that the App Target application artifacts were started successfully.

Example 10-6 App Target startup log verification

```
WSVR0221I: Application started:
HTM_PredefinedTasks_V700_WBSFRMRS.AppTarget
WSVR0221I: Application started: REST Services Gateway
WSVR0221I: Application started: Fabric_Catalog_WBSFRMRS.AppTarget
WSVR0221I: Application
started:HTM_PredefinedTaskMsg_V700_WBSFRMRS.AppTarget
WSVR0221I: Application started: BPEContainer_WBSFRMRS.AppTarget
WSVR0221I: Application started: TaskContainer_WBSFRMRS.AppTarget
WSVR0221I: Application started: Fabric_Trace_WBSFRMRS.AppTarget
WSVR0221I: Application started: Fabric_Tools_WBSFRMRS.AppTarget
```

In the Integrated Solutions Console navigate to **Servers** → **Clusters** → **WebSphere application server clusters** to confirm that the clusters are started, as shown in Figure 10-14.

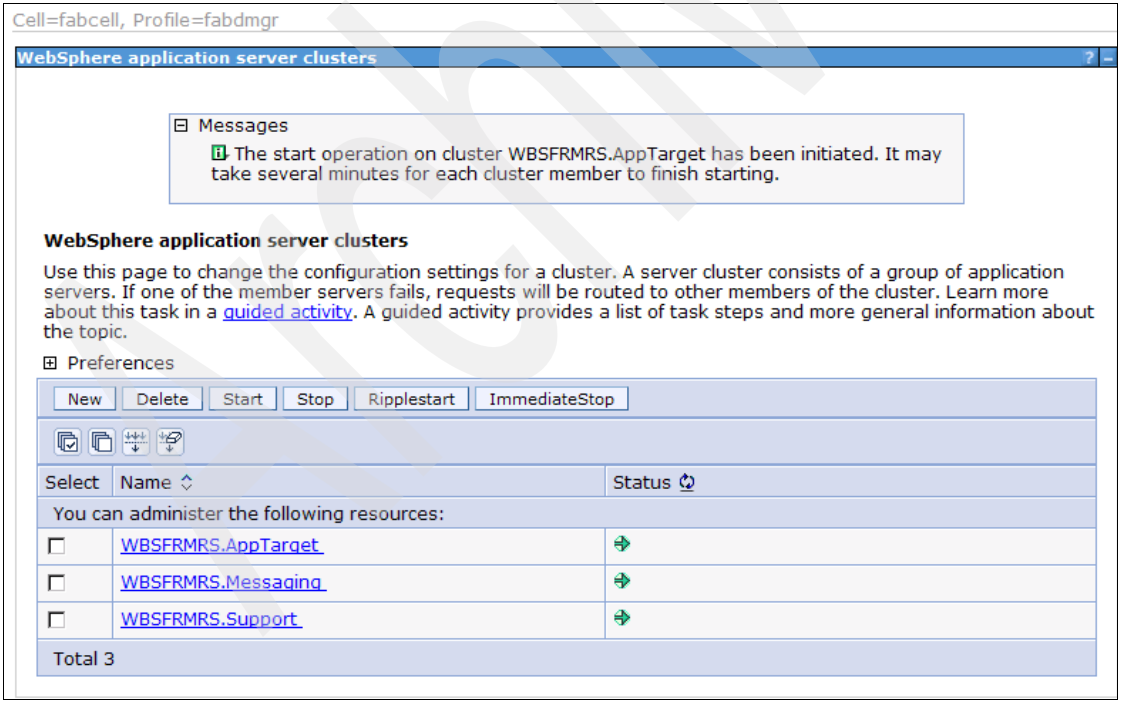


Figure 10-14 Cluster views with clusters started

The successful start of the clusters automatically starts the deployment environment. In the Integrated Solutions Console verify that the deployment environment is successfully started by clicking **Servers** → **Deployment Environments**, as shown in Figure 10-15.

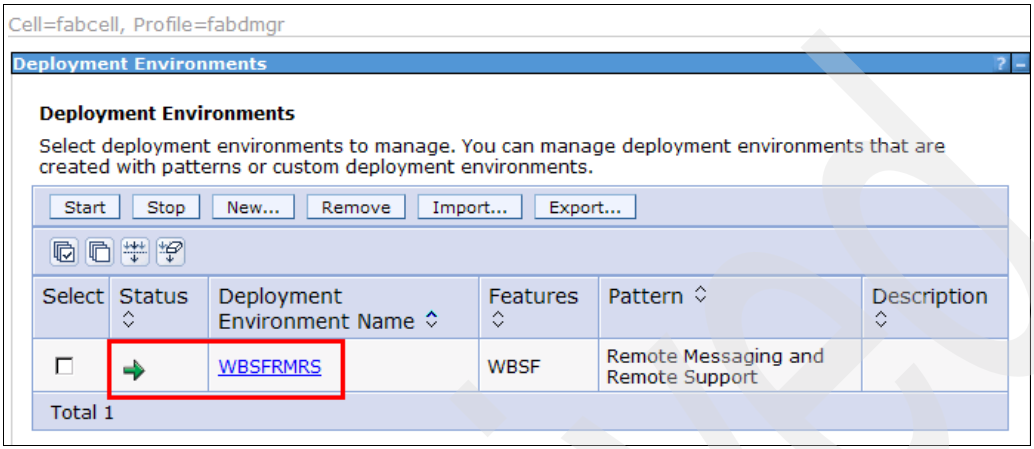


Figure 10-15 Deployment environment started

10.5.5 Installing and testing a WebSphere Business Services Fabric sample application

This section describes the steps to install and test a sample composite business application. The sample application contains a WebSphere Enterprise Service Bus Mediation Flow containing the Fabric Context Extractor wired to a Fabric Dynamic Assembly Component. It also contains a set of Fabric artifacts containing Business Services and policies. The sample application is packaged in a file named `FabricSample.zip` and can be downloaded from Appendix A, "Additional material" on page 421. The steps include:

1. Launch the WebSphere Business Services Fabric Administration Console from one of the application servers using the url:

   `http://<host_name>:<server_WC_defaulthost_port>/fabric/login.jsp`

   In our environment it is:

   `http://fabnode1:9081/fabric/login.jsp`

Log in into the console by entering the admin user ID `fabadmin` (in our environment) and password, as shown in Figure 10-16.

Figure 10-16 WebSphere Business Services Fabric Administrative Console

2. In the console expand **Governance Manager** and click **Import/Export.** Import the following files in the order listed below from the `FabricSample.zip` file, as shown in Figure 10-17:

   – `AppGlossary.zip` (contains the business vocabulary)
   – `BService.zip` (contains Business Services)
   – `SimpleApp.zip` (Contains the business application)

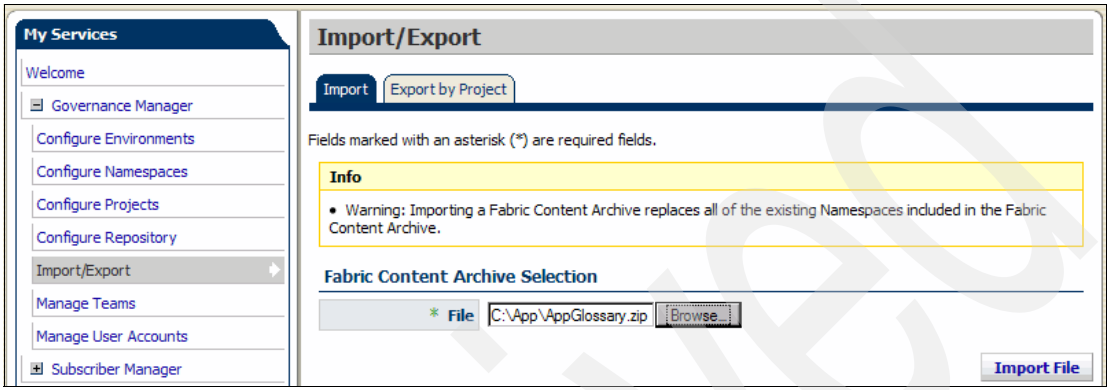


Figure 10-17 Import/Export panel in WebSphere Business Services Fabric console

3. For each file imported ignore the message:

   ```
   Warning: Importing a Fabric Content Archive replaces all of the
   existing Namespaces included in the Fabric Content Archive.
   ```

4. These imports create new namespaces and projects artifacts.

   a. Verify that the App Simple (Technical) and App-Simple (Business) artifacts are created by clicking **Governance Manager** → **Configure Namespaces**, as shown in Figure 10-18.

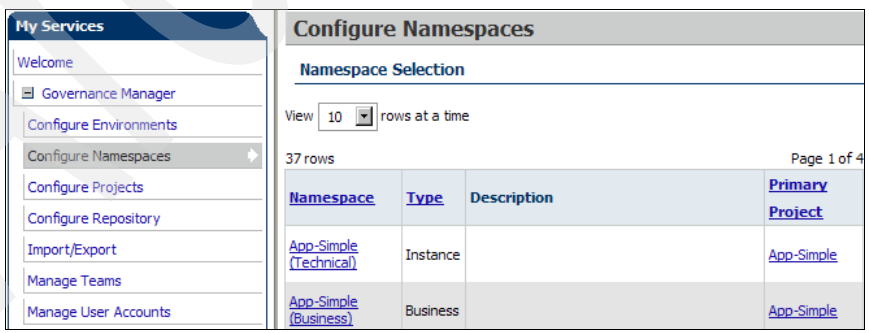


Figure 10-18 Configure Namespaces panel in WebSphere Business Services Fabric console

Archived

b. Verify that the App-Simple and BService-Simple artifacts are created by clicking **Governance Manager** → **Configure Projects**, as shown in Figure 10-19.

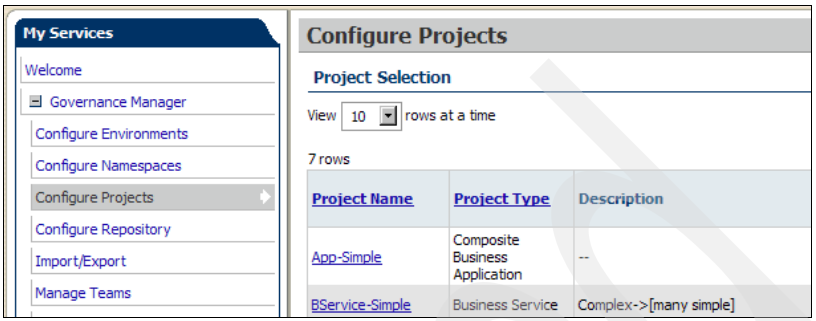


Figure 10-19 Configure Projects panel in WebSphere Business Services Fabric console

5. Install the `SimpleApp.ear` file shipped in the `FabricSample.zip` file through the WebSphere Process Server Integrated Solutions Console following these steps:
   a. Launch the Integrated Solutions Console and click **Applications** → **Application Types** → **WebSphere enterprise application** → **Install**. Enter the location of the `SimpleApp.ear` file. Click **Next**.
   b. Click the **Fast Path - Prompt only when additional information is required** radio button. Click **Next** and accept the defaults in all the panels except for the Map modules to servers panel.
   c. On the Map modules to servers panel ensure that the SimpleWeb is applied to cluster=WBSFRMRS.AppTarget.

Archived

6. Start the installed SimpleApp application by clicking **Start** on the Enterprise Applications panel and ensure it is started, as shown highlighted in Figure 10-20.

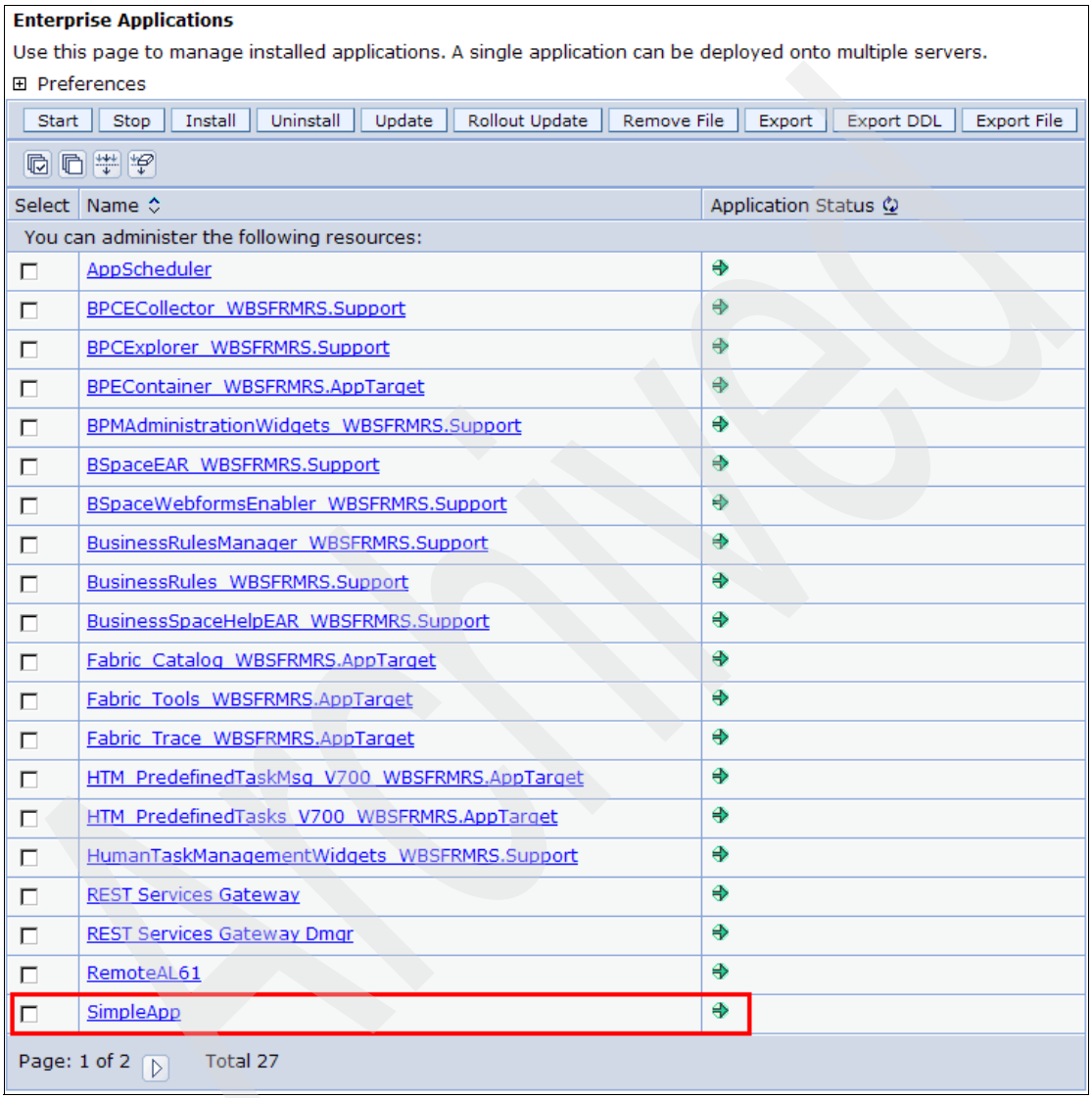


Figure 10-20 Integrated Solutions Console: Enterprise Application started panel

7. Using any SOAP UI tool, perform these steps:

   a. Import the `SimpleApp-soapui.xml` file that represents a SOAP UI project from the `FabricSample.zip`.

   b. Invoke a SOAP request against one of the WebSphere Business Services Fabric servers using the URL:

      `http://<server>:<defaulthost>/SimpleWeb/sca/Simple-HTTP`

      in our environment this URL is:

      `http://fabnode1:9081/SimpleWeb/sca/Simple-HTTP`

When a SOAP request is invoked the context extractor extracts the context from the incoming request, processes it, and passes it on to the dynamic assembly component, which in turn selects the correct endpoint or service based on the business context or policies and sends back the appropriate response. Example 10-7 shows the SOAP request sent to fabnode1.

Example 10-7 Sample App SOAP request

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sim="http://Simple/simpleInterface">
   <soapenv:Header/>
   <soapenv:Body>
      <sim:operation1>
         <input1>
            <decimal>12.2</decimal>
            <boolean>true</boolean>
            <date1>2007-02-06-06:00</date1>
            <text>hello world</text>
            <integer>32</integer>
            <enumeration>VALUE1</enumeration>
         </input1>
      </sim:operation1>
   </soapenv:Body>
</soapenv:Envelope>
```

Archived

Specifically, the sample demonstrates two functions (that is, read and process the different data types supported by the context extractor) and based on the input passed as a date, it sends back the response as true or false.
Example 10-8 shows the expected SOAP response for the SOAP request in Example 10-7 on page 357.

Example 10-8 Sample App SOAP response

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Body>
      <in:operation1Response xmlns:in="http://Simple/simpleInterface">
         <output1>true</output1>
      </in:operation1Response>
   </soapenv:Body>
</soapenv:Envelope>
```

A *true* response in `<output1>true</output1>` verifies that the WebSphere Business Services Fabric runtime is functioning correctly. This completes the topology verification process.

10.5.6 Business Space for Fabric

WebSphere Business Services Fabric provides predefined templates on Business Space that help a business user to create, deploy, and govern composite business applications.

To launch Business Space provided by WebSphere Business Services Fabric follow these steps:

1. Business Space is launched by using the URL:

   `https://<server_host>:<WC_defaulthost_secure _port>/BusinessSpace`

   For our environment the URL is:

   `http://fabnode1:9082/BusinessSpace`

Archived

Launching Business Space provides a login page, as shown in Figure 10-21. Log in using a user ID fabadmin (in our environment) and password.

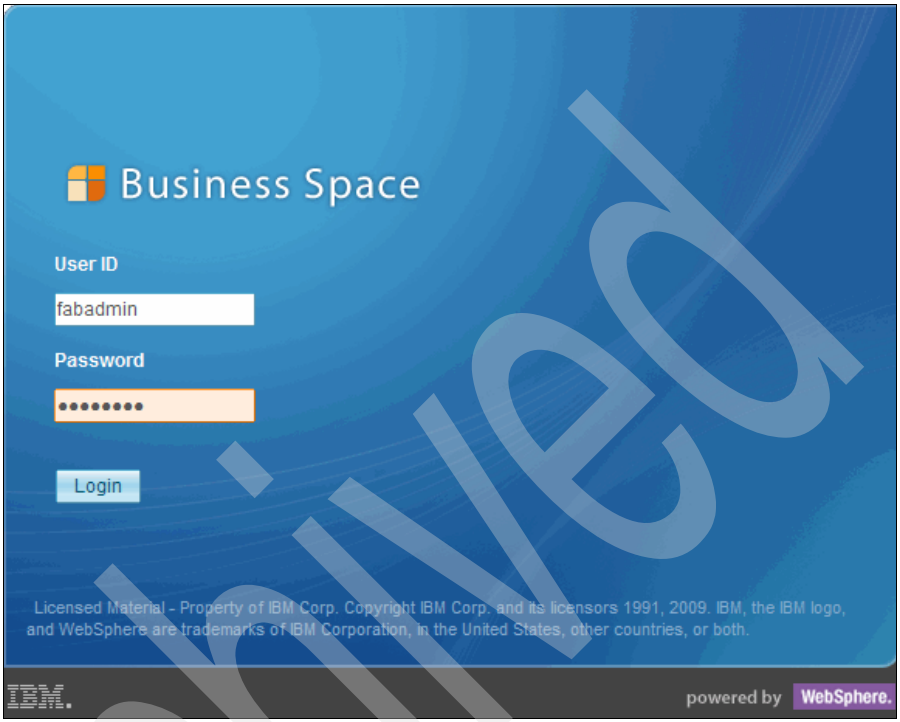


Figure 10-21 Business Space: login page

Archived

2. On the Welcome to Business Space home page click **Manage Spaces** to launch the space manager. Click **Create Space** from the Space Manager page to create a new WebSphere Business Services Fabric Business Space. Enter the required values, as shown in Figure 10-22, and click **Save**.

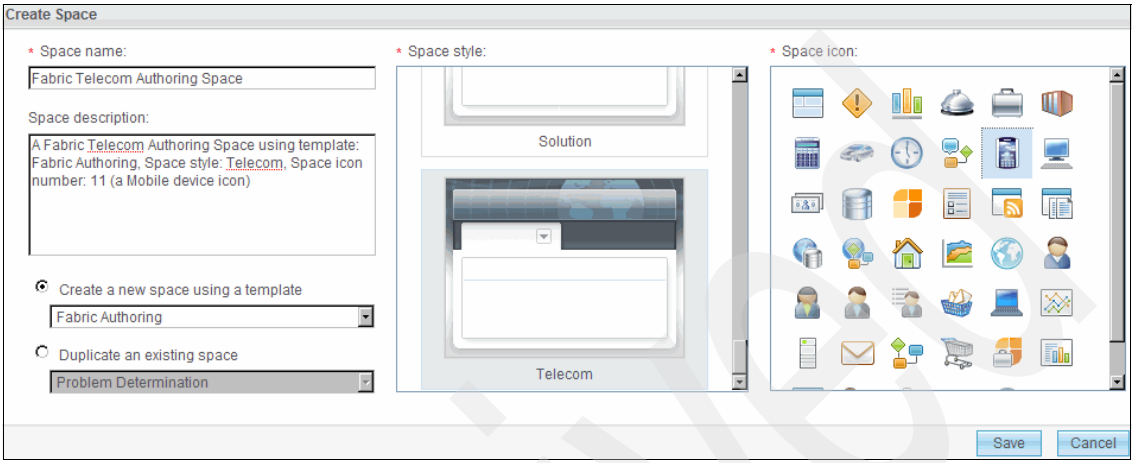


Figure 10-22 Business Space: Create Space page

3. Click the newly created **Fabric Telecom Authoring Space** link to launch this Business Space. This Business Space is launched with the Getting Started page. Click the edit page to see various widgets provided by WebSphere Business Services Fabric to build and customize the space, as shown in Figure 10-23.

