WebSphere software

IBM

WebSphere Business Process Management V6.2 Production Topologies

- Building and extending WebSphere Process Server topologies
- Incorporating WebSphere Business Services Fabric
- Integrating WebSphere Business Monitor

Martin Keen
Naveen Balani
Addison Goering
Sila Kissuu
Leon Matthews
Thomas McManus
Catherine Rivi
Mohamed ShamsEldin Salem
Jim Thorpe
Srinivasa Vadlamudi

Redbooks

ibm.com/redbooks

International Technical Support Organization

WebSphere Business Process Management V6.2 Production Topologies

June 2009

SG24-7732-00

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

First Edition (June 2009)

This edition applies to Version 6.2 of WebSphere Process Server, WebSphere Business Services Fabric, and WebSphere Business Monitor.

Contents

Notices

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

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

AIX®	developerWorks®	Rational®
AlphaBlox®	FileNet®	Redbooks®
ClearCase®	HACMP™	Redbooks (logo) ®
ClearQuest®	i5/OS®	System z®
Cool Blue™	IBM®	Tivoli®
DataPower®	Lotus Notes®	WebSphere®
DB2 Universal Database™	Lotus®	z/OS®
DB2®	Notes®	

The following terms are trademarks of other companies:

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

SUSE, the Novell logo, and the N logo are registered trademarks of Novell, Inc. in the United States and other countries.

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

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

SAP NetWeaver, SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

EJB, Enterprise JavaBeans, J2EE, J2ME, Java, JavaBeans, JavaServer, JDBC, JMX, JSP, JVM, Solaris, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Excel, 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.

Preface

IBM®'s WebSphere® Dynamic Process Edition V6.2 is a comprehensive set of role-based, service-oriented architecture (SOA)-enabled product capabilities providing customers with the ability to continuously optimize processes and adapt them to rapidly changing needs. This IBM Redbooks® publication addresses the configuration, administration, and security of the key runtime environments in WebSphere Dynamic Process Edition:

- WebSphere Process Server V6.2
- WebSphere Business Services Fabric V6.2
- WebSphere Business Monitor V6.2

Part 1, "Overview" on page 1, of this book introduces production topology concepts and terminology, and provides security considerations.

Part 2, "Building topologies for WebSphere Process Server" on page 97, provides a series of step-by-step instructions for selecting and creating a production topology environment based on WebSphere Process Server deployment environment patterns. You will learn how to secure this environment and administer it. This part also contains chapters on extending these topologies, monitoring them with IBM Tivoli® Monitoring, and accessing them with Business Space powered by WebSphere and Lotus® Forms Client.

Part 3, "Extending the Remote Message and Remote Support topology" on page 383, extends the Remote Messaging and Remote Support production topology for WebSphere Process Server, describing step-by-step how to add WebSphere Business Services Fabric and WebSphere Business Monitor to that topology.

Finally, Part 4, "Four Cluster production topology" on page 507, describes how to build a complete WebSphere Dynamic Process Edition production topology from the ground up using the new Four Cluster production topology.

A separate publication covering z/OS® titled *z/OS: WebSphere Business Process Management V6.2 Production Topologies*, SG24-7733, is also available.

The team that wrote this book

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

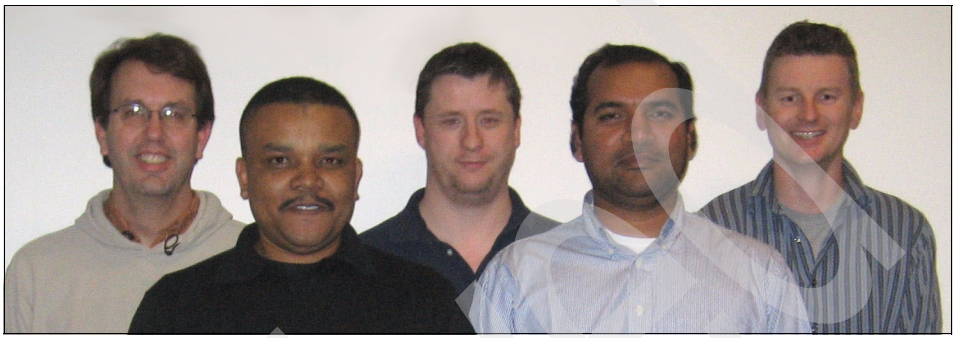


Figure 1 From left to right: Addison, Sila, Leon, Srini, and Martin

Martin Keen is a Consulting IT Specialist at the ITSO, Raleigh Center. He writes extensively about WebSphere products and SOA. He also teaches IBM classes worldwide about WebSphere, SOA, and ESB. Before joining the ITSO, Martin worked in the EMEA WebSphere Lab Services team in Hursley, UK. Martin holds a bachelor's degree in Computer Studies from Southampton Institute of Higher Education.

Naveen Balani works as a Software Architect with IBM India Software Labs (ISL). He leads the design and development activities for the WebSphere Business Service Fabric product out of ISL. He likes to research new technologies and is a regular contributor to IBM developerWorks®, having written about Web services, ESB, JMS, service-oriented architecture (SOA), architectures, open source frameworks, semantic Web, J2ME™, pervasive computing, and Spring, Ajax, and various IBM products. He is also a coauthor of *Beginning Spring Framework 2* and *Getting Started with IBM WebSphere Business Services Fabric V6.1*, SG24-7614.

Addison Goering is a Courseware Developer and Instructor with WebSphere Education. He has 10 years of experience in developing and delivering education with a specific focus on WebSphere products for the past nine years. His area of focus is WebSphere Business Services Fabric.

Sila Kissuu is a Senior SOA Solutions Architect with IBM Software Group - Application and Integration Middleware Software. He has over 10 years of experience in architecture and development of enterprise BPM solutions. Sila

has attained multiple IBM professional and product certifications and holds degrees in Economics and Computer Science.

Leon Matthews is an Advisory Software Engineer with the IBM Software Group in the USA. He has 10 years of experience in development of enterprise-ready transformation products and consultancy and project management of customer applications. His areas of expertise include WebSphere Process Server and WebSphere Transformation Extender.

Thomas McManus is a Senior Software Engineer with IBM SWG Business Partner Technical Strategy and Enablement. He has 10 years of experience deploying, administering, and securing middleware topologies. Tom is an IBM Certified SOA Solution Architectural Designer, IBM Certified Administrator for SOA Solutions—WebSphere Process Server V6.0, and IBM Certified Solution Developer—Web Services Development.

Catherine Rivi is a Technical Writer and Software Tester on the ISSW for IBM team. She has over 15 years of experience in testing and documentation of WebSphere products. In addition to testing internal IBM applications and new product releases, she edits articles and tutorials on BPM and other topics. She is an IBM Certified System Administrator - WebSphere Application Server Network Deployment.

Mohamed ShamsEldin Salem is a Senior IT Specialist with the IBM software group in Cairo Technology and Development Center (C-TDC) Egypt. He has six years of experience working in WebSphere Business Monitor information development, development, and SWAT teams. Mohamed has experience in installation, configuration, and security for the WebSphere product stack. He provides technical support for WebSphere products in Europe and Africa regions in critical customer situations. Mohamed holds a bachelor's degree in Computer Engineering from Cairo University - Egypt. He is a certified software solution developer for WebSphere Business Monitor, WebSphere Integration Developer, and WebSphere Portal Server.

Jim Thorpe is a Senior Software Engineer on the WebSphere Business Monitor Development team. He has 25 years of experience with IBM working on numerous hardware and software products. He is also a Lab Advocate for the New York State Department of Taxation and Finance.

Srinivasa Vadlamudi is an Advisory Software Engineer with the WebSphere Process Server Level2 Support team in the USA. He has over 10 years of experience in the Software Development, Software Configuration Management, and Release Engineering fields. He is currently pursuing his MBA from North Carolina State University in Raleigh, NC. His areas of expertise include WebSphere Process Server, and in his previous roles in IBM he worked extensively on Rational® ClearCase® and ClearQuest®.

Special thanks to the authors of *WebSphere Business Process Management V6.1.2 Production Topologies*, SG24-7665: Peter Daly, Ryan Malynn, Thomas McManus, Karen Poyer, Julia Reder, Mohamed Shamseldin Salem, Kevin Senior, Jeffrey Slone, and Vignesh Velusamyravindran.

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

Eric Herness
IBM BPM Cheif Architect

Karri S Carlson Neumann
Advisory Software Engineer in the BPM Customer First Bring Up Lab in Rochester Minnesota

Jim Pailistrant
Worldwide Tivoli Pre-Sales Technical Enablement Engineer

Vincent Belfoure, Paul Pacholski, Josh Bock, Julie Reese, Dennis Miller, Jared Michalec, Christopher Walk, Michael Steele, Stuart Jones, Birgit Schwarz, Manfred Haas, Kurt Fleckenstein, and Shyam Nagarajan
IBM Worldwide BPM Technical Sales

John Alcorn, Richard Johnson, Dave Enyeart, Luis Sanchez, Phil Piper, and Varadarajan (Varad) Ramamoorthy
IBM WebSphere Business Monitor Development

Stephen Gibney and Peter Daly
IBM Software Services for WebSphere

Mohamed Saeed and Hui Ming Zhong
IBM WebSphere Business Monitor SWAT team

Marilza Maia
SOA Advanced Technology, IBM Software Group

Authur Kevin McGrath
IBM WebSphere Instructor

Shu Jun Tang
WebSphere Process Server Validation, IBM China Development Lab

Bobby Pham
IBM Business Partner Technical Strategy and Enablement (BPTSE)

Become a published author

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

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks 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

Part 1

Overview

1

IBM business process management products and concepts

This chapter describes the IBM business process management (BPM) suite. It outlines how each product is used in the BPM life cycle, and includes key terminology and concepts related to BPM and network deployment. It contains the following sections:

- The IBM BPM suite
- IBM WebSphere Dynamic Process Edition
- Unified interface for BPM suite: Business Space powered by WebSphere
- Additional products for BPM solutions
- Network deployment concepts

1.1 The IBM BPM suite

The IBM Business Process Management (BPM) suite is a set of collaborative, role-based capabilities that allow you to model, simulate, execute, rapidly change, monitor, and optimize business processes. The IBM BPM suite offers a choice between two foundational offerings:

- The IBM WebSphere Dynamic Process Edition
- The IBM FileNet® Active Content Edition

A diagram of these offerings and their components is shown in Figure 1-1.

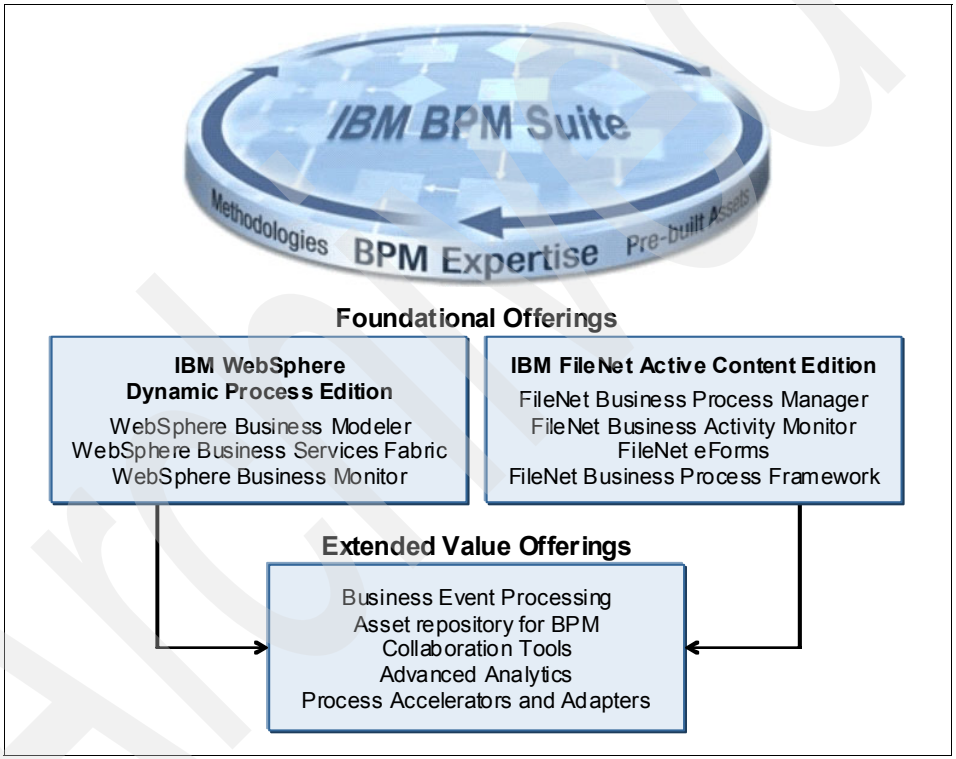


Figure 1-1 The IBM BPM Suite and its two foundation offerings

WebSphere Dynamic Process Edition provides a comprehensive foundation for modeling, deploying, continuously optimizing, and monitoring dynamic business processes that include human workflows. WebSphere Dynamic Process Edition consists of:

- WebSphere Business Modeler
- WebSphere Business Services Fabric
- WebSphere Business Monitor

IBM FileNet Active Content Edition supports process optimization where content is core to your processes or you have integrated compliance requirements. FileNet Active Content Edition can manage the full life cycle of complex business processes dealing with unstructured information assets in areas such as human resources, accounts payable and invoice processing, and contract life cycle management.

This IBM Redbooks publication focuses on building production topologies that contain WebSphere Dynamic Process Edition. It also covers how to extend those topologies with additional products and capabilities.

1.2 IBM WebSphere Dynamic Process Edition

IBM WebSphere Dynamic Process Edition is the core offering from the IBM BPM suite. WebSphere Dynamic Process Edition is a comprehensive set of role-based products, enabled for service-oriented architecture (SOA), that allow you to optimize processes continuously and adapt them to rapidly changing needs. It includes three products:

- IBM WebSphere Business Modeler Advanced V6.2

  This contains tools for business users to visualize, understand, document, and simulate business processes including human workflows and dynamic service selection.

- IBM WebSphere Business Services Fabric V6.2

  This is an SOA-based process engine capable of unique dynamic execution of business processes determined at run time based on business service policies and diverse, managed service selection. WebSphere Business Services Fabric includes WebSphere Process Server, a high-performance integration server for business processes.

- IBM WebSphere Business Monitor V6.2

  This provides comprehensive business activity monitoring that provides a real-time view of your business processes and operations.

Figure 1-2 shows how these and other IBM products are used in a typical BPM life cycle.

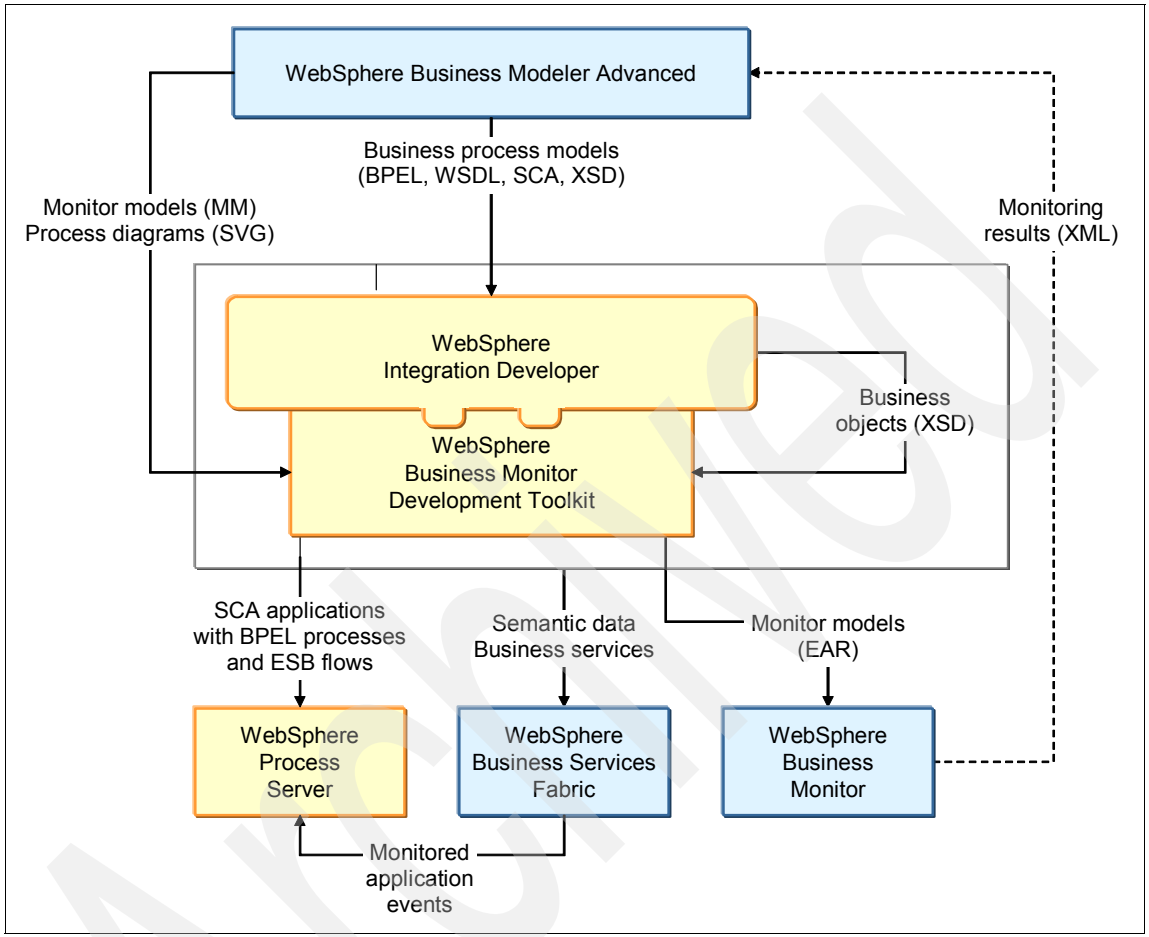


Figure 1-2 Using IBM products throughout the BPM life cycle

The stages of the life cycle and products that can be used in each stage are:

- Model and simulate business processes.

  Business users document and model a business process using WebSphere Business Modeler. The model is then refined and improved through simulation and analysis of *what if* scenarios. The model can be published to colleagues for review and modification using the WebSphere Business Modeler Publishing Server. A monitor model is created that defines the goals for the business process and indicates how the results should be measured. See 1.2.1, “WebSphere Business Modeler” on page 8.

- Deploy business processes.

  IT specialists complete the monitor model and implement the business process using WebSphere Integration Developer. The monitor model guides how the business process is enabled to emit events that are correlated with business occurrences. The resulting Service Component Architecture (SCA) application and other artifacts are deployed into the WebSphere Process Server run time environment. (WebSphere Process Server is included as part of WebSphere Dynamic Process Edition.) The execution of the business process relies on the infrastructure of WebSphere Application Server and WebSphere Enterprise Service Bus. See 1.2.2, “WebSphere Process Server” on page 9, 1.2.3, “WebSphere Application Server” on page 10, and 1.2.4, “WebSphere Enterprise Service Bus” on page 10.

- Dynamically assemble and customize business services.

  WebSphere Business Services Fabric is used to dynamically assemble composite business applications from business services. Descriptive metadata governs the run time execution of the services based on policies, intended delivery channels, and roles. This allows for flexible processes that can rapidly respond to a changing business climate or other demands. See 1.2.5, “WebSphere Business Services Fabric” on page 11.

- Monitor, analyze, and optimize business processes in real time and respond accordingly.

  WebSphere Business Monitor is used to monitor and analyze business events over time, collect and display data, and feed improvements back into the model. The run time values of the previously defined metrics and key performance indicators (KPI) are used to gauge the performance of the business process, identify bottlenecks, and emit warnings when thresholds are exceeded. The results are accessed through dashboards in Business Space powered by WebSphere. See 1.2.6, “WebSphere Business Monitor” on page 11.

- Access a single interface for BPM activities and collaboration.

  Day to day, business users collaborate and interact with multiple sources of business data across the BPM suite through a single graphical user interface.

Business Space powered by WebSphere provides an integrated view of the business process. Each BPM feature is mapped to a business role and exposed through out-of-the-box and custom Web 2.0 pages, where you can view KPIs, work on human tasks, and modify rules and policies. See 1.3, “Unified interface for BPM suite: Business Space powered by WebSphere” on page 12.

1.2.1 WebSphere Business Modeler

WebSphere Business Modeler is used to document, visualize, and report on business process models. It works with an asset repository to manage assets across the BPM life cycle, which increases the reuse and traceability of process model components. The WebSphere Business Modeler product family includes the following three editions:

- WebSphere Business Modeler Basic
- WebSphere Business Modeler Advanced
- WebSphere Business Modeler Publishing Server

While Basic is considered an entry-level process modeling tool, Advanced and Publishing Server consist of more feature-rich tools that allow greater control over process management.

WebSphere Business Modeler Advanced

WebSphere Business Modeler Advanced offers all of the capabilities to document, visualize, and report on business process models that Basic offers, while adding modeling, simulation, and analysis capabilities. You can model your business with drag-and-drop capabilities to help set up simulations. This makes it easy to analyze workloads and bottlenecks.

New in V6.2 is a direct-deploy scenario that allows you to directly deploy models into the WebSphere Process Server run time. V6.2 also introduces the Interactive Process Design tool for defining and testing a complete BPM solution in a sandbox environment prior to deployment. It includes a test server for deploying the process and a preconfigured test business space for validating process and monitoring logic.

Business analysts can model a process, browse for pre-built business services, create forms through editors, and then test these processes on a server. The testing is not emulation. It is testing on a server to see how the process will run on the server when deployed. A preconfigured Business Space is generated as part of this testing.

WebSphere Business Modeler Publishing Server

IBM WebSphere Business Modeler Publishing Server provides a way to publish business processes and related business information, such as process models, organization diagrams, dashboard designs, and user interface form images, to a secure website. Publishing business processes and other BPM assets in a Web-based format allows various stakeholders from around the world to view and contribute to the development of best practices in the business processes.

With WebSphere Business Modeler Publishing Server you can import user-interface form views for human tasks and then review and comment on the forms. You can also display process models created with Business Process Modeling Notation (BPMN). BPMN is a standard graphical notation for drawing business practice models.

1.2.2 WebSphere Process Server

WebSphere Process Server is an SCA-compliant integration server for business processes that provides a fully converged, standards-based process engine underpinned by WebSphere Application Server. Along with WebSphere Enterprise Service Bus, it is a strategic product for integration and modernization of IT assets, including core systems using SOA.

SCA is a model for application development that separates the application function from the implementation details. SCA defines *modules* and *components* that are connected using standard interfaces. A module performs or supports a specific business function and can be deployed directly. A component is a discrete, reusable unit that provides published interfaces and references other components' interfaces.

Following the principles of SCA, WebSphere Process Server has a single invocation model, a single data model, and a component-based framework. Everything in WebSphere Process Server is a component. These components have an interface and can be wired together to form a module. This modular arrangement enables the changing of any part of an application without affecting the other parts. For example, a module of a given type can be replaced with a module of the same type without the need to modify the business process.

WebSphere Process Server V6.2 provides enhanced flexibility and control over process instances, and a new direct-deploy scenario that allows you to directly deploy modules from WebSphere Business Modeler into the WebSphere Process Server run time. It also offers new widgets for Business Space, and a new explorer-style Web application for browsing the service integration bus.

1.2.3 WebSphere Application Server

WebSphere Application Server is the foundation of the IBM WebSphere software family and a key building block for SOA. It provides a transaction engine for building, running, integrating, and managing dynamic applications.

WebSphere Application Server Network Deployment allows you to run services in a reliable, scalable, highly available environment to ensure that business opportunities are not lost due to application downtime.

1.2.4 WebSphere Enterprise Service Bus

WebSphere Enterprise Service Bus is the mediation layer that runs on top of the transport layer within WebSphere Application Server Network Deployment. As such, WebSphere Enterprise Service Bus provides prebuilt mediation functions and easy-to-use tools to enable rapid construction and implementation of an ESB as a value-add on top of WebSphere Application Server.

For integration to be successful, SOA needs a single invocation model and a single data model. WebSphere Enterprise Service Bus uses SCA as its invocation model and Service Data Objects (SDOs) for its data representation. The Common Event Infrastructure (CEI) provides basic event management services, which enable monitoring of service components.

There are four basic tasks that an ESB must perform:

- Route messages among services.
- Transform message formats when necessary.
- Convert protocols for the consumer and provider.
- Handle events from different services.

WebSphere Enterprise Service Bus conforms to all Web services standards to achieve these basic capabilities. It uses SOAP with either Java™ Message Service (JMS) or HTTP. It can also communicate with WebSphere MQ, WebSphere Message Broker, or an adapter.

The modules in charge of performing the operations for WebSphere Enterprise Service Bus are called *mediation components*. These mediation components are built using WebSphere Integration Developer. To aid developers, this tool has features similar to an assembly diagram editor, a mediation flow editor, and a visual debugger. When created, the mediation modules are deployed to WebSphere Enterprise Service Bus.

1.2.5 WebSphere Business Services Fabric

WebSphere Business Services Fabric is an SOA-enabled product for discovering, assembling, deploying, managing, and governing composite business applications (CBAs). It consists of the Foundation Pack (the run time and management environment) and the Tool Pack (the design and assembly environment).

WebSphere Business Services Fabric enables the assembly of existing and new business services into CBAs. These policy-driven CBAs are adapted at run time based on semantic knowledge to provide the correct business service functionality. The functionality is delivered through a preferred communication channel based on the business context, content, and contract of the service request. These constraints prescribe variations in the business service, dynamically altering the process behavior or end-point routing.

WebSphere Business Services Fabric V6.2 contains predefined Business Space widgets for authoring and changing business services, assembling services into an end-to-end business process, and configuring business services at run time. WebSphere Business Services Fabric also includes enhanced industry content packs. Industry content packs are prebuilt industry-specific assets (patterns, templates, and code) used to accelerate business service deployment. The assets are based on standards and models for a variety of industries such as telecommunications, insurance, and health care.

1.2.6 WebSphere Business Monitor

WebSphere Business Monitor is a comprehensive business activity monitoring solution that provides a near real-time view of business performance. It monitors activities or processes by receiving and processing business events, called *common base events*, from business applications. The events reflect business activity. Information processed from events is stored in the Business Monitor database.

To monitor business operations, WebSphere Business Monitor offers the following functions:

- Captures business-related data based on the monitor model that you design and install
- Extracts the measurement variables from the data
- Transforms the variables into metric and key performance indicator (KPI) values
- Displays the measurement values on dashboards in Business Space

- Provides business intelligence insight through dimensional analysis and reporting
- Enables you to define actions to take when specified situations occur
- Identifies and notifies you of operation failures for inspection and analysis

WebSphere Business Monitor V6.2 now includes the ability to view historical KPI data over time to see trends and to predict future behavior. It also provides a new feature for creating alerts at runtime. Monitored data can now be viewed in Excel®, Lotus SameTime, Lotus Notes®, and on Apple iPhones. Version 6.2 also includes industry content (monitor models, customized dashboards, and sample events) related to the insurance, banking, health care, and other industries.

1.3 Unified interface for BPM suite: Business Space powered by WebSphere

IBM provides a set of capabilities for building a unified personalized view across the various products of the BPM suite. Business Space powered by WebSphere is a common UI shipped with the following run time products:

- WebSphere Business Monitor
- WebSphere Process Server
- WebSphere Business Services Fabric
- WebSphere Business Modeler Publishing Server

You can use Business Space to assemble integrated interfaces that access a variety of data sources through REpresentational State Transfer (REST) APIs. The capabilities include:

- Managing and monitoring from mobile devices
- Modifying rules, policies, and processes
- Reviewing KPIs, and managing task lists and staff productivity
- Analyzing metrics, and optimizing process models and dashboards

Using out-of-the-box templates and product-specific widget palettes, you can combine process content from across these BPM products to manage your processes from a central view. Various BPM features are exposed in the interface depending on the role of the BPM professional (for example, business leader, business analyst, or solution administrator). The abilities include:

- View KPI information, notifications, alerts, and historical analyses within the Business Space.
- Work with human tasks and business rules, govern changes to business artifacts, and work with business calendars. Additionally, WebSphere Process Server provides content for the IT administrator to assess the health of their BPM systems and applications, as well as ensure smooth operation.
- Create new processes or immediately change existing processes (without IT involvement) by modifying business service policy attributes linked to these processes from within Business Space. Examples of these attributes include a new role, channel, region, or a broad corporate policy.
- Share and collaborate on process models, dashboard designs, and user interaction forms within Business Space.

Complete documentation on Business Space is in the *Business Space powered by WebSphere Information Center*, found at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp?topic=/com.ibm.bspace.620.help.framework.doc/welcome/dochome.html

1.4 Additional products for BPM solutions

IBM offers additional products that can be used to extend the value of your BPM solution:

- You can monitor and manage business services by tracking service flows and ensuring that service levels are met using IBM Tivoli Composite Application Manager for SOA.
- You can use IBM Lotus Forms as the interface for human tasks and processes in WebSphere Process Server.

Each of these products is described in the following sections.

IBM Tivoli Composite Application Manager for SOA

IBM Tivoli Composite Application Manager for SOA (ITCAM for SOA) provides monitoring, controlling, and discovery features for SOA. It consists of integrated

management tools that ensure high availability and performance of your SOA solution. With ITCAM for SOA, you can:

- Keep track of your service flow through a services topology view that shows service-to-service relationships, service status, and metrics.
- Manage heterogeneous SOA platforms including the WebSphere family, Microsoft® .NET, and BEA WebLogic.
- Manage and monitor the system to ensure that service levels are met through built-in alerts, message mediations, situations, and workflows.
- Gain insight into service use, flows, and relationships.

When integrated with WebSphere Services Registry and Repository, ITCAM for SOA can provide important run time governance solutions. For example, it can help you to identify rogue services in your environment, show you the impact to a business process when services degrade, and dynamically affect routing decisions made by WebSphere Enterprise Service Bus. The routing decisions are based on real-time performance monitoring information that indicates the health and suitability of a service endpoint.

IBM Lotus Forms

IBM Lotus Forms provides a way to build a secure, dynamic front-end to forms-based business processes:

- Lotus Forms Server API: This API is a collection of programming tools for developing applications that process Extensible Forms Description Language (XFDL) forms, including XForms data models within XFDL forms. You can create and manage applications that analyze, route, validate, and create electronic forms.
- Lotus Forms Designer: Lotus Forms Designer is a design environment that lets you build forms by dragging and dropping design elements onto a canvas. You can also use the Designer to add business logic and data modeling to create forms that complement complex workflows.
- Lotus Forms Viewer: Lotus Forms Viewer is a client-side program that lets you open, fill out, sign, submit, and save XFDL forms. The Viewer can display forms as either a stand-alone application or embedded within a Web browser.

1.5 Network deployment concepts

This section defines the following concepts:

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

1.5.1 Components of a WebSphere network deployment environment

WebSphere Application Server Network Deployment solutions are built from the components discussed in this section.

Cells

A WebSphere cell is a logical 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 managed container for one or more application servers. Typically, a single node corresponds to a single machine. A node consists of a node agent, by which the node is controlled, and the application servers hosted on that node.

Node agents

The WebSphere node agent is an architectural component that enables the deployment manager for the cell to remotely manage the node, its application servers, and their applications.

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 manager runs a single application, a Web-based configuration front-end known as the Integrated Solutions Console (or administrative console), through which you can perform nearly all management tasks.

Figure 1-3 shows a cell topology that consists of two nodes, each running a node agent. Each node hosts 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, in turn, communicates with node agents, which in turn 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.

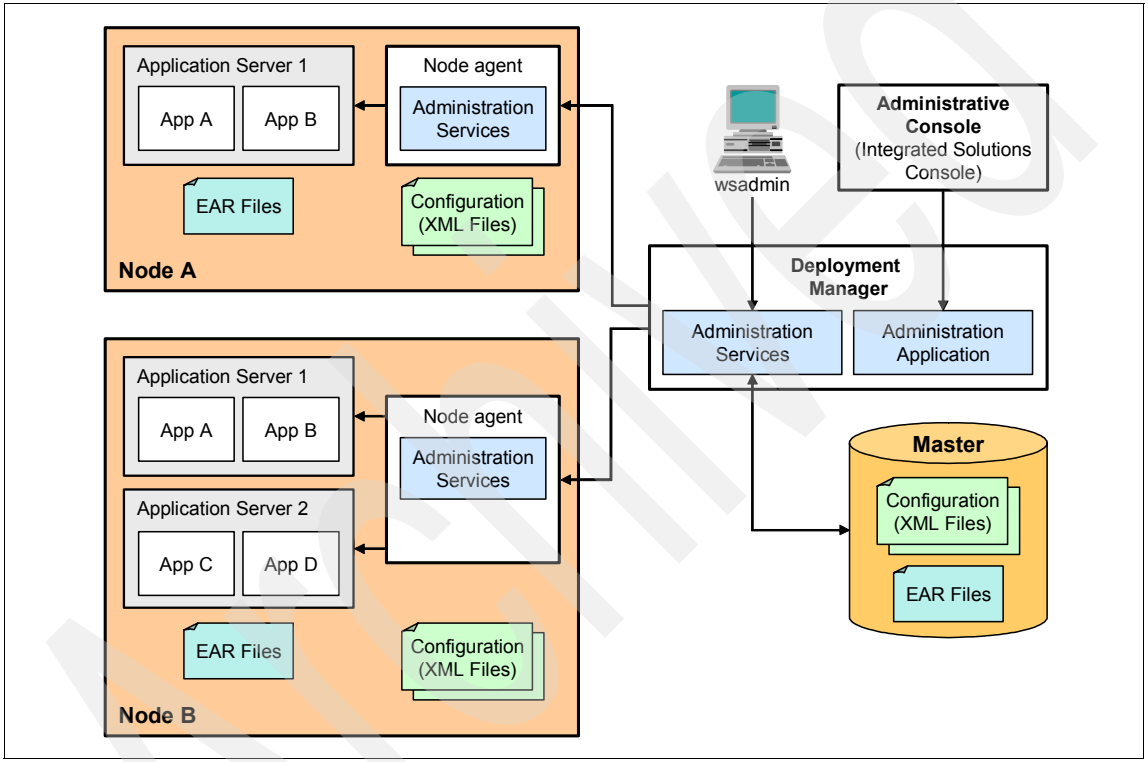


Figure 1-3 A cell topology

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 is a specific instance of a WebSphere Application Server run time environment. The WebSphere Application Server installation program places the files that it creates into one of two separate environments. It installs the core product files in one location, and in a separate location it creates an initial profile, which is a run time execution environment that includes configuration files, the default location for deployed applications, logs, and other data. All profiles on a machine can share the same core product files, which they cannot modify. There are different types of profiles, for example, deployment manager profile and custom profile, each supporting a specific functional environment for the user.

1.5.2 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 resiliency, or both.

Vertical clustering

In a vertical cluster, multiple application servers are placed onto the same node in order to better use the available resources (Figure 1-4). 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 members' node fails.

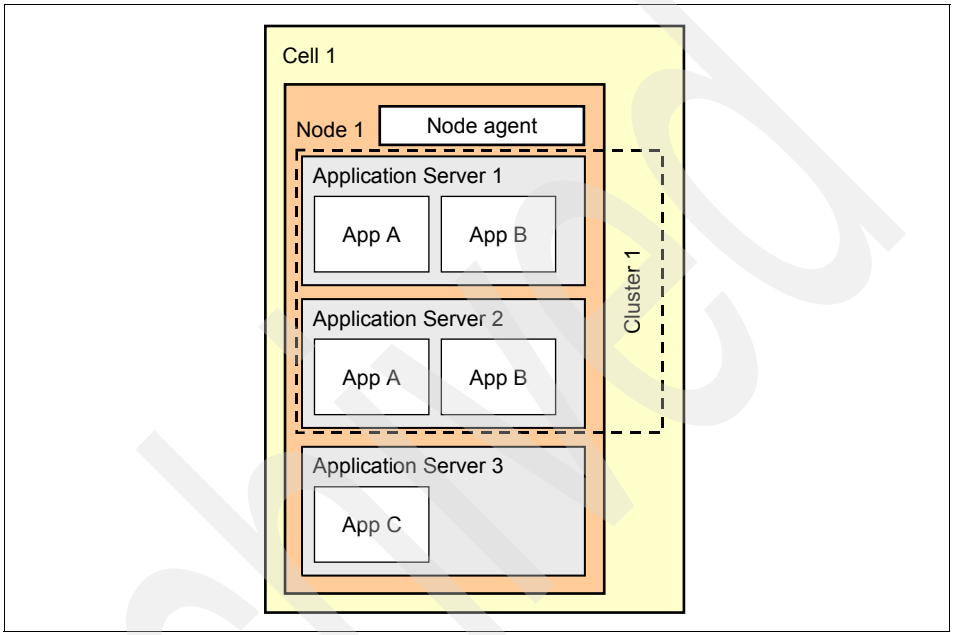


Figure 1-4 A vertically clustered WebSphere environment

Horizontal clustering

In a horizontal cluster, multiple application servers are distributed across nodes in order to use more physical resources (Figure 1-5). 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.

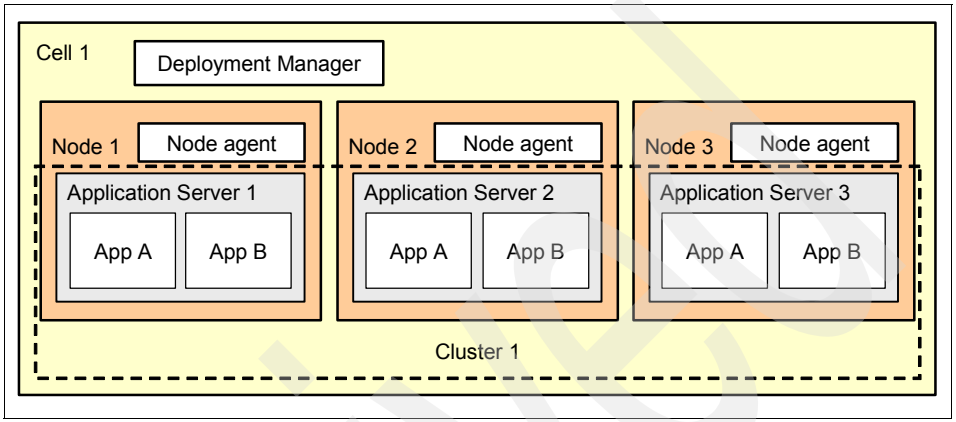


Figure 1-5 A horizontally clustered WebSphere environment

1.5.3 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 1-6).

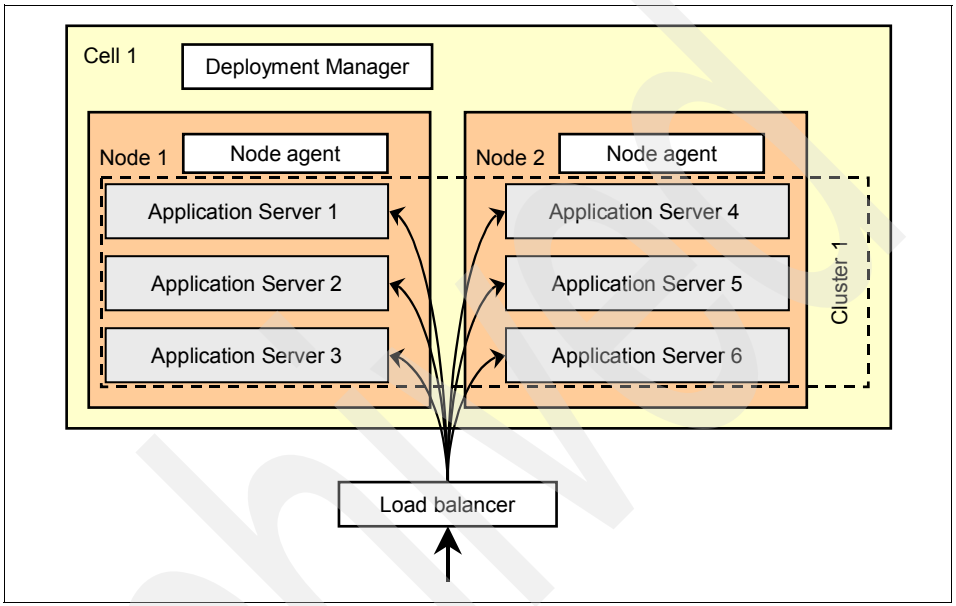


Figure 1-6 A load-balanced WebSphere environment

1.5.4 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.

Additionally, sharing of data is critical to sustain transactions if a particular application server or its node fails, as shown in Figure 1-7. In this case, another application server would be unable to continue a partially completed transaction without information about the current state of the transaction in question. Where data is not shared between application servers, all transactions started on a server that subsequently fails is lost.

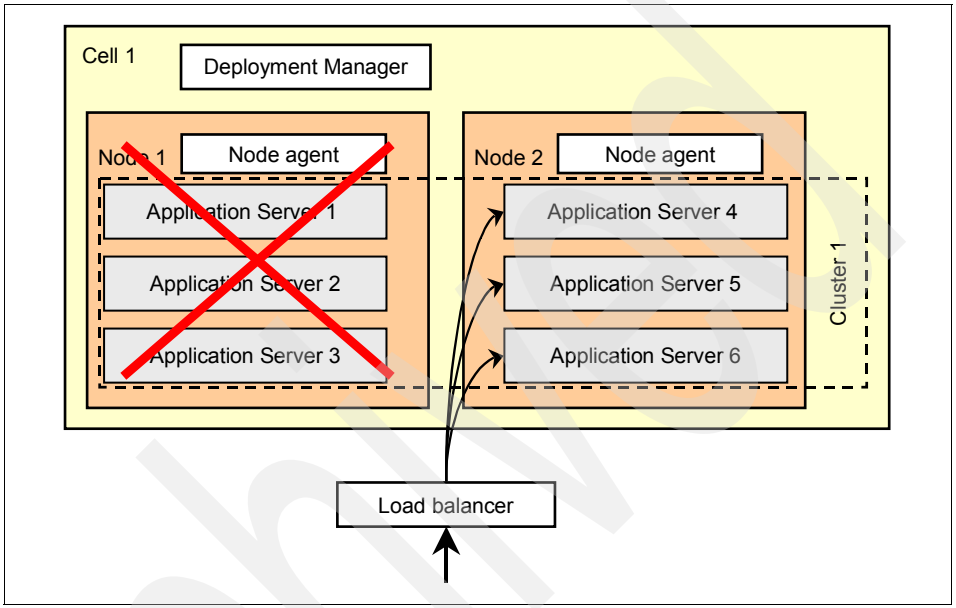


Figure 1-7 Failover in a clustered WebSphere environment

2

Sample business application scenario used in topologies

This chapter introduces the business scenario used in this IBM Redbooks publication. The scenario is a vehicle loan application system at the fictional ITSOBank.

This chapter contains the following sections:

2.1 Introduction

This publication uses a sample application for deployment on the various runtime topologies created. The application is a vehicle loan process used by a fictional organization called ITSOBank. The application collects and analyzes a loan applicant's information and provides a suitable loan customized for the applicant.

This section describes what the application does, its input data, and the required software to build and deploy it. It also describes how to obtain the project and EAR files for replication in your environment.

Typically, a monitor model would also be defined to track the vehicle loan process, although for simplicity we did not define a monitor model in this chapter.

2.1.1 The vehicle loan application

ITSOBank is a fictional financial institution in North America that offers a variety of products related to account management (deposits) and loan processing. In this book we consider the vehicle loan processing system with the following use cases:

- Apply loan.
- Verify customer.
- Perform credit check.
- Perform vehicle identification number (VIN) lookup.
- Calculate risk rating.
- Provide loan.

The actors participating in this system are:

- Customer

  Performs the financial transaction with the bank, which involves applying for a vehicle loan. ITSOBank customers are classified in three categories:

  - Premium
  - Regular
  - New

  Premium and regular customers are customers who already have established a relationship with the bank. They have either a loan account or a savings account.

- Loan officer

  Applies for loans on behalf of the customer.

- Credit verification provider

  An external service provider that verifies the creditworthiness of the customer.

- Loan provider

  An entity that finances the loan, typically ITSOBank. However, there are cases where other banks (lenders) express interest in providing loan facilities to the customer through ITSOBank. Such lenders are known as external loan providers.

Sequence of tasks in the process

The ITSOBank vehicle loan processing application involves the following steps:

1. The loan process is initiated when a customer's loan application is received.
2. The credit score of the customer requesting the loan is checked. The loan process uses a credit verification provider to obtain the credit score. The type of customer can be premium, regular, or new. This step invokes the service of a credit check service provider.

   Note: In the WebSphere Process Server version of this process the credit verification step is implemented as a human task using a Lotus Forms user interface.

3. A vehicle number verification is performed. This step is executed by using a vehicle identification number (VIN) lookup service.
4. The results from the vehicle verification are sent to the rating service to calculate the risk rating of a customer.

   Based on the customer's rating score and the customer type, the loan interest rates will differ. Premium customers with a low-risk rating receive the lowest rate of interest.

Sample input data

The sample application uses specific values to drive specific outcomes. The value 12345678901234567 indicates a *clean* VIN, meaning that there are no known peculiarities with the identified vehicle that would inhibit unencumbered transfer of ownership. Any other numeric value is considered *not clean*.

The calculation of the credit score is shown in Table 2-1.

Table 2-1 Customer type and credit score based on ID

Customer ID	Customer type based on customer ID	Credit score based on customer ID
100	Premium	800
200	Regular	650
Any other value	New	500

The interest applied to a loan is determined by a combination of the type of customer and her credit ranking, as shown in Table 2-2.

Table 2-2 Interest rate based on the rating score and customer type

Credit rating	Customer type	Annual rate of interest ($%)
Low	Premium	4.565
Low	Regular or new	6.850
Medium	Any (premium, regular, or new	8.585
High	Any (premium, regular, or new	10.545

2.1.2 Prerequisite software and downloads

The following WebSphere BPM products were used in the development of the vehicle loan application:

- Business Space powered by WebSphere V6.2
- WebSphere Business Modeler Advanced V6.2
- WebSphere Integration Developer V6.2 and Business Services Composition Studio V6.2

After development, the application can be deployed to either of two runtime environments. The production topology demonstrated in this publication shows deployment to either of the following:

- WebSphere Process Server V6.2
- WebSphere Business Services Fabric V6.2

Chapter 12, “Monitoring a production topology” on page 341, discusses technical monitoring of the vehicle loan application using IBM Tivoli Composite Application Manager (ITCAM) for SOA. It also provides a brief introduction to the capabilities of WebSphere Business Services Fabric’s Performance Manager.

You can also extend your topology to include WebSphere Business Monitor V6.2 for real-time monitoring of business events for the vehicle loan application, as discussed in Chapter 15, “Incorporating WebSphere Business Monitor into a production topology” on page 433.

To download the vehicle loan application project and Enterprise Archive (EAR) files, refer to the instructions in Appendix A, “Additional material” on page 597. The directory named \Scenarios contains two subdirectories, one for each runtime. The EAR subdirectory contains a deployable version of the vehicle loan process for the appropriate runtime environment.

2.2 Implementation of the vehicle loan process

This section describes the development methodology used to create the vehicle loan application and how it is implemented in the various runtimes. The development life cycle begins with the definition of business intent in Business Space. The process is modeled with WebSphere Business Modeler, developed in WebSphere Integration Developer, and deployed to either WebSphere Process Server or WebSphere Business Services Fabric. WebSphere Business Monitor is used to monitor Key Performance Indicators (KPIs) and extract business metrics, which are available for display from either WebSphere Business Monitor or Business Space. The WebSphere Business Services Fabric’s Performance Manager provides summary metrics on service endpoint invocation for business services. This section outlines the sequence of steps used in each of these stages and illustrates some of the views and diagrams from the corresponding products. The following topics are covered:

- Overview of the development life cycle
- The ITSOBank use case
- Vehicle loan process with Business Space
- Vehicle loan process with WebSphere Business Modeler
- Vehicle loan process with WebSphere Business Services Fabric
- Vehicle loan process with WebSphere Process Server

2.2.1 Overview of the development life cycle

The design and construction of the vehicle loan process involves multiple products from the WebSphere BPM product suite, as shown in Figure 2-1.

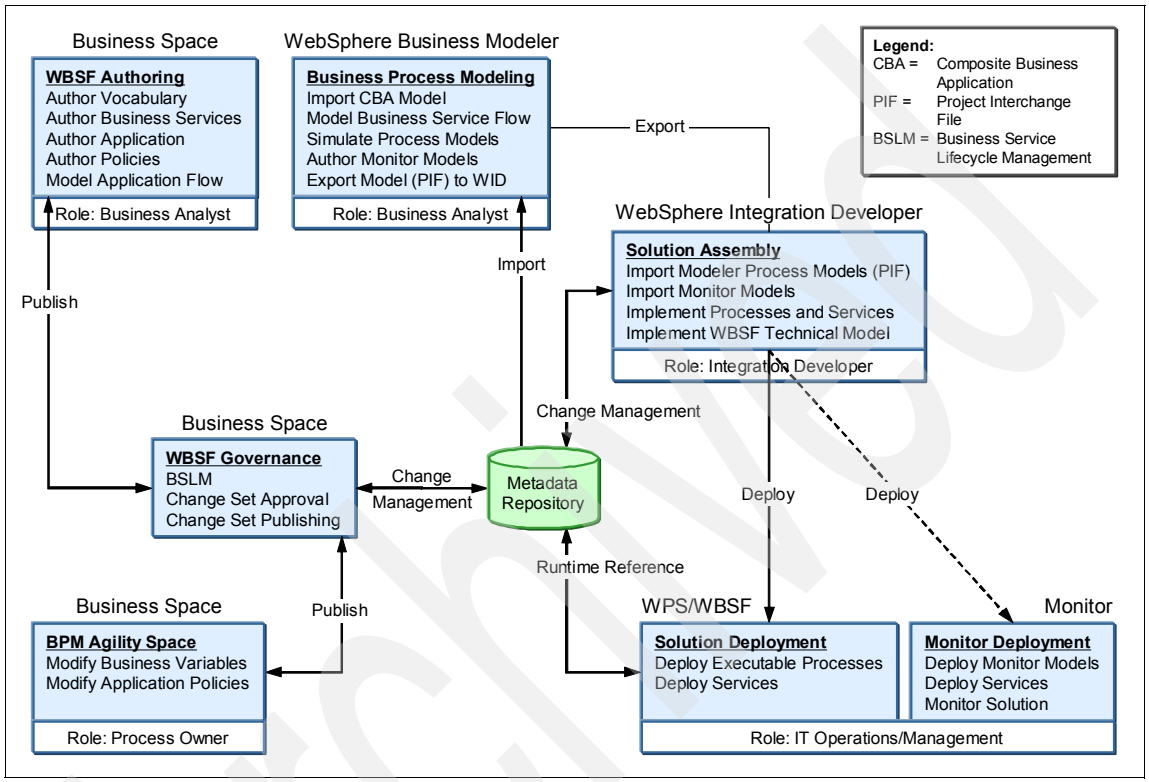


Figure 2-1 Product interactions in the development life cycle

A business leader or business analyst uses the WebSphere Business Services Fabric authoring template in Business Space to create composite business applications (CBAs), model application flows, author business vocabulary and business services, and to apply governance. These artifacts are stored in a common metadata repository, the WebSphere Process Server runtime database.

A business process modeler then points the WebSphere Business Modeler tool to the repository to import artifacts created in the Business Space. After further refinement the process modeler hands off the business process models to the integration developer, who uses WebSphere Integration Developer, who adds execution semantics to create technical models that can be deployed on WebSphere Business Services Fabric, WebSphere Process Server, and WebSphere Business Monitor.

Once the application is deployed the business analyst can use Business Space for a variety of tasks, such as viewing Key Performance Indicators dashboards, managing processes and human tasks, and adjusting business policies.

2.2.2 Defining the loan process using Business Space

The business intent for the vehicle loan process is initially expressed in Business Space by following this procedure:

1. The business analyst or business leader logs into Business Space and creates a new WebSphere Business Services Fabric Authoring space.
2. The analyst opens the vocabulary space to define a business vocabulary for the ITSOBank. The vocabulary comprises channels, roles, and business concepts, as shown in Figure 2-2.

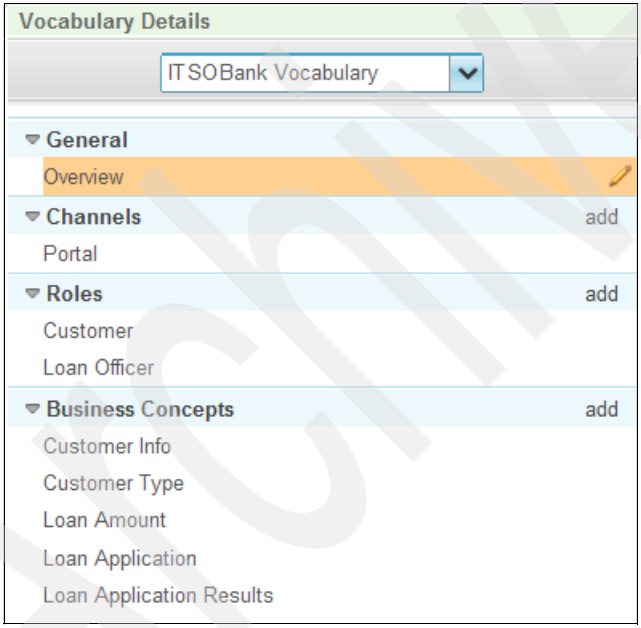


Figure 2-2 ITSOBank Vocabulary

3. In the Business Service space the analyst adds business services and policies required to support the vehicle loan process.

4. The ITSOBank loan application is then defined in the application space. Here the analyst models the application flow as illustrated in Figure 2-3.

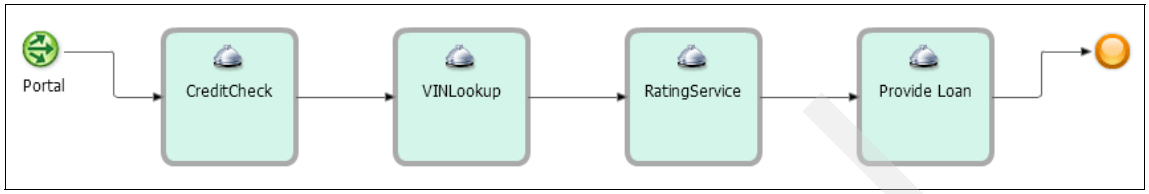


Figure 2-3 Conceptual application flow for the vehicle loan process

After adding business services and application policies for the ITSOBank loan application, application details will look as shown in Figure 2-4.

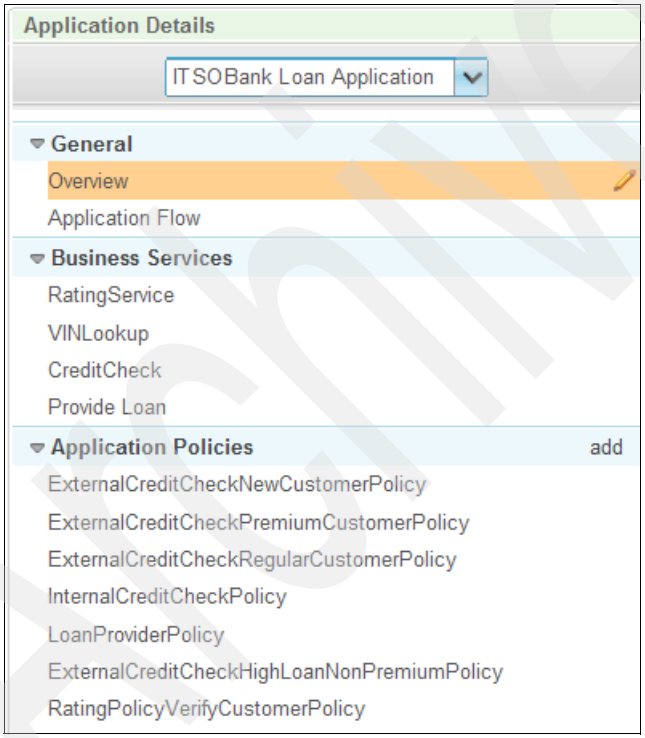


Figure 2-4 Business Space application setup for the ITSOBank loan process

5. The business analyst saves her work into the master repository by bringing up the governance space to submit her change set for approval, as shown in Figure 2-5.

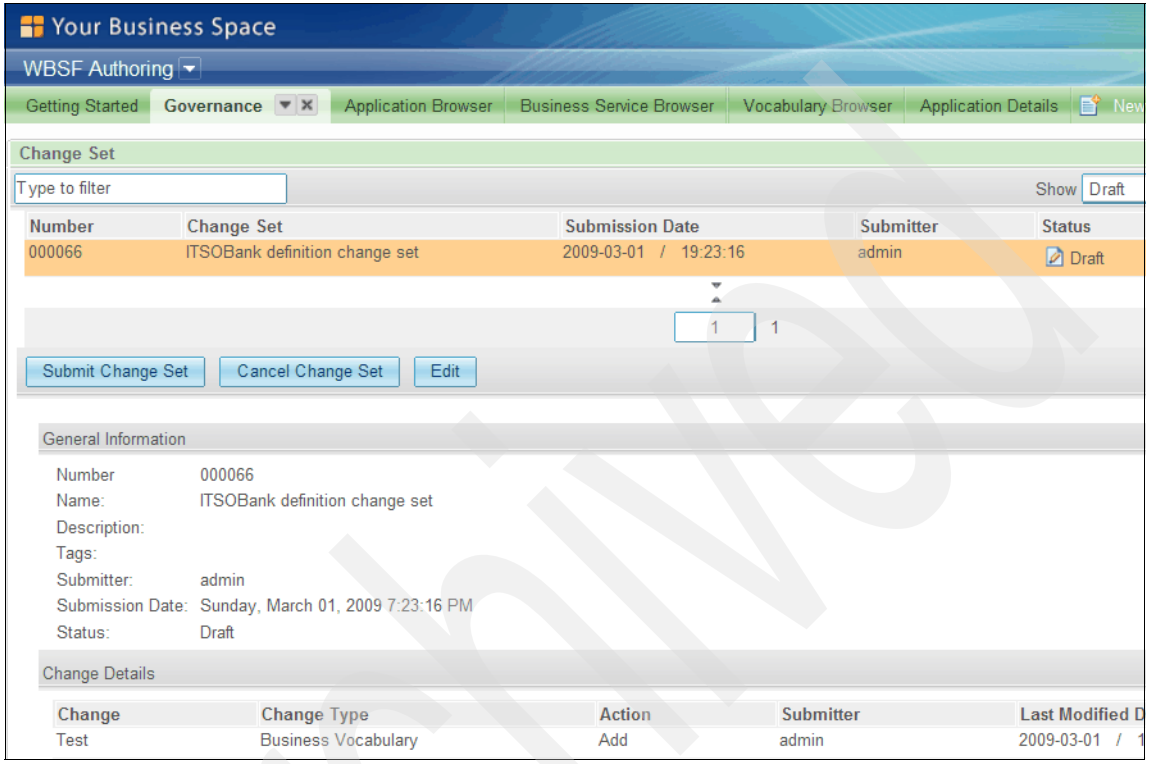


Figure 2-5 Submitting changes to master repository

6. Upon verification of changes submitted by the business analyst, the WebSphere Business Services Fabric administrator approves and publishes the change set to the master repository.

2.2.3 Modeling the loan process using WebSphere Business Modeler

Once a business analyst has completed the application definition work in Business Space, a business analyst with process modeling responsibilities imports application metadata into WebSphere Business Modeler. Application flows and business services are further refined to incorporate data flows, key performance indicators, and business measures, among other facets of process modeling. The analyst also adds monitoring hints into the model that will be used by WebSphere Business Monitor. The model is then available to the integration developer for technical implementation in WebSphere Integration Developer.

2.2.4 Developing and deploying the WebSphere Business Services Fabric runtime

ITSOBank has implemented the vehicle loan process for WebSphere Business Services Fabric. WebSphere Business Services Fabric introduces the concept of a *business service*. A business service represents a business function whose behavior can be adapted at run time. A business service is based on the operating context of the request and the policies established to meet the service consumer's need.

In order to implement the vehicle loan process with WebSphere Business Services Fabric, the integration developer will use the Business Services Composition Studio (in WebSphere Integration Developer) to describe and create the vehicle loan process. In this version of the vehicle loan process, the dynamic assembly capabilities of WebSphere Business Services Fabric are used. This dynamic assembly capability enables ITSOBank to extract points of variability in the vehicle loan process (in this case, determining the loan provider to use). This creates a linear process where we are able to apply assertions and business policies at runtime, giving the flexibility that is needed by business.

The vehicle loan process with WebSphere Business Services Fabric is shown in Figure 2-6.

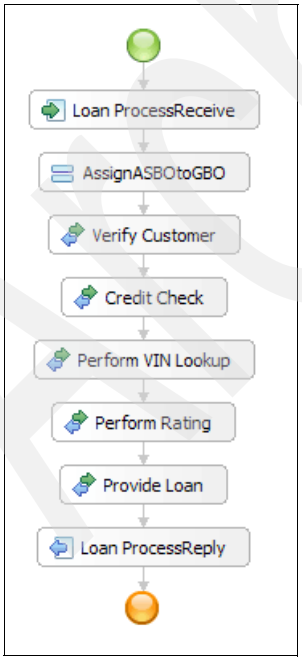


Figure 2-6 The vehicle loan process model in WebSphere Business Services Fabric

In this version of the process all significant points of variability have been extracted and modeled as WebSphere Business Services Fabric policies. The result is an optimized straight-through process implemented as a micro-flow.

2.2.5 Developing and deploying WebSphere Process Server runtime

The WebSphere BPM steps for the ITSOBank vehicle loan process are:

1. A business analyst defines the process model by using WebSphere Business Modeler to analyze, simulate, model, and define business measures (key performance indicators and metrics) for the vehicle loan process.

   The business analyst uses a process diagram to compose the process flow visually. A process diagram is a graphical representation of a business process flow, consisting of activities and the connections between these activities.

2. The model generated by WebSphere Business Modeler is imported into WebSphere Integration Developer as a set of Business Process Execution Language (WS-BPEL) artifacts for further processing.

3. The integration developer (technical person) works with WebSphere Integration Developer to implement the vehicle loan process. With WebSphere Integration Developer, the developer assembles an integrated application for the vehicle loan process model, using reusable service components (such as verify customer, credit check, and VIN lookup). These components are shown in the Service Component Architecture (SCA) assembly diagram in Figure 2-7.

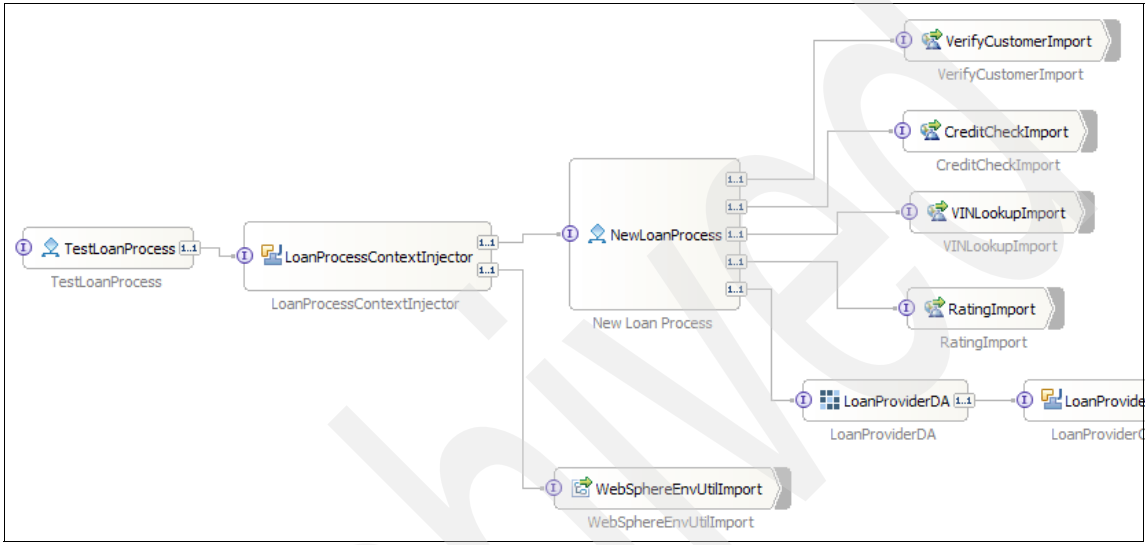


Figure 2-7 SCA assembly diagram of the loan application

4. The integration developer visually composes how the vehicle loan process should execute these reusable service components in a process flow.
5. The vehicle loan process application is assembled and packaged into an enterprise archive file for deployment to WebSphere Process Server.

Alternatively, modules exported from WebSphere Business Monitor can be directly deployed to WebSphere Process Server without using WebSphere Integration Developer. For more information about this approach see:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp?topic=/com.ibm.d2dscenario.doc/doc/wps/tmde_impmde.html

2.3 Deploying and testing the vehicle loan application

The completed vehicle loan application is deployed to WebSphere Business Services Fabric.

In 14.4, “Deploying and testing the vehicle loan application” on page 426, we provide instructions for deploying this sample application to WebSphere Business Services Fabric.

3

Business Process Management production topologies

This chapter provides an introduction to the WebSphere Process Server components and to topology patterns. This chapter presents the four WebSphere Process Server deployment environment topology patterns included in the administrative console and in the profile management tool:

- Single Cluster topology (or bronze topology)
- Remote Messaging topology (or silver topology)
- Remote Messaging and Remote Support (or gold, or ND7 topology)
- Custom topology

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

The WebSphere Process Server topology can be extended to include other WebSphere Business Process Management (BPM) products such as WebSphere Business Services Fabric and WebSphere Business Monitor. These topologies are also introduced in this chapter.

3.1 Introduction

A WebSphere Process Server topology is the physical layout of the deployment environment required to meet your business needs for capacity, availability, and scalability. A key aspect of the WebSphere Process Server topology design involves the number of physical machines (in distributed environments), the number of servers on those machines, and 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 non-WebSphere Process Server supporting resources such as a user registry (for security), one or more HTTP servers (for Web content), necessary firewalls, load balancers, and so forth.

You should carefully plan any WebSphere Process Server 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

To make the topology design and implementation process easier, WebSphere Process Server V6.2 includes a set of deployment environment patterns that represent the most common production topologies.

The deployment patterns offer a repeatable, automated method of 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 V6.2. Whether you perform a manual install or use the deployment topology patterns, there are a number of different components to consider in creating the topology.

3.2 WebSphere Process Server components

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

- “Databases” on page 39
- “Service integration buses” on page 39
- “Business Process Choreographer” on page 40
- “WebSphere Process Server applications” on page 40

3.2.1 Databases

WebSphere Process Server uses multiple databases to hold, store, and track information. WebSphere Process Server makes use of the following databases:

- Common database (WPRCSDB)

  This database is used as a repository for various components in WebSphere Process Server. It must be created prior to starting WebSphere Process Server. The common database contains information regarding the components:

  - Application Scheduler
  - Business Rules
  - Mediations
  - Recovery
  - Relationships
  - Selectors

- 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.

- Business Process Observer database (OBSVRDB)

  This database is used by the BPC Observer application to store event information from the CEI bus in an event collector table.

- Messaging engine database (MEDB)

  This database is used by the Service Component Architecture (SCA) system and application buses, the CEI bus, and the Business Process Choreographer bus.

- Event database (EVENT)

  This database contains information regarding the Event Service, such as Common Based Events.

In addition, both WebSphere Business Services Fabric and WebSphere Business Monitor have their own databases.

3.2.2 Service integration buses

A service integration bus is a managed communication mechanism that supports service integration through synchronous and asynchronous messaging. A bus consists of interconnecting messaging engines. A messaging engine is a component, running inside a server, that manages messaging resources for members of the bus. Applications are connected to a messaging engine when

they connect to a bus. WebSphere Process Server makes use of the following service integration buses:

- SCA system bus

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

- SCA application bus

  This bus supports the asynchronous communication between WebSphere Business Integration Adapters and other SCA components.

- Common Event Infrastructure bus

  This bus is used to transmit common base events asynchronously to a Common Event Infrastructure (CEI) server.

- Business Process Choreographer bus

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

3.2.3 Business Process Choreographer

Business Process Choreographer (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

  This component executes and manages business processes written in BPEL.

- Human Task Manager

  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.

3.2.4 WebSphere Process Server applications

WebSphere Process Server provides a variety of Web-based application tools.

- BPC Tools

  BPC Explorer is a Web application for managing the life cycle of business processes and human tasks. BPC Explorer can be used to repair business processes, update custom properties for processes, and manage work assignments for human tasks. Optionally, it can generate reports and display statistical data about the execution of business processes and human tasks.

> **Note:** In the 6.2 release, BPC Explorer houses the function of the BPC Observer. This is now referred to as BPC tools.

- Business rules manager (BRM)

  The BRM assists business analysts in browsing and modifying business rule values. Using templates, an analyst can edit business rule values, create new rules, create new conditions within a decision table, and publish changes to business rule definitions at run time.

In addition to these WebSphere Process Server-specific applications, Business Space powered by WebSphere can be used to interact with WebSphere Process Server. Business Space is a browser-based, graphical interface included in WebSphere Process Server that allows application users to create, manage, and integrate Web interfaces across the BPM Suite.

3.2.5 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, such as event generation, transmission, persistence, and consumption. Events can be published to the CEI server for possible distribution to JMS queues and topics. If enabled, events might also be persisted to a database.

Typically, WebSphere Process Server uses CEI to get business event information to WebSphere Business Monitor.

3.3 WebSphere Process Server deployment environment patterns

A WebSphere Process Server deployment environment can easily be created using the IBM-supplied deployment environment patterns. The deployment environment patterns included in the administrative console and the profile management tool represent the most common deployment environments used in production. Each pattern centers around the number of WebSphere Process Server clusters and cluster members and how these clusters are grouped and allocated across nodes.

Any WebSphere Process Server deployment contains three 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 upon your needs. The three sets of functions in the WebSphere Process Server environment are:

- Application deployment target

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

- Supporting infrastructure

  The supporting infrastructure includes the CEI and other infrastructure services used to support your environment, such as the Business Process Choreographer Explorer, Business Rules Manager, and Business Spaces.

- 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 WebSphere Process Server components (for example, the internal navigation queues used by long-running business processes).

Each of the provided deployment environment patterns creates a different number of clusters to support the required functions. The deployment environment patterns included in WebSphere Process Server V6.2 are:

- Single Cluster (bronze)

  In this pattern, the messaging infrastructure, the application deployment target, and the support functions are contained in a single cluster (named AppTarget). This pattern is discussed in 3.3.1, “Single Cluster topology pattern” on page 44.

- Remote Messaging (silver)

  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 (named AppTarget)
  - One for the messaging infrastructure (named Messaging)

  This pattern is discussed in 3.3.2, “Remote Messaging topology pattern” on page 47.

- Remote Messaging and Remote Support (gold)

  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 pattern is discussed in 3.3.3, “Remote Messaging and Remote Support topology pattern” on page 49.

- Custom deployment environments

  If none of the deployment environment patterns meets your requirements, you may create a custom deployment environment. This pattern is discussed in 3.3.4, “Custom topology patterns” on page 51.

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, imported, and re-used to create the topology on any number of systems. For example, you may wish 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 deployment environment:

- Create the deployment environment when you install the software, using the installation wizard or silent installation.
- Install the software on the host systems. Use the Profile Management Tool or **manageprofiles** command to create the deployment environment.
- Install the software on the host systems. Use the Profile Management Tool or **manageprofiles** command to create deployment manager and custom profiles. Create the deployment environment using the Integrated Solutions Console of the deployment manager.
- Install the software on the host systems. Use the Profile Management Tool or **manageprofiles** command to create deployment manager and custom profiles. Create the deployment environment using the **wsadmin** utility.

The third method was used to create the topology used in the lab environment for this publication. Regardless of which method you use to create the deployment environment, you can still manage some 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) will not be reflected in the deployment environment descriptor.

3.3.1 Single Cluster topology pattern

The Single Cluster topology pattern, also known as the bronze topology, 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, and demonstration environments.

Creating this deployment environment is described in detail in Chapter 6, “Configuring a Single Cluster topology” on page 135.

A Single Cluster topology sample configuration for WebSphere Process Server is shown in Figure 3-1.

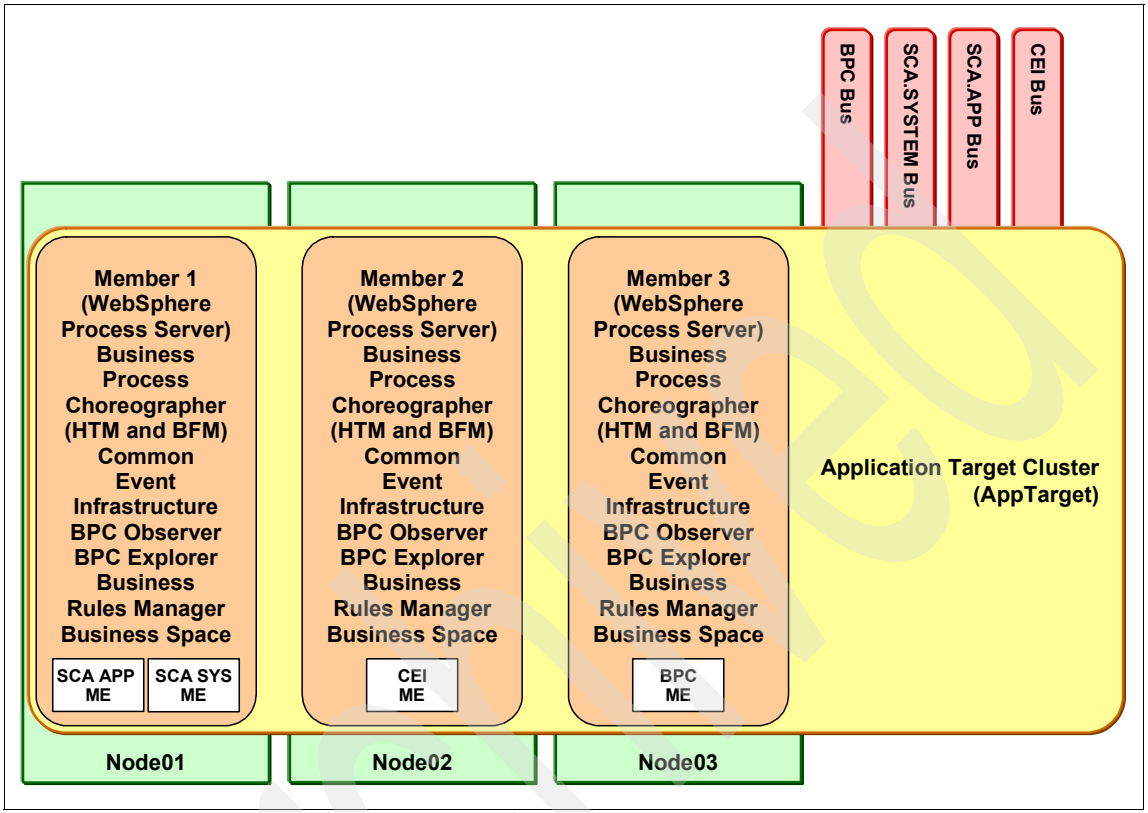


Figure 3-1 Example Single Cluster topology

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 3-1 on page 45, the messaging engines are split 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.

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 needed 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.

The fact that messaging engines read and write locally also creates a unique set of issues if you attempt to partition the destinations. When you create more than one active set of messaging engines, partitioning results. 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 server. However, this configuration can create issues for your applications.

If you partition destinations when the applications and messaging engines are in the same cluster, you will no longer have the ability to maintain message order. This is true even if you attempt to enable event sequencing in WebSphere Process Server. Partitioned destinations can create unpredictable behavior if one or more messaging engines fails in a Single Cluster topology. If you are prepared to endure possible unpredictable behavior and the loss of message order, partitioning the destinations in a Single Cluster topology may be acceptable. However, this configuration is 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

3.3.2 Remote Messaging topology pattern

The Remote Messaging topology pattern, also known as the silver topology, 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.

A Remote Messaging sample topology is shown in Figure 3-2.

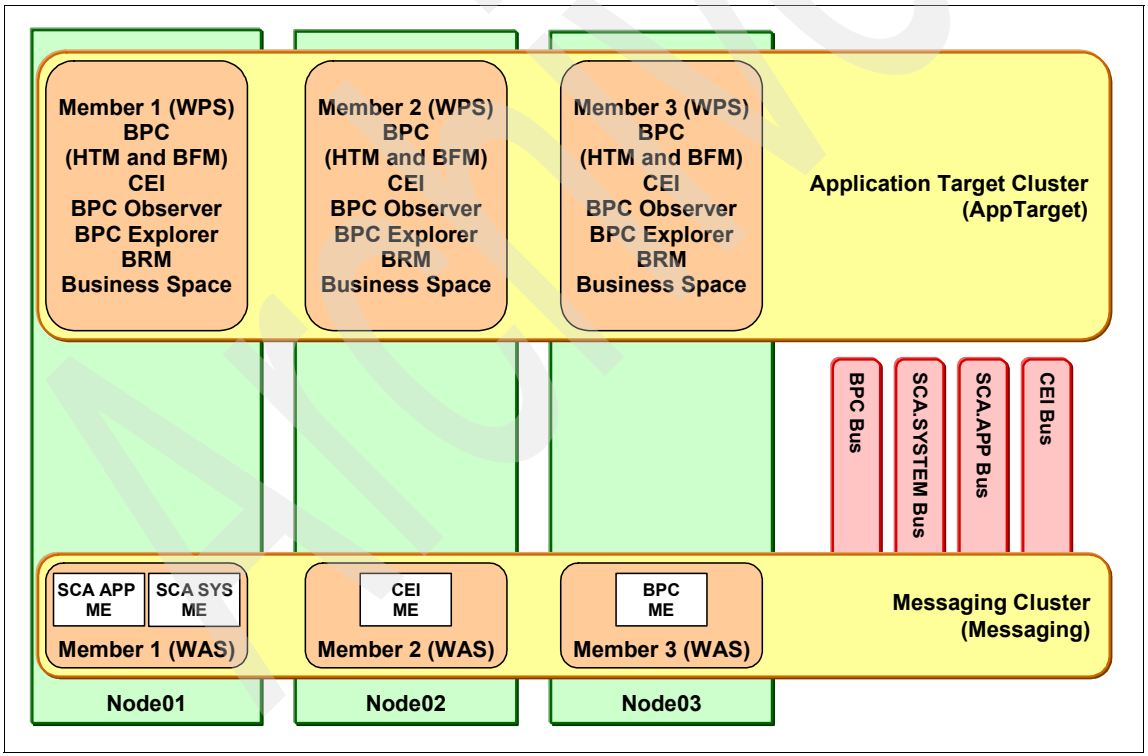


Figure 3-2 Example Remote Messaging topology

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.
- The Messaging cluster is a member of all four of the required WebSphere Process Server buses:
  - 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 3-2 on page 47, the messaging engines are split 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 four of the messaging engines, and the server that starts first will automatically run all four of the engines.

Note that the behavior of the messaging engines in a Remote Messaging topology is different from the 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.