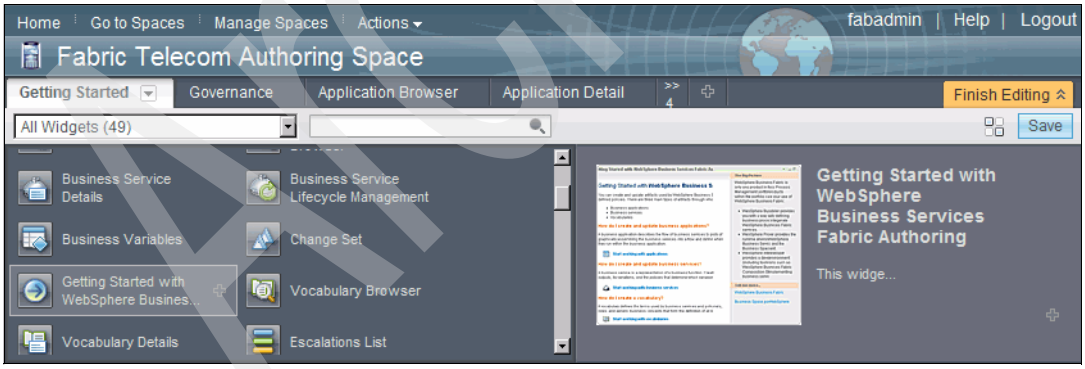


Figure 10-23 Fabric Telecom Authoring Space sample

The above steps demonstrate how to launch and create of a new Business Space for WebSphere Business Services Fabric using the predefined templates.

11

WebSphere Dynamic Process Edition

This chapter provides step-by-step instructions for building the WebSphere Dynamic Process Edition V7.0 production topology on a SUSE Linux Enterprise Server 10 64-bit environment. The products included in this topology are:

- WebSphere Process Server
- WebSphere Business Monitor
- WebSphere Business Services Fabric

We start to build a standard Remote Messaging, Support and Web deployment environment topology. In a second step, we manually add a fifth cluster for the monitor applications to separate the business applications from the monitoring applications.

© Copyright IBM Corp. 2010. All rights reserved.

11.1 Topology summary

Figure 11-1 shows a diagram of the topology that we create in this chapter. In this topology, all of the functional components (user applications, messaging infrastructure, Common Event Infrastructure service (CEI), Monitor model, Fabric application, and support applications) run in different clusters. The messaging engines run on one cluster member at a time. The active messaging engines are shown in white.

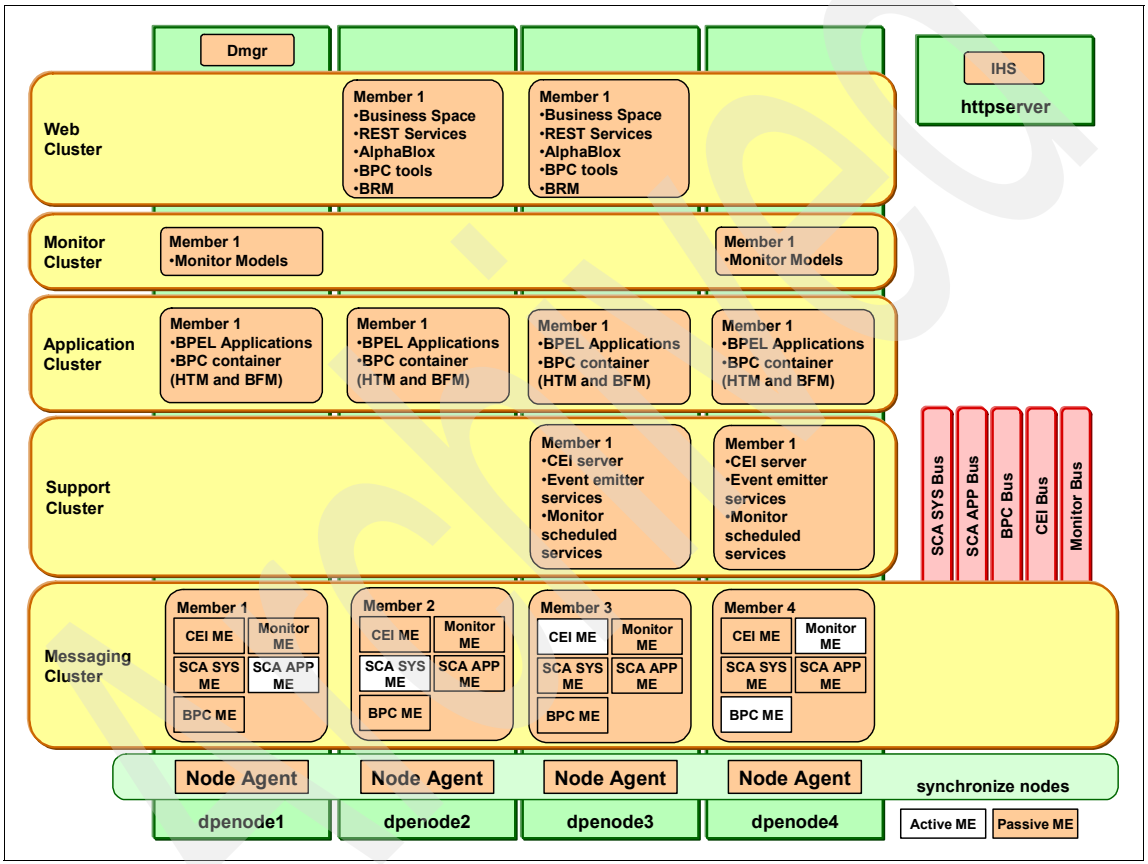


Figure 11-1 Five-cluster topology used in this chapter

Note: This chapter builds a five-cluster topology for WebSphere Dynamic Process Edition. The normal practice is to create a four-cluster deployment environment. Adding a fifth cluster (the monitor cluster) is an optional step to isolate WebSphere Business Monitor from WebSphere Process Server and WebSphere Business Services Fabric. We show that additional step in this chapter.

Additionally, we have a Web server running on host httpserver. All incoming HTTP requests go through the Web server at default port 80. The deployment manager also runs on host dpenode1 because of limited hardware. For a production environment it commonly run on a separate node.

11.2 Prerequisites for creating this topology

The WebSphere Dynamic Process Edition installation described in this chapter uses fully qualified host names. Figure 11-1 on page 362 shows the host names used in this topology. The following host names must be defined on all servers that are part of this topology. In our case we added host names to `/etc/hosts` on each server.

Table 11-1 Host names

Host name	Description
itsodb2.itso.ral.ibm.com	This server hosts the DB2 instance.
itsoldap.itso.ral.ibm.com	This server hosts the Lightweight Directory Access Protocol (LDAP) Tivoli Directory Server.
dpenode1.itso.ral.ibm.com	This server runs the WebSphere Application Server Deployment Manager and the first WebSphere Dynamic Process Edition node.
dpenode2.itso.ral.ibm.com	Second WebSphere Dynamic Process Edition node.
dpenode3.itso.ral.ibm.com	Third WebSphere Dynamic Process Edition node.
dpenode4.itso.ral.ibm.com	Fourth WebSphere Dynamic Process Edition node.
httpserver.itso.ral.ibm.com	IBM HTTP server as the central entrance host.

Archive

Before you start creating this topology, you must have completed all applicable prerequisite steps described in Chapter 3, “Preparing your topology” on page 55:

- Install a database server as described in “Software versions used in this book” on page 56.
- Install an LDAP server as per “Software versions used in this book” on page 56.
- Define the appropriate users in your user registry as per 3.4.2, “Configuring LDAP as the user account registry” on page 99.
- Install WebSphere Dynamic Process Edition as per 3.2, “Software installation” on page 57. It is important to un-tar/un-zip the two separate software packages for WebSphere Dynamic Process Edition *into the same directory*. Depending on your hardware and environment, installing the software silently could take time (over one hour in a shared virtual machines environment) and you do not get any progress indication while the installation is running.
- Create the required databases listed in Table 11-2 in the DB2, as described in 3.3, “Database and schema creation” on page 82. For WebSphere Dynamic Process Edition this is a two-step process. First you must generate the DDL scripts for Monitor and in a second step, generate the DDL scripts for WebSphere Process Server. Two database templates named dpe.wbm.dbDesign and dpe.wps.dbDesign will be delivered in Appendix A, “Additional material” on page 421 and can be used as a starting point. Table 11-2 shows the needed databases and the relevant section with further details about how to create the artifacts.

Table 11-2 Databases used for WebSphere Dynamic Process Edition

Database	Section with further details
WPRCSDB	3.3.2, “The Common database” on page 86
BPEDB	3.3.3, “The Business Process Choreographer database” on page 87
MEDB	3.3.4, “The messaging engine database” on page 87
OBSVRDB	3.3.5, “The business process reporting database” on page 88
BSPCDB	3.3.6, “The Business Space database” on page 88
EVENT	3.3.7, “The EVENT database” on page 89
MONDB	3.3.8, “The MONITOR database” on page 89

Note: Tables for the event database are not created now, but after the deployment environment has been created, so do not execute the steps in 3.3.7, “The EVENT database” on page 89, yet.

- Create a deployment manager profile of template type `dmgr.wdpe` as described in 3.4, “Creating a deployment manager profile” on page 91. The sample response file named `dpe_dmgr_create.resp` that we used in this chapter can be found in Appendix A, “Additional material” on page 421.
- Install and add the IBM HTTP Server to the cell as described in 3.6.1, “Installing IBM HTTP Server” on page 111, and 3.6.2, “Add IBM HTTP Server to the cell” on page 111.
- Create and federate four custom nodes as per 3.5, “Creating custom profiles” on page 107. A sample response file named `dpe_node_create.resp` can be found in Appendix A, “Additional material” on page 421.

11.3 Configuring topology using deployment environments in Integrated Solutions Console

The following procedure creates a Remote Messaging, Support and Web topology using the Integrated Solutions Console. In a manual step, a separate Monitor cluster is created to separate the business processes from the monitoring processes. Before you begin, ensure that the deployment manager and all nodes are running, and that their logs are error free.

Archive

11.3.1 Define and configure a deployment environment

To do this:

1. Log in to the Integrated Solutions Console as the primary administrative user that you defined when you created the profile. Here we use `dpeadmin`.
2. Navigate to **Servers Deployment Environments**. Click **New**. The system displays the first page of the deployment environment configuration wizard with **Create new deployment environment** selected. Enter the name of your deployment environment (we use `dpe`), as shown in Figure 11-2, then click **Next**.

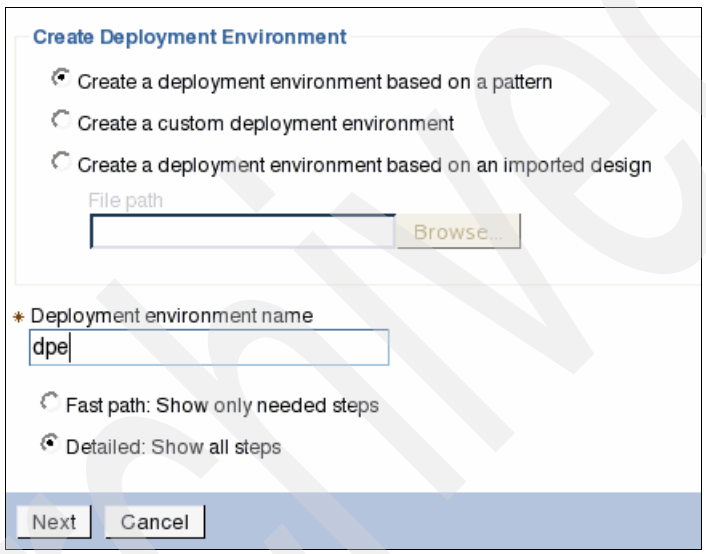


Figure 11-2 Create new deployment environment named dpe

Archived

3. Ensure that **WBSF** is selected as the deployment environment type on the create new deployment environment wizard, as shown in Figure 11-3, then click **Next**.

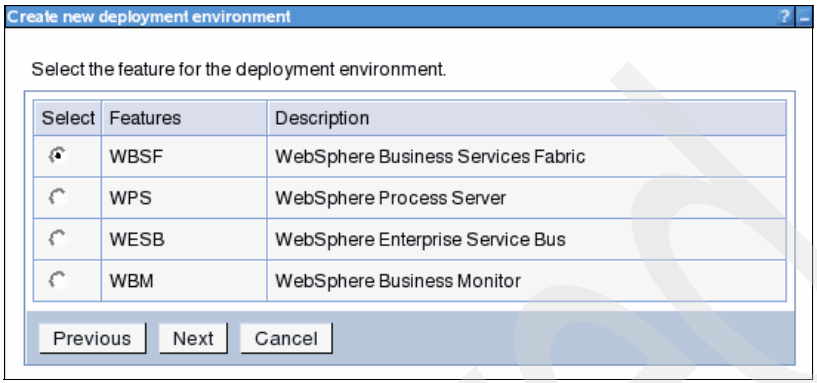


Figure 11-3 Select WebSphere Business Services Fabric

4. Select **WebSphere Business Monitor** as an optional feature, as shown in Figure 11-4.

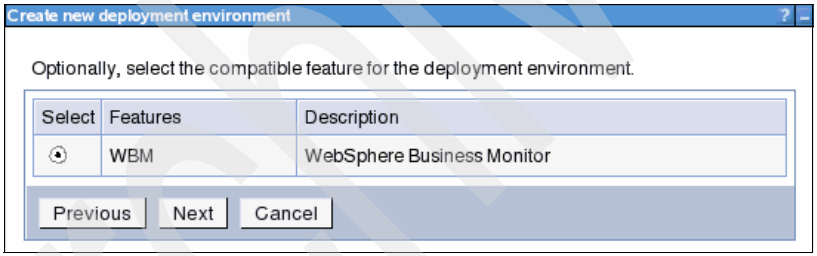


Figure 11-4 Select WebSphere Business Monitor as optional feature

Archived

5. Change the deployment environment pattern to **Remote Messaging, Remote Support, and Web**, as shown in Figure 11-5.

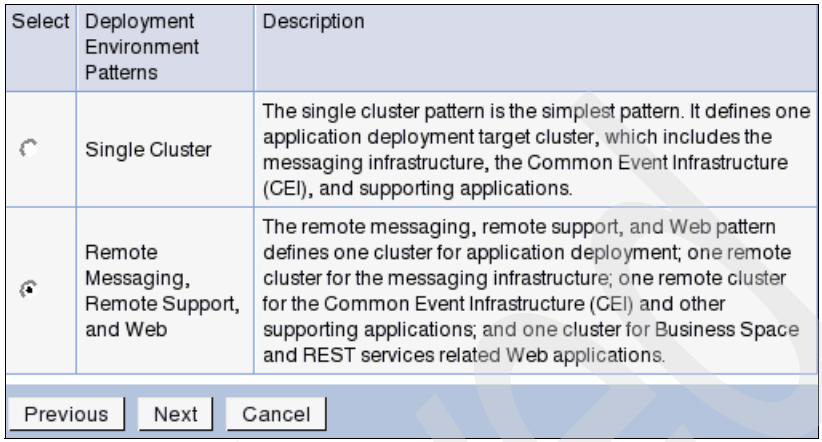


Figure 11-5 Use Remote Messaging and Remote Support deployment pattern

Archived

6. You will now run through a series of steps in the wizard, specifying how the deployment environment should be created. For step 1, select all four nodes, as shown in Figure 11-6.

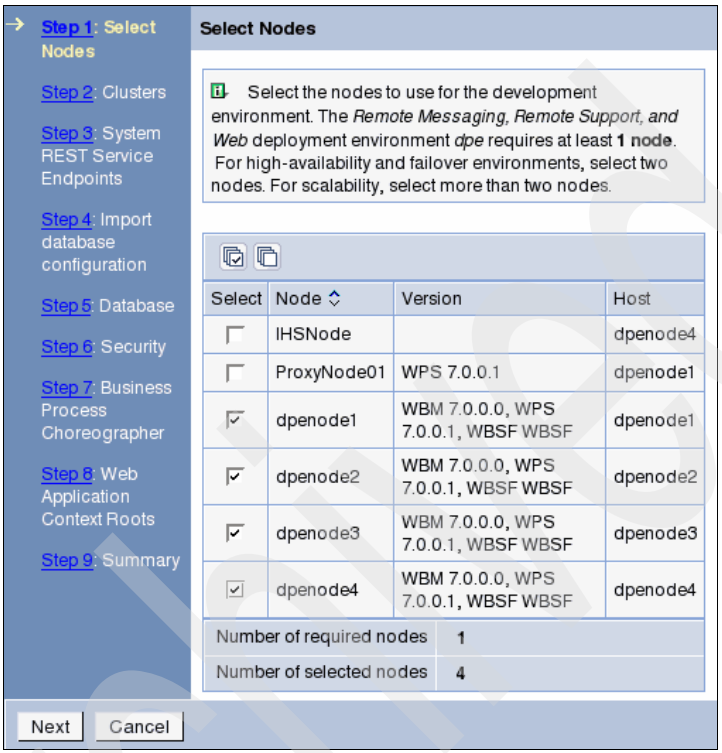


Figure 11-6 Select the nodes for your deployment environment

Archived

7. For step 2, map the clusters to your nodes, as shown in Figure 11-7.

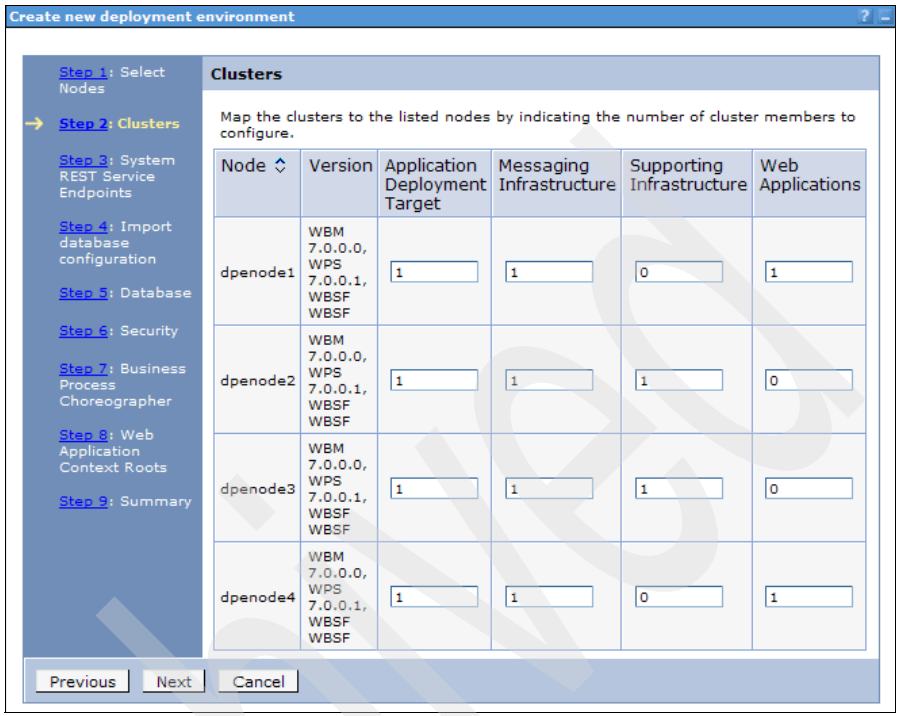


Figure 11-7 Map cluster members to nodes

Archive

8. For step 3, configure the Representational State Transfer (REST) service endpoint and point it to your Web server, as shown in Figure 11-8.

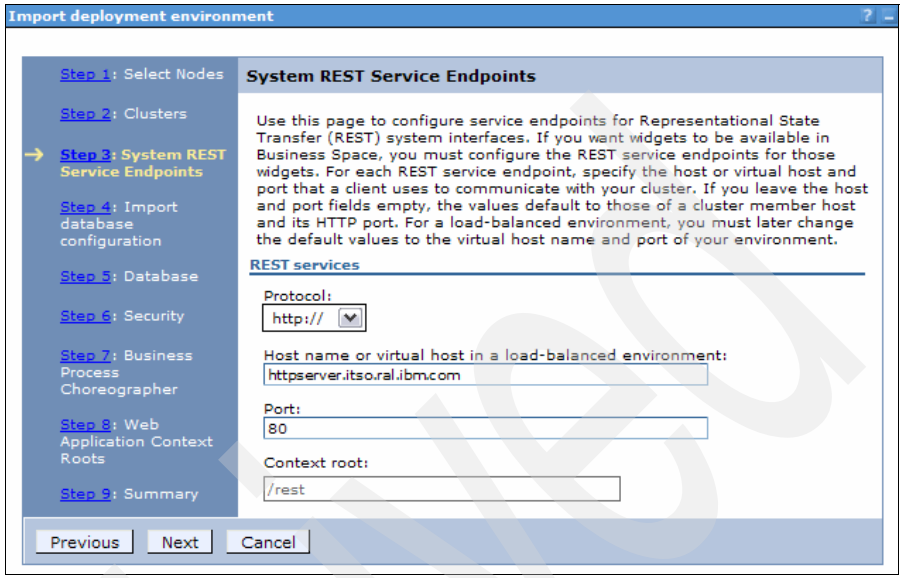


Figure 11-8 Configure REST service endpoints: use the Web server

Archived

9. For step 4, import the database configuration for WebSphere Process Server (`dpe.wps.dbDesign`) and for WebSphere Business Monitor (`dpe.wbm.dbDesign`) that you created earlier in the database creation section in 11.2, “Prerequisites for creating this topology” on page 363, as shown in Figure 11-9.

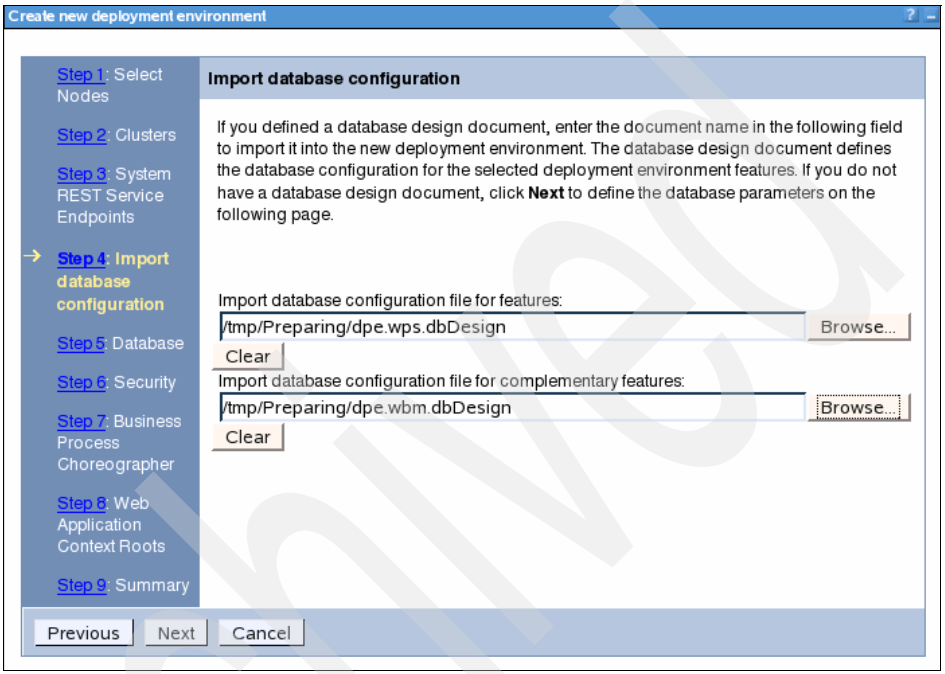


Figure 11-9 Import previously generated database configuration

Archive

10. For step 5, verify that the Create Tables check boxes for all your databases are not selected, as indicated in Figure 11-10, since the database setup was done already with the database generation tool.