If a single bus has more than one 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.

Thus, partitioned destinations are strongly discouraged in a remote messaging scenario.

3.3.3 Remote Messaging and Remote Support topology pattern

The Remote Messaging and Remote Support topology pattern, also known as the gold topology, is the preferred topology for production environments. This topology provides three separate clusters:

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

Creating this deployment environment using the Remote Messaging and Remote Support pattern is described in detail in Chapter 7, “Configuring Remote Messaging and Remote Support” on page 165. A Remote Messaging and Remote Support sample topology is shown in Figure 3-3.

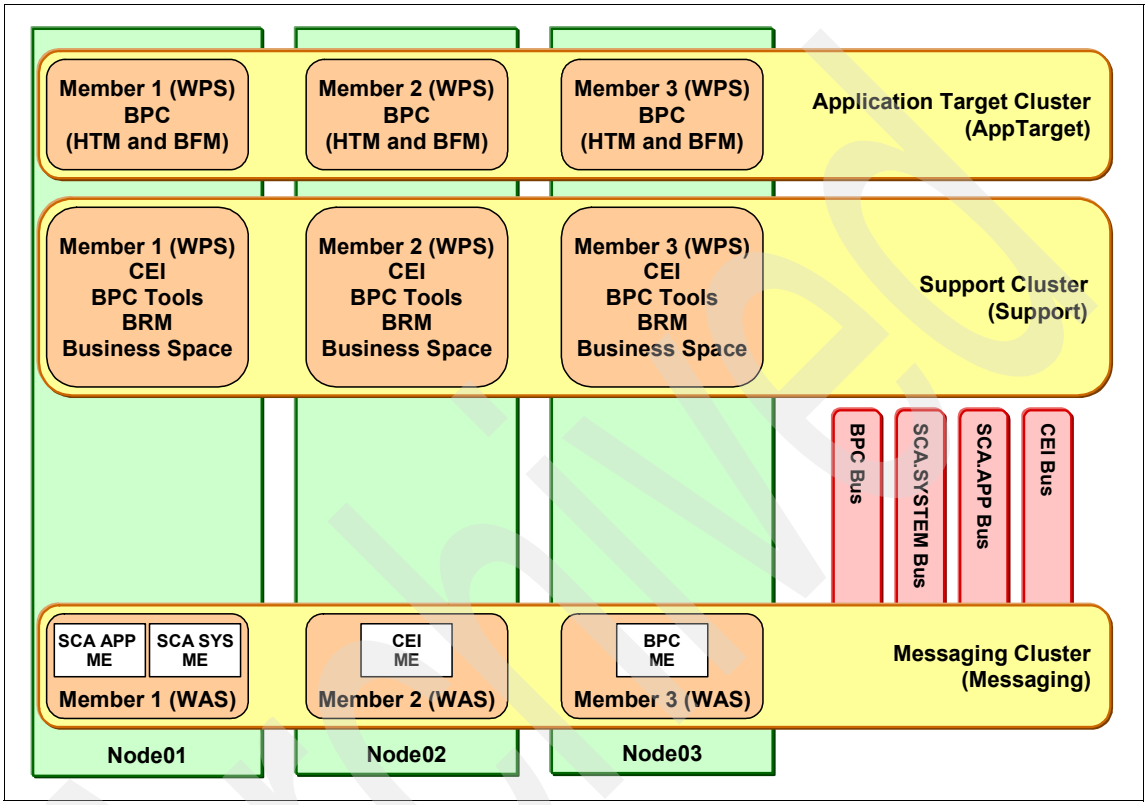


Figure 3-3 Example Remote Messaging and Remote Support topology

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

- 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 3-3 on page 50.
  - 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.

You should note that the behavior of the messaging engines in a Remote Messaging and Remote Support topology is identical to the behavior discussed in the Remote Messaging topology description. 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 the message order. This is true even if you attempt to enable event sequencing in WebSphere Process Server.

Partitioned destinations can create additional issues when the messaging engines are remote. 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. Thus, partitioned destinations are strongly discouraged in a Remote Messaging and Remote Support scenario.

3.3.4 Custom topology patterns

If none of the three default deployment environment patterns are suitable to your needs, you can create a custom topology. Also, you can use the administrative console 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. Using custom topologies is discussed in Chapter 8, “Configuring a custom topology” on page 199. 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 wish 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 wish 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 wish 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 Observer 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. Chapter 11, “Advanced production topologies” on page 269, describes how to manually extend the Remote Messaging and Remote Support topology. The principles discussed in that chapter also apply to the creation of custom topology patterns.

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.

3.4 Four Cluster topology pattern

The Four Cluster topology pattern is the recommended topology for WebSphere Dynamic Process Edition production environments. It is also the preferred topology for environments where WebSphere Process Server and WebSphere Business Monitor coexist. This topology provides four separate clusters:

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

> **Note:** There is no deployment environment wizard support for this pattern in WebSphere Dynamic Process Edition 6.2. You must create this pattern manually.

This topology is described in Chapter 16, “Creating a WebSphere Dynamic Process Edition production topology” on page 509.

A Four Cluster sample topology is shown in Figure 3-4. This sample incorporates WebSphere Process Server, WebSphere Business Services Fabric, and WebSphere Business Monitor into one cell.

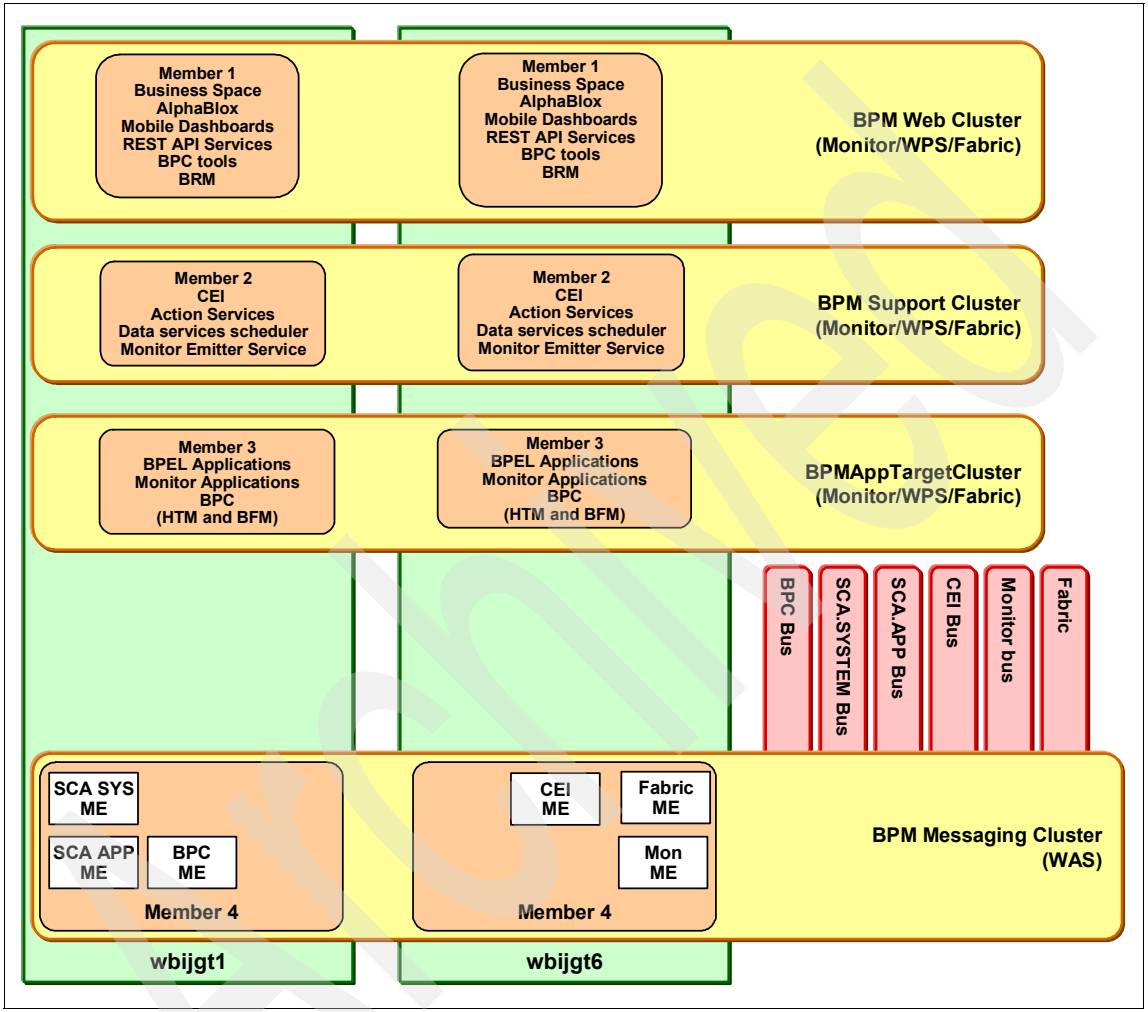


Figure 3-4 Four Cluster topology

Note the following aspects of this example:

- The application cluster will contain:
  - BPC and (HTM and BFM)
  - BPEL applications
  - Monitor model (moderator/application logic)
- The support cluster will contain:
  - CEI
  - Action services
  - Data services scheduler
  - Monitor Event Emitter service
- The Web cluster will contain:
  - Business SpaceREST API Services
  - BPC tools
  - Business Rules Manager
  - Monitor widgets
  - Monitor Alphablox widgets
  - Monitor mobile dashboards
  - Fabric widgets
  - WPS widgets
  - Alphablox
  - Fabric EARs
- The messaging cluster will have the required buses:
  - SCA.SYSTEM bus
  - SCA.APPLICATION bus
  - CEI bus
  - BPC bus
  - Monitor bus
  - Fabric bus

3.5 Selecting an appropriate topology

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

- Available hardware resources
- 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

In general, the Remote Messaging and Remote Support topology pattern is the most suitable production topology, but the choice ultimately depends upon your unique, individual requirements.

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

3.5.1 Single Cluster topology pattern

A Single Cluster topology is ideal for limited hardware. Because all of the components are installed in the same cluster, fewer physical machines are required. Because each server instance must run the supporting applications and your integration applications, however, the memory requirements for the individual Java Virtual Machines (JVMs) are much greater. In addition, one or more members of the cluster must also run the messaging engines required for asynchronous interactions. Thus, Single Cluster topologies are typically used for proof of concept, development, and testing environments.

Combining all aspects of the WebSphere Process Server 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 components is easier to administer. Instead of 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. Thus, 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 some additional administrative effort in creating and maintaining the policies.

3.5.2 Remote Messaging topology pattern

For environments that must support numerous human tasks, long-running business processes, state machines, and asynchronous interactions, a Remote

Messaging topology has advantages over the Single Cluster topology. Separating the messaging infrastructure into a separate cluster removes the messaging overhead from the application target cluster. This reduces the memory requirements for the application target cluster members. This topology also differs from the Single Cluster topology in terms of the hardware required. Because there are now two clusters with multiple cluster members, the hardware requirements are greater for distributed environments.

From an administrative perspective, the requirements of the Remote Messaging topology are greater than those of the Single Cluster topology. Additional clusters and additional cluster members increase the administrative effort required. In addition, distributing the messaging engines across the members of the messaging cluster requires the creation and maintenance of policies.

In the Remote Messaging topology, the supporting applications and the CEI components are still part of the application target cluster. Thus, for environments that make extensive use of CEI, the Remote Messaging topology may not be ideal either. For small to medium-sized businesses, or for businesses without extensive monitoring or auditing requirements, this topology is generally suitable.

The scalability options for the Remote Messaging 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 maximum of three servers (the SCA.SYSTEM and SCA.APPLICATION engines should be active on the same server). Thus, adding more than three cluster members to the messaging cluster has no effect on the processing capability of the messaging infrastructure. Scaling the application target cluster is relatively easy. If you need additional processing capability for your applications or for the supporting infrastructure, you can simply add additional nodes and members to the application target cluster.

3.5.3 Remote Messaging and Remote Support topology pattern

For the vast majority of customers (especially those with large computing infrastructures), the Remote Messaging and Remote Support topology is the preferred environment. The hardware requirements for distributed platforms are more intensive, but having three (or more) clusters with multiple members performing specific functions allows you greater flexibility in adjusting and tuning memory 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 provide 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. Because expanding the messaging infrastructure beyond three cluster members has no affect on processing capability, the scalability limitations present in the Remote Messaging policy also apply to the Remote Messaging and Remote Support topology.

As with the Remote Messaging topology, the Remote Messaging and Remote Support topology provide an ideal environment for long-running business processes, state machines, human tasks, and asynchronous interactions (including JMS and MQ/JMS bindings). Because 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 (including environments with WebSphere Business Monitor). 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.

3.5.4 Custom topology

By allowing you to define your own environment, the custom topology is by far the most flexible. As mentioned previously, the supplied topology patterns (Single Cluster, Remote Messaging, and Remote Messaging and Remote Support), deploy all of the WebSphere Process Server components to their default locations. You may or may not need the additional overhead associated with these components. For example, if your organization has no need for the CEI, you could create a custom topology that removes CEI support and the BPC Observer from your environment. Similarly, if your organization has governance rules that prevent you from taking advantage of the Business Rules Manager, you could remove it from your deployment.

Aside from giving you the ability to precisely control the individual components deployed in your environment, the advantages of custom topologies are similar to

those in the Remote Messaging and Remote Support topology. The disadvantages are also similar.

3.5.5 Four Cluster topology

The Four Cluster topology is the recommended starting topology when you are starting with WebSphere Dynamic Process Edition. WebSphere Dynamic Process Edition make 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
- AlphaBlox® for dimensional view of data

Aside from giving you the ability to precisely control the individual components deployed in your environment, the advantages of the Four Cluster topology are similar to those in the Remote Messaging and Remote Support topology. The disadvantages are also similar but with one limitation: This topology cannot be built using the Deployment Environment Topology wizard.

3.5.6 Condensed topology selection criteria

Consider the information listed in Table 3-1, which is a quick guide to selecting your production topology. This table provides a condensed list of the advantages and disadvantages of each of the topology patterns.

Table 3-1 Considerations for selecting a topology

Consideration	Single Cluster topology	Remote Messaging topology	Remote Messaging and Remote Support topology	Four Cluster topology
Number of clusters to maintain	One cluster for all components	One cluster for applications and for the support infrastructure One cluster for messaging	One cluster for applications One cluster for the support infrastructure One cluster for messaging	One cluster for applications One cluster for Web interfaces One cluster for support infrastructure One cluster for messaging

Consideration	Single Cluster topology	Remote Messaging topology	Remote Messaging and Remote Support topology	Four Cluster topology
Hardware requirements	Can be implemented on limited hardware	More hardware required for distributed environments	More hardware required for distributed environments	Most hardware intensive
Asynchronous interactions	Use should be minimal	Use must be balanced against resource availability	Ideal environment for asynchronous interactions	Ideal environment for asynchronous interactions
Long-running processes, state machines, and human tasks	Use should be minimal	Use must be balanced against resource availability	Ideal environment for interruptible processes, state machines, and human tasks	Ideal environment for interruptible processes, state machines, and human tasks
Heavy CEI activity	Not recommended (Light CEI use should be balanced against resource usage.)	Not recommended (Light CEI use should be balanced against resource usage.)	Ideal environment for heavy CEI use	Ideal environment for heavy CEI use
Administrative burden	Relatively small	Requires additional effort	Requires additional administrative effort	Requires most administrative effort
Scalability	All components scaled at the same rate	Messaging cluster scalability limited (no benefit beyond three servers) All other components scaled at the same rate	Easy to scale All functions separated Messaging cluster scalability still limited (no benefit beyond three servers)	Easiest to scale All functions separated Messaging cluster scalability still limited (benefit comes when other BPM products are introduced)

3.6 Incorporating other products into a Remote Messaging and Remote Support topology

In addition to the provided topology patterns and custom topologies, WebSphere Process Server production topologies can also include other WebSphere products. This book includes information about configuring WebSphere Business

Services Fabric and WebSphere Business Monitor in the Remote Messaging and Remote Support topology pattern.

If you intend to include WebSphere Business Services Fabric and WebSphere Business Monitor in your production environment, we suggest that, for performance reasons, you use the Remote Messaging and Remote Support topology. If you choose to include other business integration products in a Single Cluster or Remote Messaging topology, you should carefully consider the impact.

3.6.1 Adding WebSphere Business Services Fabric

You can add WebSphere Business Services Fabric to a WebSphere Process Server production cell. When WebSphere Business Services Fabric is added to the Remote Messaging and Remote Support topology, the messaging cluster is a member of the WebSphere Business Services Fabric bus, and the WebSphere Business Services Fabric core applications are added to the application target cluster. This topology is represented in Figure 3-5.

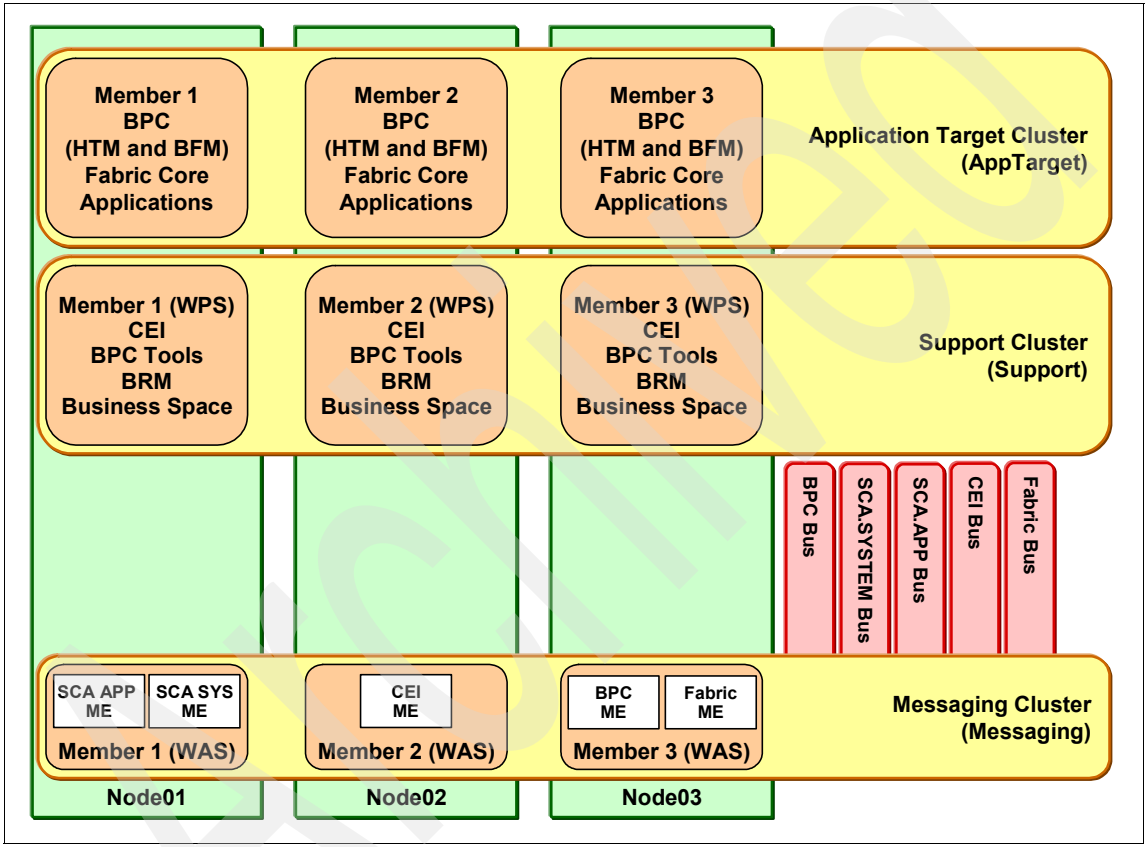


Figure 3-5 WebSphere Business Services Fabric deployment scenario

Chapter 14, “Incorporating WebSphere Business Services Fabric into a Remote Messaging and Remote Support topology” on page 385, provides detailed instructions on how to add WebSphere Business Services Fabric to the WebSphere Process Server Remote Messaging and Remote Support topology.

3.6.2 Adding WebSphere Business Monitor

If you are adding WebSphere Business Monitor into your production topology, the recommended practice is to deploy WebSphere Process Server and WebSphere Business Monitor in the same cell. Doing so allows both products to share the CEI (also referred to as local CEI). There are two options for doing this:

- Extend WebSphere Business Monitor onto an existing WebSphere Process Server Remote Messaging and Remote Support topology. This topology is represented in Figure 3-6.

  For guidance on creating this topology, see Chapter 15, “Incorporating WebSphere Business Monitor into a production topology” on page 433.

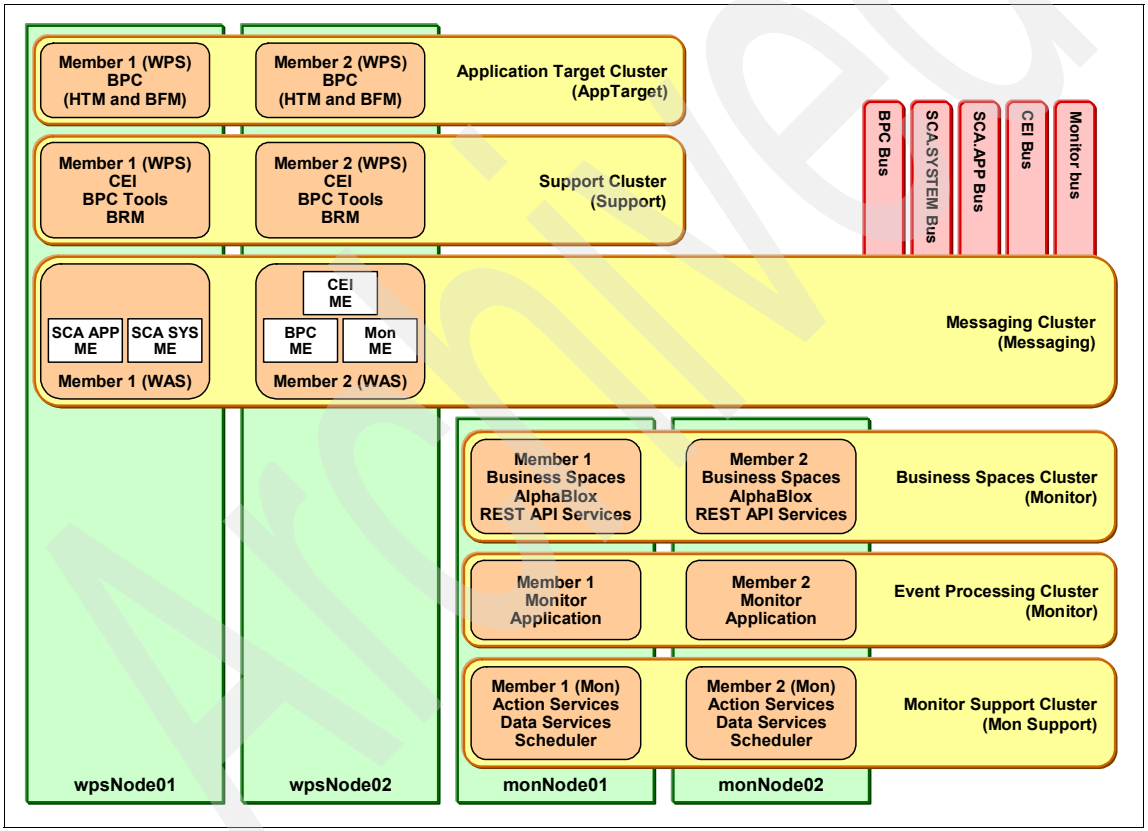


Figure 3-6 WebSphere Business Monitor deployment scenario

- Create a Four Cluster topology, as described in 3.4, “Four Cluster topology pattern” on page 53. This is the recommended approach for newly created production topologies. For guidance on creating this topology, see

Chapter 16, “Creating a WebSphere Dynamic Process Edition production topology” on page 509.

Regardless of the topology that you select, there are multiple ways in which to scale WebSphere Business Monitor as your needs grow:

- Add an additional cluster to which monitor models could be deployed.
- Add additional members to the cluster where Monitor models are deployed. Furthermore, if you define core groups, you could pin specific monitor models to particular members. For failover purposes, you can specify another member.

4

Security considerations for BPM

This chapter addresses security considerations when building a Business Process Management (BPM) solution using WebSphere Process Server, WebSphere Business Monitor, and WebSphere Business Services Fabric. It contains the following sections:

- Security overview
- Security in WebSphere Application Server
- Security for a WebSphere Process Server solution
- Access control for WebSphere Business Services Fabric
- Access control for WebSphere Business Monitor
- Additional security considerations

4.1 Security overview

Before discussing the security features of WebSphere Application Server, there are some core concepts that should be defined carefully. In this section we review security fundamentals to be able to make the correct decision in choosing the suitable security mechanism.

4.1.1 Authentication and authorization

Authentication is the methodology of identifying users or entities for the purposes of preventing impersonation. Authentication techniques include user IDs, passwords, digital certificates, and private keys.

Authorization is the methodology of selectively granting or denying users access to resources. The resources could be roles, access control lists, system files, applications, or even a business process or task.

4.1.2 Roles and groups

Groups are a registry concept. Roles are a logical concept that is application specific. Roles can be assigned to a user or a group.

4.1.3 Directory and registry

A directory (or repository) is a concrete instance of user data such as LDAP. A registry is a container in which users are identified. A single registry might span multiple directories/repositories. The logical collection of users in a registry is also known as user realm.

4.2 Security in WebSphere Application Server

WebSphere Application Server is the foundation on which WebSphere Process Server, WebSphere Business Monitor, and WebSphere Business Services Fabric are built. This section discusses WebSphere Application Server security considerations that are required to secure related WebSphere products environments.

4.2.1 Overview of security provided by WebSphere Application Server

WebSphere Application Server provides a security infrastructure and mechanisms that protect sensitive Java 2 Platform, Enterprise Edition (J2EE) resources and administrative resources.

WebSphere Application Server security consists of the four components shown in Figure 4-1:

- Application security
- Administrative security
- Java 2 security
- Operating system security

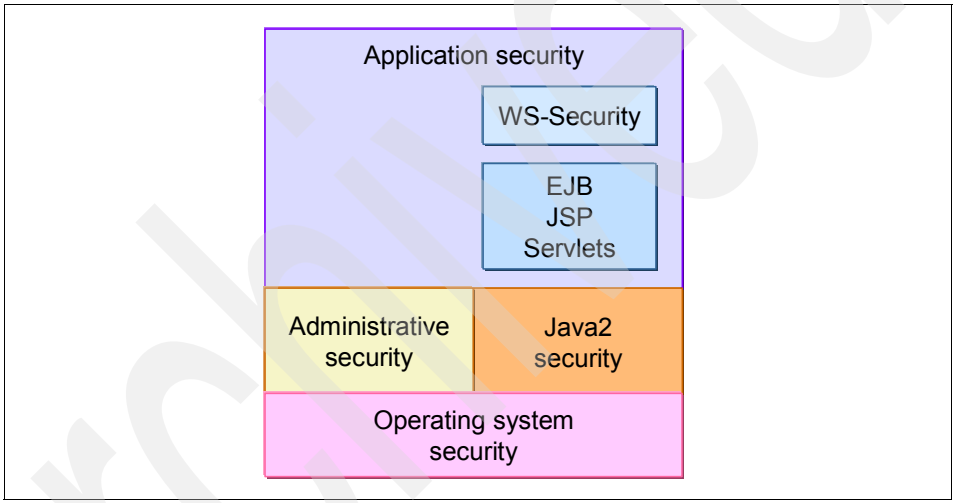


Figure 4-1 WebSphere Application Server security components

For an in-depth look into WebSphere Application Server security, refer to IBM Redbooks publication *IBM WebSphere Application Server V6.1 Security Handbook*, SG24-6316.

4.2.2 Application security

Application security provides application isolation and requirements for authenticating users and controlling their access to the applications in your environment. Application security must be enabled in case declarative security is used by any application deployed in the application server. However, if your application relies only on programmatic security (for example, using the HttpServletRequest interface method getRemoteUser(), where authentication is

already done on the Hypertext Transfer Protocol (HTTP) server side), you do not necessarily have to enable application security.

Security roles are logical and declared at development time. These logical roles are mapped to real users/groups at deployment time. Security roles allow for access control and are associated with J2EE artifacts such as servlets, JSPs, and EJBs.

As an administrator or configurator, you must understand from the development team what security roles to expect, and what users should act on behalf of the particular role. For example, you may work for a bank, and the application you are installing has a role named manager. Does this mean the branch manager, any manager of employees, or some other type of manager? It will be your job to make sure that the correct users and groups are assigned to the role. Within WebSphere Application Server, it is a best practice to assign groups to roles. This is allows for a more flexible, yet secured, environment.

Some applications use Message-Driven Beans (MDBs) and have configured them to use runAs roles. The runAs role is an identity assertion, and the MDB will always run as the user that you have mapped to the role. An authentication alias is an artifact used to define a user that will be mapped to a runAs role.

We recommend requiring a document from the application team, as listed in Table 4-1, which lists and describes the roles for the application. Have a column to enter the actual user or group that you will assign during deployment.

Table 4-1 Sample table of roles and group/user mapping for a retail/banking application

Description of role	**Security role**	**runAs role**	**Administrator assigned users or groups**
Access to functions only for bank managers	bankManager	No	jones or bankmgr
Access to functions for bank tellers	bankteller	No	branchempl
Access to functions for customers	customer	No	all authenticated
Access to run evening accounting	accountingRec	Yes	accntProcess

4.2.3 Administrative security

Administrative security represents the security configuration that affects the entire security domain. The security domain consists of all the servers that are configured with the same user registry realm name. The basic requirement for a security domain is that the access ID returned by the registry from one server be the same access ID as that returned from the registry on any other servers within the same security domain.

Enabling administrative security activates a wide variety of security settings for WebSphere Application Server. While values for these settings can be specified, they take effect only when administrative security is activated. These settings include authentication of users, the use of Secure Sockets Layer (SSL), the choice of user account repository, and application security.

User account repositories

WebSphere Application Server supports several user registries. User registries manage the identities (user names, passwords, and other information) of entities that interact with the system. The available user registries are:

- Federated repositories
- Standalone LDAP registry
- Local operating system
- Standalone Custom registry

Important: The Network Deployment environment does not support the local operating system registry.

Authentication mechanisms

WebSphere Application Server uses Lightweight Third Party Authentication (LTPA) as the default authentication mechanism. LTPA supports forwardable credentials. For security reasons, a configurable expiration time is set on the credentials. The use of LTPA allows you to enable single sign-on (SSO) for your security domain.

In addition, WebSphere Application Server supports using third-party authentication mechanisms through a trusted relationship. This relationship is established using Trust Association Interceptors (TAI). WebSphere Application Server provides four TAIs:

- IBM Tivoli Access Manager (Policy Director)
- WebSEAL Version 5.1
- Simple and Protected GSS-API Negotiation Mechanism (SPNEGO)
- Session Initiation Protocol (SIP)

Single sign-on

When a client request must flow through multiple systems within the enterprise, the client should not have to authenticate several times. The client should be authenticated once. The authenticated context is propagated to downstream systems, which can apply access control.

One use case for WebSphere Application Server integrates Web applications with backend Enterprise Information Systems (EISs). WebSEAL, which is a part of Tivoli Access Manager, can front the Web application and perform authentication on its behalf.

You can configure WebSEAL for trust association with downstream servers, such as WebSphere Application Server. Trust association between two processes means that they have authenticated with each other and trust messages from each other. With trust association, one server can authenticate clients and forward the authenticated context to trusted servers. The trusted servers do not need to authenticate the request again. Figure 4-2 illustrates a trust association between WebSEAL and WebSphere Process Server that is established using SSL.

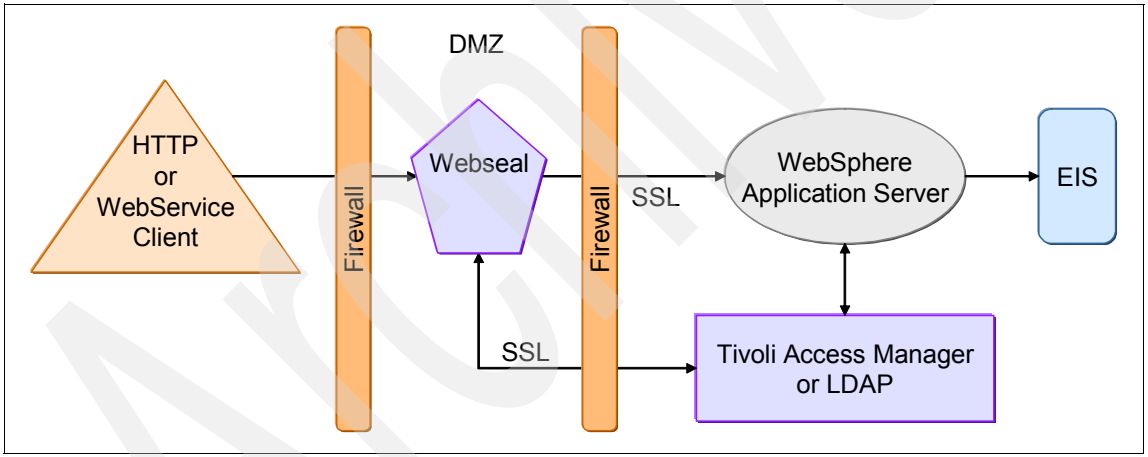


Figure 4-2 Single sign-on

> **Important:** Trust should be limited. When building the SSL infrastructure, limit the number of signer certificates to those that are used for your connections. This limits the clients that can complete the SSL handshake.

If the target Enterprise Information System has its own user registry, you can map the identity from the request to an identity in the target system. By default, WebSphere Application Server supports many-to-one credential mapping. You

can map the identities from the incoming requests to one preconfigured identity in the target EIS security domain. For one-to-one credential mapping, WebSphere Application Server provides a programming interface for developers to create their own custom mapping modules.

Confidentiality and integrity

WebSphere Application Server provides industry-accepted ways to protect the security of data or messages as they flow across the network and out of the network while maintaining the data's integrity and confidentiality:

- Confidentiality

  Confidentiality, or privacy, is the desire that only the sender and the receiver be able to inspect the contents of the message or data. This desire is fulfilled through an encryption protocol. The protocol packages the data with a symmetric key. This key comes from a negotiation just prior to the data being sent. Once this occurs, the data can be read, thus assuring the confidentiality of the data.

- Integrity

  Integrity ensures that there is no unauthorized modification of the stored and transmitted data by using a signature. A signature is created based on a key that the sender is authorized to have. Unauthorized network analyzers do not have this key. When the receiver gets the message, it creates a signature using the message contents. If the two signatures match, the receiver honors the message. If the signatures are different, an error is returned to the sender.

Transport layer security is a function that provides both privacy and data integrity between two communicating applications. The protection occurs in a layer of software on top of the base transport protocol (for example, on top of TCP/IP). These may sound familiar because they are often discussed together. Most of the encryption protocols provide both data confidentiality and integrity. WebSphere Application Server provides support infrastructure for confidentiality and integrity with SSL and WS-Security.

The most commonly known encryption protocol is Secured Sockets Layer (SSL). SSL is also referred to as Transport Layer Security (TLS). SSL provides security over the communications link through encryption technology, ensuring the integrity of messages in a network. Because communications are encrypted between two parties, a third party cannot tamper with messages. SSL also provides confidentiality (ensuring the message content cannot be read), replay detection, and out-of-sequence detection.

With the growth of Web services comes WS-Security. WS-Security is a message-level standard based on securing SOAP messages through XML digital signature, confidentiality through XML encryption, and credential propagation

through security tokens. WS-Security for WebSphere Application Server V6 and later is based on standards that are included in the OASIS Web Services Security Version 1.0 specification, the Username Token Version 1.0 Profile, the X.509 Token Version 1.0 Profile, and a SOAP with Attachments (SWA) Version 1.0 Profile.

One advantage of WS-Security is that it can be configured by the application to be used. The administrator would adapt the applications declarations to her environment.

Service integration bus

A service integration bus is the messaging infrastructure for WebSphere Application Server. Security can be enabled for the bus if administrative security has been enabled for the application server. Access to the bus and resources on the bus is role-based and administered through the WebSphere Application Server wsadmin tool and partially through the Integrated Solutions Console.

Access to the service integration bus is determined by user or group membership in the Bus Connector role. When both administrative security and the bus security are enabled, access to the bus is checked when a user tries to connect to a bus. By default, only the server group is assigned with this role.

The IBM Redbooks publication *IBM WebSphere Application Server V6.1 Security Handbook*, SG24-6316, reviews the messaging roles and destinations and how they can be secured on the bus.

4.2.4 Java 2 security

Java 2 security provides a policy-based, fine-grained access control mechanism that increases overall system integrity by checking for permissions before allowing access to certain protected system resources. Java 2 security guards access to system resources such as file I/O, sockets, and properties. Java 2 Platform, Enterprise Edition (J2EE) security guards access to Web resources such as servlets, JavaServer™ Pages (JSP™) files, and Enterprise JavaBeans™ (EJB™) methods.

Although Java 2 security is supported, it is disabled by default. You can configure Java 2 security and administrative security independently of one another. Disabling administrative security does not disable Java 2 security automatically. You must explicitly disable it.

For more information about Java 2 security with WebSphere Application Server based products refer to the following Web page:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/csec_rsecmgr2.html

4.2.5 Operating system security

You do not want your operating system compromised. You should install and run WebSphere Application Server as a non-root user. However, there are limitations to the operation of WebSphere Application Server as a non-root user. These are documented in the Information Center, at the following Web page:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/cins_nonroot.html

4.3 Security for a WebSphere Process Server solution

WebSphere Process Server security is based on WebSphere Application Server security. The security for WebSphere Process Server is mainly divided into two parts. The first part is related to administrative security of WebSphere Process Server and the second part is the security of the applications running on WebSphere Process Server itself. This section discusses:

- Overview of business integration security
- Access control for SCA container
- Access control for Business Process Choreographer container
- Access control for Common Event Infrastructure container
- Securing SCA modules
- Access control for the Business Calendar Manager
- People resolution and directories

4.3.1 Overview of business integration security

To provide security to the Business Process Choreographer (BPC) and Service Component Architecture (SCA) runtimes, WebSphere Process Server exploits the following WebSphere Application Server security features:

- Application security
- Administrative security
- Java 2 security

SCA adds two components to the application security component of WebSphere Application Server, as shown in Figure 4-3:

- SCA modules
- SCA runtime

BPC adds a third component called BPC runtime to the application security component of WebSphere Application Server, as shown in Figure 4-3.

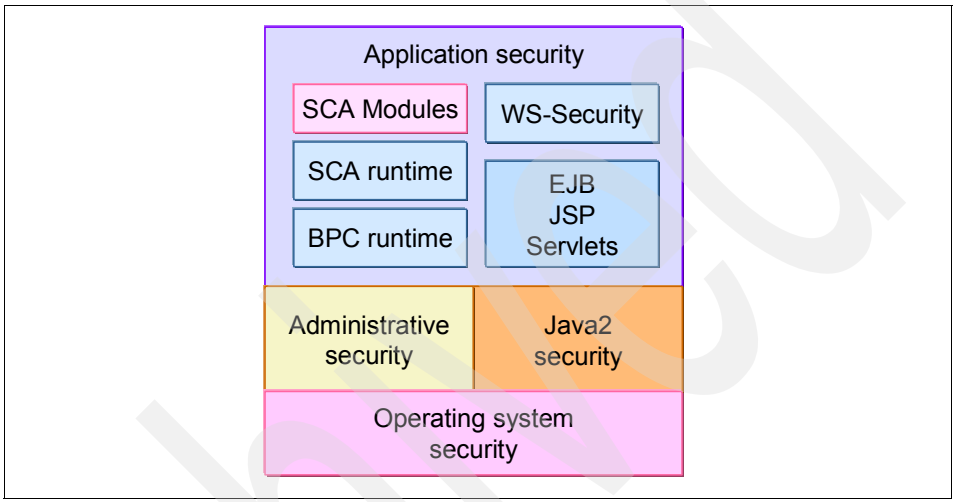


Figure 4-3 WebSphere Process Server security components

WebSphere Process Server makes extensive use of the service integration bus to send and receive messages. Asynchronous invocation in Service Component Architecture (SCA) is implemented using messages that are sent and received over the bus. The integration environment is not secure if you do not secure the bus.

The bus can hold messages until a consumer is ready to consume the message. The bus can store messages either in a database or on disk. Storing in a database is more secure. If you decide to let the bus store messages on a disk, the disk must be protected with operating system security.

The service integration bus supports authentication for connecting to the bus and role-based access control for accessing the destinations and sending, receiving, and browsing messages. Default access control grants permissions to all authenticated users. For a more secure environment, grant permissions only to a limited set of users or groups.

Data is potentially sent over the network between a remote client, such as an adapter and a messaging engine, and between two messaging engines (on

different nodes). To ensure the privacy of this data, encrypt the communication link with the SSL protocol.

The WebSphere Process Server runtime components have message driven beans (MDBs) that are configured with a runAs role. The deployment environments wizard collects the user name and password for the runAs role and creates these authentication aliases.

From the Security hyperlink in the Integrated Solutions Console, there are two ways to modify the aliases:

- Click **Security** → **Business Integration Security**.
- Click **Security**→ **Secure administration, applications, and infrastructure** → **Java Authentication and Authorization Service** → **J2C authentication data**.

The WebSphere Process Server runtime also has supporting applications that should also be more closely governed than the defaults. Review the roles available for each container and the supporting applications so that you can understand what access you will want to grant to certain groups in your organization.

4.3.2 Access control for SCA container

WebSphere Process Server uses container-managed aliases to authenticate with the bus. These aliases, shown in Table 4-2, are set up during creation of the deployment environment.

Table 4-2 SCA-related authentication aliases

Authentication alias	Description	Notes
SCA_Auth_Alias	Used by runtime to authenticate with the messaging engine	User name and password entered on the SCA configuration window of the Create new deployment environment wizard
SCAAPP<db name>_Auth_Alias	Authentication alias for the SCA Application Bus ME data source	User name and password entered on the Database configuration window of the Create new deployment environment wizard
SCASYS<db name>_Auth_Alias	SCA System Bus ME data source authentication alias	User name and password entered on the Database configuration window of the Create new deployment environment wizard

To allow the SCA buses to talk to one another, the user ID for the bus will need to be part of the bus connector role. By default, the SCA_Auth_Alias ID is added to the bus connector role. The security role for the failed event manager is shown in Table 4-3.

Table 4-3 Failed Event Manger roles

Application name	Security role	Description	Notes
wpsFEMgr_6.2.0	WBIOperator	Everyone	Users assigned to this role have administrator privileges. This role is also referred to as the system administrator for Failed Event Manager.

4.3.3 Access control for Business Process Choreographer container

The BPC runtime uses container-managed aliases to authenticate with the bus and datastore. These authentication aliases, shown in Table 4-4, are set up during creation of the deployment environment.

Table 4-4 BPC runtime related authentication alias

Authentication alias	Description	Notes
BPEAuthDataAliasJMS_<node>_<server>	BPC messaging engine datasource user ID	User name and password entered on the BPC configuration window of the Create new deployment environment wizard
BPEAuthDataAlias<DbType>_<node>_<server>	BPC datasource user ID	User name and password entered on the Database configuration window of the Create new deployment environment wizard
JMSAPIUser	Authentication for business flow manager MDB to process asynchronous API calls	User name and password entered on the BPC configuration window of the Create new deployment environment wizard
EscalationUser	Authentication for human task manager MDB to process asynchronous API calls	User name and password entered on the BPC configuration window of the Create new deployment environment wizard

The BPC runtime is installed as an Enterprise Application Archive (EAR) file with security roles that must have users and groups assigned (Table 4-5 on page 78). At a minimum, all of the APIUser roles should be *all authenticated*. You may wish to restrict this even more based on what the development staff has created with these APIs.

Table 4-5 Business Process Choreographer components with Access Control

Application name	Security role	Default permission	Notes
BPEContainer_<deploymentEnvironment.cluster>	BPESystemAdministrator	User or group entered on the Business Process Choreographer configuration window of the Create new deployment environment wizard	Users assigned to this role have all privileges. This role is also referred to as the system administrator for business processes.
	BPESystemMonitor	All authenticated users	Users assigned to this role can view the properties of all business process objects. This role is also referred to as the system monitor for business processes.
	BPEAPIUser	All authenticated users	Users assigned to this role can access BPE Container APIs that are publicly exposed.
	CleanupUser	All authenticated users	Users assigned to this role can run the cleanup jobs. This user ID is used as the J2EE run-as role for the Business Flow Manager and Human Task Manager cleanup services. The cleanup user must be a member of the administrator group.
	JMSAPIUser	All authenticated users	Users assigned to this role can access business flow manager message-driven beans to process asynchronous API calls.

Application name	Security role	Default permission	Notes
TaskContainer_ <deployment Environment. cluster>	TaskSystemAdministrator	User or group entered on the Business Process Choreographer configuration window of the Create new deployment environment wizard	Users assigned to this role can administer the business flow manager and the human task manager. Users for this role have all privileges for the Business Process Choreographer.
	TaskSystemMonitor	User or group entered on the Business Process Choreographer configuration window of the Create new deployment environment wizard	Users assigned to this role can view the properties of all of the task objects. This role is also referred to as the system monitor for human tasks.
	TaskAPIUser	All authenticated users	Users assigned to this role can access Task Container APIs that are publicly exposed.
	CleanupUser	All authenticated users	Users assigned to this role can run the cleanup jobs. This user ID is used as the J2EE run-as role for the Business Flow Manager and Human Task Manager cleanup services. The cleanup user must be member of the administrator group.
	EscalationUser	All authenticated users	Users assigned to this role can access human task manager message -driven beans to process asynchronous API calls.

Application name	Security role	Default permission	Notes
BPCExplorer_ <deployment Environment. cluster>	CleanupUser	All authenticated users	Users assigned to this role can run the cleanup jobs. This user ID is used as the J2EE run-as role for the Business Flow Manager and Human Task Manager cleanup services. The cleanup user must be member of the administrator group.
BPCObserver_ <deployment Environment. cluster>	ObserverUser	All authenticated users	Users assigned to this role can use the Business Process Choreographer Observer.
BusinessSpace Manager	administrator	All authenticated users	Users assigned to this role can administer the business space manager.
BusinessRules Manager_ <deployment Environment. cluster>	BusinessRuleUsers	All authenticated users	Users assigned to this role can use the Business Rules Manager.
	NoOne		Required if Tivoli Access Manager is part of the deployment, as it requires a role for indicating who absolutely cannot access the application.
	AnyOne	All authenticated users, everyone	Anyone can use the Business Rules Manager.

4.3.4 Access control for Common Event Infrastructure container

The Common Event Infrastructure (CEI) runtime uses container-managed aliases to authenticate with the bus and datastore. These authentication aliases, listed in Table 4-6, are set up during creation of the deployment environment. If these aliases are not set up correctly, the server does not function correctly when security is turned on.

Table 4-6 Common Event Infrastructure authentication aliases

Authentication alias	Description	Notes
CommonEventInfrastructure JMSAuthAlias	Used by runtime to authenticate with the messaging engine	User name and password entered on the CEI configuration window of the installer
EventAuthAlias<DBType>r	Used by runtime to authenticate with the database	User name and password entered on the CEI configuration window of the installer

The CEI runtime is enabled as a service with security roles that must have users and groups assigned, as shown in Table 4-7. For greater detail about the uses for each role, refer to the Information Center article *Security and the Common Event Infrastructure*, available at the following Web page:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.websphere.cei.620.doc/doc/ccei_security.html

Table 4-7 CEI components with Access Control: Event Service

Roles	Default permission
eventAdministrator	All authenticated users
eventConsumer	All authenticated users
eventUpdater	All authenticated users
eventCreator	All authenticated users
catalogAdministrator	All authenticated users
catalogReader	All authenticated users

4.3.5 Securing SCA modules

SCA provides you with two additional qualifiers. These are defined in WebSphere Integration Developer for each module as a quality of service (QoS) property. You can also secure components developed by users using the following SCA qualifiers:

- securityPermission

  In this qualifier, you specify the role that has the permission to invoke the secured method.

- securityIdentity

  This qualifier is the same as J2EE runAs identity. The value of this qualifier is a role that is mapped to an identity during deployment. The invocation takes the identity specified.

SCA components are developed using WebSphere Integration Developer. A module with securityPermission is exported from WebSphere Integration Developer as an EAR and installed into WebSphere Process Server.

During the installation, you can assign users to roles using any of the following choices:

- Everyone

  This is equivalent to no security.

- All authenticated

  Every authenticated user is a member of the role.

- Mapped User

  Individual users are added.

- Mapped Groups

  In a real-world enterprise, the administrator should use groups defined in your federated repositories instead of individual users.

Access control for SCA components

Components implement interfaces that have methods. You can secure an interface or method using the SCA qualifier securityPermission. Components are defined using the Service Component Definition Language (SCDL). In the sample SCDL in Example 4-1, access to the one-way invoke method is restricted to users who are members of the role manager.

Example 4-1 SCDL with security qualifiers

```
<?xml version="1.0" encoding="UTF-8"?>
<scdl:component xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:java="http://www.ibm.com/xmlns/prod/websphere/scdl/java/6.0.0"
xmlns:ns1="http://sample.recovery.security/Itarget"
xmlns:scdl="http://www.ibm.com/xmlns/prod/websphere/scdl/6.0.0"
xmlns:wsdl="http://www.ibm.com/xmlns/prod/websphere/scdl/wsdl/6.0.0"
displayName="secure" name="Component1">
<interfaces>
<interface xsi:type="wsdl:WSDLPortType" portType="ns1:Itarget">
<method name="onewayinvoke">
<scdl:interfaceQualifier xsi:type="scdl:SecurityPermission"
role="manager"/>
</method>
</interface>
</interfaces>
<references/>
<implementation xsi:type="java:JavaImplementation"
class="sca.component.java.impl.Component1Impl1">
</implementation>
</scdl:component>
```

For more information about security considerations with WebSphere Process Server, refer to the developerWorks article *WebSphere Process Server security overview*, available at the following Web page:

http://www.ibm.com/developerworks/websphere/library/techarticles/0602_khangaonkar/0602_khangaonkar.html

4.3.6 Access control for the Business Calendar Manager

WebSphere Process Server V6.2 introduces a new business space widget called the security manager widget. The new widget is used to configure role assignments for business calendar usage. The default ID BPMAdmin has the authority to add and remove users from the BPMrolemanager role, which in turn has the authority to remove members from resource roles.

For more information about Business Calender Manager, refer to the Information Center topic *Security for Business Calendar Manager* available at the following Web page:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.websphere.wps.z.620.doc/doc/csec_rolebased.html

4.3.7 People resolution and directories

BPC uses people directory providers as adapters for accessing people directories. You can configure the virtual member manager, LDAP, the user registry, and the system people directory providers to retrieve user information.

The decision about which people directory provider to use depends on the support that you need from people resolution. To exploit all of the people assignment features offered by BPC, use the virtual member manager.

All of the people directory configurations require that WebSphere Application Server administrative and application security are enabled. For more information refer to the Information Center article *People directory providers and configurations*, available at the following Web page:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.websphere.bpc.620.doc/doc/bpc/cpeopledirectory_provider.html

For more information about the overall usage of people directories refer to the developerWorks article *Authorization and staff resolution in Business Process Choreographer: Part 1: Understanding the concepts and components of staff resolution*, available at the following Web page:

http://www.ibm.com/developerworks/websphere/techjournal/0710_lind/0710_lind.html

Instance-based roles

Instance-based roles are valid for individual task and escalation instances, or the templates that are used to create task or escalation instances. Role-based authorization requires that administration and application security is enabled for the application server.

A task instance or an escalation instance is not assigned directly to a person. Instead, it is associated with predefined roles to which people are assigned. Anyone that is assigned to an instance-based role can perform the actions for that role. The association of users to instance-based roles is determined either by people assignment or as the result of task actions.

People are assigned to the following roles at runtime by people assignment, based on the user and user group information that is stored in a people directory:

- Potential creator
- Potential starter
- Potential owner
- Reader
- Editor
- Administrator
- Escalation receiver

The following roles are associated with only one user and are assigned as the result of a task action:

- Originator
- Starter
- Owner

For a complete list, refer to the Information Center article *Instance-based roles for business processes and activities*, available at the following Web page:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.websphere.bpc.620.doc/doc/bpc/c6bpel_auth_instance.html

4.4 Access control for WebSphere Business Services Fabric

This section addresses access control considerations specifically for WebSphere Business Services Fabric.

WebSphere Business Services Fabric security roles

The WebSphere Business Services Fabric runtime uses container-managed aliases to authenticate with the bus and datastore. These authentication aliases, listed in Table 4-8, are set up during configuration of the environment.

Table 4-8 WebSphere Business Services Fabric authentication aliases

Authentication alias	Description	Notes
FABRIC_JDBC_AUTH	Fabric database authentication alias	User name and password for the four WebSphere Business Services Fabric databases: ► Business services repository ► Governance manger ► Performance manager ► Messaging engine
Fabric_Bus_AuthAlias	Fabric bus authentication alias	User name and password for the WebSphere Business Services Fabric service integration bus. This user name will need to be added to the following locations: ► Fabric bus's connector role ► Fabric activationSpecs – Hub Event Activation – Hub Request Activation – DAPerfMon Activation ► DAEventConnection Factory

The WebSphere Business Services Fabric installation pre-populates the six groups to the Fabric Tools application, as listed in Table 4-9. You can either create these groups in your federated repositories or you can add your own groups to these roles and remove the pre-populated roles.

Table 4-9 WebSphere Business Services Fabric security roles: FabricTools and Fabric Catalog

Security role	Default permission	Notes
FabricAdministrator	Group provided at installation: FabricAdministrators Administrators	The system administrator trumps all other roles and can access everything in the system.
FabricStudioUser	Group provided at installation: FabricStudioUsers	The FabricStudioUser role has full access to the Composition Studio to use secure services for Replication, Changelist Submission, and Governance Status and must be able to freely use the BSRViewer to see repository metadata. The FabricStudioUser role also has read-only access to governance views such as Environments, Repository, Namespaces, Projects, Teams, and Changelists, that are necessary for interacting with the governance model. Composition Studio users can create projects and namespaces in a local environment where they have more control and have Governance Administrator import Fabric Content Archives with this content.
FabricGovernanceAdministrator	Group provided at installation: FabricGovernanceAdministrators	The FabricGovernanceAdministrator role controls all changes made to data stored in the Business Services Repository, including the assignment of users to teams and the definition of projects and namespaces
FabricPerformanceUser	Group provided at installation: FabricPerformanceUsers	The FabricPerformanceUser role can view and fully use the Performance Manager.

Security role	Default permission	Notes
FabricSubscriberManager	Group provided at installation: FabricSubscriberManagers	The FabricSubscriberManager has full access to the Subscriber Manager, enabling them to perform all required subscriber management functions.
FabricBasicUser	Group provided at installation: FabricBasicUsers	This role provides read-only access to the Business Service Repository and Governance Manager. It establishes an appropriate access level for a user who must log into the WebSphere Business Services Fabric for z/OS using a browser.

4.5 Access control for WebSphere Business Monitor

The WebSphere Business Monitor runtime uses container-managed aliases to authenticate with the bus and datastore. These authentication aliases, shown in Table 4-10, are set up during configuration of the environment.

Table 4-10 WebSphere Business Monitor authentication aliases

Authentication Alias	Description	Notes
MonitorAlphabloxAlias	Authentication for MonitorAlphabloxAlias	User name and password for AlphaBlox. Deprecated in V6.2.
MonitorBusAuth	Authentication for MONITOR.<cellName>.Bus and Action Services QueueConnFactory	User name and password for monitor bus.
MonitorQueueConnectionFactoryAuth	Authentication for MonitorQueueConnectionFactoryAuth	User name and password for monitor queues.
Monitor_JDBC_Alias	Authentication for the monitor database	User name and password for the monitor database.

WebSphere Business Monitor models can be grouped into resource groups to allow easy administration of data access permissions. Permissions must be assigned to a resource group by way of a three-way binding. This binding consists of a resource group, a role, and a user or group of users.

Monitor Data Security always has a root resource group defined. All resource groups other than root are considered children of root. All resources are visible to the root resource group. By default, all resources are deployed to the root resource group.

A resource can be a member of only one resource group. The roles that can be assigned to a user or group within a resource group are defined by WebSphere Business Monitor. Table 4-11 indicates the roles and the actions that can be completed for each role.

Table 4-11 WebSphere Business Monitor Data Security Roles

Roles	Notes
Business-Manager	This role provides basic read-only access to public (shared) KPIs within a resource group.
Personal-KPI-Adminstrator	This role gives users the authority to create non-shared (personal) KPIs. The created KPI can be viewed and updated only by the owner and a KPI-Administrator.
Public-KPI-Adminstrator	This role gives users the authority to create shared (public) or non-shared (personal) KPIs. Shared (public) KPIs can be used and viewed by other users. Only the owner or a KPI-Administrator can make changes to a shared (public) KPI.
KPI-Adminstrator	This role gives users all the authority associated with KPI administration. Users of this role can create both shared (public) and non-shared (personal) KPIs. In addition, KPI-Administrators can change the ownership of any KPI.
SuperUser	Full access.

The roles referenced in Table 4-12 are used for WebSphere Business Monitor dashboards. These roles encompass access to AlphaBlox, REST APIs, and Business Space.

Table 4-12 WebSphere Business Monitor dashboard security roles

Application name	Security role	Default permission
AlphabloxPlatform	AlphabloxAdministrator	All authenticated users
	AlphabloxDeveloper	All authenticated users
	AlphabloxUser	All authenticated users
ApplicationStudio	AlphabloxAdministrator	All authenticated users
	AlphabloxUser	All authenticated users
IBM_WBM_REST_SERVICES	monitorusers	All authenticated users
IBM_BSPACE_WIDGETS	Administrator	All authenticated users

4.6 Additional security considerations

This section provides resources and advice about additional security considerations.

4.6.1 Creating a secured link between two cells

When configuring your business integration systems, you may have to link two completely different cells together into a configuration referred to as a *cross-cell* or *cross-linked* configuration. This is a configuration where two standalone WebSphere Process Server environments inter-communicate, or WebSphere Business Monitor inter-communicates with WebSphere Process Server at a different cell. These connections are probably two SCA modules where the import is bound through synchronous or asynchronous bindings.

You want to configure your processes to communicate using a secured channel, so you must configure SSL so that the consuming cell has the signer certificate of the producing cell. If this is bidirectional, then you must exchange signers between the cells.

More information about this topic can be found in the in the Information Center article *Exchanging signer certificates*, available at the following Web page:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/tsec_sslexchangesigncerts.html

If you are going to trust the other cell, then you can swap the Lightweight Third Party Authentication (LTPA) key. Follow the instructions in the Information Center article *Managing LTPA keys from multiple WebSphere Application Server cells*, available at the following Web page:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tsec_sslmanagelptakeys.html

Synchronous communications

The synchronous communication configuration closely resembles an EJB client application. The consuming cell looks up the module in the namespace of the producing cell, then binds to the bootstrap port of the producing cell. The lookup call can be done in two ways:

- Programatically

  You can look up a remote Java Naming and Directory Interface (JNDI) namespace using a provider URL like corbaloc::<hostname>:<port>. This gives the developer control, but it is not a flexible solution. These values can also be looked up from a properties file, which will provide more flexibility but is not a centrally managed solution.

- Declarative

  Instead of creating name space bindings from a program, you can configure them with the Integrated Solutions Console. Name servers add these configured bindings to the name space view by reading the configuration data for the bindings. Configured bindings are created each time that a server starts, even when the binding is created in a transient partition of the name space. One use of configured bindings is to provide fixed qualified names for server application objects. The required steps to create name space bindings are provided in the Information Center article *Configuring name space bindings*, available at the following Web page:

  http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tnam_view_bindings.html

When building the name space binding, use Common Object Request Broker Architecture (CORBA) object binding settings. Your lookup string would look like Example 4-2.

Example 4-2 Declarative lookup string

```
context.lookup("providerCell/com/ibm/bpe/api/BusinessFlowManagerHome");
```

Because you are in a trusted cell and you are authenticated, your user identity flows to the provider cell. The user must have permissions to execute the

routines, so it must be a member of the group assigned to BPEAPI, TASKAPI, or JMSAPI roles. Which role depends on which set of APIs it will be calling.

Configuring asynchronous communications

This configuration is a little more complicated. You must create the same configuration changes in both cells. This allows both cells to send messages back and forth to both SCA modules:

1. Define a foreignBus in your cell (Example 4-3).

   Example 4-3 Create foreign bus

   ```
   AdminTask.createSIBForeignBus('[-bus SCA.APPLICATION.WPSCell01.Bus
   -name SCA.APPLICATION.WPSCell02.Bus -routingType Direct -type SIBus
   -inboundUserid SCA -outboundUserid SCA]')
   ```

2. Add users to destination roles (Example 4-4).

   Example 4-4 Add role to destination

   ```
   AdminTask.addUserToDestinationRole('[-bus
   SCA.APPLICATION.WPSCell01.Bus -type ForeignDestination -foreignBus
   SCA.APPLICATION.WPSCell02.Bus -destination SCAApp.Response -role
   sender -user SCA]')'
   ```

3. Add user to bus connector role (Example 4-5).

   Example 4-5 Add user to bus connector role

   ```
   $AdminTask addUserToBusConnectorRole {-bus
   SCA.APPLICATION.WPSCell01.Bus -user SCA}
   ```

4. Create a service bus link (Example 4-6).

   Example 4-6 Create bus link

   ```
   AdminTask.createSIBLink('[-bus SCA.APPLICATION.WPSCell01.Bus
   -messagingEngine MECluster.000-SCA.APPLICATION.WPSCell01.Bus -name
   WPSCell02Link -foreignBusName SCA.APPLICATION.WPSCell02.Bus
   -bootstrapEndpoints 9.16.41.7:7286:BootstrapSecureMessaging
   -remoteMessagingEngineName
   MECluster.000-SCA.APPLICATION.WPSCell02.Bus -description SIBLink
   -protocolName InboundSecureMessaging -authAlias SCA_Auth_Alias]')
   ```

5. Create a SIB destination (Example 4-7).

Example 4-7 Create SIB destination

```
AdminTask.createSIBDestination('[-bus SCA.APPLICATION.WPSCell01.Bus
-name SCAApp.Request -type FOREIGN -foreignBus
SCA.APPLICATION.WPSCell02.Bus -description -reliability
ASSURED_PERSISTENT -maxReliability ASSURED_PERSISTENT
-overrideOfQOSByProducerAllowed true -sendAllowed true ]')
```

6. Create a remote queue (Example 4-8).

Example 4-8 Create remote SIB JMSQueue

```
AdminTask.createSIBJMSQueue('ReuCell(cells/WPSCell01|cell.xml)',
'[-name SCAApp.Request -jndiName jms/SCAAppRequest -description
-queueName SCAApp.Request -deliveryMode Application -readAhead
AsConnection -busName SCA.APPLICATION.WPSCell02.Bus]')
```

7. Create a local queue (Example 4-9).

Example 4-9 Create local SIB JMSQueue

```
AdminTask.createSIBJMSQueue('ReuCell(cells/WPSCell01|cell.xml)',
'[-name SCAApp.Response -jndiName jms/SCAAppResponse -description
-queueName SCAApp.Response -deliveryMode Application -readAhead
AsConnection -busName SCA.APPLICATION.WPSCell01.Bus]')
```

For a more detailed explanation of configuring SCA cross-cell review the article *Instructions for configuring SCA cross-cell communications,* available on the following Web page:

http://www.ibm.com/support/docview.wss?uid=swg21216929

For a more information about cross-cell configuration for WebSphere Business Monitor with remote CEI, refer to the Information Center article *Configuring a remote CEI server to use WebSphere Business Monitor* at the following link

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.monitor.install.doc/admin/cei_cfg_for_rem_monitor.html

For defining JMS SIB security and problem determination, review the IBM Redpaper publication *WebSphere Application Server V6.1: JMS Problem Determination*, REDP-4330.

4.6.2 Security considerations tips

Every organization has a similar goal, which is to run in a highly secure environment. However, each organization has corporate security policies that govern the configuration of its environment. The WebSphere Process Server deployment environment comes configured securely with a file registry. This security configuration contains the following information:

- An authentication alias for every MDB.
- Certain roles that you assigned to users or groups during the initial configuration.
- Roles not configured during the deployment environment wizard are assigned all authenticated.

From reading the previous sections, you are aware of the numerous roles and authentication aliases that you must manage. If security is simple, the system can be easily compromised. There are, however, several practices that can make your job easier. These practices might reduce your flexibility, so you must weigh their benefits against your needs and corporate security policies.

Create two IDs for each infrastructure ID

If you have corporate policies that require password changes every *X* number of days, creating a secondary ID allows you to change passwords without creating a system outage. There are specific instructions to change passwords without outages. These same instructions work for bus runAs role user IDs as well. If you follow this methodology, make sure that the second ID is also assigned to the role (for example, bus connector role).

Use groups for infrastructure IDs

You may decide to have a different user for each messaging engine. If so, you must add each user to the bus connector role so that the containers have access to the bus. One way to reduce the users associated with the bus connector role is to create a user repository group for your messaging engine user IDs and assign this group the bus connector role. If you are creating a secure bus link between two or more cells, you may add the users acting on behalf of the foreign bus in a group and assign that group to the different roles.

Consolidate authentication aliases

As you review your system, you may notice that there are multiple authentication aliases running with the same user ID and password. For example, if you have decided to configure all of your data access authentication aliases to run as one ID, you may create a new one and reconfigure the environment to just use this alias. This reduces the number of locations where you will need to change the

password or user ID in the future. One drawback to this is that there are certain Integrated Solutions Console panels, such as the Business Integration Security, that will no longer be useful for these IDs.

The groups listed in Table 4-13 and Table 4-14 on page 96 are used to administer different aspects of your BPM environment.

Table 4-13 Business Process Management System Administrators

Group	User	Password	Description
admin	wasadmin	passw0rd	This group is the WebSphere Process Server Administrators.
	wps	passw0rd	(Primary Admin ID)
	wsadmin	passw0rd	
security	wpssec	passw0rd	This group is the WebSphere Process Server administrative role adminstrator.
	wassec	passw0rd	
	John	passw0rd	
	tom	passw0rd	
monadmin	monitor	passw0rd	This group is the WebSphere Business Monitor Administrators. This group must be mapped when installing the WebSphere Business Monitor.
	mohamed	passw0rd	
fabadmin	fabric	passw0rd	This group is the WebSphere Business Services Fabric Administrators. This group must be mapped when installing the WebSphere Business Services Fabric.
	Addison	passw0rd	

Table 4-14 System users

Group	User	Password	Description
wpsusers	hutch	passw0rd	This group is the WebSphere Process Server operators.
	mohamed	passw0rd	
	gates	passw0rd	
monuser	tom	passw0rd	This group is the WebSphere Business Monitor.
	John	passw0rd	
	mohamed	passw0rd	
wpscfg	naveen	passw0rd	This group is the WebSphere Process Server configurators.
	John	passw0rd	
	cath	passw0rd	

Part 2

Building topologies for WebSphere Process Server

5

Preparing your topology

This chapter provides instructions for installing and configuring the prerequisite software used in the topologies discussed in this publication. There are common installation and configuration tasks across each topology described in this IBM Redbooks publication. Later chapters refer to this chapter for instructions when creating the topology.

This chapter contains step-by-step procedures for the following tasks:

- Prerequisite software installation
- Database creation
- Profile creation via GUI
- Profile creation via scripting
- Post installation configuration

5.1 Prerequisite software installation

This section discusses and describes the installation process for the foundational products used in the topologies discussed in this book. Both silent and GUI installation methods are discussed. Sample scripts are provided for silent installation options.

5.1.1 Software versions

To create any of the topologies in this publication, the following software is required. The instructions are for Linux® systems, but the general process is the same for all distributed platforms:

- SUSE® Linux Enterprise Server 10 SP1
- WebSphere Process Server V6.2
- IBM DB2® Universal Database™ V9.5
- IBM Tivoli Directory Server V6.2
- IBM HTTP Server V6.1

5.1.2 Software installation

This section contains pointers to instructions for installing the software versions required by the topologies discussed in this book.

Installing IBM DB2 V9.5

Note: Silent installation of DB2 is covered in the Information Center article available at the following Web page:

http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.uprun.doc/doc/c0007503.htm

An example response file is shown in Example 5-1.

Example 5-1 Example DB2 response file db2response.txt

```
*  Product Installation
LIC_AGREEMENT= ACCEPT
PROD       = ENTERPRISE_SERVER_EDITION
FILE  = /opt/ibm/db2/V9.1
INSTALL_TYPE= TYPICAL
*-----------------------------------------------
*  Das properties
*-----------------------------------------------
```

```
DAS_CONTACT_LIST= LOCAL
DAS_USERNAME= dasuser1
DAS_GROUP_NAME= dasadm1
DAS_HOME_DIRECTORY= /home/dasuser1
DAS_PASSWORD= passw0rd
* ------------------------------------------------
*  Instance properties
* ------------------------------------------------
INSTANCE= inst1
inst1.TYPE= ese
inst1.NAME= db2inst1
inst1.GROUP_NAME= db2grp1
inst1.HOME_DIRECTORY= /home/db2inst1
inst1.PASSWORD= passw0rd
inst1.AUTOSTART= YES
inst1.SVCENAME= db2c_db2inst1
inst1.PORT_NUMBER= 50000
inst1.FCM_PORT_NUMBER= 60000
inst1.MAX_LOGICAL_NODES= 4
*  Fenced user
inst1.FENCED_USERNAME= db2fenc1
inst1.FENCED_GROUP_NAME= db2fgrp1
inst1.FENCED_HOME_DIRECTORY= /home/db2fenc1
inst1.FENCED_PASSWORD= passw0rd
*------------------------------------------------
*  Installed Languages
*------------------------------------------------
LANG       = EN
```

Installing IBM Tivoli Directory Server

Note: The installation and configuration of IBM Tivoli Directory Server is described in the Information Center article at the following Web page:

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

The LDIF file used for this environment is included in the additional materials supplied with this book in Appendix A, “Additional material” on page 597.

Installing WebSphere Process Server base product

Note: Silent installation of WebSphere Process Server is covered in the Information Center article available at the following Web page:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp?topic=/com.ibm.websphere.wps.62.doc/doc/iins_rf_wps.html

Example 5-2 is the response file that we used.

Example 5-2 Example WebSphere Process Server V6.2 response file

```
-OPT silentInstallLicenseAcceptance="true"
-OPT disableOSPrereqChecking="true"
-OPT disableNonBlockingPrereqChecking="true"
-OPT installType="installNew"
-OPT wpsInstallType="typical"
-OPT samplesSelected="false"
-OPT installLocation="/opt/ibm/WebSphere/ProcServer"
-OPT useExistingWAS="false"
-OPT profileType="none"
```

Installing the update installer

WebSphere Update Installer V7 is needed to install several iFixes to WebSphere Business Services Fabric.

Note: The version of the update installer used in this book is available at the following Web page:

ftp://ftp.software.ibm.com/software/websphere/appserv/support/tools/UpdateInstaller/7.0.x/LinuxIA32/7.0.0.1-WS-UPDI-LinuxIA32.zip

The installation of WebSphere Update Installer V7 is covered in the Information Center article available at the following Web page:

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.installation.base.doc/info/aes/ae/tins_updi_install.html

5.1.3 Add a Web server

Note: Silent installation of the IBM HTTP Server is covered in the Information Center article available from the following Web page:

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_silentinstall.html

Example 5-3 is a sample response file.

Example 5-3 Example response file for IBM HTTP server

```
-OPT silentInstallLicenseAcceptance="true"
-OPT allowNonRootSilentInstall=false
-OPT disableOSPrereqChecking="true"
-OPT installLocation="/opt/IBM/HTTPServer"
-OPT installGSKit="true"
-OPT httpPort="80"
-OPT adminPort="8008"
-OPT createAdminAuth="false"
-OPT adminAuthUser="wasadmin"
-OPT adminAuthPassword="passw0rd"
-OPT adminAuthPasswordConfirm="passw0rd"
-OPT runSetupAdmin="true"
-OPT createAdminUserGroup=true
-OPT setupAdminUser="wasadmin"
-OPT setupAdminGroup="wasadmin"
-OPT installPlugin="true"
-OPT webserverDefinition="webserver1"
-OPT washostname="itsodmgr"
```

5.2 Database creation

In this section the databases are created before beginning the process of creating the topology.

5.2.1 Overview

Following common practice, a database is created with one name and a schema within that database with a different name. These names are listed in Table 5-1.

Table 5-1 Database details including owner, database name and schema

Instance (owner)	Database name	Schema name	Comments
db2inst1	WPRCSDB	COMMONDB	The common database. The default schema name is the same as the instance owner.
db2inst1	BPEDB	BPC	The BPC database. The default schema name is the same as the instance owner.
db2inst1	OBSVRDB	OBS	The Business Process Observer database. The default schema name is the same as the instance owner.
db2inst1	MEDB	SCASYS	The SCA System messaging data store. There is no default schema name.
db2inst1	MEDB	SCAAPP	The SCA Application messaging data store. There is no default schema name.
db2inst1	MEDB	CEIME	The CEI messaging data store. There is no default schema name.
db2inst1	MEDB	BPCME	The BPC messaging data store. There is no default schema name.
db2inst1	EVENT		The Event database for CEI events. Note that there is no specific schema associated with this database, so it uses the instance owner.
db2inst1	BSPCDB	BSPACE	The database for Business Space powered by WebSphere.

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 5-1 on page 104, where the instance owner db2inst1 manages four databases and, for example, the MEDB database has four schemas.

To create all the databases, you must copy scripts to the DB2 system. These scripts are located in <install_root>/dbscripts. For example, on Linux they can be found in `/opt/ibm/WebSphere/ProcServer/dbscripts`. We recommend that you copy the entire folder to the DB2 system under the instance owner directory.

In this edition of the book, the databases are created before profile creation. Consequently, some of the files must be edited to insert schema names. You can defer table creation until after the deployment topology is generated and the default database scripts are used to generate valid SQL scripts with the schema names already embedded. You can then run these scripts directly.

5.2.2 Common database

Using the command-line interface to DB2, create the common database on the DB2 system using the scripts copied from the deployment manager.

1. Log in to the DB2 system as the instance owner for the common database, as given in Table 5-1 on page 104.
2. Change to the dbscripts folder `cd ~/dbscripts/CommonDB/DB2`.

3. Make adjustments to the following files:
   - `configCommonDB.sh`
   - `createDBTables.sh`
   - `insertTable_CommonDB.sql`

   a. Edit the file `configCommonDB.sh` and change the values for #DB_NAME# and #DB_USER# to the values given in Table 5-1 on page 104. In this example they are WPRCSDB and db2inst1, respectively. The modified text is shown in Example 5-4.

   Example 5-4 Corrected details for the file configCommonDB.sh

```
##############################
#  DB_NAME will be replaced
##############################
DB_NAME=WPRCSDB
##############################
#  DB_USER will be replaced
##############################
USER_NAME=db2inst1
```

   b. Edit the file `createDBTables.sh`. Look for the line `db2 set current schema=$DB_USERID` and change this to the value for the schema name. In this example this line becomes `db2 set current schema=COMMONDB`.

   c. Edit the file `insertTable_CommonDB.sql` and for each of the insert statements change the values as follows:
      - #MajorVersion#: 6
      - #MinorVersion#: 2
      - #RefreshPackLevel#: 0
      - #FixpackLevel#: 0

   Before and after examples are shown in Example 5-5 and Example 5-6.

   Example 5-5 Changes to the insertTable_CommonDB.sql file: Before

```
INSERT INTO SchemaVersionInfo VALUES ('recovery.ejb',
#MajorVersion#, #MinorVersion#, #RefreshPackLevel#, #FixpackLevel#,
0):
```

   Example 5-6 Changes to the insertTable_CommonDB.sql file: After

```
INSERT INTO SchemaVersionInfo VALUES ('recovery.ejb', 6, 2, 0, 0,
0);
```

4. Create the database, schema, and tables with the following command:

   ```
   ./configCommonDB.sh createDB
   ```

   After the database is created you will be asked for the instance owner password. The remainder of the table creation is then performed. Check the output carefully for errors.

5. Check the database using the command-line interface to DB2:

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

   Sample output is shown in Example 5-7. An additional database table called MSGLOG (with schema ESBLOG) is also created by configCommonDB.sh.

Example 5-7 Output of 'db2 list tables for schema COMMONDB' (the output is truncated)

```
db2 list tables for schema COMMONDB
Table/View                      Schema
APPTIMESTAMP                    COMMONDB
BYTESTORE                       COMMONDB
BYTESTOREOVERFLOW               COMMONDB
CUSTPROPERTIES                  COMMONDB
FAILEDEVENTBOTYPES              COMMONDB
FAILEDEVENTDETAIL               COMMONDB
FAILEDEVENTMESSAGE              COMMONDB
FAILEDEVENTS                    COMMONDB
MEDIATION_TICKETS               COMMONDB
PERSISTENTLOCK                  COMMONDB
RELN_METADATA_T                 COMMONDB
SCHEMAVERSIONINFO               COMMONDB
WSCH_LMGR                       COMMONDB
WSCH_LMPR                       COMMONDB
WSCH_TASK                       COMMONDB
WSCH_TREG                       COMMONDB
```

5.2.3 Business Process Choreographer database

When creating the Business Process Choreographer database you can either create a simple database for testing purposes or follow common practice for production topologies, which is to use a dedicated tablespace and disks for performance. Both of these options are outlined below.

Creating a test database

For a simple database, where performance is not important, perform the following steps:

1. Change to the appropriate folder:

   ```
   cd ~/dbscripts/ProcessChoreographer/DB2
   ```

2. Edit the file `createDatabase.sql` and change the line that connects to the database to include the user name and password. Also add a schema name here. An example is shown in Example 5-8.

   Example 5-8 Corrected details for the file createDatabase.sql

   ```
   -- create the database
   CREATE DATABASE BPEDB USING CODESET UTF-8 TERRITORY en-us;
   -- connect to the created database:
   -- Use CONNECT TO BPEDB USER xxx when another user should become
   owner of the schema
   CONNECT TO BPEDB USER db2inst1 using 'passw0rd';
   CREATE SCHEMA BPC;
   set current schema=BPC;
   ```

3. Create the database with the **`db2 -tf createDatabase.sql`** command.

4. In production environments, you may wish to separate the database log directory to a separate file system. Example 5-9 shows how to do this. Note that /u1 and /u2 are on separate file systems with dedicated disk storage systems.

   Example 5-9 Creating a separate log directory

   ```
   CREATE DATABASE BPEDB AUTOMATIC STORAGE YES ON
   /u1/data/db2inst1/BPEDB USING CODESET UTF-8 TERRITORY en-us;
   CONNECT TO BPEDB USER db2inst1 using 'passw0rd';
   UPDATE DATABASE CONFIG FOR BPEDB USING NEWLOGPATH
   /u2/log/db2inst1/BPEDB;
   CONNECT RESET;
   ```

5. If you are not creating a higher performance database, continue on to 5.2.4, “Process Observer database” on page 109.

Creating a higher performance database

For a higher performing database follow these instructions. In production topologies the tablespaces would use their own high-performance disks.

1. Change to the appropriate folder:

   ```
   cd ~/dbscripts/ProcessChoreographer/DB2
   ```

2. Edit the file `createTablespace.sql`. Change each occurrence of @location@ to your chosen location (for example, /home/db2inst1/db2inst1/NODE0000).

3. Edit the file `createSchema.sql`. Change each occurrence of the phrase @SCHEMA@ to your chosen schema name (for example, BPC).

4. Create the database, tablespace, and schema:

   ```
   db2 "CREATE DATABASE BPEDB USING CODESET UTF-8 TERRITORY en-us"
   db2 connect to BPEDB USER db2inst1 using 'passw0rd'
   db2 “CREATE SCHEMA BPC”
   db2 -tf createTablespace.sql
   db2 -tf createSchema.sql
   db2 connect reset
   ```

5.2.4 Process Observer database

You can create the Business Process Observer database in a similar way to the Business Process Choreographer database, a simple one for testing purposes or a higher performance one for production environments. Both methods are described below.

Creating a test database

For a simple database where performance is not important, use these instructions:

1. Change to the appropriate folder:

   ```
   cd ~/dbscripts/ProcessChoreographer/DB2
   ```

2. Edit the file `createDatabase_Observer.sql` and change the line that connects to the database to include the user name and password. Also add a schema name here. An example is shown in Example 5-10 after the changes have been made.

   Example 5-10 Corrected details for the file createDatabase_Observer.sql

   ```
   -- create the database
   CREATE DATABASE OBSVRDB USING CODESET UTF-8 TERRITORY en-us;
   -- connect to the created database:
   -- Use CONNECT TO OBSVRDB USER xxx when another user should become
   owner of the schema
   CONNECT TO OBSVRDB USER db2inst1 using 'passw0rd';
   CREATE SCHEMA OBS;
   set current schema=OBS;
   ```

3. Create the database with the following command:

   ```
   db2 -tf createDatabase_Observer.sql
   ```

4. If you are not creating a higher performance database, you may now go on to 5.2.5, “Messaging engine database resources” on page 111.

Creating a higher-performance database

For a database that is higher performing follow these instructions. In real production topologies the tablespaces would use their own high-performance disks.

1. Change to the appropriate folder:

   ```
   cd ~/dbscripts/ProcessChoreographer/DB2
   ```

2. Edit the file `createTablespace_Observer.sql`. Change each occurrence of @location@ to your chosen location (for example, /home/db2inst1/db2inst1/NODE0000).

3. Edit the file `createSchema_Observer.sql`. Change each occurrence of @SCHEMA@ to your chosen schema name (for example, OBS).

4. Create the database, tablespace, and schema:

```
db2 "CREATE DATABASE OBSVRDB USING CODESET UTF-8 TERRITORY en-us"
db2 connect to OBSVRDB USER db2inst1 using 'passw0rd'
db2 “CREATE SCHEMA OBS”
db2 -tf createTablespace_Observer.sql
db2 -tf createSchema_Observer.sql
db2 connect reset
```

5.2.5 Messaging engine database resources

This section describes how to create the messaging engine database resources.

Creating messaging engine schemas

Before creating the messaging engine schemas, the DDL files must first be generated on the deployment manager:

1. Log in to the deployment manager. Generate four schemas.

```
cd /opt/ibm/WebSphere/ProcServer/bin

./sibDDLGenerator.sh -system db2 -platform unix -schema SCAAPP -user
db2inst1 -statementend \; > /tmp/SCAAPP.ddl

./sibDDLGenerator.sh -system db2 -platform unix -schema SCASYS -user
db2inst1 -statementend \; > /tmp/SCASYS.ddl

./sibDDLGenerator.sh -system db2 -platform unix -schema BPCME -user
db2inst1 -statementend \; > /tmp/BPCME.ddl

./sibDDLGenerator.sh -system db2 -platform unix -schema CEIME -user
db2inst1 -statementend \; > /tmp/CEIME.ddl
```

2. Transfer all four generated ddl files to the DB2 system under the db2inst1 user’s home folder.

Creating messaging engine database

Verify that the files were transferred to the database host and db2inst1 user as noted in “Creating messaging engine schemas” on page 111. The creation of the database and schemas will be done on the DB2 host.

Log in to the DB2 system as the instance owner, then run these commands:

```
db2 "CREATE DATABASE MEDB USING CODESET UTF-8 TERRITORY en-us"
db2 connect to MEDB USER db2inst1 using 'passw0rd'
db2 -tf SCAAPP.ddl
db2 -tf SCASYS.ddl
db2 -tf CEIME.ddl
db2 -tf BPCME.ddl
db2 connect reset
```

5.2.6 Event database

Log in to the DB2 system as the instance owner, then run the following commands:

```
db2 "CREATE DATABASE EVENT USING CODESET UTF-8 TERRITORY en-us"
db2 connect to EVENT USER db2inst1 using 'passw0rd'
db2 connect reset
```

5.2.7 Business Space database

Log in to the DB2 system as the instance owner, then run the following commands:

```
db2 "CREATE DATABASE BSPCDB USING CODESET UTF-8 TERRITORY en-us"
db2 connect to BSPCDB USER db2inst1 using 'passw0rd'
db2 create schema BSPACE
db2 connect reset
```

5.2.8 Verify database tables

At this point it is a good idea to verify that the tables were created properly in the above steps. You created six databases in the above steps:

- BPEDB
- MEDB
- OBSVRDB
- WPRCSDB
- EVENT
- BSPCDB

Perform the following:

1. Log in to the DB2 system as the instance owner, then run the following command:

   ```
   db2 list database directory
   ```

2. Verify that each of the databases that were created exist.

5.2.9 Next steps

At this point you have two choices as to how to proceed. You can create a topology using the steps in 5.3, “Profile creation (GUI)” on page 113, or you can create the same topology silently, using the scripts described in 5.4, “Profile creation (scripting)” on page 127. Security considerations for these options are described in Chapter 4, “Security considerations for BPM” on page 65.

5.3 Profile creation (GUI)

This is a brief outline of the steps required to create the topology. The steps are described in the sections that follow.

1. Create a deployment manager profile. See 5.3.1, “Deployment manager profile” on page 113.
2. Create a node (custom) profile on each system and federate into the cell. See 5.3.2, “Node profiles” on page 123.
3. Generate a deployment topology using the deployment environments. These are discussed throughout Part 2, “Building topologies for WebSphere Process Server” on page 97.
4. Populate the EVENT database. See 5.5, “Populate the event database” on page 132.

5.3.1 Deployment manager profile

To create a deployment manager profile:

1. Log in to the deployment manager machine as the root user.
2. Run the profile management tool:

   ```
   /opt/ibm/WebSphere/ProcServer/bin/ProfileManagement/pmt.sh
   ```

   A splash window is displayed. Click **Next**.

3. At the Welcome to the Profile Management tool window, click **Next**.
4. In the Environment Selection window (Figure 5-1), click **WebSphere Process Server** and click **Next**.

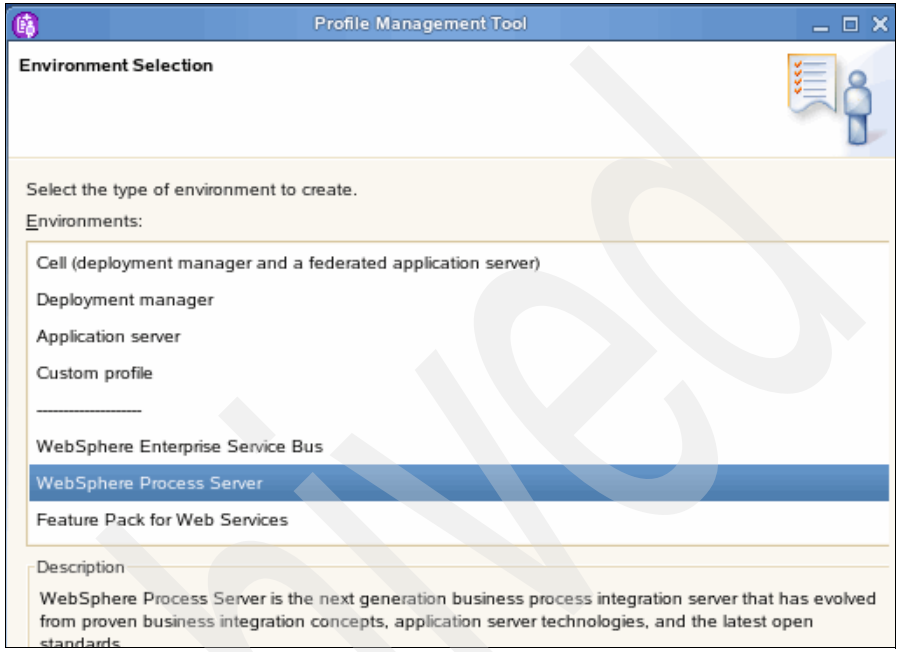


Figure 5-1 Environment Selection window

5. In the Profile Type Selection window (Figure 5-2), click **Deployment manager profile** and click **Next**.

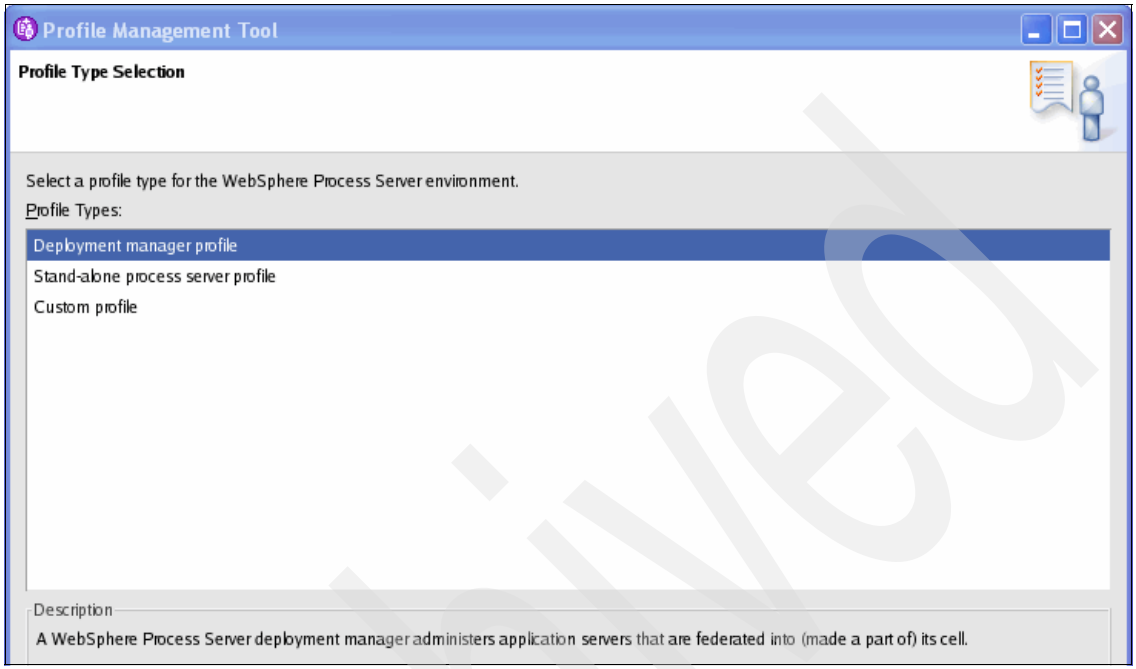


Figure 5-2 The Profile Type Selection window

6. In the Profile Creation Options window (Figure 5-3), click the **Advanced profile creation** radio button and click **Next**.

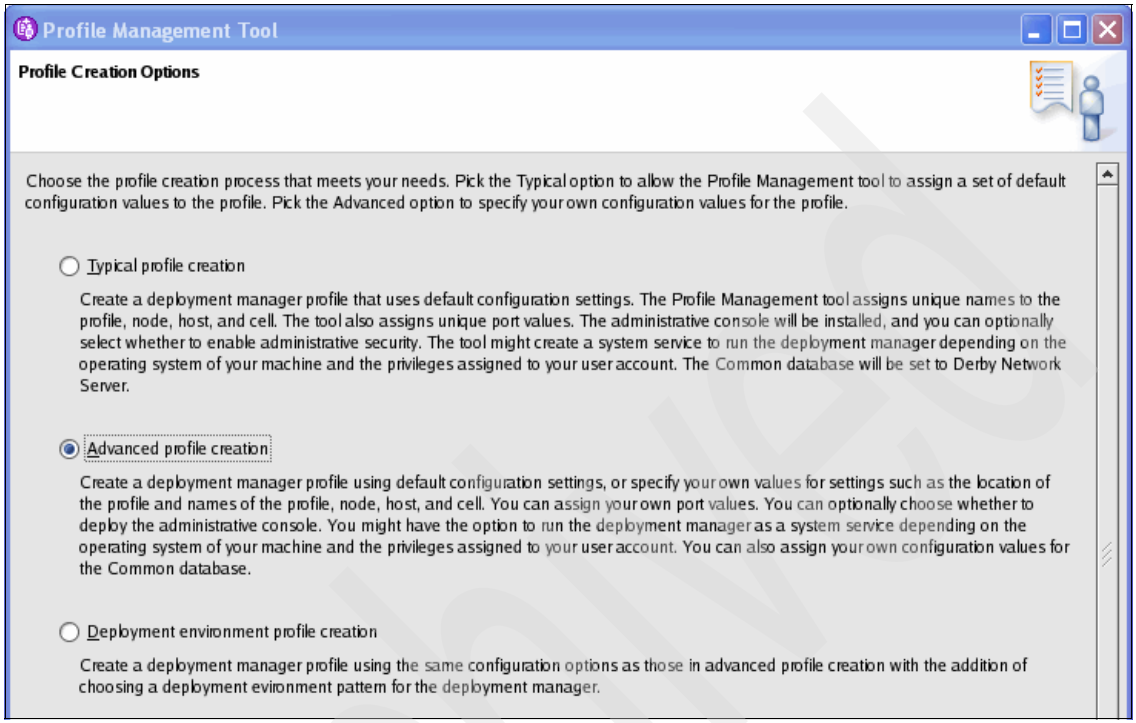


Figure 5-3 The Profile Creation Options window

7. In the Optional Application Deployment window, leave the Deploy the administrative console check box selected and click **Next**.

8. In the Environment Selection window, click **WebSphere Process Server** and click **Next**.
9. In the Node, Host, and Cell Names window (Figure 5-4), enter `CellManager01` in the Node name text box. Do not change the host name. Enter `WPSCell01` in the Cell name text box. Click **Next**.

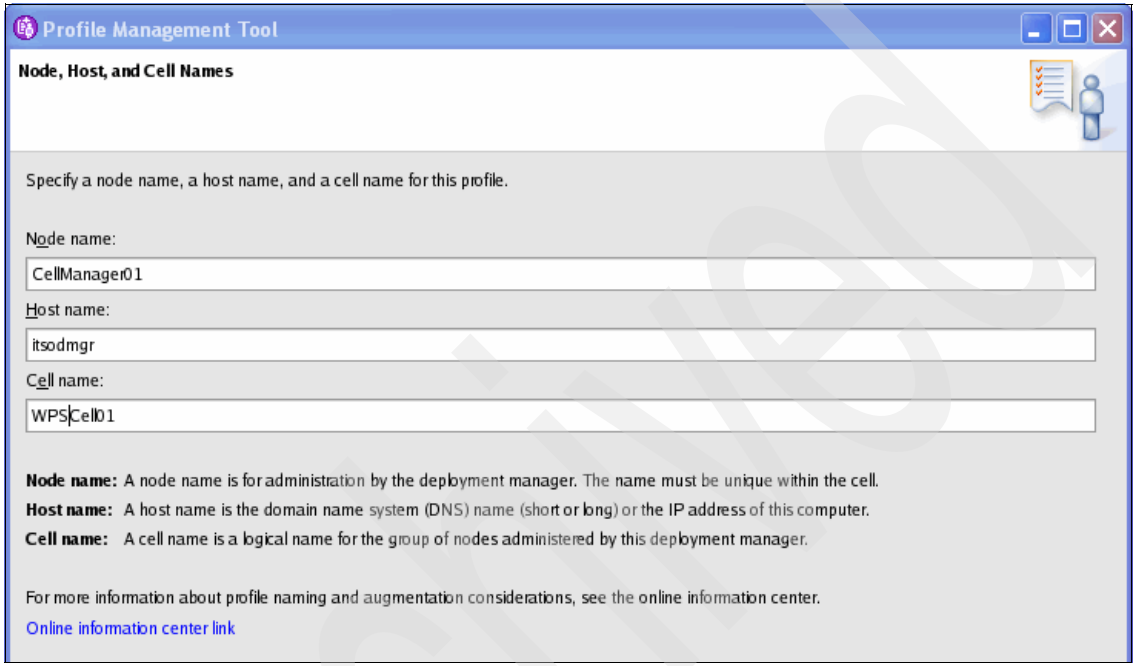


Figure 5-4 The Node, Host, and Cell Names window

10. In the Administrative Security window, clear the Enable administrative security check box and click **Next**. Security will be added to the topology later.
11. In the Port Values Assignment window, accept the default values and click **Next**.
12. In the Linux Service Definition window, leave the default value (cleared) for the Run the deployment manager process as a Linux service check box. Click **Next**.

13. Configure the following items in the Database Configuration window (Figure 5-5):
   - In the Choose a database product text box, select **DB2 Universal** from the drop-down menu.
   - Enter the value `WPRCSDB` in the Database name text box.
   - Select the **Delay execution of database scripts for new or existing database** check box, and click **Next**.

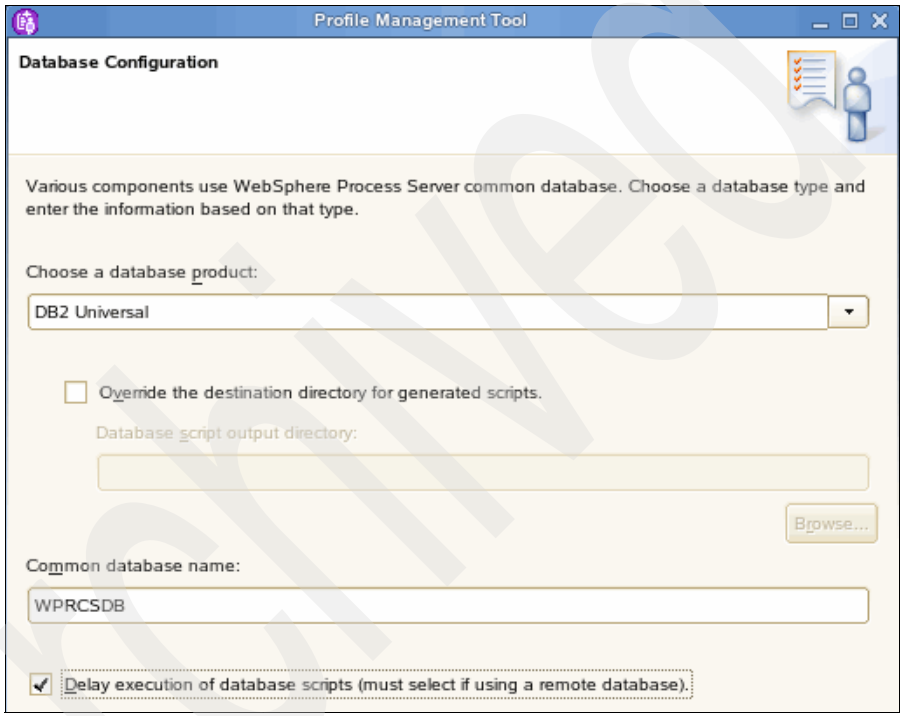


Figure 5-5 Database Configuration window

14. Configure the following items in the Database Configuration (Part 2) window (Figure 5-6):

   a. Enter the value `db2inst1` in the Username text box to authenticate with the database text box from Table 5-1 on page 104.

   b. Enter your password in the Password for database authentication text box. As you enter the password in the first box, a note will appear at the top of the window with the message `Please confirm your database password`. Enter your password again in the Confirm password text box. This text box disappears after you enter the value in the Confirm password text box.

   c. Leave the Location (directory) of JDBC™ driver classpath files text box with the default values.

   d. Enter the host name or IP address of your DB2 Server, itsodb2, in the Database server host name (for example IP address) text box.

   e. Enter a value of `50000` for Server port and click **Next** (Figure 5-6).

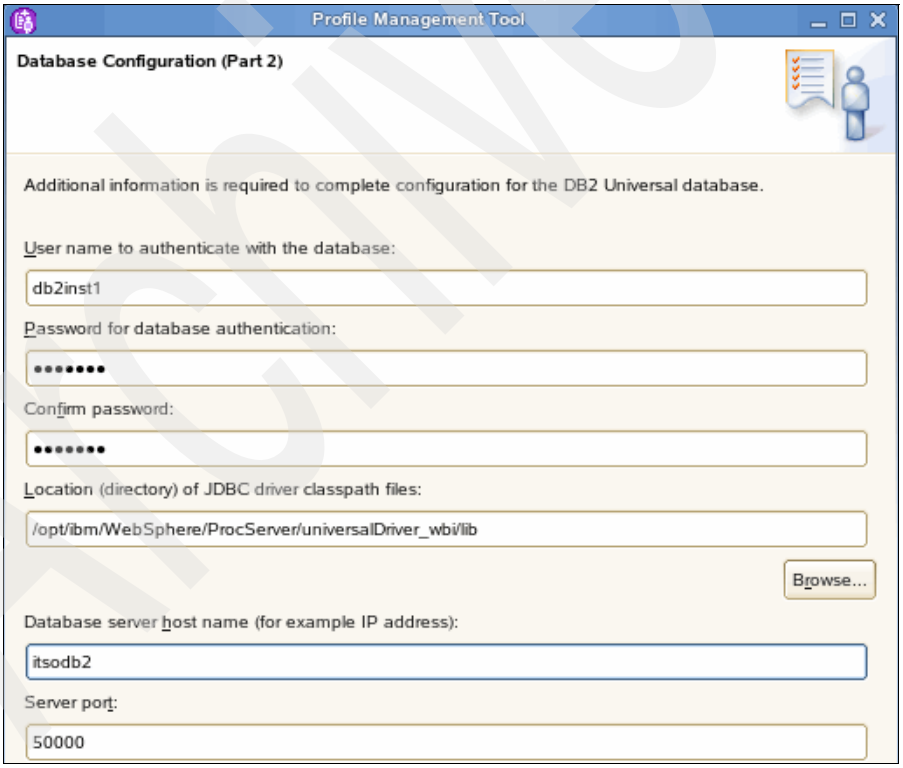


Figure 5-6 Database Configuration (Part 2)

15. In the Profile Creation Summary window (Figure 5-7), check the values and click **Create**. This takes some time to complete. A Profile Creation Progress window is displayed during this process.

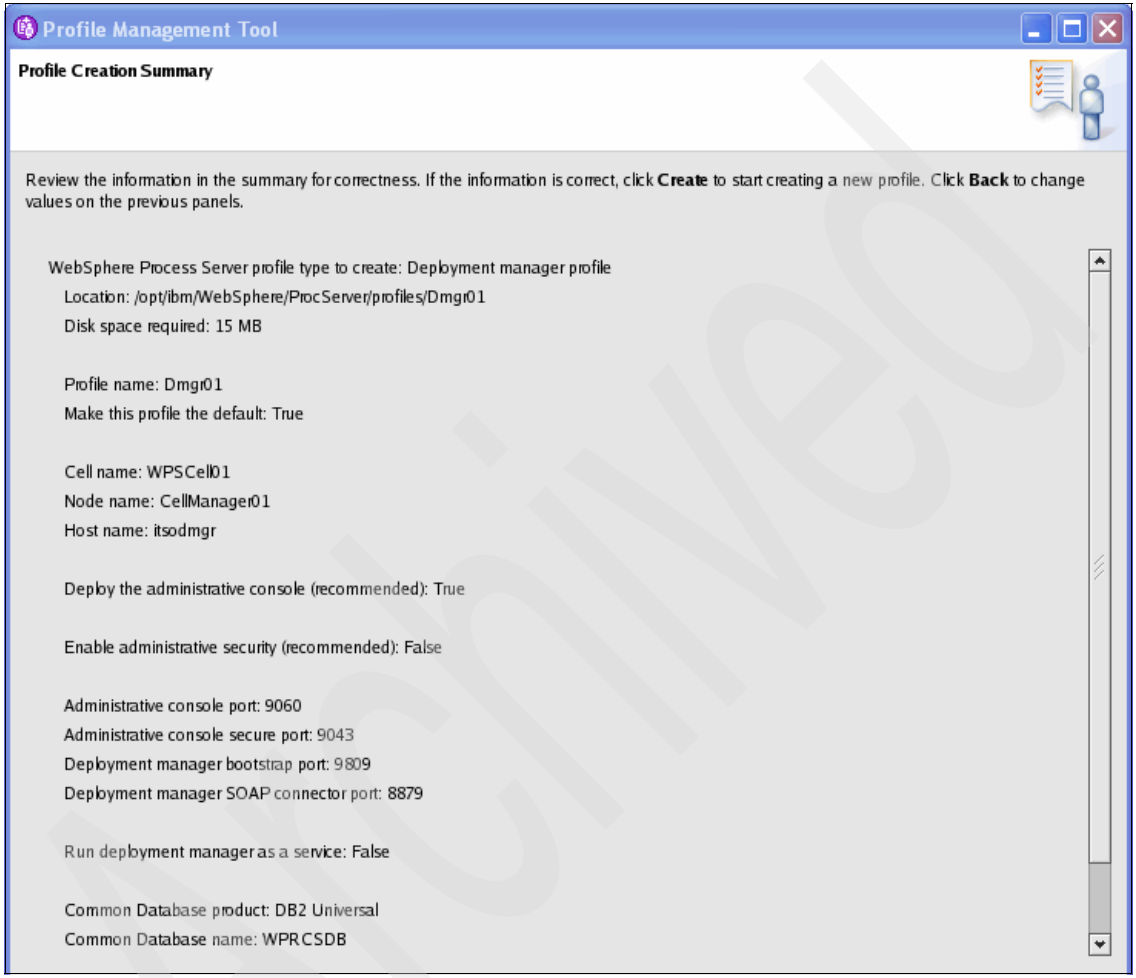


Figure 5-7 The Profile Creation Summary window

16. In the Profile Creation Complete window (Figure 5-8) make sure that the profile creation was successful. Clear the Launch the First steps console check box. Click **Finish**.

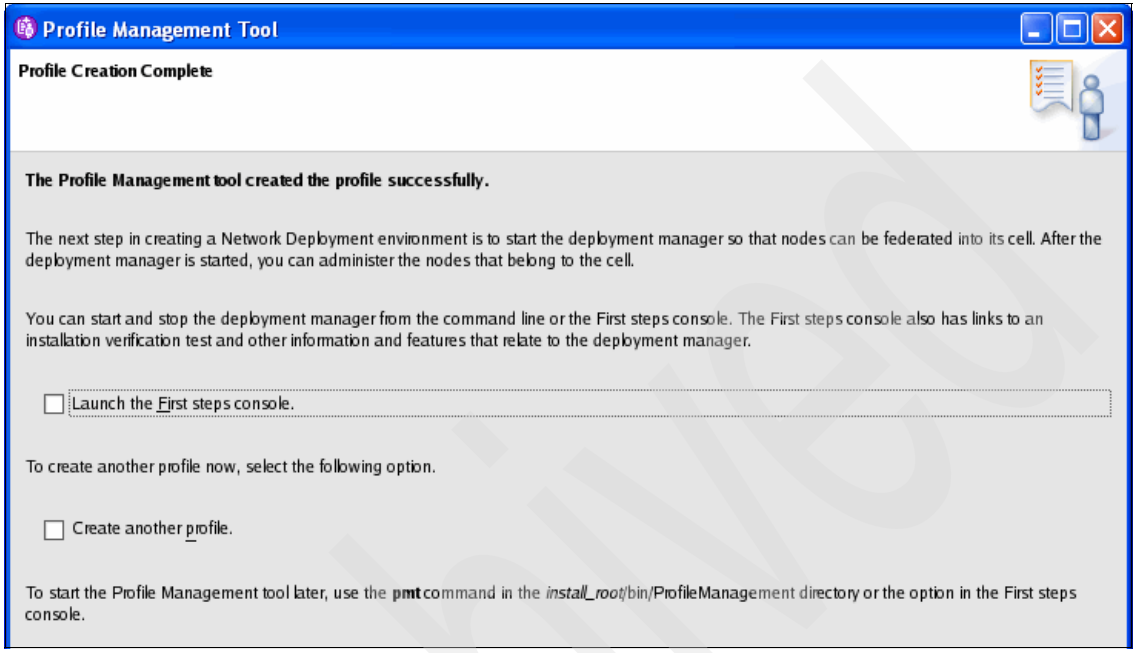


Figure 5-8 The Profile Creation Complete window

17. Start the deployment manager by navigating to:

    `/opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/bin`

18. Enter the command:

    `./startManager.sh`

19. Verify that you receive a message indicating that the process started successfully.

20. Log in to the administrative console (Figure 5-9) by using the URL `http://<host_name>:9060/ibm/console`, where <*host_name*> is the host name of the deployment manager or its IP address.

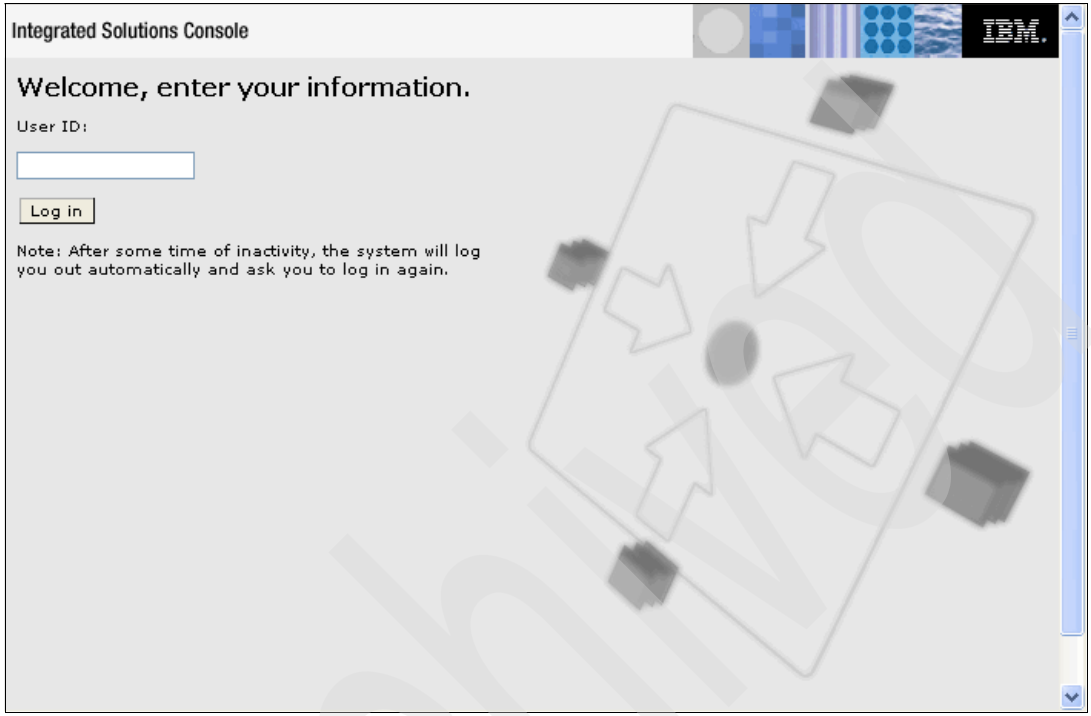


Figure 5-9 The administrative console login window

There are some changes needed for the deployment to be accurate because a schema name of COMMONDB was used. By default, the WPRCSDB database does not have a schema name. It uses the instance owner. Perform the following steps to make these changes:

1. Navigate to **Resources** → **JDBC** → **JDBC Providers**. There is only one provider in the scope (Cell:WPSCell01). Click this provider.
2. Under Additional Properties click **Data sources** and you will see the following two data sources defined:
   - ESBLoggingMediationDataSource
   - WBI_DataSource
3. Click **WBI_DataSource**.
4. Scroll down and under Authentication alias for XA recovery, click the **Use component-managed authentication** alias radio button. Click **OK**.

5. Click **WBI_DataSource** and under Additional Properties click **Custom Properties**.
6. Scroll down the list and click **currentSchema**, enter the value `COMMONDB`, then click **OK**.
7. Scroll down the list, click **cliSchema**, and enter the value `COMMONDB`. Click **OK**.
8. Click **Save** at the top of the page.
9. Repeat steps 2–8, this time clicking **ESBLoggingMediationDataSource**. This enables the component-managed authentication alias. Use the schema value ESBLOG. Save your changes.
10. Log out of the Integrated Solutions Console.
11. Navigate to:

    ```
    /opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/bin
    ```

12. Execute the the following commands to restart the deployment:

    ```
    ./stopManager.sh
    ./startManager.sh
    ```

5.3.2 Node profiles

> **Important:** Before starting node creation you must ensure that the system time on the deployment manager and the system time on the nodes is within 5 minutes of each other.

Before you begin to create the nodes make sure that the deployment manager is running, because you will federate the nodes as part of the creation process. Note that many of the windows in this process are similar to the windows for deployment manager creation, so the different windows are shown here.

1. Log in to the first node (itsonode1) as the root user.
2. Navigate to:

    ```
    /opt/ibm/WebSphere/ProcServer/bin/ProfileManagement
    ```

3. Run the profile management tool:

    ```
    ./pmt.sh
    ```

4. After a splash window is displayed, the Welcome to the Profile Management tool window is displayed. Click **Next**. The Environment Selection window is displayed.

5. Click **WebSphere Process Server** and click **Next**. The Profile Type Selection window is displayed (Figure 5-10).
6. Click **Custom profile** and click **Next**. The Profile Creations Options window is displayed.

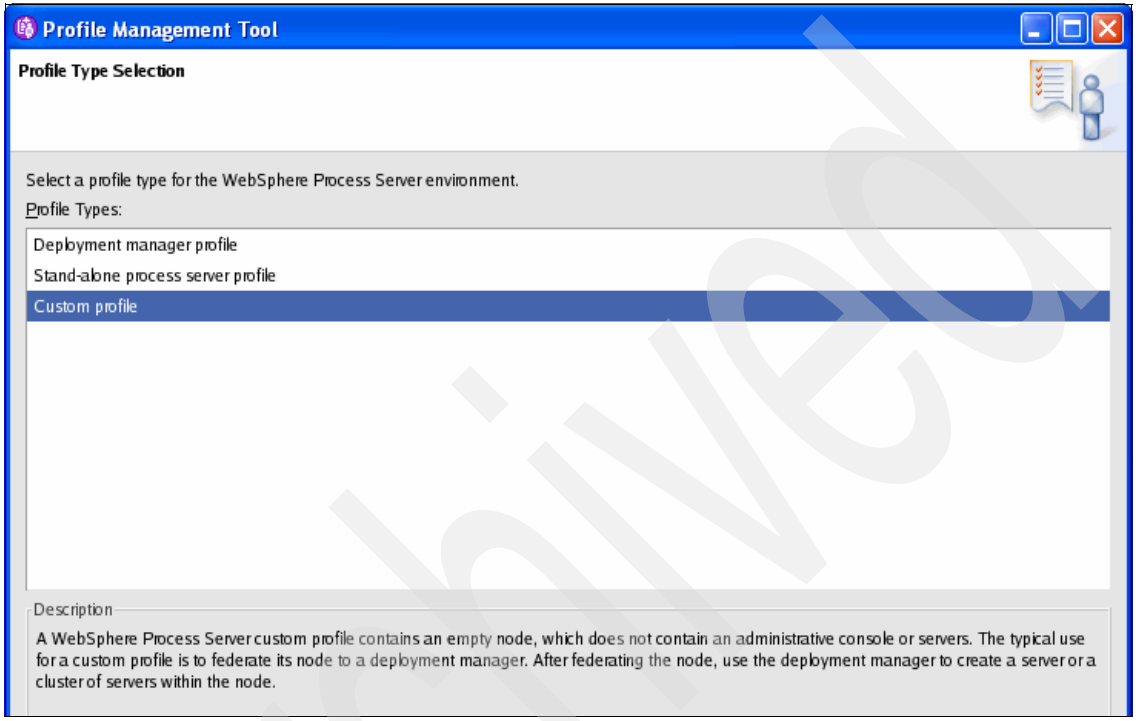


Figure 5-10 The Profile Type Selection window with a custom profile selected

7. Click **Advanced profile creation** and click **Next**. The Profile Name and Location window is displayed.
8. Leave the profile name and profile directory values at their defaults. Click **Next**. The Node and Host Names window is displayed.
9. Enter `wpsNode1` for the node name, accept the default for the host name, and click **Next**. The Federation window is displayed.

10. Enter the host name for the deployment manager (for example, itsodmgr) in the Deployment manager host name or IP address text box (Figure 5-11). Leave all other values at their defaults. You do not need a user name and password because security will not be enabled at this stage. Therefore, leave these values empty. Click **Next**.

> **Note:** You may elect to select to federate this node later if you wish to federate all nodes to the deployment manager after all nodes are created and the deployment manager is started. If you select Federate this node later, you must manually federate each node with the **addNode** command.

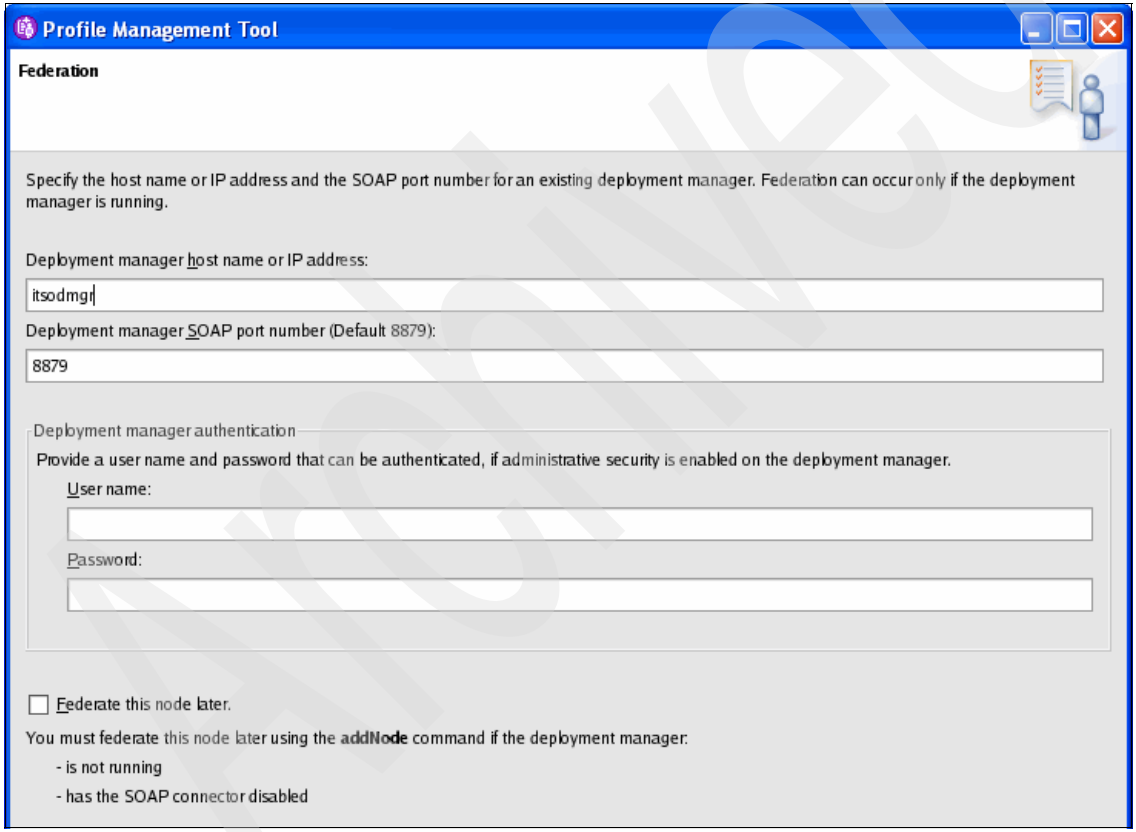


Figure 5-11 The Federation window

11. The Port Values Assignment window is displayed. Accept all the default values and click **Next**. The Database Configuration window is displayed (Figure 5-12).

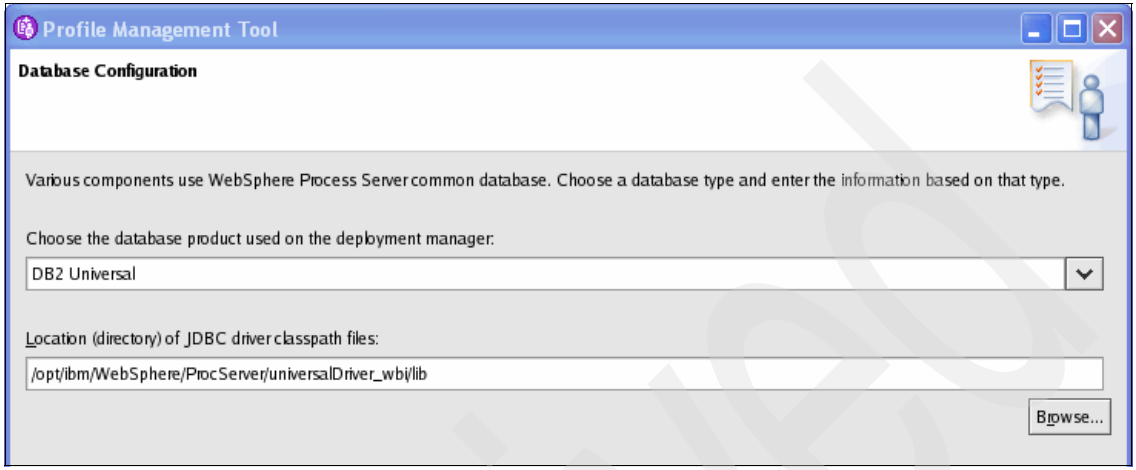


Figure 5-12 The Database Configuration window for a custom profile

12. In the Choose the database product used on the deployment manager drop-down menu, click **DB2 Universal**. Leave the other value at the default and click **Next**. The Profile Creation Summary window is displayed.
13. Check the values and click **Create**. This takes some time to complete. A Profile Creation Progress window is displayed. When profile creation is complete, the Profile Creation Complete window is displayed.
14. Ensure that the profile creation was successful. Clear the Launch the First steps console radio button. Click **Finish**.

In case of failure: If the process creation fails the most likely causes are:

- No connectivity between your node and the deployment manager.
- Time synchronization between the node and the deployment manager must be within 5 minutes of one another.

You can now log in to the other node and perform the same series of steps to create a custom profile there. In the Profile Name and Location window, you may wish to change the profile name to Custom02. Similarly, change the Profile directory to end in Custom02 and on the Node and Host name window use wpsNode02 as the Node name. These should be the only changes.

Creating the node profiles automatically starts the node agent so you can log in to the Integrated Solutions Console and verify that the nodes are available. In the Integrated Solutions Console navigate to **System Administration** → **Node agents** and on the right-hand side you should see you newly created nodes running (Figure 5-13).

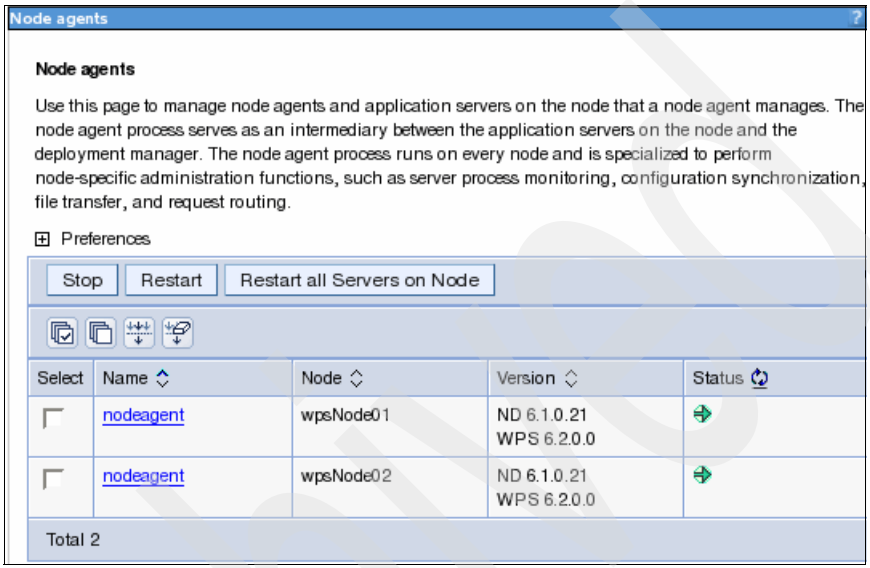


Figure 5-13 List of running node agents

5.4 Profile creation (scripting)

This section demonstrates the silent install process. Start from point where the databases have been created and the WebSphere Process Server product has been installed but no profiles have been created. This is a brief outline of the steps required to create the topology:

1. Create a properties file. See 5.4.1, “Create a properties file” on page 128.
2. Create a deployment manager profile. See 5.4.2, “Deployment manager profile” on page 129.
3. Create the node profiles. See 5.4.3, “Node profiles” on page 131.
4. Generate a deployment topology using the deployment environments. These are discussed throughout Part 2, “Building topologies for WebSphere Process Server” on page 97.
5. Populate the EVENT database. See 5.5, “Populate the event database” on page 132.

5.4.1 Create a properties file

Many of the values used in silent installation are the same as used when installing through the Integrated Solutions Console (for example, database name, database user, and database password). Therefore, start by creating a simple properties file to contain these values. Edit a file called properties.sh with the contents shown in Example 5-11. The values that may need to be changed are dmgrName, dmgrPort, dbHost, dbPort, dbUser, and dbPass.

Example 5-11 The properties.sh file

```
#!/bin/sh

# Basic locations of product install and profiles.

wasDir=/opt/ibm/WebSphere/ProcServer # WPS install location
profDir=${wasDir}/profiles           # Profiles location
binDir=${wasDir}/bin                 # WPS binaries

# Cell configuration
dmgrName=itsodmgr   # Host name or IP
dmgrPort=8879       # SOAP Connector port
cellName=WPSCell01  # Cell Name

# If global security is enabled we need these values

adminUser=wasadmin # WPS User
adminPass=passw0rd # WPS Password

# DB2 configuration information

dbName=WPRCSDB  # Common DB Name
dbHost=itsodmgr # Common DB Host
dbPort=50000    # Common DB Port
dbUser=db2inst1 # Common DB User
dbPass=passw0rd # Common DB Password
dbJDBC=${wasDir}/universalDriver_wbi/lib # JDBC Driver location

# Messaging engine schema names
schemaNames="BPCME CEIME SCASYS SCAAPP"
```

The values should be self-explanatory. This file will be read by the other files used for creating a deployment manager and node profiles.

5.4.2 Deployment manager profile

Create a deployment manager profile silently using the script createDmgr.sh. This script can be found in the associated zip file referenced in Appendix A, “Additional material” on page 597. Make sure that it is located in the same folder as the `properties.sh` file. This script takes the following three optional parameters:

- Cell name (default WPSCell01)
- Deployment manager name (default Dmgr01)
- Node name (default CellManager01)

Once you have inspected this file, you can run it with the default values by executing the **`sh ./createDmgr.sh`** command.

After a short time, the deployment manager will be created. The output is shown in Example 5-12.

Example 5-12 The output of createDmgr.sh

```
INSTCONFSUCCESS: Success: Profile Dmgr01 now exists. Please consult
/opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/logs/AboutThisProfile.txt
for more information about this profile.
```

Note: The script and properties file assumes there are no port conflicts. The properties file uses the default ports when creating a deployment manager. You can specify different ports using the -startingPort value (for example, -startingPort 20000) or using the -portsFile option (for example, -portsFile Myports.props), and list the ports explicitly in the given file.

Make the following changes to the deployment manager:

1. Change the SCA_Auth_Alias password from the default of SCA to your own value (passw0rd).
2. Adjust the currentSchema custom property for the two data sources created:
   - For the data source ESBLoggerMediationDataSource, the currentSchema should be ESBLOG.
   - For the data source WBI_DataSource, the value should be COMMONDB.
3. Make sure that the data sources use the same authentication alias for XA recovery.

These steps are performed using a Jython script named changeDmgr.py based on the toolkit library, also provided in the additional materials of this book. Edit the file called `changeDmgr.py`. You should edit the SCA_Auth_Alias password (the

third parameter to modifyJ2CAuthData) to your needs. This file must be placed in the same folder as the toolkit libraries.

Verify that the deployment manager is stopped. Run this script as follows on the deployment manager:

```
/opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/bin/wsadmin.sh -lang
jython -conntype NONE -f changeDmgr.py
```

The output is shown in Example 5-13.

Example 5-13 The output of changeDmgr.py

```
====== Modify JAAS Auth Alias SCA_Auth_Alias, if it exists ======
Modification of SCA_Auth_Alias was successful.

====== Add Custom Property currentSchema to WBI_DataSource ======
Modifying currentSchema values

====== Add Custom Property currentSchema to
ESBLoggerMediationDataSource ======
Modifying currentSchema values

====== Add Custom Property cliSchema to WBI_DataSource ======
Modifying cliSchema values

====== Add Custom Property cliSchema to ESBLoggerMediationDataSource
======
Modifying cliSchema values
```

Note: The first time that you run wsadmin.sh, the system will process many JAR files leading to many lines of output. This only happens once, and the output is not shown in this example.

Finally, start the deployment manager:

```
/opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/bin/startManager.sh
```

5.4.3 Node profiles

Before starting node creation: You must ensure that the system time on the deployment manager and the system time on the nodes is within 5 minutes of each other.

Before creating the nodes, make sure that the deployment manager is running because you will federate the nodes as part of the creation process. You will create the first profile silently using a script named createNode.sh. This script can be found in the associated zip file referenced in Appendix A, “Additional material” on page 597.

Inspect the `createNode.sh` file. Make sure that it is in the same folder as the location of the `properties.sh` file.

Note that this script takes two optional parameters:

- A profile name (default Custom01)
- A node name (default wpsNode01)

Once you have created this file, you can run it by executing the following command:

```
./createNode.sh Custom01 wpsNode01
```

After a short time, your node will be created and federated into the cell. Federating the node automatically starts the node agent. The output is shown in Example 5-14.

Example 5-14 The output of createNode.sh

```
INSTCONFSUCCESS: Success: Profile Custom01 now exists. Please consult
/opt/ibm/WebSphere/ProcServer/profiles/Custom01/logs/AboutThisProfile.t
xt for more information about this profile.
```

For additional nodes, edit this file and change the values of profName and nodeName, but because these are parameters to the script, you can create the second node with the following command:

```
./createNode.sh Custom02 wpsNode02
```

After a short time, your node will be created and federated into the cell.

5.5 Populate the event database

The final task before generating the topology is to create the event database tables. The scripts to do this are available in the associated zip file referenced in Appendix A, “Additional material” on page 597.

- cr_db2.db2
- cr_tbl.db2
- cr_ts.db2
- dbConfigureCr.sh
- ins_metadata.db2

The first task is to copy these scripts to the DB2 system under the instance owner. The steps below assume that you have copied the files into the home folder of the instance owner.

1. Log in to the DB2 System as the instance owner.
2. Change the directory to the scripts just copied, then run the following command:

   ```
   echo “db2inst1:passw0rd” | ./dbConfigureCr.sh 1 | tee output.log
   ```

   In this command, replace db2inst1 with the instance owner and passw0rd with your chosen password. This creates the database and tables.
3. The database and tables will be created. Check the file `output.log` for any messages.

   Note: These DB2 commands may report various informational messages. This includes the following messages:

   ```
   SQL0598W  Existing index "BPCME.SIB000PKIX" is used as the index
   for the primary key or a unique key.  SQLSTATE=01550
   ```

   Or:

   ```
   SQL20189W  The buffer pool operation (CREATE/ALTER) will not take
   effect until the next database startup due to insufficient
   memory.  SQLSTATE=01657
   ```

   These are not errors. You can ignore these messages.

4. Log off the DB2 system.

5.6 Post-installation configuration

This section describes two post-installation tasks.

5.6.1 Add a Web server to the administrative console

During the installation of the Web server, a script called configurewebserver1.sh is created to simplify the integration with the administrative console. This script is created in the `/opt/IBM/HTTPServer/Plugins/bin` folder.

Copy this script to your deployment manager in the folder `/opt/ibm/WebSphere/ProcServer/bin` and run it with the following command:

```
sh ./configurewebserver1.sh -ihsAdminPassword passw0rd
```

After you run this script. the Web server should appear under **Servers** → **Web servers** within the administrative console. It allows you to start and stop the Web server and generate and propagate the plug-in. You must ensure that the IBM HTTP Server admin server is running to use this functionality.

5.6.2 Install sample application

You can verify your configuration by using a sample application supplied with this book. For more details see Chapter 2, “Sample business application scenario used in topologies” on page 23.

6

Configuring a Single Cluster topology

This chapter provides full instructions for creating a *bronze* topology, which uses the Single Cluster topology pattern for WebSphere Process Server V6.2. In this topology, all the functional pieces (user applications, messaging infrastructure, CEI, and support applications) run in the same cluster.

6.1 Single Cluster topology creation prerequisites

In this section the databases are created before beginning the process of creating the topology. In the single server topology, the databases are local. We assume that the base product has been installed but that no profiles have been created.

6.1.1 Creating the required databases in DB2

We need the following databases in this topology:

- Common database (WPRCSDB)
- Business Process Choreographer (BPC) database
- Messaging engine database (MEDB) with three schemas
- Event database for Common Event Infrastructure (CEI)
- Business Process Choreographer reporting function (OBSVRDB)
- Business Space database (BSPCDB)

Refer to Table 5-1 on page 104 for details on database name and schema.

Use the instructions in 5.2.2, “Common database” on page 105, to create the common database (WPRCSDB).

Use the instructions in 5.2.3, “Business Process Choreographer database” on page 107, to create the BPC database.

Use the instructions in 5.2.4, “Process Observer database” on page 109, to create the Business Choreographer reporting function database (OBSVRDB).

Use the instructions in 5.2.5, “Messaging engine database resources” on page 111, to generate the DDL for each of the Messaging Engine schemas:

- BPCME
- CEIME
- SCASYS
- SCAAPP

And follow the instructions in 5.2.5, “Messaging engine database resources” on page 111, to create the ME database (MEDB).

Use the instructions in 5.2.6, “Event database” on page 112, to create the EVENT database.

Use the instructions in 5.2.7, “Business Space database” on page 112, to create the Business Space database (BSPCDB).

6.1.2 Create a Deployment Manager profile

There are two options to create a deployment manager profile:

- To create a deployment manager profile using the Profile Management Tool (the graphical option), see 5.3.1, “Deployment manager profile” on page 113.

> **Note:** You cannot use the Profile Management Tool to create or augment profiles on 64-bit platforms (except for i5/OS®) or on the Linux on System z® platform. To create profiles on these platforms, you must use the **`manageprofiles`** command. See more information about the **`manageprofiles`** command at:
>
> http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.websphere.wps.620.doc/doc/tins_manageprofiles_create.html

- To create a deployment manager profile silently, see 5.4.2, “Deployment manager profile” on page 129.

Remember to perform the post-creation changes. That is, add COMMONDB as the schema name and modify the SCA_Auth_Alias. The script changeDmgr.py can be used or the administrative console.

6.1.3 Create the custom node profile

There are two options to create custom node profiles:

- To create node profiles using the Profile Management Tool (the graphical option), see 5.3.2, “Node profiles” on page 123. See the above note regarding the restriction of using Profile Management Tool on 64-bit platforms.
- To create node profiles silently, see 5.4.3, “Node profiles” on page 131.

6.2 Configuring through the 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.

6.2.1 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.

1. Log in to the Integrated Solutions Console as any user. We are not using global security at this point.
2. Navigate to **Servers** → **Deployment Environments**. Click **New**. The system displays the first page of the Deployment Environment Configuration wizard with **Create a new deployment environment** selected, as shown in Figure 6-1.

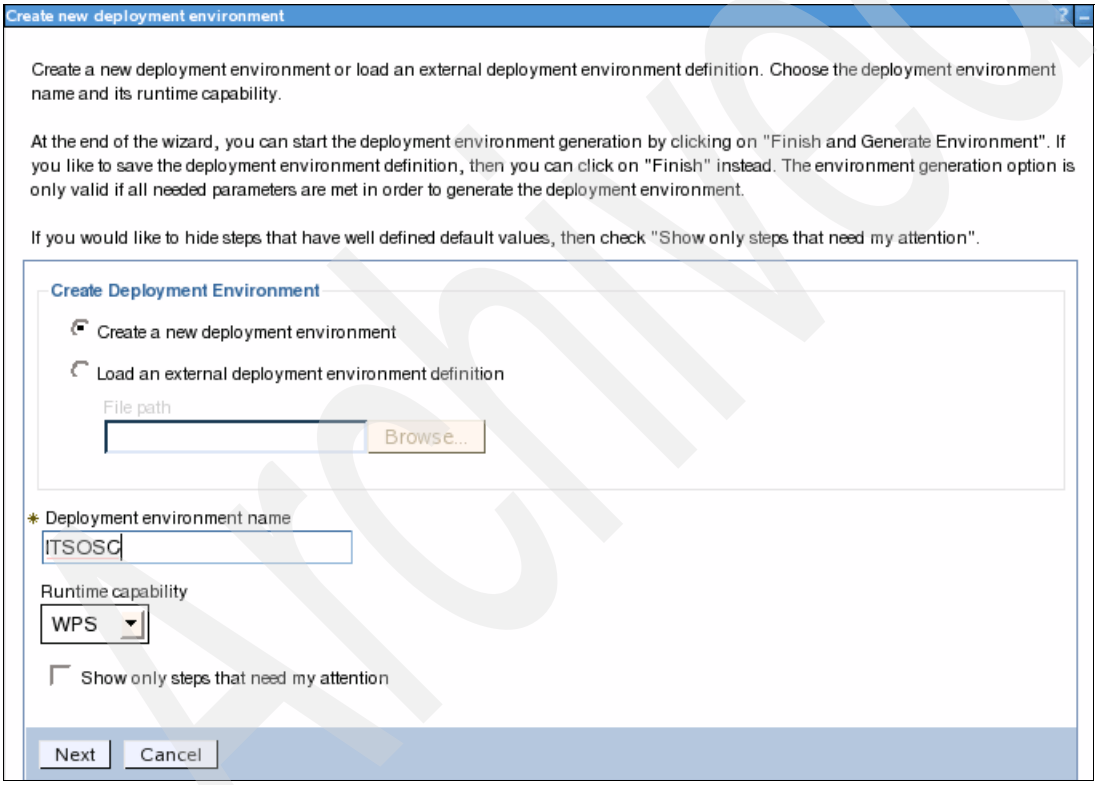


Figure 6-1 Create new deployment environment window

3. Enter `ITSOSC` (for example) in the Deployment environment name text box. Make sure that the Runtime capability drop-down box is set to **WPS**. Click **Next**. The Deployment Environment Patterns window opens (Figure 6-2).

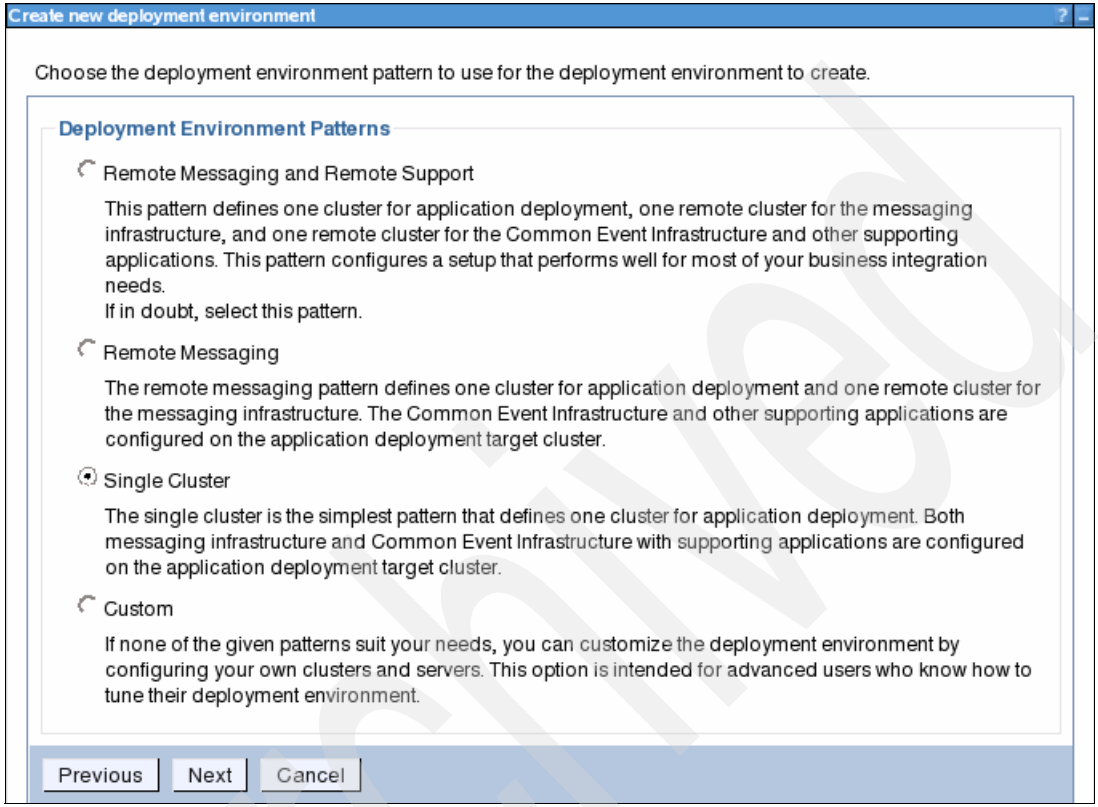


Figure 6-2 Deployment Environment Patterns window

4. Select the **Single Cluster** radio button. This is the *bronze* topology. Click **Next**. The Select Nodes window appears (Figure 6-3).

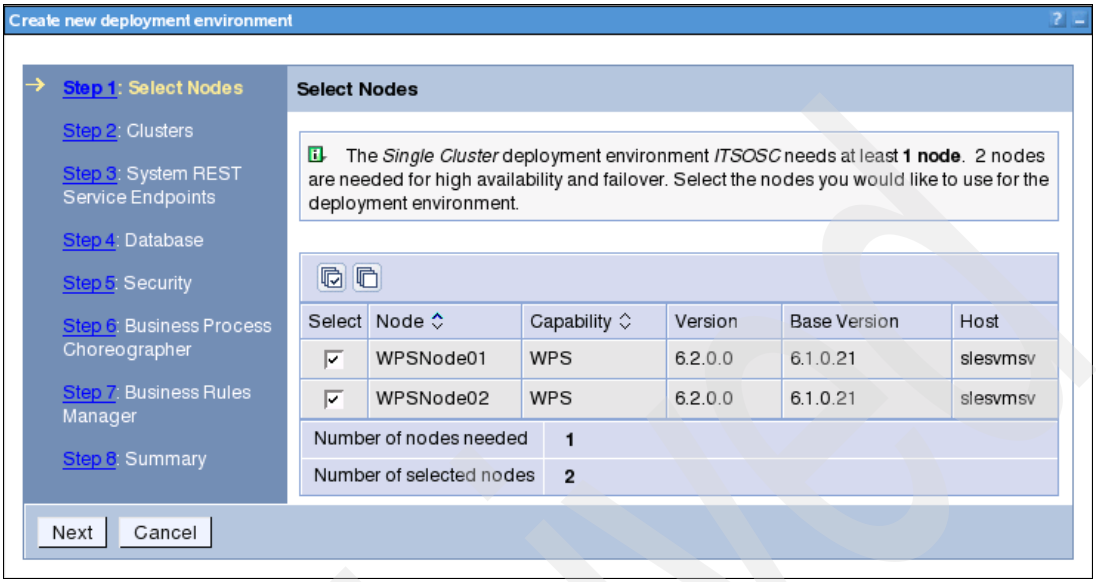


Figure 6-3 Select Nodes window

You will now see an 8-step process outlined beginning with *select nodes*. Our topology consists of just two nodes, so we use both of them, but in a larger environment you can select a sub-set of the entire node list.

5. Click the check box for both nodes and click **Next**. The Clusters window appears (Figure 6-4).

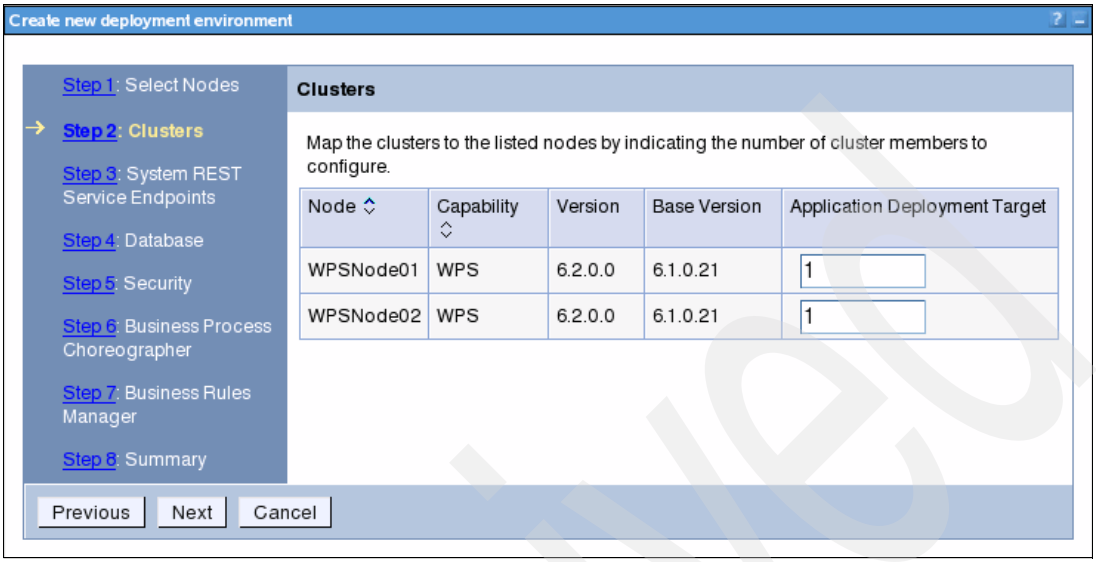


Figure 6-4 Clusters window, where one can choose the distribution of servers

The next window shows the distribution of the clusters. The messaging, application deployment target, and application support functions are contained in a single cluster. You will be creating one cluster with two servers in the cluster and one server per node.

6. Leave the values at the defaults, which gives you one server for the single cluster, ITSOSC.AppTarget on both nodes, and click **Next**. The System Representational State Transfer (REST) Endpoints configuration window is displayed, as shown in Figure 6-5.

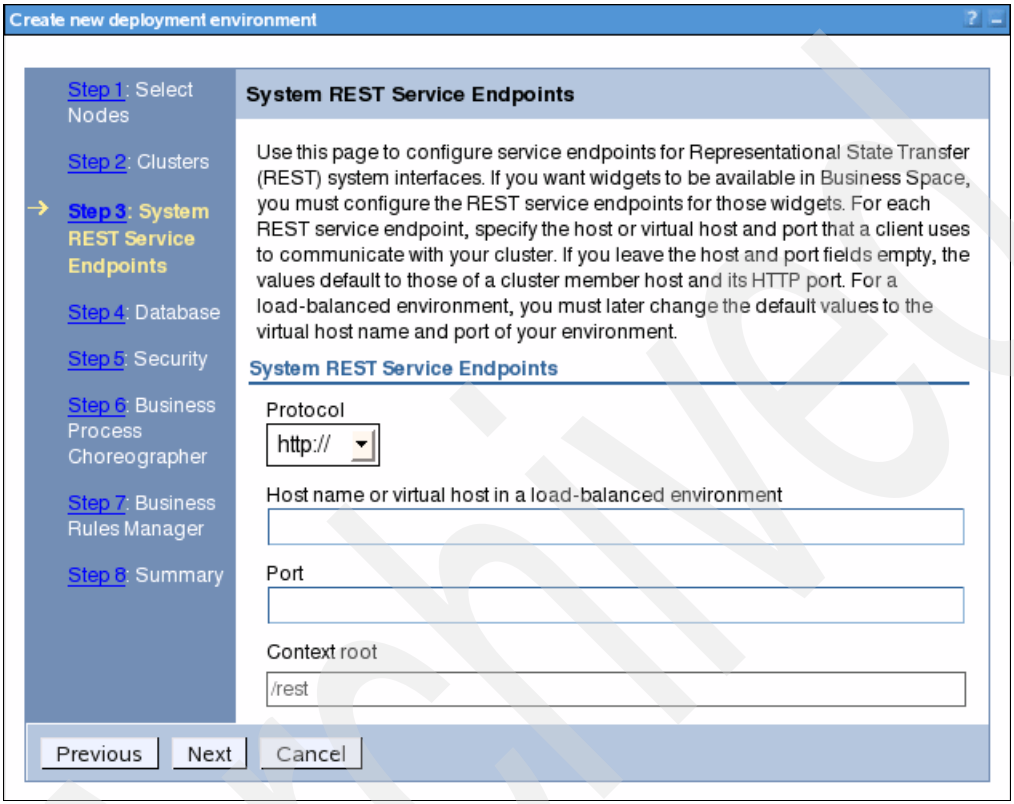


Figure 6-5 System REST Service Endpoints configuration window

The REST implementation allows for easy-to-use HTTP services that are language-independent and platform-independent, stateless, scalable, and easily parseable. The REST APIs are extremely useful for creating AJAX-style Web applications, in the same vein as the Business Space dashboards. Business Space dashboards enable a business user to visualize business performance data using various widgets. An illustration of Business Space powered by WebSphere can be found in more detail in the Chapter 13, “Using Business Space powered by WebSphere and Lotus Forms Client” on page 355.

7. Leave the values at the defaults, which means that the Host and Port fields are empty. Click **Next**. The Database window is displayed (Figure 6-6).

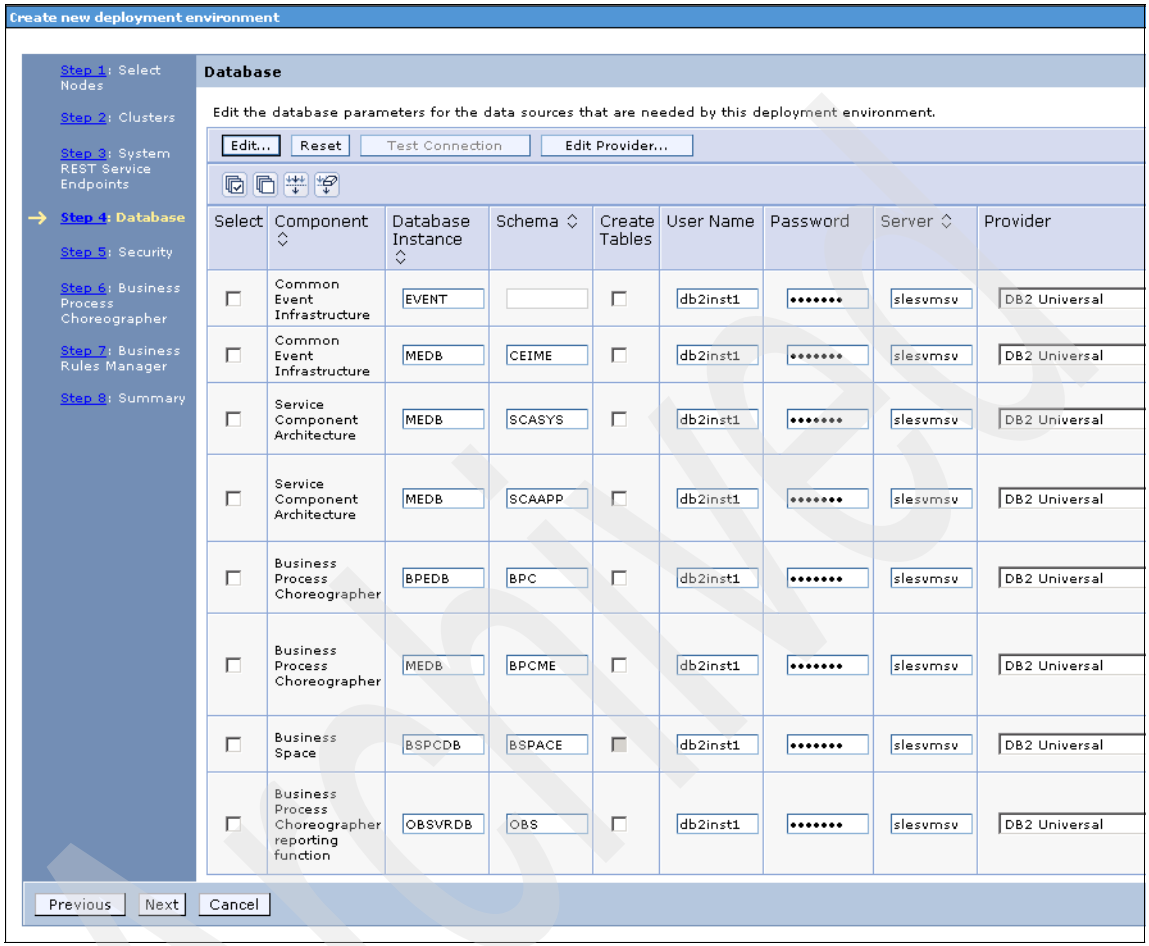


Figure 6-6 The Database configuration window

The Database window is the most complex, and care must be taken to edit this table correctly. Refer to Table 6-1 for a description of the fields and how they relate to the databases that you created earlier.

Table 6-1 Database instances

Database instance	Description	Comments
EVENT	Event server data source	This database does not exist yet. We create it after deployment of the topology. Note that this does not support a schema name.
MEDB	CEI Messaging Engine data source	Created earlier with schema CEIME.
MEDB	SCA System Bus Messaging Engine data source	Created earlier with schema SCASYS.
MEDB	SCA Application Bus Messaging Engine data source	Created earlier with schema SCAAPP.
BPEDB	Business Process Choreographer data source	Created earlier with schema BPC.
MEDB	Business Process Choreographer Messaging Engine data source	Created earlier with schema BPCME.
BSPCDB	Business Space data source	Created earlier with schema BSPACE.
OBSVRDB	Business Process Choreographer Event Collector data source	Created earlier with schema OBS.

8. Fill in the Database window form with the details shown in Table 6-1 on page 144. Figure 6-6 on page 143 does not show the full window details for space reasons, but the description of each value is given on the far right of the window. Make sure that the Create Tables column is cleared for each value and click **Next**. The security window is displayed in Figure 6-7.

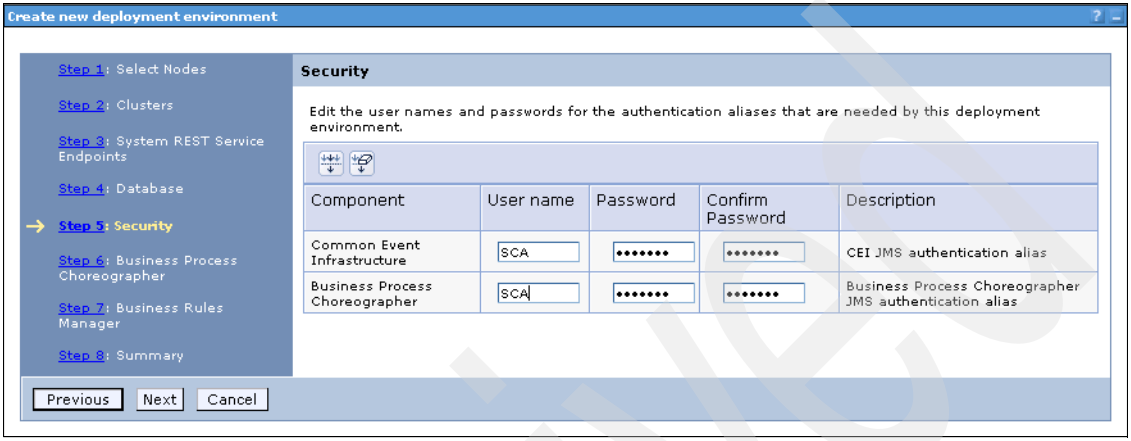


Figure 6-7 Security configuration window

9. Leave the user names for both CEI and BPC as SCA and enter a password. This user must be in LDAP later. Click **Next**. The Business Process Choreographer configuration window is displayed (Figure 6-8).

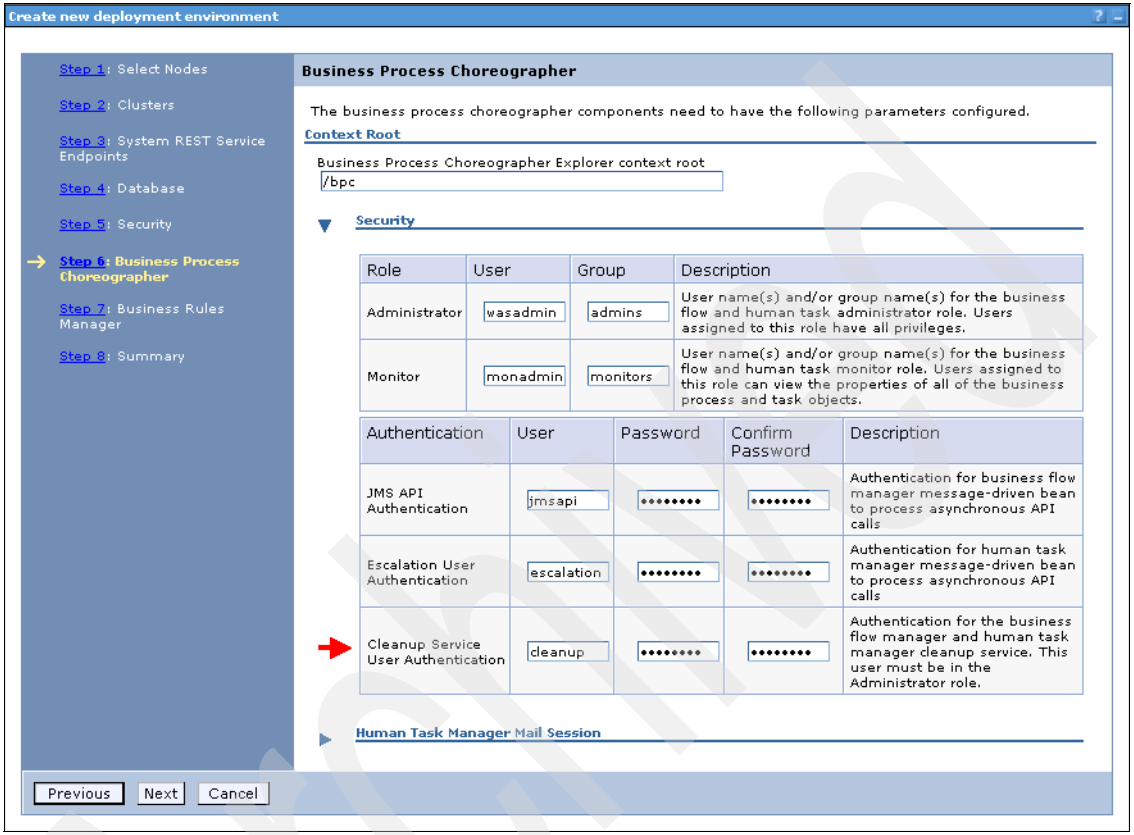


Figure 6-8 The Business Process Choreographer configuration window

Note: You want completed instances to be deleted automatically after keeping them for a while. New in WebSphere Process Server 6.2, you can use the Integrated Solutions Console to configure the cleanup service to schedule jobs that periodically delete eligible instances.