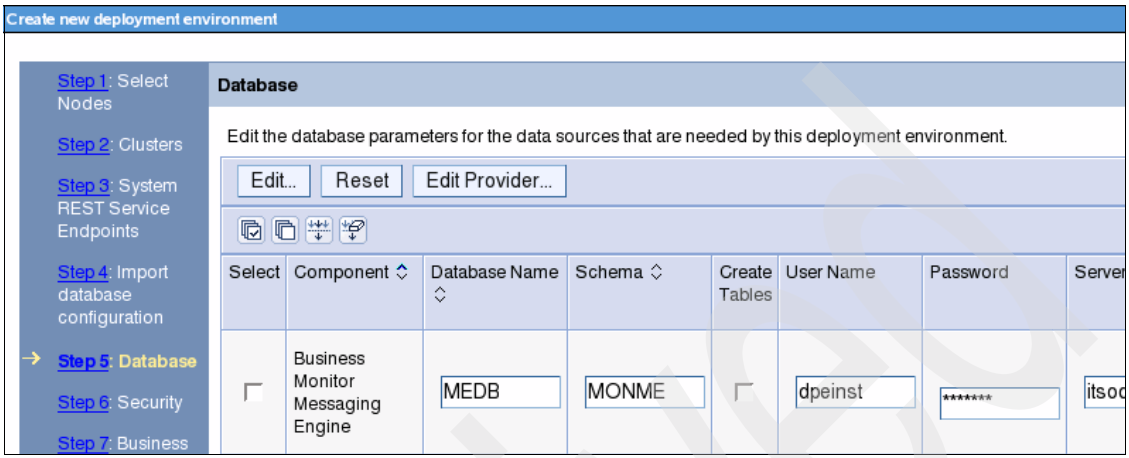


Figure 11-10 Configure the databases

Archive

11. For step 6, enter your user names and passwords as they are defined in your LDAP server, as shown in Figure 11-11. In this environment, the following user names are used:
    - ceijms: Common Event Infrastructure
    - sca: Service Component Architecture
    - bpcjms: Business Process Choreographer
    - abx: WBM_Alphablox

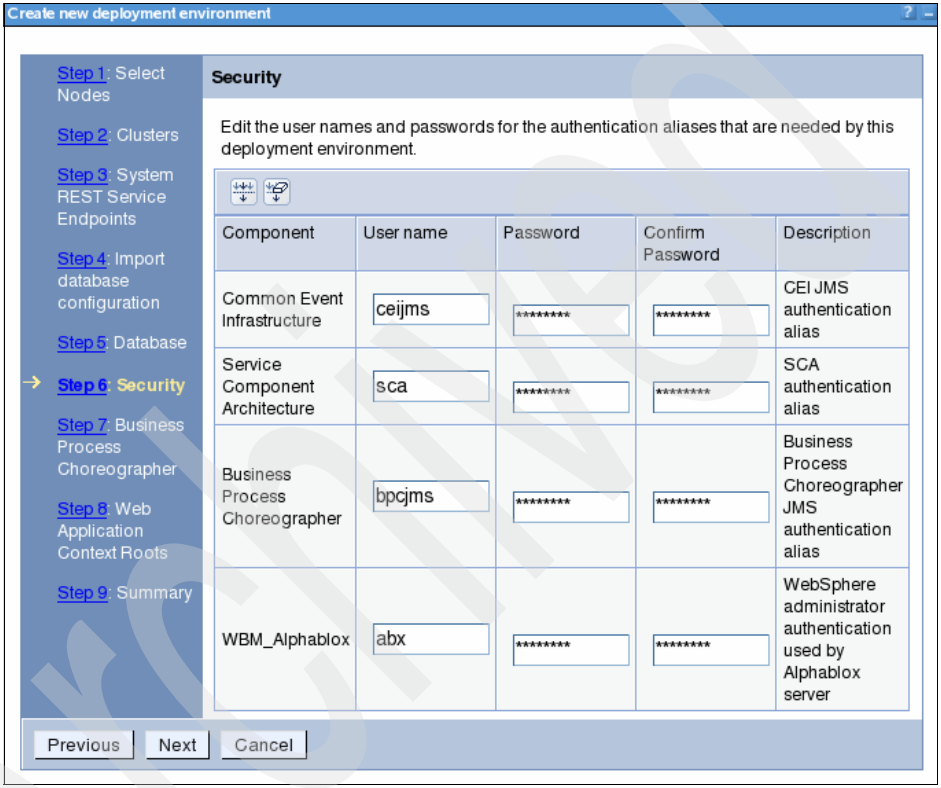


Figure 11-11 User and password used as defined in your LDAP server

Archive

12. For step 7, again enter your user names and passwords and decide whether to select or unselect the **Enable e-mail service** check box, as indicated in Figure 11-12 and Figure 11-13 on page 376. In this environment, the following users
are used:
    - dpeadmin: Administrator
    - monadmin: Monitor
    - jmsapi: JMS API Authentication
    - escalation: Escalation User Authentication
    - cleanup: Administration job user authentication

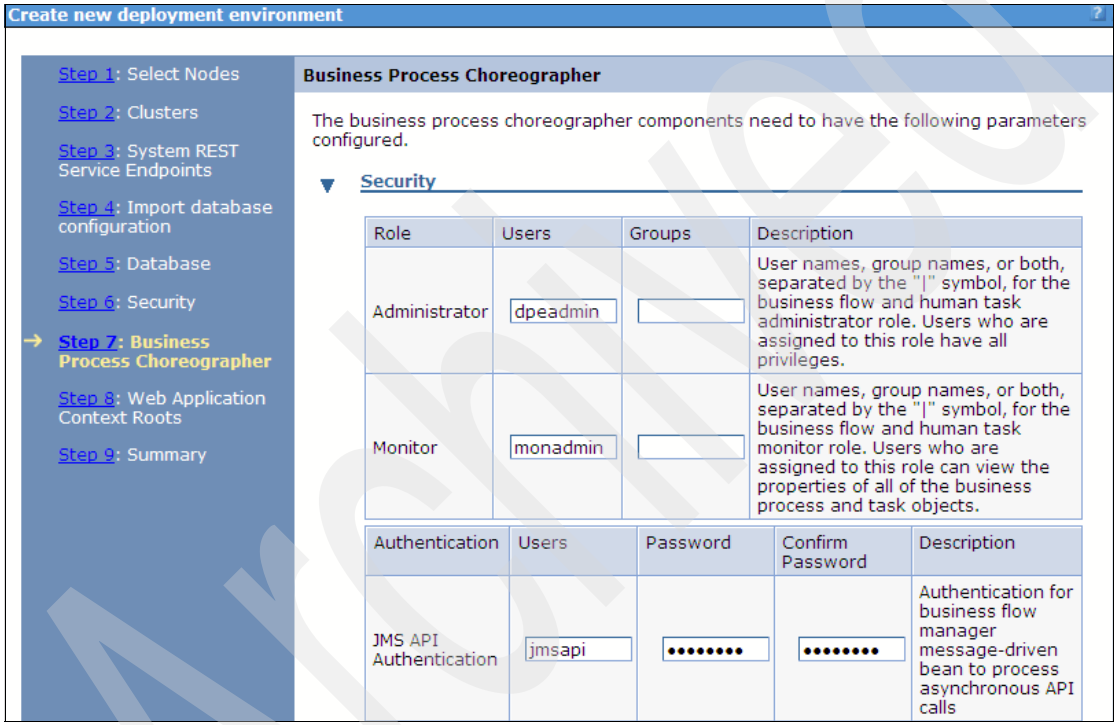


Figure 11-12 Security settings: part 1

Archived

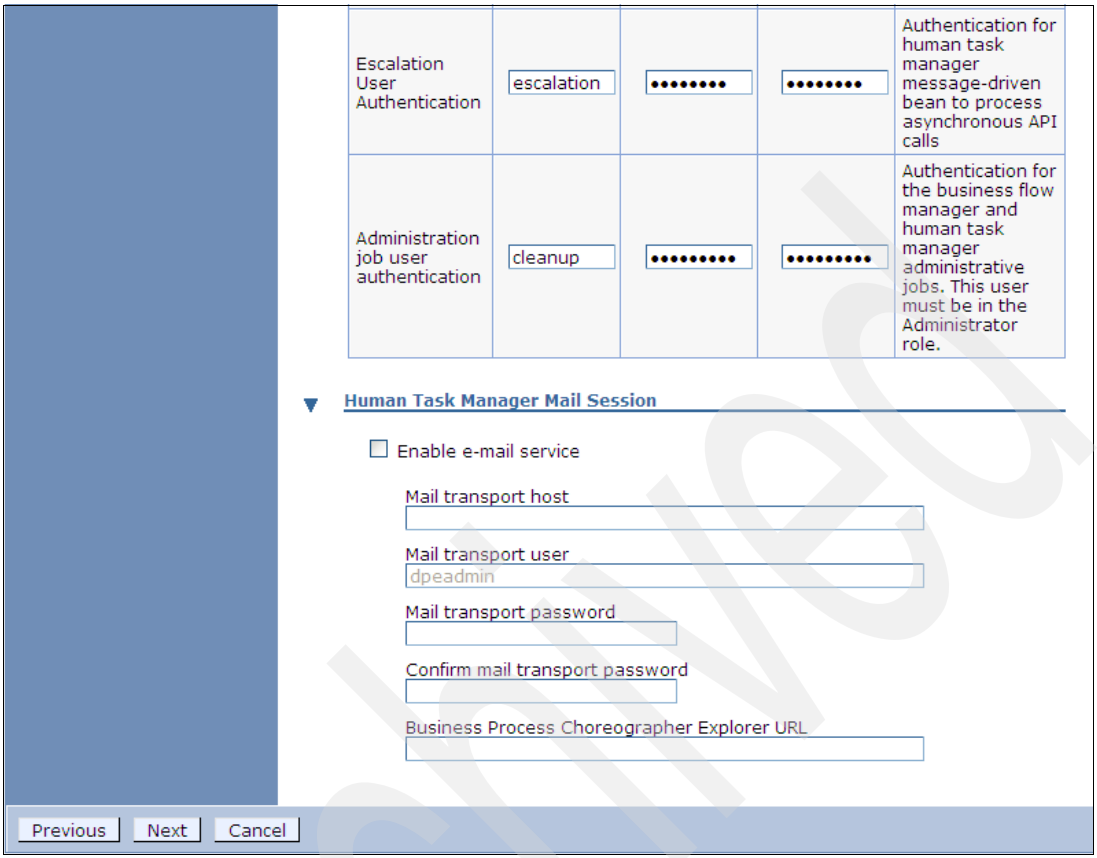


Figure 11-13 Security settings: part 2

13. Keep the defaults for step 8.
14. Step 9 displays the summary page. Click **Finish** and generate the environment later. Doing it this way gives you the opportunity to save your deployment environment to an `xml` file and change or reuse it (see also 11.4, “Configuring the topology manually using wsadmin” on page 380).

Archive

11.3.2 Generate the deployment environment

To do this:

1. Click the newly generated deployment environment named **dpe** and click **Generate Environment**.
2. After several minutes, your environment should be created successfully. Save your changes.
3. Go back to your dpe deployment environment and click **Configuration Done**, as indicated in Figure 11-14, since your database setup was already done in a previous step.

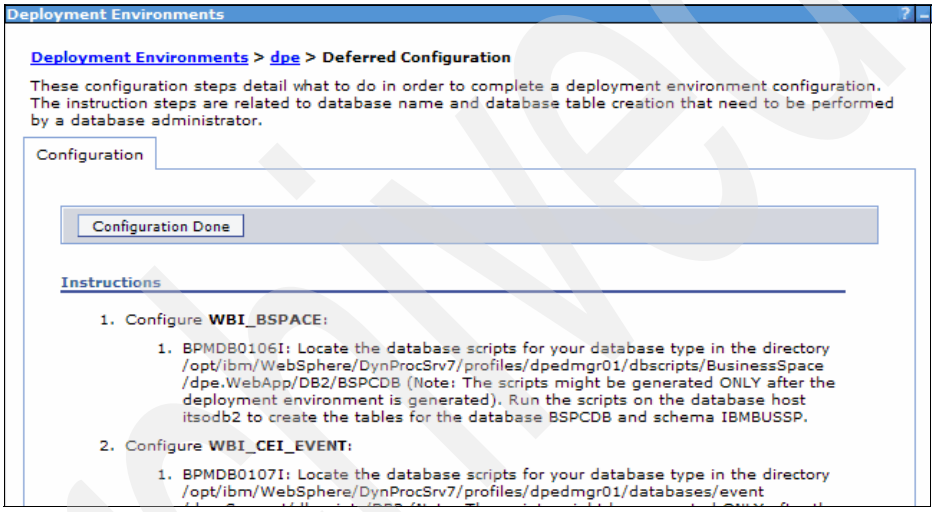


Figure 11-14 Deferred configuration pane

Archived

11.3.3 Create the Monitor cluster using Integrated Solutions Console

To do this:

1. Go to **Servers** → **WebSphere application server clusters** and create a new cluster named `dpe.Monitor`, as indicated in Figure 11-15.

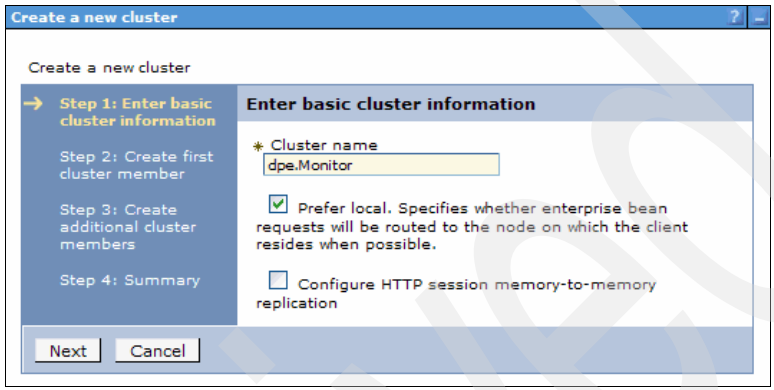


Figure 11-15 Create new cluster dpe.Monitor

Archived

2. Create the first cluster member `dpe.Monitor_member01`, as shown in Figure 11-16.

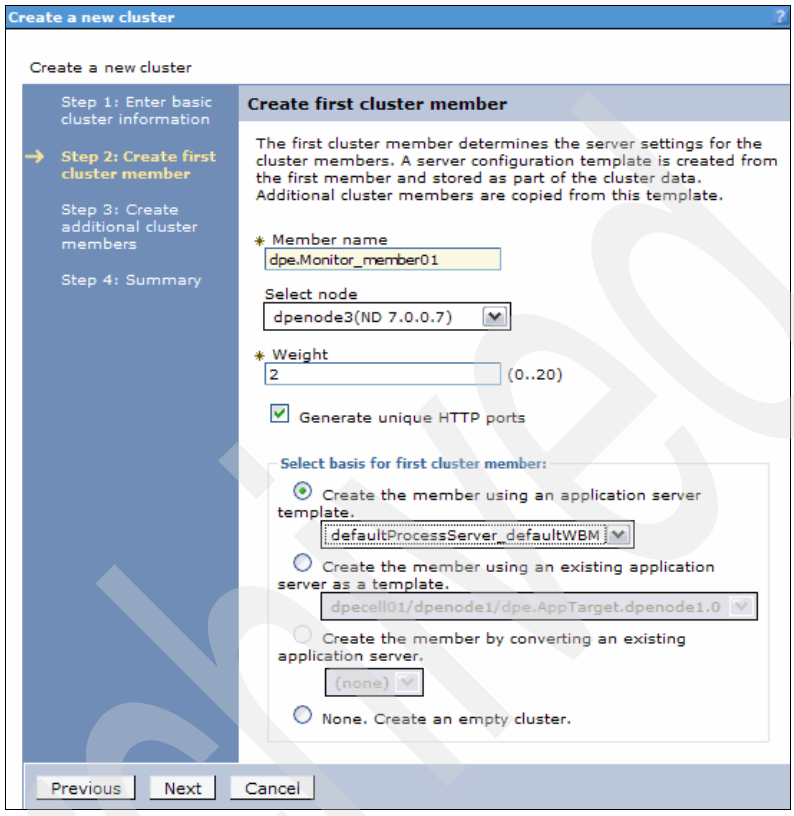


Figure 11-16 Create first cluster member named dpe.Monitor_member01

Archived

3. Create a second cluster member `dpe.Monitor_member02`, as shown in Figure 11-17.

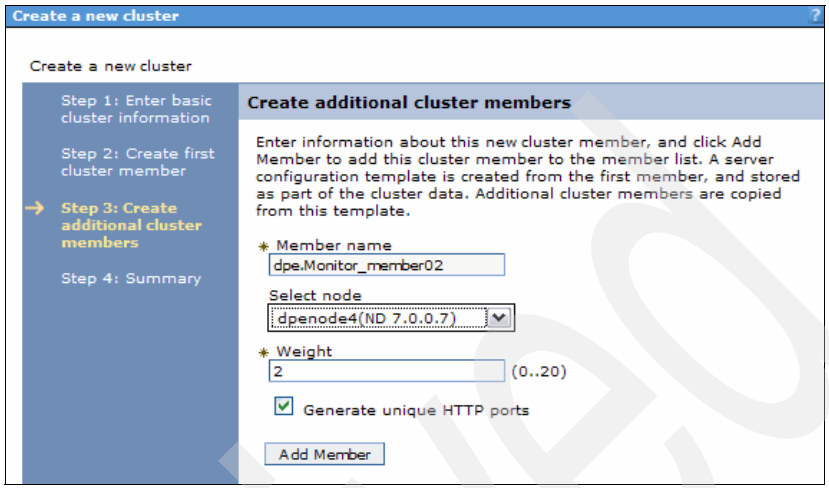


Figure 11-17 Create a second cluster member named dpe.Monitor_member02

4. On the Summary panel, click **Finish** to generate the cluster. Now these clusters are available:
   - `dpe.AppTarget`
   - `dpe.Messaging`
   - `dpe.Support`
   - `dpe.WebApp`
   - `dpe.Monitor`

11.4 Configuring the topology manually using wsadmin

To configure a topology using the command line, you need a topology definition file. The typical way to create one is by defining a deployment environment topology using the Integrated Solutions Console and export it. To do this:

1. Define a deployment environment topology as you did in 11.3.1, “Define and configure a deployment environment” on page 366.
2. Export the topology by opening the Integrated Solutions Console, navigating to **Servers** → **Deployment Environments**, selecting the relevant topology definition, and clicking **Export**.

Archived

Note: You also can generate a new topology definition from scratch using **wsadmin** commands. We do not cover that approach in this book. More information is available at the InfoCenter:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp?topic=/com.ibm.websphere.wps.doc/doc/tcnfg_create_depenv_cmd.html

Once you have a file representing a deployment environment topology (let us assume that it is stored in /tmp/dpe.xml), you can import it and generate the environment using **wsadmin**.

Attention: Before performing the steps below you must increase the SOAP timeout setting in the soap.client.props file located in the deployment manager profile under Properties (in our environment this is /opt/ibm/WebSphere/DynProcSrv7/profiles/dpedmgr01/properties/soap.client.props). Set the timeout value to 0, which means disable the timeout:

```
com.ibm.SOAP.requestTimeout=0
```

1. Start wsadmin in this way:

   ```
   /opt/ibm/WebSphere/DynProcSrv7/profiles/dpedmgr01/bin/wsadmin.sh
   -lang jython -conntype SOAP -host dpenode1 -port 8879 -user dpeadmin
   -password itso4you
   ```

2. Enter the following command on the wsadmin prompt:

   ```
   AdminTask.importDeploymentEnvDef(['-filePath', '/tmp/dpe.xml',
   '-topologyName', 'dpe'])
   ```

   You will receive this response:

   ```
   'Successfully imported topology instance dpe'
   ```

3. Execute the following command on the next wsadmin prompt:

   ```
   AdminTask.generateDeploymentEnv (['-topologyName', 'dpe'])
   ```

   You will receive see this response:

   ```
   'Topology dpehas been successfully configured.'
   ```

4. If this was successful, save your settings with this command:

   ```
   AdminConfig.save()
   ```

5. Exit the wsadmin shell:

   ```
   exit
   ```

Reset the SOAP timeout value in the property file soap.client.props to the value used before. The default value is 180 seconds.