Cleanup User Authentication User and Password: Cleanup User is the run-as user ID for the Business Flow Manager and Human Task Manager cleanup service. This user must be in the business administrator role. More information about the cleanup service is available at:

`http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.websphere.bpc.620.doc/doc/bpc/tadmin_cleanup.html`

10. Perform the following steps to specify groups and users for authorization:
    a. For the administrator role, use the following values for the User and Group text boxes:
        - User: `wasadmin`
        - Group: `admins`
    b. For the monitor role, use the following values for User and Group text boxes:
        - User: `monadmin`
        - Group: `monitors`
    c. For the JMS API authentication, use the following values for the user and password:
        - User: `jmsapi`
        - Password: `passw0rd`
    d. For the escalation user authentication, use the following values for the user and password:
        - User:`escalation`
        - Password: `passw0rd`
    e. For the cleanup user authentication, use the following values for the user and password:
        - User:`cleanup`
        - Password: `passw0rd`

        When we enable LDAP, these users and groups must in the LDAP instance.
11. Clear the Enable e-mail service check box in the Human Task Manager Mail Session section because we will not be using human tasks with e-mail escalations. If you require this, you must also provide the other details.
12. Click **Next**. The Business Rules Manager window is displayed.
13. Click **Next**. The Summary window is displayed.
14. On the Summary panel, check your settings and click **Finish and Generate Environment**.

15. Save the changes. The Deployment Environments window (Figure 6-9) is displayed, showing the current status of our environment.

    If you hover the mouse over the status line you will see that it is not configured. This means that we have a definition of an environment but that no resources have been created yet.

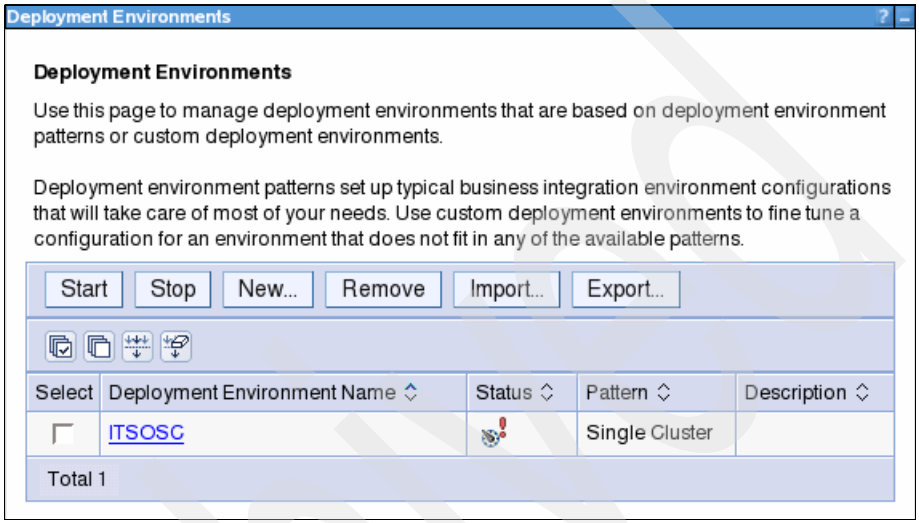


Figure 6-9 Deployment Environment status window

16. Click the **ITSOSC** link to display the Configuration window (Figure 6-10). This window shows the status of the single cluster that we have defined, which currently is not configured.

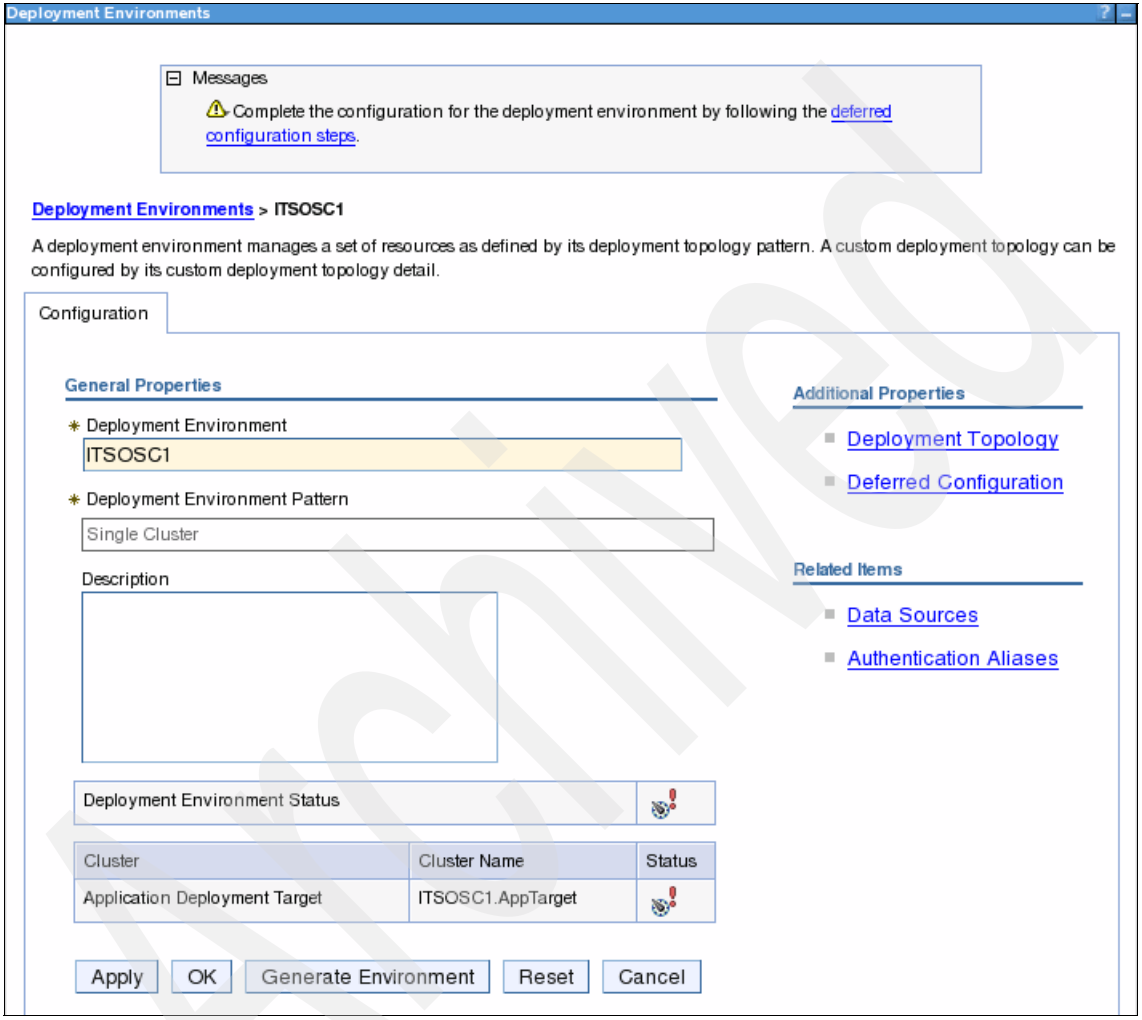


Figure 6-10 The Deployment Environments configuration window

17. Under Additional Properties, on the right side of the window, you can click **Deployment Topology** to see that the nodes are running but the clusters are not configured. Click **Cancel** to return to the Deployment Environments Configuration window.

18. Under Related Items, on the right side of the window, you can click **Data Sources** to show the database, schema and JNDI names that have been

defined. Click **Cancel** to return to the Deployment Environments Configuration window.

19. Click **Generate Environment**. A Configuration Status window is displayed, as shown in Figure 6-11.

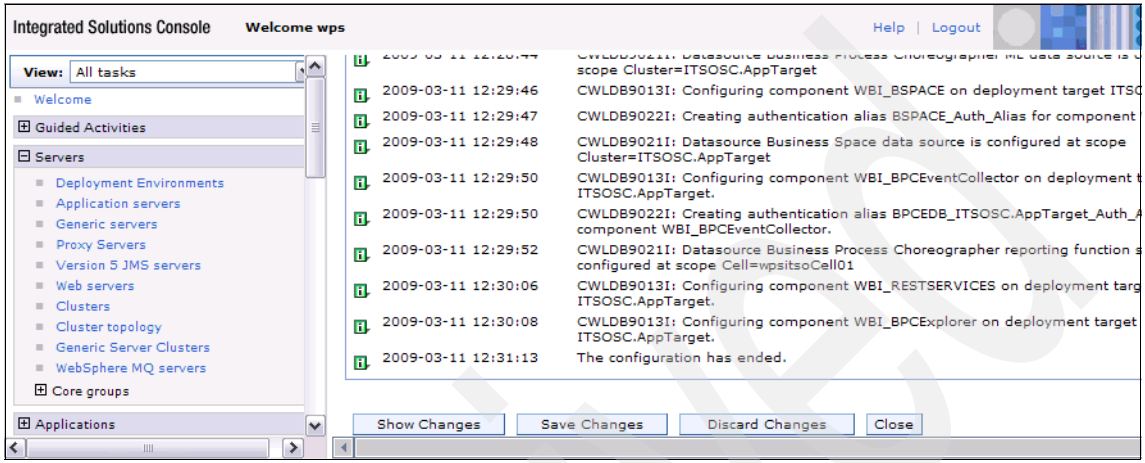


Figure 6-11 Completion of Generate Environment step

20. When complete, click **Save Changes**. The environment will now have a status of stopped (Figure 6-12).

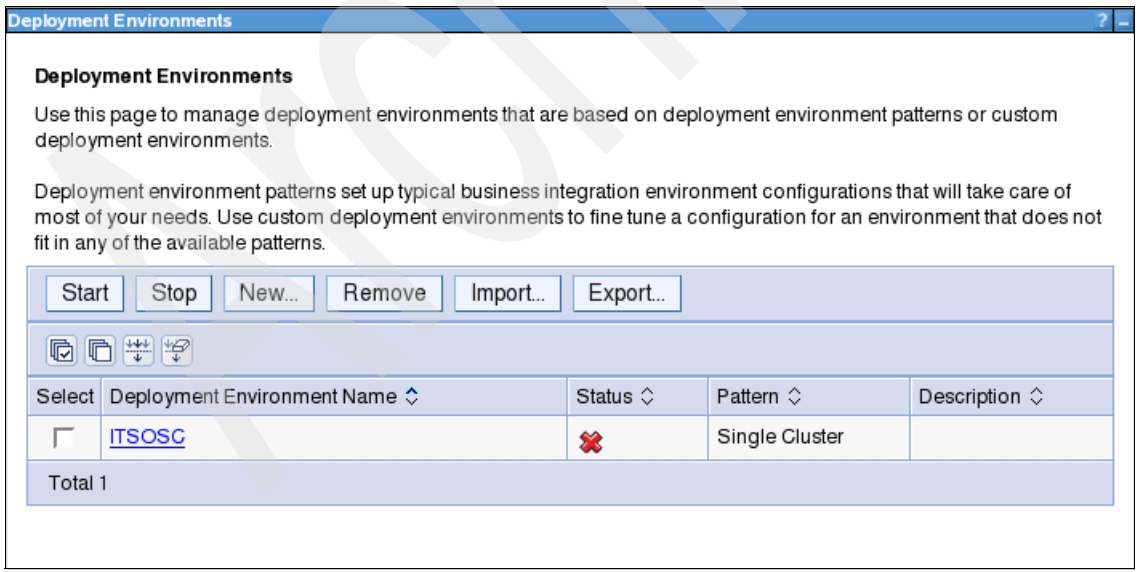


Figure 6-12 The Deployment Environments status window

6.2.2 Creating the event database tables

The final task before starting the environment is to create the event database (EVENT) tables. The scripts to do this are now available on the deployment manager under the deployment manager profile:

`/opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/databases/event/ITSOSC.AppTarget/dbscripts/db2`

The first task is to copy these scripts over to the DB2 system under the instance owner. The steps below assume that you have copied the files across into the home folder of the instance owner.

1. Log in to the DB2 System as the instance owner.
2. Change the directory to the scripts just copied, then run the following command:

   ```
   ./cr_event_db2.sh 2>&1 | tee output.log
   ```

3. Enter `1` for a server connection because we are on the DB2 system itself.
4. Enter the instance owner name.
5. Enter the instance owner password.
6. The database and tables will be created. Check the file `output.log` for any messages. If the script ran without any errors, you should see the messages shown in Example 6-1.

Example 6-1 cr_event_db2.sh script executed successfully

```
COMMIT
DB20000I  The SQL command completed successfully.

connect reset
DB20000I  The SQL command completed successfully.

The Event Service DB2 database EVENT created successfully.
```

Note: These DB2 commands may report various informational messages. This includes the following messages:

```
SQL0598W  Existing index "BPCME.SIB000PKIX" is used as the index
for the primary key or a unique key.  SQLSTATE=01550
```

Or:

```
SQL20189W  The buffer pool operation (CREATE/ALTER) will not take
effect until the next database startup due to insufficient
memory.  SQLSTATE=01657
```

These are not errors. You can ignore these messages.

7. Log off from the DB2 system.

6.2.3 Checking the database connectivity

Before you start the deployment environment you must check database connectivity:

1. Log in to the administrative console and navigate to **Resources** → **JDBC** → **JDBC Providers**. You will see that there are two providers now at different scopes (one for the cell and one for the ITSOSC.AppTarget cluster).
2. Click the first one where the scope is Cell=<cell name>. In our example it is WPSCell01. Under Additional Properties, click **Data sources**. You will see three data sources. Select the check box next to each data source and click **Test connection** to make sure that they have connectivity.

3. We now must define new variables. Navigate to **Environment** → **WebSphere variables** and select **Cluster=ITSOSC.AppTarget** as the scope. Click **New**. Create a new variable called DB2_UNIVERSAL_JDBC_DRIVER_PATH with the value /opt/ibm/WebSphere/ProcServer/universalDriver_wbi/lib, as shown in Figure 6-13. Click **OK** and then **Save** your changes.

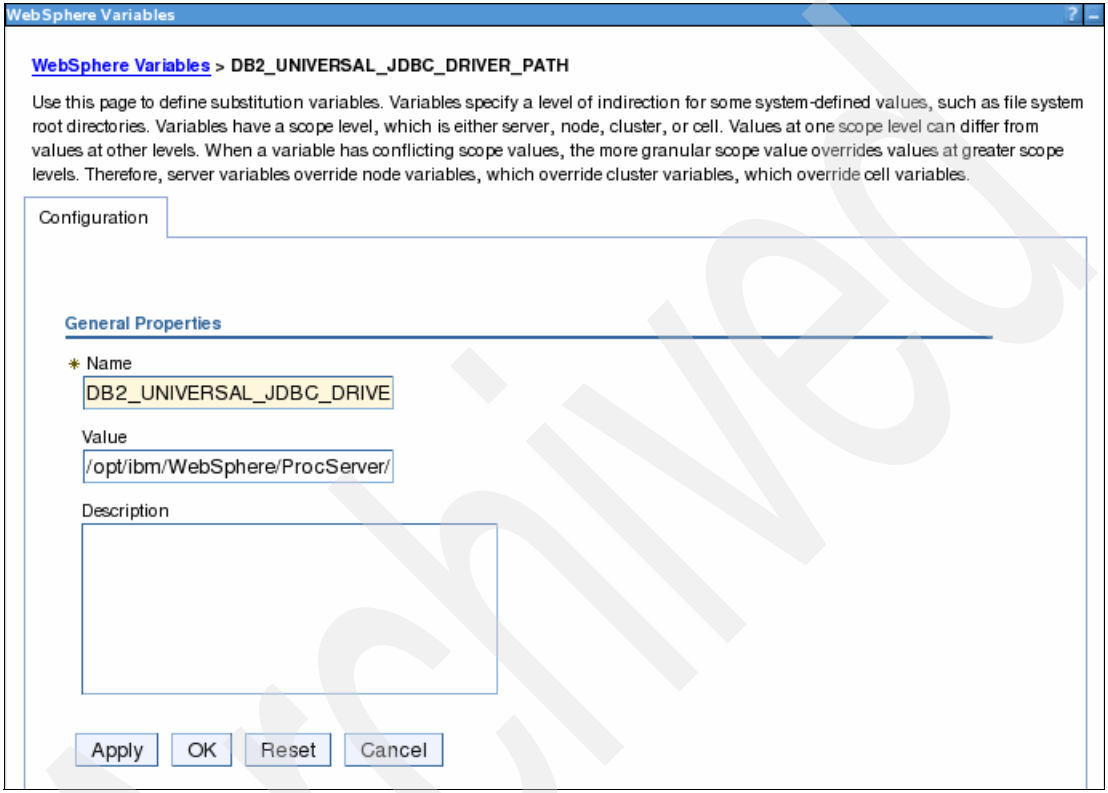


Figure 6-13 Creating new WebSphere variable window

4. Navigate to **System Administration** → **Save changes to master repository**. Click the **Synchronize changes with Nodes** check box, and click **Save**.
5. Navigate to **System Administration** → **Node agents**. Select both node agents and click **Restart**.

> **Note:** This may expire your login to the Integrated Solutions Console, so you may be required to log back in.

6. Navigate to **Resources** → **JDBC** → **JDBC Providers**. Click the provider that is shown as a link, at the scope Cluster=ITSOSC.AppTarget.

> **Note:** If you do not see this, make sure that the scope at the top of the page is set to All Scopes.

7. Under Additional Properties, click **Data sources**. You will see one data source for each of the schemas that we created earlier. In each case, we must make sure that the authentication alias is correctly set before we click **Test connection**.
8. Click **Business Process Choreographer ME data source** and scroll down the page until you reach a heading of Component-managed authentication alias. Select **BPCME_00_Auth_Alias** from the drop-down list. Under Authentication alias for XA recovery, select the **Use component-managed authentication alias** radio button, and click **OK**.
9. Save and synchronize the changes. Once saved, you should be returned to the Data sources page. You can now check the connectivity by selecting the **Business Process Choreographer ME data source** check box and clicking **Test connection**.
10. Perform the same actions for the other three data sources using the values shown in Table 6-2.

Table 6-2 Authentication aliases for messaging engines

Data source name	Authentication alias
CEI ME data source	CEIME_ITSOSC.AppTarget_Auth_Alias
SCA Application Bus ME data source	SCAAPPME_00_Auth_Alias
SCA System Bus ME data source	SCASYSME_00_Auth_Alias

6.2.4 Installing a Web server

Install a Web server as described in 5.6.1, “Add a Web server to the administrative console” on page 133.

6.2.5 Completing the deployment environment configuration

In this section you will start the deployment environment. Perform the following steps to complete the topology configuration:

1. Log in to the administrative console.
2. Navigate to **Servers** → **Deployment Environments**. Click the **ITSOSC** link (which is currently stopped).
3. Under Additional Properties, click **Deferred Configuration**. A list of tasks required to complete the configuration is displayed. Because we created all the databases before starting any deployment and have just finished the configuration of the event database, these tasks have been completed. Click **Configuration Done**, save the changes, and click **Close**.
4. Navigate to **Servers** → **Deployment Environments**. Select the **ITSOSC** check box and click **Start**. The ITSOSC deployment environment will immediately change to Started, but you must wait while the application servers start.
5. Navigate to **Servers** → **Clusters** and you will see that the server cluster's state is now Partial Start (Figure 6-14).

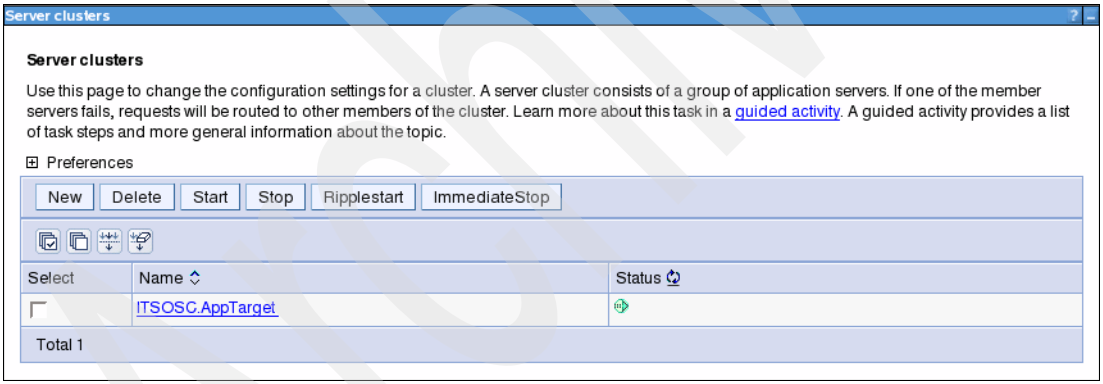


Figure 6-14 The Server Clusters window

6.2.6 Completing and verifying the configuration

You are now ready to complete the configuration and verify it. For instructions on completing and verifying the configuration, see 6.3, "Post-creation configuration and verification" on page 156.

6.3 Post-creation configuration and verification

In this section we add functonality to the deployment topology and demonstrate simple checks to perform to verify that the topology was created successfully.

6.3.1 Configuring CEI logging

To configure CEI logging:

1. In the Integrated Solutions Console, navigate to **Servers** → **Clusters** → **ITSCSO.AppTarget**.
2. On the right-hand side, under Business Integration, expand **Business Process Choreographer** and 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 or Human Task Manager check boxes, or both, depending on your requirements.
4. Save and synchronize your changes.

6.3.2 Configuring shared transaction logging

This section introduces considerations for shared transaction logging. It contains the following sections:

- High availability considerations for the transaction manager
- Create the shared directories for the transaction logs
- Changing the transaction manager log settings
- Policies for transaction manager peer recovery

High availability considerations for the transaction manager

The WebSphere Application Server transaction manager (used by WebSphere Process Server) writes to its transaction recovery logs when it handles global transactions (XA transactions) that involve two or more resources. Transaction recovery logs are stored on disk and are used for recovering in-flight transactions from system crashes or process failures. By default, each cluster member maintains its own transaction log.

To keep the transaction logs highly available and to enable transaction peer recovery, you must place the recovery logs on a highly available file system, such as IBM SAN FS or NAS, for all the application servers within the same cluster to access. All application servers must be able to read from and write to the logs. In

addition to configuring a highly available file system, you must decide whether to use automated or manual peer recovery for the transaction manager. In either case transaction manager policies must also exist.

For more details on high-availability considerations for the transaction logs, refer to the IBM Redbooks publication *WebSphere Application Server Network Deployment V6: High Availability Solutions*, SG24-6688.

Create the shared directories for the transaction logs

Once you have decided on a highly available file system, you must configure the transaction log directory setting for each server in the cluster. You can configure the location of the transaction log directory using either the Integrated Solutions Console or commands. The configuration is stored in the `serverindex.xml` `node-level` configuration file.

Each server must be able to access the log directories of other servers in the same cluster. For this reason, do not leave this setting unset. If you do not set a directory, the application server assumes a default location within the appropriate profile directory, which might not be accessible to other servers in the cluster.

Each server in the cluster must also have a unique transaction log directory, to avoid attempts by multiple servers to access the same log file. For example, you could use the name of each server as part of the log directory name for that server.

To set the transaction log directory for the cluster members:

1. In the administrative console, expand **Servers** and click the **Clusters** link.
2. Click the check box for the cluster that you wish to modify and click **Stop**.

3. Once the cluster is stopped, click the link for the cluster that you wish to modify. Figure 6-15 shows the transaction log settings for the AppTarget cluster and its members.
4. In the Additional Properties section, click the **Cluster members** link.
5. Click the link for the first cluster member.
6. In the Container Settings section, expand **Container Services** and click the **Transaction Service** link (Figure 6-15).

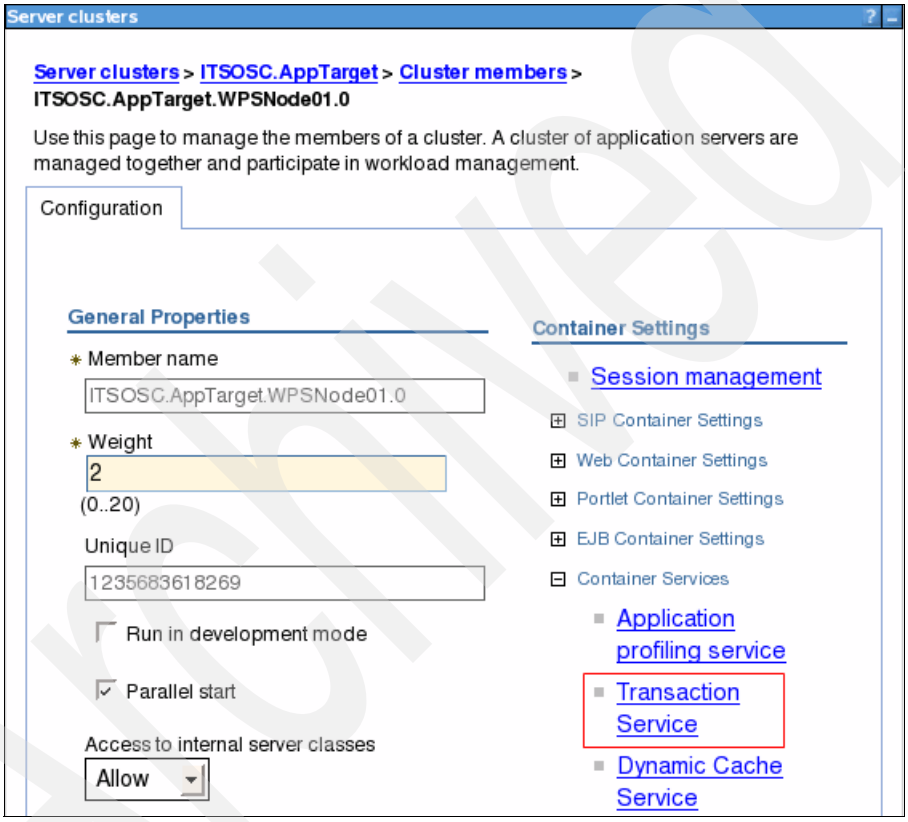


Figure 6-15 Transaction Service link from Container Settings

7. In the General Properties section, enter an appropriate value in the Transaction log directory text box. See Figure 6-16.

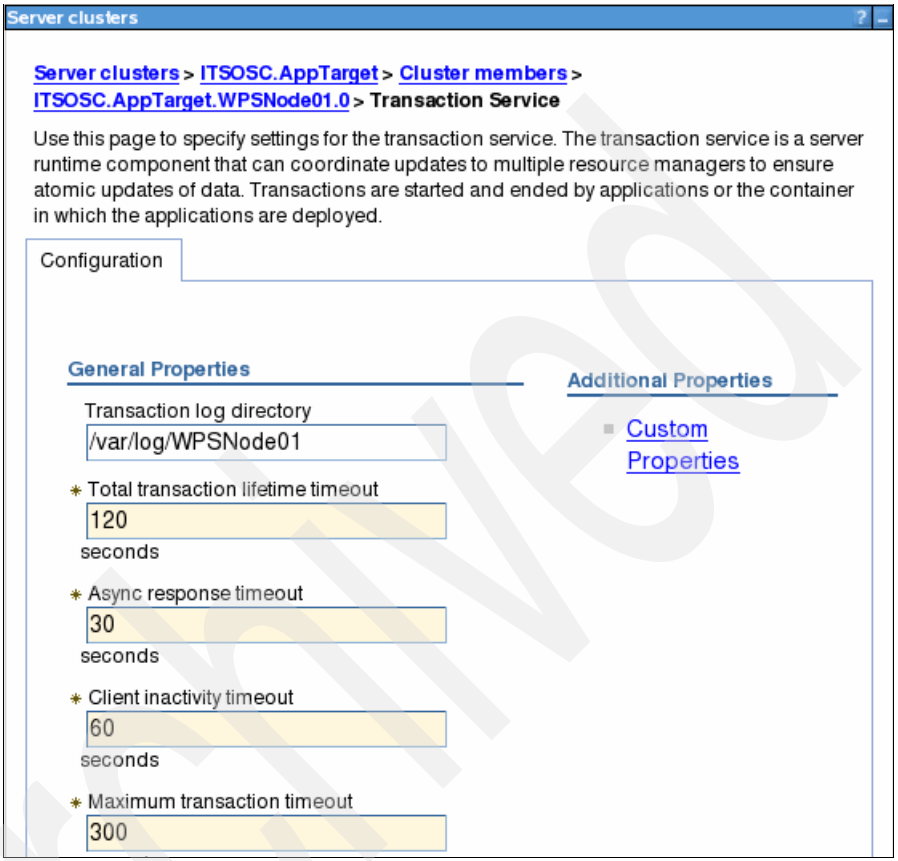


Figure 6-16 Transaction log directory

> **Tip:** If you are using NFS, it is advisable to use the hard option in the NFS mount command (**`mount -o hard`**) to avoid data corruption.

Click **OK**.

8. Save the changes to the master configuration.
9. Wait for automatic synchronization to complete and click **OK**, or manually synchronize the nodes.
10. Copy the existing transaction logs to the shared file system. Make sure that the location and file permissions are correct.

Changing the transaction manager log settings

Once you have configured the transaction log location for the cluster members, you must enable transaction log failover for the cluster.

To enable transaction log recovery:

1. In the administrative console, expand **Servers** and click the **Clusters** link.
2. Click the link for the cluster that you wish to modify (the following figures show the transaction log settings for the AppTarget cluster and its members).
3. In the Configuration tab, in the General Properties section, click the **Enable failover of transaction log recovery** check box. See Figure 6-17.

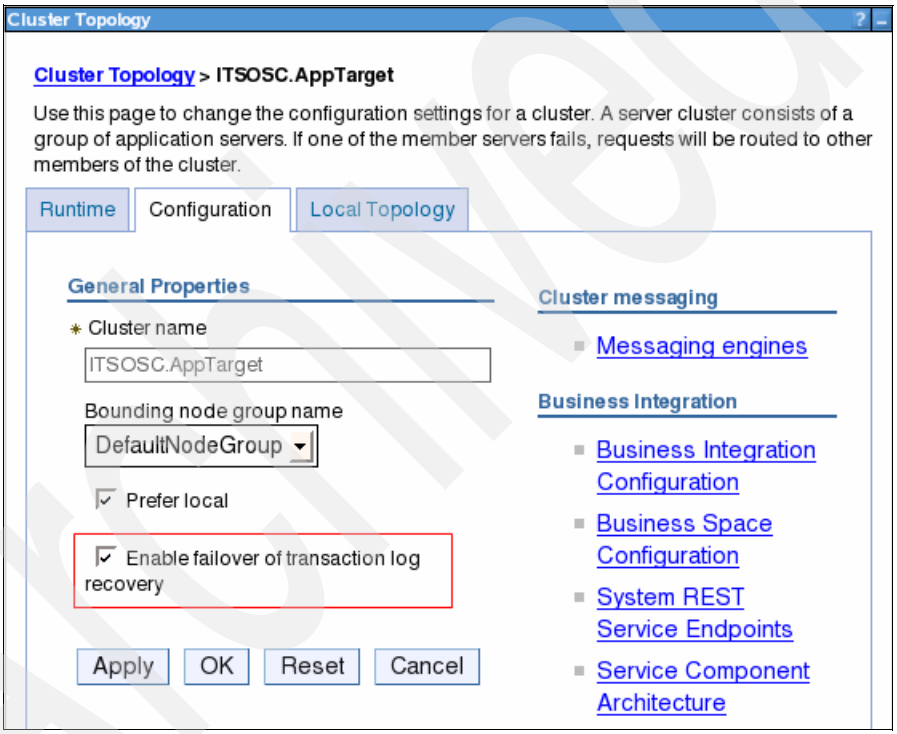


Figure 6-17 Transaction log recovery failover enablement

Click **OK**.

4. Save the changes to the master configuration.
5. Wait for automatic synchronization to complete and click **OK**, or manually synchronize the nodes.
6. Start the cluster.

Policies for transaction manager peer recovery

In order for transaction log failover to work correctly, you must have one or more policies in place. In each WebSphere Process Server deployment, a default transaction manager policy is created to control failover of the transaction manager service. This policy is a *one of n policy* similar to the policies created for the messaging engines in Chapter 11, “Advanced production topologies” on page 269.

A *one of n policy* means that only one server in a cluster can run the transaction manager service at any given time. If the running transaction manager service fails, the default transaction manager policy, called *Clustered TM Policy*, specifies that the service can fail over to another cluster member. The default policy also enforces failback. If the failed transaction manager becomes available, the transaction manager service will fail back to it.

If you are using automated failover, the default transaction manager policy is likely sufficient for your needs. To examine the default transaction manager policy:

1. In the administrative console, expand **Servers** → **Core groups**.
2. Click the **Core group settings** link.
3. Click the **DefaultCoreGroup** link.
4. In the Additional Properties section, click the **Policies** link.
5. Click the link for **Clustered TM Policy**. See Figure 6-18.

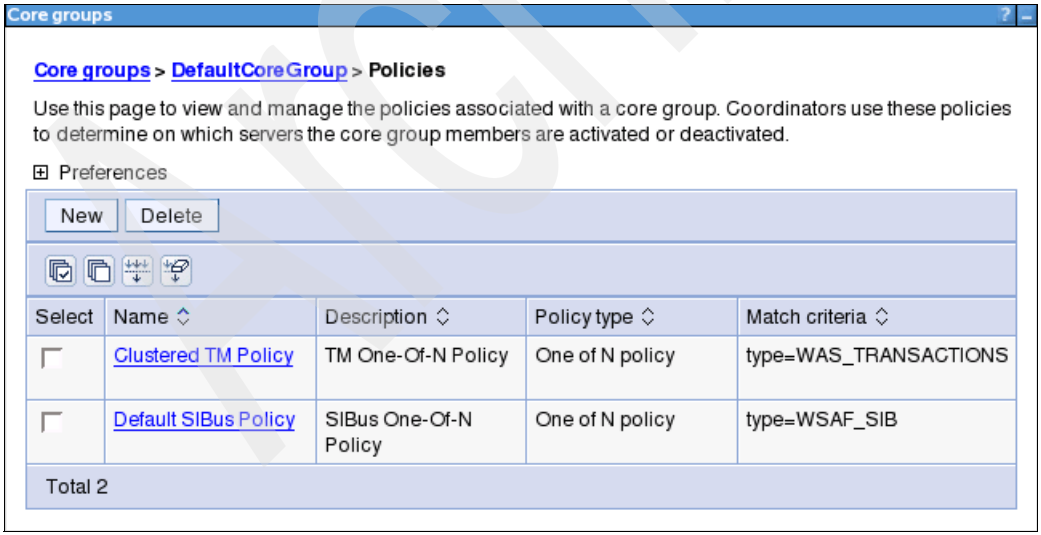


Figure 6-18 Transaction Manager policy window

6. Examine the properties of the policy. See Figure 6-19.

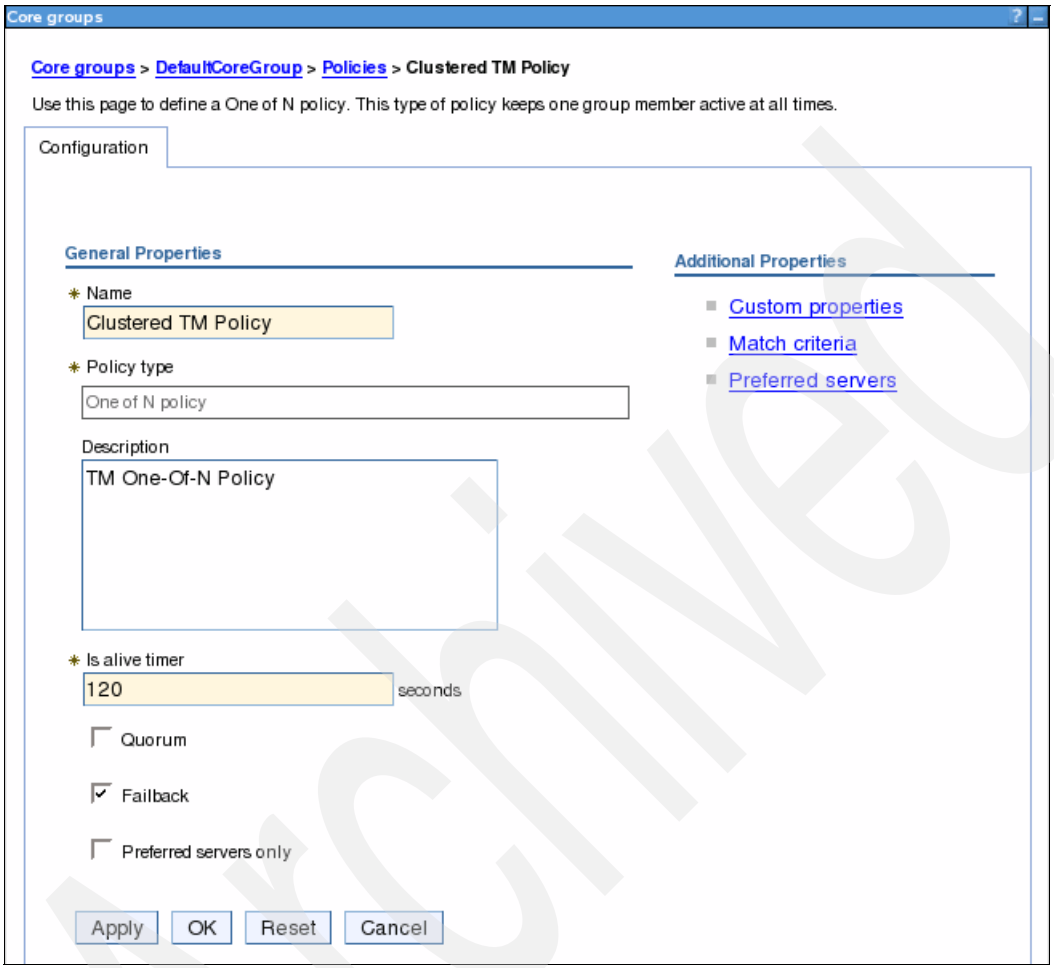


Figure 6-19 Default transaction manager policy

7. Click **Cancel**.

6.3.3 Installing the sample application

This publication provides a sample vehicle loan process created for the fictitious company ITSOBank. You can use this vehicle loan process to test the topology that you have built in this chapter. For more information about the vehicle loan process, refer to Chapter 2, “Sample business application scenario used in topologies” on page 23. To obtain the additional material supplied with this book, refer to Appendix A, “Additional material” on page 597.

When you have obtained the additional materials, navigate to the `Scenarios\WPS\EAR\v6.2_Plain` directory. From here, copy the `ITSOApp.ear` and `ITSO_implApp.ear` files to the deployment manager. Installation follows the normal process and is described here:

1. Navigate to **Applications** → **Install New Application**.
2. Choose **Remote file system**, click **Browse**, and navigate to the location of the uploaded EAR files.
3. Click the **ITSO_implApp.ear** radio button and click **OK**.
4. Select the radio button **Prompt me only when additional information is required**, as shown in Figure 6-20. Click **Next**.

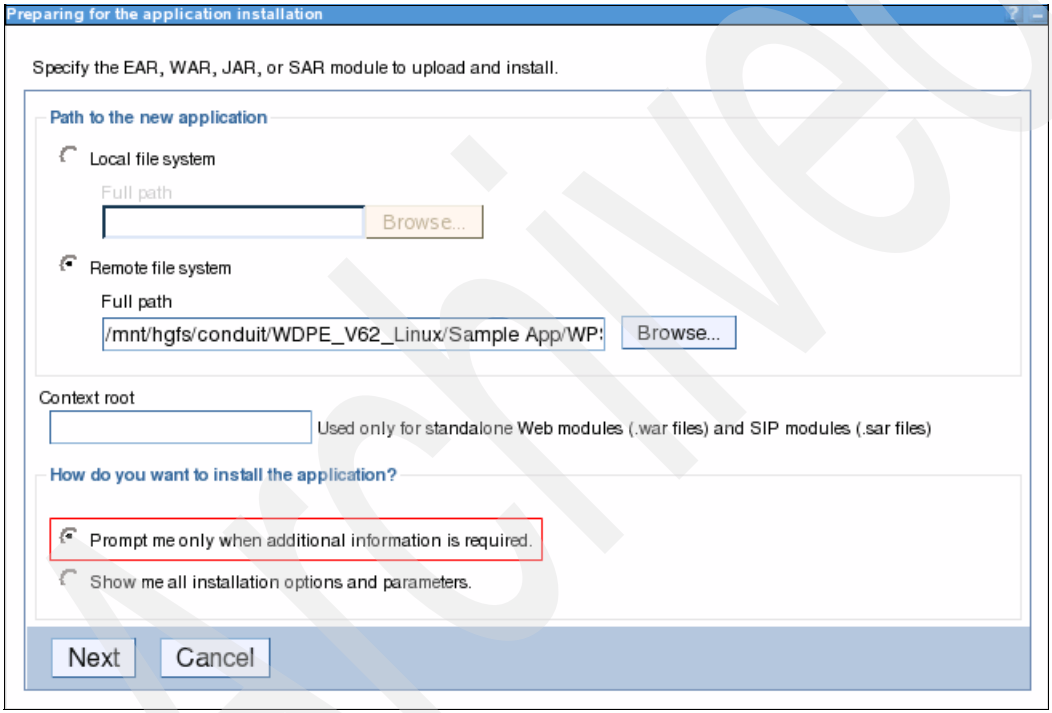


Figure 6-20 Preparing for application install window

5. On Step 1: Select installation options, click **Next**.
6. On Step 2: Map modules to servers, select **WebSphere:cell=slesvmsvCell01,cluster=ITSOSC.AppTarget**, select the **ITSO_implWeb** check box, and click **Apply**, then click **Next**.
7. On Step 3: Summary click **Finish**.

8. When you see the message `Application ITSO_implApp installed successfully`, click the **Save** link, then **OK**.

Repeat this process for the `ITSOApp.ear` file.

You can check that the Web server plug-in file is correctly updated:

1. Navigate to **Servers** → **Web servers** and click **webserver1**.
2. Under Additional Properties, click **Plug-in properties**.
3. Under Plug-in properties, click the **View** button.
4. Scroll down the page and you should see the lines shown in Example 6-2.

Example 6-2 Plug-in details showing ITSO application URL

```
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
Name="/ITSO_implWeb/*"/>
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
Name="/ITSOWeb/*"/>
```

5. Navigate to **Servers** → **Web servers**. Select the **webserver1** check box and click **Propagate Plug-in**.
6. Navigate to **Applications** → **Enterprise Applications**. Select the **ITSOApp** and **ITSO_implApp** check boxes, and click **Start**.
7. Log in to the Business Process Choreographer console. In our environment we used the following URL:

   `http://itsodmgr/bpc`

8. Click **My Process Templates**, select **New Loan Process**, and click **Start Instance**. Provide some test input data and click **Submit**. This should launch the business process. If the environment is working correctly, it returns a response.

To uninstall this or any other enterprise application from the server, issue the following commands:

```
cd /opt/IBM/WebSphere/ProcServer/ProcessChoreographer/admin

../../bin/wsadmin.sh -lang jacl -f bpcTemplates.jacl -uninstall
"<AppName>" -force
```

Note: We do not recommend using the -force option in a production environment.

7

Configuring Remote Messaging and Remote Support

This chapter provides full instructions for creating a *gold* topology, which uses the Remote Messaging and Remote Support (RMRS) Deployment Environment pattern for WebSphere Process Server V6.2. In this topology you create three clusters:

- An Application Cluster to support WebSphere Process Server applications and mediations
- A Messaging Cluster to support the messaging engine infrastructure
- A 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, formerly known as BPC Observer

These clusters are configured over two nodes, and each node has a single cluster member.

Furthermore, you can create the topology using two distinct methods:

- Through the administrative console and template guided activities
- Using wsadmin scripting

7.1 Prerequisites for creating the RMRS topology

This chapter builds on top of the work that is already described in the previous chapters. A general overview of the steps required in creating this topology consists of the following sections:

- Installing the products
- “Creating the required databases in DB2” on page 166
- “Create a Deployment Manager profile” on page 167
- “Create the custom node profile” on page 167
- “Creating a deployment environment topology” on page 169
- Generate the environment
- Test and verify the topology

7.1.1 Creating the required databases in DB2

In this section the databases are created before beginning the process of creating the topology. In the single server topology, the databases are local. In the RMRS topology, the databases are remote from the WebSphere Process Server farm. In the RMRS topology, the work of creating these databases is normally carried out by the database team.

You need the following databases in this topology:

- Common database (WPRCSDB)
- Business Process Choreographer (BPC) database
- Messaging engine database (MEDB with three schemas)
- Event database for Common Event Infrastructure (CEI)
- Business Process Choreographer reporting function (OBSVRDB)
- Business Space database (BSPCDB)

Refer to Table 5-1 on page 104 for details on database name and schema.

Use the instructions in 5.2.2, “Common database” on page 105, to create the common database (WPRCSDB).

Use the instructions in 5.2.3, “Business Process Choreographer database” on page 107, to create the BPC database (BPEDB).

Use the instructions in 5.2.5, “Messaging engine database resources” on page 111, to generate the DDL for each of the Messaging Engine schemas:

- BPCME
- CEIME
- SCASYS
- SCAAPP

Follow the instructions in 5.2.5, “Messaging engine database resources” on page 111, to create the ME database (MEDB).

Use the instructions in 5.2.6, “Event database” on page 112, to create the CEI database (EVENT).

Use the instructions in 5.2.4, “Process Observer database” on page 109, to create the Business Process Choreographer reporting function database (OBSVRDB).

Use the instructions in 5.2.7, “Business Space database” on page 112, to create the Business Space database (BSPCDB).

Create the Business Space database

Log in to the DB2 system as the instance owner, then run the following commands:

```
db2 "CREATE DATABASE BSPCDB USING CODESET UTF-8 TERRITORY en-us"
db2 connect to BSPCDB USER db2inst1 using 'dbpass'
db2 connect reset
```

7.1.2 Create a Deployment Manager profile

There are two options for creating a deployment manager profile:

- To create a deployment manager profile using the profile management tool (the graphical option), see 5.3.1, “Deployment manager profile” on page 113.
- To create a deployment manager profile silently, see 5.4.2, “Deployment manager profile” on page 129.

Remember to perform the post-creation changes. That is, add COMMONDB as the schema name and modify the SCA_Auth_Alias. The script changeDmgr.py can be used or the administrative console.

7.1.3 Create the custom node profile

There are two options for creating custom node profiles:

- To create node profiles using the profile management tool (the graphical option), see 5.3.2, “Node profiles” on page 123.
- To create node profiles silently, see 5.4.3, “Node profiles” on page 131.

7.2 Configuring the topology using the Integrated Solutions Console

This section contains instructions for configuring and deploying a near-production quality Remote Messaging and Remote Support topology. The topology includes a database server (DB2), an LDAP server, and two nodes to provide the clustering required. The databases hosted use other schema names (rather than the default). We do not show how to make various components highly available using technologies such as HACMP™ or HADR.

7.2.1 Creating a deployment environment topology

This section describes how to create a Remote Messaging and Remote Support topology using the Integrated Solutions Console. Before beginning, ensure that the deployment manager and nodes are running.

1. Log in to the administrative console as any user. We are not using global security at this point.
2. Navigate to **Servers** → **Deployment Environments**. Click **New**. The Create new deployment environment window opens (Figure 7-1).

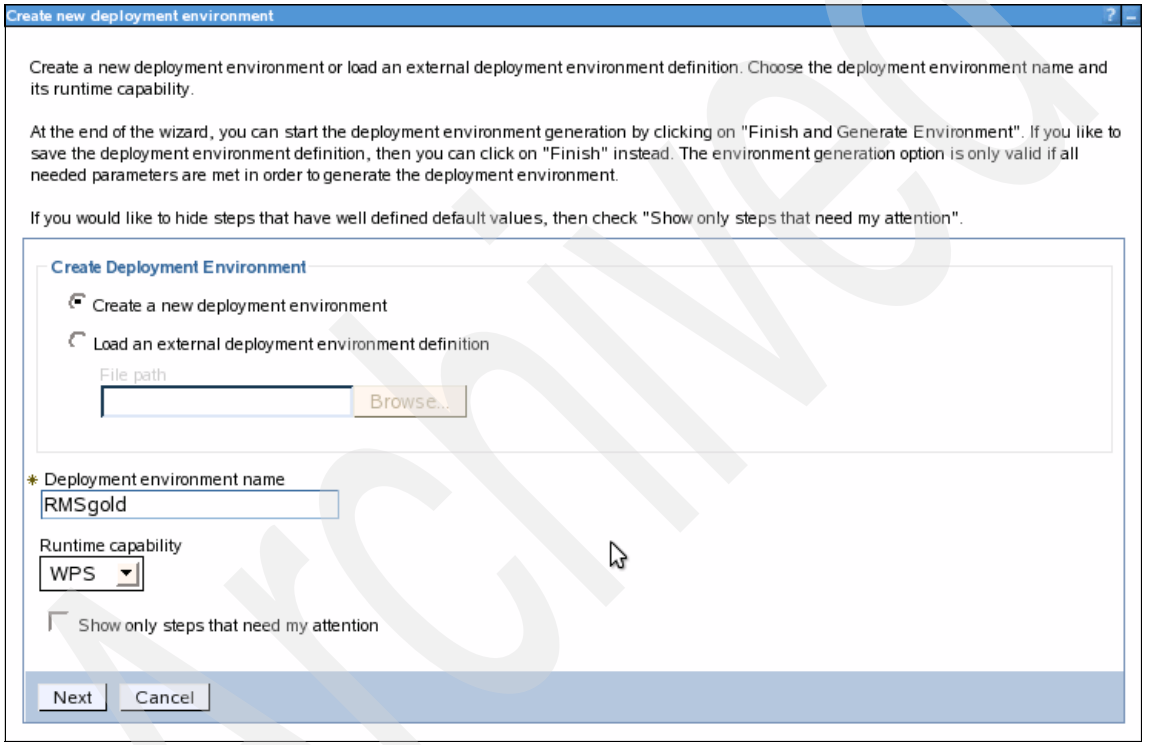


Figure 7-1 Create new deployment environment window

3. Leave the Create a new deployment environment radio button selected. Enter `RMSgold` in the Deployment environment name text box. Make sure that Runtime capability is set to WPS, and click **Next**. The Deployment Environment Patterns window opens (Figure 7-2).

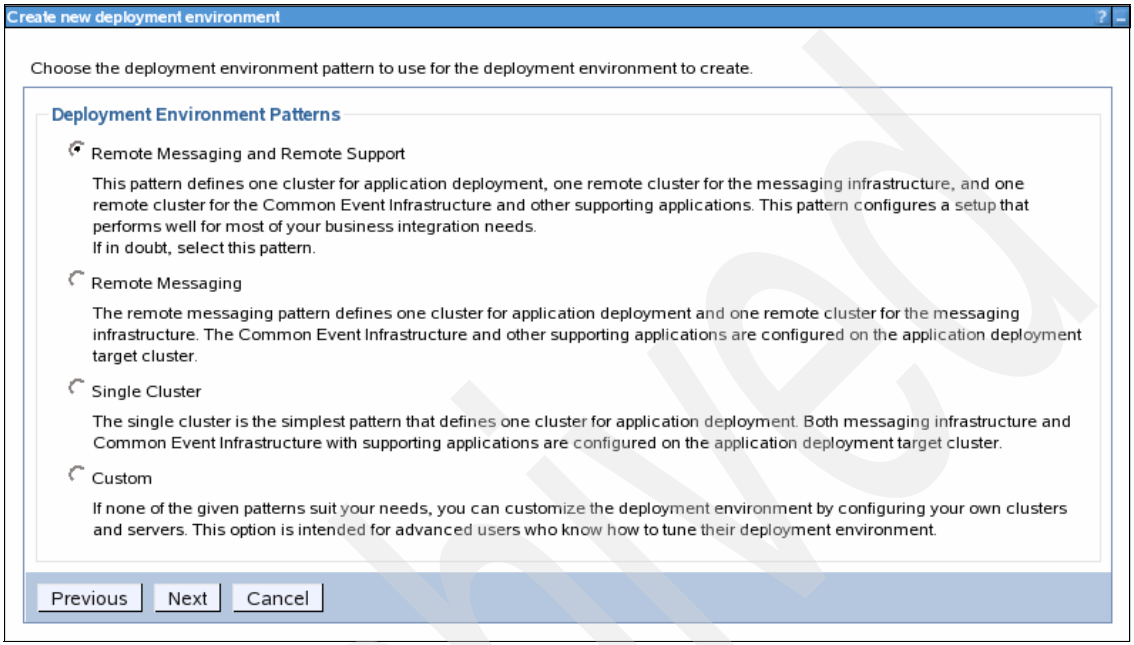


Figure 7-2 Deployment Environment Patterns window

4. Select the **Remote Messaging and Remote Support** radio button. This is the gold topology. Click **Next**. The Select Nodes window appears (Figure 7-3).

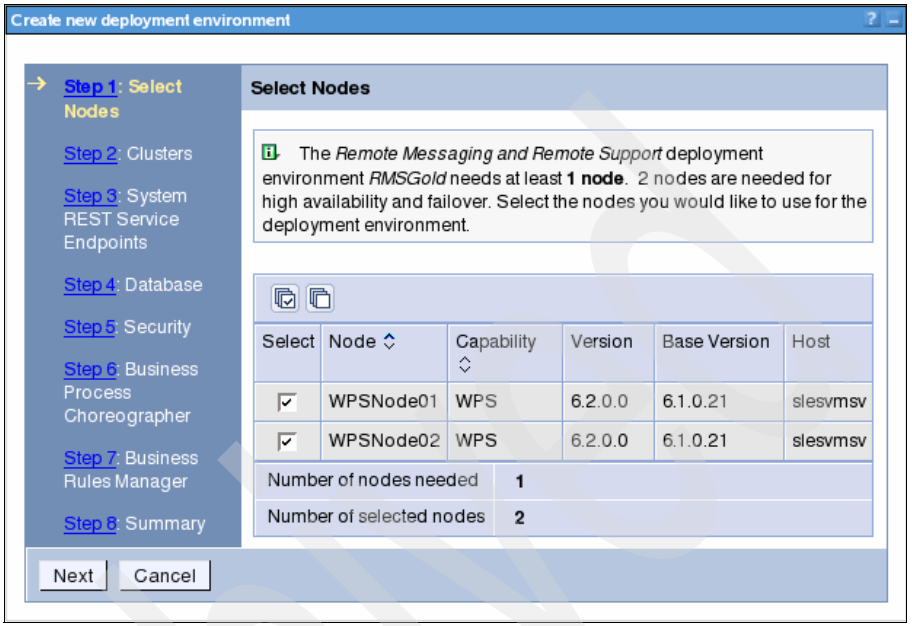


Figure 7-3 Select Nodes window

You will now see an 8-stage process outlined beginning with *select nodes*. Our topology consists of just two nodes, so you will use both of them, but, in a larger environment, you can select a sub-set of the entire node list.

5. Click the check box for both nodes and click **Next**. The Clusters window appears (Figure 7-4).

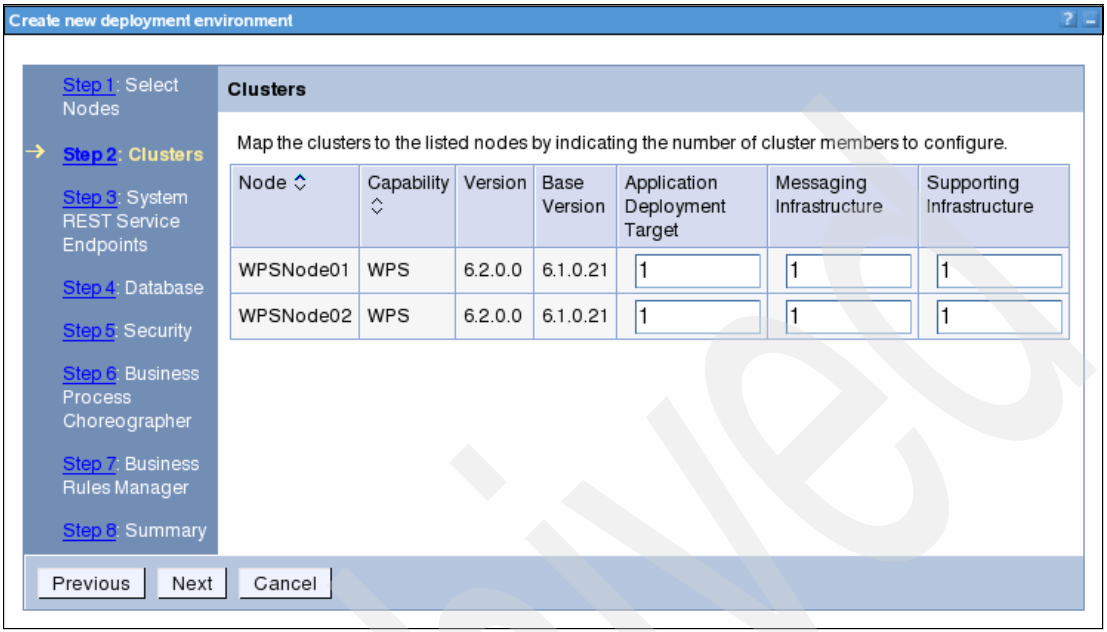


Figure 7-4 Clusters window, where you can map the nodes within the clusters

This window shows the distribution of the clusters. The Remote Messaging and Remote Support topology has three clusters:

- Application Deployment Target is the cluster for WebSphere Process Server applications.
- Messaging Infrastructure is the cluster for messaging engines.
- Supporting Infrastructure is the cluster for CEI and other services.

In this example, you create three clusters with one server in each cluster and one server per node.

6. Leave the values at the defaults, which gives us one server for each cluster on both nodes, and click **Next**. The System REST Service Endpoints window appears (Figure 7-5).

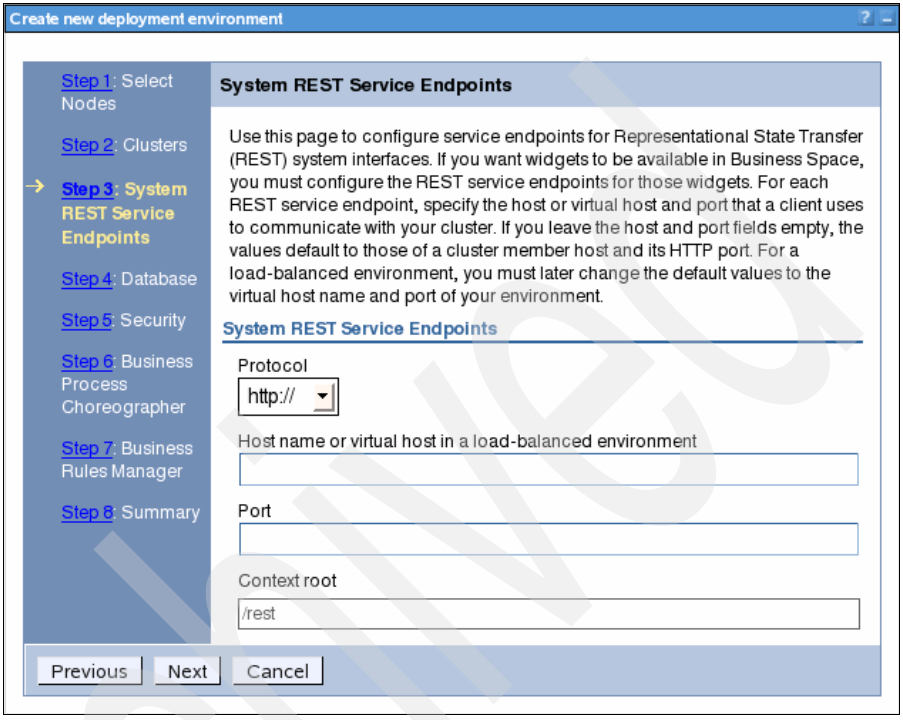


Figure 7-5 System REST Service Endpoints window

The REST implementation allows for easy-to-use HTTP services that are language-independent and platform-independent, stateless, scalable, and easily parseable. The REST APIs are extremely useful for creating AJAX-style Web applications, in the same vein as the Business Space dashboards. Business Space dashboards enable a business user to visualize business performance data using various widgets. An illustration of Business Space powered by WebSphere can be found in Chapter 13, “Using Business Space powered by WebSphere and Lotus Forms Client” on page 355.

7. Leave the values at the defaults and click **Next**. The System Database window appears (Figure 7-6).

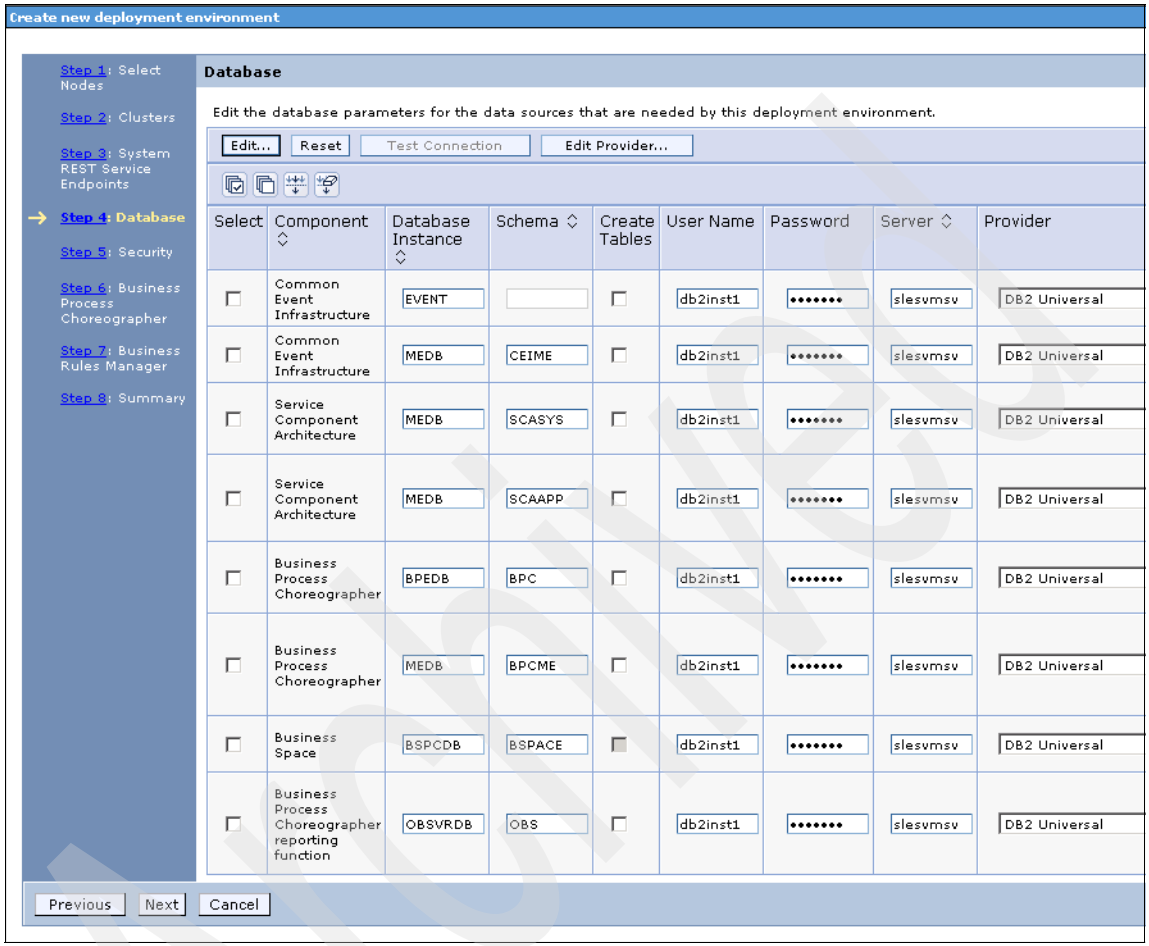


Figure 7-6 The Database configuration window

The Database window is the most complex, and care must be taken to edit this table correctly. Refer to Table 7-1 for a description of the fields and how they relate to the databases that you created earlier.

Table 7-1 Database instances

Database instance	Description	Comments
EVENT	Event server data source	This database does not exist yet. We create it after deployment of the topology. Note that this does not support a schema name.
MEDB	CEI Messaging Engine data source	Created earlier with schema CEIME.
MEDB	SCA System Bus Messaging Engine data source	Created earlier with schema SCASYS.
MEDB	SCA Application Bus Messaging Engine data source	Created earlier with schema SCAAPP.
BPEDB	Business Process Choreographer data source	Created earlier with schema BPC.
MEDB	Business Process Choreographer Messaging Engine data source	Created earlier with schema BPCME.
BSPCDB	Business Space data source	Created earlier with schema BSPACE.
OBSVRDB	Business Process Choreographer Event Collector data source	Created earlier with schema OBS.

8. Fill in the form with the details shown in Table 7-1 on page 175. Figure 7-6 on page 174 does not show the full window details for space reasons, but the description of each value is given on the far right of the window. Make sure that the Create Tables column is cleared for each value, and click **Next**. The security window appears (Figure 7-7).

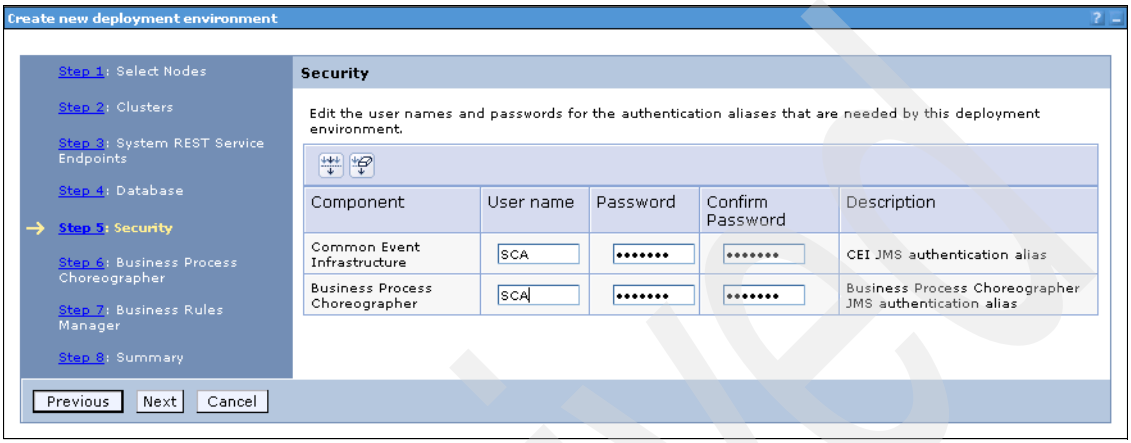


Figure 7-7 Security Configuration window

9. Leave the user names in both cases to be Service Component Architecture (SCA) and enter a password. This user will need to be in LDAP later. Click **Next**. The Business Process Choreographer window appears (Figure 7-8).

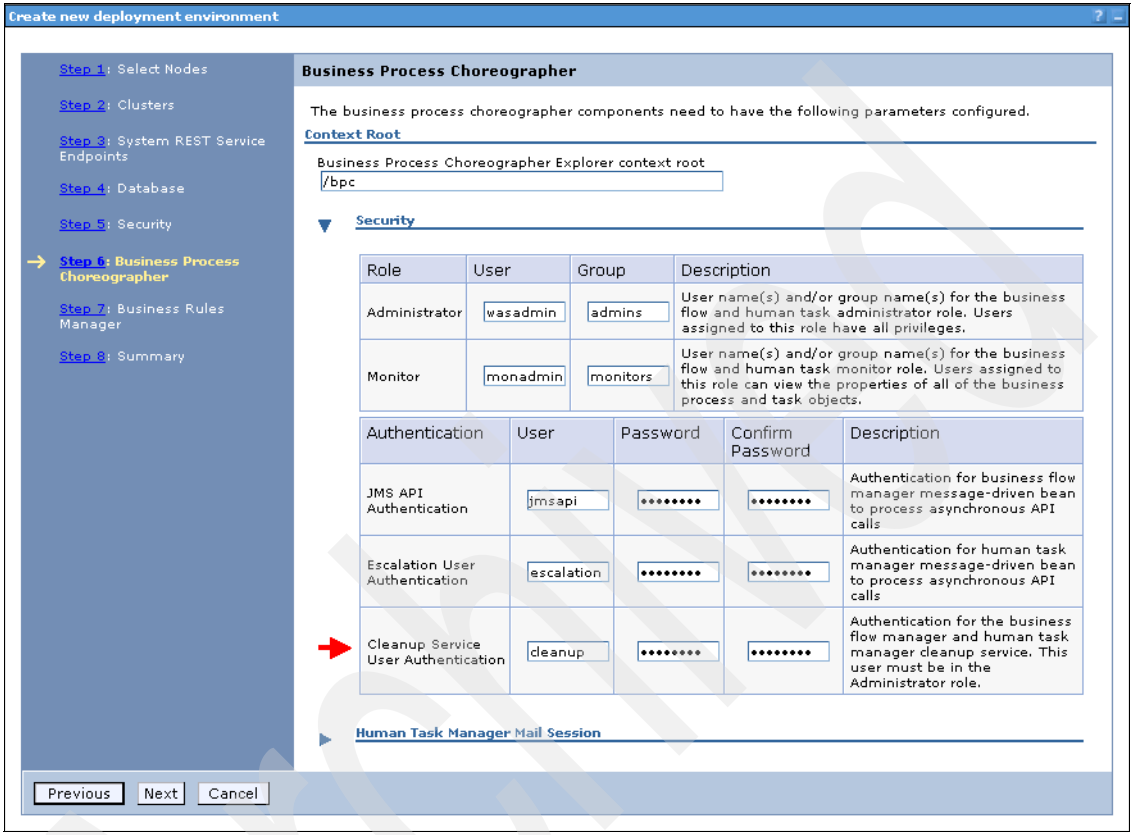


Figure 7-8 Business Process Choreographer window

10. Perform the following steps to specify groups and users for authorization:
    a. For the administrator role, use the following values for the User and Group text boxes:
        - User: `wasadmin`
        - Group: `Admins`
    b. For the monitor role, use the following values for the User and Group text boxes:
        - User: `monadmin`
        - Group: `Monitors`
    c. For the JMS API authentication, use the following values for the user and password:
        - User: `jmsapi`
        - Password: `passw0rd`
    d. For the escalation user authentication, use the following values for the user and password:
        - User: `escalation`
        - Password: `passw0rd`
    e. For the cleanup user authentication, use the following values for the user and password:
        - User: `cleanup`
        - Password: `passw0rd`

        When we enable LDAP, these users and groups must be created in the LDAP database.

    > **Note:** New in WebSphere Process Server 6.2, you can use the Integrated Solutions Console to configure the cleanup service to schedule jobs that periodically delete eligible instances. For more information about the cleanup service go to:
    >
    > http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.websphere.bpc.620.doc/doc/bpc/tadmin_cleanup.html

11. Clear the Enable e-mail service check box in the Human Task Manager Mail Session section because we will not be using human tasks with e-mail escalations. If you require this, you must also provide the other details. Click **Next**. The Business Rules Manager window appears.
12. Click **Next**. The Summary window is displayed.
13. On the Summary panel, check your settings and click **Finish**.

> **Note:** Do not click Finish and Generate Environment because you want to review your settings before generating the environment. Do not try to start the Deployment Environment because the EVENT database does not yet exist and this will cause the deployment to fail.

14. Click the **RMSgold** link to display the Configuration window (Figure 7-9). This window shows the status of the three clusters that you just defined, all of which are currently not configured.

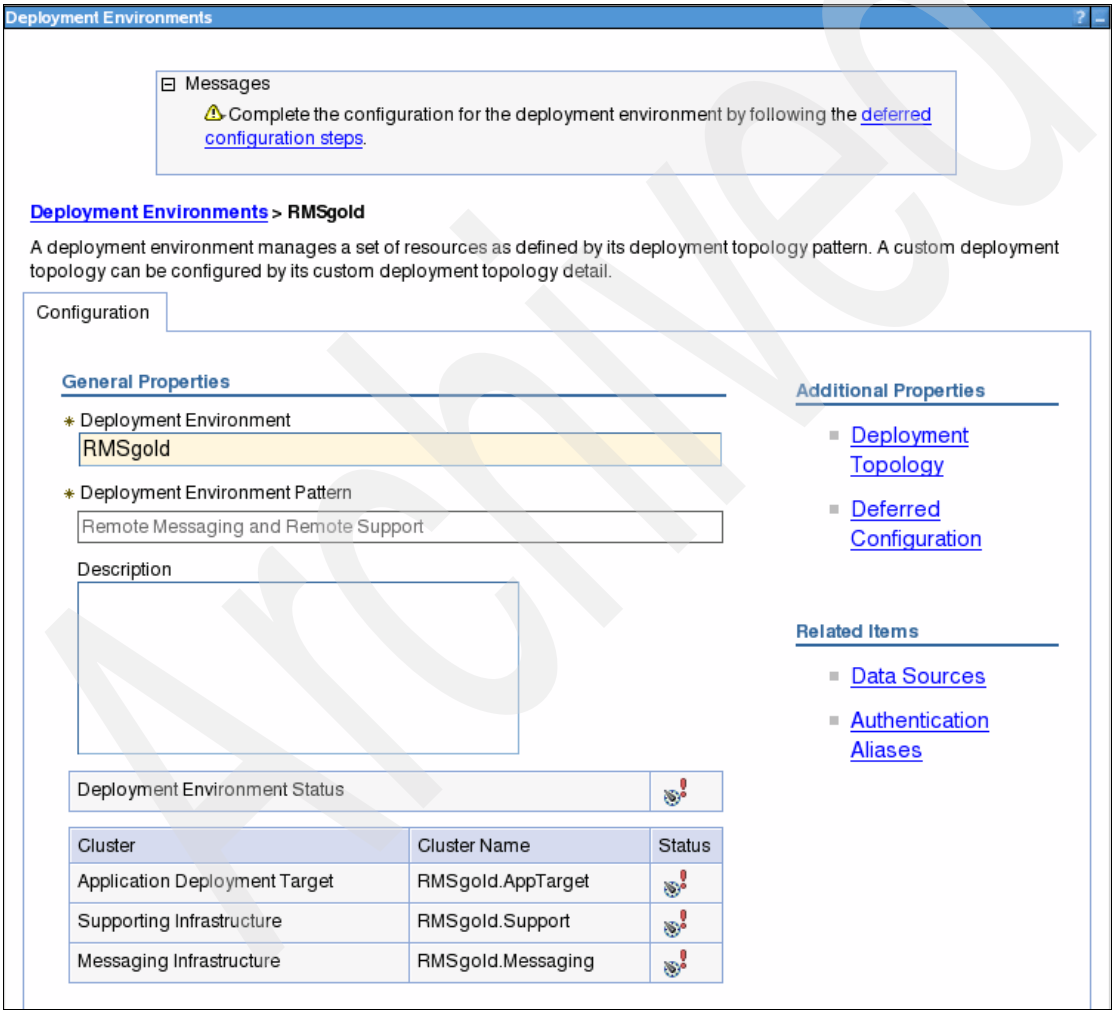


Figure 7-9 RMSgold Deployment Environment configuration window

15. Under Additional Properties, on the right side of the window, you can click **Deployment Topology** to see that the nodes are running but the clusters are not configured. Click **Cancel** to return to the Deployment Environments Configuration window.

16. Under Related Items, on the right side of the window, you can click **Data Sources** to show the database, schema, and JNDI names that have been defined. Click **Cancel** to return to the Deployment Environments Configuration window.

17. Click **Generate Environment**. A Configuration Status window is displayed, as shown in Figure 7-10.

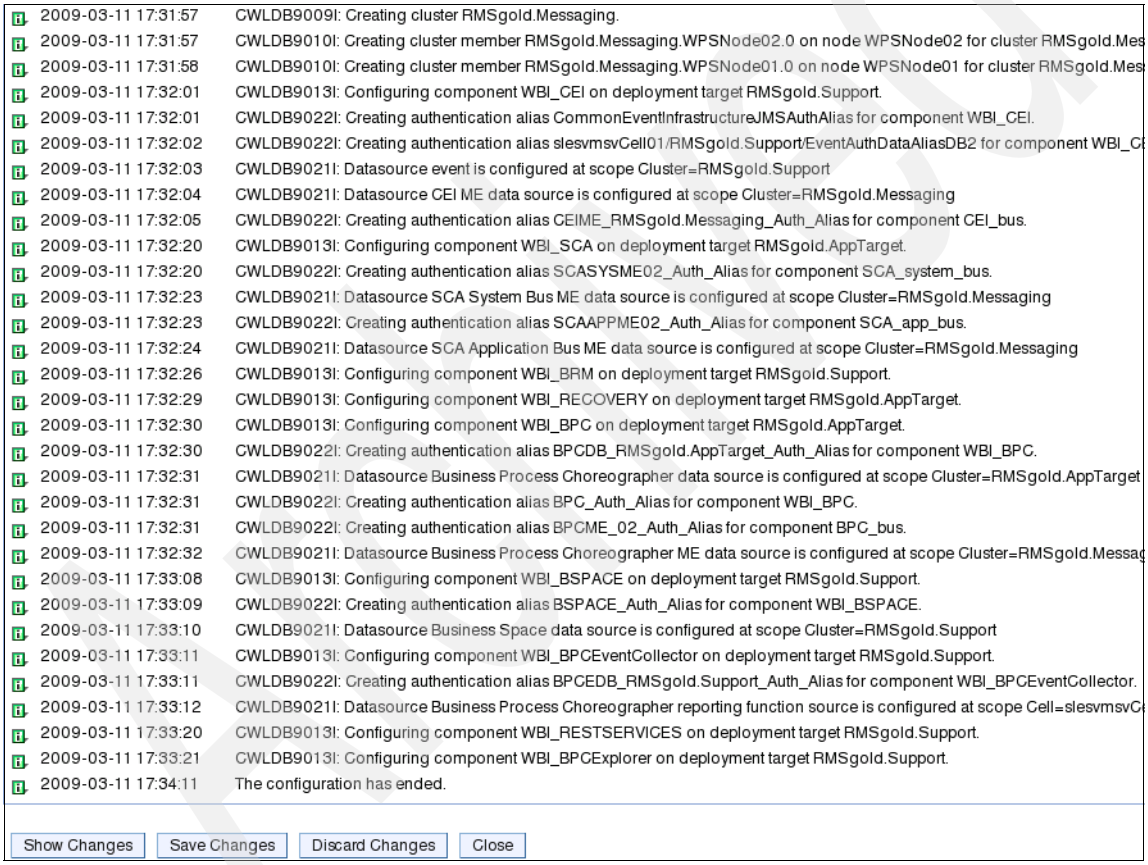


Figure 7-10 Completion of Generate Environment window

18. When complete, click **Save Changes**. The environment will now have a status of stopped (Figure 7-11).

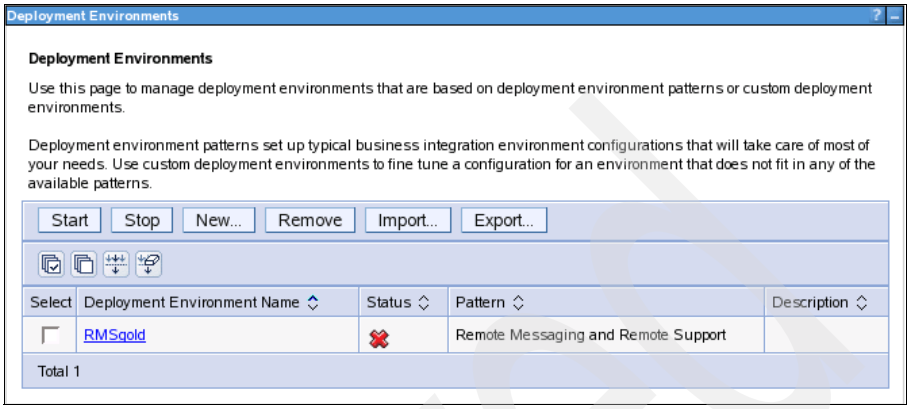


Figure 7-11 Deployment Environment status window

19. Log out of the Integrated Solutions Console.

7.2.2 Creating the event database tables

The final task before starting the environment is to create the event database tables. The scripts to do this are now available under the deployment manager profile's directory. For example:

```
/opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/databases/event/RMSgold.
Support/dbscripts/db2
```

The first task is to copy these scripts over to the DB2 system under the instance owner. The steps below assume that you have copied the files across into the home folder of the instance owner.

1. Log in to the DB2 System as the instance owner.
2. Change the directory to the scripts just copied, then run the following command:

   ```
   ./cr_event_db2.sh 2>&1 | tee output.log
   ```

3. Enter `1` for a server connection, because we are on the DB2 system itself.
4. Enter the instance owner name.
5. Enter the instance owner password.
6. The database and tables will be created. Check the file `output.log` for any messages.

Note: These DB2 commands may report various informational messages. This includes the following messages:

```
SQL0598W  Existing index "BPCME.SIB000PKIX" is used as the index
for the primary key or a unique key.  SQLSTATE=01550

SQL20189W  The buffer pool operation (CREATE/ALTER) will not take
effect until the next database startup due to insufficient
memory.  SQLSTATE=01657
```

These are not errors. You can ignore these messages.

7. Log off the DB2 system.

7.2.3 Checking database connectivity

Before you start the environment you must check database connectivity:

1. Log in to the Integrated Solutions Console and navigate to **Resources** → **JDBC** → **JDBC Providers**. You will see that there are four providers now at different scopes: one provider at the cell scope level and three additional providers (one for each of the separate clusters at the cluster scope), as shown in Figure 7-12.

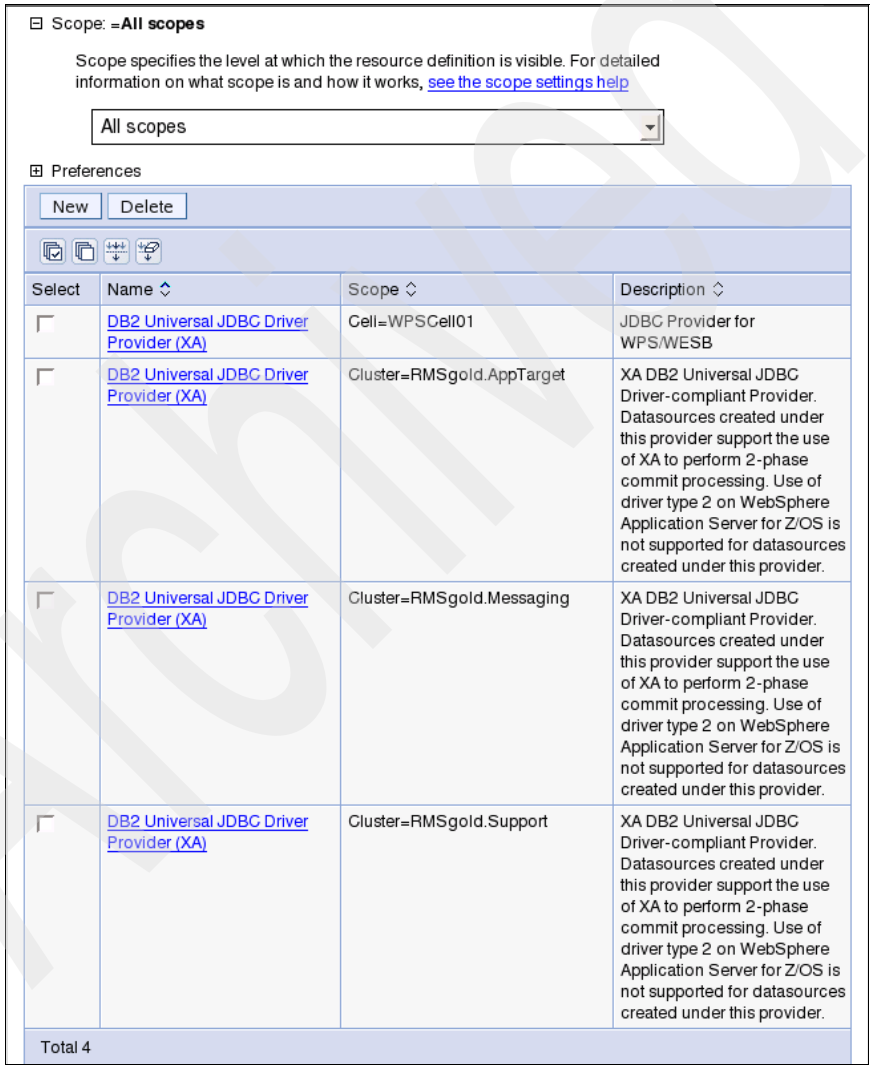


Figure 7-12 JDBC Providers at All scopes window

2. Click the first provider where the scope is Cell. In our example it is WPSCell01. Under Additional Properties, click **Data sources**. You will see three data sources. Select the check box next to each data source and click **Test connection** to make sure that they have connectivity.
3. We now must define new variables. Navigate to **Environment** → **WebSphere variables** and select **Cluster=RMSgold.Support** as the scope. Click **New**. Create a new variable called DB2_UNIVERSAL_JDBC_DRIVER_PATH with the value /opt/ibm/WebSphere/ProcServer/universalDriver_wbi/lib. Save your changes.
4. Repeat step 3 for the two other cluster scopes (RMSgold.AppTarget and RMSgold.Messaging).
5. Navigate to **System Administration** → **Save changes to master repository**. Click the **Synchronize changes with the nodes** check box, and click **Save**.
6. Navigate to **System Administration** → **Node agents**. Select both node agents and click **Restart**. This may expire your login to the administrative console, so you may be required to log back in.
7. Navigate to **Resources** → **JDBC** → **JDBC Providers**. Click the provider link at the scope Cluster=RMSgold.Support (Figure 7-13).

	DB2 Universal JDBC Driver Provider (XA)	Cluster=RMSgold.Support	XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on WebSphere Application Server for Z/OS is not supported for datasources created under this provider.

Figure 7-13 JDBC provider link at cluster scope window

> **Note:** When checking database connectivity, if you do not see the desired provider, make sure that the scope at the top of the page is set to All Scopes.

8. Under Additional Properties, click **Data sources**, select the check box of the data sources, and click **Test connection**.
9. Navigate to **Resources** → **JDBC** → **JDBC Providers**. Click the provider link at the scope Cluster=RMSgold.AppTarget.
10. Under Additional Properties, click **Data sources**, select the check box (there is only one), and click **Test connection**.

11. Navigate to **Resources** → **JDBC** → **JDBC Providers**. Click the provider link at the scope Cluster=RMSgold.Messaging.
12. Under Additional Properties, click **Data sources**. You will see one data source for each of the schemas that we created earlier. In each case, we must make sure that the authentication alias is correctly set before we test the connection.
13. Click the **Business Process Choreographer ME data source** link and scroll down the page until you reach a heading of Component-managed authentication alias. Select **BPCME_00_Auth_Alias** from the drop-down list. Under Authentication alias for XA recovery, select the **Use component-managed authentication alias** radio button and click **OK**.
14. Save and synchronize the changes. This takes a moment. Once saved, you should be returned to the Data sources page. You can now check the connectivity by selecting the **Business Process Choreographer ME data source** check box and clicking **Test connection**.
15. Perform the same actions for the other three data sources using the values shown in Table 7-2.

Table 7-2 Authentication aliases for messaging engines

Data source name	Authentication alias
CEI ME data source	CEIME_RMSgold.Messaging_Auth_Alias
SCA Application Bus ME data source	SCAAPPME_00_Auth_Alias
SCA System Bus ME data source	SCASYSME_00_Auth_Alias

7.2.4 Completing the topology configuration

In this section you will start the deployment environment. To complete the topology configuration:

1. Log in to the Integrated Solutions Console.
2. Navigate to **Servers** → **Deployment Environments**. Click the **RMSgold** link (which is currently stopped).
3. Under Additional Properties, click **Deferred Configuration**. A list of tasks required to complete the configuration is displayed. Because we created all the databases before starting any deployment and have just finished the configuration of the event database, these tasks have been completed. Click **Configuration Done**, save the changes, and click **Close**.

4. Navigate to **Servers** → **Deployment Environments**. Select the **RMSgold** check box and click **Start**. The RMSgold deployment environment status will immediately change to Partial Start (Figure 7-14).

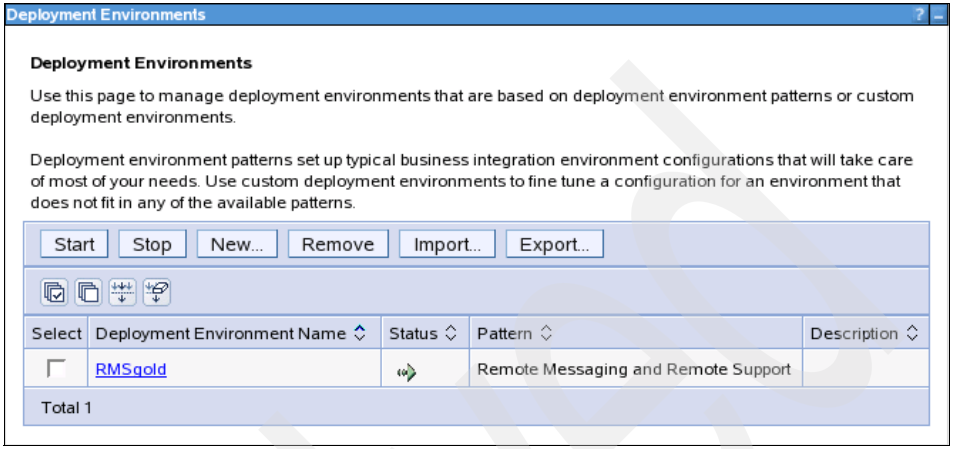


Figure 7-14 Deployment Environment status window

5. Navigate to **Servers** → **Clusters** and you will see that the status of each server cluster is Partial Start (Figure 7-15).

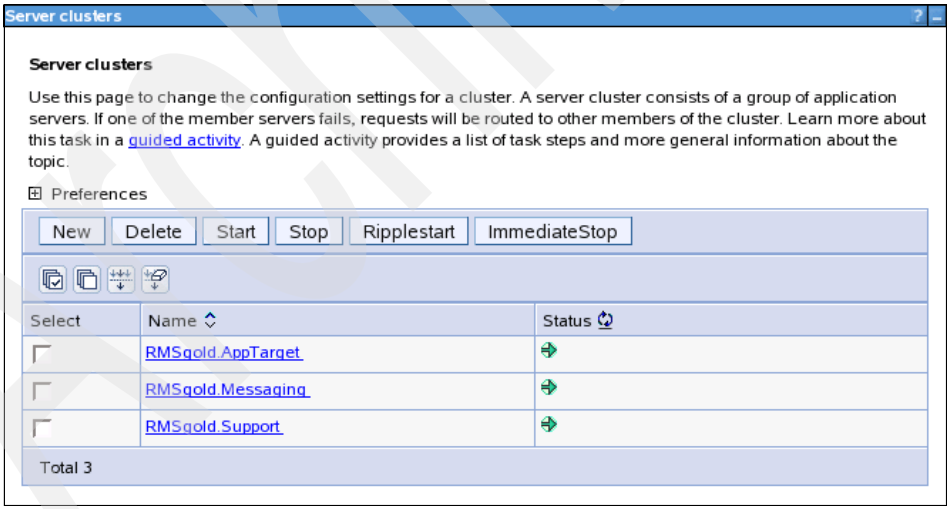


Figure 7-15 Server Clusters window

6. Wait until all the clusters are in Started State. The arrows will be solid green.

7.3 Post-creation configuration and verification

In this section we add functonality to the deployment topology and perform some simple checks to verify that the topology was created successfully.

7.3.1 Configuring CEI logging

To configure CEI logging:

1. In the Integrated Solutions Console, navigate to **Servers** → **Clusters** → **RMSgold.AppTarget**.
2. On the right, under Business Integration, expand **Business Process Choreographer** and 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 or Human Task Manager check boxes, or both, depending on your requirements (Figure 7-16).

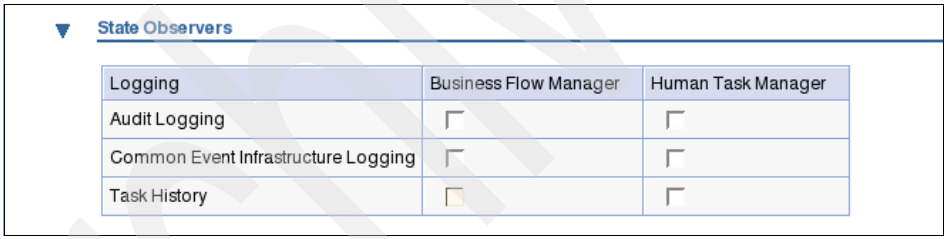


Figure 7-16 Configure CEI logging window

4. Save and synchronize your changes.

7.3.2 Configuring shared transaction logging

This section introduces considerations for shared transaction logging. It contains the following sections:

- “High-availability considerations for the transaction manager” on page 188
- “Create the shared directories for the transaction logs” on page 188
- “Change the transaction manager log settings” on page 191
- “Policies for transaction manager peer recovery” on page 192

High-availability considerations for the transaction manager

The WebSphere Application Server transaction manager (used by WebSphere Process Server) writes to its transaction recovery logs when it handles global transactions (XA transactions) that involve two or more resources. Transaction recovery logs are stored on disk and are used for recovering in-flight transactions from system crashes or process failures. By default, each cluster member maintains its own transaction log.

To keep the transaction logs highly available and to enable transaction peer recovery, it is necessary to place the recovery logs on a highly available file system, such as IBM SAN FS or NAS, for all the application servers within the same cluster to access. All application servers must be able to read from and write to the logs. In addition to configuring a highly available file system, you must decide whether to use automated or manual peer recovery for the transaction manager. In either case transaction manager policies must also exist.

For more details on high availability considerations for the transaction logs, refer to the IBM Redbooks publication *WebSphere Application Server Network Deployment V6: High Availability Solutions*, SG24-6688.

Create the shared directories for the transaction logs

Once you have decided upon a highly available file system, you must configure the transaction log directory setting for each server in the cluster. You can configure the location of the transaction log directory using either the administrative console or commands. The configuration is stored in the `serverindex.xml` node-level configuration file.

Each server must be able to access the log directories of other servers in the same cluster. For this reason, do not leave this setting unset. If you do not set a directory, the application server assumes a default location within the appropriate profile directory, which might not be accessible to other servers in the cluster.

Each server in the cluster must also have a unique transaction log directory to avoid attempts by multiple servers to access the same log file. For example, you could use the name of each server as part of the log directory name for that server.

To set the transaction log directory for the cluster members:

1. In the administrative console, expand **Servers** and click the **Clusters** link.
2. Click the check box for the cluster that you wish to modify and click **Stop**.
3. Once the cluster is stopped, click the link for the cluster that you wish to modify.
4. In the Additional Properties section, click the **Cluster members** link.

5. Click the link for the first cluster member.
6. In the Container Settings section, expand **Container Services** and click the **Transaction Service** link. See Figure 7-17.

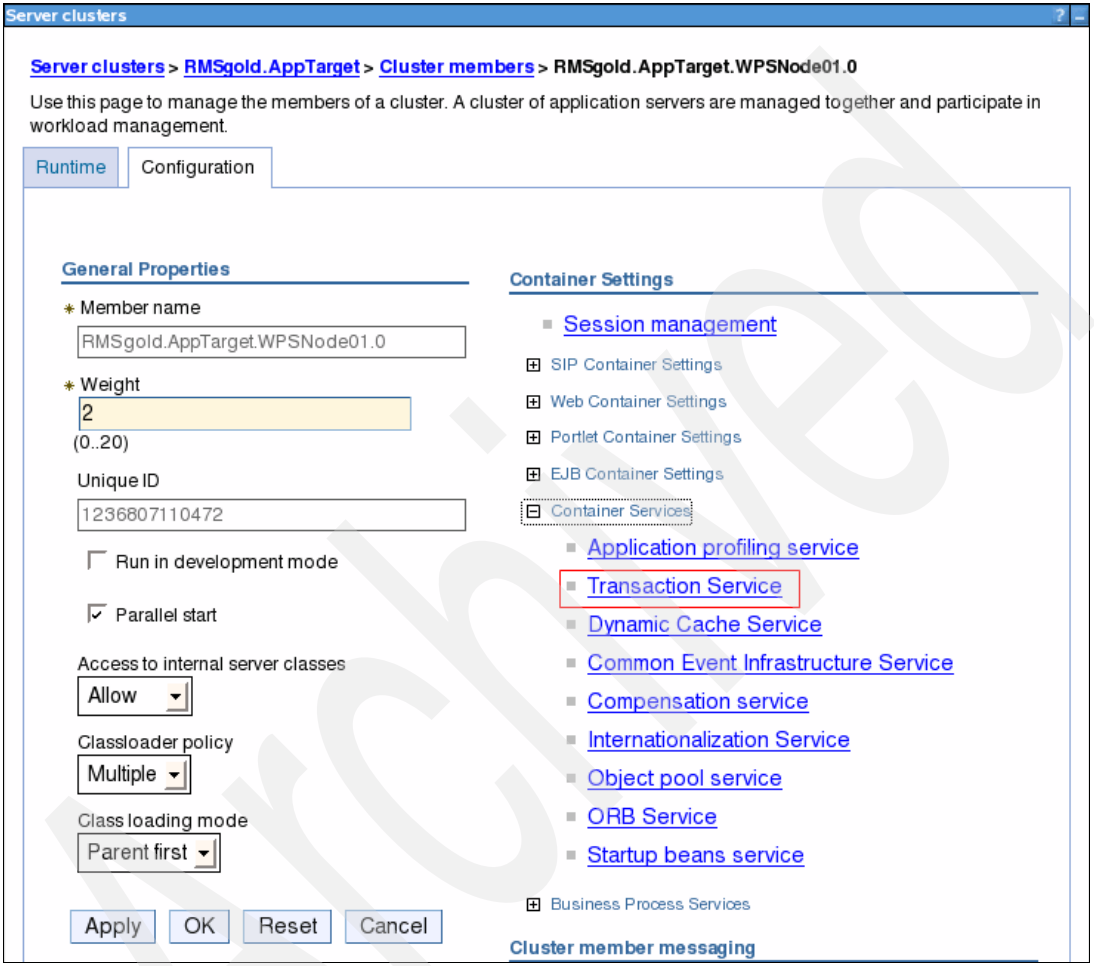


Figure 7-17 Transaction Service link from Container Settings

7. In the General Properties section, enter an appropriate value in the Transaction log directory text box. See Figure 7-18.

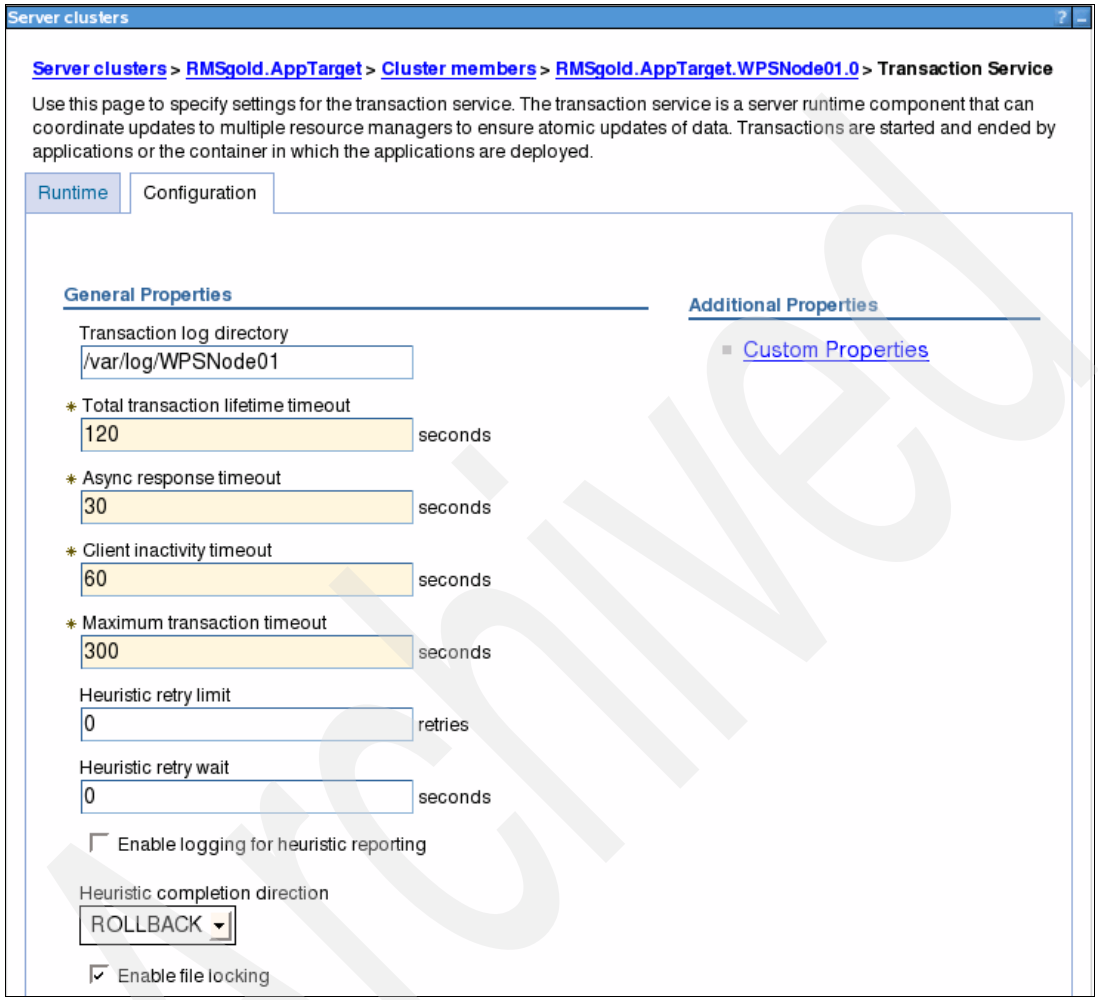


Figure 7-18 Transaction log directory for WPSNode01 window

> **Tip:** If you are using NFS, we recommend using the hard option in the NFS mount command (**`mount -o hard`**) to avoid data corruption.

8. Click **OK**.
9. Save the changes to the master configuration.

10. Wait for automatic synchronization to complete and click **OK**, or manually synchronize the nodes.
11. Copy the existing transaction logs to the shared file system. Make sure that the location and file permissions are correct.

Change the transaction manager log settings

Once you have configured the transaction log location for the cluster members, you must enable transaction log failover for the cluster. To enable transaction log recovery:

1. In the Integrated Solutions Console, expand **Servers** and click the **Clusters** link. Make sure that the clusters are in stopped state.
2. Click the link for the cluster that you wish to modify. (The following images show the transaction log settings for the AppTarget cluster and its members.)
3. In the Configuration tab, in the General Properties section, click the **Enable failover of transaction log recovery** check box. See Figure 7-19.

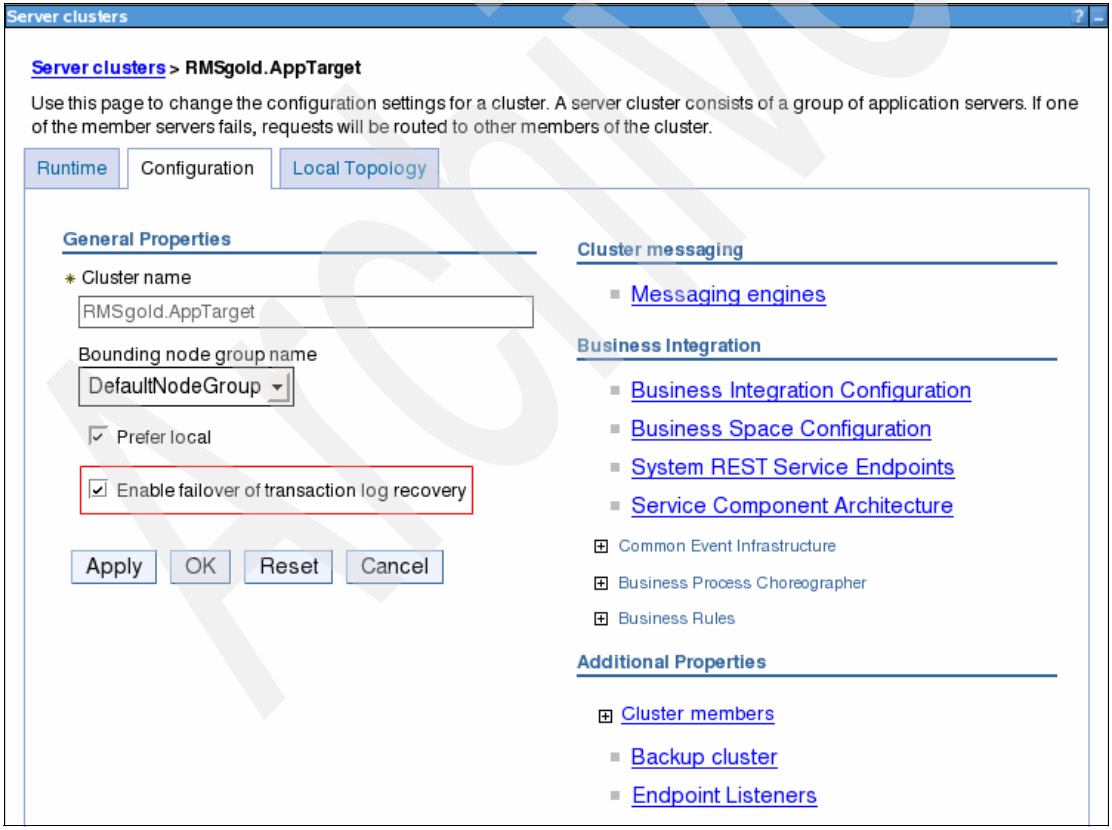


Figure 7-19 Enable failover of transaction log recovery window

Click **OK**.

4. Save the changes to the master configuration.
5. Wait for automatic synchronization to complete and click **OK**, or manually synchronize the nodes.
6. Start the cluster.

Policies for transaction manager peer recovery

In order for transaction log failover to work correctly, you must have one or more policies in place. In each WebSphere Process Server deployment, a default transaction manager policy is created to control failover of the transaction manager service. This policy is a *one of n policy* similar to the policies created for the messaging engines in Chapter 11, “Advanced production topologies” on page 269.

A *one of n* policy means that only one server in a cluster can run the transaction manager service at any given time. If the running transaction manager service fails, the default transaction manager policy, called *Clustered TM Policy*, specifies that the service can fail over to another cluster member. The default policy also enforces failback. If the failed transaction manager becomes available, the transaction manager service will fail back to it.

If you are using automated failover, the default transaction manager policy is likely sufficient for your needs. To examine the default transaction manager policy:

1. In the administrative console, expand **Servers** → **Core groups**.
2. Click the **Core group settings** link.
3. Click the **DefaultCoreGroup** link.
4. In the Additional Properties section, click the **Policies** link.

5. Click the link for **Clustered TM Policy**. See Figure 7-20.

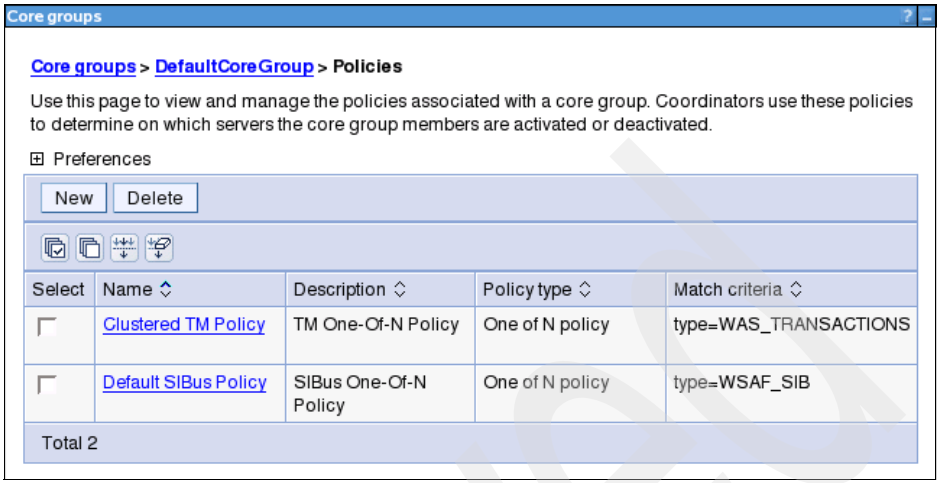


Figure 7-20 DefaultCoreGroup policies window

6. Examine the properties of the policy. See Figure 7-21.

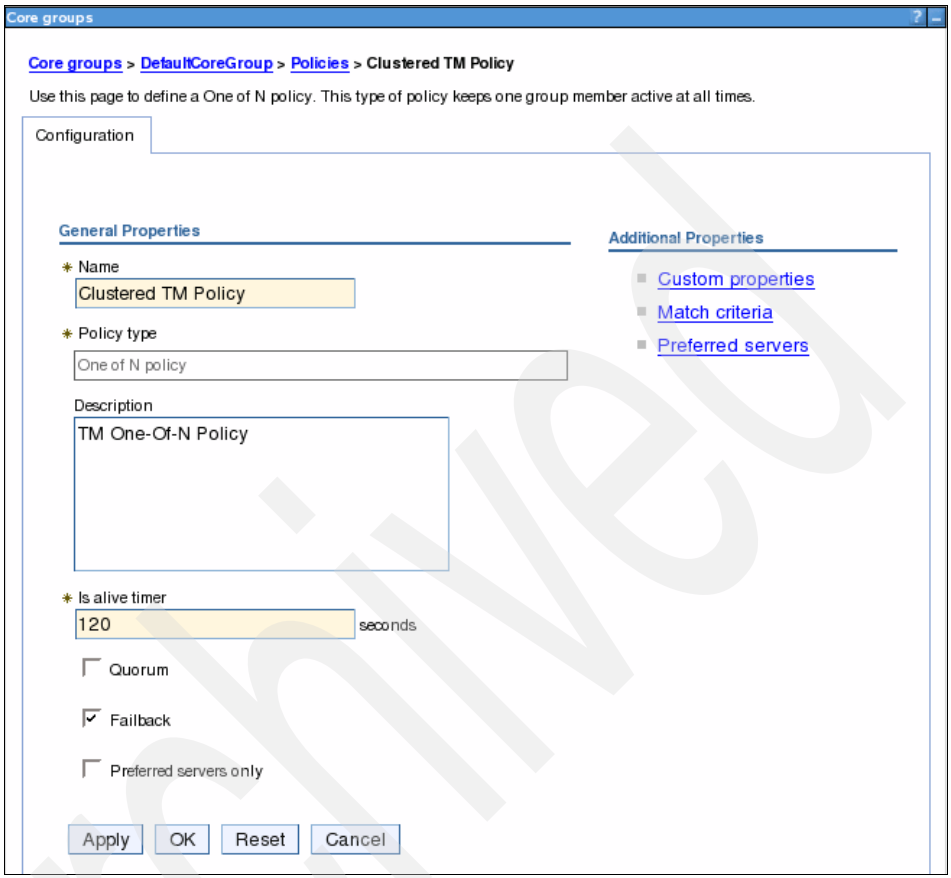


Figure 7-21 Default transaction manager policy

7. Click **Cancel**.

7.3.3 Installing the sample application

This publication provides a sample vehicle loan process created for the fictitious company ITSOBank. You can use this vehicle loan process to test the topology that you have built in this chapter. For more information about the vehicle loan process, refer to Chapter 2, “Sample business application scenario used in topologies” on page 23. To obtain the additional material supplied with this book, refer to Appendix A, “Additional material” on page 597.

When you have obtained the additional materials, navigate to the `Scenarios\WPS\EAR\v6.2_Plain` directory. From here, you must copy the

`ITSOApp.ear` and `ITSO_implApp.ear` files to the deployment manager. Installation follows the normal process and is described here:

1. Navigate to **Applications** → **Install New Application**.
2. Choose **Remote file system**, click **Browse**, and navigate to the location of the uploaded EAR files.
3. Click the **ITSO_implApp.ear** radio button, then click **OK**.
4. Select the radio button **Prompt me only when additional information is required**, as shown in Figure 7-22. Click **Next**.

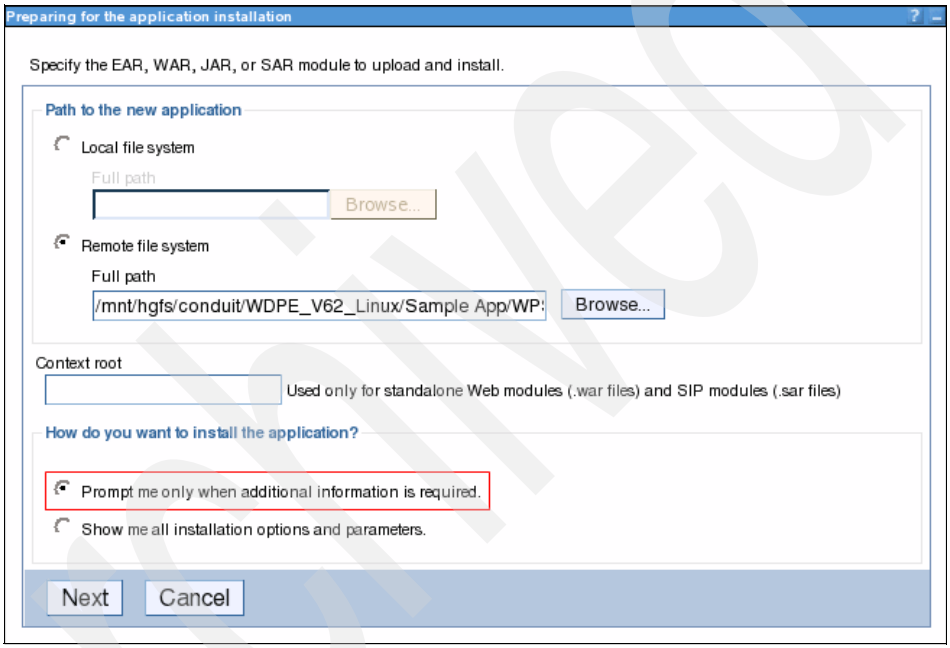


Figure 7-22 Preparing for the application installation window

5. On Step 1: Select installation options, click **Next**.
6. On Step 2: Map modules to servers, select **WebSphere:cell=slesvmsvCell01,cluster=RMSgold.AppTarget**, select the **ITSO_implWeb** check box, click **Apply**, then click **Next**.
7. On Step 3: Summary click **Finish**.
8. When you see the message `Application ITSO_implApp installed successfully`, click the **Save** link, then click **OK**.
9. Repeat the process for the `ITSOApp.ear` file. This time, select both the **ITSOWeb** and **ITSOEJB** check boxes.

You can check that the Web server plug-in file is correctly updated:

1. Navigate to **Servers** → **Web servers** and click **webserver1**.
2. Under Additional Properties, click **Plug-in properties**.
3. Under Plug-in properties, click the **View** button.
4. Scroll down the page and you should see the lines shown in Example 7-1.

Example 7-1 Plug-in details showing ITSO application URL

```
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
Name="/ITSO_implWeb/*"/>
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
Name="/ITSOWeb/*"/>
```

5. Navigate to **Servers** → **Web servers**. Select the **webserver1** check box and click **Propagate Plug-in**.
6. Navigate to **Applications** → **Enterprise Applications**. Select the **ITSOApp** and **ITSO_implApp** check boxes, then click **Start**.
7. Log in to Business Process Choreographer console. In our environment we used the following URL:

   `http://slesvmsv:9080/bpc`

8. Click **My Process Templates**, select **New Loan Process**, and click **Start Instance**. Provide some test input data and click **Submit**. This should launch the business process. If the environment is working correctly, it returns a response, as shown in Figure 7-23.

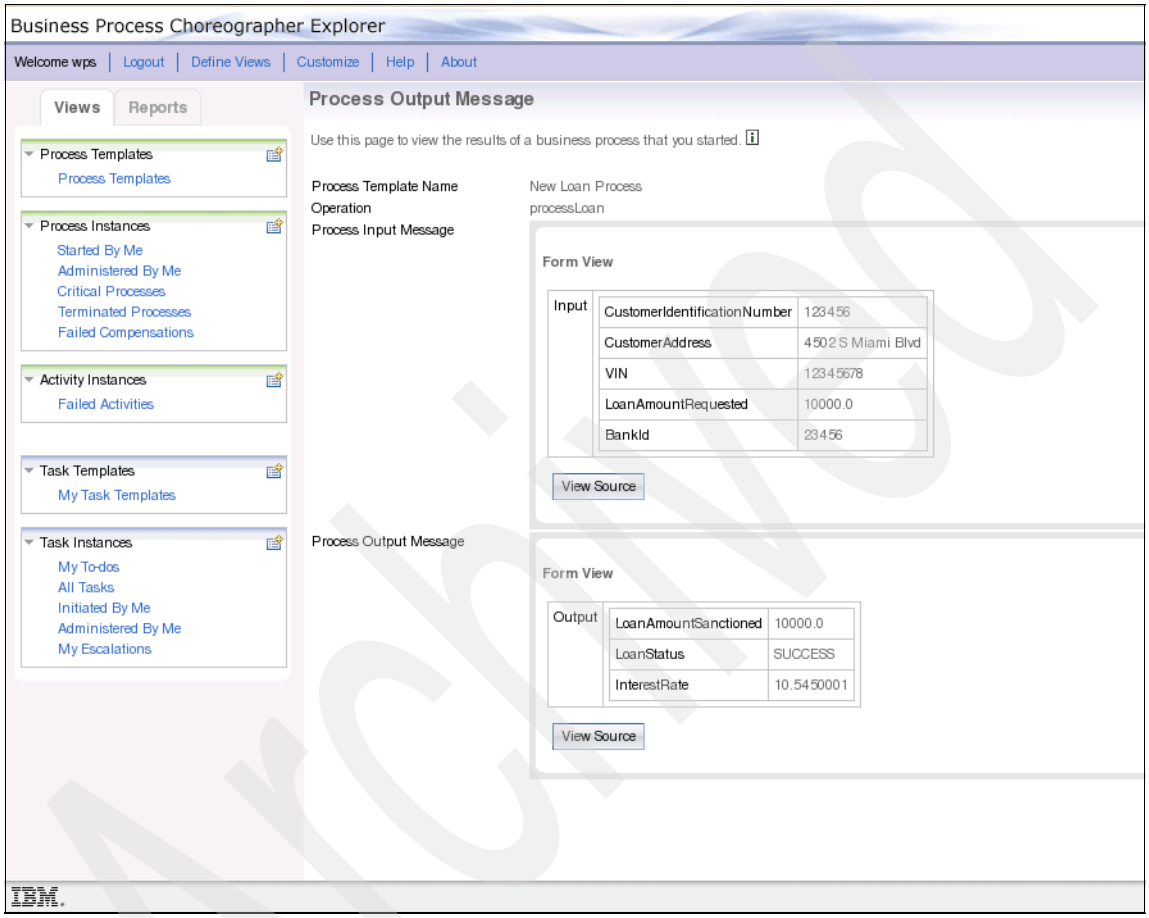


Figure 7-23 Business Process Choreographer Process Output Message

To uninstall this or any other enterprise application from the server, issue the following commands:

```
cd /opt/IBM/WebSphere/ProcServer/ProcessChoreographer/admin

../../bin/wsadmin.sh -lang jacl -f bpcTemplates.jacl -uninstall
"<AppName>" -force
```

8

Configuring a custom topology

This chapter provides instructions for creating a custom topology, which includes Remote Messaging, Application, and Support clusters. We do not use Common Event Infrastructure (CEI).

In this topology we create three clusters:

- An application cluster to support WebSphere Process Server applications
- A messaging cluster to support the messaging engine infrastructure
- A support cluster to host support applications such as the BPC Explorer and the Business Rules Manager

These clusters are configured over two nodes, so each cluster has two members.

8.1 Custom topology creation prerequisites

To begin, start from the same position as the Remote Messaging and Remote Support topology. We assume that the base product has been installed but that no profiles have been created.

8.1.1 Creating the required databases in DB2

You need the following three databases in this topology:

- Common database (WPRCSDB)
- Business Process Choreographer (BPC) database
- Messaging engine database (MEDB) with three schemas

You are not using the CEI, so the event database, the observer database, and the schema for CEI in the messaging database are not required.

Use the instructions in 5.2.2, “Common database” on page 105, to create the common database (WPRCSDB).

Use the instructions in 5.2.3, “Business Process Choreographer database” on page 107, to create the BPC database (BPEDB)

Use the instructions in 5.2.5, “Messaging engine database resources” on page 111, to generate the DDL for each of the Messaging Engine schemas, and the ME database:

- BPCME
- SCASYS
- SCAAPP
- MEDB

8.1.2 Create a Deployment Manager profile

There are two options for creating a deployment manager profile:

- To create a deployment manager profile using the profile management tool (the graphical option), see 5.3.1, “Deployment manager profile” on page 113.
- To create a deployment manager profile silently, see 5.4.2, “Deployment manager profile” on page 129.

Remember to perform the post-creation changes. That is, add COMMONDB as the schema name and modify the SCA_Auth_Alias. The script changeDmgr.py can be used or the administrative console.

8.1.3 Create the custom node profile

There are two options for creating custom node profiles:

- To create node profiles using the profile management tool (the graphical option), see 5.3.2, “Node profiles” on page 123.
- To create node profiles silently, see 5.4.3, “Node profiles” on page 131.

8.1.4 Create the custom clusters

Before we run the wizard, we must create our clusters manually. We have three clusters called AppTarget, Messaging, and Support. All three will have two members and will be created using the defaultProcessServer template:

1. Log in to the administrative console and navigate to **Servers** → **Clusters**. Click **New**.
2. Enter `AppTarget` as the name of the cluster and click **Next** (Figure 8-1).

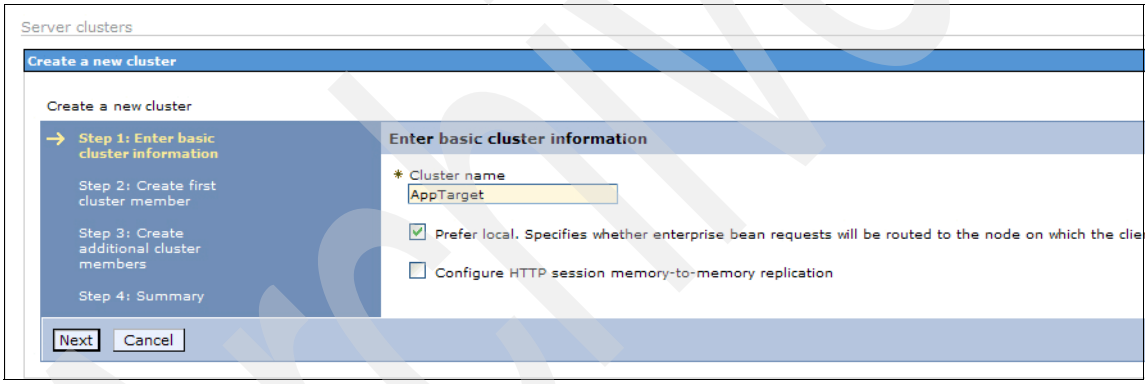


Figure 8-1 Cluster create window

3. In the Create first cluster member window (Figure 8-2) perform the following steps:
   a. Enter `AppTargetServ01` for the member name.
   b. Select **wpsNode01** from the Select node drop-down menu.
   c. Under Select basis for first cluster member, click the **Create the member using an application server template** radio button.
   d. Select **defaultProcessServer** from the drop-down menu of templates and click **Next**.

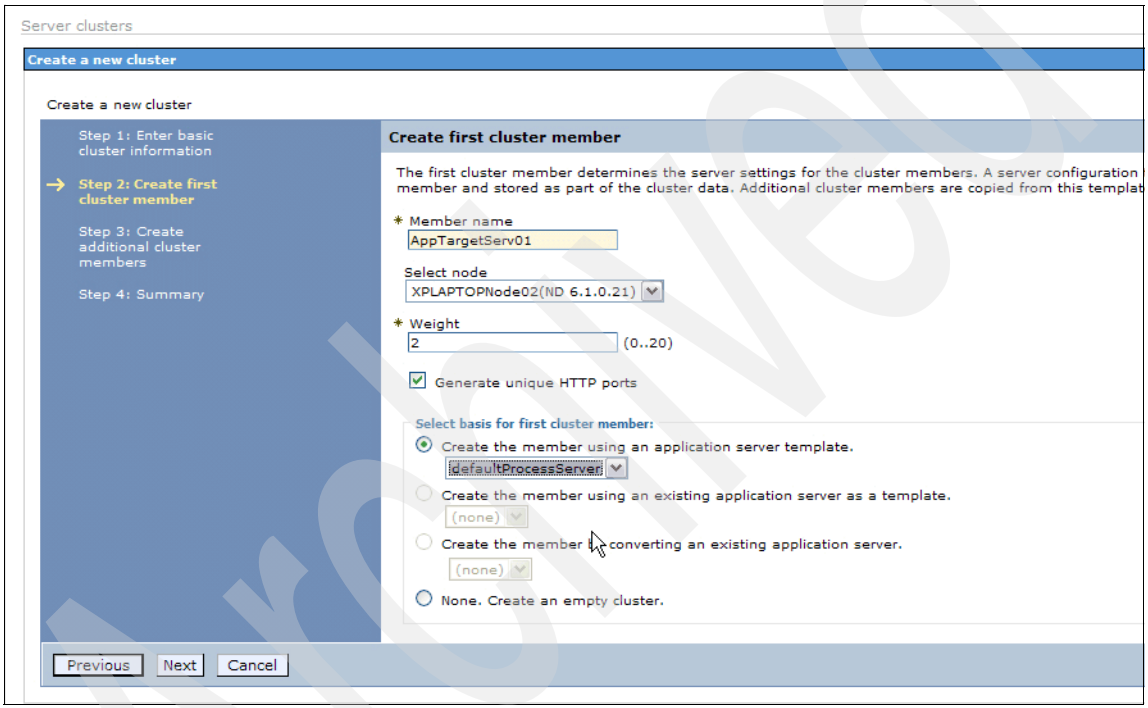


Figure 8-2 Add first cluster member

4. In The Create additional cluster members window (Figure 8-3), enter `AppTargetServ02` for the member name, select **wpsNode02**, and click the **Add Member** button. Click **Next**.

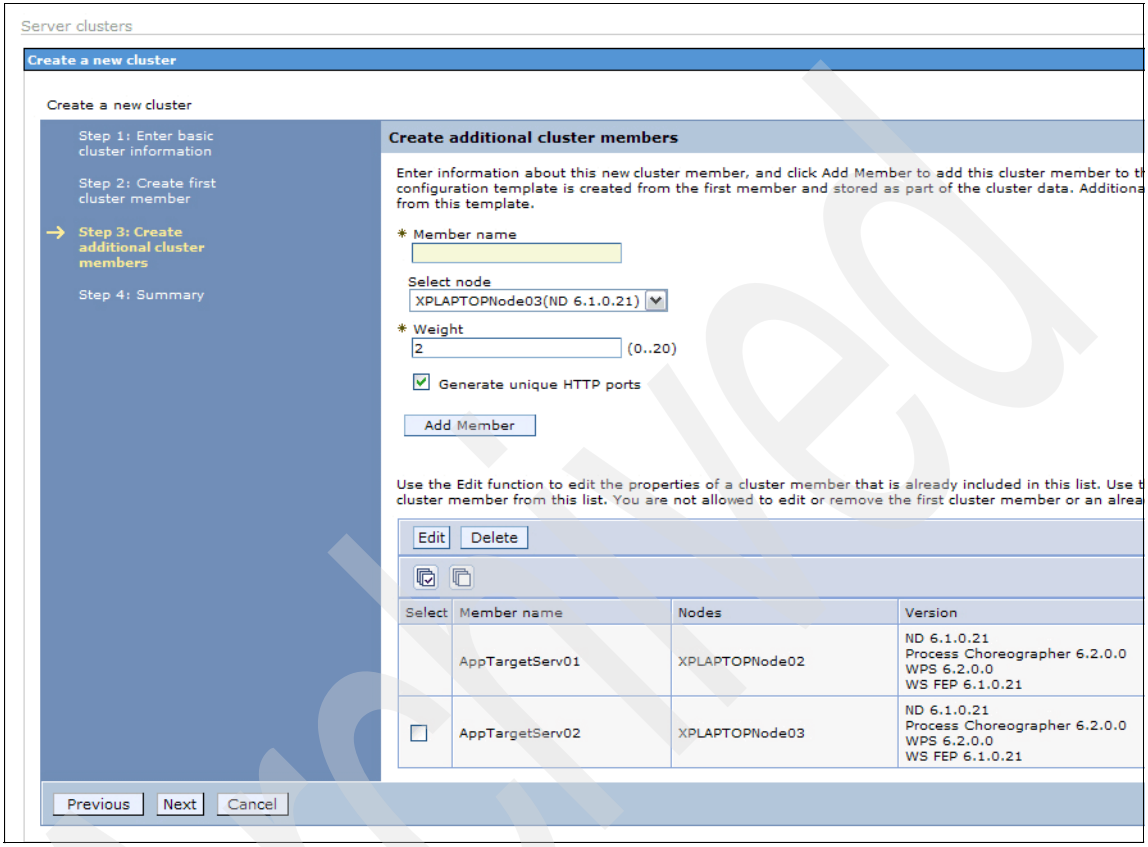


Figure 8-3 Creating additional cluster members

5. In the Summary window click **Finish**. The new cluster will be created.
6. Repeat these steps to create a cluster called Messaging and a cluster called Support, each with two members. Save and synchronize your changes.

8.1.5 Using the custom topology wizard

To run the custom topology wizard in the Integrated Solutions Console:

1. Log in to the administrative console and navigate to **Servers** → **Deployment Environments**. Click **New**.
2. Click the Create a new deployment environment radio button. Enter the name **myCustom** and click **Next** (Figure 8-4).

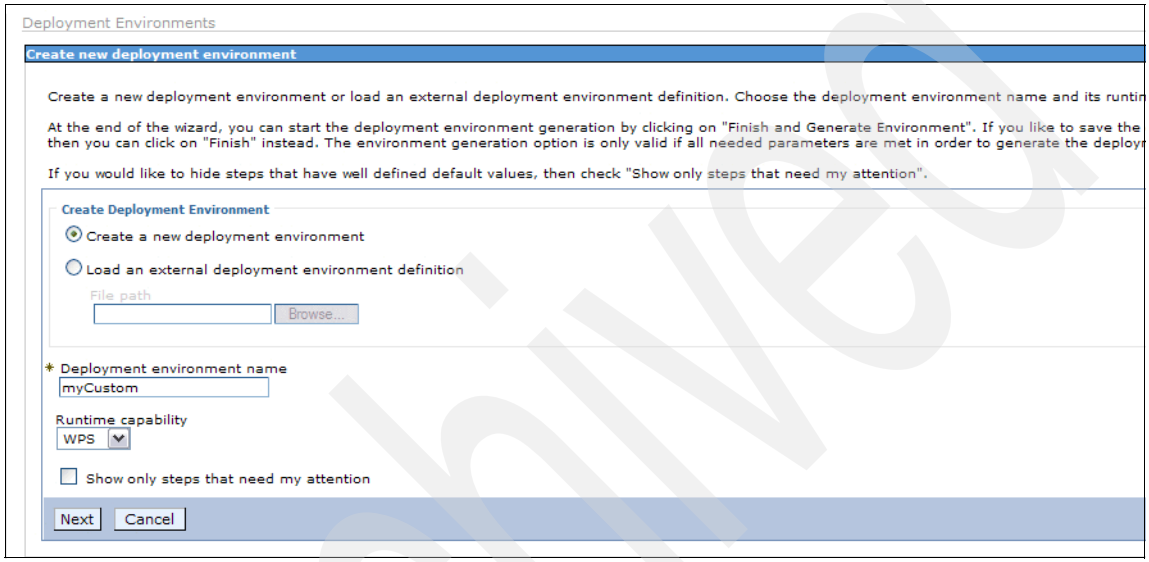


Figure 8-4 Create a new deployment environment

3. In the Deployment Environment Patterns window, click **Custom**, then **Next**.
4. In the Step 1: Deployment environment window, click the **Cluster** radio button, select **AppTarget** from the drop-down menu, and click **Add**.

5. Repeat step 4 to add the **Messaging** and **Support** clusters from the drop-down menu. In Figure 5 we have added the AppTarget and Support clusters and have just added the Messaging cluster. This window has two stages, and we are currently at the first stage where we add clusters to the topology, but have not configured them yet.

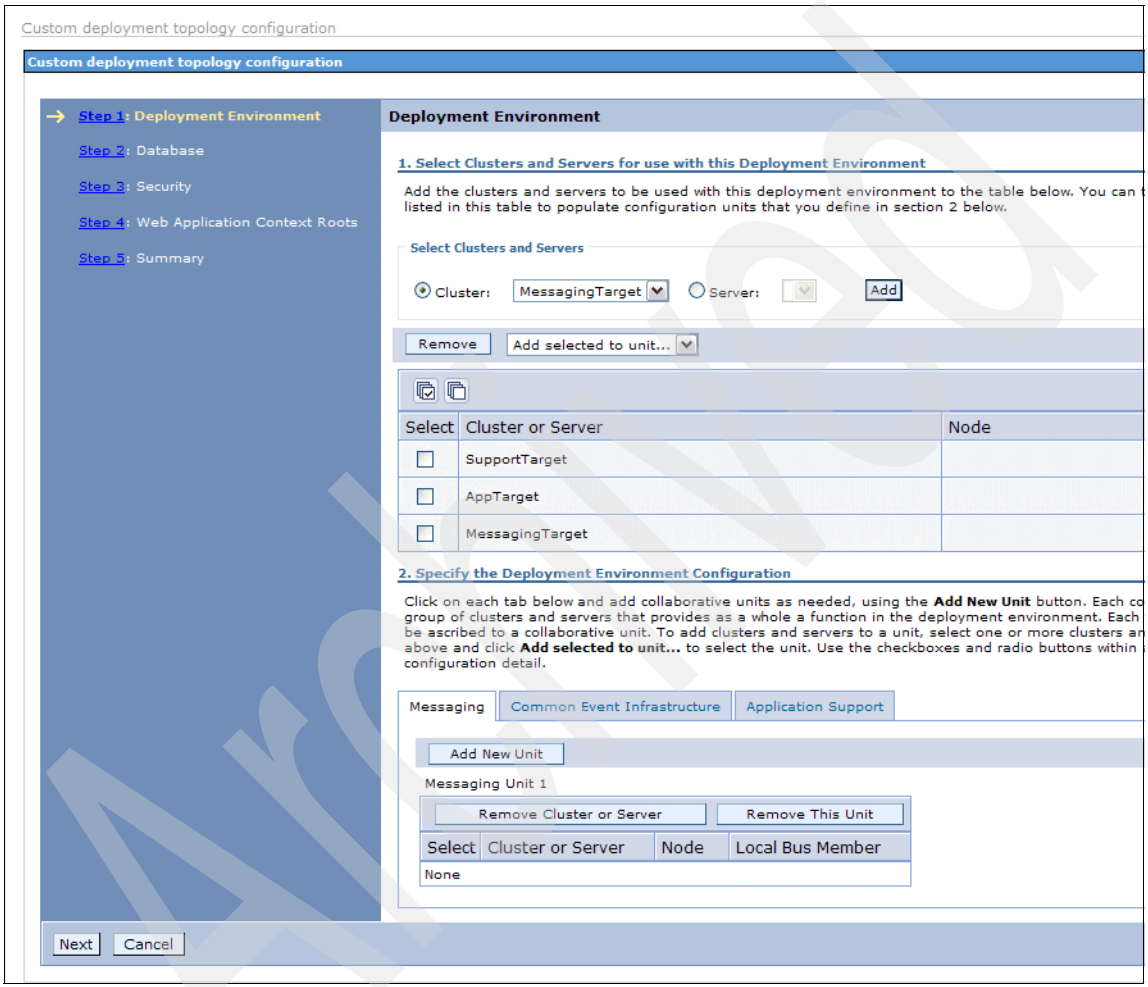


Figure 8-5 Adding cluster to the myCustom Deployment Environment

6. Add all three clusters to each of the configurations. First, add the clusters to the messaging configuration:

   a. Click the **Messaging** tab on the lower half of the window, then select all three clusters in the upper part of the window.

b. Select **Messaging unit 1** from the Add selected to unit drop-down menu. The page will refresh.

7. Click the **Local Bus Member** radio button for the Messaging cluster, as shown in Figure 8-6.

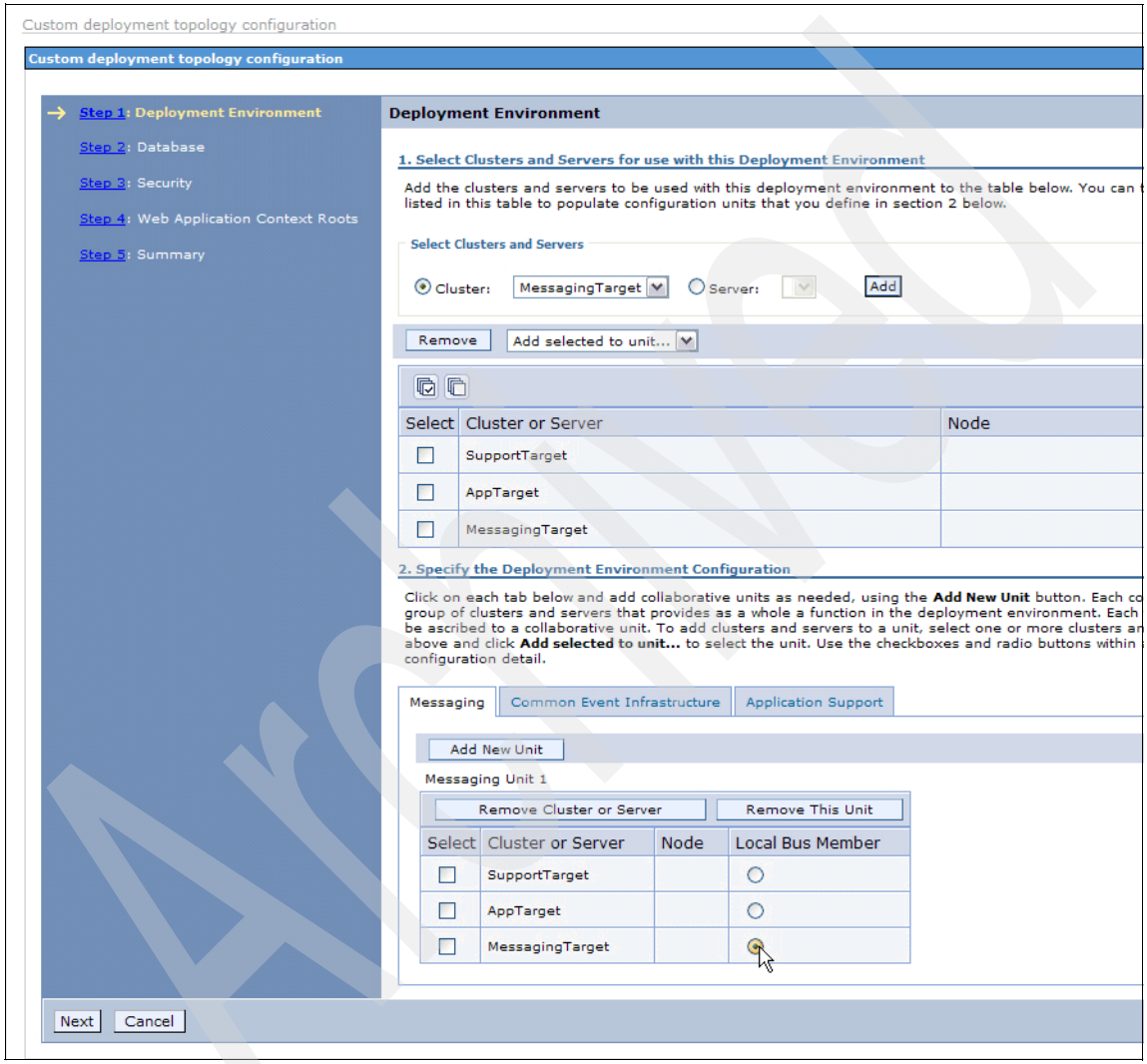


Figure 8-6 Adding the messaging cluster to the messaging unit

8. Click the **Common Event Infrastructure** tab. Select all three clusters and select **Common Event Infrastructure Unit 1** as the unit.

9. Click the **Application Support** tab. Select all three clusters and select **Application Support Unit 1** as the unit. The page will refresh.
10. Enable **Service Component Architecture** on the AppTarget, Messaging, and Support clusters. Note that this enables further buttons within the window.
11. Enable **Business Process Choreographer Container** on the AppTarget cluster. This will enable further options within the window.
12. Enable **Business Process Choreographer Explorer** and **Business Rules Manager** on the Support cluster. There is no option for Business Process Event Collector because we have not used CEI.

The completed window is shown in Figure 8-7. Click **Next**.

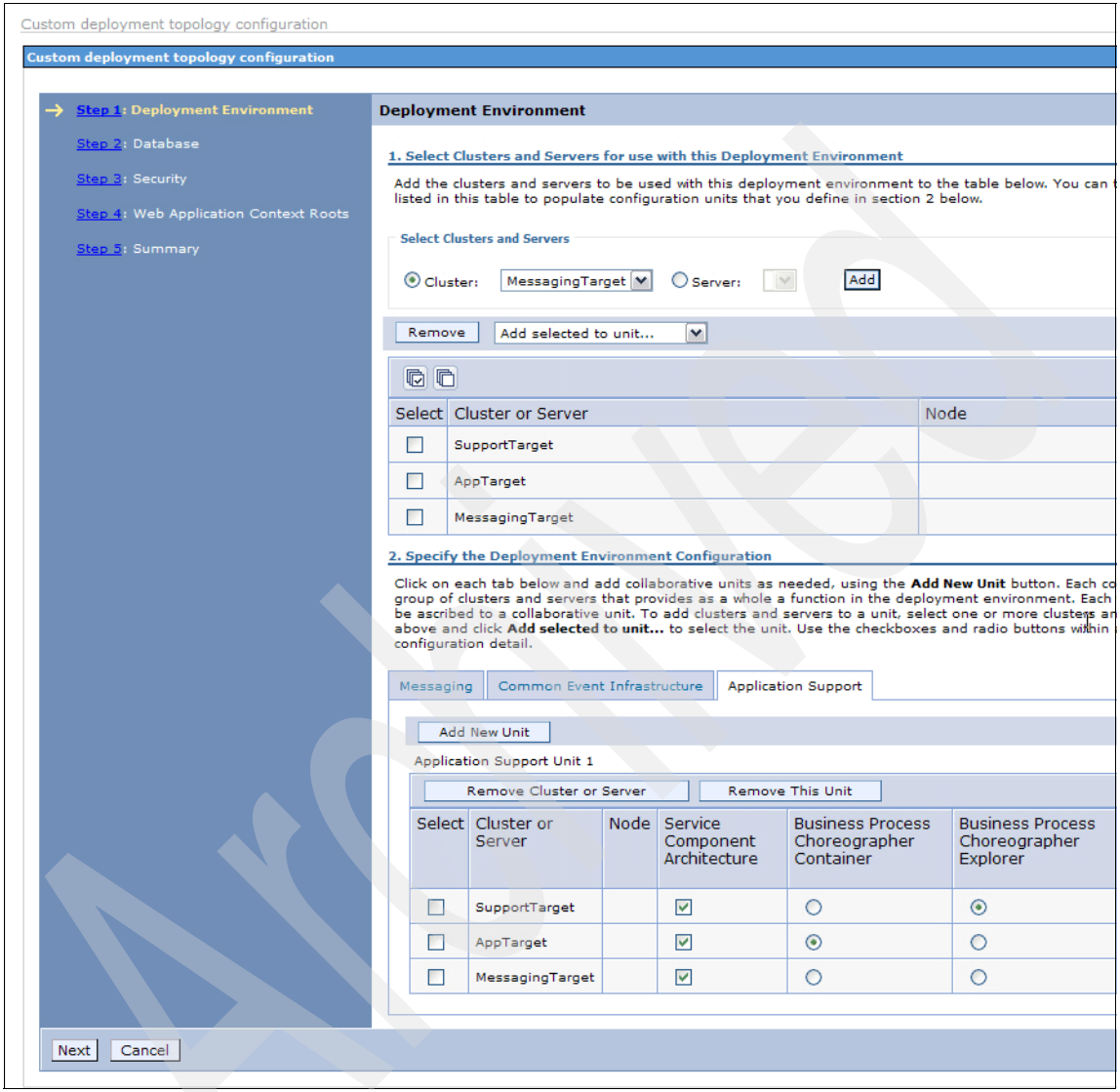


Figure 8-7 Assigning components to the three clusters

13. In the databases window, there are four data sources (one for the BPC database and three for the messaging engines). Fill in the details, making sure that all check boxes in the Create Tables column are cleared, as shown in Figure 8-8. Click **Next**.

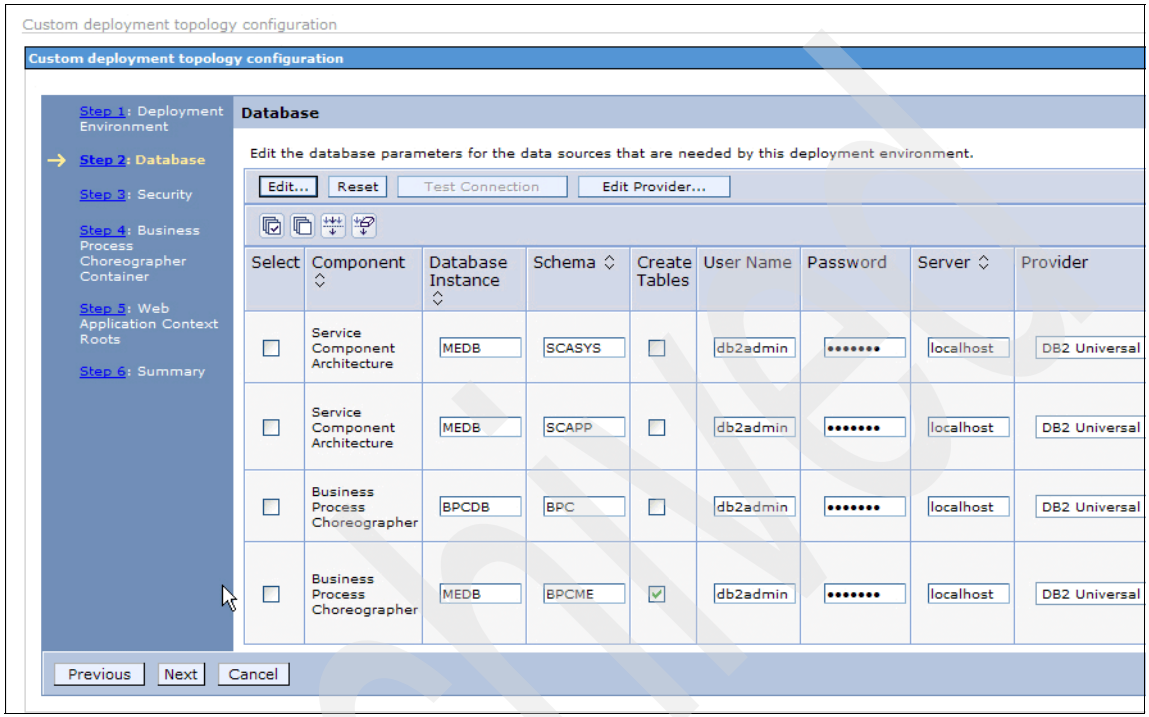


Figure 8-8 Completed data sources window

14. Click the **Step 3: Security** link. In the Security window (Figure 8-9), enter the following values:

   – User: `SCA`
   – Password: `passw0rd`

   Click **Next**.

Figure 8-9 Completed BPC security window

15. In the Business Process Choreographer Container window (Figure 8-10 on page 211), enter the following details:

   a. For the administrator role, use the following values for the User and Group text boxes:

      - User: `wasadmin`
      - Group: `Admins`

   b. For the monitor role, use the following values for the User and Group text boxes:

      - User: `monadmin`
      - Group: `Monitors`

   c. For the JMS API authentication, use the following values for the user and password:

      - User: `jmsapi`
      - Password: `passw0rd`

   d. For the escalation user authentication, use the following values for the user and password:

      - User: `escalation`
      - Password `passw0rd`

e. For the cleanup user authentication, use the following values for the user and password:

- User: `cleanup`
- Password: `passw0rd`

Later on when we enable LDAP, these users and groups must be in the LDAP database. Expand the **Human Task Manager** and clear the **Enable e-mail service** check box. The completed window is shown in Figure 8-10.

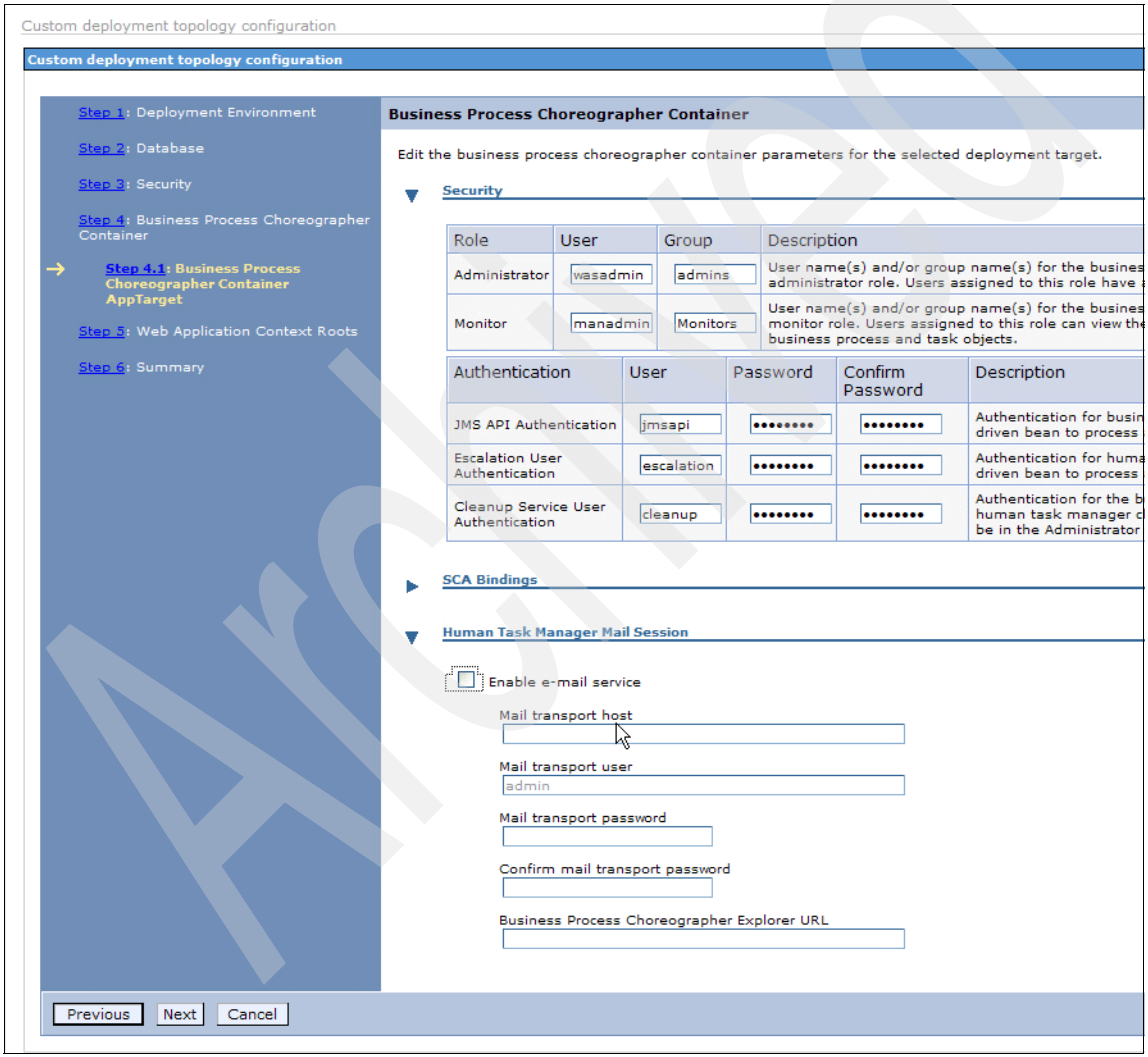


Figure 8-10 Completed BPC container window

16. Click **Next**, because there is nothing to change in the Web Application Context Roots window. The Summary window appears.
17. Click **Finish**.
18. Save the configuration and synchronize the nodes.
19. Click the newly created deployment environment and click **Generate**. A progress window is displayed, as shown in Figure 8-11. Save and synchronize your changes.

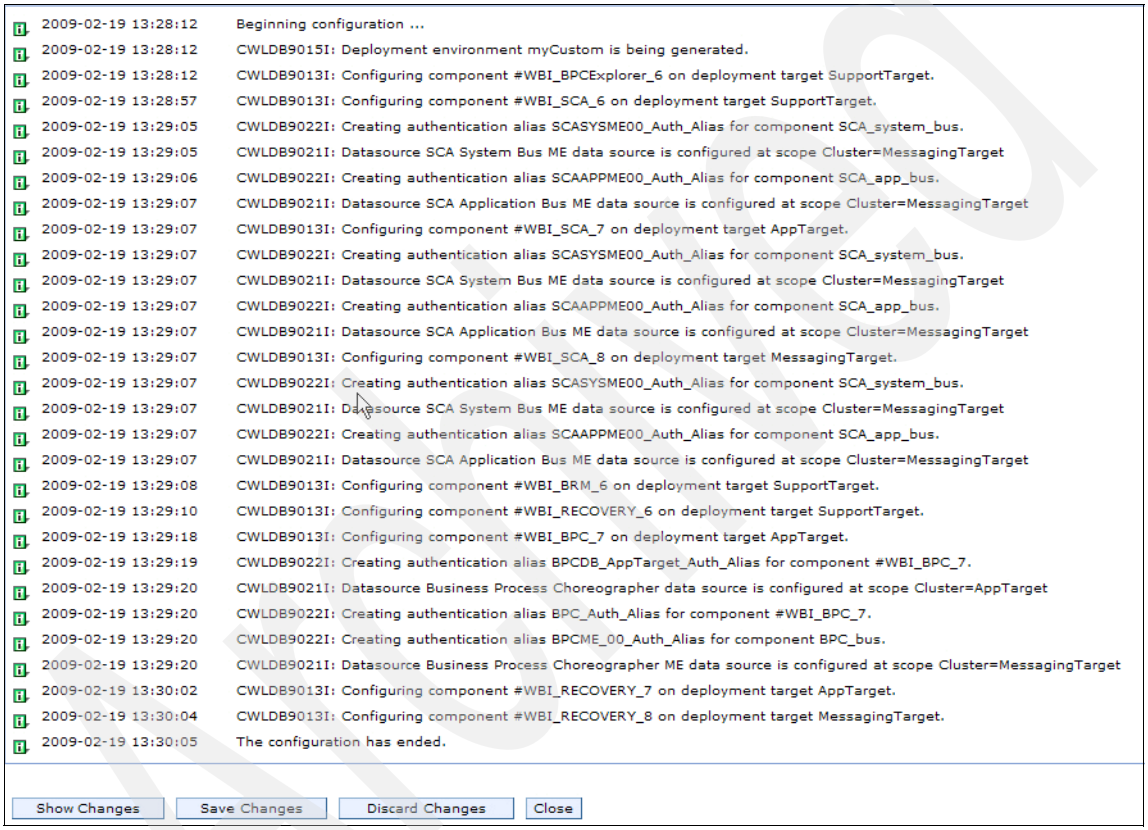


Figure 8-11 Generate Environment logs messages during creation of resources

8.2 Making required post-creation changes

You have now created a custom deployment topology, but before you start, you must make some post-creation changes just as we have done for the Remote Messaging and Remote Support topology:

1. Check that the WebSphere Variable DB2UNIVERSAL_JDBC_DRIVER_PATH is set with the value /opt/ibm/WebSphere/ProcServer/universalDriver_wbi/lib on the AppTarget and Messaging clusters.
2. Set the authentication aliases on the data sources and add the virtual hosts.
3. Start the environment once these changes are made. You are now ready to start your new topology.

Note: At the time of writing it is not possible to start the custom environment from the Servers → Deployment Environments window. You must start the clusters from Servers → Clusters in this order:

1. Messaging
2. Support
3. AppTarget

9

Administering a production topology

This chapter describes administration of a production topology. Administering topologies involves creating and configuring deployment environments, exporting them to other systems, and performing day-to-day operational tasks. This chapter also discusses administrative tasks for other components of your BPM solution. It contains the following sections:

9.1 Administering deployment environments

One of the easier ways to configure and administer your WebSphere Process Server environment is using deployment environments. From the Integrated Solutions Console, click **Servers** → **Deployment Environments** to get to the main window (Figure 9-1).