11.5 Create Monitor cluster manually using the command line

To do this:

1. Start wsadmin in this way:

   ```
   /opt/ibm/WebSphere/DynProcSrv7/profiles/dpedmgr01/bin/wsadmin.sh
   -lang jython -conntype SOAP -host dpenode1 -port 8879 -user dpeadmin
   -password itso4you
   ```

2. Call:

   ```
   AdminTask.createCluster('[-clusterConfig [-clusterName dpe.Monitor
   -preferLocal true]]')
   ```

   You receive a response similar to:

   ```
   'dpe.Monitor(cells/dpecell01/clusters/dpe.Monitor|cluster.xml#Server
   Cluster_1266714815487)'
   ```

3. Create the first Monitor member `dpe.Monitor_member01` on node dpenode1:

   ```
   AdminTask.createClusterMember('[-clusterName dpe.Monitor
   -memberConfig [-memberNode dpenode1 -memberName dpe.Monitor_member01
   -memberWeight 2 -genUniquePorts true -replicatorEntry false]
   -firstMember [-templateName defaultProcessServer_defaultWBM
   -nodeGroup DefaultNodeGroup -coreGroup DefaultCoreGroup]]')
   ```

   You receive a response like:

   ```
   'dpe.Monitor_member01(cells/dpecell01/clusters/dpe.Monitor|cluster.x
   ml#ClusterMember_1266715347880)'
   ```

4. Create the second Monitor member `dpe.Monitor_member02` on node dpenode4:

   ```
   AdminTask.createClusterMember('[-clusterName dpe.Monitor
   -memberConfig [-memberNode dpenode4 -memberName dpe.Monitor_member02
   -memberWeight 2 -genUniquePorts true -replicatorEntry false]]')
   ```

   The response will be something like:

   ```
   'dpe.Monitor_member02(cells/dpecell01/clusters/dpe.Monitor|cluster.x
   ml#ClusterMember_1266715387991)'
   ```

5. Save your changes by calling:

   ```
   AdminConfig.save()
   ```

Archived

11.6 Post-creation configuration and verification

This section describes applicable tests to verify that your environment is working correctly.

11.6.1 Verify server startup without errors

Restart your deployment manager and all your node agents on dpenode1, dpenode2, dpenode3, and dpenode4 and check the log files for any errors. The `SystemOut.log` can be found in the `logs` directory of your profile. For example, for the deployment manager on our environment it is located under `/opt/ibm/WebSphere/DynProcSrv7/profiles/dpedmgr01/logs/dmgr`. Restart all your clusters and check the logs for any errors.

11.6.2 Creating event database tables

If you decide to persist your CEI events (see also 11.6.5, “Disabling CEI event persistence” on page 385), you must create and populate the event database tables. Refer to 3.3.7, “The EVENT database” on page 89, for information about how to do this.

11.6.3 Test data sources

Verify that you can access all the defined data sources without any errors. Log in to your Integrated Solutions Console, go to **Resources** → **JDBC** → **Data sources**, and select all data sources. Click **Test connection** to test your data sources. You should receive a success message for all data sources.

Archive

11.6.4 Test message engines are running

Log in to the Integrated Solutions Console, go to **Service integration** → **Buses**, and verify that all your message engines are up and running. You willsee the buses defined as shown in Figure 11-18.

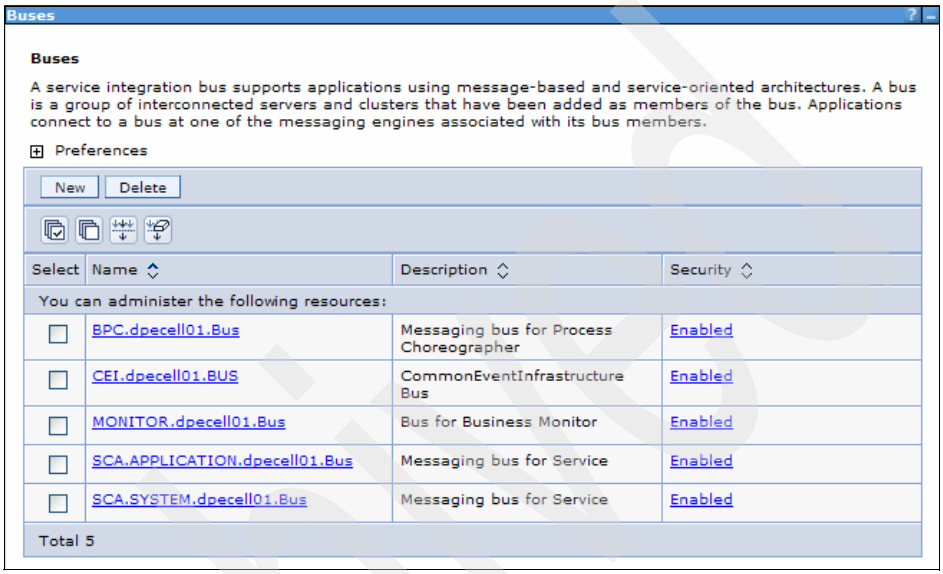


Figure 11-18 Defined buses in WebSphere Dynamic Process Edition

Click each bus and go to **Messaging engines** to verify that the message engine is up and running, as shown in Figure 11-19, for the bus CEI.dpecell01.BUS. You will see a green arrow for each bus defined to support this topology.

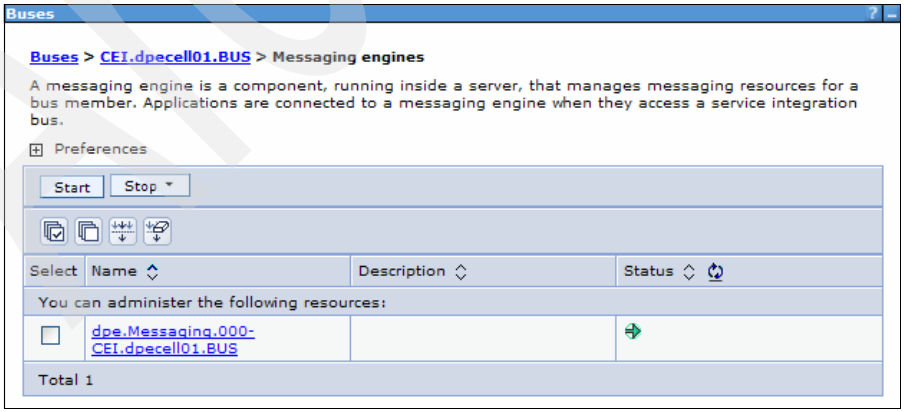


Figure 11-19 Verify that your message engine is up and running

11.6.5 Disabling CEI event persistence

For a production system, disable persistence of CEI events to a database, as this can have significant performance overhead. To do this:

1. Open the Integrated Solutions Console and navigate to **Service Integration → Common Event Infrastructure → Event Service → Event Services**.
2. Click **Default Common Event Infrastructure event server**. Click **Enable event data store** to unselect it. Ensure that it is *not* selected, as shown in Figure 11-20.

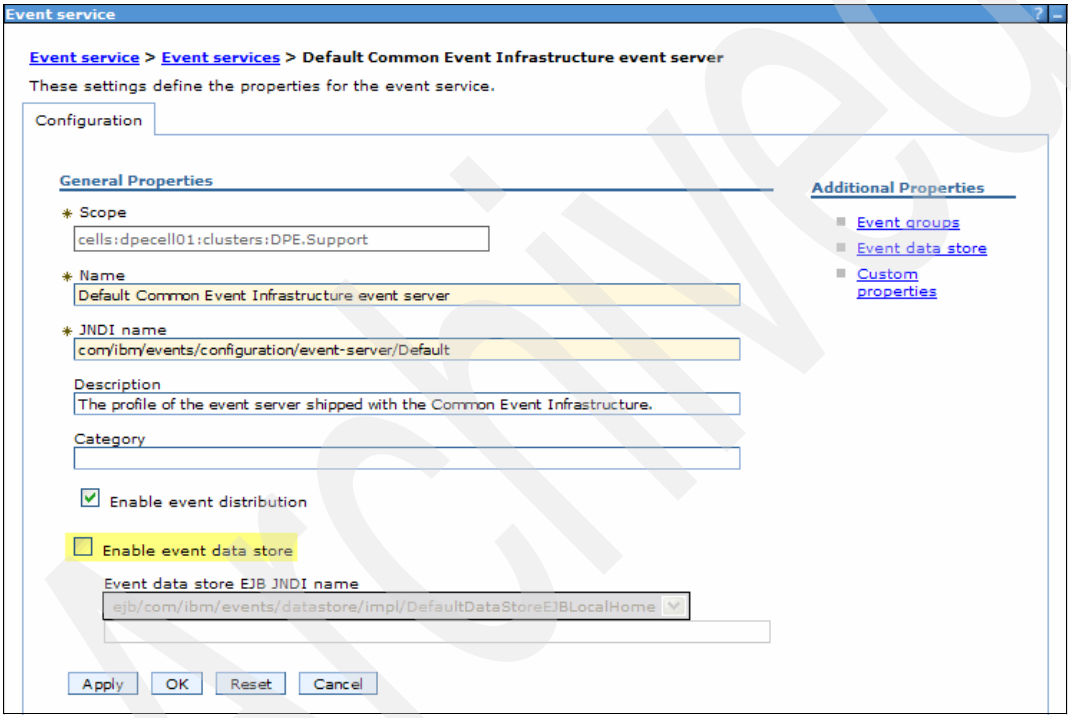


Figure 11-20 Disable event persistence in production environment

3. Click **OK**. Save and synchronize your changes.

Archive

Command-line equivalent

Alternatively, you may use the **wsadmin** commands shown in Example 11-1 with wsadmin to accomplish the same steps. Make sure to replace the cell name with the cell name that you are using.

Example 11-1 wsadmin script to set CEI event persistence

```
eventServerProfile = AdminConfig.list('EventServerProfile',
AdminConfig.getid( '/Cell:dpecell01/')).split("\n")[0]
AdminConfig.modify(eventServerProfile, '[[enableDataStore false]]')
AdminConfig.save()
```

11.6.6 Test the Alphablox admin page

Open a Web browser and go to the Alphablox admin URL:

`http://`*`httpserver`*`/AlphabloxAdmin`

You will see the Alphablox admin page displayed.

Archived

Troubleshooting: Alphablox admin page does not show up

First, check the `SystemOut.log` of your dpe.WebApp cluster for any exceptions. If you see an `Access is denied` message as in Example 11-2, add the role *Admin Security Manager* to your Alphablox user abx, as shown in Figure 11-21. Go to **Users and Groups** → **Administrative user roles** and do the mapping. A restart of your WebApp cluster is required to activate this change.

Example 11-2 Error message in SystemOut.log

```
SRVE0292I: Servlet Message -
[AlphabloxPlatform#AlphabloxServer.war]:.MonitorStartup: 2/17/10
2:10:47 PM [FATAL] Web Application Server Adapter: ADMN0022E: Access is
denied for the getDeploymentDescriptor operation on Application MBean
because of insufficient or empty credentials.
```

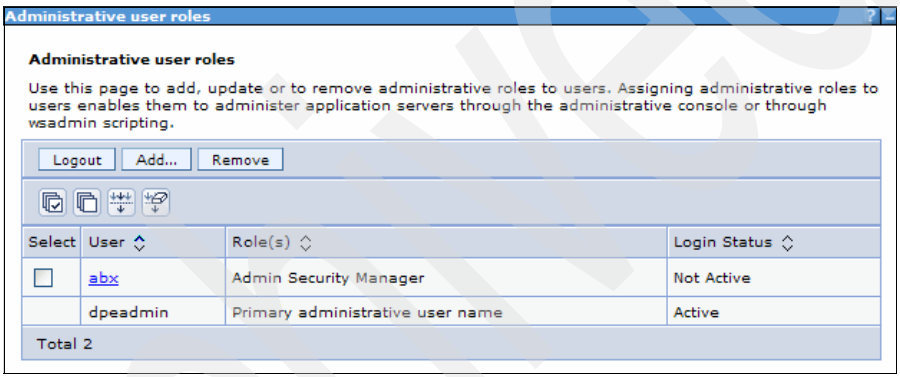


Figure 11-21 Map user abx to role Admin Security Manager

Archive

If the log files does not show any errors and you still do not get the Alphablox admin page, verify that you have mapped the application AlphabloxPlatform to both your WebApp cluster and your Web server, as shown in Figure 11-22. Regenerate and propagate the Web server.

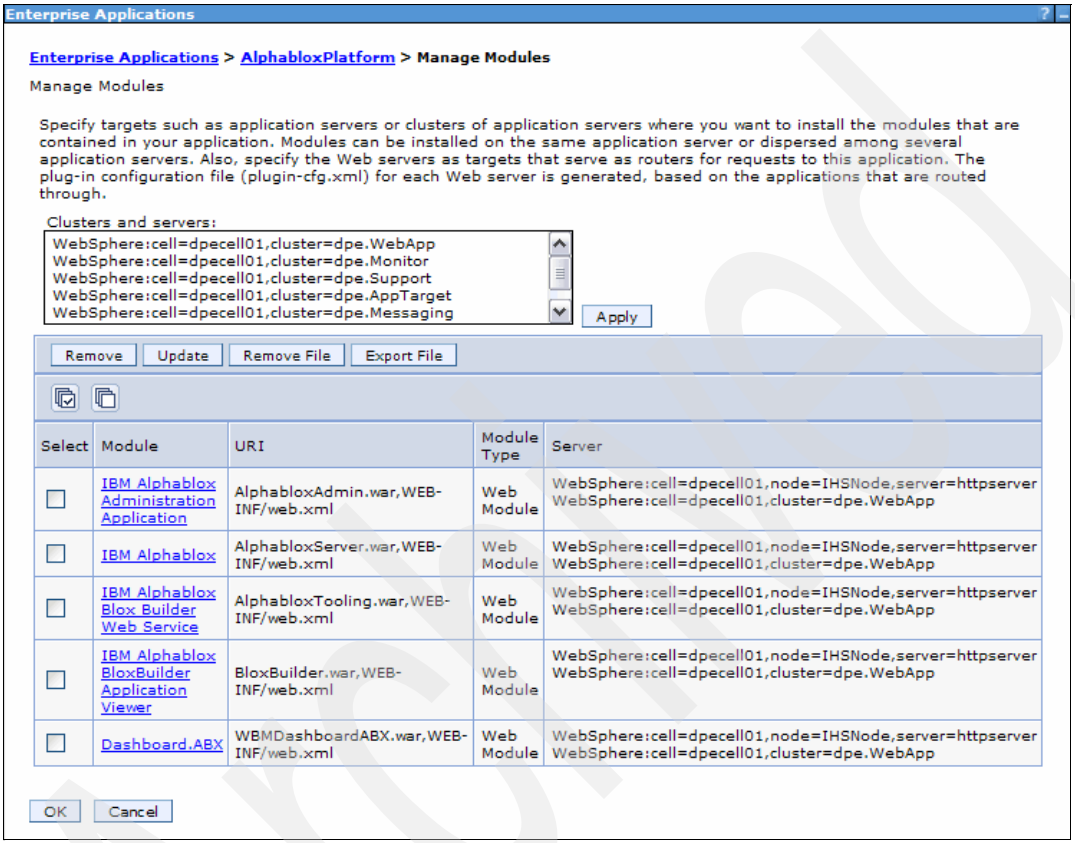


Figure 11-22 Map Alphablox modules to Web server and WebApp cluster

If you still cannot see your Alphablox admin page, verify the following settings:

1. Check your dpe.WebApp cluster and which ports are used. Go to **Clusters** → **WebSphere application server clusters** → **dpe.WebApp** → **Cluster members** → **clusterMember** → **Ports** and write down the ports for the entry `WC_defaulthost` and `WC_defaulthost_secure`. In our environment this is port 9082 for WC_defaulthost and port 9445 for WC_defaulthost_secure.
2. Go to **Environment** → **Virtual hosts** → **default_host** → **Host Aliases** and search for entries, in our case, of port 9082 and 9445.
3. If no entries can be found, add an asterisk (*) entry for port 9082 and another asterisk entry for 9445.

4. Regenerate and propagate the Web server plug-in.
5. Restart your dpe.WebApp cluster.

11.6.7 Test Business Space

Open the URL `http://`*`httpserver`*`/BusinessSpace` in a Web browser. You will see the Business Space welcome page.

Troubleshooting: Business Space page does not show up

Verify the following settings:

1. Check whether the following applications are mapped to the Web server and the WebApp cluster:
   - BSpaceEAR_dpe.WebApp
   - BSpaceWebformsEnabler_dpe.WebApp
   - BusinessSpaceHelpEAR_dpe.WebApp
   - mm.was_dpe.WebApp
   - REST Services Gateway
2. Make sure that you have entries in your virtual hosts alias section for your WebApp cluster, as explained in 11.6.6, “Test the Alphablox admin page” on page 386.
3. Additionally, you might verify the REST service endpoint provider settings as described 5.5.3, “Configuring REST service endpoints” on page 179.

11.6.8 Test the Business Process Choreographer (BPC)

Open the URL `http://`*`httpserver`*`/bpc` in your browser. You should see the BPC Explorer view.

If you do not see the BPC Explorer page, verify that the application BPCExplorer_dpe.WebApp is mapped to your Web server and your WebApp cluster.

To verify that the BPC Explorer and the human task manager are working correctly, run the test case described in 4.5.8, “Verification: human task manager” on page 142.

11.6.9 Test the Business Rule Manager

Open the URL `http://`*`httpserver`*`/br` in your browser. You will see the Business Rules Manager page displayed.

Archived

If you do not see the page, verify that you mapped the application BusinessRulesManager_dpe.WebApp to your Web server and your WebApp cluster.

11.6.10 Test WebSphere Business Services Fabric

Open the URL `http://<httpserver/fabric/` in your browser. You will see the Business Services Fabric Welcome page displayed.

If you do not see the page, verify that you mapped the following applications to your Web server and your WebApp cluster:

- Fabric_Tools_dpe.AppTarget
- Fabric_Catalog_dpe.AppTarget

A functional test of the WebSphere Business Services Fabric application can be found in 10.5.5, “Installing and testing a WebSphere Business Services Fabric sample application” on page 352.

Archived

11.6.11 Test WebSphere Business Monitor

To emit an event from any BPEL process, we must enable the Common Event Infrastructure logging in the Business Flow Manager. Go to **Servers** → **Clusters** → **WebSphere application server clusters** and select the cluster in which the BPEL applications are running. In our environment, this is the dpe.AppTarget cluster. Open the **Business Process Choreographer** section under Business Integration and click **Business Flow Manager**. Activate the **Enable Common Event Infrastructure logging** check box under State Observers, as shown in Figure 11-23.

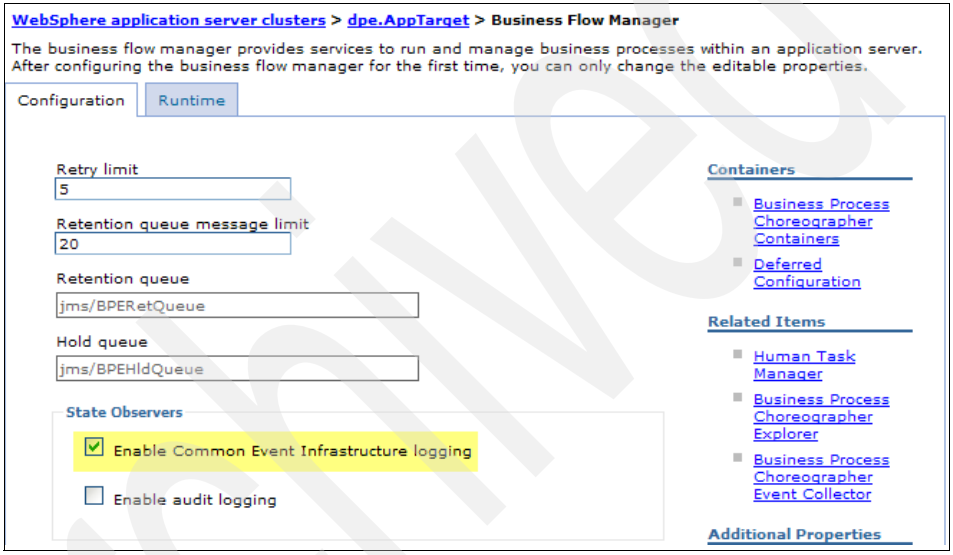


Figure 11-23 Enable Common Event Infrastructure logging for Business Flow Manager

Archived

We must redo the above step for the Human Task manager. Go back to your dpe.AppTarget and select the link **Human Task Manager** in the Business Process Choreographer section under Business Integration. Select the **Enable Common Event Infrastructure logging** check box, as shown in Figure 11-24.

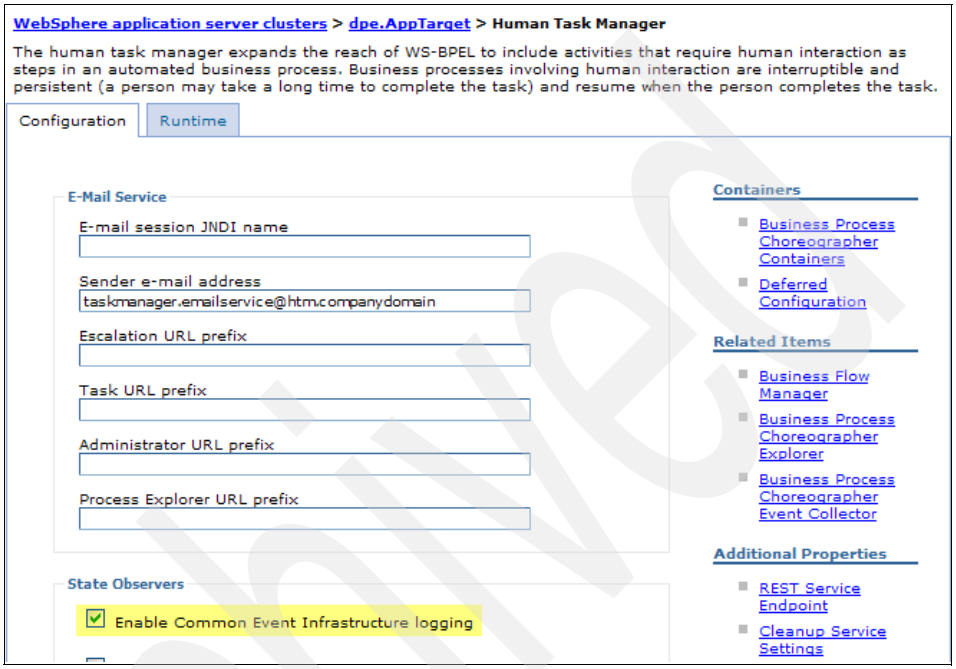


Figure 11-24 Enable Common Event Infrastructure logging for the human task manager

To apply your changes, restart your AppTarget cluster.

11.6.12 Verify the topology using a sample application

As a last step deploy the HelloWorld sample application provided with the additional material of this book to validate all components. Appendix B, “Verification using the sample application” on page 423, describes the steps to install and test the HelloWorld application.

Archived

12

WebSphere Business Compass

This chapter describes how to configure a WebSphere Business Compass topology based on the Single Cluster topology pattern.