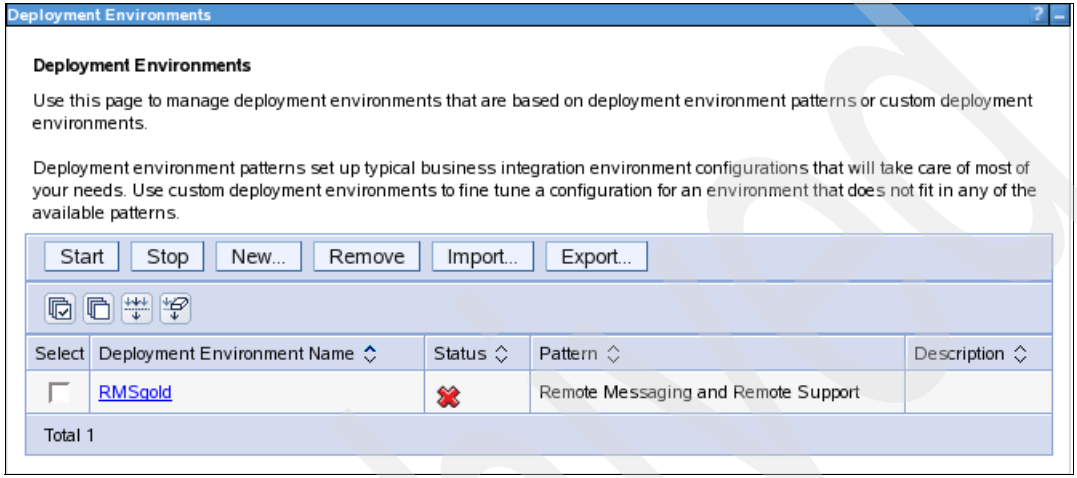


Figure 9-1 Main Deployment Environments window

This window enables you to start and stop existing environments. The **New** and **Remove** buttons enable you to build a new environment based on a pattern or remove an existing environment. The **Export** button creates a backup of configuration patterns and **Import** generates environments based on previously configured environments.

From the Deployment Environments window, you can perform the following tasks, described in the sections that follow:

- "Creating a new deployment environment" on page 216
- "Starting and stopping deployment environments" on page 220
- "Reviewing and changing deployment environments" on page 221
- "Exporting and importing deployment environments" on page 226

Creating a new deployment environment

In Section 7.2.3, "Creating a deployment topology" on page 66, you built a Remote Messaging and Remote Support deployment environment using the deployment environment's **Create new deployment environment** wizard.

To review the wizard:

1. Click the **New** button shown in Figure 9-1 on page 216. The wizard will guide you through naming, choosing and populating the components in the environment. In the **Create new deployment environment** window (Figure 9-2), where you will provide a deployment environment name and runtime capability (WPS or WESB).

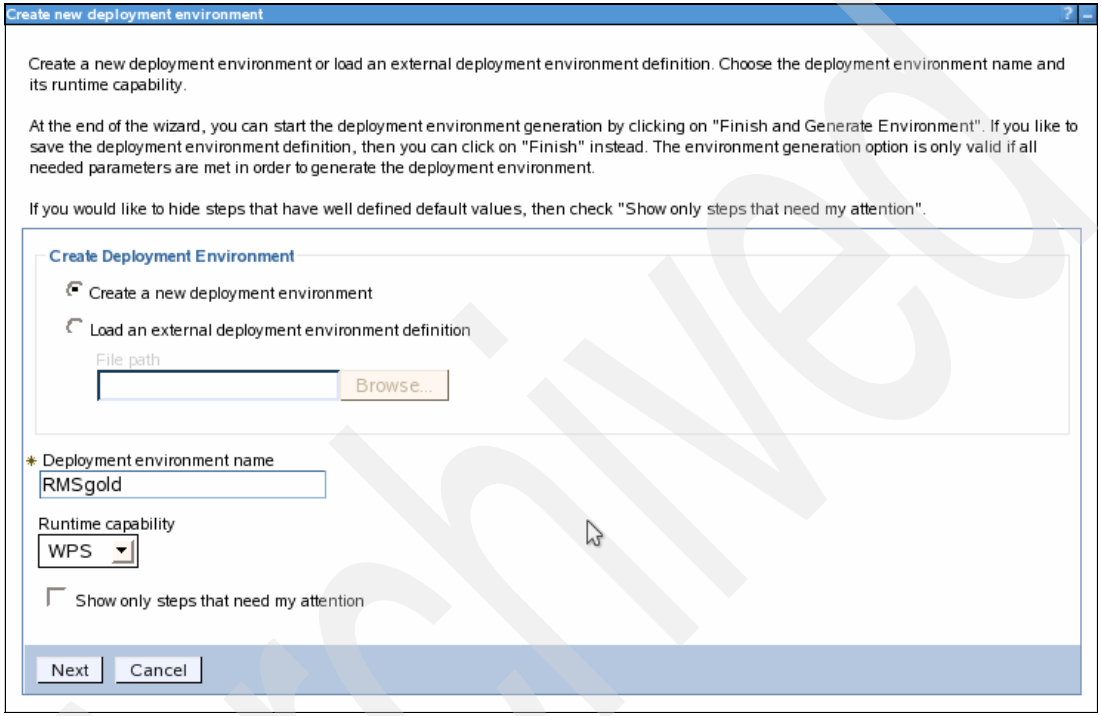


Figure 9-2 Create new deployment environment

2. Enter `RMSgold` in the Deployment environment name text box.

3. Click **Next.** The window in Figure 9-3 will help you decide which deployment environment pattern to choose.

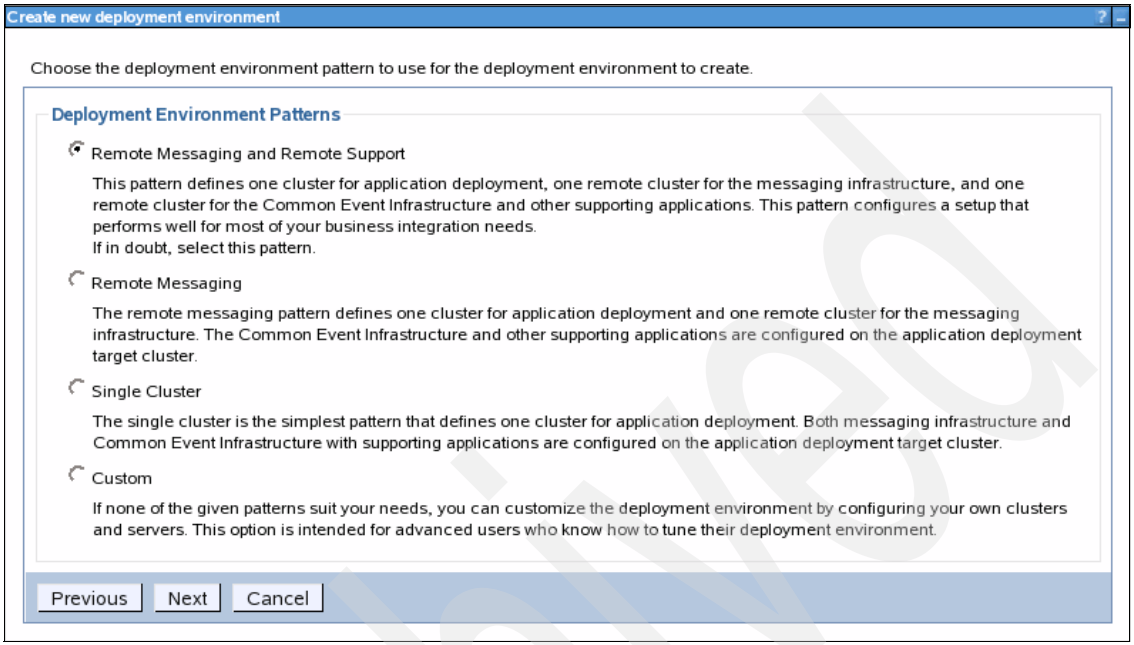


Figure 9-3 Deployment Environment Patterns

4. Click the appropriate radio button.
5. Click **Next.**

The next eight steps require specifying the values that describe the configuration of the deployment environment. Any artifacts that can be discovered by the wizard will be available to you. Figure 9-4 shows the additional steps needed to complete this particular pattern.

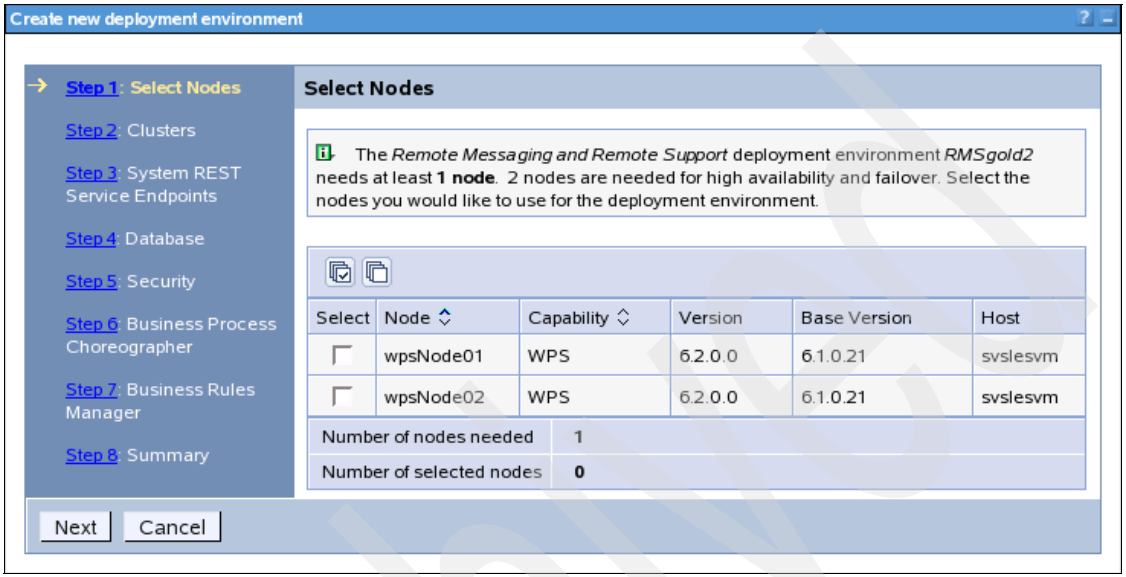


Figure 9-4 Steps to complete creating a new deployment environment

6. Click **Finish** to save the environment definition, or click **Finish and Generate** to generate the entire deployment environment.

Starting and stopping deployment environments

Once the deployment environment pattern is built, it can be centrally managed from the Deployment Environments window of the Integrated Solutions Console.

1. Click **Servers** → **Deployment Environments** to view the configuration information. This window provides a high-level status of the environment.
2. From the window shown in Figure 9-5, you may start or stop the deployment environment. The current status of the environment is stopped.

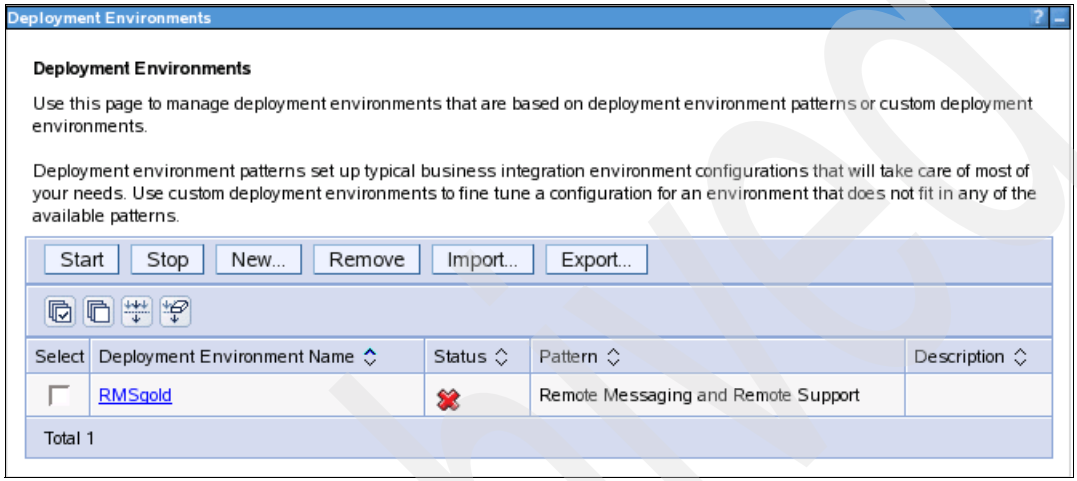


Figure 9-5 A deployment environment with stopped status

To start the environment, perform the following steps:

a. Click the desired environment's check box.
b. Click **Start**. The environment status indicator will turn green.

To stop the environment, perform the following steps:

a. Click the desired environment's check box.
b. Click **Stop**. The environment status indicator will turn red.

> **Note:** New in V6.2, the wsadmin commands **`startDeploymentEnv`** and **`stopDeploymentEnv`** can be used to start and stop a Deployment Environment, for example:
>
> ```
> wsadmin>$AdminTask startDeploymentEnv {-topologyName RMSgold}
>
> wsadmin>$AdminTask stopDeploymentEnv {-topologyName RMSgold}
> ```

3. Verify the deployment environment's status by clicking **Servers** → **Clusters**. Review the status column. The Clusters view refreshes the status on an

interval. Click the refresh arrows to the right of the title status to refresh the status manually.

Note: New in V6.2, wsadmin command **showDeploymentEnvStatus** queries and displays the status of a Deployment Environment from command line, for example:

```
wsadmin>$AdminTask showDeploymentEnvStatus {-topologyName RMSgold}
```

Reviewing and changing deployment environments

To see more details of the environment:

1. Click the **RMSgold** link shown in Figure 9-5 on page 220.

   The configuration window shown in Figure 9-6 can be used to manage the resources of the deployment topology. These resources are the data sources, authentication aliases, deployment topology, and deferred configuration.

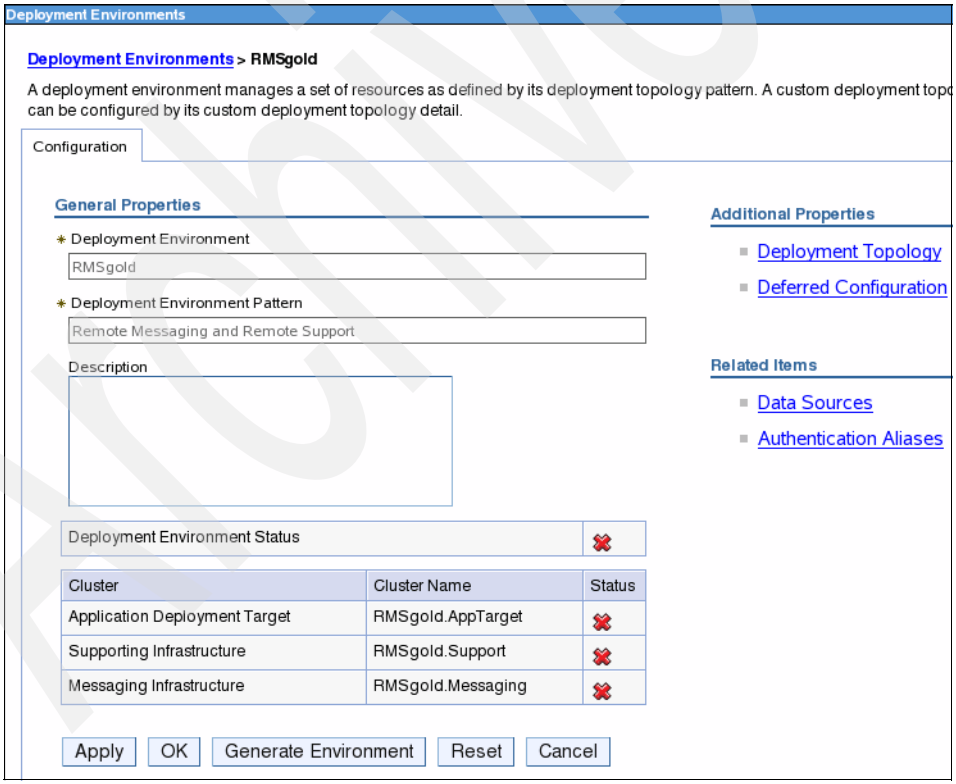


Figure 9-6 RMSgold Configuration window

2. From the **Deployment Environments** → **RMSgold** configuration window (Figure 9-6 on page 221), click **Deployment Topology** under **Additional Properties**. This window shows you the status of the nodes and clusters. It also allows you to increase or decrease the number of cluster members per node or cluster in this environment.

   a. To increase cluster members, perform the following steps in the window shown in Figure 9-7.

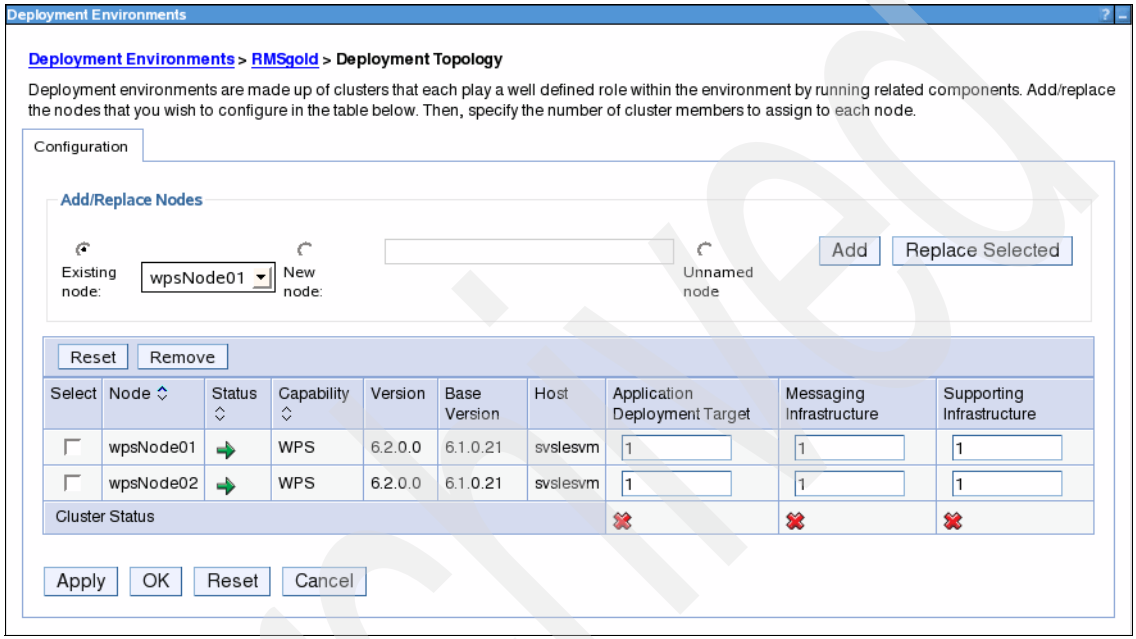


Figure 9-7 RMSgold Deployment Topology window

   i. Select the check box for the `wpsNode01` row.

   ii. In the Application Deployment Target column, increase the number for the `wpsNode01` row to 2.

   iii. Click **OK** and **Save**.

   iv. Review the **Servers** → **Application Servers** window. Ensure that there is an additional server in stopped state.

   v. Start the deployment environment, cluster, or server so that it can be deployed and managed by the environment.

b. To decrease cluster members, perform the following steps in the window shown in Figure 9-7 on page 222:

   i. Select a the check box for the `wpsNode01` row.

   ii. In the Application Deployment Target column, decrease the number for the `wpsNode01` row to 1.

   iii. Click **OK** and **Save**.

   iv. Click **Servers** → **Application Servers** to verify that the number of servers has decreased.

3. From the **Deployment Environments** > **RMSgold** configuration window, click **Data sources** under **Related Items**. From the Data sources window shown in Figure 9-8, you can perform the tasks listed after Figure 9-8.

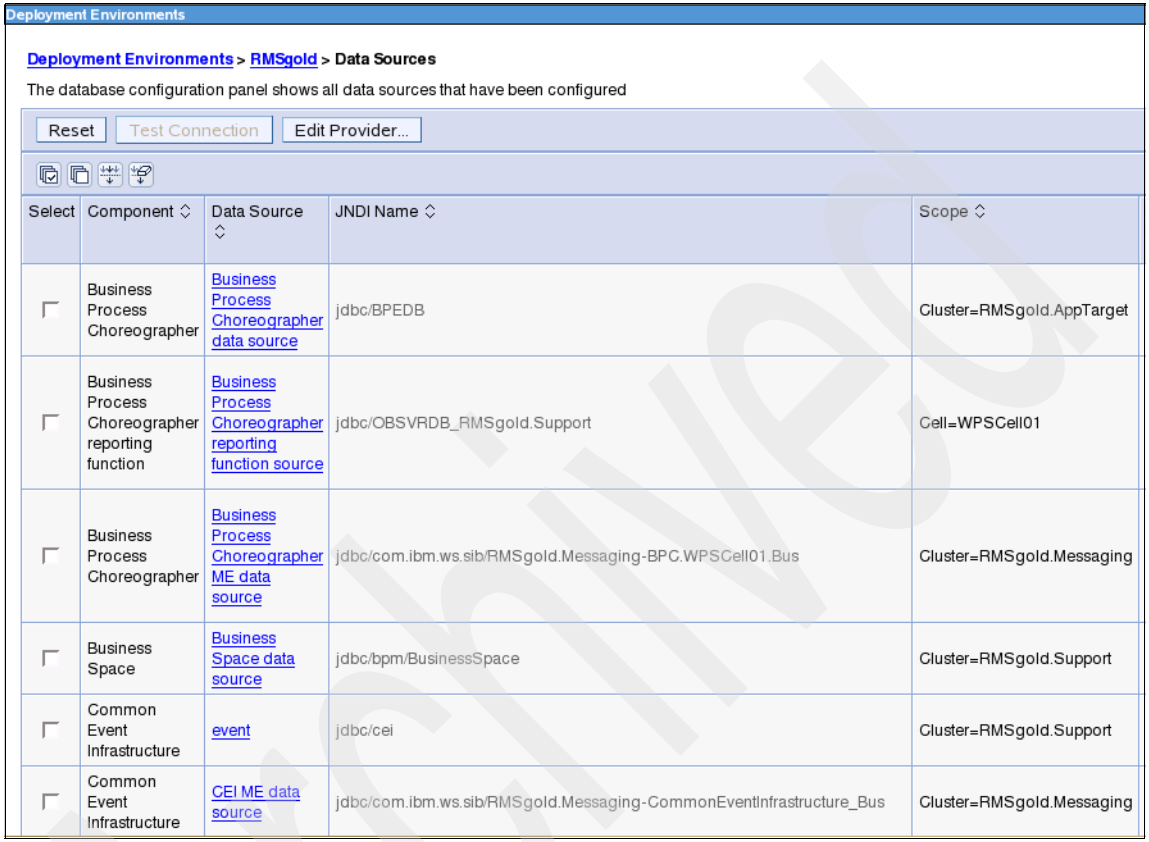


Figure 9-8 Data Sources window

- Review the Data sources defined for this deployment environment.
- Alter the following fields in the data source configuration as necessary:
  - Instance
  - Schema
  - User
  - Password
  - JDBC Provider
- Test the connection prior to saving the configuration.
- Edit the database provider.

To edit the database provider, perform the following steps:

i. Check **Business Process Choreographer**.

ii. Click **Edit Provider**.

iii. Edit the Database Provider Configuration window values shown in Figure 9-9. Select the appropriate node and driver paths.

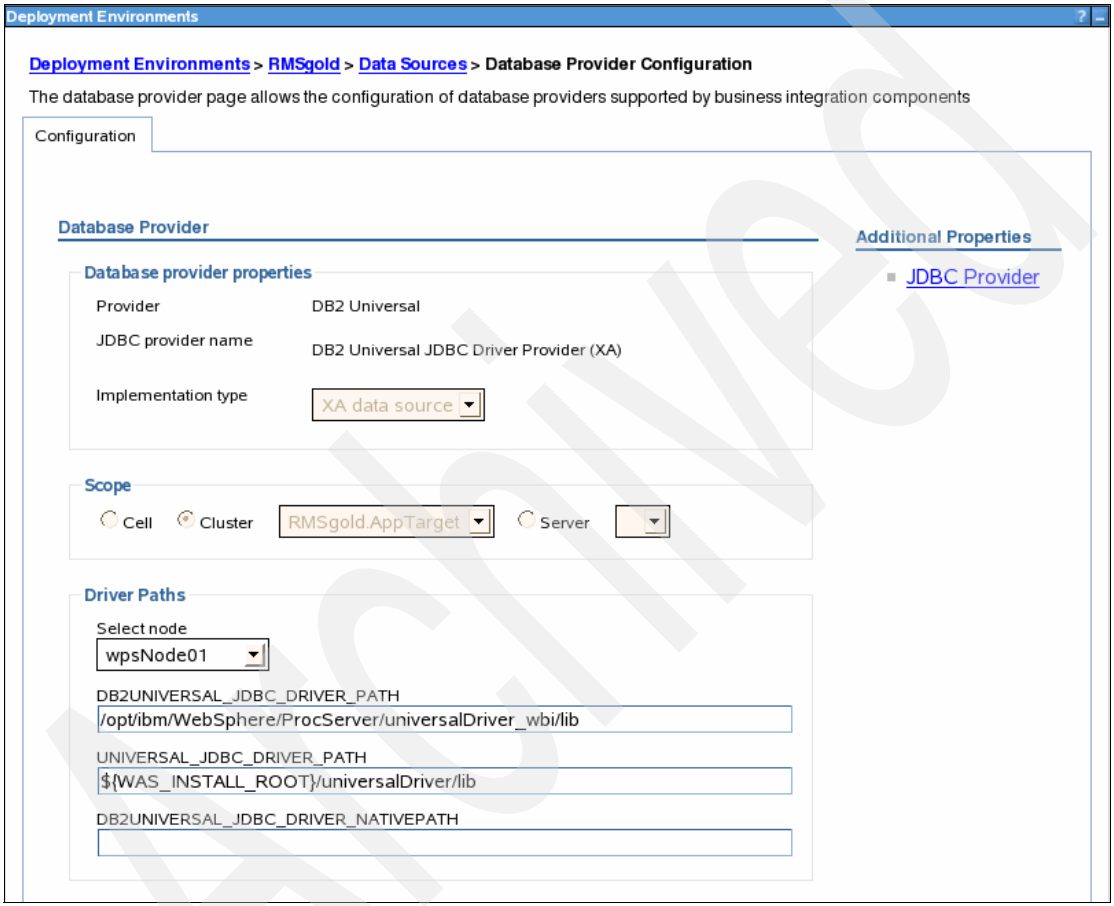


Figure 9-9 Database Provider Configuration window

iv. Click **OK** and click **Save** after changes have been made.

v. Generate and restart the environment for these changes to be propagated.

4. From the **Deployment Environments** → **RMSgold** configuration window, click **Authentication Aliases**. Under Related Items, you can change the user name and password for all exposed authentication aliases. These changes will be propagated out once you generate the environment.

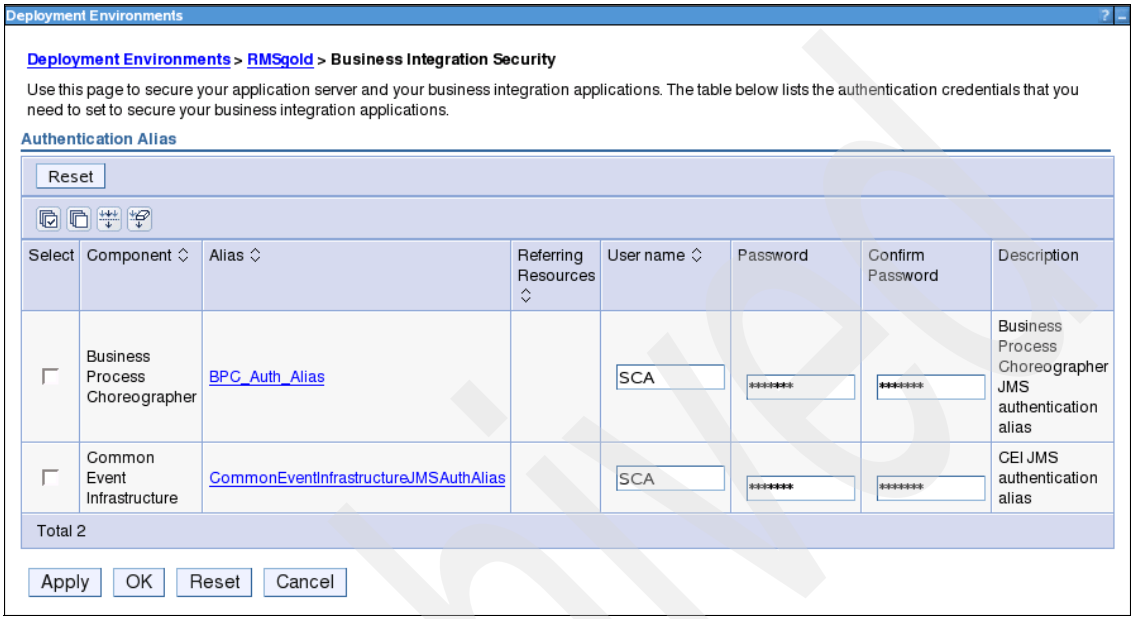


Figure 9-10 Authentication Alias window

To change the user name and password:

a. Edit the user name text box with a new value.

b. Edit the password text box with a new password.

c. Edit the confirm password text box with a new password.

d. Click **OK** and **Save**.

e. Generate and restart the environment for these changes to be propagated.

Exporting and importing deployment environments

Exporting and then importing your configuration is an efficient method of promoting an environment from system or stress testing to user acceptance testing, then to a production environment. To do this, configure the deployment environment based on a pattern, such as Remote Messaging and Remote Support. As testing progresses, you adjust the configuration to add or subtract Application Target cluster members based on throughput requirements.

Important: A 1:1 relationship of cluster members to nodes or other clusters is not necessary. For example, the application target can have eight cluster members and messaging and support only two cluster members.

Once the RMSGold environment is ready to be promoted to a test environment (for example, RMSUAT), follow these steps to build the RMSUAT environment:

1. Export the RMSgold deployment environment. This is an XML file.
2. Make a copy of the generated XML file and name it `RMSUAT.xml`.
3. Review the `RMSUAT.xml` file to verify that the values are correct for your new environment.

   Important: The hostName text box will be changed automatically for you during the import. The host name will be derived from the federated nodes into the new RMSUAT cell.

4. Perform the following steps to edit `RMSUAT.xml` to match the new environment.
   a. Open the XML file in an editor.
   b. Delete the name value pairs of deferredConfigTime and deferredConfigUser from the RMSUAT.xml code shown in Example 9-1. This is the audit message displayed when you clicked **Configuration Done** in the Deferred Configuration window.

Example 9-1 Top of generated deployment environment export XML file

```
<?xml version="1.0" encoding="ASCII"?>
<wbitopology:WBITopology xmi:version="2.0"
xmlns:xmi="http://www.omg.org/XMI"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wbitopology="http://www.ibm.com/wbi/schemas/6.1/wbitopology.xmi"
name="RMSgold" version="6.2.0.0"
deferredConfigTime="2009-02-16T18:17:53.346-0500"
deferredConfigUser="wpsadmin">
  <pattern id="Reference" name="Remote Messaging and Remote Support"
version="6.2.0.0"/>
```

   c. Convert the RMSgold environment naming to RMSUAT:
      i. Complete a find and replace of RMSgold with RMSUAT.
      ii. Review the names of your clusters, service integration bus names, and the scope of the authentication aliases.
   d. Change the database server name and port change, if needed.

e. If the cell name changes, it must be edited in the authentication alias and service integration bus names. Example 9-2 shows the Support Topology's CEI database component.

Example 9-2 Support topology entry in RMSUAT.xml file

```
<components id="WBI_CEI" name="WBI_CEI" version="6.2.0.0"
topologyRole="Support" baseRuntimeId="WAS" level="1">
    <dataSrc component="WBI_CEI" createTable="false"
dbcomponent="WBI_CEI_EVENT">
      <authAlias name="WPSCell01/RMSUAT.Support/EventAuthDataAliasDB2"
userName="uatinst1" password="{xor}OzOvPiws" component="WBI_CEI"
description="CEI Event data source authentication alias"
dbcomponent="WBI_CEI_EVENT"/>
      <properties name="databaseName" value="EVENT" type=""/>
      <properties name="driverType" value="4" type=""/>
      <properties name="serverName" value="uatDB2" type=""/>
      <properties name="portNumber" value="50000" type=""/>
      <attributes name="jndiName" value="jdbc/cei"/>
      <attributes name="name" value="event"/>
      <attributes name="description" value="Event server data source"/>
      <attributes name="dataStoreHelperClassName"
value="com.ibm.websphere.rsadapter.DB2UniversalDataStoreHelper"/>
      <provider scope="Cluster=RMSUAT.Support"
databaseType="DB2_UNIVERSAL" providerType="DB2 Universal JDBC Driver
Provider" implementationType="XA data source"
dbcomponent="WBI_CEI_EVENT"/>
    </dataSrc>
```

f. Verify that the userName, serverName, and port are correct for the new environment. Examine the rest of the file for other values that must be changed.

g. Save the file.

5. Move it to a location where you will run the Integrated Solutions Console.
6. From the Integrated Solutions Console, click **Server** → **Deployment Environments** and then click **Import**.
7. Click **Browse**.
8. Locate the `RMSUAT.xml` file.

9. Select the **Show only steps that need my attention** check box, as shown in Figure 9-11.

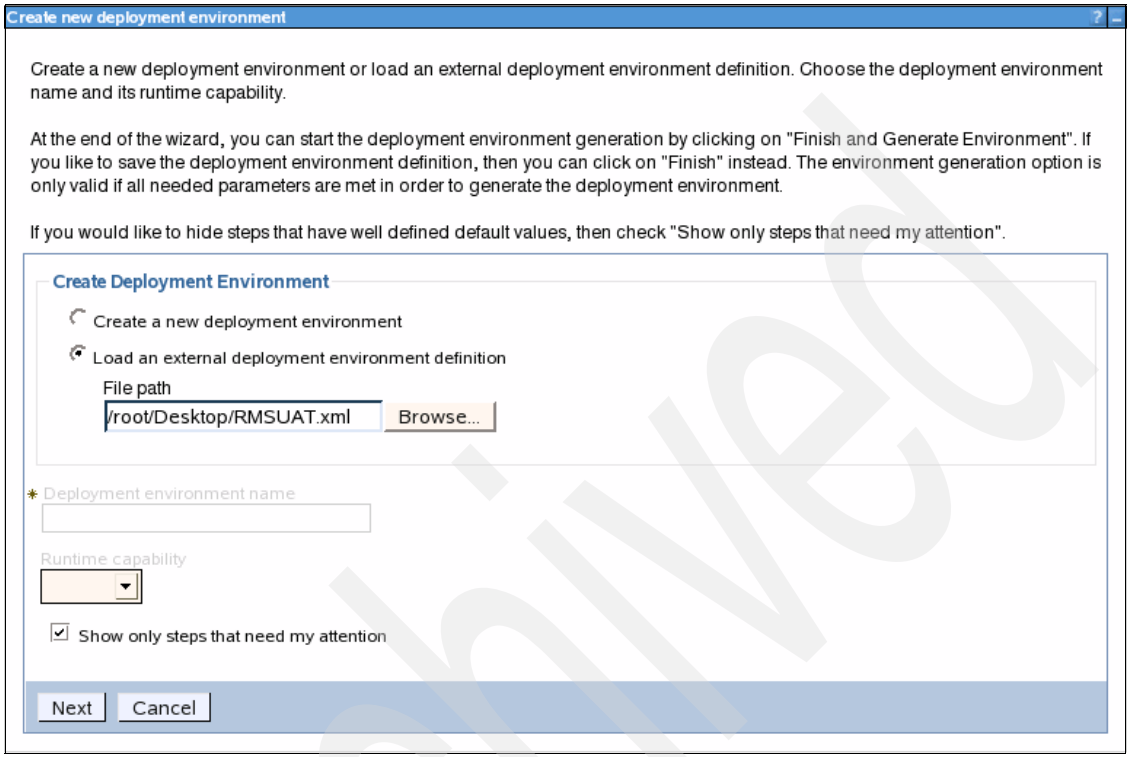


Figure 9-11 Import Deployment Environment from exported XML file

Note: Because the RMSGold name changed to RMSUAT on line 2 of the file, the text box for deployment environment name is not required. If you use the same name the wizard requires a new name.

10. Complete the steps in the Import Wizard as you did when creating a new deployment environment. The Summary page is as shown in Figure 9-12.

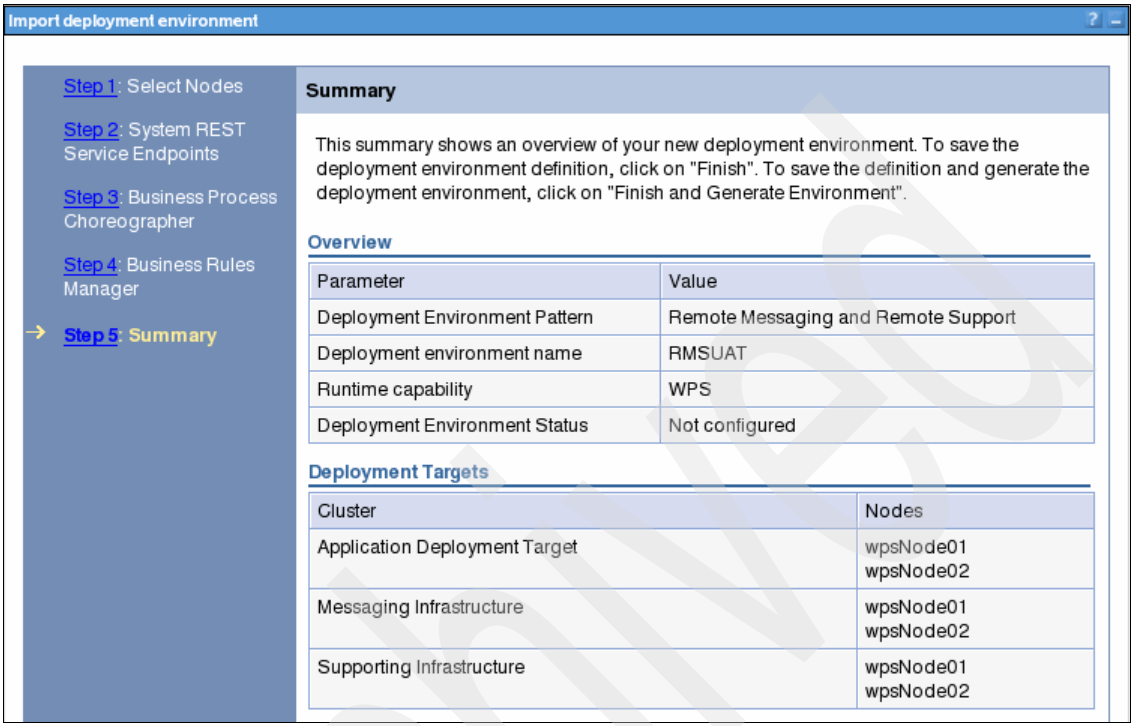


Figure 9-12 Import deployment environment Summary window

9.2 Administering Business Process Choreographer

This section describes administration considerations for the Business Process Choreographer.

Using compensation

This function is enabled by default. To verify that the Compensation Service is enabled:

1. Click **Servers** → **Application Servers** → **RMSGold.AppTarget.wpsNode01.0**.
2. Click **Container Services** → **Compensation Service**.
3. **Enable service at server startup** should be selected.

4. Adjust options based on your system's needs:
   - The compensation handler retry limit defaults to unlimited retries.
   - The compensation handler retry interval defaults to 30 seconds.

> **Note:** This service is enabled at the server level, not the cluster level.

Improving the performance of Business Process navigation

A long-running process spans multiple transactions. By default, a transaction is triggered by a Java Messaging Service (JMS) message. To improve the performance of process navigation, you can configure the Business Flow Manager to use a work-manager-based implementation for triggering transactions instead of JMS messages. Refer to the following Web page for more information:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.websphere.bpc.620.doc/doc/bpc/t5tuneint_processnavigation.html

9.3 Administering Common Event Infrastructure

In WebSphere Process Server V6.2, Common Event Infrastructure is configured to standard practices. Disabling the event data store can give you better performance with less maintenance. All events will be distributed by the event service. To disable this data store:

1. Click **Service Integration** → **Common Event Infrastructure** → **Event service**. Click **Event services** under Additional Properties. Click **Default Common Event Infrastructure event server**.
2. Clear the Enable event data store check box, as shown in Figure 9-13.

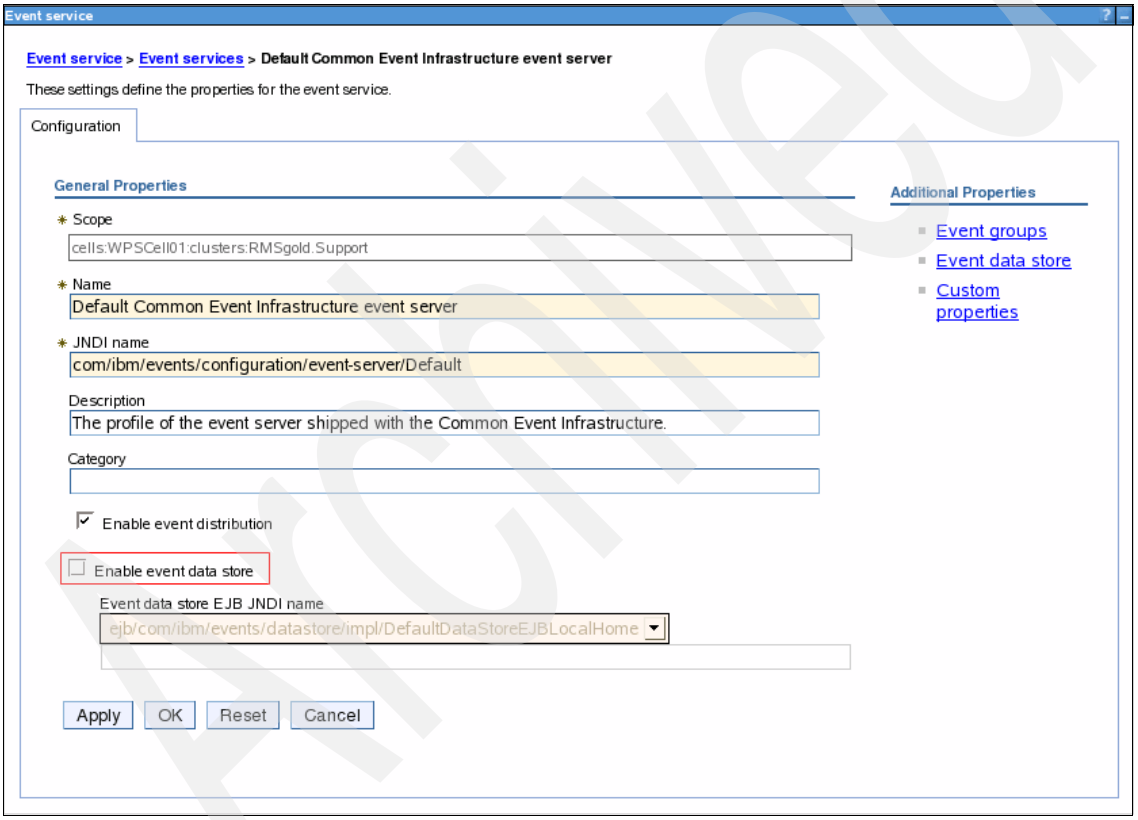


Figure 9-13 Disable create event data store

3. Click **OK**, then click **Save**.
4. Restart **RMSgold.Support**.

For additional performance best practices, refer to IBM Redpaper publication *IBM WebSphere Business Process Management V6.1 Performance Tuning*, REDP-4431.

9.4 Changing a database password

One common administrative problem is changing a database password to comply with corporate security guidelines. This section describes a method for accomplishing this task without an outage. With any change to an authentication alias, the server using this alias must be restarted. It is nearly impossible not to disrupt any in-flight processing, change a password to the database with only one user ID, change the authentication alias, and restart the server.

However, this can be done using clusters and two database IDs. You will want to work with the database administrator to create two database IDs that can be used to access the same tables with the same privileges. The trick is to stagger the database user IDs' password expiration. If you have a requirement to change the password once per month, then one ID should expire on the *first* day of the month and the other on the *fifteenth*.

For this demonstration, let us call the users First and Fifteenth. You bring the system up on January 1st and the authentication alias is set to the user First. You now have 30 days to change the database password. On January 15th, Fifteenth's password gets changed by the database administrator. Some time between January 15th and February 1st, the WebSphere Process Server administrator should change the authentication alias from First to Fifteenth. Once the authentication aliases are changed, the administrator should issue a ripple start of the cluster using the database. This is issued by checking the cluster by clicking the **Ripplestart** button from the **Servers** → **Clusters** window in the Integrated Solutions Console shown in Figure 9-14 on page 234.

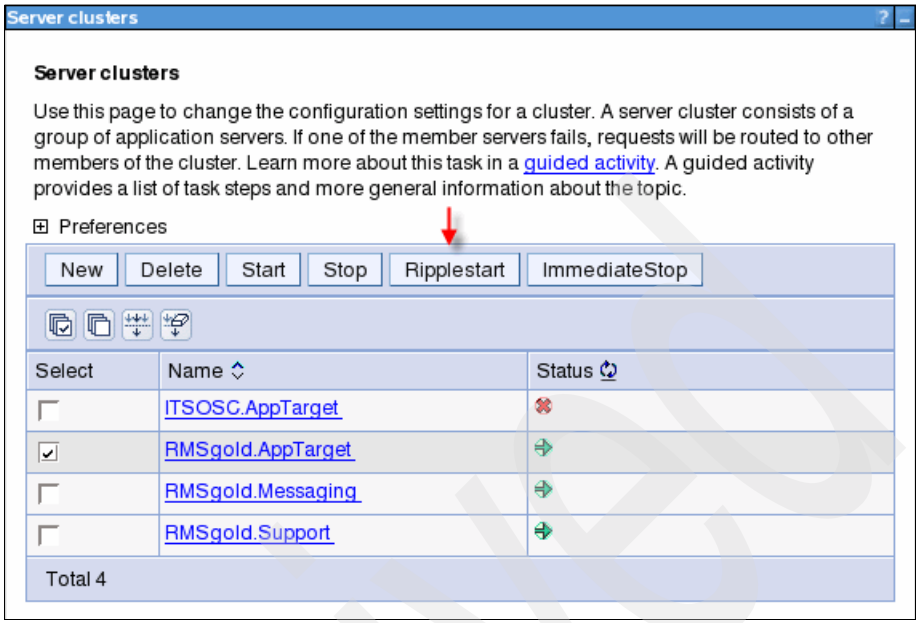


Figure 9-14 Ripplestarting a cluster to pick up new password

This stops one server in the cluster at a time. When the server stops it quiesces the incoming work and completes it before stopping. When the server restarts, it uses the Fifteenth user ID to make database calls. This server takes on new work while the next cluster member is quiescing work to restart. This happens until all cluster members are restarted.

Important: As long as the JDBC connections are XA compliant, any in-flight transaction is coordinated by the transaction manager. If your processes access only one resource manager, queue, or database per transaction, then XA compliance is not be a concern.

9.5 Managing failed events

This section provides guidance on using Failed Events Manager in WebSphere Process Server.

9.5.1 What is an event

An event is a Service Data Object (SDO) that is received by a WebSphere Process Server application. An SDO is made up of data and a reference to the business operation which should be executed by the application. When WebSphere Process Server receives the event, the SDO is processed by the appropriate business application based on the referenced business operation.

Every system based on business processes contains events. There are always processes and events that fail. The expectation is that a well-developed application is designed by business knowledgeable people and the business should know how to best handle failed events and process. The application's exception and fault handling code is responsible for handling business failures.

Most system level failures appear as a communication issue. There are two types of communication:

- Synchronous

  Synchronous communication is blocking. A call is initiated and the thread waits for a response before processing further. In case of failure, the invoking application is responsible for failure capture and retry logic. There is no administrative action available for a WebSphere Process Server administrator.

- Asynchronous

  Asynchronous communication is not blocking. The call is initiated and the event is placed on a queue. The receiving process is listening on the queue to process the event and reply to the calling process. If there is business exception or fault in the receiving process, the application is responsible for failure capture and retry logic. There is no administrative action available for a WebSphere Process Server administrator.

If two SCA components are communicating asynchronously, and there is a failure (such as the system is not available), WebSphere Process Server has built-in retry logic. Five retries is the default. If the retry logic fails, the event is considered failed, and the WebSphere Process Server Recovery Service (WPSRS) moves the event to the failed event queue. The WPSRS persists the event into a database. The WebSphere Process Server administrator can take administrative action using the Failed Event Manager.

Important: Because adapters are an asynchronous technology, configurations that make use of adapters have high chance of generating failed events.

9.5.2 How to use the Failed Event Manager

The Failed Event Manager, which is built into the Integrated Solutions Console, is a Web-based tool that enables an administrator to view, modify, resubmit, or delete the failed events.

To launch the Failed Event Manager:

1. Open the Integrated Solutions Console.
2. Click **Integration Applications** → **Failed Event Manager**.

The Failed Event Manager (Figure 9-15) allows you to search for failed events. There are seven default searches and one custom search. If there is a system failure, you may narrow your desired result set by either destination or date.

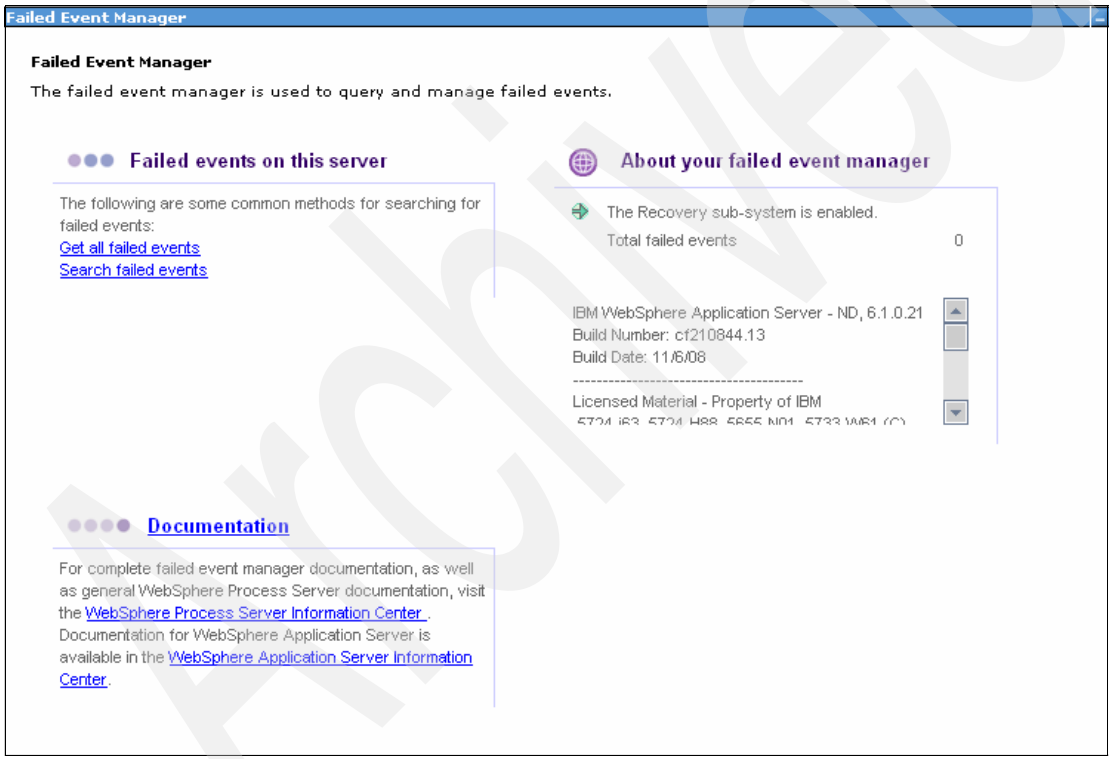


Figure 9-15 Failed Event Manager main window

Important: If the About your failed event manager section (shown in Figure 9-15) says that the recovery sub-system is disabled, verify that the SCA container is started. In this configuration, it is the RMSGold.AppTarget cluster. If this does not enable the recovery sub-system, then review the following Web page from the support site:

http://www-1.ibm.com/support/docview.wss?rs=2307&uid=swg21293460

Once your search is complete, you may need to take some action. Administrators and operators are allowed to take action on the Failed Event Manager. If you are in another role, then you will not see the buttons shown in Figure 9-16.

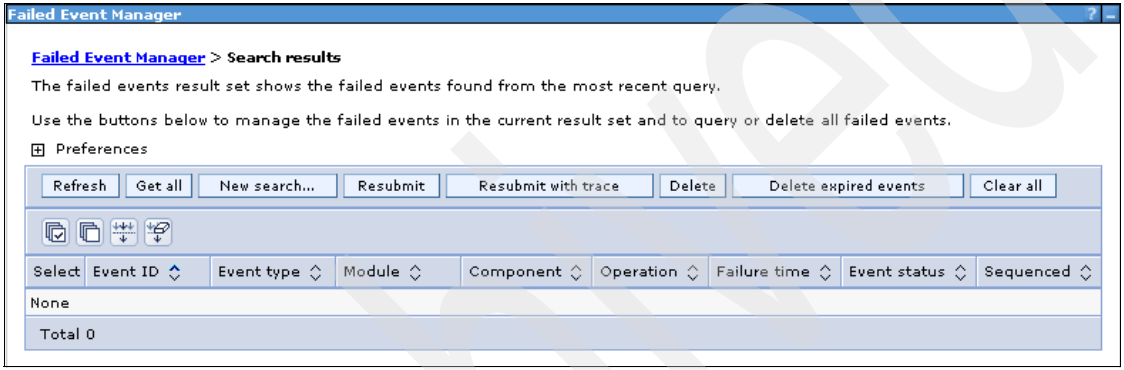


Figure 9-16 Failed Event Manager actions

The Failed Event Manager shows you information about the failed event so that you can take some action on it.

Note: Actions taken on failed events are business related. For example, the resubmission of a failed event might result in a financial transaction of some time (such as money being spent). Therefore, system administrators need business and application knowledge to act on failed events.

If the destination module was stopped and this was the reason that the event failed, you should resubmit the event as follows:

1. Check the box in the select column next to the event that you wish to resubmit.
2. Click **Resubmit**.
3. Click **Refresh**.

This should clear the event. If it still appears with a new failure time, resubmit with trace to discover why the event failed, as follows:

1. Check the box in the select column next to the event that you wish to resubmit.
2. Click **Resubmit with trace**.
3. From the Resubmit with trace window, specify the Trace Control text box with trace specification.
4. Click **Resubmit**.

> **Important:** You cannot resubmit an event that has expired. If the event has not expired, you can edit the expiration date prior to resubmitting.

When a failed event has expired or you do not wish to resubmit it, then you delete this event. There are three options in the Failed Event Manager window to do this:

- Delete

  Click this button to delete a specific event.
- Delete expired events

  Click this button to delete any events with an expired date.
- Clear all on server

  Click this button to delete all events in the Failed Event Manager.

10

Securing a production topology

This chapter addresses securing WebSphere Process Server for the Remote Messaging and Remote Support topology pattern. As the primary administrator, you will configure security for the cluster, including how to customize the integration with an LDAP server.

10.1 Securing a BPM topology

A Business Process Management (BPM) infrastructure must be properly secured. Out of the box, WebSphere Process Server comes with the following:

- File registry security
- SSL enabled and configured with central key management
- Messaging infrastructure roles assigned
- Database access configured
- A secured Integrated Solutions Console
- People directory provider set up to retrieve users and groups from the configured security repository

This is a good start. You will need to adapt this configuration to your company's security policies and current infrastructure. This includes encrypting communications with processes external to your WebSphere Process Server cell, configuring your company's user repositories, and mapping groups to administrative roles.

10.2 Setting up SSL infrastructure

In WebSphere Process Server V6.2 SSL is centrally managed and configured by default. You only need to configure external resources, such as Tivoli Directory Server, DB2, and HTTP Server. We recommend the following functions:

- Add the signer certificate from your LDAP server and database to the WebSphere Process Server trust store at the proper scope. By default, this would be the cell scope.
- The WebSphere plug-in will be populated with the WebSphere Process Server signer certificate for you. You may have to propagate the KDB file to the HTTP Server.

For more information about the new SSL central management feature refer to the IBM WebSphere Developer Technical Journal article *SSL, certificate, and key management enhancements for even stronger security in WebSphere Application Server V6.1*, available at the following Web page:

http://www.ibm.com/developerworks/websphere/techjournal/0612_birk/0612_birk.html

Important: If you are planning cross-cell single-sign on, then you must exchange signer certificates with the other cell.

10.2.1 Available user account repositories

There are four supported user account repositories that you can select when configuring security.

- Federated repositories
- Local operating system
- Standalone LDAP
- Standalone custom registry

For a BPM configuration, you will want to use federated repositories. A federated repository allows you to search multiple providers with one query.

Federated repositories still give you the flexibility to use LDAP, a custom registry, or both. WebSphere Business Monitor requires the use of federated repositories, so you will be configuring federated repositories using LDAP rather than a standalone LDAP registry.

Important: Any time that you are using more that one machine, the local operating system user account registry is not supported.

10.2.2 Enabling security to use LDAP

To enable the use of federated repository:

1. Start the deployment manager.
2. Sign in to the Integration Solutions Console using the user ID `admin` and the password `admin`.
3. Click **Security** → **Secure administration, applications, and infrastructure**.
4. On the Secure administration, applications, and infrastructure panel, look under User account repository and click **Configure**, as shown in Figure 10-1.

Figure 10-1 Configure button

5. On the Federated repositories panel, under Related Items, click **Manage repositories**, as shown in Figure 10-2.

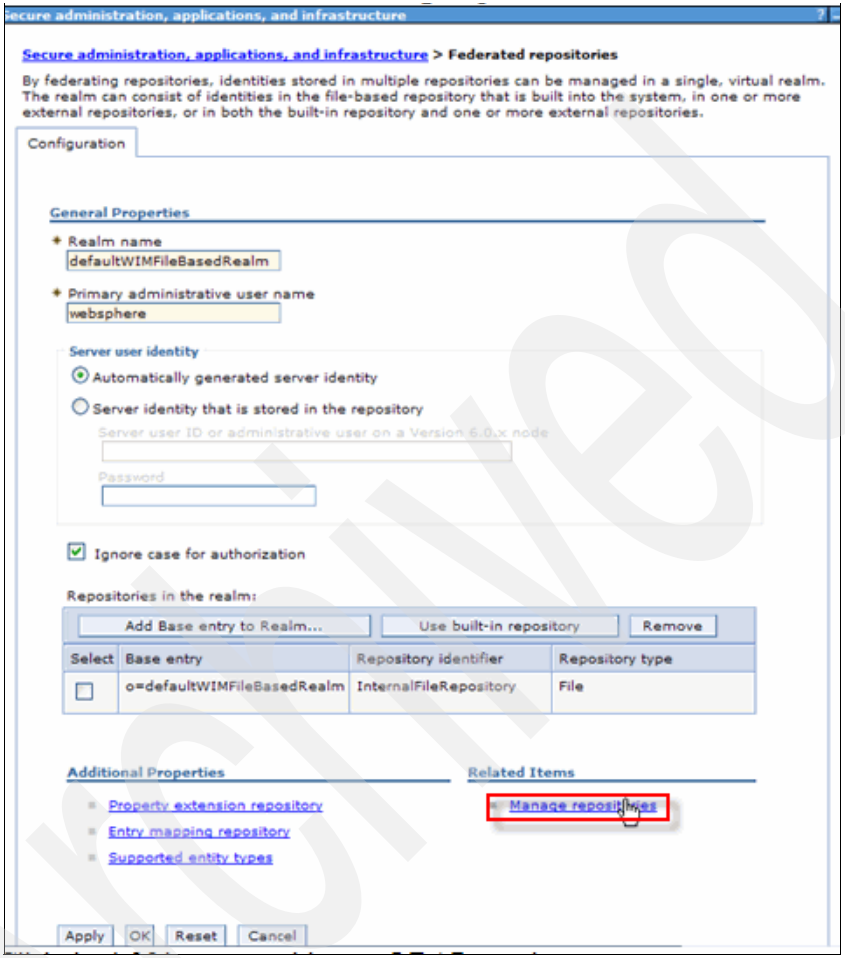


Figure 10-2 Federated repositories configuration panel

On the Manage repositories panel, there should be one entry for the file Internal file repository that is used by default (Figure 10-3).

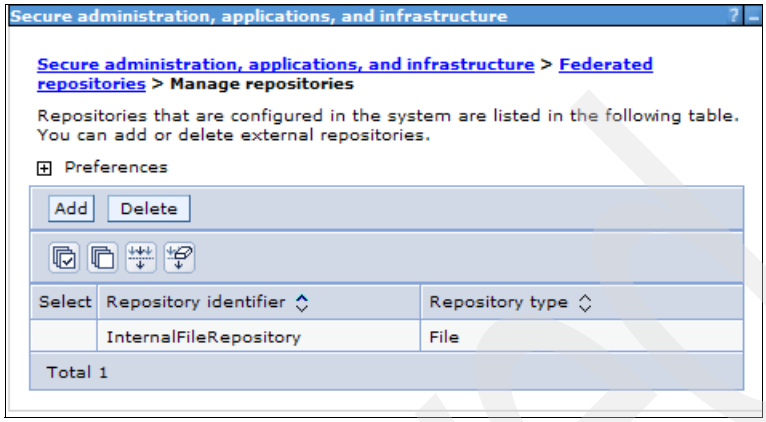


Figure 10-3 Manage repositories panel

6. Click **Add** to create our new federated repository.

7. On the New Repository panel (Figure 10-4) set the following fields:
   a. Set Repository identifier to `wbi602a.raleigh.ibm.com`.
   b. Set Directory type to `IBM Tivoli Directory Server Version 6`.
   c. Set Primary host name to `wbi602a.raleigh.ibm.com`.
   d. The Port value of 389 is correct.
   e. Set Bind distinguished name to `uid=wps,cn=People,O=IBM`.
   f. Set Bind password to `passw0rd`.
   g. Click **OK**.

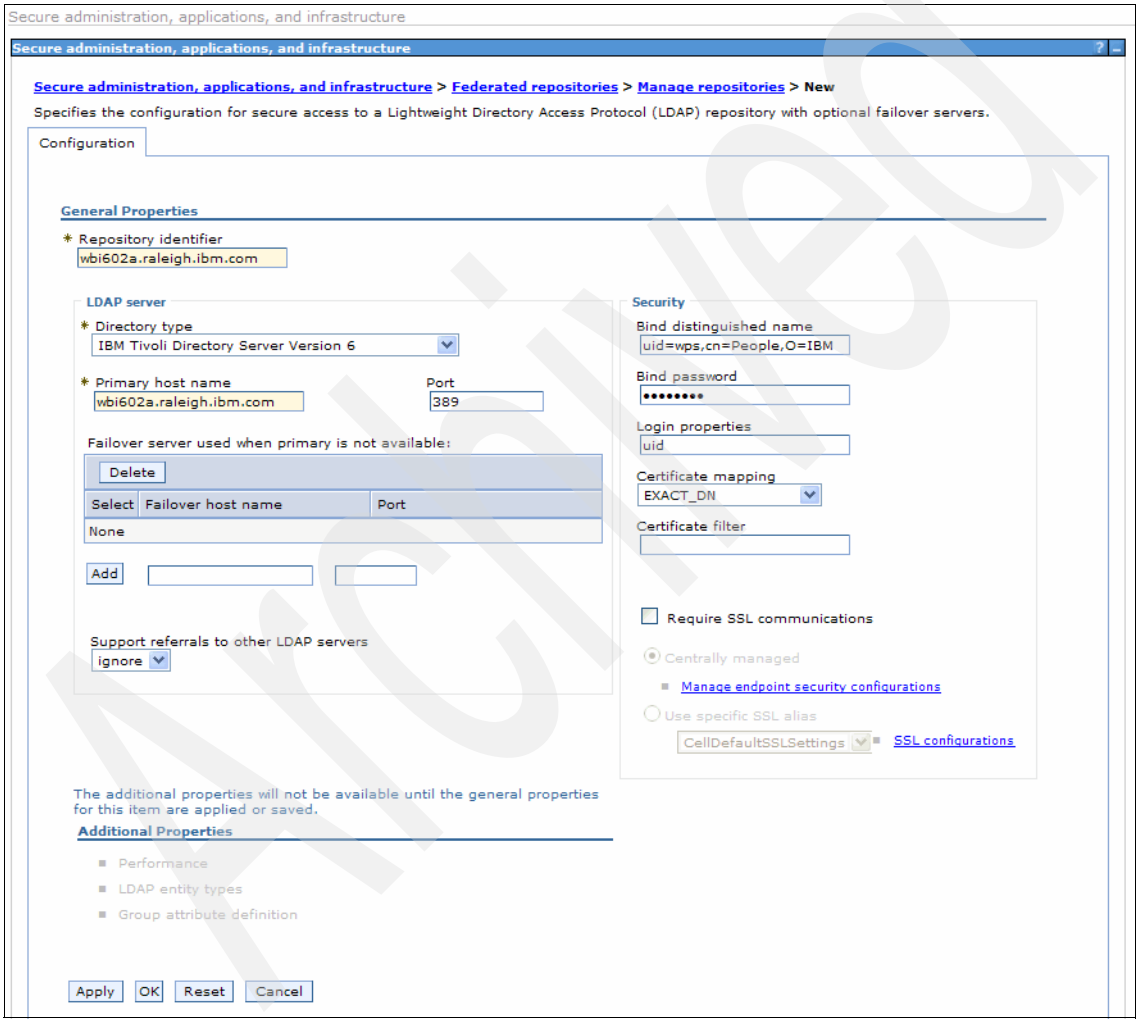


Figure 10-4 New repository configuration panel

   h. Click **Save** to save your configuration changes.

8. At this point, we should now see two federated repositories listed, as shown in Figure 10-5.

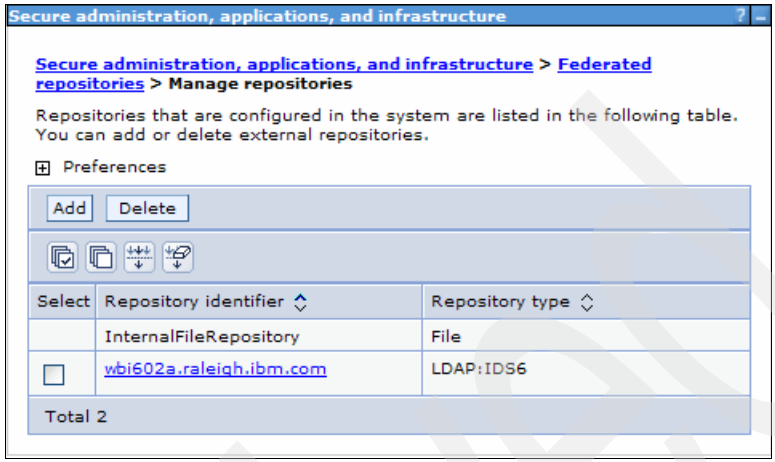


Figure 10-5 Manage repositories panel

9. Click the bread crumb **Federated repositories** to return to this panel.
10. Click **Add Base entry to Realm**, as shown in Figure 10-6.

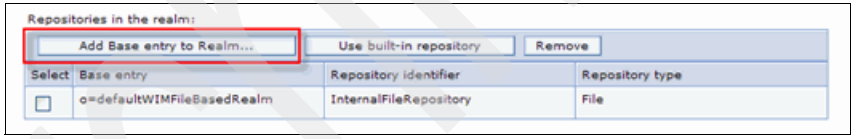


Figure 10-6 Add Base entry to Realm

11. On the next panel (Figure 10-7) set the following fields:
    a. Set “Distinguished name of a base entry that uniquely identifies this set of entries in the realm” to `cn=People,O=IBM`.
    b. Set “Distinguished name of a base entry in this repository” to `cn=People,O=IBM`.
    c. Click **OK**.
    d. Click **Save** to save your configuration changes.

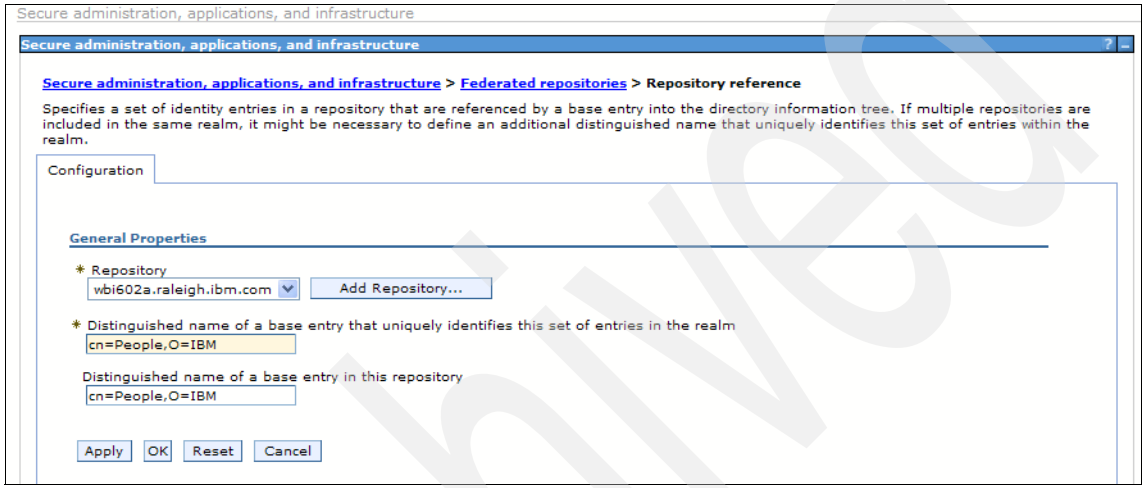


Figure 10-7 Configuration tab

12. The next step is to remove the default InternalFileRepository repository. Note that this is an optional step and it is not required. To remove this repository:
    a. Click the bread crumb to get to the Federated repositories panel.
    b. Check the check box for the **Repository Identifier** that has the value **InternalFileRepository**.
    c. Click **Remove**.
    d. Click **Save** to save your configuration changes.

13. There is one more change to make in order to finish enabling the use of LDAP security. Click the bread crumb to get to the Federated repositories panel. The panel shown in Figure 10-8 should now appear. Change the following:

    a. Set Realm name to `wbi602a.raleigh.ibm.com`.

    b. Set Primary administrative user name to `wps`.

    c. Click **OK**.

    d. Click **Save** to save your configuration changes.

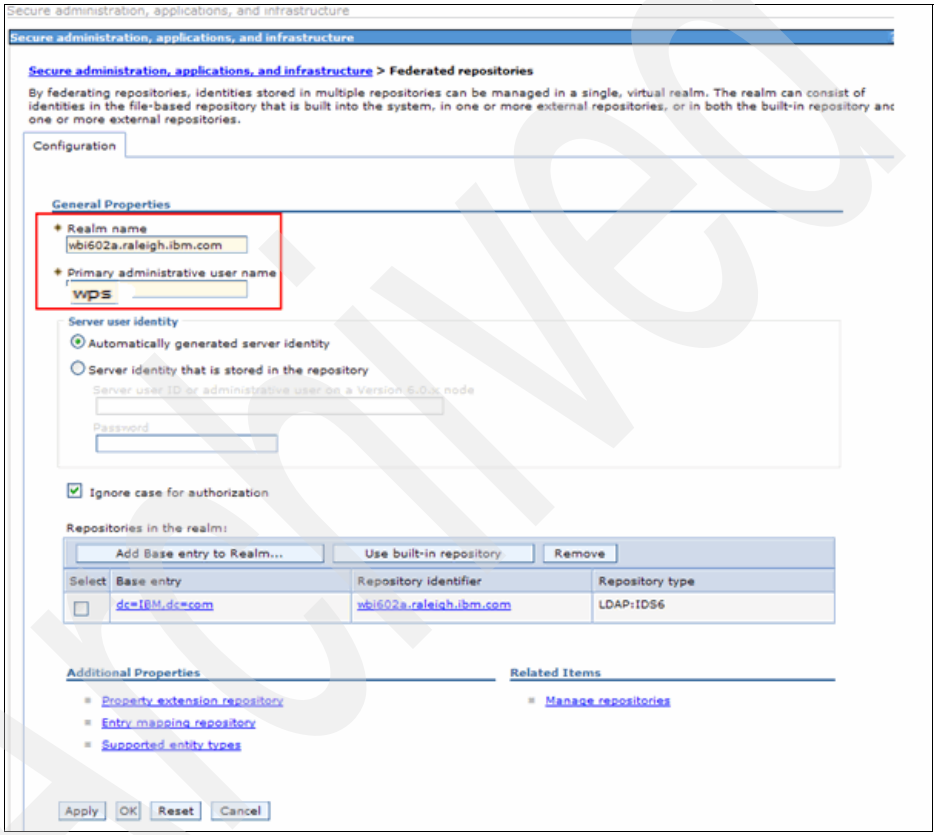


Figure 10-8 General Properties

14. To summarize at this point, we have now changed our security such that instead of using a file-based repository, we are now using LDAP. When we first created our deployment manager profile and set administrative security, we specified a user ID of admin with a password of admin. Now what we have done is changed such that we will now use user ID *wps* with password *passw0rd* when signing into the Integrated Solutions Console in the future.

15. It is necessary to stop the deployment manager and then restart it.
16. When the deployment manager has finished starting, bring up a browser and try to sign in to the Integrated Solutions Console with the new user ID and password.
17. This is also a very good time to go and examine the user IDs that currently exist and are set to admin. They should be modified to use our new LDAP user ID wps. Follow these steps:
    a. Go to the Integrated Solutions Console.
    b. Click **Security** → **Secure administration, applications, and infrastructure**.
    c. Expand **Java Authentication and Authorization Service** and then click **J2C authentication data**. The list of authentication aliases will appear as Figure 10-9 illustrates.

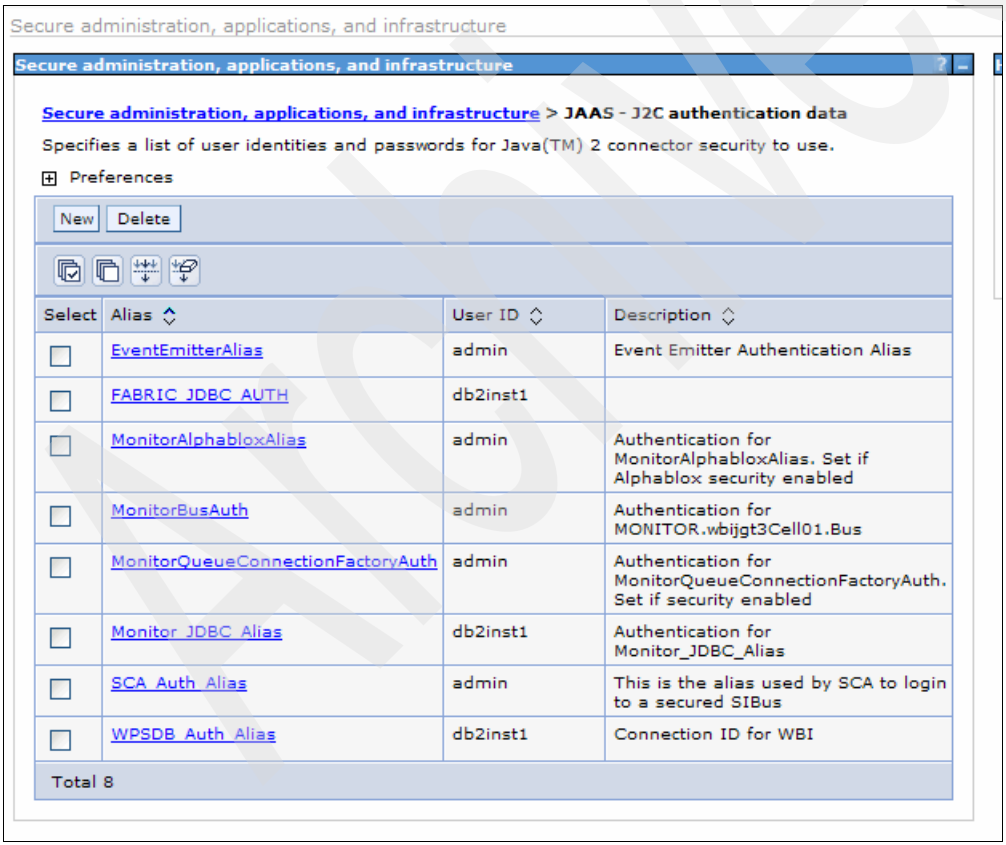


Figure 10-9 List of authentication aliases created after creating a deployment manager profile

d. Examine all user ID values looking for *admin* and change them to *wps*. The password should be *passw0rd*.

e. Save all changes made.

10.2.3 Administrative security for LDAP

Now that all of the repository definition is completed, you will configure global security:

1. Select **Federated repositories** from the Available realm definitions list box in the User Account Repository section of the Global Security window.
2. Click **Set as current**.
3. Select the **Enable Administrative Security** check box.
4. Clear the Use Java 2 security check box.
5. If you are enabling Java 2 security, select the **Warn if applications are granted custom permissions** check box, to debug any initial problems.

6. Make sure that the Service Component Architecture (SCA) modules you are deploying are Java 2 security-ready. Your window should resemble Figure 10-10.

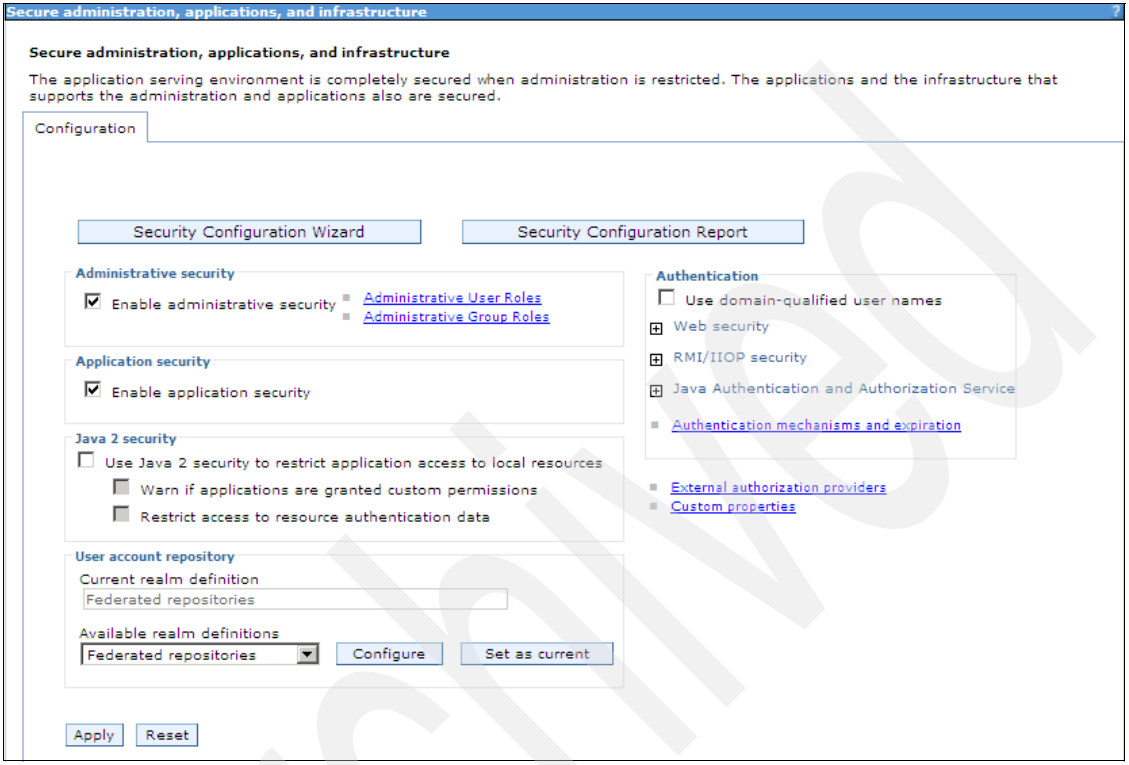


Figure 10-10 Global security settings window

7. Click **System administration** → **Save Changes to Master Repository**.
8. Check **Synchronize changes with Nodes**.
9. Click **Save**.

> **Important:** Make sure that all of your nodes are currently running in the cell, otherwise the synchronize changes with nodes will only synchronize with running federated nodes. If your nodes are out of synchronization, there is a command-line tool called synchNode that must be executed from the profiles directory and the node must be stopped.

10. Restart your node managers and deployment manager.

11. Perform the following steps to verify your configuration by querying a user from the LDAP repository:
    a. Open the Integrated Solutions Console and log in as wps, the primary admin ID.
    b. Click **Users and Groups** → **Manage Users**.
    c. Enter `leon` or another user from your populated repository.
    d. Click **Search**.

The user leon should return an entry as shown in Figure 10-11.

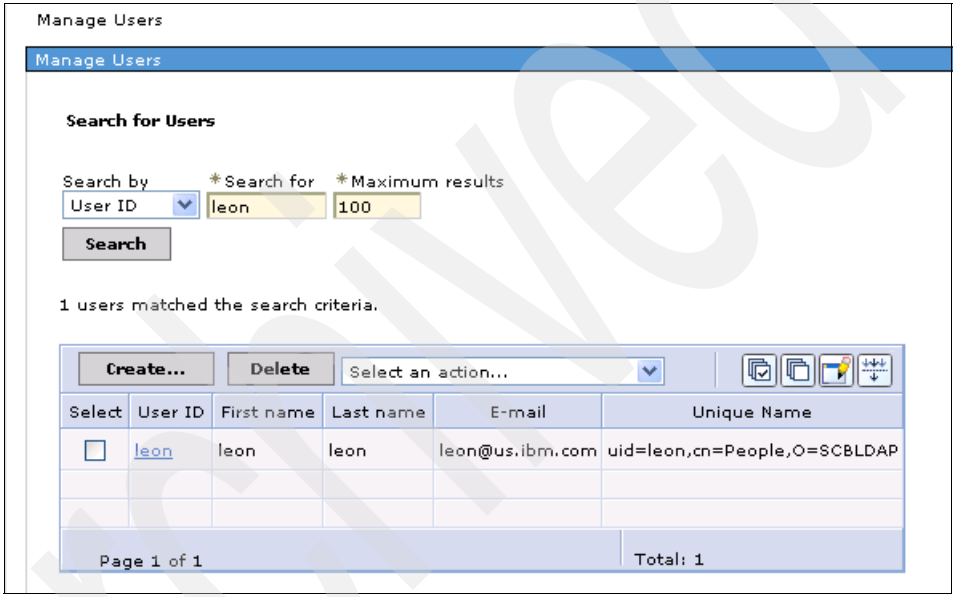


Figure 10-11 User search result

Note: We recommend mapping groups to the administrative roles, thus limiting the number of people using the primary admin identity.

10.2.4 Service integration bus security

To verify that the service integration bus is secured:

1. Launch the Integrated Solutions Console.
2. Click **Service Integration** → **Buses**.

3. Review the Security column. Each entry should be Enabled. Figure 10-12 shows that all of the buses are secured.

Figure 10-12 Service integration bus security window

4. Click **Enabled** for any of the buses to see the security configuration shown in Figure 10-13.

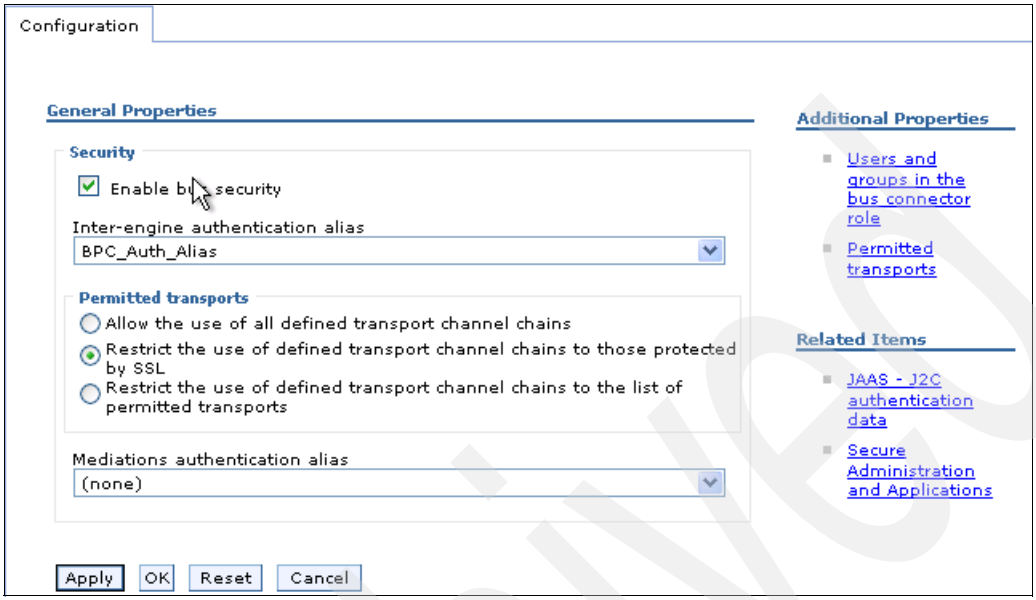


Figure 10-13 BPC bus security

5. Review the configuration. Ensure that the **Restrict the use of defined transport channel chains to those protected by SSL** radio button is selected (to restrict non-SSL channel chains).
6. Click **OK** and **Save**.
7. Click **Buses** → **Security for bus BPC.<Cell Name>.Bus**.

8. Under Additional Properties click **Users and groups in the bus connector role**. This window, shown in Figure 10-14, allows you to add and delete users and groups from this role.

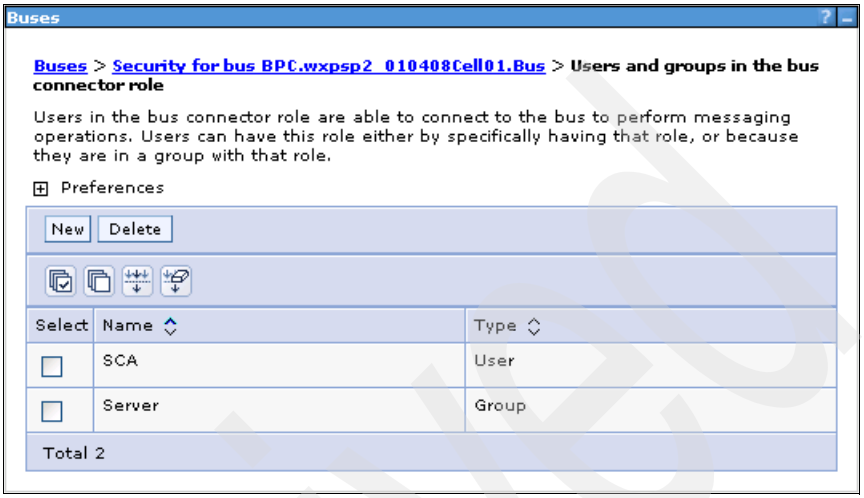


Figure 10-14 Users and groups with bus roles

9. Click **New**. This launches the Create user or group in the bus connector role window, shown in Figure 10-15. In this window, you may grant permissions to an existing user or group.

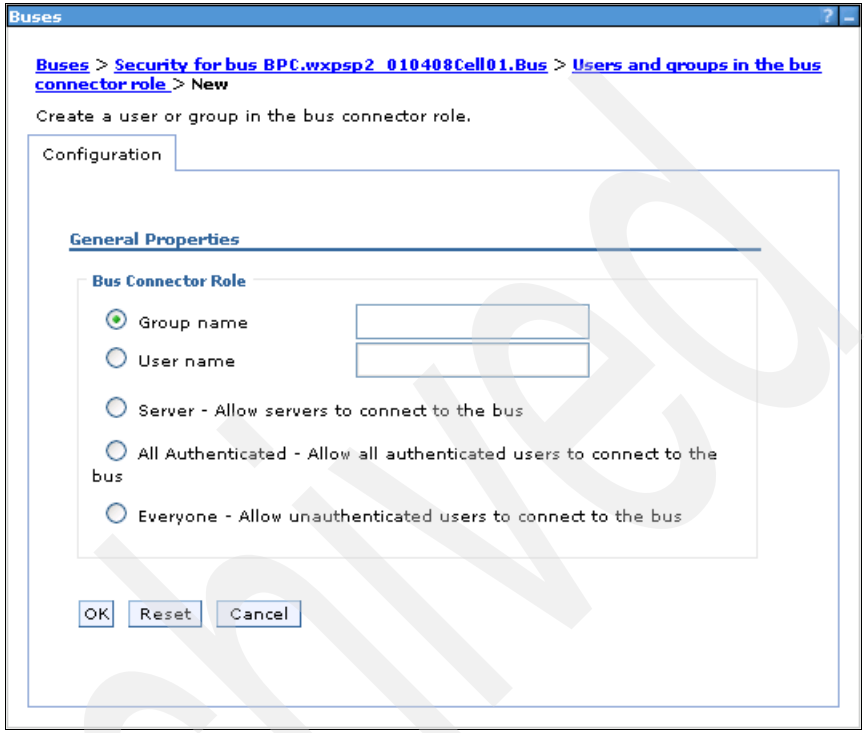


Figure 10-15 Add a group or user with a bus role

10. Click **Group name** or the appropriate radio button.
11. Enter the group or user name that you want to permit to connect to the bus. This would apply in a cross-cell configuration, as described in 4.6.1, “Creating a secured link between two cells” on page 90.

10.2.5 Map groups to administrative roles

To configure the administrative user and group roles, you must log in as either wps or the server’s primary admin ID. This gives you the authority to map other groups and users to roles. The first role that you should assign is the adminsecuritymanager, because this allows you to delegate authority without sharing the primary administrative user name and password.

For this example, you want to map the admins group to be administrators. Users in this group will have nearly full administrator privileges. The only access that they will not have is to map users and groups to administrative roles. Using Table 10-1, map the groups to roles.

Table 10-1 Groups for administrative roles

Group	Role	Description
admins	Administrator	All users in the admins group will have access to change anything in the cell, except mapping administrative roles.
wpsuser	operators	All users in the wpsuser group will have access to start and stop anything in the cell.
security	Adminsecuritymanager	All users in the security group can assign users and groups to the administrative roles.

Mapping groups to roles

To map groups to roles:

1. Log in to the Integrated Solutions Console as the primaryAdminID (wps in this book).
2. Click **Users and groups** → **Administrative Group Roles**.
3. Click **Add**.
4. Enter `security` in the Group name text box.

5. Click **adminsecuritymanager**, as shown in Figure 10-16. To assign multiple roles to a particular group, press Ctrl and click the role.
6. Click **OK** and **Save**.

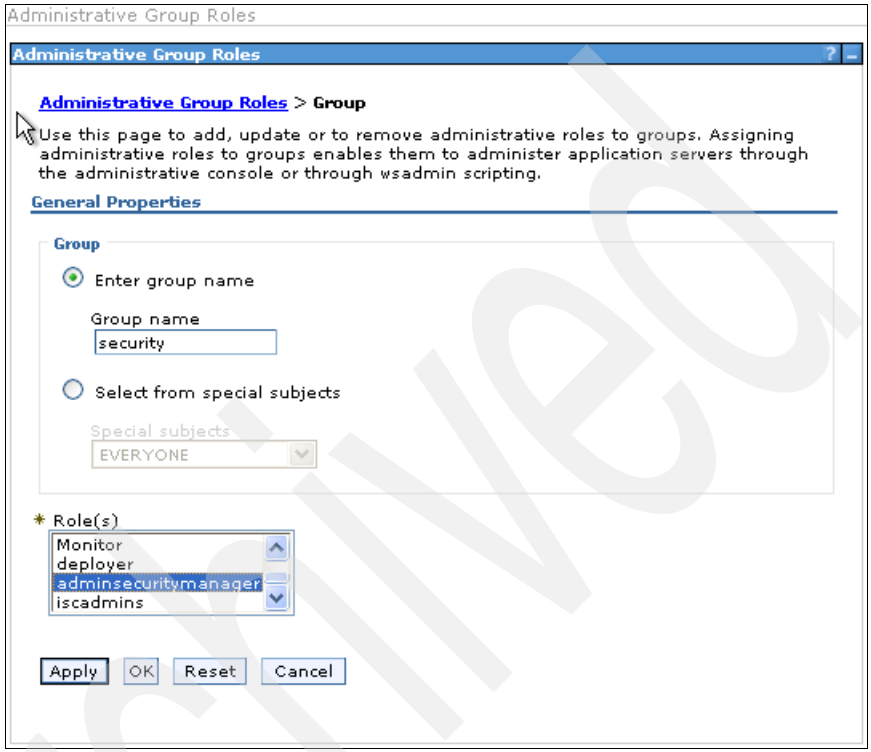


Figure 10-16 Adding groups with roles

7. Repeat for the other two groups in Table 10-1 on page 256. Your administrative groups roles window should look like Figure 10-17.

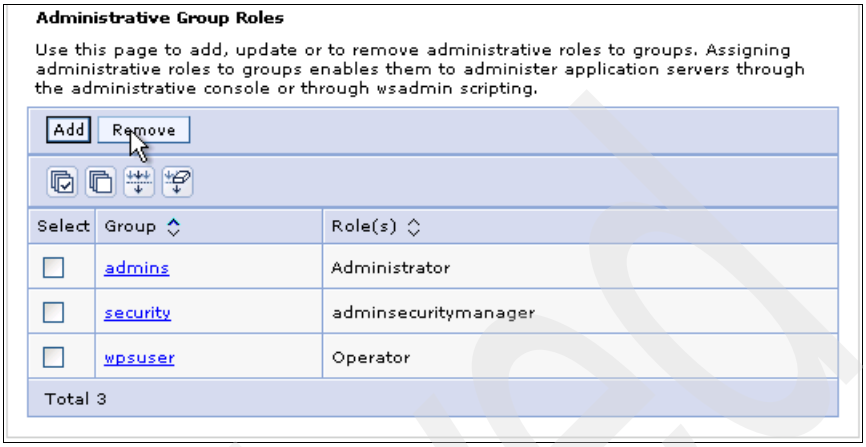


Figure 10-17 Completed group roles window

10.2.6 Mapping groups to the business integration containers and supporting applications

Each container or supporting application is a J2EE application, and the applications are controlled using roles. These roles are defined in 4.3, “Security for a WebSphere Process Server solution” on page 73. To use the security roles to user/groups mapping feature in the Integrated Solutions Console, you must be in a group that is assigned the administrative role of either configurator or administrator.

1. Log into the Integrated Solutions Console.
2. Click **Applications** → **Enterprise Applications** → **BPEContainer <deployment_environment>.AppTarget**.
3. Under Detail Properties click **Security role to user/group mapping**.
4. Check the **select** box for a role. In this example you will use BPEAPIUser.
5. Click **Look up users** or **Look up groups**. In this example, use **Look up groups**.
6. Enter either ***** or a specific value to the search box.
7. Click **Search**.
8. Highlight **wpsuser** or your user or group.

9. Click **>>** to add the group to the role. The window is shown in Figure 10-18.

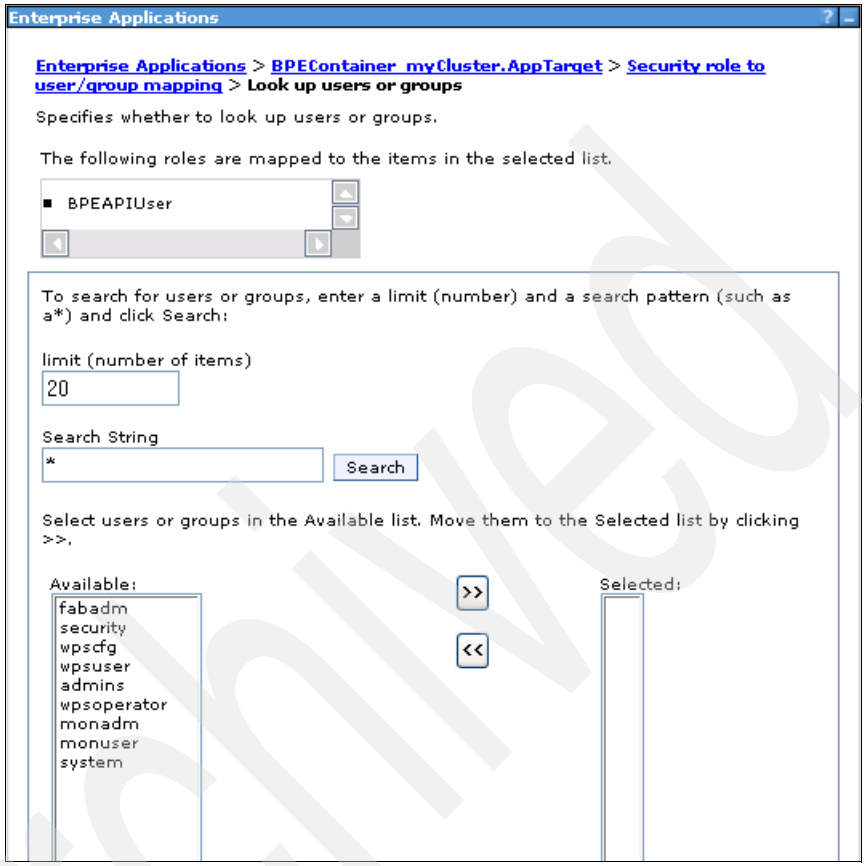


Figure 10-18 Lists of groups to add to a role

10. Click **OK**.

Once this is complete, the group wpsuser has been added to the BPEAPIUser role. These are the same steps that you would use for each set of roles below.

Important: In the rest of this section you will see the permissions that are set by default. You will want to build a table based on the needs of your business and the contents of your registries. See Table 10-2 on page 260 for an example based on our sample LDAP configuration.

Table 10-2 Table to use to secure your management applications

Application	Security role	Administrator assigned users or groups
BPEContainer_<deployment Environment.cluster>	BPESystemAdministrator	admins
	BPESystemMonitor	wpsusers
	BPEAPIUser	wpsusers
	WebClientUser	wpsusers
	JMSAPIUser	jmsapi
TaskContainer_<deployment Environment.cluster>	TaskSystemAdministrator	admins
	TaskSystemMonitor	wpsusers
	TaskAPIUser	wpsusers
	EscalationUser	escalation
BPCExplorer_<deployment Environment.cluster>	WebClientUser	wpsusers
BusinessSpaceManager	administrator	admins
BusinessRulesManager_ <deploymentEnvironment. cluster>	BusinessRuleUsers	wpsusers
Event Service	eventAdministrator	admins
	eventConsumer	wpsusers, admins
	eventUpdater	wpsusers, admins
	eventCreator	admins
	catalogAdministrator	admins
	catalogReader	admins
wpsFEMgr 6.2.0	WBIOperator	wpsusers,admins

Business Process Choreographer

BPC consists of multiple J2EE Enterprise Application Archives (EAR) files:

- Two container EARs
- Three management application EARs

Mapping roles for the container EARs

Figure 10-19 shows the roles available for the BPEContainer EAR. To map roles for the BPEContainer EAR:

1. Click **Applications** → **Enterprise Applications** → **BPEContainer_<deployment_environment>.AppTarget**.
2. Under Default Properties click **Security role to user/group mapping**.
3. Refer to Table 10-2 on page 260 to map your groups to the roles.

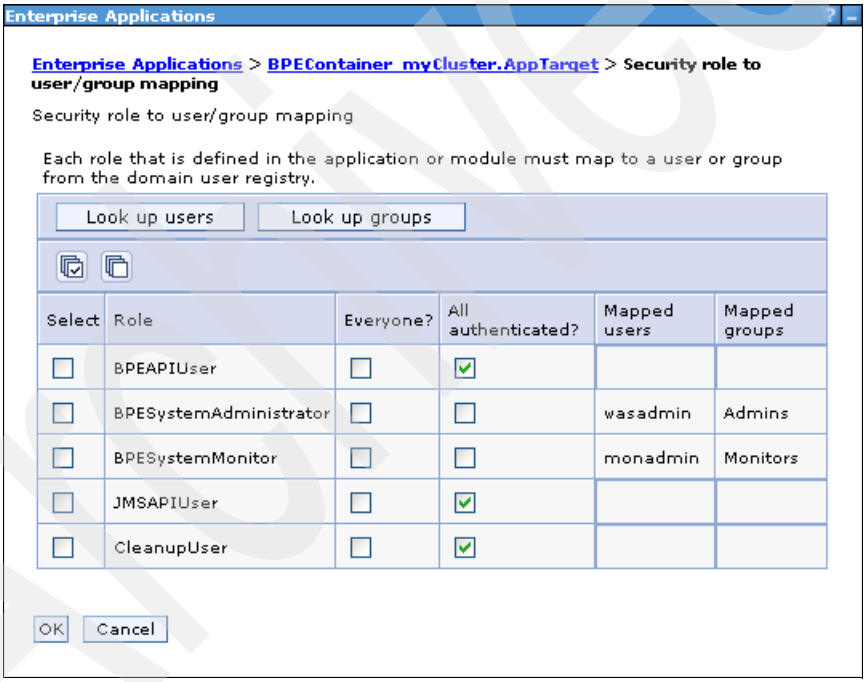


Figure 10-19 BPC Users list

Note: You will see that there are users and groups already populated. This occurred during the initial configuration through the wizard.

To map roles for the TaskContainer EAR:

1. Click **Applications** → **Enterprise Applications** → **TaskContainer_<deployment_environment>.AppTarget**.
2. Under Default Properties click **Security role to user/group mapping**.
3. Refer to Table 10-2 on page 260 to map your groups to the roles.

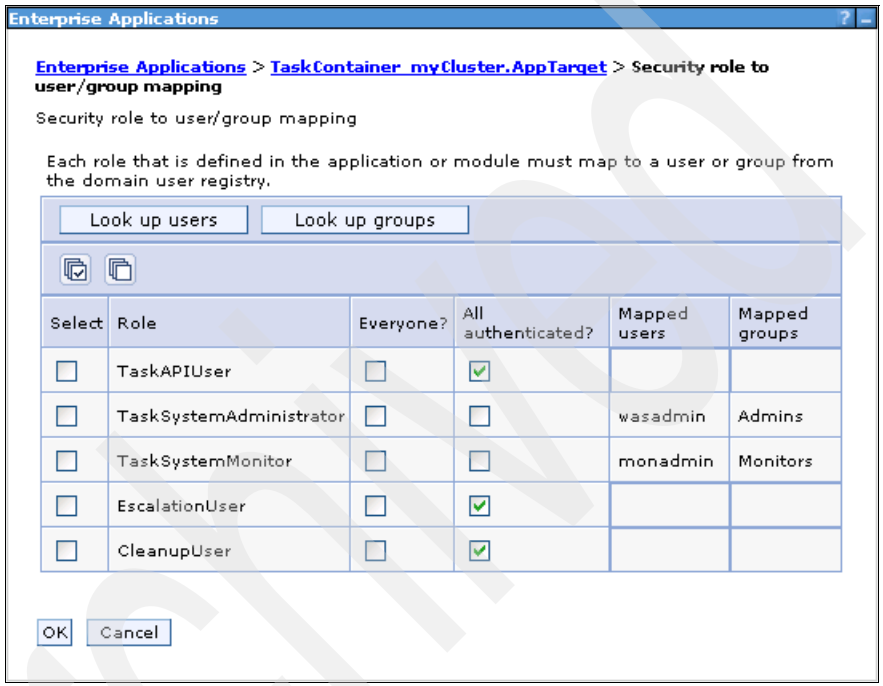


Figure 10-20 Task Container roles list

Note: You can also navigate to the mapping window for the container application by clicking **Applications** → **SCA Modules** → **TaskContainer <deployment_environment>.AppTarget** → **Security role to user/group mapping**.

Management applications

The BPC Container has three management applications that will give you the flexibility to grant certain groups permissions to certain but not all functions. These applications are as follows:

- Business Space Manager

  The Business Space Manager is where you manage your business spaces. This includes creating and deleting, adding pages, and setting who can view and edit privileges.

  The Business Space Manager displays the business spaces that you own and the spaces for which you are a viewer or an editor. The Business Space Manager consists of a toolbar, an area that lists the spaces and pages, and an area that displays information about the selected space or page. Based on the Access Control Lists that you set for your own Business Space, there is internal authorization checking.

  Figure 10-21 shows a single role of administrator. This role has access to administer every single business space in the system, not just its own. Users without this role can only administer their own Business Space.

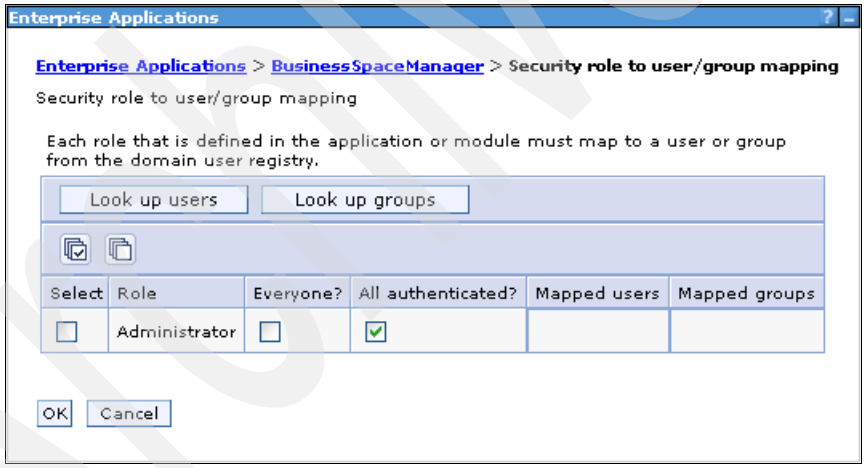


Figure 10-21 Business Space manager roles list

To do this:

a. Click **Applications** → **Enterprise Applications** → **BusinessSpaceManager_<deployment_environment>.AppTarget**.

b. Under Default Properties click **Security role to user/group mapping**.

c. Refer to Table 10-2 on page 260 to map your groups to the roles.

- BPC Explorer

  BPC Explorer is a Web application that implements a generic Web user interface for interacting with business processes and human tasks.

  Figure 10-22 shows a single WebClientUser role. Users assigned this role can view and act on only those tasks that have been assigned to them.

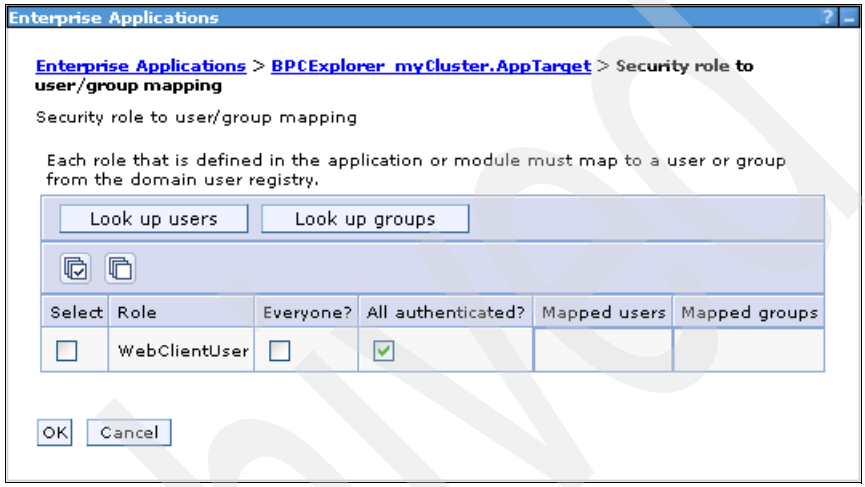


Figure 10-22 BPC Explorer roles list

Note: The Business Process Choreographer Observer application has been merged into the Business Process Choreographer Explorer reporting function. Information about this function can be found at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp?topic=/com.ibm.websphere.bpc.620.doc/doc/bpc/c2observer.html

- Business Rules Manager

  The Business Rules Manager is the main WebSphere Process Server tool that a business analyst uses for rule authoring.

Figure 10-23 shows a single BusinessRulesUser role. Users assigned this role will be able to update business rules. The NoOne role is required if Tivoli Access Manager is part of the deployment, as it requires a role for indicating who absolutely cannot access the application. This role does not need to map to anything valid.

Perform the following steps to check the `web.xml` file of the Business Rule Manager Web application to verify what resources these roles are securing:

1. Click **Applications** → **Enterprise Applications** → **BusinessRulesManager_<Support_Cluster_Name>**.
2. Under Default Properties click **Security role to user/group mapping**.
3. Refer to Table 10-2 on page 260 to map your groups to the roles.

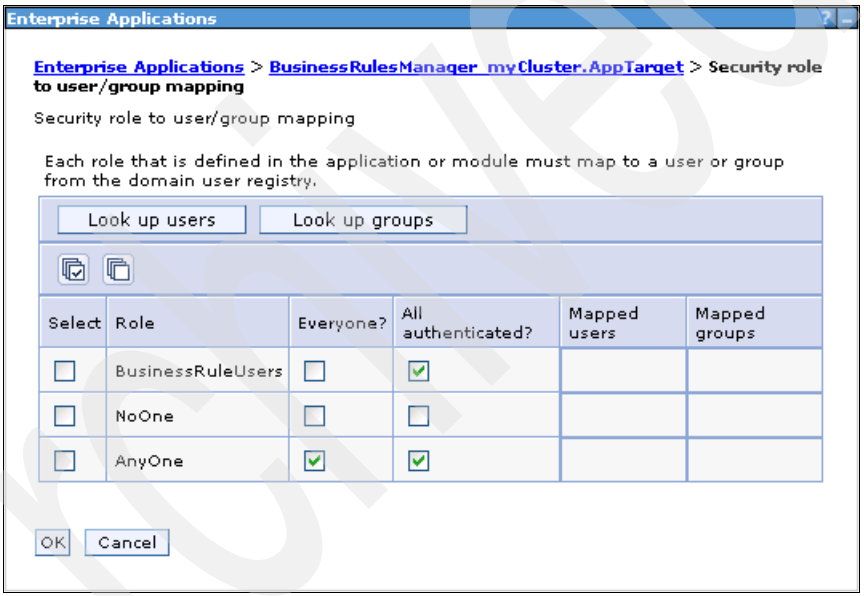


Figure 10-23 Business Rules Manager roles list

For more information about how this manager works, refer to the Information Center article *How the business rules manager works*, available at the following Web page:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp?topic=/com.ibm.websphere.wps.620.doc/doc/cbre_busiru_overview.html

Service Component Architecture

SCA has one management application, Failed Event Manager. Use the Failed Event Manager to find and manage WebSphere Process Server failed events on all servers in a cell. The interface enables you to view and edit the data for a failed event, resubmit a failed event, or delete a failed event.

Figure 10-24 shows a single WBIOperator role. Users assigned this role will be able to use the application.

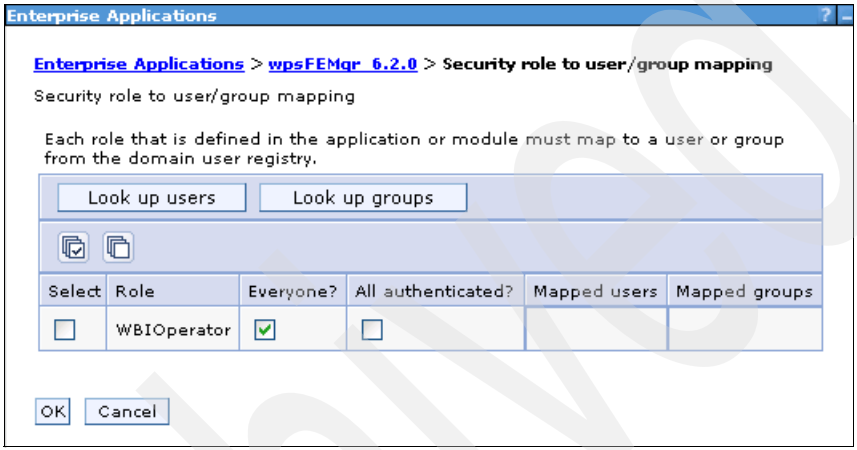


Figure 10-24 Failed Event Manager

Follow these steps:

1. Click **Applications** → **Enterprise Applications** → **wpsFEMgr_6.1.2**.
2. Under Default Properties click **Security role to user/group mapping**.
3. Refer to Table 10-2 on page 260 to map your groups to the roles.

Common Event Infrastructure

The event service is the conduit between event sources and event consumers. The event service receives events submitted to emitters by event sources. It stores events in a persistent data store, and then distributes them asynchronously to subscribed event consumers. In addition, the event service supports synchronous queries of historical events from the persistent store.

Figure 10-25 shows multiple roles. Users assigned these roles will gain access to interfaces referenced in 4.3.4, “Access control for Common Event Infrastructure container” on page 81.

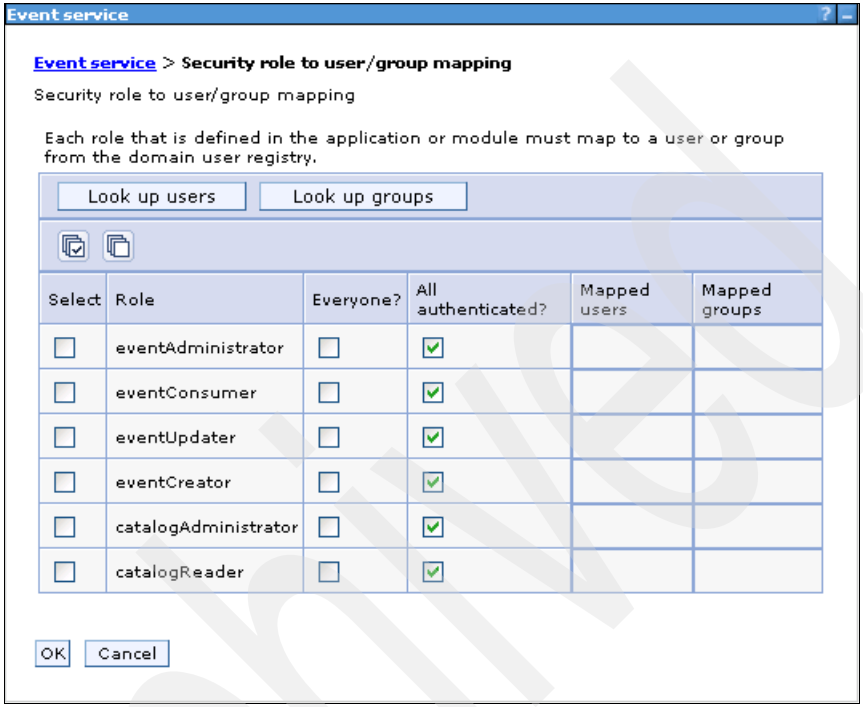


Figure 10-25 Common Infrastructure roles list

Follow these steps:

1. Click **Service Integration** → **Common Event Infrastructure** → **Event Service**.
2. Under Default Properties click **Security role to user/group mapping**.
3. Refer to Table 10-2 on page 260 to map your groups to the roles.

10.2.7 Administrative action for securing components

One of the roles of the configurator and administrator is to secure the SCA modules based on application-defined roles. Have the development team provide you with a table with the roles that they have defined and a description of their purpose. When you install the modules, you will need to assign users and groups to these roles. The procedure is shown in the Information Center article *Deploying (installing) secure applications*, available at the following Web page:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.websphere.wps.620.doc/doc/tsec_deploying.html

If the application is communicating with an external resource through Web services, the communication links will most likely be encrypted. The administrator may need to work with external vendors to properly secure this transportation channel chain.

11

Advanced production topologies

This chapter discusses ways to extend the Remote Messaging and Remote Support topology to provide additional processing capability. The following topics are discussed:

- Adding cluster members to the clusters created during Remote Messaging and Remote Support deployment environment generation
- Adding clusters to the Remote Messaging and Remote Support topology
- Distributing messaging engines across cluster members in the Remote Messaging and Remote Support topology

While this chapter only discusses extending the Remote Messaging and Remote Support topology, the principles discussed here apply to the other supported topologies as well.

11.1 Reasons for extending a topology

There are three reasons for extending a topology from a Remote Messaging Remote Support (ND7) topology:

- 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.

    Note: Isolation is a double-edge sword. Consider an extended topology with multiple application targets, each with its own BPCDB.

    With this topology, when querying the BPCDB (for example, to claim a human task), the result would only show tasks from within a single BPCDB. 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, increases failover for messaging engines, and increases startup times for both messaging engines and application targets. Similarly, if new versions of applications are being deployed, the number of destinations will increase further.

Deploying numerous applications, especially those that use large objects, would cause 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.

- Simplicity

  Growing topologies by adding new application and messaging clusters is a repeatable and proven process and provides a broad level of functionality.

  It is possible to grow a topology by adding single application clusters without adding any new messaging clusters. However, this still leads to the possibility of overburdening a messaging engine cluster. It is therefore advisable to maintain a single 1-1 ratio of messaging engines to application clusters.

11.2 Extending Remote Messaging and Remote Support topology

In production and performance testing environments, you may discover the need to add additional processing capability to one or more of the clusters included in the Remote Messaging and Remote Support topology. Extending the Remote Messaging and Remote Support topology can be done in any of the following ways:

- Extending the application target cluster by adding nodes and cluster members
- Extending the application target and messaging infrastructure capabilities by adding both an additional application cluster and an additional messaging cluster
- Extending the messaging cluster's capabilities by distributing messaging engines across cluster members
- Extending the support cluster by adding nodes and cluster members

This is not an exhaustive list of methods for extending the Remote Messaging and Remote Support topology. It represents the more common methods of adding processing capability that are discussed in this chapter.

To implement the extended topologies discussed in this chapter, a Remote Messaging and Remote Support topology was created. This topology is represented in Figure 11-1.

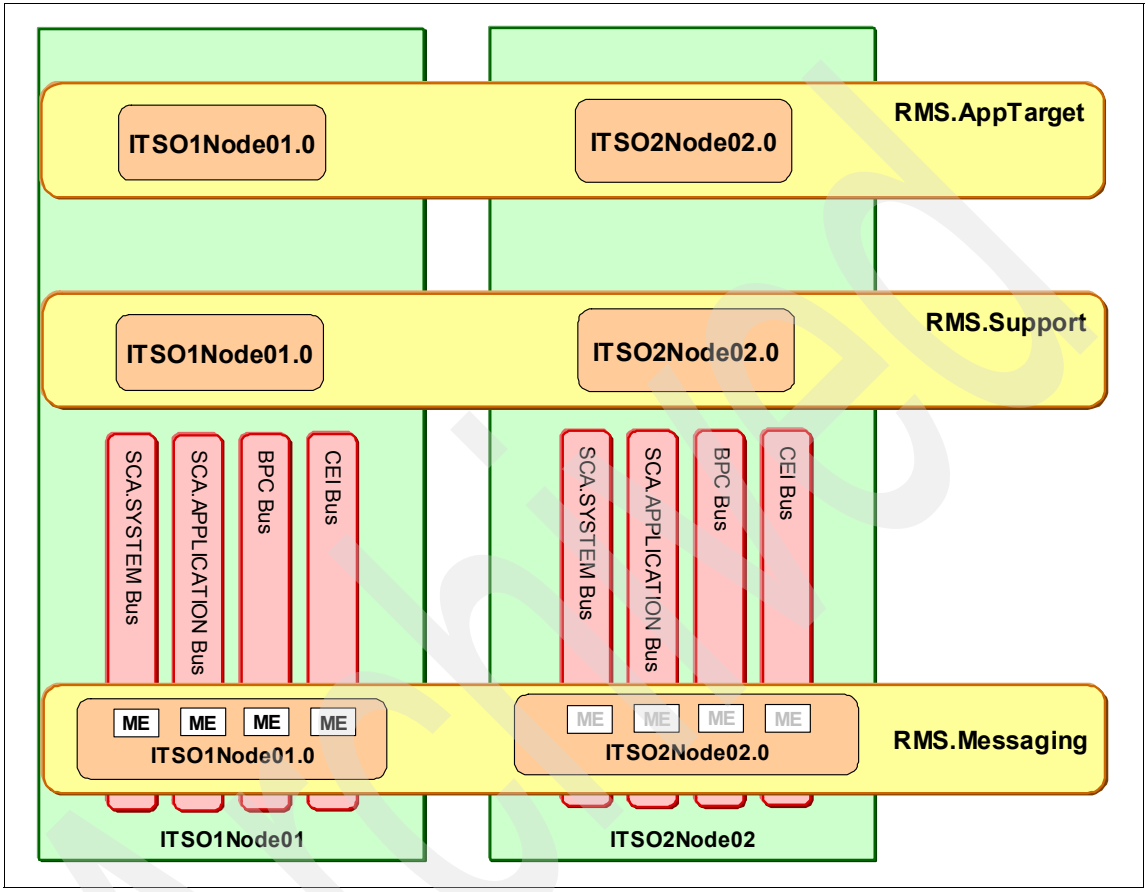


Figure 11-1 Topology created for this chapter

This topology was created using the Deployment Environments window in the Integrated Solutions Console. This environment contains two machines:

- ITSO1
- ITSO2

ITSO1 contains Node01. ITSO2 contains Node02.

Node01 houses the following members:

- A member of the RMS.AppTarget cluster (RMS.AppTarget.ITSONode01.0).
- A member of the RMS.Support cluster (RMS.Support.ITSO1Node01.0).
- A member of the RMS.Messaging cluster (RMS.Messaging.ITSO1Node01.0). All four messaging engines are started on this server.

Node02 houses the following members:

- A member of the RMS.AppTarget cluster (RMS.AppTarget.ITSONode02.0).
- A member of the RMS.Support cluster (RMS.Support.ITSO1Node02.0).
- A member of the RMS.Messaging cluster (RMS.Messaging.ITSO1Node02.0). All four messaging engines are joined on this server.

Later in this chapter this environment is expanded by adding nodes, servers, and clusters.

11.3 Adding nodes and cluster members

If you need additional processing capability for the application target cluster in the Remote Messaging and Remote Support topology, you have the option of adding nodes and cluster members. In typical representations of the Remote Messaging and Remote Support topology, there are usually two to three nodes, each with one cluster member. This is not mandatory. Should you need to add additional cluster members to the application target cluster or the support cluster, it is possible to do so.

There are a number of reasons why you may wish to add additional cluster members to the application target cluster, such as:

- To increase application processing capability
- To create additional capacity for migration or application updates
- To provide adequate failover capability

You may add additional nodes and server instances to the application cluster, or you may add additional servers to an existing node in the application cluster (if the hardware is capable of supporting the additional Java Virtual Machines (JVMs) and the resulting additional memory required). If you add additional cluster members to existing hardware, be sure that you will not overwhelm the system's capabilities. Adding cluster members to the Remote Messaging and Remote Support topology is represented in Figure 11-2.

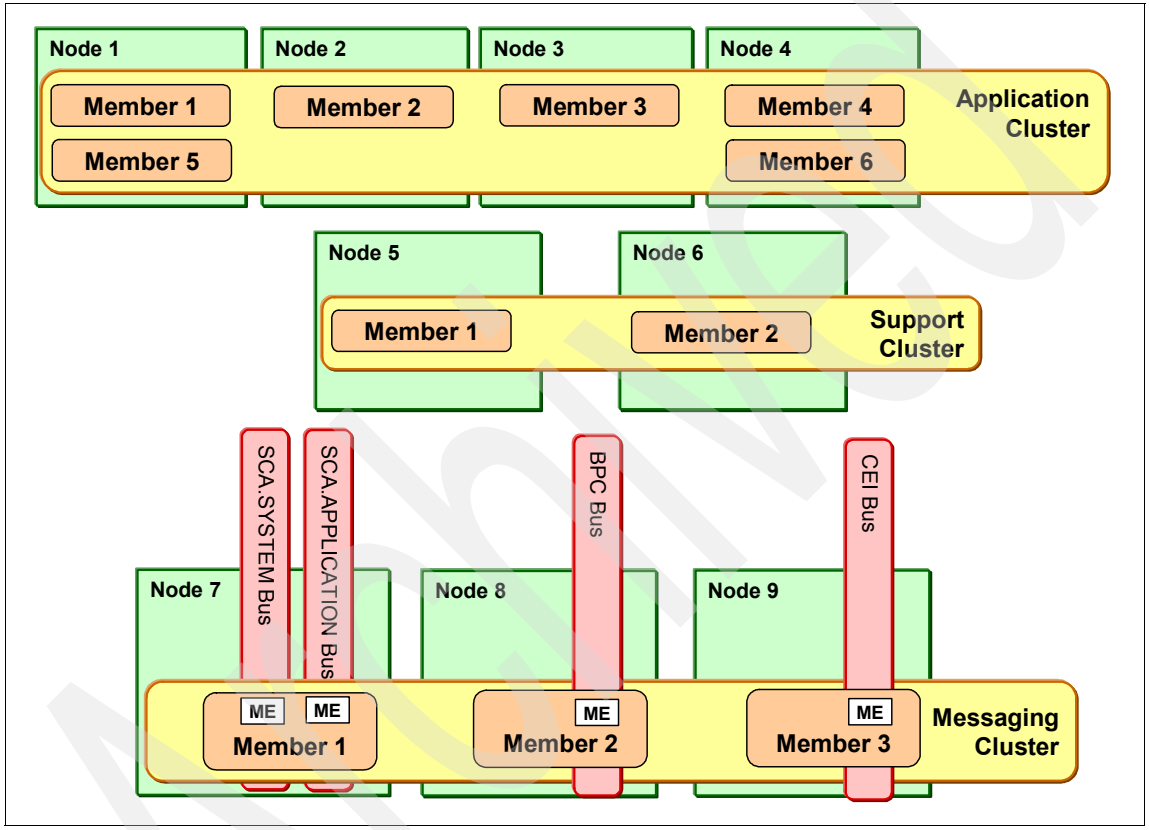


Figure 11-2 Adding servers and nodes to the application cluster

The topology represented here began as a Remote Messaging and Remote Support topology with two nodes. Each cluster had a single cluster member (server instance) on each node. To add additional processing capability, two additional nodes were added to the application target cluster, and an additional node was added to the messaging cluster (this configuration is discussed in 11.5, "Distributing messaging workload using policies" on page 318). In addition, two of the nodes in the application target clusters were extended by adding cluster members to them.

Note the following aspects of this topology:

- Adding nodes or more than three cluster members to the messaging cluster does not add additional processing capability. The preferred topology is to have one active instance of each messaging engine (with the remaining engines on standby). If you distribute the engines across the cluster members, the SCA.SYSTEM and SYS.APPLICATION engines are on one server (these should always be kept together), the Business Process Choreographer (BPC) engine is on a second server, and the Common Event Infrastructure (CEI) engine is on a third server. Adding a fourth, fifth, or sixth cluster member does not increase messaging capacity.
- Partitioning destinations in the messaging cluster (by creating multiple active instances of each messaging engine) can give you additional workload management capabilities. However, this configuration should be avoided due to issues with potential message loss, lack of event ordering, and so forth. These issues are discussed in Chapter 3, “Business Process Management production topologies” on page 37.
- You are not required to have the same number of cluster members in each cluster. If you find that you need additional application processing, but that the support cluster performs to your satisfaction, you can add additional application cluster members without adding support cluster members.
- Because adding messaging cluster members does not provide additional processing capability, this cluster has a maximum of three members. This is true even if the application cluster has several more members than the messaging cluster.

In our lab, the Remote Messaging and Remote Support topology created in the Integrated Solutions Console was extended to provide additional messaging and application processing capability. A third node (ITSO2Node03) was added to the topology on machine ITSO2. This node houses a member of the RMS.AppTarget cluster and a member of the RMS.Messaging cluster. Adding the third cluster member to the application target cluster provides additional application processing, while the third member of the messaging cluster allows you to split the messaging engines across cluster members. The resulting topology is represented in Figure 11-3.

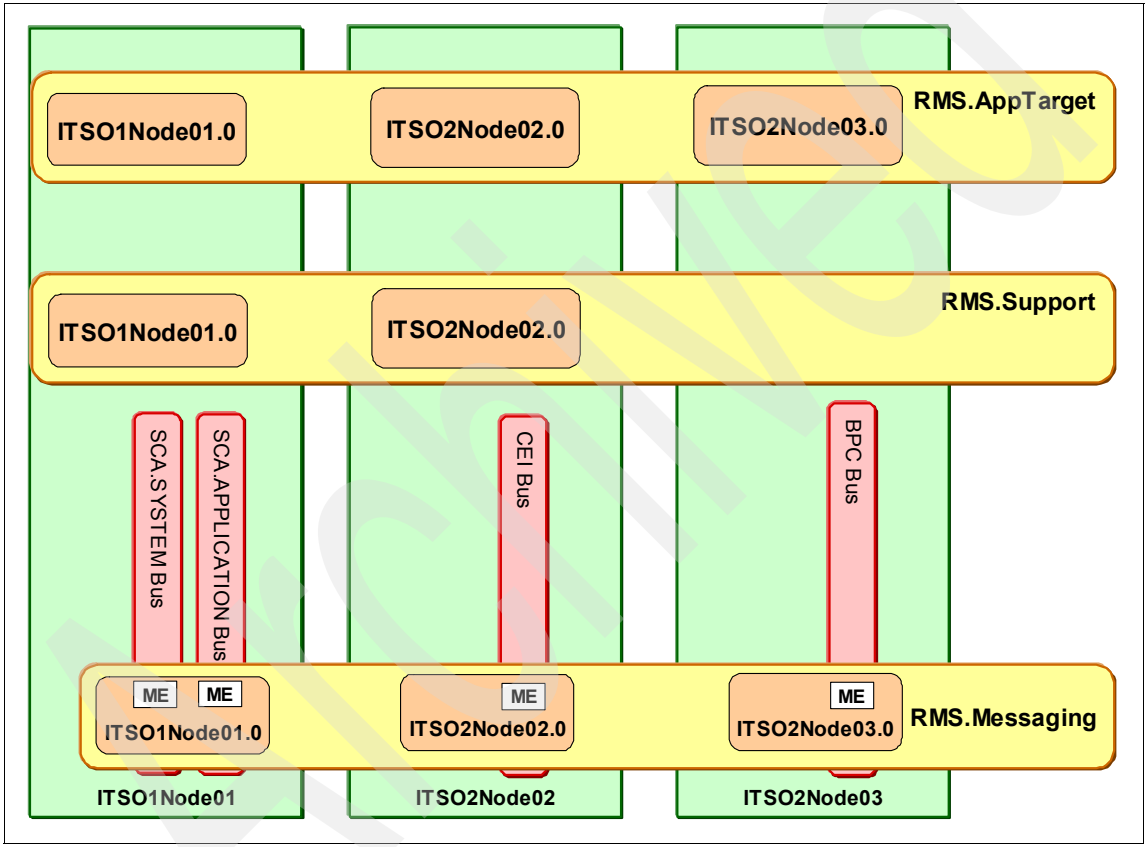


Figure 11-3 Remote Messaging and Remote Support topology with additional nodes and cluster members

Distributing the messaging engines across the messaging cluster members is discussed in 11.5, “Distributing messaging workload using policies” on page 318.

To add an additional node and additional cluster members to the Remote Messaging and Remote Support topology, perform the following steps.

> **Note:** These steps (and the naming conventions used) assume that you created your initial Remote Messaging and Remote Support environment using the template-driven deployment process.

1. Use the profile management tool to create a new custom profile on a new machine that you are including in the topology (if you need additional hardware) or on an existing machine that is already part of the topology (if you just need additional processing capability and the existing hardware supports it). In our lab, a new custom profile was created on machine ITSO2.
2. Federate the node.
3. In the Integrated Solutions Console, expand **Servers** and click the **Deployment Environments** link (Figure 11-4).

Figure 11-4 Deployment environments link

4. Click the link for your deployment environment in the Deployment Environment Name column. In our lab, the name of the deployment environment was RMS (Figure 11-5).

Select	Deployment Environment Name	Status	Pattern	Description
☐	RMS	✖	Remote Messaging and Remote Support	

Figure 11-5 Deployment Environment Name link

5. In the Additional Properties section, click the **Deployment Topology** link (Figure 11-6).

Figure 11-6 Deployment Topology link

6. Click the **Existing node** radio button and select the newly federated node from the drop-down list. In our lab, the newly federated node was named ITSO2Node03 (Figure 11-7).

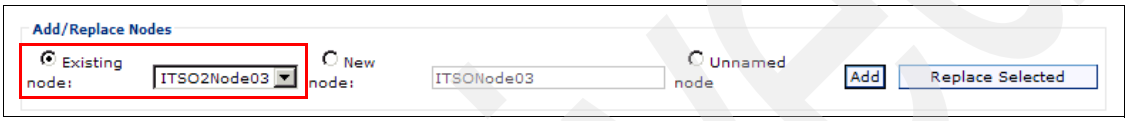


Figure 11-7 Add a new node to the deployment environment

7. Click the **Add** button to add the node to the topology.
8. Enter `1` in the Application Deployment Target column, enter `1` in the Messaging Infrastructure column, and enter `0` in the Supporting Infrastructure column. This creates a single server instance in the RMS.Messaging cluster and a single server instance in the RMS.AppTarget cluster on the new node (Figure 11-8).

Select	Node	Status	Capability	Version	Base Version	Host	Application Deployment Target	Messaging Infrastructure	Supporting Infrastructure
☐	ITSO1Node01		WPS	6.1.2.0	6.1.0.17	ITSO1	1	1	1
☐	ITSO2Node02		WPS	6.1.2.0	6.1.0.17	ITSO2	1	1	1
☐	ITSO2Node03		WPS	6.1.2.0	6.1.0.17	ITSO2	1	0	1

Figure 11-8 Specify the number of servers

9. Click **OK**.
10. Click the **Save** link to save your changes to the master configuration.
11. If you have automatic synchronization enabled, when the synchronization process is complete, you should see the message `The configuration synchronization complete for cell`. Click **OK**. Otherwise, manually synchronize the changes.

12. In the Integrated Solutions Console, expand **Servers** and click the **Clusters** link.

13. Click the link for the application target cluster. In our lab, this value was RMS.AppTarget (Figure 11-9).

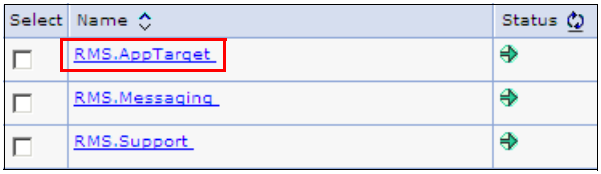


Figure 11-9 Application target cluster link

14. In the Additional properties section, click the **Cluster members** link (Figure 11-10).

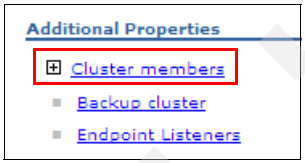


Figure 11-10 Cluster members link

15. In the cluster members table, click the check box for the newly added cluster member and click **Start** (Figure 11-11).

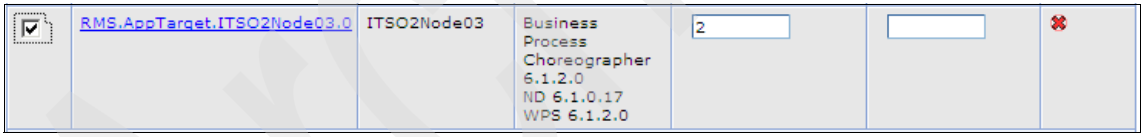


Figure 11-11 Cluster members table

16. Verify that the new cluster member starts without error by checking `SystemOut.log` for exceptions.

17. Repeat the previous steps to start the new messaging cluster member, RMS.Messaging.ITSO2Node03.0.

18. (Optional) Verify the structure of the topology by examining the cluster topology diagram:

   a. In the Integrated Solutions Console, expand **Servers** and click the **Cluster topology** link.

   b. Expand **RMS.AppTarget**, expand **Nodes**, expand each of the individual nodes listed, and expand **Cluster members**. You should see three nodes, each with a single server instance, as depicted in Figure 11-12.

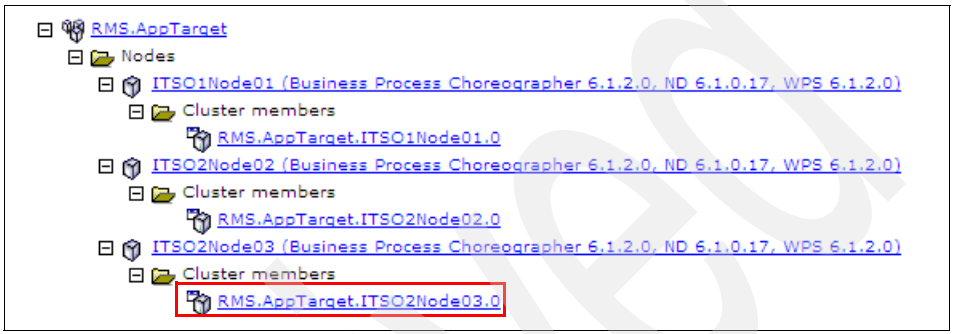


Figure 11-12 Cell topology with additional application cluster member

   c. Expand **RMS.Messaging**, expand **Nodes**, expand each of the individual nodes listed, and expand **Cluster members**. You should see three nodes, each with a single server instance, as depicted in Figure 11-13.

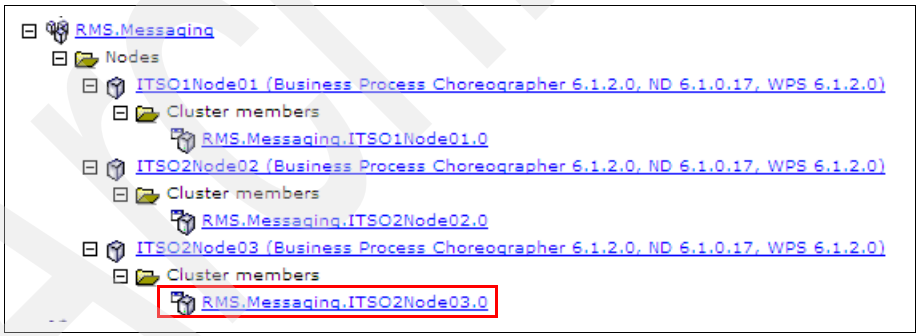


Figure 11-13 Cell topology with additional messaging cluster member

11.4 Adding WebSphere Process Server application clusters

There are many reasons why you might add an additional WebSphere Process Server application cluster to your cell topology, such as:

- The need to isolate application functionality for your organization's business units due to regulatory or governance requirements (You may deploy the applications for human resources to one cluster, while the applications for the accounting group are deployed to a separate cluster.)
- The need to isolate applications because they have unique runtime requirements (heavy asynchronous traffic versus primarily synchronous traffic)
- The need to isolate different application versions
- The need to provide additional application processing capability (Creating a new application cluster instead of adding members to the existing cluster adds administrative complexity to your topology.)
- The need to work around application bottlenecks

If you decide to deploy applications to two separate application clusters, keep in mind that there are several possible limitations to this topology, including:

- The names of the Service Component Architecture (SCA) components within your applications must be unique in the cell. Therefore, if you deploy the same applications to both clusters, you must rename the SCA modules in the second application instance so that they are unique. This creates additional administrative and development requirements that you would not otherwise have.
- The additional application target cluster will require a new set of database tables for the BPC. Creating a new schema or database to house the data for the additional BPC will add performance tuning and administrative requirements to the topology.
- If you create the additional application cluster on existing hardware, you must consider how the additional JVMs will affect the available memory and how this will affect your existing performance-tuning scenario.
- If you deploy the same application to both clusters, in addition to unique module names, you must also have unique context roots for your Web modules.

When you deploy applications to two separate application target clusters without modifying the messaging cluster, all the destinations for the applications in both application clusters are deployed to the messaging cluster, just as they would be

if all the applications were in a single application cluster. If you determine that the messaging cluster is a bottleneck, you may increase the messaging capacity in one of two ways:

- Distribute the messaging engines across servers in the messaging cluster. See 11.5, “Distributing messaging workload using policies” on page 318.
- Create an additional messaging cluster. See 11.4.2, “Adding application cluster and additional messaging cluster” on page 283.

11.4.1 Adding an additional application cluster

This topology contains a second WebSphere Process Server cluster. It is leveraging the existing messaging cluster. Adding a second WebSphere Process Server application cluster to your cell topology consists of the following steps:

1. Create a second BPC database. The existing WPRCSDB will be shared by both application clusters. There can only be one WPRCSDB per cell.
2. Create the second application cluster and add the required cluster members.
3. Configure SCA support for the cluster. This configures the cluster to use the remote messaging cluster that is a member of the SCA.SYSTEM and SCA.APPLICATION buses.
4. Deploy the BPC in the cluster. This configures the cluster to use the remote messaging cluster that is a member of the BPC bus.
5. Configure the Common Event Infrastructure destination for the application cluster. Because the Common Event Infrastructure (CEI) destination used by the support cluster is configured at the cell level, the additional application cluster leverages the existing support cluster for CEI event propagation.

The addition of a second application cluster to the Remote Messaging and Remote Support topology created for this book is represented in Figure 11-14.

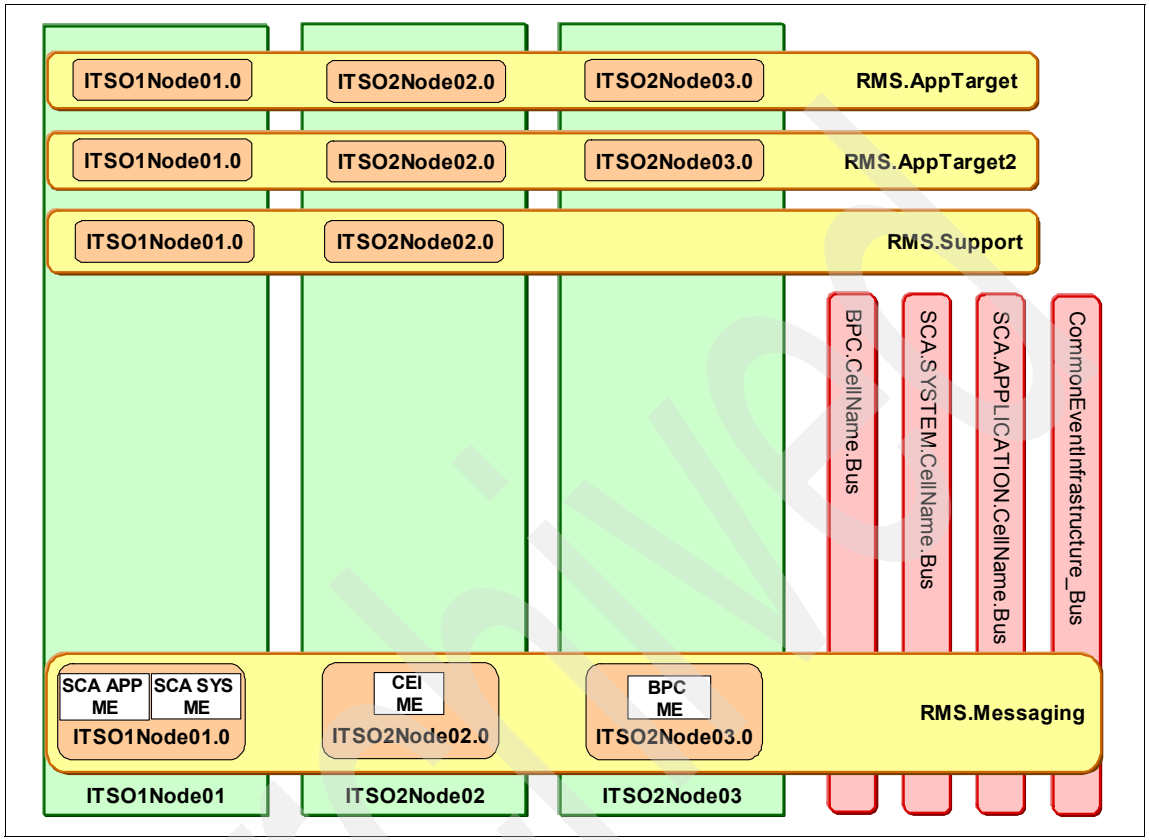


Figure 11-14 Gold topology with second application cluster

11.4.2 Adding application cluster and additional messaging cluster

If you need additional capacity for your applications and for the messaging infrastructure, you can add an additional messaging cluster and an additional application cluster. Currently, implementing a single application target cluster and two messaging clusters is not supported. You cannot split the destinations for a single set of applications across two messaging clusters.

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. Currently, creating duplicate buses is not supported. 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 wish to add additional nodes and cluster members to the support cluster to prevent bottlenecks.

The default behavior of the messaging infrastructure when there are two application clusters and two messaging clusters is depicted in Figure 11-15.

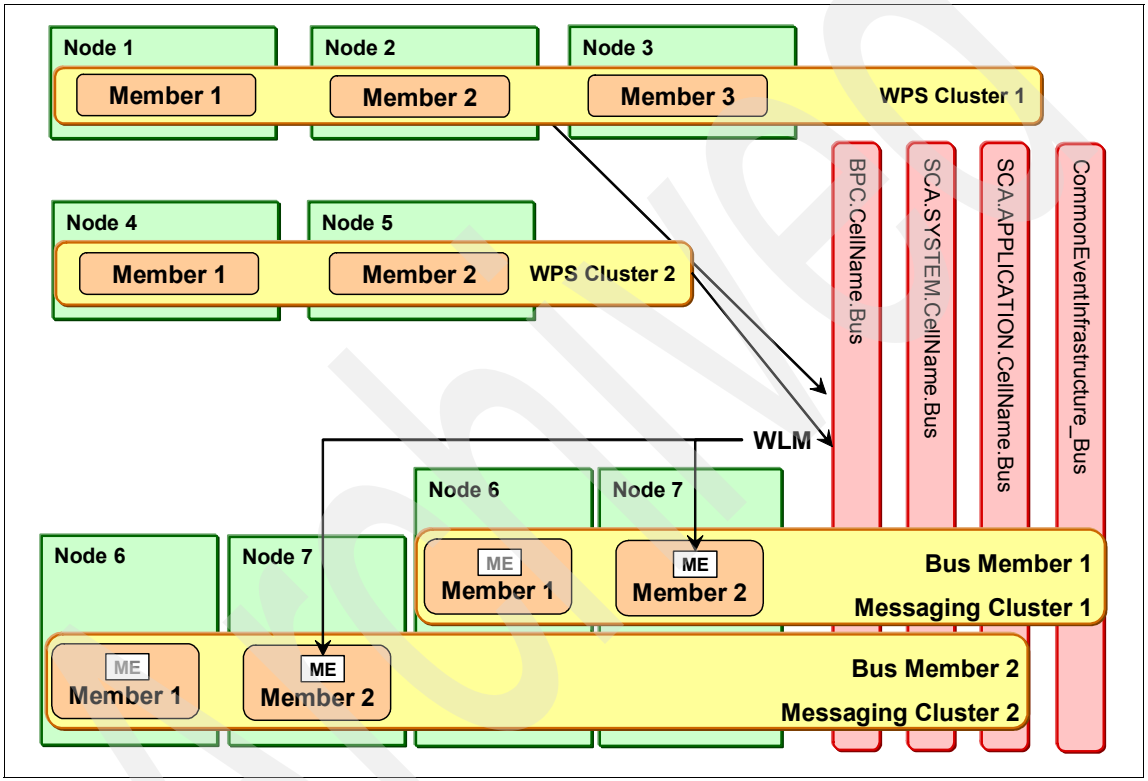


Figure 11-15 Messaging behavior with two application clusters and two messaging clusters

When you have two application clusters and two messaging clusters, as you deploy applications to the application target clusters, you identify which messaging cluster will hold the destinations used by the applications for asynchronous communication. When an application needs access to a destination, it connects to the appropriate bus and then to the messaging engine in the cluster where the destinations are housed.

At run time, the workload manager controls to which applicable messaging engine the application is ultimately directed. This decision is based on several factors such as proximity. The resulting connection may or may not be to the desired messaging engine. If your application connects to the applicable messaging engine in cluster one, but the destinations exist in messaging cluster two, this can create a pass-through condition. The messages produced by the application are sent to the messaging engine in cluster one, which then forwards the messages to the applicable messaging engine in cluster two.

If the messaging engine in the cluster that houses the application's destinations is down, a condition called store-and-forward results. In Figure 11-16 on page 286, an application in WPS Cluster 2 needs access to a destination that was created in messaging cluster 2. However, the applicable messaging engine in that cluster is down. Because the application cannot place the message in the appropriate destination, the workload manager will connect the application to the messaging engine in messaging cluster 1. Because the message is intended for a destination in the other messaging cluster, the messaging engine on messaging cluster 1 will create a temporary queue for the message and will deliver it to the messaging engine on messaging cluster 2 when the messaging engine becomes available.

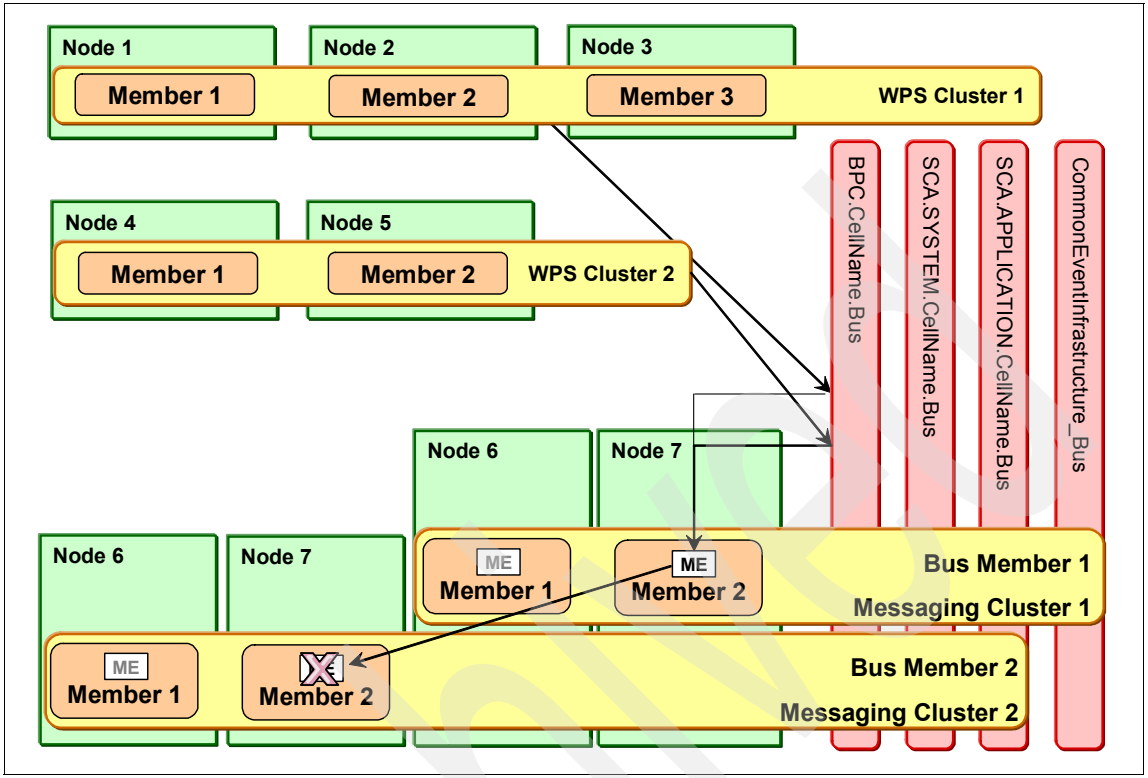


Figure 11-16 Messaging store-and-forward

The desirability of store-and-forward depends on your environment. If you want message delivery to continue even if one of the messaging clusters is down, you may find this option acceptable. However, if the performance hit incurred by the messaging engine on the surviving messaging cluster is unacceptable, you may find this option untenable.

In order to avoid pass-through and store-and-forward, and to isolate each application cluster to a dedicated messaging cluster, you must configure target significance for each JMS connection factory and activation specification in your environment. If you have a large number of destinations, this can be a time-consuming task. Consider this effort carefully before deciding on a dual-cluster topology with messaging isolation. The behavior of the messaging infrastructure after applying target significance to the activation specifications and connection factories is depicted in Figure 11-17.

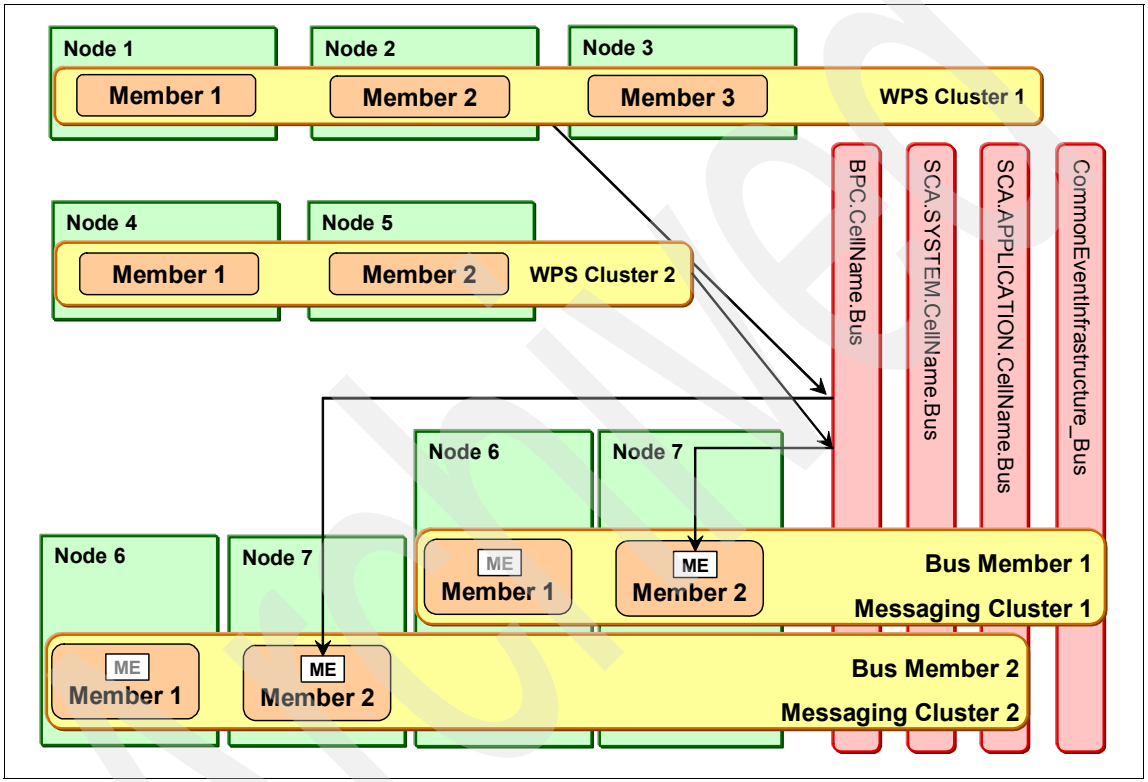


Figure 11-17 Messaging behavior with target significance

In Figure 11-17, each application target cluster uses a specific set of messaging engines in a specific messaging cluster. For SCA messaging and BPC messaging, WPS cluster 1 will use the messaging engines on messaging cluster 1. For SCA and BPC messaging, WPS cluster 2 will use messaging cluster 2. The target significance property for the activation specifications and connection factories determines this behavior. Because the CEI resources are defined at the cell level, both application target clusters will use messaging cluster 1 for CEI message traffic.

Creating additional application target cluster and messaging clusters

Creating an additional application target cluster and an additional messaging cluster consists of the following steps:

1. Add an additional application target cluster:
   a. Create a second BPC database.
   b. Create the second application cluster and add the number of required cluster members.
   c. Configure SCA support for the application cluster.
   d. Deploy the BPC in the application cluster.
   e. Configure the CEI destination for the application cluster.
2. Add an additional messaging cluster:
   a. Create an additional messaging engine database.
   b. Create the second messaging cluster.
   c. Configure SCA support for the additional messaging cluster.
   d. Configure target significance for the connection factories and activation specifications for both application clusters.

The addition of a second application cluster and a second messaging cluster to the Remote Messaging and Remote Support topology created for this publication is represented in Figure 11-18.

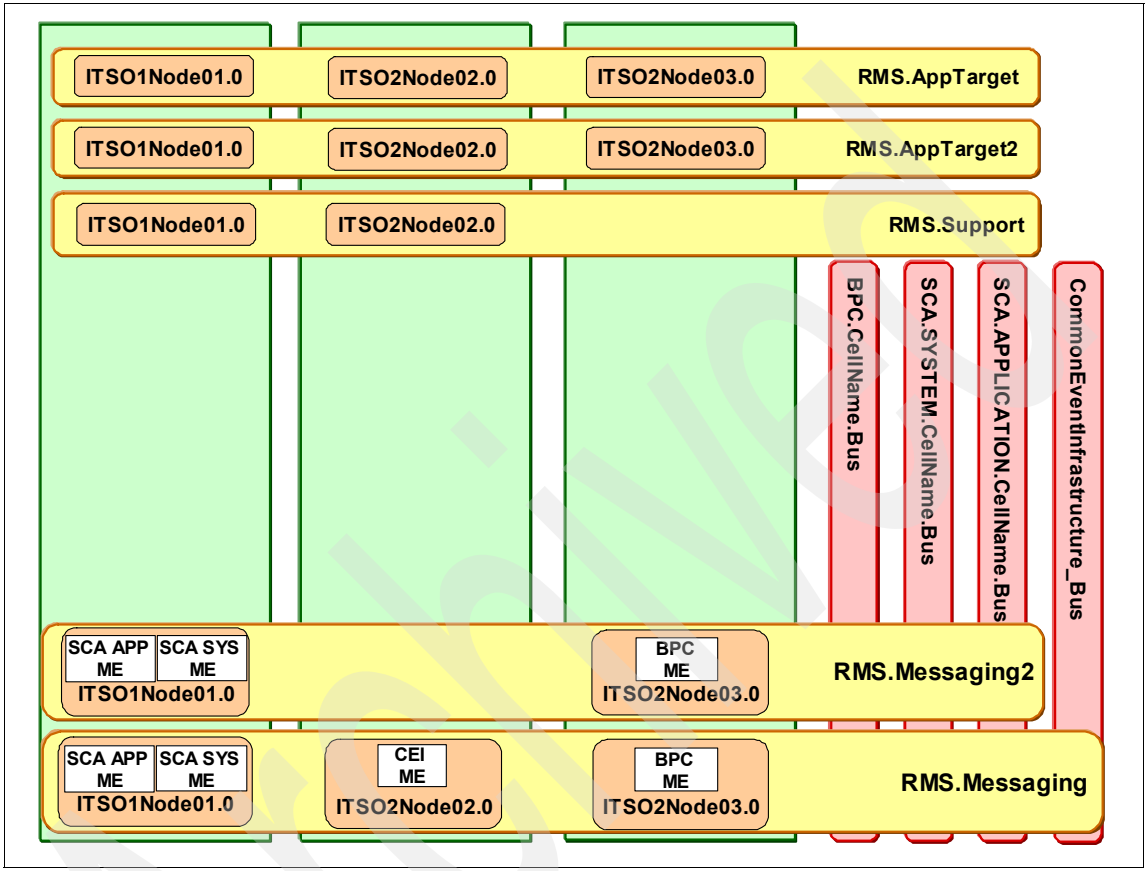


Figure 11-18 Gold topology with two application clusters and messaging clusters

Adding an additional application target cluster

The following steps were used to implement this extended topology. Each step is described in the sections that follow.