This chapter discusses the following topics:

- The topology used in this chapter and when to choose this topology
- The software prerequisites and installation parameters used to build this topology
- Detailed step-by-step instructions about how to configure the topology using a graphical user interface (GUI) and how to install in silent mode
- How to validate the topology by deploying a sample application

Archived

© Copyright IBM Corp. 2010. All rights reserved.

12.1 Topology summary

This section outlines the steps to install WebSphere Business Compass using a Single Cluster topology pattern. The installation steps were performed on three SUSE Linux Enterprise Server 10 64-bit servers. Figure 12-1 depicts the topology and shows the distribution of all main WebSphere Business Compass components.

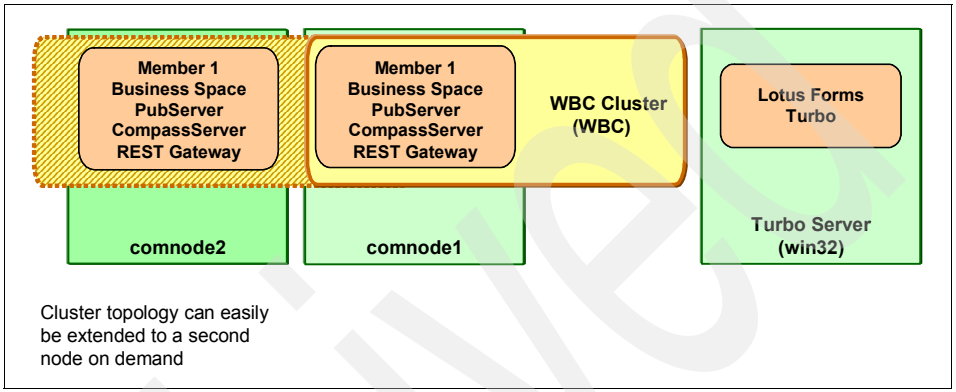


Figure 12-1 WebSphere Business Compass topology

The WebSphere Business Compass installation described in this chapter is based on the Single Cluster topology pattern. The Single Cluster topology was chosen in favor of a standalone server installation for the following reasons:

- Topology offers the possibility to do vertical or horizontal scaling if the load on the WebSphere Business Compass server increases by adding new cluster members.
- Topology offers the option to set up a highly available topology by adding new nodes and cluster members to the WebSphere Business Compass cell.
- Installation effort is similar to a standalone server installation with a remote database.

Note: Lotus Forms Turbo is an optional component of WebSphere Business Compass. It is installed on a dedicated Windows server to avoid interference with WebSphere Business Compass components and to facilitate the installation.

Archived

12.2 Prerequisites for creating this topology

Before installing WebSphere Business Compass, ensure that you have the required prerequisites. This section describes the software and system requirements and the installation parameter required to install the product.

12.2.1 Prepare your operating system

Verify that your hardware and operating system conforms to the requirements. Detailed instructions can be found in the InfoCenter. For hardware and software requirements consult the following section in the InfoCenter:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/topic/com.ibm.btools.help.monitor.inst.doc/plan/prep_os_lin.html

Note: Also check whether your operating system kernel parameters must be adapted. In our example on SUSE Linux Enterprise Server 10 we increased the number of allowed open files (ulimit -n 8192).

Information about your operating-system-specific instructions can be found in the following section of the InfoCenter:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/topic/com.ibm.btools.help.modeler.collab.publish.inst.doc/plan/prep_os_lin.html

The WebSphere Business Compass installation described in this chapter uses fully qualified host names. Table 12-1 shows the host names used in this topology. These host names must be defined on all servers that are part of this topology. In our case we added hostnames to the `/etc/hosts` file on each server.

Table 12-1 Host names

Host name	Description
itsodb2.itso.ral.ibm.com	This server hosts the DB2 instance.
itsoldap.itso.ral.ibm.com	This server hosts the LDAP Tivoli Directory Server.
comnode1.itso.ral.ibm.com	WebSphere Business Compass node.
comweb.itso.ral.ibm.com	IBM HTTP server for browser access to Business Space
turbo.itso.ral.ibm.com	Host name of Lotus Form Turbo server

12.2.2 Install database server

The topology described in this chapter uses DB2 Enterprise Edition 9.5 FixPack 5. Refer to “Software versions used in this book” on page 56.

12.2.3 Install LDAP server

Tivoli Directory Server 6.2 is used as an LDAP server in this topology as per “Software versions used in this book” on page 56.

12.2.4 Define users in LDAP user registry

This topology uses a federated user repository with users defined in an Lightweight Directory Access Protocol (LDAP) server. Users used in this topology and instructions about how to set up WebSphere Application Server security can be found in 3.4.2, “Configuring LDAP as the user account registry” on page 99

12.2.5 Install WebSphere Business Compass

There are two options for installing the WebSphere Business Monitor binaries:

- The graphical installation method, using the IBM Installation Manager
- The silent installation using IBM Installation Manager response files

Both methods are described in 3.2, “Software installation” on page 57.

The response files `run_wbc` and `wbc_response.xml` are used to install the WebSphere Business Compass binaries and can be found in Appendix A, “Additional material” on page 421.

Note: In order to reuse an existing DB2 database instance choose the **WebSphere Business Compass only** installation option during binary installation using installation manager. The installation option **Install WebSphere Business Compass and Database management system** installs a new IBM DB2 instance on the WebSphere Business Compass server.

Archived

12.2.6 Create and configure deployment manager

Create a deployment manager profile, set up the user registry integration, add IBM HTTP Server to the cell, and modify configuration settings, as per 3.4, “Creating a deployment manager profile” on page 91.

Table 12-2 shows the key properties and values used for the WebSphere Business Compass topology setup.

Table 12-2 Topology key properties

Topology component name	Value
WebSphere Business Compass cell name	wbccell
WebSphere Business Compass node name	comnode1
WebSphere Business Compass deployment manager node name	wbcdmgrnode
Integrated Solutions Console	9034
Dmgr bootstrap port	9809
Dmgr SOAP Connector admin port	8879

The response file `wbc_dmgr_create.resp` used to set up the deployment manager can be found in Appendix A, “Additional material” on page 421.

12.2.7 Create and federate custom node

Create and federate one custom node (comnode1) as per 3.5, “Creating custom profiles” on page 107.

The response file `wbc_node_create.resp` used to set up the custom node can be found in Appendix A, “Additional material” on page 421.

12.2.8 Install IBM HTTP Server

In this topology, an IBM HTTP server is used in front of the WebSphere Business Compass server. Refer to 3.6.1, “Installing IBM HTTP Server” on page 111, for instructions about how to set up and configure the HTTP server.

Archived

12.2.9 Create databases

This product does not use the DDT tool to create the database. We run the sql scripts, which are created as part of the deployment manager profile creation. Once you have created your deployment manager and installed Business Space, you can create the database. Refer to 12.3.3, “Create WebSphere Business Compass and Business Space databases” on page 402 for detailed instructions.

12.3 Installing WebSphere Business Compass using Integrated Solutions Console

This section describes how to install WebSphere Business Compass graphically using the Integrated Solutions Console.

12.3.1 Create a cluster

To do this:

1. Create a cluster to install WebSphere Business Compass components. In the Integrated Solutions Console navigate to **Servers** → **Clusters** → **WebSphere application server clusters** and click **New.** On the next page the cluster creation wizard is displayed.
2. On the first page of the wizard enter `WBCCluster` as the cluster name and click **Next**, as in Figure 12-2.

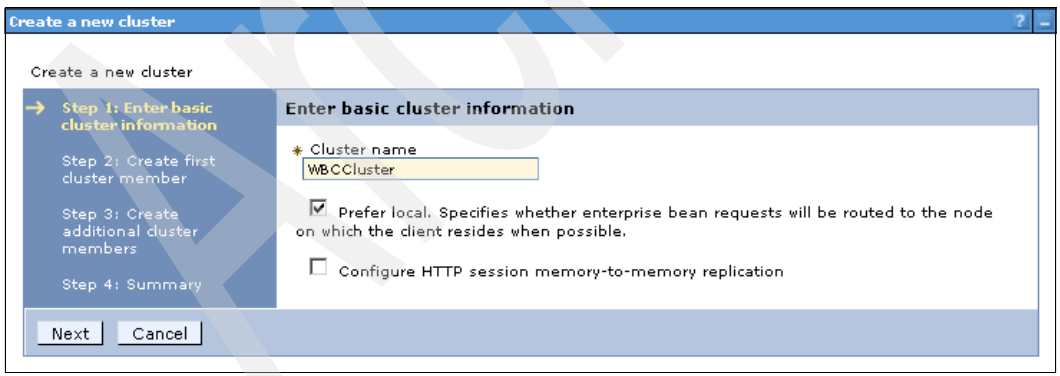


Figure 12-2 Enter cluster name

Archived

3. In the next step enter the cluster member information. Enter `WBCCluster.comnode1.0` as the member name, select `wbcnode1` on from the Select node list box, and leave the Application server template list box as default, as in Figure 12-3.

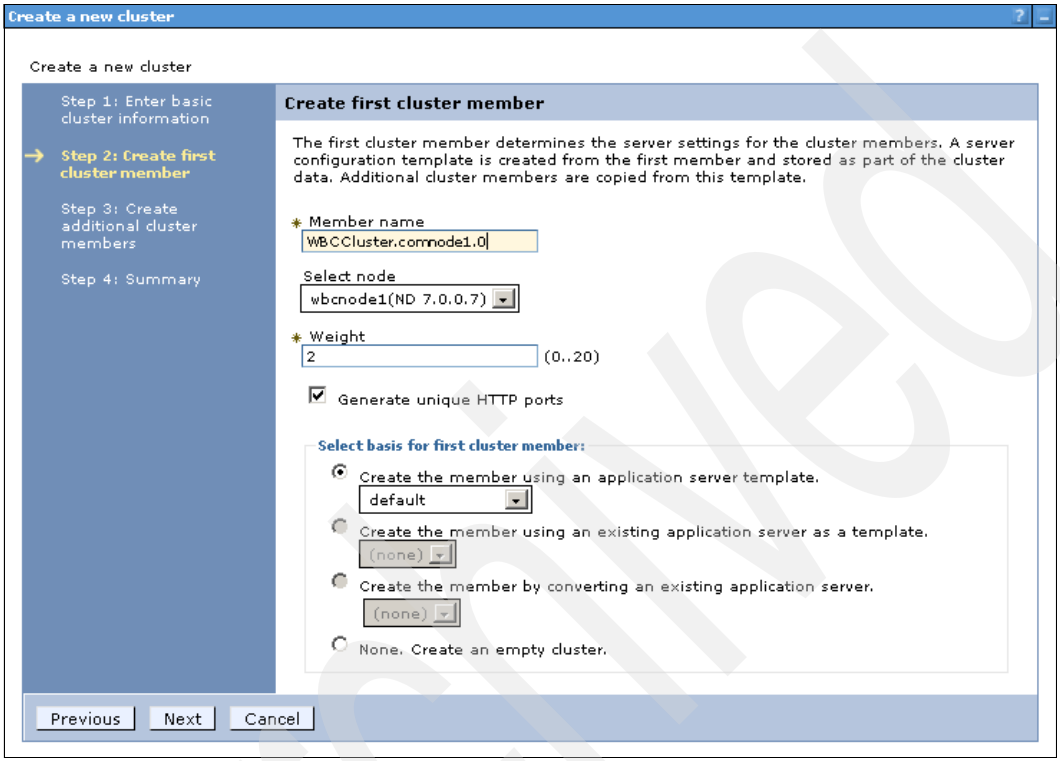


Figure 12-3 Enter cluster member parameter

4. On the Create additional cluster member page click **Next**.

Archived

5. Review the summary page and click **Finish**, as in Figure 12-4.

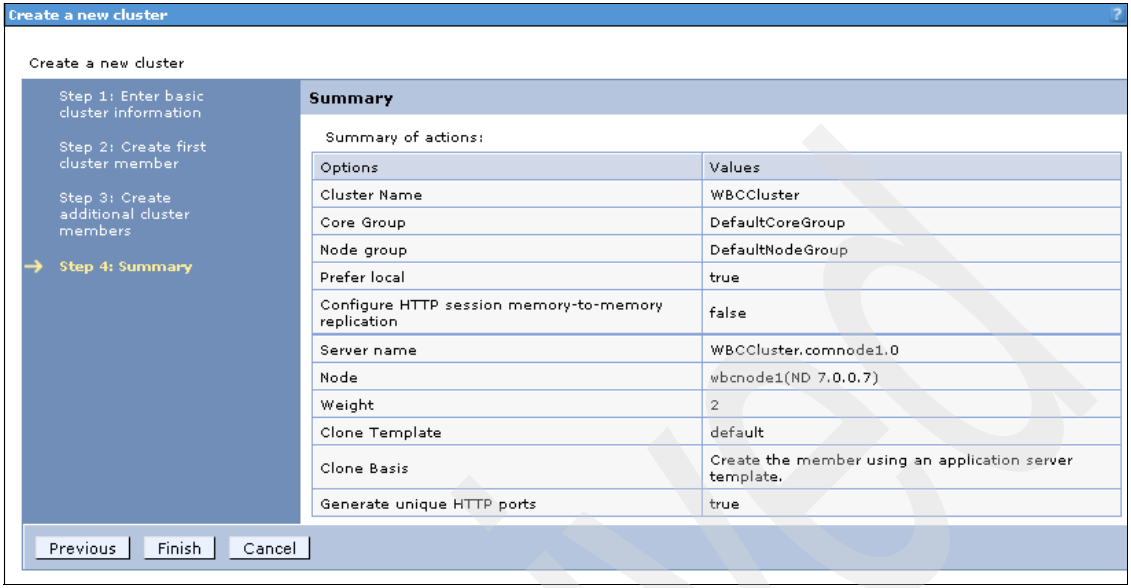


Figure 12-4 Cluster creation summary panel

A new cluster has been created.

Archived

12.3.2 Install Business Space

To do this:

1. Navigate to **Servers** → **Clusters** → **WebSphere application server clusters** and click the newly created cluster **WBCCluster**.
2. Click **Business Space Configuration**, as in Figure 12-5.

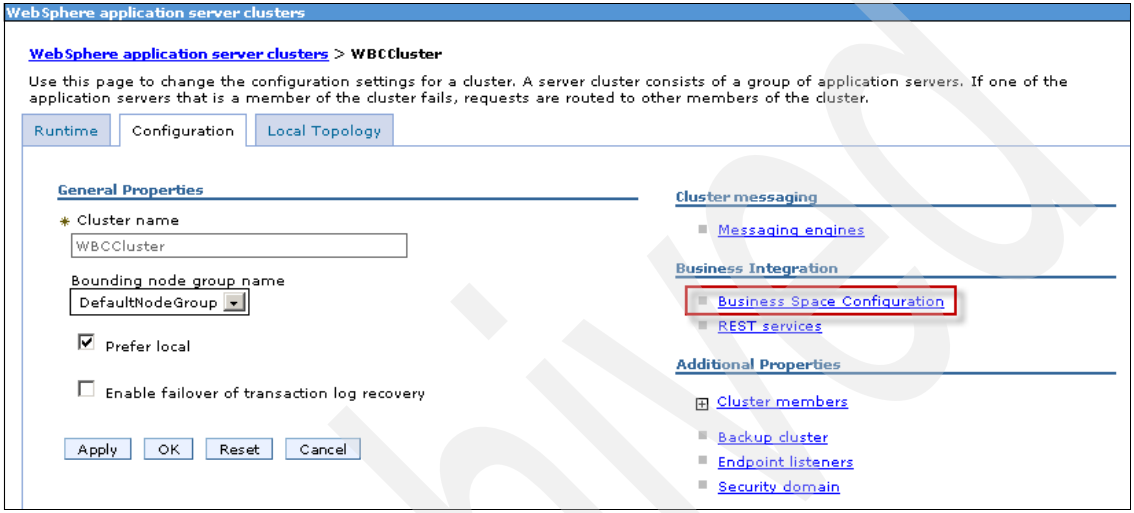


Figure 12-5 Business Space configuration

3. On the next page click **Install Business Space service**. Enter `IBMBUSSP` as the database schema name and select **Publishing Server Datasource**. Click **OK** to install Business Space.

Archived

4. The Integrated Solutions Console displays a summary page, as in Figure 12-6. The installation was successful. You can ignore the error message related to the Representational State Transfer (REST) service endpoints configuration. Continue and click **Save** to finish the installation.

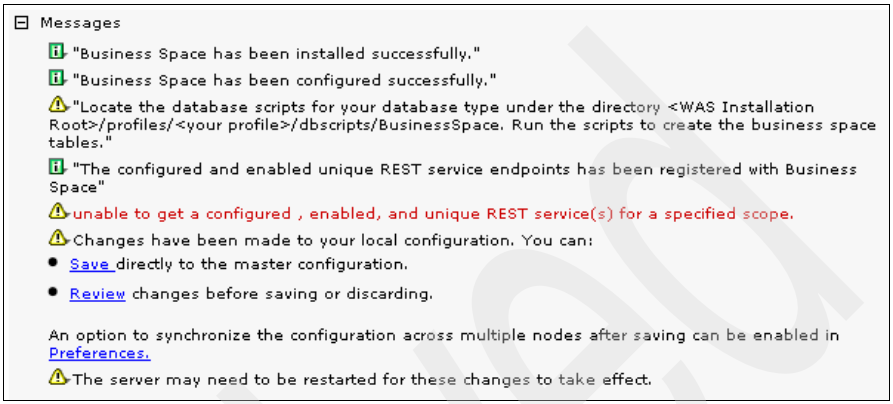


Figure 12-6 Business Space installation result panel

12.3.3 Create WebSphere Business Compass and Business Space databases

To do this:

1. Copy the database creation script `createWPBSDB.sql` from the deployment manger profiles directory `/opt/ibm/WebSphere/BusComp/profiles/wbcdmgr/dbscripts.wpbs` to the `/software/compass` directory on the database server itsodb2.
2. Find and replace $DBPWD$ in the `createWPBSDB.sql` file with your database instance administrator password in single quotation marks (for example, `'itso4you'` in our case).
3. Do the same for the Business Space scripts (no substitution needed this time). Copy the entire `WBCDB` directory into the `/opt/ibm/WebSphere/BusComp/profiles/wbcdmgr/dbscripts/BusinessSpace/WBCCluster/DB2/WBCDB to /software/compass` directory on the database server itsodb2.
4. Open a remote shell to itsodb2 and log in as DB2 WebSphere Business Compass instance user cominst.
5. Execute the following **db2** command to create the WebSphere Business Compass database:

```
db2 -tf /software/createWPBSDB.sql
```

6. Change to the `/software/WBCDB` directory and execute `configBusinessSpaceDB.sh`.
7. Enter a DB2 user name and password as requested.
8. Verify the outcome of the **db2** commands.

12.3.4 Install WebSphere Business Compass applications

To do this:

1. In the Integrated Solutions Console navigate to **Applications** → **New Application**.
2. Click **New Enterprise Application**.
3. Select **Remote file system** and browse to `/opt/ibm/WebSphere/BusComp/installableApps.wpbs on your deployment manager node`.
4. Select **PublishingServer.ear** and click **OK**. Click **Next**.

Archived

5. Use the Fast Path option and deploy the application using default parameters, as summarized in Figure 12-7.

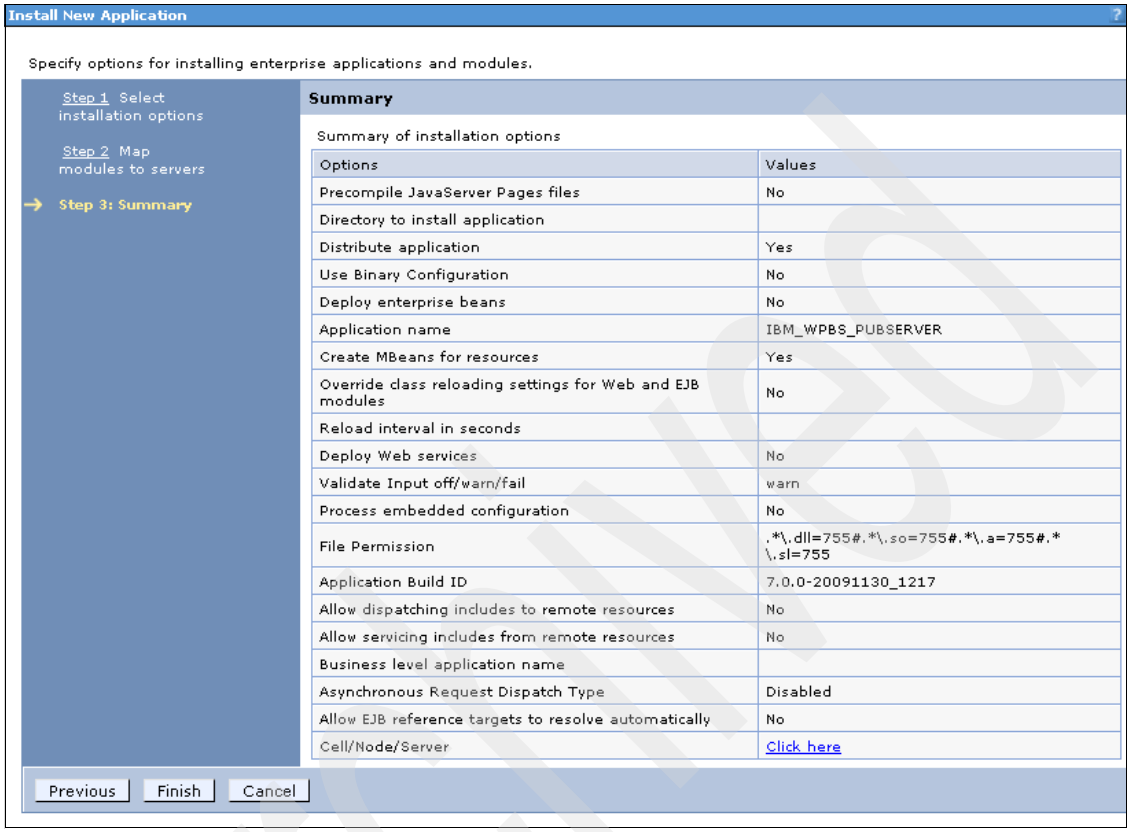


Figure 12-7 Install PublishinServer.ear

6. Repeat steps 1 to 4 for BusinessLeaderApp.ear.

At this stage WebSphere Business Compass is installed.

12.4 Post-creation configuration and verification

The following steps describe how to complete and verify the configuration.

Archived

12.4.1 Start WebSphere Business Compass

Restart the Deployment Manager and all nodes using the stop and start commands shown in Example 12-1.

Example 12-1 Commands to stop deployment manager and nodes

```
/opt/ibm/WebSphere/BusComp/profiles/wbcnode1/bin/stopServer.sh
WBCCluster.comnode1.0 -username wbcadmin -password itso4you
/opt/ibm/WebSphere/BusComp/profiles/wbcdmgr/bin/stopManager.sh
-username wbcadmin -password itso4you
/opt/ibm/WebSphere/BusComp/profiles/wbcnode1/bin/stopNode.sh -username
wbcadmin -password itso4you
/opt/ibm/WebSphere/BusComp/profiles/wbcdmgr/bin/startManager.sh
/opt/ibm/WebSphere/BusComp/profiles/wbcnode1/bin/startNode.sh
/opt/ibm/WebSphere/BusComp/profiles/wbcnode1/bin/startServer.sh
WBCCluster.comnode1.0
```

When the cluster is started it is required to check the `SystemOut.log` of the deployment manager, the node agent, and the cluster member to verify that there are no errors. The `SystemOut.log` file is in the logs folder of the wbcdmgr and wbcnode1 profile. For example, in our environment the `SystemOut.log` file for the WebSphere Business Compass cluster is in the `/opt/ibm/WebSphere/BusComp/profiles/wbcnode1/logs/WBCCluster.comnode1.0/` folder.

Example 12-2 gives an example of the messages shown during startup of the AppTarget cluster showing that all the applications artifacts were started successfully.

Example 12-2 Startup of AppTarget cluster

```
WSVR0221I: Application started: BusinessLeaderApp
WSVR0221I: Application started: mm.was_WBCCluster
WSVR0221I: Application started: BSpaceEAR_WBCCluster
WSVR0221I: Application started: BSpaceWebformsEnabler_WBCCluster
WSVR0221I: Application started: widgets_pubserver_WBCCluster
WSVR0221I: Application started: IBM_WPBS_PUBSERVER
WSVR0221I: Application started: widgets_busleader_WBCCluster
```

In the Integrated Solutions Console navigate to **Servers** → **Server Types - WebSphere Application Servers** to confirm that the server is started, as shown in Figure 12-8.

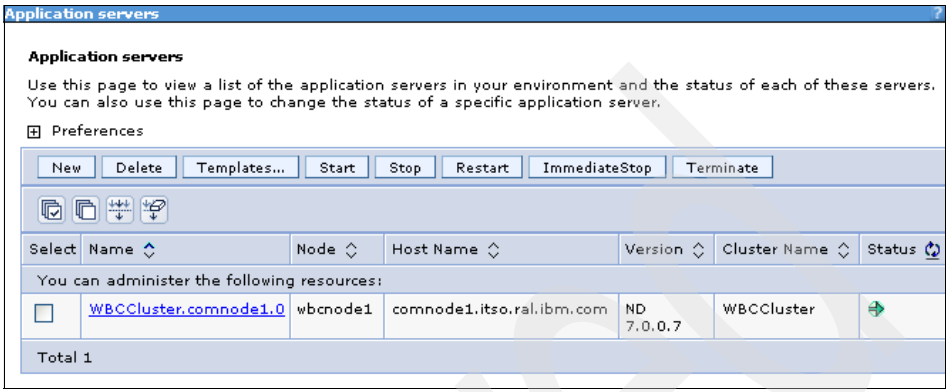


Figure 12-8 Cluster view with cluster member started

12.4.2 Verify database connectivity

Before you start the environment, check the database connectivity to ensure that the deployment environment has been generated correctly. Follow these steps to verify the database connectivity:

1. Log in to the administrative console and navigate to **Resources** → **JDBC** → **JDBC Providers** → **Data Sources**.
2. Make sure that **All scopes** is selected.
3. Select all of the data sources and click **Test connection** to make sure that they have connectivity. You will see a message that the connection was successful for each.

Command-line equivalent

You can use a Jython script like the one shown in Example 12-3 to check database connectivity using wsadmin instead. You must change the cell name to the cell name that you are using.

Example 12-3 Checking database connectivity using wsadmin

```
cellName = "wbccell"

dataSources = AdminConfig.list('DataSource', AdminConfig.getid(
'/Cell:' + cellName + '/')).split("\n")

for dataSource in dataSources:
```

```
if not dataSource.startswith('DefaultEJB'):
   print dataSource
   AdminControl.testConnection(dataSource)
```

12.4.3 Install Lotus Forms Turbo

Installation has been done on a dedicated Microsoft Windows XP workstation.

Install binaries

To do this:

1. Execute the downloaded file `LotusFormsTurbo-3.5.1-Fixpack1-Win32.exe` and extract content to a temporary directory of your choice (for example, to `\software`).
2. Navigate to the temporary directory and execute `LFTurbo_351_Win32.exe`.
3. Step to the installation wizard and accept the default values. In summary, choose the **Standard - Installs Lotus Forms Turbo database** installation option on the Select Database page and specify as `lftadmin` as the user name and `itso4you` as the password.

Configure single sign-on with WebSphere Business Compass

We must establish a single sign-on domain between the underlying WebSphere Application Server of Lotus Forms Turbo and WebSphere Business Compass:

1. We must change the WebSphere user registry. To avoid any interference with Windows services, stop and remove the WebSphere Application Server windows service by executing the following commands in the `C:\Program Files\IBM\Lotus Forms\Turbo\3.5\WebSphere\bin` dicrectory:

   ```
   WASService.exe -stop server1
   WASService.exe -remove server1
   and
   WASService.exe -stop IBMWebformServerTranslator35
   WASService.exe -remove IBMWebformServerTranslator35
   ```

2. Open the `C:\Program Files\IBM\Lotus Forms\Turbo\3.5\repos\config\default.properties` file. Navigate to the access.secure.wim.user and access.secure.wim.pass properties and write down the user name and the auto-generated password. Navigate to the access.supportAccountControl property and change it to `false`. We use a LDAP directory as a user registry, so we must disable Lotus Forms Turbo advance user management features.

Archived

3. Start the Lotus Forms Turbo server. Navigate to the `C:\Program Files\IBM\Lotus Forms\Turbo\3.5\WebSphere\profiles\AppSrv01\bin` directory and execute:

   ```
   startServer server1
   startServer TranslatorServer
   ```

4. Log in to Lotus Forms Turbo Integrated Solutions Console to configure LDAP security. The default URL is `https://localhost:9043/admin`. Follow the steps in 3.4.2, “Configuring LDAP as the user account registry” on page 99, using `wasadmin` as the primary administrative user name and `itso4you` as the password.

5. Restart Lotus Forms Turbo by executing the following commands in the `C:\Program Files\IBM\Lotus Forms\Turbo\3.5\WebSphere\profiles\AppSrv01\bin` directory:

   `stopServer.bat server1` (Use the user and password that you wrote down in step 2 on page 407 when the login page displays.)
   `stopServer.bat TranslatorServer` (same user/password as before)
   `startServer.bat server1`
   `startServer.bat TranslatorServer`

   Log in to Lotus Forms Turbo Integrated Solutions Console using the LDAP user `wasadmin` and password `itso4you`. If you are able to log in, you have successfully configured LDAP security on Lotus Forms Turbo.

   Security is enabled on the WebSphere Business Compass server and on the Lotus Forms Turbo server. WebSphere Business Compass allows business users to design Lotus Turbo forms inside Business Space. To avoid a second login when accessing Lotus Forms Turbo, a single sign-on (SSO) domain must be set up on both servers. Perform the next steps to set up SSO.

Archive

6. Configure SSO on WebSphere Business Compass server comnode1:
   a. Log in to WebSphere Business Compass Integrated Solutions Console using LDAP user `wbcadmin` and password `itso4you`.
   b. Navigate to **Security** → **Global Security** in the WebSphere Business Compass Integrated Solutions Console and click **LTPA in section Authentication,** as in Figure 12-9.

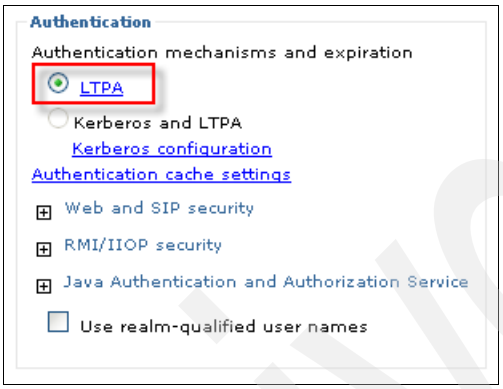


Figure 12-9 Export LTPA keys from WebSphere Business Compass

   c. In the Cross-cell single sign-on section enter the password `itso4you` in the Password and Confirm password fields and enter `/tmp/ltpa.key` in the Fully qualified key file name field. Click **Export keys** to export the key file.
   d. Copy the exported key file to the Lotus Forms Turbo server's `C:\temp` directory.
   e. Still in the WebSphere Business Compass Integrated Solutions Console, navigate to **Security** → **SSL certificate and key management** → **Key stores and certificates** → **NodeDefaultTrustStore** → **Signer certificates**.

Archived

f. Click **Retrieve from port** and enter the Lotus Forms Turbo server connection parameter, as in Figure 12-10.

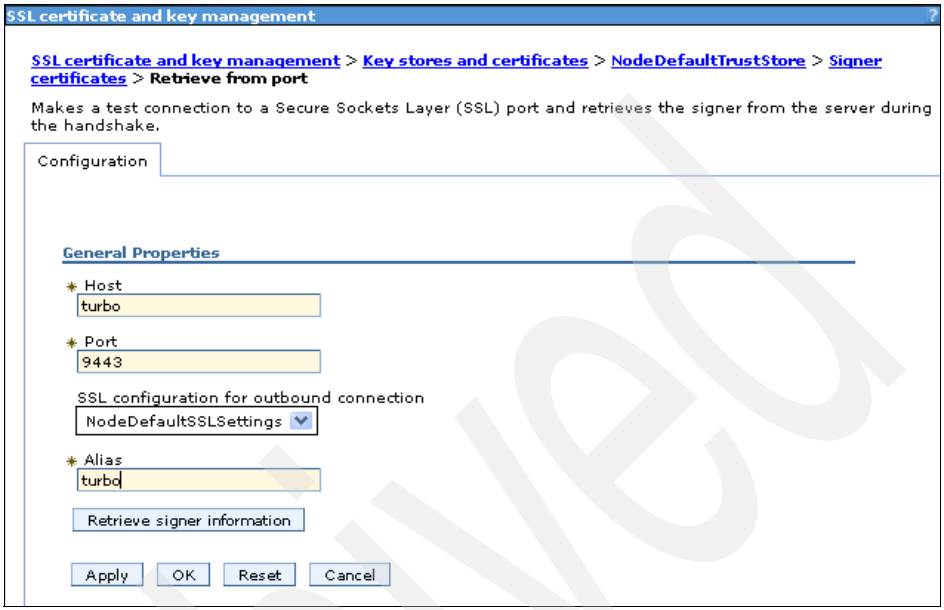


Figure 12-10 Import Lotus Forms Turbo server certificate

g. Click **Retrieve Signer information**. Review the SSL information and click **OK** to import the certificate into the trust store.

Archived

h. Navigate to **Security** → **Global Security** → **Authentication** → **Web and SIP Security** → **Single sign-on (SSO)**, specify the SSO domain `itso.ral.ibm.com`, and activate the option **Interoperability Mode**, as shown in Figure 12-11.

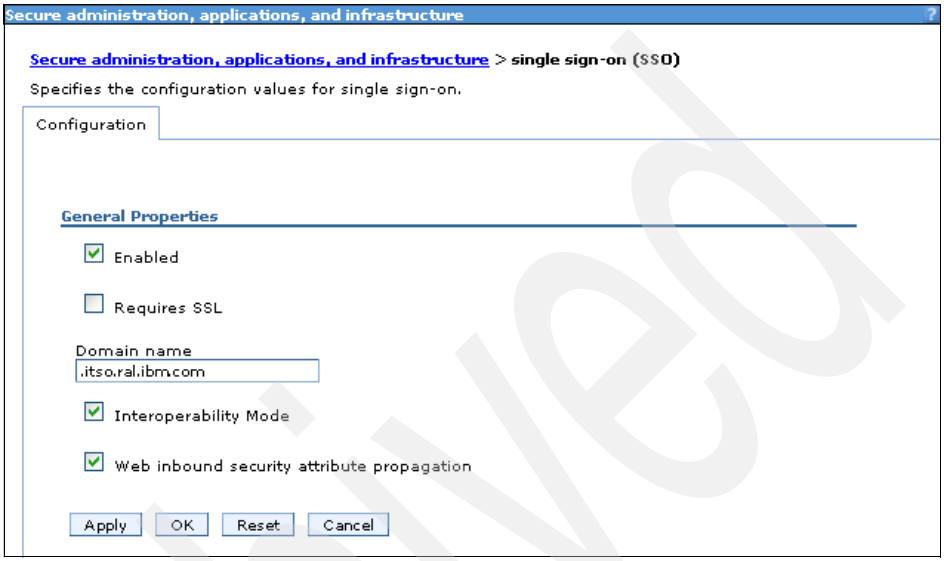


Figure 12-11 Enter SSO domain for WebSphere Business Compass server

i. Click **OK** and save your configuration.

7. Configure SSO on Lotus Forms Turbo server `turbo`:

a. Log in to the Lotus Forms Turbo Integrated Solutions Console and navigate to **Security** → **Secure administration, applications, and infrastructure**. Click **Authentication mechanisms and expiration** under the Authentication section, as in Figure 12-12.

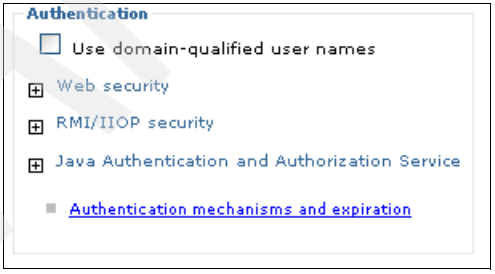


Figure 12-12 LTPA key import

Archived

b. In the Cross-cell single sign-on section use the same password that you used in the key export (`itso4you`) for the Password and Confirm password fields and use the previously exported key file `C:\temp\ltpa.key` in the Fully qualified key file name field. Click **Import keys** to import the keys. You get a message saying:

```
The keys were successfully imported from the file
C:\temp\ltpa.key.
```

c. Navigate to **Security → SSL certificate and key management → Key stores and certificates → NodeDefaultTrustStore → Signer certificates**. Click **Retrieve from port** and enter the WebSphere Business Compass connection parameter, as shown in Figure 12-13. Use the host name or IP address of WebSphere Business Compass server.

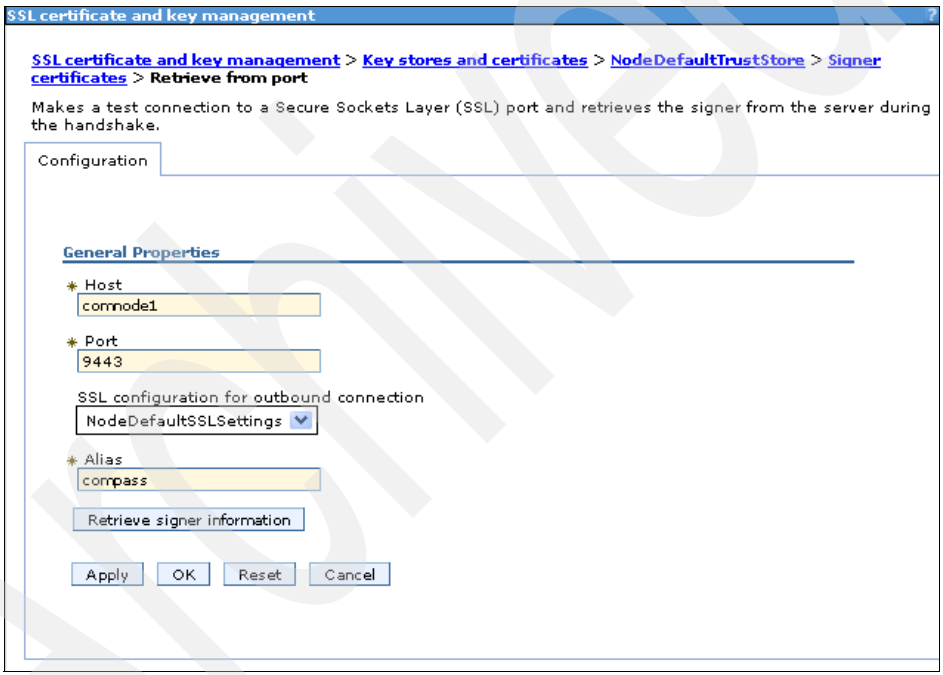


Figure 12-13 Retrieve SSL certificate from WebSphere Business Compass node

d. Click **Retrieve Signer information**. Review the SSL information and click **OK** to import the certificate into the trust store.

e. Save your configuration.

f. Still in the Integrated Solutions Console navigate to **Security → Secure administration, applications, and infrastructure → Authentication → Web Security → single sign-on (SSO)** and specify the same SSO

Archived

domain `.itso.ral.ibm.com` that you used on WebSphere Business Compass, as shown in Figure 12-11 on page 411.

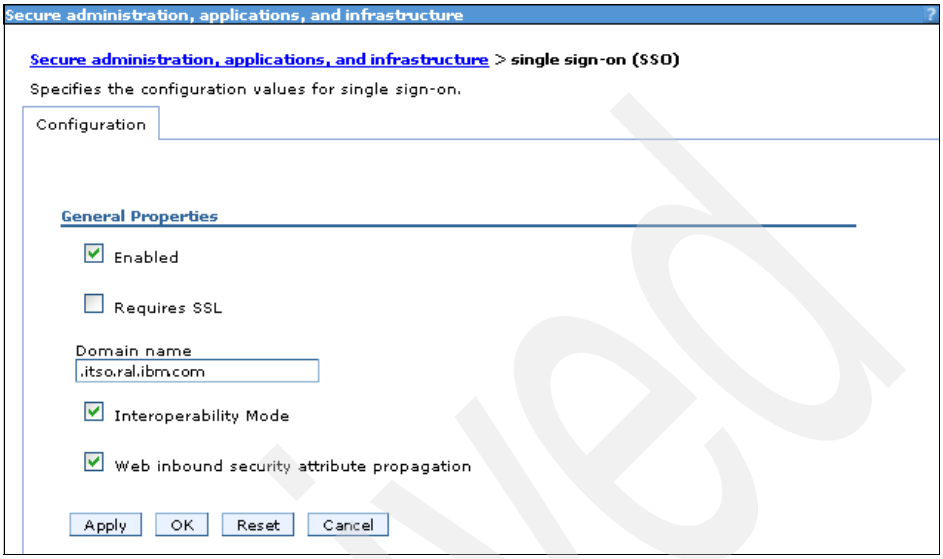


Figure 12-14 Enter SSO domain for Lotus Forms Turbo Server

8. Click **OK** and save your configuration.
9. Stop the Lotus Forms Turbo Server:
   - `cd C:\Program Files\IBM\Lotus Forms\Turbo\3.5\WebSphere\profiles\AppSrv01\bin`
   - `stopServer.bat server1 -user wasadmin -password itso4you`
   - `stopServer.bat TranslatorServer -user wasadmin -password itso4you`
10. Stop the WebSphere Business Compass cell servers (deployment manager, nodes, and cluster). For details see Example 12-1 on page 405.
11. Update Lotus Forms Turbo endpoints in WebSphere Business Compass. Log in as `cominst` on the database server itsodb2 and execute the following commands:

```
db2 connect to WBCDB
db2 “update bl_endpoints set url='http://turbo.itso.ral.ibm.com:9080
' where name='FORM_SERVER_URL'“
db2 “update bl_endpoints set url='http://turbo.itso.ral.ibm.com:8085
' where name='FORM_TRANSLATOR_URL' “
```

12. Start the WebSphere Business Compass cell and Lotus Forms Turbo server.

Archived

12.4.4 Verify installation using Business Space

To do this:

1. Point you browser to:

   `https://comnode1.itso.ral.ibm.com:9443/BusinessSpace`

2. Log in to Business Space using user `wbcadmin` and password `itso4you`.
3. Click **Go to Spaces** (Figure 12-15).

Figure 12-15 Go to spaces

4. In the dialog Go to Spaces click **Business Design Home**. The Business Design Space opens.
5. On the right upper corner click **Import the business design sample** (Figure 12-16).

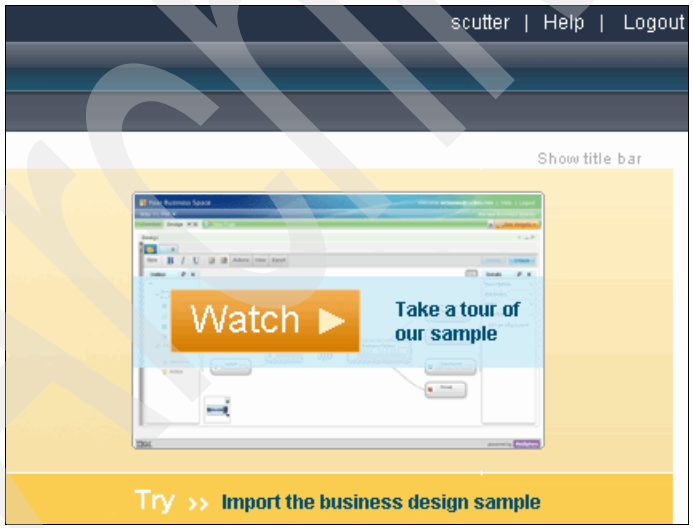


Figure 12-16 Import samples

Archived

6. Click **Expedited Reservation and Check in Process** (Figure 12-17).

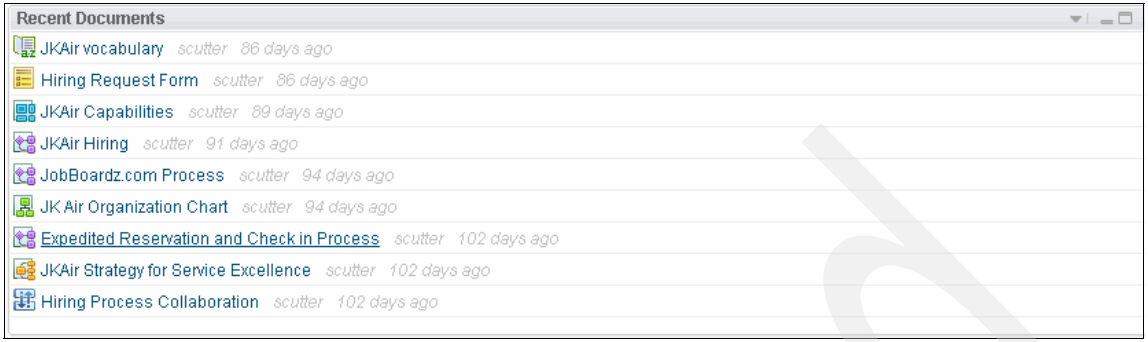


Figure 12-17 Open sample document

7. You might need to install Adobe® Flash Player when indicated.
8. Staying in the Design tab, you again click link **Expedited Reservation and Check in Process**. A new tab is opened and the process is shown, as in Figure 12-18.

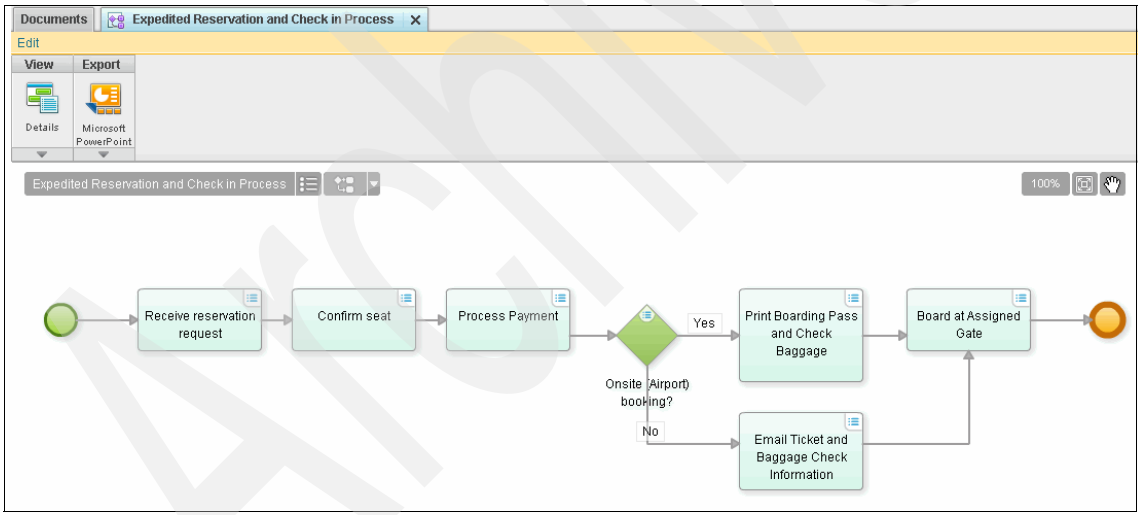


Figure 12-18 Extended reservation and check in process

9. Click the **Documents** tab in the upper left corner.

Archived

10. To verify the Lotus Forms Turbo integration, create a new form by clicking **Form Document**, as shown in Figure 12-19.

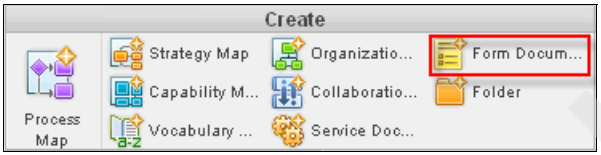


Figure 12-19 Create a new forms document

11. Enter the name of the form `Test` in the Name field and click **OK**.
12. In the next page choose **Design a new form** and click **OK**, as in Figure 12-20. If this option is not available, the Lotus Forms Turbo integration failed. Verify the Lotus Forms Turbo and WebSphere Business Compass log files, especially for SSL-related exceptions or handshake and LTPA token exceptions.

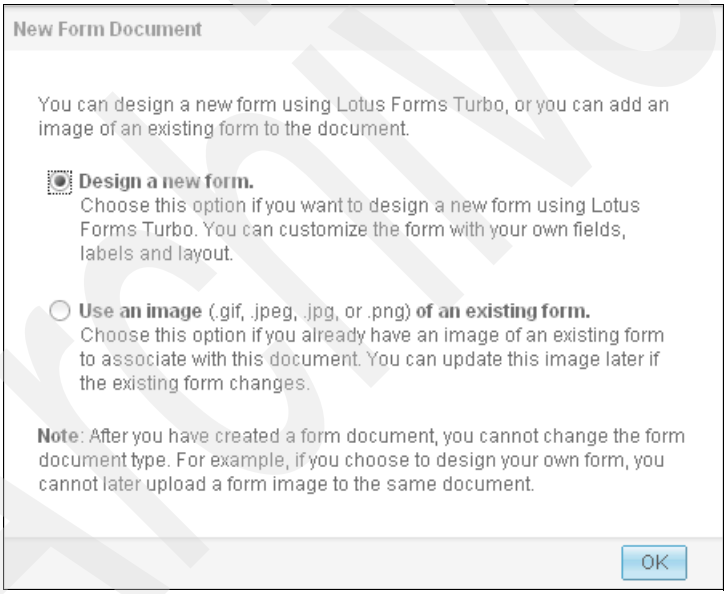


Figure 12-20 Create a new Lotus Forms Turbo document

Archived

13. You will see the Lotus Forms Turbo design page. Drag and drop form elements from the palette on the canvas to verify that integration works, as shown in Figure 12-21.

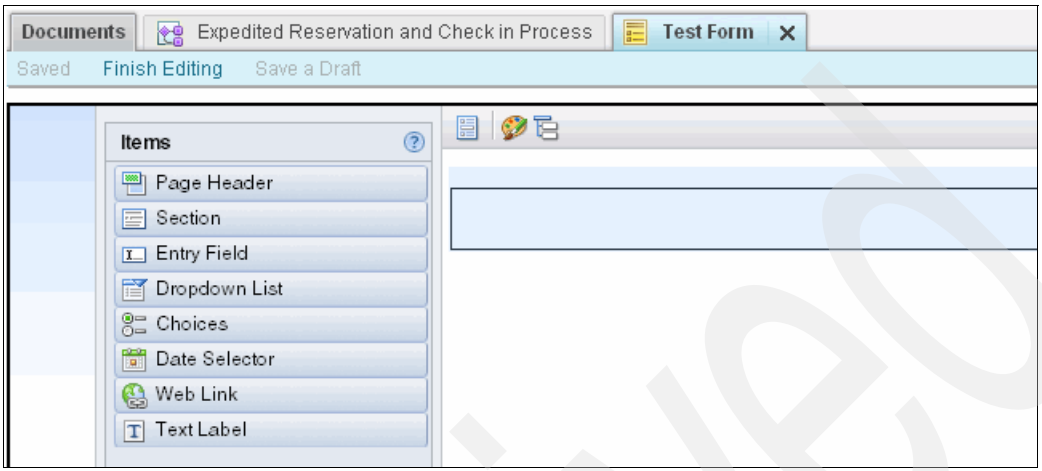


Figure 12-21 Lotus Forms Turbo design space

14. Click **Finish Editing** and close the test form.

Archived

Archived

Part 3

Appendixes

Archived

© Copyright IBM Corp. 2010. All rights reserved.

Archived

A

Additional material

This book refers to additional material that can be downloaded from the Internet as described below.

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/SG247854

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the IBM Redbooks form number, SG247854.

How to use the Web material

Create a subdirectory (folder) on your workstation,and unzip the contents of the Web material zip file into this folder.

© Copyright IBM Corp. 2010. All rights reserved.

Archived

B

Verification using the sample application

This chapter describes how to install and run a sample WebSphere Process Server module and a sample WebSphere Business Monitor model. This chapter discusses the following topics:

- Installing a WebSphere Process Server module
- Installing a WebSphere Business Monitor module
- Importing and emitting sample events using the recorded event management
- Executing the WebSphere Process Server process using Business Process Choreographer Explorer
- Configuring a monitor model to consume events from a remote common event infrastructure.
- Creating a test Business Space

© Copyright IBM Corp. 2010. All rights reserved.

Verification using the sample application

Follow this procedure to install and test the HelloWorld application and monitor model. This application helps you to verify that:

- The process runs successfully in WebSphere Process Server
- Events are being emitted from WebSphere Process Server and consumed by WebSphere Business Monitor
- Results are being displayed on the Business Space dashboard.

The sample application can be found in Appendix A, “Additional material” on page 421.

Install WebSphere Process Server module

This section describes how to install the WebSphere Process Server module HelloWorldModuleApp. To deploy the module perform these steps:

1. In the Integrated Solutions Console navigate to **Applications** → **Monitor Models** and click **Install**.
2. Navigate to the location where your application source is, as shown in Figure B-1.

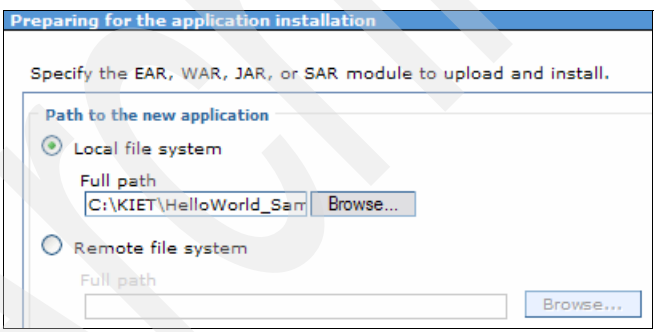


Figure B-1 Installing sample HelloWorld application for end-to-end verification of environment built

3. Select the **detailed** check box.
4. Click **Next** for the Installation options window.

Archived

5. In step 2, check the module and map it to run on the AppTarget cluster, as shown in Figure B-2.

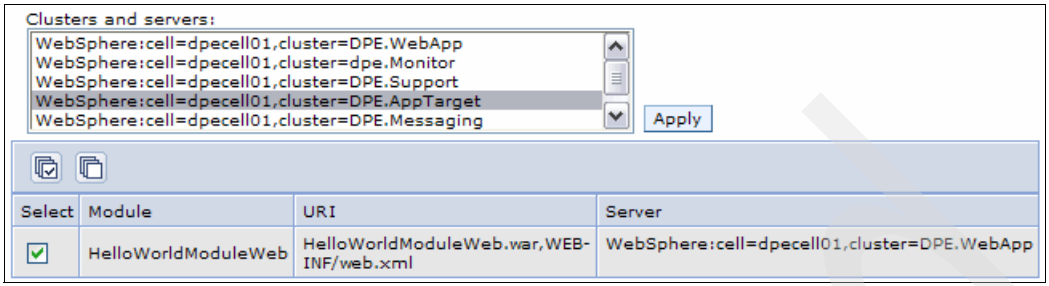


Figure B-2 Mapping HelloWorld BPEL module

6. Click **Apply**.
7. Keep clicking **Next** for steps 3 through 7 until you reach the Summary screen.
8. Click **Finish**. This application does not need to be mapped to the Web server, so the warning is OK.
9. Click **Save** when the installation is successful.
10. Navigate to **Applications** → **Application Types** → **WebSphere enterprise applications**. In the list click the check box to the left of the HelloWorldModuleApp application and click **Start**. Verify that the application successfully starts.

Install the monitor model

This section describes how to install the WebSphere Business Monitor model HelloWorldModuleMMApplication. To deploy the monitor module perform the steps:

1. In the WebSphere Business Monitor Integrated Solutions Console navigate to **Applications** → **Monitor Models** and click **Install**.
2. Click **Browse** and locate the `HelloWorldMMApplication.ear` file on your hard disk. Then click **Next**.
3. On the next page click **Fast Path**, then click **Next**.
4. On the Step 1: Select installation options page accept the default settings and click **Next**.
5. On page Step 2: Map modules to servers click both check boxes next to the **HelloWorldMMModelLogic** and **HelloWorldMMModerator** modules in the module table, then click the **AppTarget** cluster and click **Apply**. Then click **Next**.

Archive

6. On page Step 3: Configure security for monitor models click **Next**. A warning appears on top of page with the message:

   `CWMAX4103W: There are no members assigned to the selected resource group.`

   Ignore the message. The assignment of the user is done after monitor model installation.

7. On page Step 4: Summary click **Finish**.
8. Check the installation log page and wait for the message:

   `HelloWorldMMApplication successfully installed.`

9. Click **Save** link to save the changes and synchronize with nodes.
10. The application starts automatically. On the Monitor Models page wait until Model HelloWorldMM status icon turns green.
11. Also verify the log files of the AppTarget on both cluster members. Verify that the message shown in Example B-1 appears. This message indicates that the monitor model has successfully been started and is waiting for incoming events. The second message only appears on one of the cluster members. In order to guarantee events sequencing only oneconsumer can be active on a cluster at a given time.

Example: B-1 Successful start of monitor model

```
0000003c CompositionUn A   WSVR0191I: Composition unit
WebSphere:cuname=HelloWorldMMApplication in BLA
WebSphere:blaname=HelloWorldM
MApplication started.

00000021 ConsumerDaemo I
com.ibm.wbimonitor.mm.HelloWorldMM.20080121000000.moderator.ConsumerDae
monHandlerImpl startDaemon() CWMRT30
05I: The Monitor Model "HelloWorldMM 20080121000000" is starting
consumption on this server or cluster member in SERIAL_MT mode with
reordering=true from NORMA
L_QUEUE_BYPASS.
```

Emit sample events with recorded event management

Next import events and play back events using the recorded events management feature of WebSphere Business Monitor:

1. In the Integrated Solutions Console navigate to **Applications** → **Monitor Services** → **Recorded Event Management** → **Events Management**.
2. Click **Delete all** to remove existing events and then click **Import Events**.
3. On the Import Events page click **Browse** and locate the `Recorded_Events_HelloWorld.xml` file provided in Appendix A, “Additional material” on page 421, and click **Import Events**.
4. A message on the top of the page shows that five events have been imported into the Events Management page.
5. Play back the events by navigating to **Applications** → **Monitor Services** → **Recorded Event Management** → **Play Back Events**.
6. Select **HelloWorldMM** as the target monitor model and click **Play Back All**.

A message on the top of the page shows that five events have been sent to the HelloWorldMM monitor model.

Create a test Business Space

The application has been successfully deployed to the WebSphere Business Monitor server. As a next step use Business Space to validate that the sample events have been processed by the HelloWorld monitor model:

1. Open the Business Space application by pointing the Web browser :

   ```
   https://<web server host>/BusinessSpace
   ```

2. Log in as the Business Space administrator.
3. On the Business Space Welcome page click **Actions** → **Created Space**. The Create Space dialog opens.

Archived

4. Enter `HelloWorld` as the space name, and in selection box Create a new Space using template choose **Blank** and click **Save**, as shown in Figure B-3.

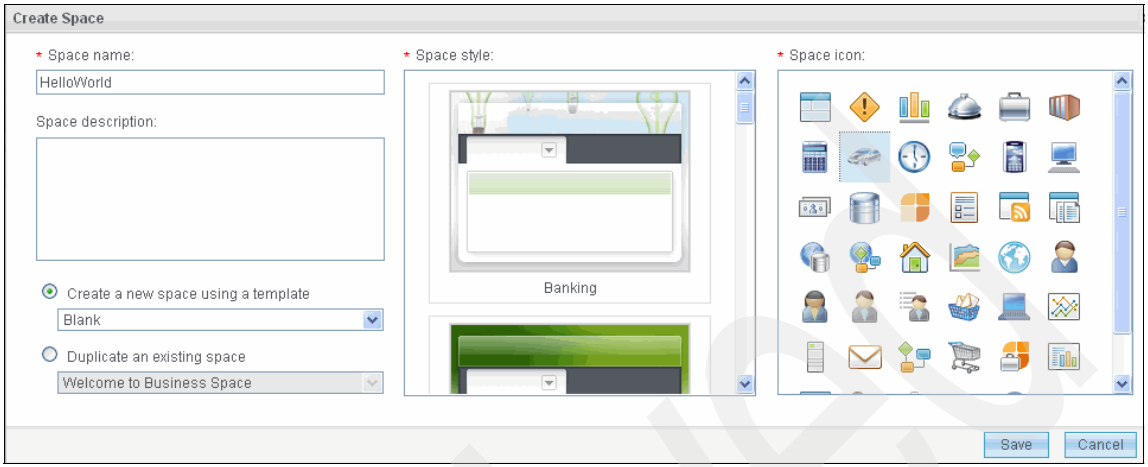


Figure B-3 Create new space

5. The new HelloWorld space opens. Click **Edit Page** in the upper left corner, as shown in Figure B-4.

Figure B-4 Toggle to edit mode

6. The widget palette opens. Add the Dimension widget, the KPI widget, and the instance widget in this order by clicking the plus sign (+) to the right of the widget, as shown in Figure B-5.

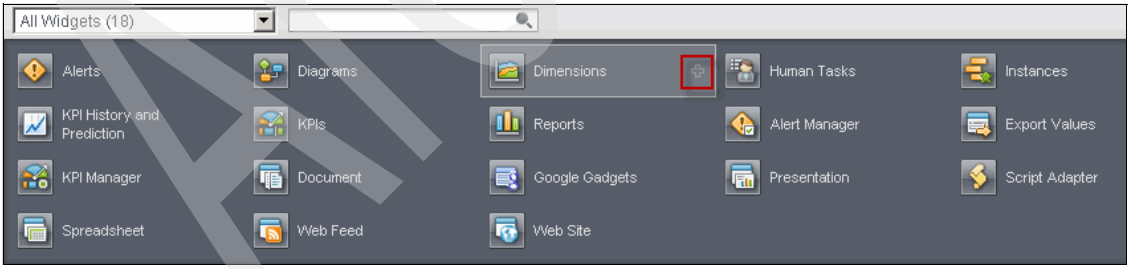


Figure B-5 Add widgets to page

7. Configure the instance widget by clicking the small down-arrow in the upper right corner of the instance widget and **Edit Settings** in the pop-up menu.

Archive

8. On the Instance Edit dialog click the **HelloWorldProcess** context and click **Set as Default**. Select the **Creation Time** column in the available columns and click **>** to move it to the Selected columns box. Do the same for **HelloWorldProcess Name**, **HelloWorldProcess State**, and **COMPLETED** and click **OK**, as shown in Figure B-6.

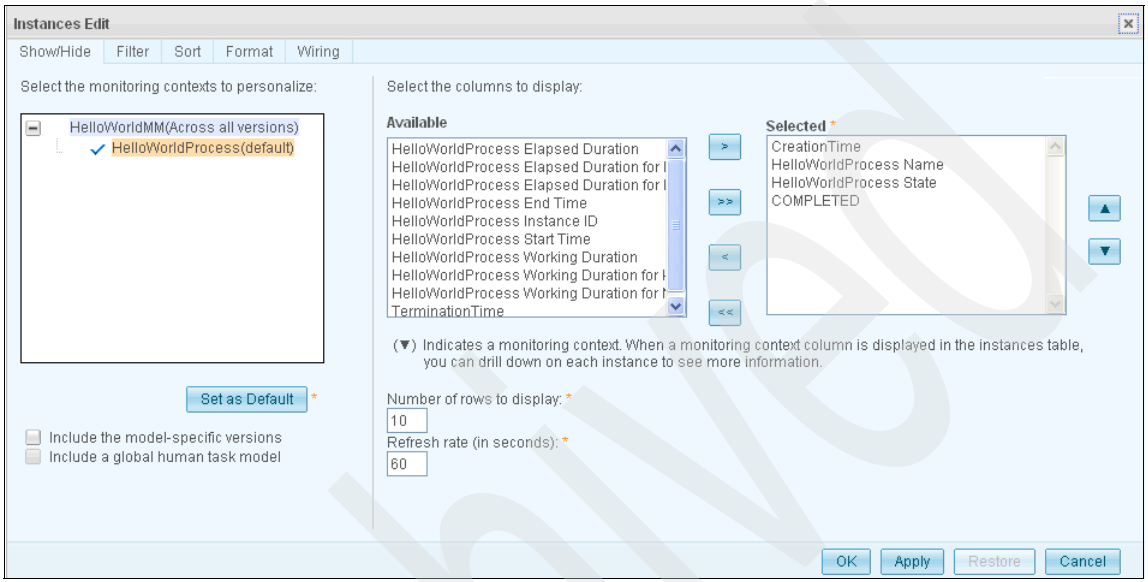


Figure B-6 Configure instance widget

Archived

9. Open the KPI widget Edit dialog and select both KPIs and click **OK**, as shown in Figure B-7.

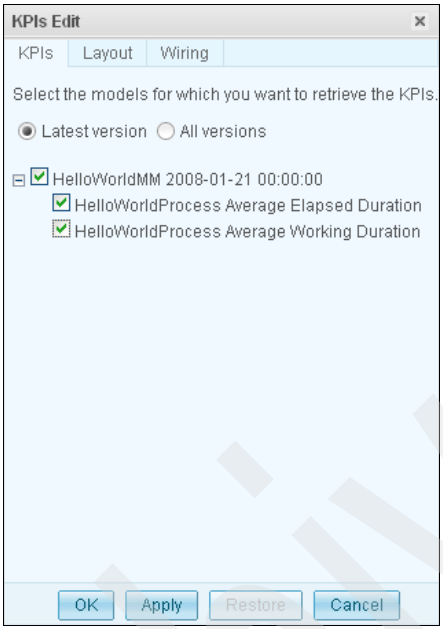


Figure B-7 Configure KPI widget

Archived

10. Open the Dimension widget Edit dialog and move the Creation Time dimension to the Column dimension list and add the Measure dimension to the Page dimension list using the arrow button (>), as shown in Figure B-8, and click **OK**.

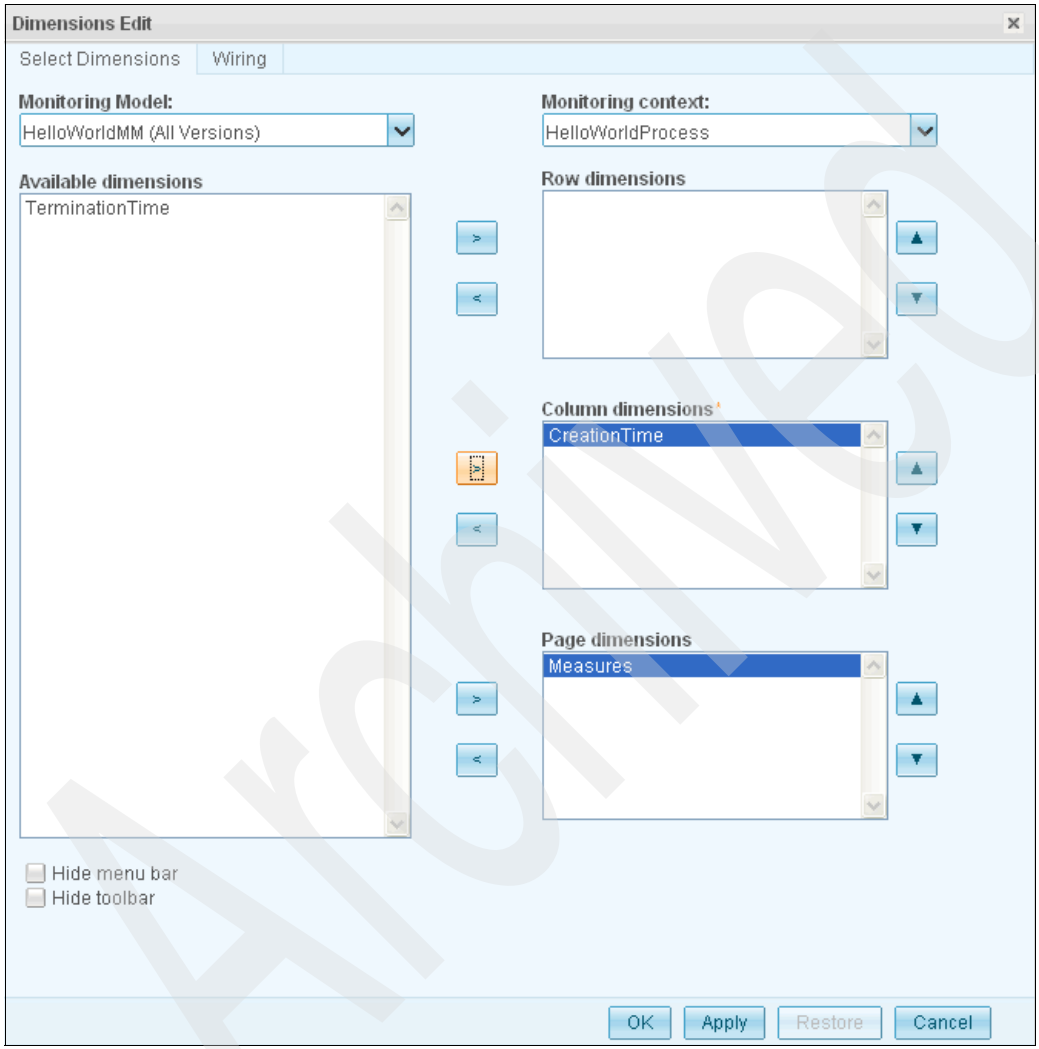


Figure B-8 Configure Dimension widget

11. Click **Save** and then **Finish Editing** to save the current page layout.

12. Verify the widget data:

    a. The instance widget data should contain one line with the following column values:

    - Creation time: <*Time of event playback*>
    - HelloWorldProcess Name: Klaus
    - HelloWorldProcess State: 3 - STATE_FINISHED
    - Completed: Yes (green star)

    b. Both KPIs displayed on the KPI widget should have an average processing time of several seconds, but different from zero.

    c. For the dimension widget, drill down one level to the year view by double-clicking **All Creation Time**. You will see one instance count in the current year.

Emit sample events with Business Space Choreographer

In this section we use the sample HelloWorld process to emit events. Use the BPC Explorer tool to get the HelloWorldModuleApp to send a few events through to the WebSphere Business Monitor HelloWorldMM model.

1. Log in to the BPC application:

   `http://<web server host>/bpc`

   You will see the page shown in Figure B-9.

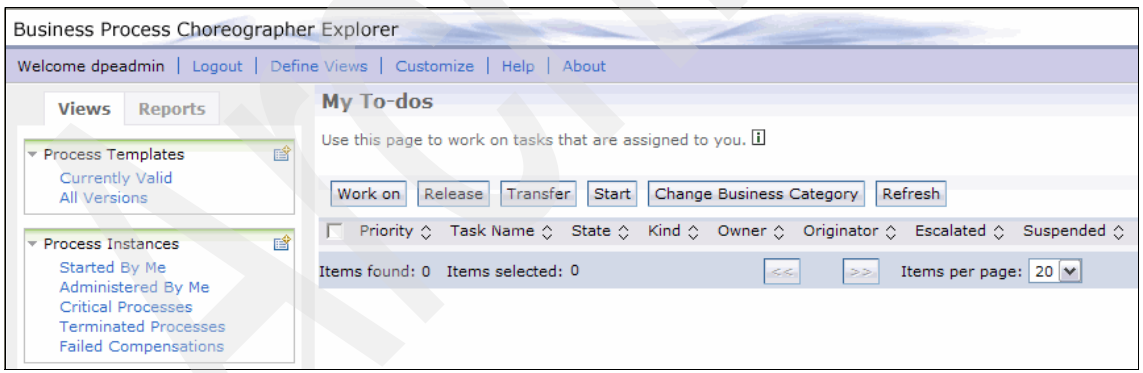


Figure B-9 BPC home page

2. Click the **All Versions** link in the Process Templates pane.

3. Click the **HelloWorldProcess** link, as shown in Figure B-10.

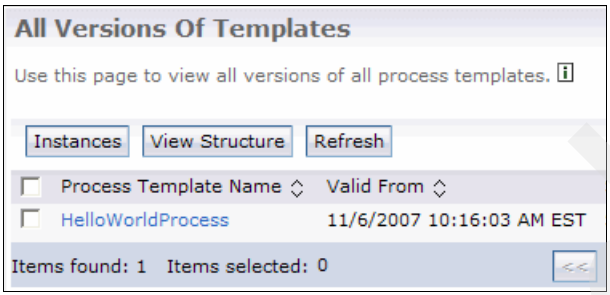


Figure B-10 Click link to enter the process

4. Click **Start Instance**, as shown in Figure B-11.

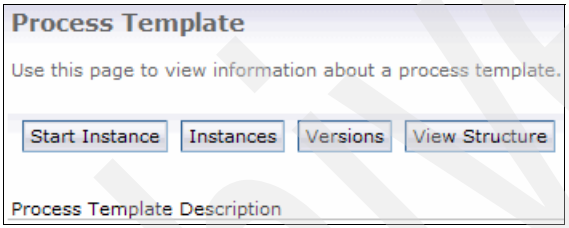


Figure B-11 Click button to start an instance

Archived

5. Enter a test string as a name and click **Submit**, as shown in Figure B-12.

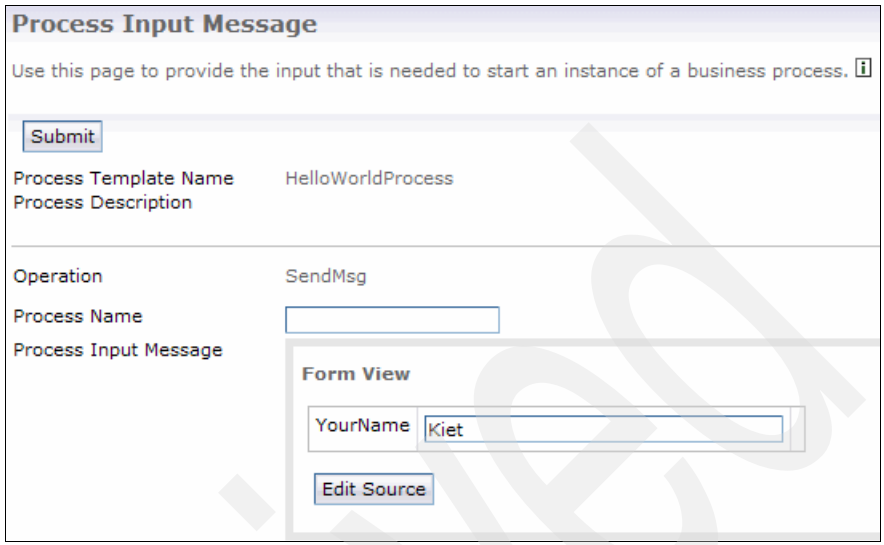


Figure B-12 Enter a test string and click Submit

6. Your event is now forwarded on to the HelloWorld model to be consumed. Use the Business Space to verify that events have been processed by the HelloWorld monitor model.
7. Verify the widget data. The instance widget data should contain a new line with the following column values:
   - Creation time: <*Time of event playback*>
   - HelloWorldProcess Name: <*The name you entered in step 5*>
   - HelloWorldProcess State: 3 - STATE_FINISHED
   - Completed: Yes (green star)
8. Both KPIs displayed on the KPI widget should have an average processing time of several seconds, but different from zero.
9. For the dimension widget, drill down one level to the year view by double-clicking **All Creation Time**. The instance count dimension will have been updated.

Archived

Verify cross-cell event distribution using the sample application

This section explains how to validate the cross-cell queue bypass event distribution by installing a business process on a remote WebSphere Process Server cell.

1. Install a the HelloWorld process on the WebSphere Process Server remote cell where the cross-cell queue bypass CEI distribution mode has been setup. Follow the instructions in “Install WebSphere Process Server module” on page 424.
2. Change the CEI distribution mode of the HelloWorld Monitor Model.
   a. In the Integrated Solutions Console navigate to the installed Monitor Models **Applications** → **Monitor Models** and click the monitor model version link named **2008-01-21T00:00:00** right to the HelloWorldMM monitor model, as shown in Figure B-13.

Figure B-13 Open Monitor Model version configuration page

   b. Change the distribution mode to Inactive in the Target list box and click **Apply**, as shown in Figure B-14.

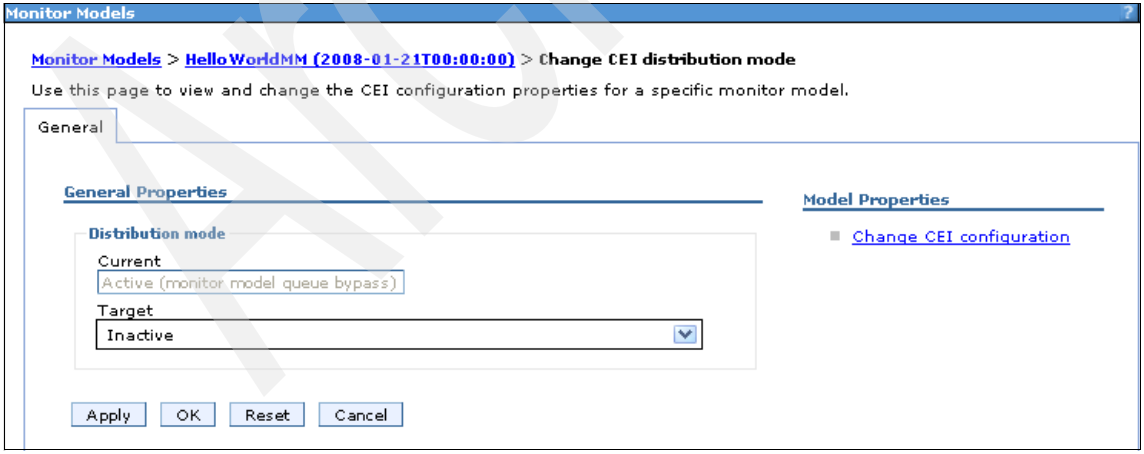


Figure B-14 Change CEI distribution mode to inactive

3. Wait until the distribution mode changes from `Active (monitor model queue bypass)` over `Active (monitor model queue bypass) to Inactive` to `Inactive`. Refresh the page to see the status changing.
4. Click the **Change CEI distribution** link, as shown in Figure B-14 on page 435.
5. On the Change CEI configuration page click **Remote** and enter the host name and RMI bootstrap port of the remote deployment manager host. Click **Enabled** and enter the remote WebSphere Process Server administrator name and password. Then click **Refresh List**, as shown in Figure B-15.

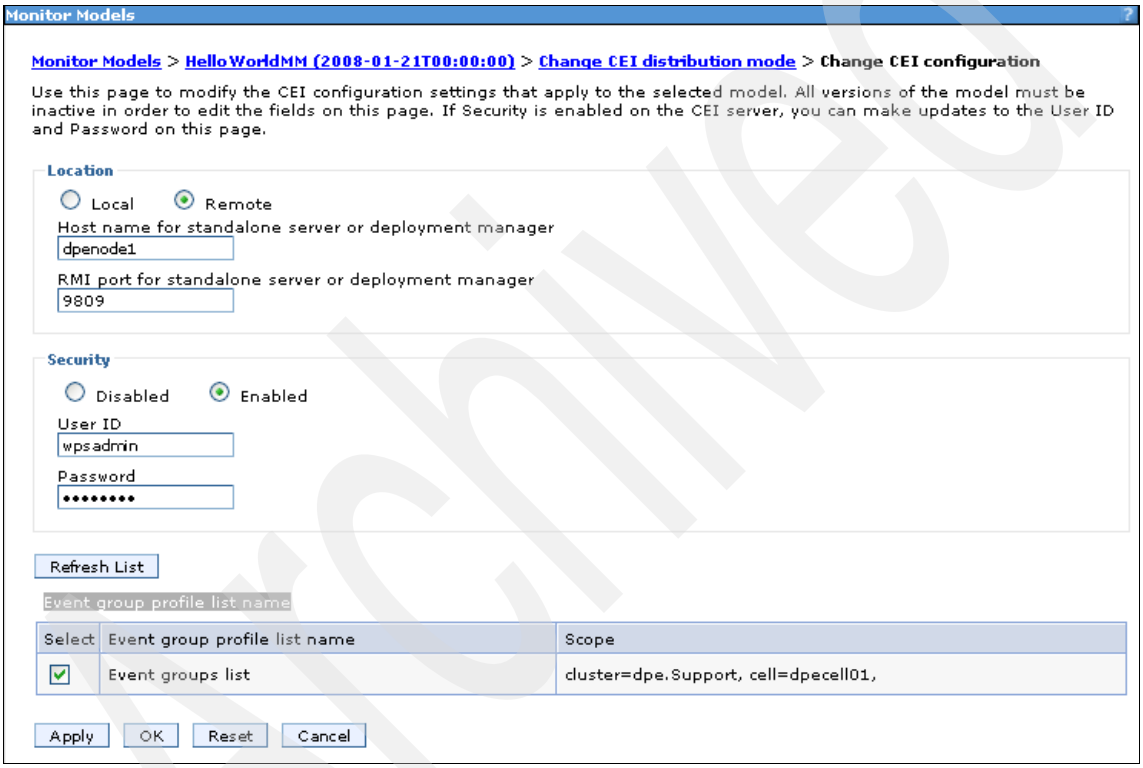


Figure B-15 CEI configuration page

6. The cluster name of the remote cell where the CEI service is deployed appears in the Event group profile list. Select the check box and click **OK**.
7. On the Change CEI distribution mode page, choose **Active (monitor model queue bypass)** from the Target list box and click **OK**.
8. Wait until current the CEI distribution shows `Active (monitor model queue bypass)`. Refresh the page to see the status changes.

Archived

9. Perform the steps described in “Emit sample events with Business Space Choreographer” on page 432 to generate events on the remote WebSphere Process Server cell.
10. Your events are now forwarded on to the HelloWorld model to be consumed. Use the Business Space to verify that the events have been processed by the HelloWolrd monitor model.
11. Verify the widget data. The instance widget data should contain a new line with the following column values:
    - Creation time: <*Time of event playback*>
    - HelloWorldProcess Name: <T*he name you entered in BPC Explorer*>
    - HelloWorldProcess State: 3 - STATE_FINISHED
    - Completed: Yes (green star)
12. Both KPIs displayed on the KPI widget should have an average processing time of several seconds, but different from zero.
13. For the dimension widget, drill down one level to the year view by double-clicking **All Creation Time**. The instance count dimension will have been updated.

Archived

Archived

C

Deployment environment definition files

When you want to define a deployment environment file for subsequent command-line deployment environment generation, typically you would use the Integrated Solutions Console. However, if you do not have access to it then you can use the template files that we supply with this book and the table in this chapter to create an appropriate file for the deployment environment required.

Table C-1 lists the templates provided as part of the additional materials. We have based these templates on V7 of the software. Future versions or fixpacks may involve changes to the format of the XML file.

Table C-1 Example template files per deployment environment

Deployment environment topology	Template name
WebSphere Process Server Single Cluster	wps.SingleCluster-DE-Template.xml
WebSphere Process Server Remote Messaging and Remote Support	wps.RMRS-DE-Template.xml
WebSphere Business Monitor Cross Cell	wbm.RMRSW-DE-Template.xml
WebSphere Dynamic Process Edition	dpe.RMRSW-DE-Template.xml
WebSphere Business Services Fabric	fab.RMRS-DE-Template.xml

© Copyright IBM Corp. 2010. All rights reserved.

Finding the required properties

Each of the template XML files has been parameterized using a consistent set of values. To find the appropriate values you require for a particular deployment environment you can read through the template file or you can use the supplied `getParams.sh` script. For example, to find out which parameters are used in the WebSphere Business Monitor cross-cell deployment environment use:

```
./getParams.sh wbm.RMRSW-DE-Template.xml
```

The output is shown in Example C-1.

Example: C-1 Parameters required in WebSphere Business Monitor Cross Cell topology

```
BusinessSpaceDatabaseName
BusinessSpaceSchemaName
CEIJMSAuthPassword
CEIJMSAuthUser
CEIMESchemaName
CellName
DBHost
DBPassword
DBPort
DBUsername
MessagingDBName
MonitorAlphabloxPassword
MonitorAlphabloxUserName
MonitorDBName
NodeName1
NodeName2
RESTProxyHost
RESTProxyPort
TopologyName
```

These are the values that should be changed in the supplied `properties.sh` file.

Generating a deployment environment definition

Once you have determined which parameters are required for your deployment environment, you must make changes to those values in the `properties.sh` file. This is a simple file with name=value pairs.

Archived

After you have edited the `properties.sh` file you can generate the XML definition using the `createXML.sh` script. For example, to generate the deployment environment XML file for WebSphere Business Monitor Cross Cell topology use:

```
./createDE.sh wbm.RMRSW-DE-Template.xml wbm.rmrsw.xml
```

The output is the `wbm.rmrsw.xml` file. If you omit the output filename it defaults to `output.xml`. The output file should now have all the appropriate parameters replaced with real values and can be used to generate a new deployment environment either using jython calls and wsadmin or you can import it into the Integrated Solutions Console.

Furthermore, a second file is created with the extension .py (for example, `wbm.rmrsw.xml.py`) that contains the jython commands to import, validate, and generate the deployment environment using the command line if you do not want to use the Integrated Solutions Console.

Archived

Archived

Abbreviations and acronyms

APIs	Application Programming Interfaces
BFM	Business Flow Manager
BPC	Business Process Choreographer
BPE	Business Process Engine
BPEDB	Business Process Choreographer database
BPM	Business Process Management
BRM	Business Rules Manager
BSPCDB	Business Space database
CBE	Common Business Events
CEI	Common Event Infrastructure
DBA	Database Administrator
DDT	Database Design Tool
DE	Deployment Environment
DMGR	Deployment Manager
DMS	Data Movement Service
EVENT	Event database
HTM	Human Task Manager
IBM	International Business Machines Corporation
IHS	IBM HTTP Server
ISC	Integrated Solutions Console
ITSO	International Technical Support Organization
JVM	Java Virtual Machine
LDAP	Lightweight Directory Access Protocol
ME	Messaging Engine
MEDB	Messaging Engine Database
MONDB	Monitor database
QoS	Quality of Service
RAL	Remote Artifact Loader
REST	Representational State Transfer
SCA	Service Component Architecture
SIB	Service Integration Bus
SOA	Service Oriented Architecture
SSO	Single Sign-On
WBCDB	WebSphere Business Compass Database
WLM	Work Load Manager

Archive

© Copyright IBM Corp. 2010. All rights reserved.

Archived

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks publications

For information about ordering these publications, see “How to get Redbooks” on page 445. Note that some of the documents referenced here may be available in softcopy only.

- *IBM HTTP Server (powered by Apache): An Integrated Solution for IBM eServer iSeries Servers*, SG24-6716
- *WebSphere Application Server Network Deployment V6: High Availability Solutions*, SG24-6688
- *WebSphere Business Process Management V6.1.2 Production Topologies*, SG24-7665

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks publications, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

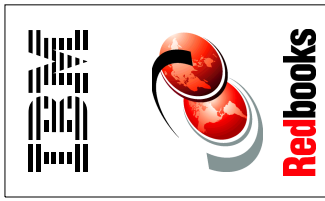
IBM Global Services

ibm.com/services

Archived

© Copyright IBM Corp. 2010. All rights reserved.

Archived



WebSphere Business Process Management V7 Production Topologies

IBM CONFIDENTIAL

Archived

Archived

WebSphere Business Process Management V7 Production Topologies

Build and extend WebSphere Process Server topologies

Explore WebSphere Dynamic Process Edition topologies

Learn by example with practical scenarios

This IBM Redbooks publication describes how to build production topologies for Business Process Management (BPM) solutions. It is aimed at IT Architects and IT Specialists who want to understand and implement these topologies. Use this book to select the appropriate production topologies for a given environment, then follow the step-by-step instructions included in this book to build these topologies. You must have a high-level understanding of WebSphere Business Process Management products to get the most out of this book. This book addresses the following WebSphere products:

- WebSphere Process Server V7
- WebSphere Business Monitor V7
- WebSphere Business Services Fabric V7
- WebSphere Enterprise Service Bus V7
- WebSphere Business Compass V7

Part 1, "Overview" on page 1, introduces the BPM products that we discuss and provides an overview of basic topology terminology. This part also provides an overview of the production topologies that we describe in this book, including a selection criteria for when to select each topology.

Part 2, "Building production topologies" on page 53, provides a series of step-by-step instructions for creating production topology environments using deployment environment patterns. We describe topologies using the Integrated Solutions Console and a command-line interface.

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

SG24-7854-00 ISBN 0738434345