1. Creating a new database for the second application cluster
2. Adding application cluster and additional messaging cluster
3. Configuring SCA support
4. Configuring the Common Event Infrastructure destination

Creating a new database for the second application cluster

When you create the new database for the second BPC in the cell, you can create a new schema in the existing BPEDB or you can create a new database. The option that you choose ultimately depends upon the database system that you are using and the performance tuning requirements. For example, in DB2, using unique databases improves performance.

To create a new DB2 database for the second application cluster:

1. Issue the command to create the database:
   a. In a DB2 command window, enter the following command to create the database:

      ```
      db2 CREATE DB BPEDB2 USING CODESET UTF-8 TERRITORY en-us.
      ```

   b. When the database is created you should see the message `The CREATE DATABASE command completed successfully`. Leave the DB2 command window open.

      > **Note:** You can create a script to generate BPEDB2, or you can customize the existing database creation script with the following command:
      >
      > `<Websphere_Process_Server_Root>\dbscripts\ProcessChoreographer\DB2\createDatabase.sql`

2. Issue the command to generate the database schema:
   a. Edit the `<Root>\dbscripts\ProcessChoreographer\DB2\createSchema.sql` file.
   b. Replace all instances of @SCHEMA@ with the name of your schema. In our lab, we used the name BPEBE02.
   c. Save and close the file.
   d. Edit the `<Websphere_Process_Server_Root>\dbscripts\ProcessChoreographer\DB2\createTablespace.sql` file.

e. Replace all instances of @location@ with the name of the DB2 node directory. In our lab, we used the directory C:\DB2\NODE0000 (Figure 11-19).

> **Note:** Adjust the directory paths for your operating system. Because the our lab machine was on Windows®, the \ character was used.

```
-----------------------------------
-- Create 4 K page tablespaces --
-----------------------------------

CREATE TABLESPACE AUDITLOG
  MANAGED BY SYSTEM
  USING( 'C:\DB2\NODE0000\AUDITLOG' );

CREATE TABLESPACE COMP
  MANAGED BY SYSTEM
  USING( 'C:\DB2\NODE0000\COMP' );

CREATE TABLESPACE INSTANCE
  MANAGED BY SYSTEM
  USING( 'C:\DB2\NODE0000\INSTANCE' );
```

Figure 11-19 DB2 node directory in createTablespace.sql

f. Save and close the file.

g. Move both files to the remote database machine or to the machine with the DB2 client installed. You may wish to put the files in `\IBM\SQLLIB\bin` for ease of use with the command window.

h. In the DB2 command window, enter the following command to connect to the BPEDB2 database:

```
db2 CONNECT TO BPEDB2 USER <Username> USING <Password>
```

In our lab, we used the following command:

```
db2 CONNECT TO BPEDB2 USER db2admin USING web1sphere
```

You should receive database connection information similar to the following:

```
Database server        = DB2/NT 9.1.3
SQL authorization ID   = DB2ADMIN
Local database alias   = BPEDB2
```

i. Issue the following command to generate the tablespaces:

```
db2 -tf createTablespace.sql
```

You should see several of the following messages:

```
The SQL command completed successfully.
```

j. Issue the following command to generate the schema:

```
db2 -tf createSchema.sql
```

You should see several of the following messages:

```
The SQL command completed successfully.
```

Once the new database is created, the next step is to create the second application target cluster.

Creating a second application cluster

To create a second application cluster:

1. In the Integrated Solutions Console navigation pane, expand **Servers** and click the **Clusters** link (Figure 11-20).

Figure 11-20 Clusters link

2. In the server clusters window, click **New**. This opens the Step 1: Enter basic cluster information window.

3. Enter `RMS.AppTarget2` in the Cluster name text box (Figure 11-21).

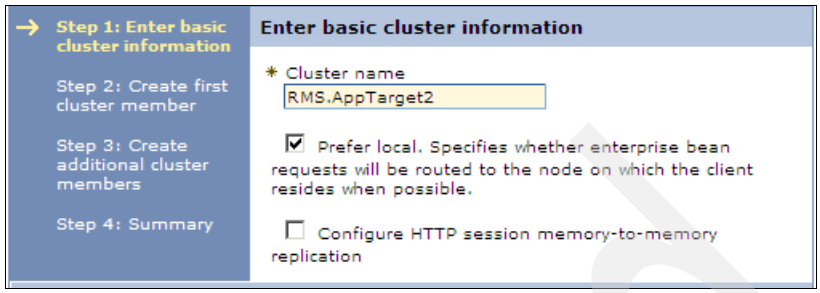


Figure 11-21 New application cluster name

Click **Next**. The Step 2: Create first cluster member window opens.

4. Perform the following steps to create the first cluster member (Figure 11-22):
   a. Enter the name of the first cluster member in the Member name text box. In our lab, we used the name RMS.AppTarget2.ITSO1Node01.0 to keep the naming conventions in line with the names generated during template-driven topology creation.
   b. Choose the appropriate node from the Select node drop-down list. In our lab, this value was ITSO1Node01.
   c. In the Select basis for first cluster member section, click the **Create the member using an application server template** radio button and choose **defaultProcessServer** from the drop-down list.

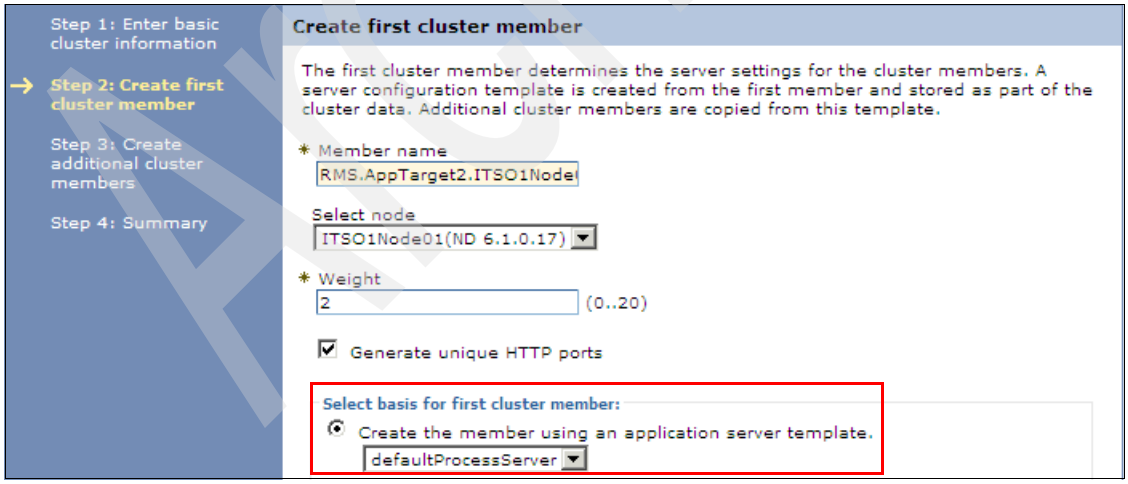


Figure 11-22 Create first cluster member

> **Important:** Because this cluster uses WebSphere Process Server functionality, you must choose the defaultProcessServer template. The default template creates a WebSphere Application Server instance.

5. Click **Next**. The Step 3: Create additional cluster members window opens.
6. Perform the following steps to create additional cluster members (Figure 11-23):
   a. Enter the name of the additional cluster member in the Member name text box. In our lab, we used the name RMS.AppTarget2.ITSO2Node02.0 to keep the naming conventions in line with the names generated during template-driven topology creation.
   b. Choose the appropriate node from the Select node drop-down list. In our lab, this value was ITSO2Node02.

Figure 11-23 Add additional cluster members

7. Click the **Add Member** button. The name of the additional cluster member should appear in the table.

8. Repeat the previous steps to add any additional cluster members. In our lab, a total of three servers were created on three separate nodes (Figure 11-24).

Member name	Nodes	Version	Weight
RMS.AppTarget2.ITSO1Node01.0	ITSO1Node01	Business Process Choreographer 6.1.2.0 ND 6.1.0.17 WPS 6.1.2.0	2
RMS.AppTarget2.ITSO2Node02.0	ITSO2Node02	Business Process Choreographer 6.1.2.0 ND 6.1.0.17 WPS 6.1.2.0	2
RMS.AppTarget2.ITSO2Node03.0	ITSO2Node03	Business Process Choreographer 6.1.2.0 ND 6.1.0.17 WPS 6.1.2.0	2

Figure 11-24 Cluster members table

Note about adding members: Adding members during cluster creation is not required. You may create the cluster with one member first and verify the cluster configuration before adding more members. For demonstration purposes, all cluster members were added at the same time in this example.

Click **Next**. The Step 4: Summary window opens.

9. Review your options and click **Finish**.
10. Click the **Save** link at the top of the window (Figure 11-25).

Messages

Changes have been made to your local configuration.
You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

An option to synchronize the configuration across multiple nodes after saving can be enabled in Preferences.

The server may need to be restarted for these changes to take effect.

Figure 11-25 Save changes to the master configuration

11. If you have automatic synchronization enabled, when the synchronization process is complete, you should see the following message:

    `The configuration synchronization complete for cell.`

    Click **OK**. Otherwise, manually synchronize the changes when you are done creating policies.

You should be returned to the Server clusters window, and you should see your newly created cluster. Do not start the cluster at this time. You will configure the remaining options before you start the cluster.

Configuring SCA support

Once you have generated a new database and created the new application cluster, the next step is to configure SCA support for the second application cluster using the Integrated Solutions Console.

To configure SCA support for the second application cluster:

1. In the Integrated Solutions Console, expand **Servers** and click the **Clusters** link.
2. In the server clusters window, click the **RMS.AppTarget2** link (Figure 11-26).

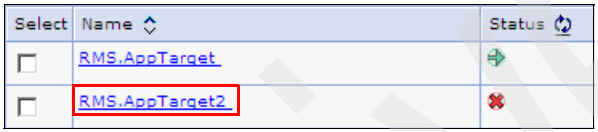


Figure 11-26 RMS.AppTarget2 link

3. In the Business Integration section, click the **Service Component Architecture** link (Figure 11-27).

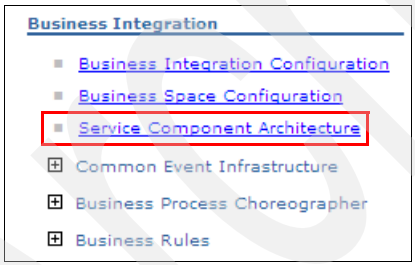


Figure 11-27 Service Component Architecture link

4. In the General Properties section, click the **Support the Service Component Architecture components** check box (Figure 11-28).

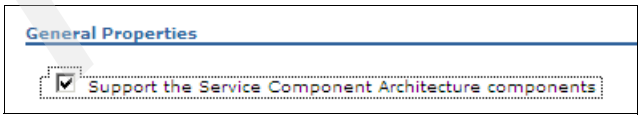


Figure 11-28 SCA support option

5. In the Bus Member Location section, click the **Remote** radio button and select the **RMS.Messaging** cluster from the drop-down list (Figure 11-29).

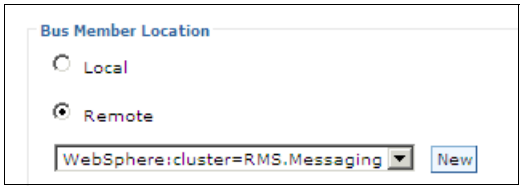


Figure 11-29 Remote bus member

When you select the remote messaging cluster, the System bus member and Application bus member sections should be populated with the same information used to enable SCA in the first application target cluster (Figure 11-30).

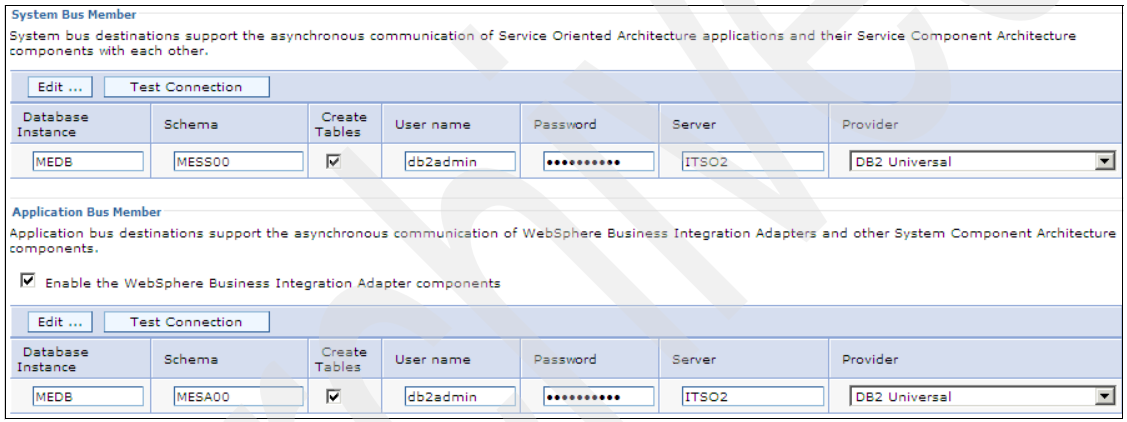


Figure 11-30 SCA.SYSTEM and SCA.APPLICATION bus properties

Click **OK**.

6. Click **Save**.
7. If you have automatic synchronization enabled, you should see the following message when the synchronization process is complete:

   `The configuration synchronization complete for cell.`

   Click **OK**. Otherwise, manually synchronize the changes when you are done creating policies.

   You should be returned to the Server clusters window, and you should see your newly created cluster. Do not start the cluster at this time. You will configure the remaining options before you start the cluster.

In addition to using the Integrated Solutions Console to configure SCA support for the messaging infrastructure, you can use the following wsadmin commands:

- **`configSCAAsyncForCluster`**

  Use this command to configure the messaging cluster to support asynchronous Service Component Architecture (SCA) applications using the SCA.SYSTEM bus.

  > **Note:** Full syntax for configSCAAsyncForCluster can be found within the 6.2 Infocenter:
  >
  > http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp?topic=/com.ibm.websphere.wps.620.doc/doc/rref_cfgscaasyncclustercmd.html

- **`configSCAJMSForCluster`**

  Use this command to configure the messaging cluster to support asynchronous communication for SCA applications using the SCA.APPLICATION bus.

  > **Note:** Full syntax for configSCAJMSForCluster can be found within the 6.2 Infocenter:
  >
  > http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp?topic=/com.ibm.websphere.wps.620.doc/doc/rref_cfgscajmsclustercmd.html

Configuring the Business Process Choreographer

After creating the additional application target cluster, you must deploy the BPC. Deploying the BPC installs the human task container and the business process container. Deploying these containers allows you to run applications containing human tasks and business processes. You can deploy the BPC using the Integrated Solutions Console or the bpeconfig.jacl script.

To configure the BPC in the second application target cluster:

1. In the Integrated Solutions Console, expand **Servers** and click the **Clusters** link.
2. In the server clusters window, click the **RMS.AppTarget2** link (Figure 11-31).

Figure 11-31 RMS.AppTarget2 link

3. In the Container Settings section, expand **Business Process Choreographer Container Settings** and click the **Business Process Choreographer Containers** link (Figure 11-32).

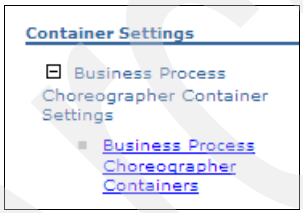


Figure 11-32 Business Process Choreographer Containers link

4. In the Data Source section, perform the following steps (Figure 11-33):
   a. In the Database Instance text box, enter the name of the database that you configured previously (`BPEDB2`).
   b. In the Schema Name text box, enter the name of the schema that you used to populate the `createSchema.sql` file (`BPEBE02`).
   c. Clear the Create Tables check box. (The tables were created when you ran the createSchema.sql script.)
   d. Populate the User Name text box with the DB2 account that you entered during template-driven deployment (in our lab this value was db2admin).
   e. Populate the Password text box with the value that you entered during template-driven deployment (in our lab this value was web1sphere).
   f. Populate the Server text box with the host name of the DB2 server (in our lab this value was ITSO2).
   g. Set the Provider to the appropriate value (in our lab the value was DB2 Universal).

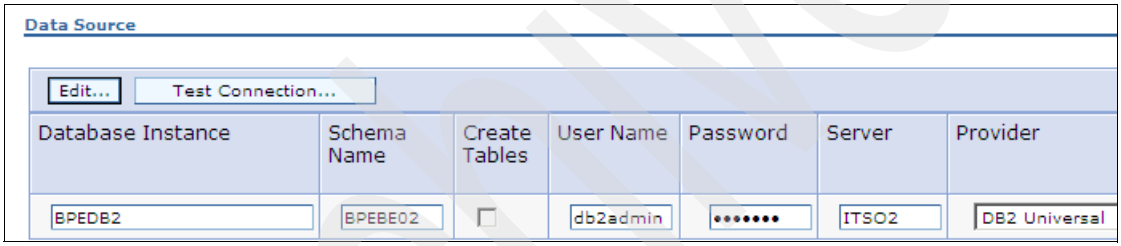


Figure 11-33 BPC data source properties

5. In the Human Task Manager Mail Session section, perform the following steps (Figure 11-34):
   a. Select the **Enable e-mail service** check box if your applications use e-mail notifications in human task escalations.
   b. In the Mail transport host text box, enter the name of the host used for the default Java mail session (in our lab this value was ITSO1).
   c. Populate the Mail transport user text box with the name of the messaging authentication account.
   d. In the Mail transport password text box, enter the password for the authentication account (in our lab this value was web1sphere).
   e. In the Business Process Choreographer Explorer URL text box, enter the URL for the Explorer client (in our lab this value was http://ITSO2/bpc).

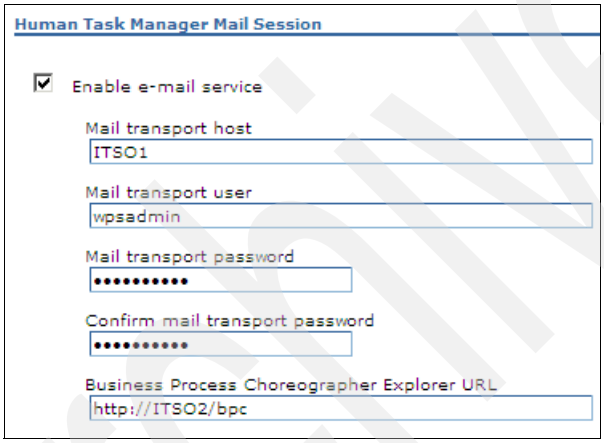


Figure 11-34 Human Task Manager Mail Session properties

6. In the Security section, enter the passwords for the authentication users that you configured during template-driven deployment. (For more information about these accounts see 4.3, “Security for a WebSphere Process Server solution” on page 73.) This is shown in Figure 11-35.

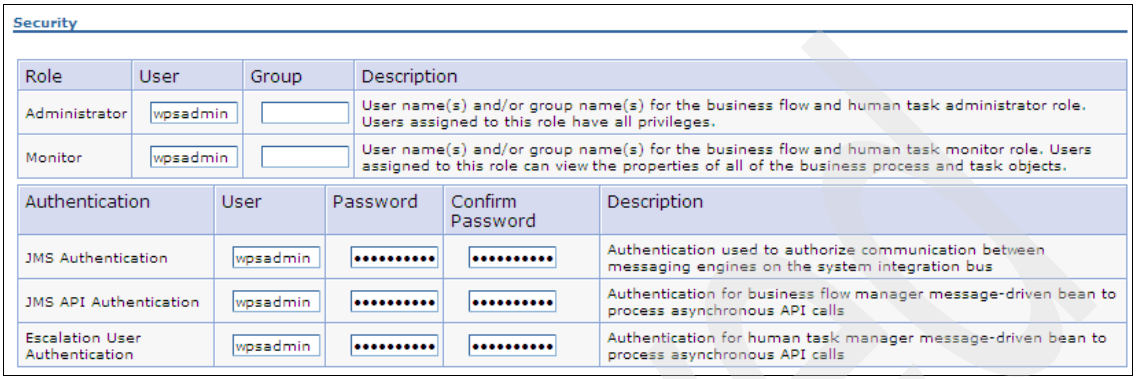


Figure 11-35 Security properties

7. In the State Observers section, if your applications produce CEI events, select the **Business Flow Manager** or the **Human Task Manager** check box (or both if you wish to monitor human tasks and business processes) for the Common Event Infrastructure Logging row. This is shown in Figure 11-36.

   Audit logging can also be used to persist business-relevant data for auditing purposes. Because of the performance implications, you should carefully consider using both audit logging and CEI logging.

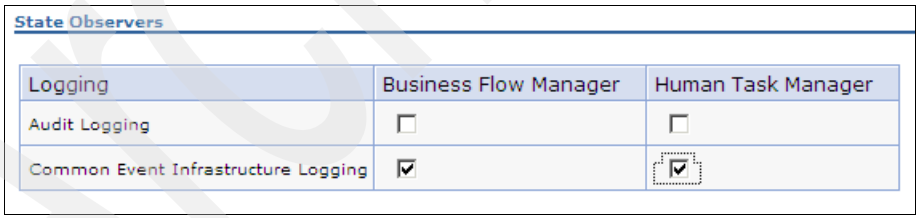


Figure 11-36 State Observers properties

8. In the SCA Bindings section, verify the following information (Figure 11-37):
   - Context root for the Business Flow Manager: /BFMIF_RMS.AppTarget2
   - Context root for the Human Task Manager: /HTMIF_RMS.AppTarget2

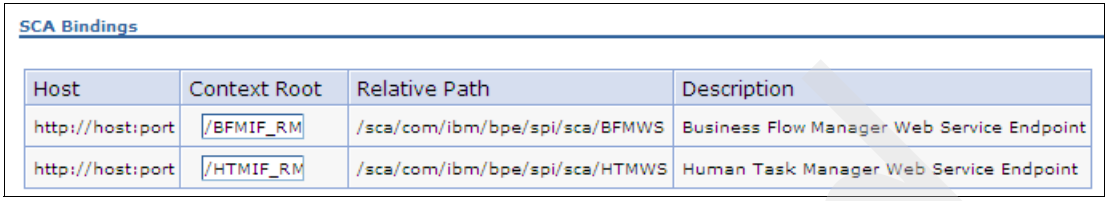


Figure 11-37 SCA Bindings properties

9. In the Bus section, clear the Use the default configuration check box. If you leave this option selected, the BPC bus and the BPC messaging engine are created in the RMS.AppCluster2 cluster. See Figure 11-38.

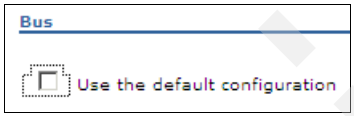


Figure 11-38 Bus properties

10. In the Bus Member Location section, click the **Remote** radio button and select the remote messaging cluster, **RMS.Messaging**, from the drop-down list (Figure 11-39). When the remote messaging cluster is selected, you should see the database properties for the BPC bus that were configured during template-driven deployment. Because both application target clusters will be using the same remote messaging cluster, these properties are the same for both application clusters.

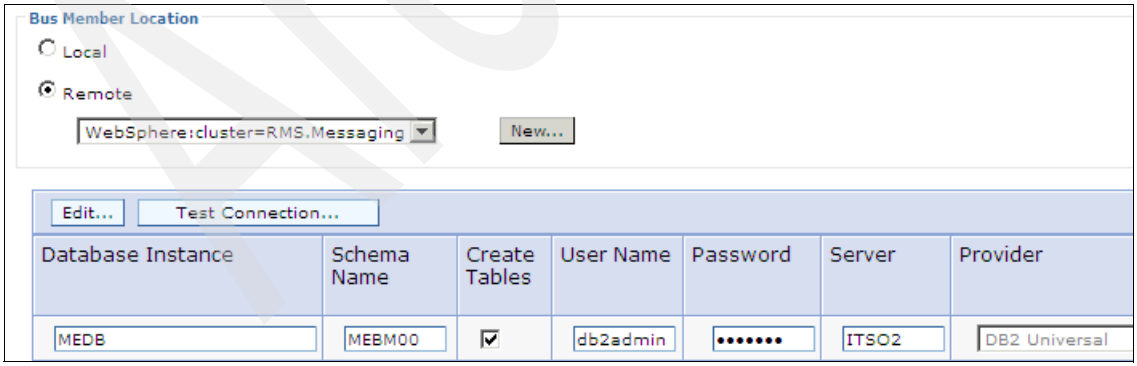


Figure 11-39 Bus Member Location properties

11. Click **OK** to deploy the BPC. When the configuration is complete, you should see the following messages in the console:

```
Application BPEContainer_RMS.AppTarget2 installed successfully.
Application TaskContainer_RMS.AppTarget2 installed successfully.
Application HTM_PredefinedTasks_V612_RMS.AppTarget2 installed
successfully.
Application HTM_PredefinedTasksMsg_V612_RMS.AppTarget2 installed
successfully.
```

12. If the configuration completes successfully, click **Save Changes**.

13. If you have automatic synchronization enabled, when the synchronization process is complete, you should see the following message:

```
The configuration synchronization complete for cell.
```

Click **OK**. Otherwise, manually synchronize the changes.

Configuring the Common Event Infrastructure destination

The final step in configuring the additional application target cluster is to configure the CEI destination. Because you configured a remote support cluster to handle CEI events for the cell, a cell-scoped CEI destination was created. The additional application cluster that you created should also use this destination.

To configure the CEI destination for the additional application cluster:

1. In the Integrated Solutions Console, expand **Servers** and click the **Clusters** link.
2. In the server clusters window, click the **RMS.AppTarget2** link (Figure 11-40).

Figure 11-40 RMS.AppTarget2 link

3. In the Business Integration section, expand **Common Event Infrastructure** and click the **Common Event Infrastructure Destination** link (Figure 11-41).

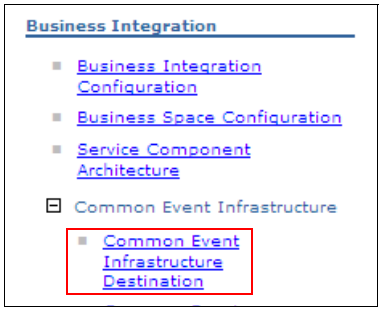


Figure 11-41 CEI Destination link

4. In the General Properties section, select the CEI destination radio button defined at the cell scope. In our lab, this value was cell/clusters/RMS.Support/com/ibm/events/configuration/emitter/Default (Figure 11-42).

Figure 11-42 Cell-scoped CEI destination

Click **OK**.

5. Click the **Save** link to save changes to the master configuration.
6. If you have automatic synchronization enabled, when the synchronization process is complete, you should see the following message:

   `The configuration synchronization complete for cell.`

   Click **OK**. Otherwise, manually synchronize the changes.
7. With the additional application target cluster configured, start the cluster.
8. Verify that the cluster members start without error by checking each member's `SystemOut.log` for exceptions.

Adding an additional messaging cluster

To add an additional messaging cluster, perform the following steps. Each of these steps is explained in the following sections.

1. Creating an additional messaging engine database. See page 306.
2. Creating the second messaging cluster. See 307.
3. Configuring SCA support for the additional messaging cluster. See page 310.
4. Configuring target significance. See page 312.

Creating an additional messaging engine database

There is no predefined WebSphere Process Server script to create a separate messaging engine database. By default, WebSphere Process Server assumes that the messaging engine data store will be incorporated into the common database (WPRCSDB). In this section, you will create a separate database for the second messaging engine cluster called MEDB2.

Once the data store is created, the schemas and tables required are created the first time that the messaging engines connect to the database. Alternately, you may create the messaging engine database and use the **sibDDLGenerator** command to create the messaging engine schemas and tables.

When you create the new data store for the second set of messaging engines in the cell, you can create new schemas in the existing MEDB or you can create a new database. The option that you choose ultimately depends upon the database system that you are using and the performance tuning required. For example, in DB2, creating a new database improves performance.

To create a new DB2 database for the second messaging cluster:

1. Issue the command to create the database:
   a. In a DB2 command window, enter the following command to create the database:

      ```
      db2 CREATE DB MEDB2 USING CODESET UTF-8 TERRITORY en-us
      ```

   b. When the database is created you should see the following message:

      ```
      The CREATE DATABASE command completed successfully.
      ```

      Close the DB2 command window.

> **Note:** You can also create a script to generate MEDB2.

c. Issue the following command to close the connection:

```
db2 CONNECT RESET
```

d. Close the DB2 command window.

Creating the second messaging cluster

To create the second messaging cluster in the cell:

1. In the Integrated Solutions Console navigation pane, expand **Servers** and click the **Clusters** link.
2. In the Server clusters window, click the **New** button. The Step 1: Enter basic cluster information window opens.
3. Enter `RMS.Messaging2` in the Cluster name text box (Figure 11-43).

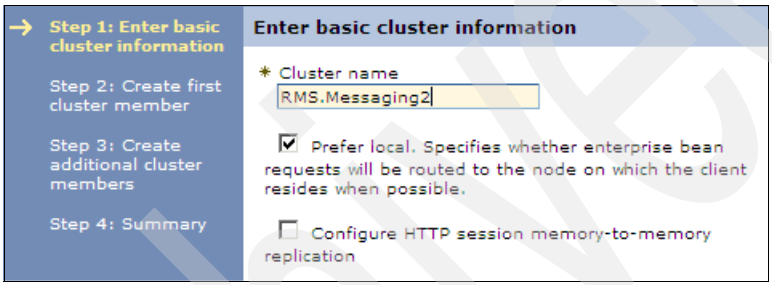


Figure 11-43 New messaging cluster name

Click **Next**. The Step 2: Create first cluster member window opens.

4. Perform the following steps to create the first cluster member (Figure 11-44):
   a. In the Member name text box, enter the name of the first cluster member (in our lab this value was RMS.Messaging2.ITSO1Node01.0).
   b. In the Select node text box, choose the appropriate node from the drop-down list (in our lab this value was ITSO1Node01).
   c. Leave the Generate unique HTTP ports check box selected.
   d. In the Select basis for first cluster member section, click the **Create the member using an application server template** radio box, and choose **default** from the drop-down list. Because the messaging components that support WebSphere Process Server are derived from base WebSphere Application Server functionality, the default WebSphere Application Server template is all that is required.

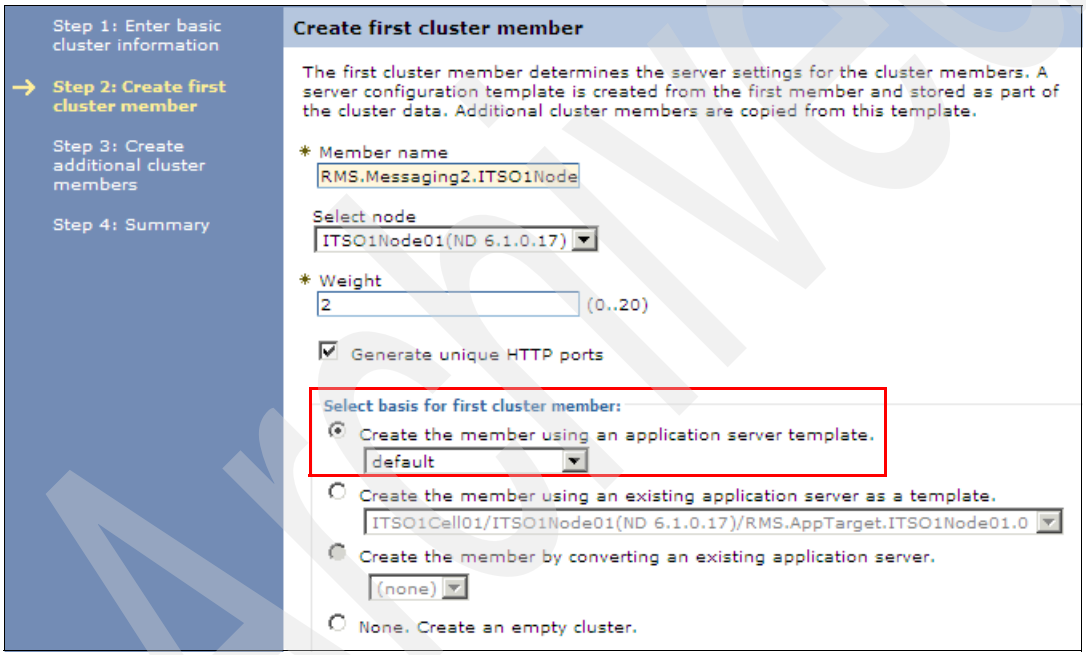


Figure 11-44 Add first member to second messaging cluster

Click **Next**. The Step 3: Create additional cluster members window opens.

5. Perform the following steps to create additional cluster members (Figure 11-45):

   a. Enter the name of the additional cluster member in the Member name text box. In our lab, the name RMS.Messaging2.ITSONode03.0 was used to keep the naming conventions in line with the names generated during template-driven topology creation.

   b. Choose the appropriate node from the Select node drop-down list. In our lab, this value was ITSO2Node03. Because the second messaging cluster will not be a member of the CEI bus, there is no need for a messaging server instance on Node02.

Figure 11-45 Add additional cluster member

6. Click **Add Member**. The name of the additional cluster member should appear in the table below.

   > **Note:** Adding members during cluster creation is not required. You may create the cluster with one member first and verify the cluster configuration before adding more members. For demonstration purposes, all cluster members were added at the same time in this example.

   Click **Next**. The Step 4: Summary window opens.

7. Review your options and click **Finish**.

8. Click the **Save** link at the top of the window.

9. If you have automatic synchronization enabled, when the synchronization process is complete, you should see the following message:

   `The configuration synchronization complete for cell.`

   Click **OK**. Otherwise, manually synchronize the changes when you are done.

   You should be returned to the Server clusters window, and you should see your newly created cluster. Do not start the cluster at this time. You will configure the remaining options before you start the cluster.

Note: If you wish to identify which messaging engines run on each of the members of the RMS.Messaging2 cluster, you must create three policies (one for the SCA.APPLICATION messaging engine, one for the SCA.SYSTEM messaging engine, and one for the BPC engine). Instructions for creating the policies necessary to implement this can be found in 11.5, “Distributing messaging workload using policies” on page 318.

Configuring SCA support for the additional messaging cluster

By default, new servers and clusters in a network deployment or managed node environment are not configured to host SCA applications and their destinations. In this section, you use the Integrated Solutions Console to configure the second remote messaging cluster to support SCA. Configuring SCA for the second messaging cluster automatically adds the cluster as a member of the SCA.SYSTEM bus and the SCA.APPLICATION bus.

To configure SCA support for the second messaging cluster:

1. In the Integrated Solutions Console, expand **Servers** and click the **Clusters** link.
2. In the Server clusters window, click the **RMS.Messaging2** link.
3. In the Business Integration section, click the **Service Component Architecture** link.
4. Select the Support the Service Component Architecture components check box and click the Local radio button (Figure 11-46).

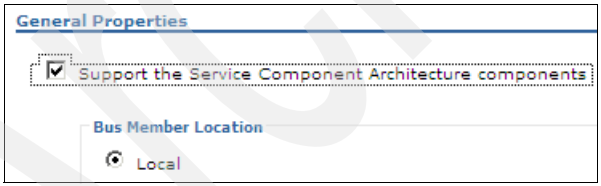


Figure 11-46 Support SCA

Because the current cluster that you are configuring will be used as the messaging engine cluster for your SCA components, the Bus Member Location is considered local.

5. In the System bus member section, perform the following steps (Figure 11-47):
   a. Enter `MEDB2` in the Database Instance text box.
   b. Enter `MESS02` in the Schema text box.
   c. Ensure that the Create Tables check box is selected. You can also create a file for the database administrator to run to create the tables using the **`sibDDLGenerator`** command.
   d. Verify that the following text boxes are populated:
      - User name (In our lab, this value was db2admin.)
      - Password (In our lab, this value was web1sphere.)
      - Server (In our lab, this value was ITSO2.)
   e. Verify that DB2 Universal is the selection in the Provider drop-down list.

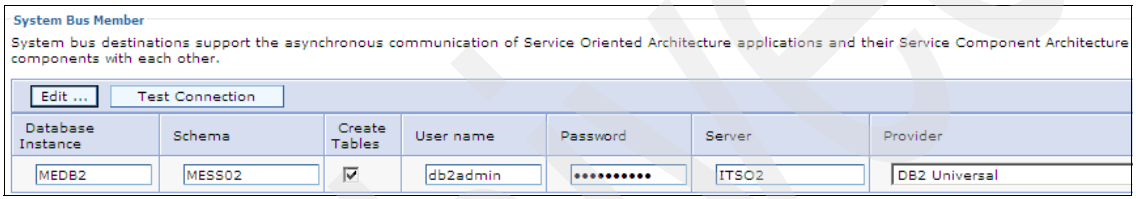


Figure 11-47 System Bus Member properties

6. In the Application Bus Member section, perform the following steps (Figure 11-48):
   a. Enter `MEDB2` in the Database Instance text box.
   b. Enter `MESA02` in the Schema text box.
   c. Ensure that the Create Tables check box is selected. You can also create a file for the database administrator to run to create the tables using the **`sibDDLGenerator`** command.
   d. Verify that the following text boxes are populated:
      - User name (In our lab, this value was db2admin.)
      - Password (In our lab, this value was web1sphere.)
      - Server (In our lab, this value was ITSO2.)
   e. Verify that DB2 Universal is the selection in the Provider drop-down list.

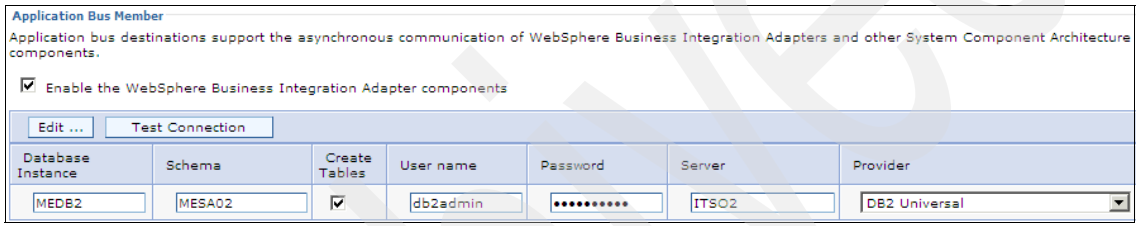


Figure 11-48 Application Bus Member properties

   Click **OK**.
7. Click the **Save** link at the top of the window to save your changes to the master configuration.
8. If you have automatic synchronization enabled, when the synchronization process is complete, you should see the following message:

   `The configuration synchronization complete for cell.`

   Click **OK**. Otherwise, manually synchronize the changes.

Configuring target significance

Because the sample used for the other chapters does not contain asynchronous interactions, the steps below use the JMS invocation sample application available from the BPC samples page at the following Web page:

`http://publib.boulder.ibm.com/bpcsamp/gettingStarted/jmsInvocation/download.htm`

To configure target significance for JMS connection factories and activation specifications:

1. Download and install the following JMS invocation sample applications, in this order:

   a. JMSInvokerApp.ear
   b. MPGConverterApp.ear

2. Configure the connection factories and activation specifications generated for the JMS invocation sample application for target significance:

   a. In the Integrated Solutions Console, expand **Resources** → **JMS** and click the **Connection factories** link. You should see two new connection factories create by the JMS invocation application EAR files (Figure 11-49).

☐	JMSInvoker.MPGConverterProcessExport_CF	JMSInvoker/MPGConverterProcessExport_CF	Default messaging provider
☐	MPGConverter.MPGConverterProcessExport_CF	MPGConverter/MPGConverterProcessExport_CF	Default messaging provider

Figure 11-49 JMS invocation application connection factories

   b. Click the **JMSInvoker.MPGConverterProcessExport_CF** link.

c. In the Connection section, perform the following steps (Figure 11-50):

   i. Ensure that the Bus name text box is populated with the name of the SCA.APPLICATION bus.

      Because this is a generic JMS export, it is handled by the SCA.APPLICATION bus. In our lab, this value was SCA.APPLICATION.ITSO1Cell01.Bus.

   ii. In the Target text box, enter the name of the SCA.APPLICATION bus messaging engine that you want the application to use.

      In our lab, the application was deployed to the RMS.AppTarget cluster (not RMS.AppTarget2), so the value used here, RMS.Messaging.000-SCA.APPLICATION.Bus, was the name of the SCA.APPLICATION messaging engine used by the RMS.Messaging cluster. This establishes an affinity between the application in RMS.AppTarget and the engine in the RMS.Messaging cluster.

      If you had deployed the application to RMS.AppTarget2, you could use the name of the SCA.APPLICATION messaging engine used by RMS.Messaging2 instead. Establishing target significance in this manner is not required. It was configured this way for convenience. You could have configured RMS.AppTarget to use RMS.Messaging2.

   iii. In the Target type text box, select **Messaging engine name**.

   iv. In the Target significance text box, select **Required**.

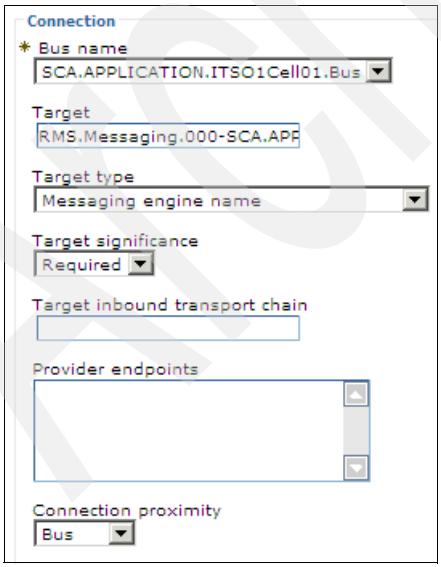


Figure 11-50 Connection factory properties

Click **OK**.

d. Click the **Save** link at the top of the window to save your changes to the master configuration.

e. If you have automatic synchronization enabled, when the synchronization process is complete, you should see the following message:

```
The configuration synchronization complete for cell.
```

Click **OK**. Otherwise, manually synchronize the changes when you are done.

f. Repeat the previous steps to configure target significance for MPGConverter.MPGConverterProcessExport_CF.

g. In the Integrated Solutions Console, expand **Resources** → **JMS** and click the **Activation specifications** link. You should see two new activation specifications associated with the JMS invocation sample application (Figure 11-51).

☐	JMSInvoker.MPGConverterProcessExport_AS	JMSInvoker/MPGConverterProcessExport_AS	Default messaging provider
☐	MPGConverter.MPGConverterProcessExport_AS	MPGConverter/MPGConverterProcessExport_AS	Default messaging provider

Figure 11-51 JMS invocation application activation specifications

h. Click the **JMSInvoker.MPGConverterProcessExport_AS** link.

i. In the Destination section, perform the following steps (Figure 11-52):

   i. Ensure that the Bus name text box is populated with the name of the SCA.APPLICATION bus.

      Because this is a generic JMS export, it is handled by the SCA.APPLICATION bus. In our lab, this value was SCA.APPLICATION.ITSO1Cell01.Bus.

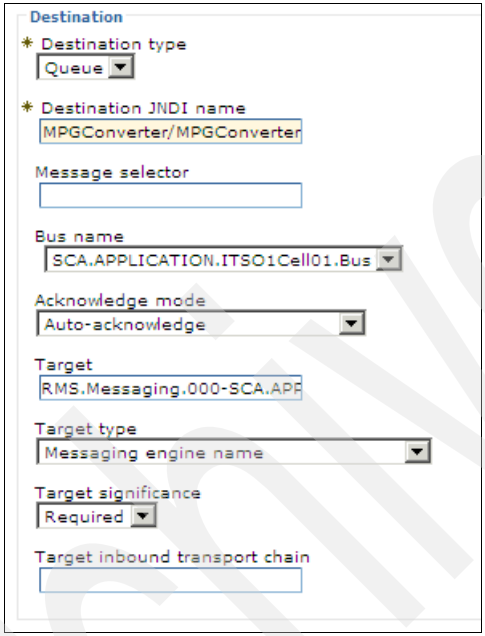


Figure 11-52 Activation specification properties

   ii. In the Target text box, enter the name of the SCA.APPLICATION bus messaging engine that you want the application to use.

      In our lab, the application was deployed to the RMS.AppTarget cluster (not RMS.AppTarget2), so the value used here (RMS.Messaging.000-SCA.APPLICATION.Bus) was the name of the SCA.APPLICATION messaging engine used by the RMS.Messaging cluster. This establishes an affinity between the application in RMS.AppTarget and the engine in the RMS.Messaging cluster.

      If you had deployed the application to RMS.AppTarget2, you could use the name of the SCA.APPLICATION messaging engine used by RMS.Messaging2 instead. Establishing target significance in this manner is not required. It was configured this way for convenience. You

could just as easily configure RMS.AppTarget to use RMS.Messaging2.

iii. In the Target type text box, select **Messaging engine name** from the drop-down list.

iv. In the Target significance text box, select **Required** from the drop-down list.

j. Click **OK**.

k. Click the **Save** link at the top of the window to save your changes to the master configuration.

l. If you have automatic synchronization enabled, when the synchronization process is complete, you should see the following message:

```
The configuration synchronization complete for cell.
```

m. Click **OK**. Otherwise, manually synchronize the changes when you are done.

n. Repeat the previous steps to configure target significance for MPGConverter.MPGConverterProcessExport_AS.

3. Once you have completed configuring target significance for any connection factories and activation specifications used by your applications, you must configure target significance for the internal JMS resources used by the human task manager and business flow manager. Repeat the previous steps to configure appropriate target significance for each of the remaining connection factories and activation specifications.

11.5 Distributing messaging workload using policies

By default, when you start the messaging cluster, the first server started will activate the messaging engine for each of the four buses required by WebSphere Process Server. This behavior is represented in Figure 11-53.

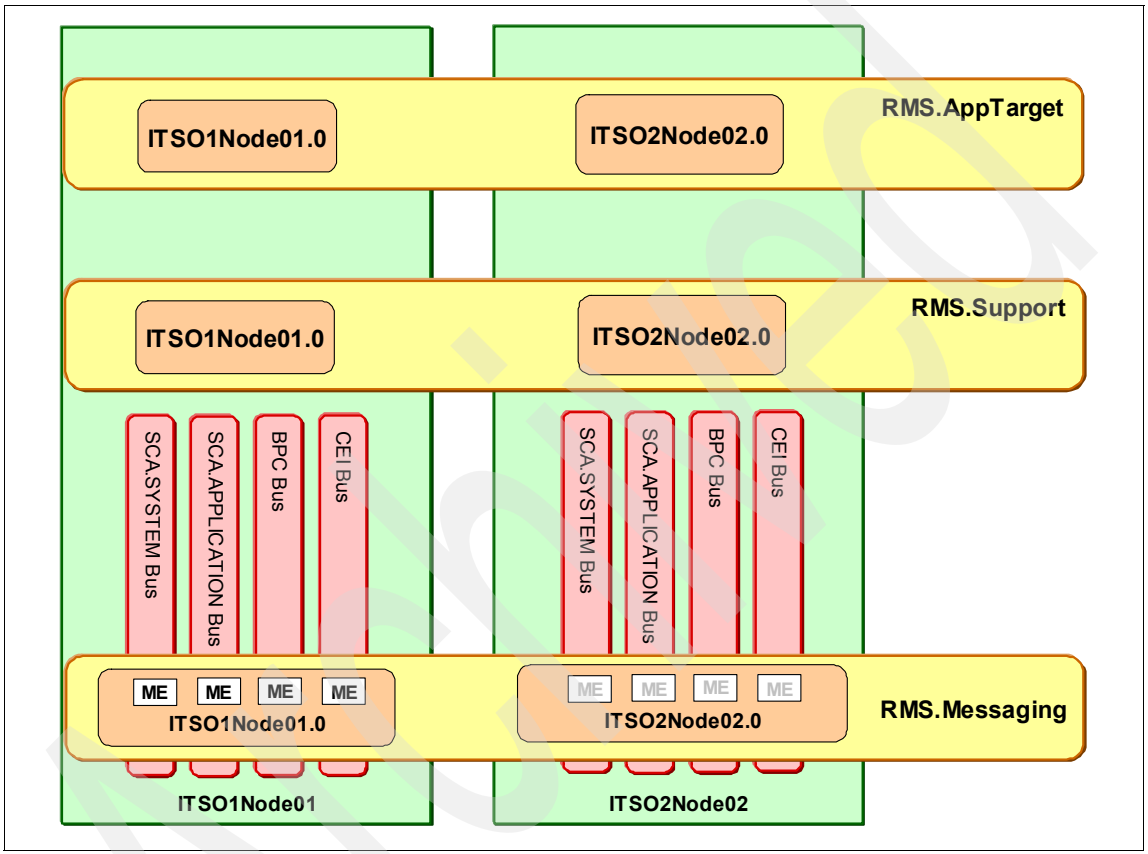


Figure 11-53 Default messaging engine behavior

Here, each of the four messaging engines is in a Started state on ITSO1Node01.0. The four messaging engines on ITSO2Node02.0 are all in a Joined state. They are on stand-by in case one of the engines on ITSO1Node01.0 fails.

To override this behavior, you must create service integration bus policies for the messaging engines that identify which servers run each of the four engines:

- 000-SCA.SYSTEM
- 000-BPC
- 000-SCA.APPLICATION
- 000-CEI

When you create the policies, you configure the preferred servers list to reflect which cluster member should run each messaging engine.

In production, you may wish to do this to ensure that the most robust machine available is always the preferred messaging server for the most heavily used messaging engine (usually 000-SCA.SYSTEM or 000-BPC). This configuration also allows you to use different servers to run each of the messaging engines. For example, you may wish to create a policy to run the SCA.SYSTEM and SCA.APPLICATION engines on one server, and two additional policies to run the CEI and BPC engines on other servers. As a best practice, you should not separate the SCA.APPLICATION and SCA.SYSTEM engines. Because they interact, you should keep these two engines on the same server.

In the lab used for this publication, the original Remote Messaging and Remote Support topology was extended from two nodes to three. The third node contains a third member of the RMS.Messaging cluster, which was created using the WebSphere Application Server template. To distribute the messaging engines across these three cluster members, four messaging engine policies were created:

- The first policy identified messaging cluster member one (ITSO1Node01.0 on node one) as the server used to run the SCA.SYSTEM engine.
- The second policy also identified cluster member one as the server used to run the SCA.APPLICATION engine.
- The third policy identified messaging cluster member two (ITSO2Node02.0 on node two) as the server used to run the CEI engine.
- The fourth policy identified messaging cluster member 3 (ITSO2Node03.0 on node three) as the server used to run the BPC engine.

This topology is represented in Figure 11-54.

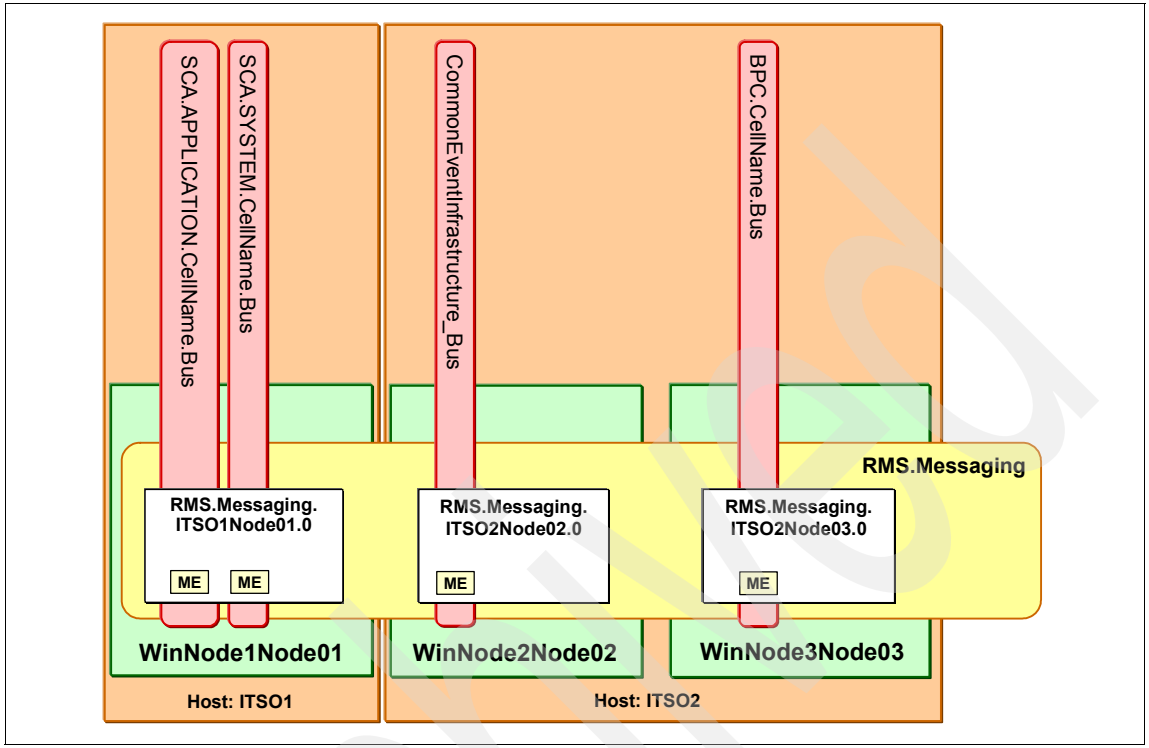


Figure 11-54 Using policies to run messaging engines on separate servers

11.5.1 Create the SCA.SYSTEM messaging engine policy

To control the startup and failover behavior of the messaging engines, you should create a policy for each of the engines. This requires a total of four policies, one for each messaging engine used by WebSphere Process Server.

The first policy that you should create is the SCA.SYSTEM messaging engine policy. This messaging engine supports asynchronous communication between SCA components and applications. It also support asynchronous communication with WebSphere (JCA) adapters.

To create the SCA.SYSTEM messaging engine policy used to implement the topology used in this publication:

1. In the Integrated Solutions Console, expand **Servers** → **Core groups** and select the **Core group settings** link (Figure 11-55).

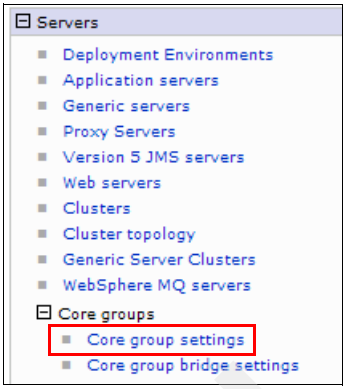


Figure 11-55 Core group settings link

2. Click the **DefaultCoreGroup** link (Figure 11-56).

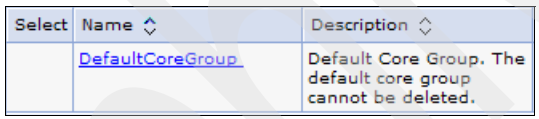


Figure 11-56 DefaultCoreGroup link

3. In the Additional Properties section, click the **Policies** link (Figure 11-57).

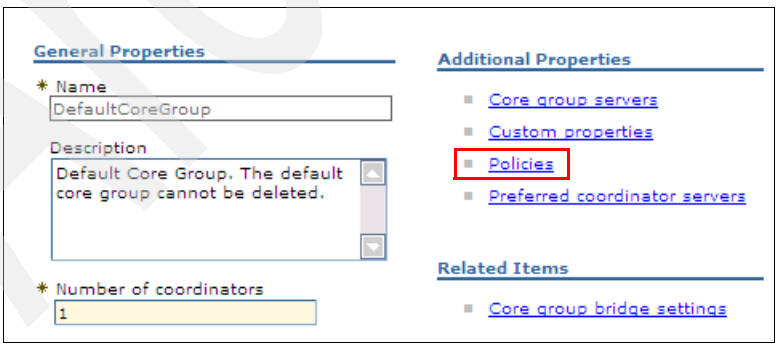


Figure 11-57 Policies link

4. Click the **New** button (Figure 11-58).

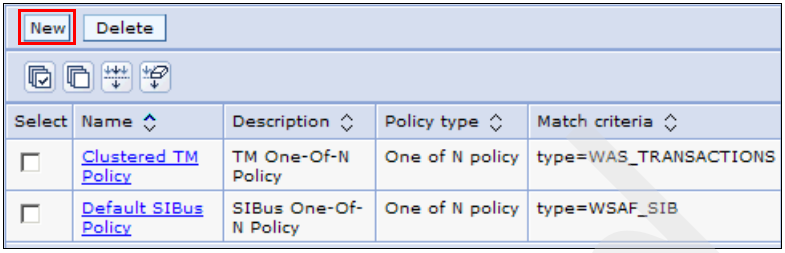


Figure 11-58 Create new policy

5. In the General Properties section, for the Policies text box, select **One of N policy** from the drop-down list (Figure 11-59).

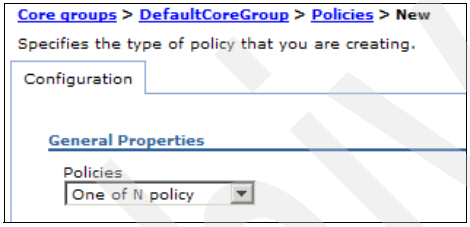


Figure 11-59 Choose policy type

Click **Next**.

6. In the General Properties section, perform the following steps (Figure 11-60):
   a. For the Name text box, enter `SCA_SYS_ME_Policy`.
   b. Ensure that the Policy type text box is set to `One of N policy`.

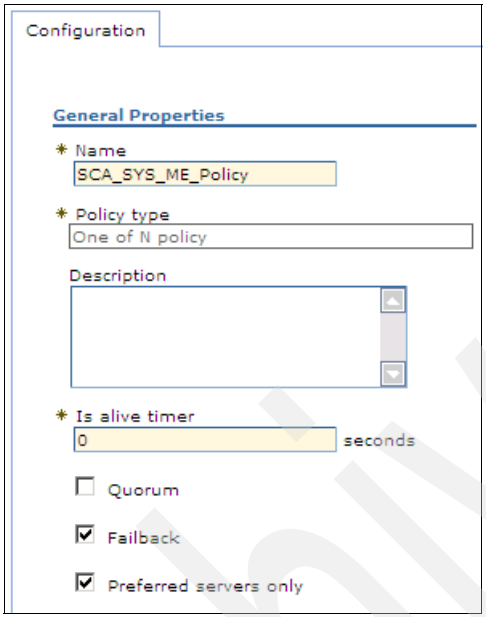


Figure 11-60 New SCA_SYS_ME_Policy

   c. Leave the Is alive timer text box set to 0 (zero).

      This text box specifies the time interval, in seconds, at which the high availability manager will check the health of all of the active high availability group members that are running this application server process. If 0 is specified, the default value of 120 seconds is used.
   d. Select the **Failback** check box.

      This option ensures that if the messaging engine fails and is started on another server, when the preferred server becomes available, the high availability manager restarts the engine on the preferred server.
   e. Select the **Preferred servers only** check box.

      By selecting the Preferred servers only check box, the messaging engine is incapable of running on a server that is not in the preferred servers list.

Click **OK**. You should be returned to the core groups window with the following message at the top of the window (Figure 11-61):

```
The policy must have at least one match criteria defined.
```

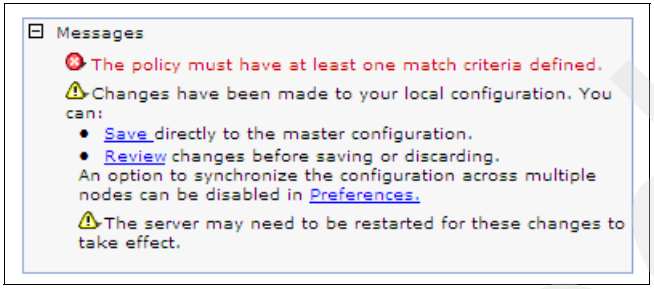


Figure 11-61 Error: No match criteria defined

7. In the Additional Properties section, click the **Match Criteria** link (Figure 11-62).

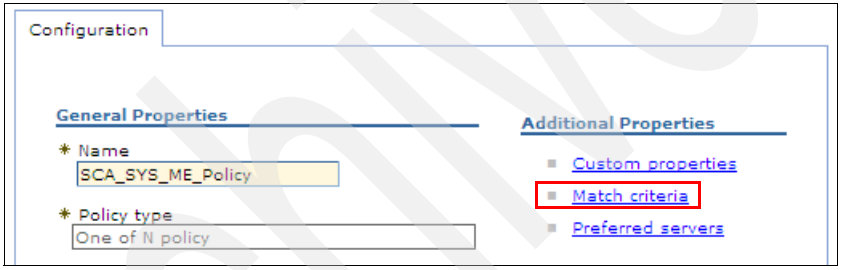


Figure 11-62 Match criteria link

8. Click the **New** button.

9. In the General Properties section, perform the following steps (Figure 11-63):
   a. In the Name text box, enter type (any messaging engine).
   b. In the Value text box, enter `WSAF_SIB`.
   c. (Optional) Enter a policy description.

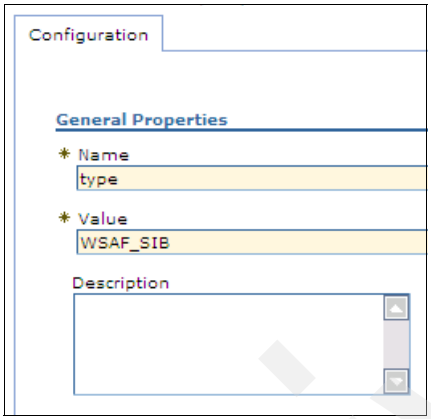


Figure 11-63 Type match criteria

   Click **OK**.

10. At the Match criteria window, click the **New** button.

11. In the General Properties section, perform the following steps (Figure 11-64):
    a. In the Name text box, enter `IBM_hc` (all messaging engines in a particular cluster).
    b. In the Value text box, enter <ClusterName>. In our lab, this value was RMS.Messaging.
    c. (Optional) Enter a description of the match criteria.

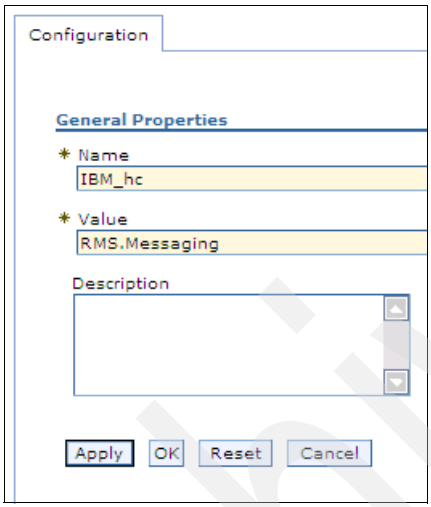


Figure 11-64 Cluster match criteria

Click **OK**.

12. When you are returned to the Match criteria window, click the **New** button.

13. In the General Properties section, perform the following steps (Figure 11-65):
    a. In the Name text box, enter `WSAF_SIB_BUS` (a particular bus).
    b. In the Value text box, enter `SCA.SYSTEM.<CellName>.Bus`. In our lab, this value was SCA.SYSTEM.ITSO1Cell01.Bus.
    c. (Optional) Enter a description of the match criteria.

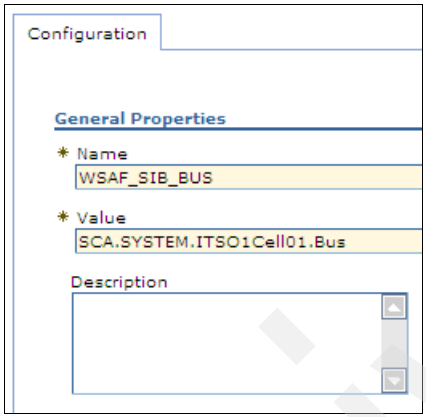


Figure 11-65 Bus match criteria

Click **OK**.

14. In the Match criteria window, click the **New** button.

15. In the General Properties section, perform the following steps (Figure 11-66):

   a. In the Name text box, enter `WSAF_SIB_MESSAGING_ENGINE` (a particular messaging engine).

   b. In the Value text box, enter `<MessagingClusterName>.000-SCA.SYSTEM.<CellName>.Bus`. In our lab, this value was RMS.Messaging.000-SCA.SYSTEM.ITSO1Cell01.Bus.

   c. (Optional) Enter a description of the match criteria.

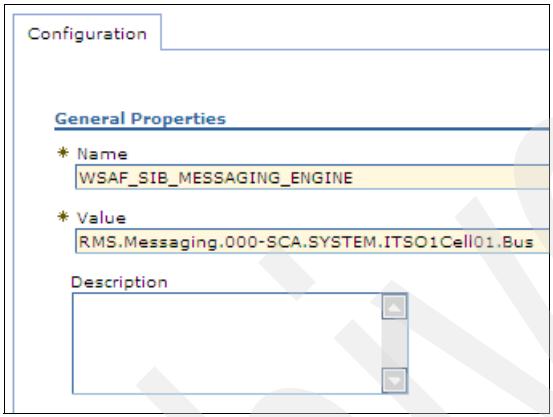


Figure 11-66 Messaging engine match criteria

Click **OK**. In the match criteria window, you should see all four of the criteria that you created (Figure 11-67).

Select	Name	Value	Description
☐	IBM_hc	RMS.Messaging	
☐	WSAF_SIB_BUS	SCA.SYSTEM.ITSO1Cell01.Bus	
☐	WSAF_SIB_MESSAGING_ENGINE	RMS.Messaging.000-SCA.SYSTEM.ITSO1Cell01.Bus	
☐	type	WSAF_SIB	

Figure 11-67 SCA_SYS_ME _Policy match criteria

Because this policy now has a match weight factor of four (because you specified four match criteria), it should override the default service integration bus policy with its match weight factor of one (type = WSAF_SIB). When multiple policies apply to the same processes, the policy with the highest

weight factor wins. You should take care to avoid creating situations where two policies have the same weight factor. If the high availability manager sees a tie, an exception is thrown.

16. Click the **SCA_SYS_ME_Policy** link in the breadcrumb trail at the top of the core groups window (Figure 11-68).

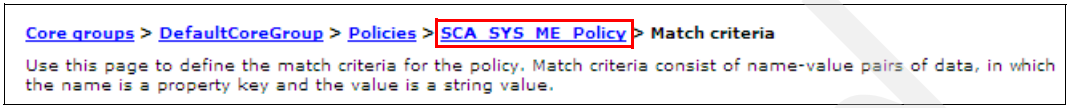


Figure 11-68 Breadcrumb trail

17. In the Additional Properties section, click the **Preferred servers** link.

18. In the Core group servers section, select the first preferred server **<HostNameNodeName>/<MessagingClusterMember>** and click **Add**. In our lab, this value was ITSO1Node01/RMS.Messaging.ITSO1Node01.0. This is shown in Figure 11-69.

Figure 11-69 Add the first preferred server

19. In the Core group servers section, select the second preferred server **<HostNameNodeName>/<MessagingClusterMember>** and click **Add**. In our lab, this value was ITSO2Node02/RMS.Messaging.ITSO2Node02.0). This is shown in Figure 11-70.

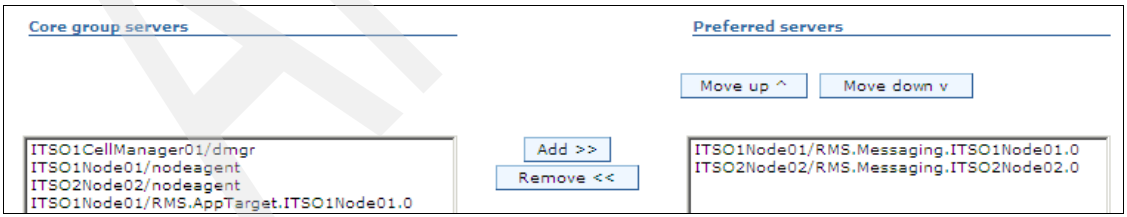


Figure 11-70 Add the second preferred server

20. In the Core group servers section, select the third preferred server **<HostNameNodeName>/<MessagingClusterMember>** and click **Add**

(Figure 11-71). In our lab, this value was ITSO2Node03/RMS.Messaging.ITSO2Node03.0.

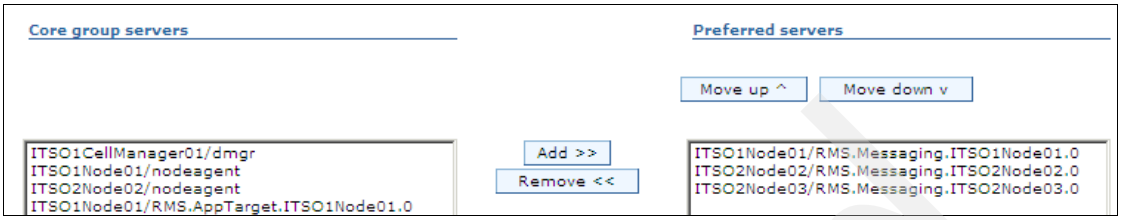


Figure 11-71 Add the third preferred server

Adding the servers to the Preferred servers list in this order should force the SCA.SYSTEM messaging engine to always start on RMS.Messaging.ITSO1Node01.0.

If this cluster member is unavailable, the high availability manager should start the messaging engine on RMS.Messaging.ITSO2Node02.0. If that server is unavailable, the high availability manager should start the messaging engine on RMS.Messaging.ITSO2Node03.0. Because you selected the Preferred servers only option, only the three servers listed can run the SCA.SYSTEM messaging engine.

Click **OK**.

21. Click the **Save** link at the top of the window (Figure 11-72).

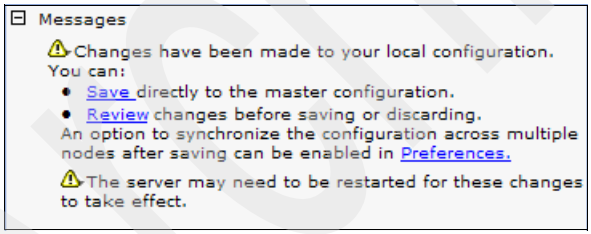


Figure 11-72 Save changes to the master configuration

22. If you have automatic synchronization enabled, when the synchronization process is complete, you should see the following message:

```
The configuration synchronization complete for cell.
```

Click **OK**. Otherwise, manually synchronize the changes when you are done creating policies.

You should be returned to the Policies window, and you should see your newly created policy.

11.5.2 Create the SCA.APPLICATION messaging engine policy

The second policy that you should create is the policy that controls the behavior of the SCA.APPLICATION messaging engine. This engine supports the SCA.APPLICATION bus, which enables asynchronous interactions for WebSphere Business Integration adapters (the non-JCA adapters) and generic JMS components. When an application is deployed, you can specify that the SCA.SYSTEM bus should be used for generic JMS components. In that scenario, the SCA.APPLICATION bus would only be used for WebSphere Business Integration adapters.

This policy should match the one that you create for the SCA.SYSTEM messaging engine. Because the two engines interact, you should keep them on the same server. To create the policy used to implement the topology used in this publication:

1. In the Policies window, click the **New** button.
2. In the General Properties section, for the Policies text box, select **One of N policy** from the drop-down list and click **Next**.
3. In the General Properties section, perform the following steps:
   a. For name, enter `SCAAppME000` (ME zero zero zero).
   b. Ensure that the policy type is set to One of N policy.
   c. Leave the Is alive timer set to 0.
   d. Select the **Preferred servers only** check box.

   Click **OK**. You should be returned to the core groups window with the following message at the top of the window.

   ```
   The policy must have at least one match criteria defined
   ```
4. In the Additional Properties section, click the **Match Criteria** link.
5. Follow the steps in 11.5.1, “Create the SCA.SYSTEM messaging engine policy” on page 320, to create the match criteria shown in Table 11-1.

Table 11-1 SCA.APPLICATION messaging engine policy match criteria

Criteria name	Value
type	WSAF_SIB
IBM_hc	RMS.Messaging
WSAF_SIB_BUS	SCA.APPLICATION.<CellName>.Bus (for example, SCA.APPLICATION.ITSO1Cell01.Bus)

Criteria name	Value
WSAF_SIB_MESSAGING_ENGINE	<ClusterName>.000-SCA.APPLICATION.<CellName>.Bus (for example, RMS.Messaging.000-SCA.APPLICATION.ITSO1Cell01.Bus)

6. Click the **SCAAppME000** link in the breadcrumb trail at the top of the core groups window.
7. In the Additional Properties section, click the **Preferred servers** link.
8. In the Core group servers section, select the first preferred server **<HostNameNodeName>/<MessagingClusterMember>** and click **Add**. In our lab, this value was ITSO1Node01/RMS.Messaging.ITSO1Node01.0.
9. In the Core group servers section, select the second preferred server **<HostNameNodeName>/<MessagingClusterMember>** and click **Add**. In our lab, this value was ITSO2Node02/RMS.Messaging.ITSO2Node02.0.
10. In the Core group servers section, select the third preferred server **<HostNameNodeName>/<MessagingClusterMember>** and click **Add**. In our lab, this value was ITSO2Node03/RMS.Messaging.ITSO2Node03.0.

    About the order in which you add servers: Adding the servers to the Preferred servers list in this order should force the SCA.APPLICATION messaging engine to always start on RMS.Messaging.ITSO1Node01.0 (the same server as the SCA.SYSTEM messaging engine). If this cluster member is unavailable, the high availability manager should start the messaging engine on RMS.Messaging.ITSO2Node02.0. If this cluster member is unavailable, the high availability manager should start the messaging engine on RMS.Messaging.ITSO2Node03.0. Because you selected the Preferred servers only option, only the three servers listed can run the SCA.APPLICATION messaging engine.

    Click **OK**.
11. Click the **Save** link at the top of the window.
12. If you have automatic synchronization enabled, when the synchronization process is complete, you should see the following message:

    ```
    The configuration synchronization complete for cell.
    ```

    Click **OK**. Otherwise, manually synchronize the changes when you are done creating policies.

    You will be returned to the Policies window and see your newly created policy.

11.5.3 Creating Common Event Infrastructure messaging engine policy

The third policy that you should create controls the behavior of the CEI messaging engine. This engine supports the CommonEventInfrastructure_Bus, which enables asynchronous event propagation for the Common Event Infrastructure.

To create the CEI messaging engine policy used to implement the topology used in this publication:

1. In the Policies window, click the **New** button.
2. In the General Properties section, for the Policies text box, select **One of N policy** from the drop-down list and click **Next**.
3. In the General Properties section, perform the following steps:
   a. For name, enter `CEI_ME000` (ME zero zero zero).
   b. Ensure that the policy type is automatically set to One of N policy.
   c. Leave the Is alive timer set to 0.
   d. Select the **Preferred servers only** check box.

   Click **OK**. You should be returned to the core groups window with the following message at the top of the window:

   ```
   The policy must have at least one match criteria defined
   ```
4. In the Additional Properties section, click the **Match Criteria** link.
5. Follow the steps in 11.5.2, “Create the SCA.APPLICATION messaging engine policy” on page 331, to create the match criteria shown in Table 11-2.

Table 11-2 CEI messaging engine policy match criteria

Criteria name	Value
type	WSAF_SIB
IBM_hc	RMS.Messaging
WSAF_SIB_BUS	CommonEventInfrastructure_Bus
WSAF_SIB_MESSAGING_ENGINE	<ClusterName>.000-CommonEventInfrastructure_Bus (for example, RMS.Messaging.000-CommonEventInfrastructure_Bus)

6. Click the **CEI_ME000** link in the breadcrumb trail at the top of the core groups window.
7. In the Additional Properties section, click the **Preferred servers** link.

8. In the Core group servers section, select the first preferred server **<HostNameNodeName>/<MessagingClusterMember>** and click **Add**. In our lab, this value was ITSO2Node02/RMS.Messaging.ITSO2Node02.0.
9. In the Core group servers section, select the second preferred server **<HostNameNodeName>/<MessagingClusterMember>** and click **Add**. In our lab, this value was ITSO1Node01/RMS.Messaging.ITSO1Node01.0.
10. In the Core group servers section, select the third preferred server **<HostNameNodeName>/<MessagingClusterMember>** and click **Add**. In our lab, this value was ITSO2Node03/RMS.Messaging.ITSO2Node03.0.

    Note about the order in which you add servers: Adding servers in this order forces the messaging engine to start on RMS.Messaging.ITSO2Node02.0. If this cluster member is unavailable, the high availability manager starts the messaging engine on RMS.Messaging.ITSO1Node01.0. If this cluster member is unavailable, the high availability manager starts the messaging engine on RMS.Messaging.ITSO2Node03.0. Because you selected the Preferred servers only option, only the three servers listed can run the CEI messaging engine.

    Click **OK**.
11. Click the **Save** link at the top of the window. If you have automatic synchronization enabled, when the synchronization process is complete, you should see the following message:

    ```
    The configuration synchronization complete for cell.
    ```

    Click **OK**. Otherwise, manually synchronize the changes when you are done creating policies.
12. You will be returned to the Policies window and see your newly created policy.

11.5.4 Business Process Choreographer messaging engine policy

The fourth policy that you should create is the policy that controls the behavior of the BPC messaging engine. This engine supports the BPC bus, which enables internal process navigation and the business flow manager's Java Messaging Service (JMS) API. In our lab environment, the BPC messaging engine was configured to run on the third node in the topology.

To create the BPC messaging engine policy used to implement the topology used in this publication, perform the following steps:

1. In the Policies window, click the **New** button.

2. In the General Properties section, for the Policies text box, select **One of N policy** from the drop-down list and click **Next**.
3. In the General Properties section, perform the following steps:

   a. For name, enter `BPC_ME000` (ME zero zero zero).
   b. Ensure that the policy type is set to One of N policy.
   c. Leave the Is alive timer set to 0.
   d. Click the **Preferred servers only** check box.

   Click **OK**. You should be returned to the core groups window with the following message at the top of the window:

   ```
   The policy must have at least one match criteria defined
   ```

4. In the Additional Properties section, click the **Match Criteria** link.
5. Follow the steps in 11.5.3, “Creating Common Event Infrastructure messaging engine policy” on page 333, to create the match criteria shown in Table 11-3.

Table 11-3 Business Process Choreographer messaging engine policy match criteria

Criteria name	Value
type	WSAF_SIB
IBM_hc	RMS.Messaging
WSAF_SIB_BUS	BPC.<CellName>.Bus (for example, BPC.ITSO1Cell01.Bus)
WSAF_SIB_MESSAGING_ENGINE	<ClusterName>.000-BPC.<CellName>.Bus (for example, RMS.Messaging.000-BPC.ITSO1Cell01.Bus)

6. Click the **BPC_ME000** link in the breadcrumb trail at the top of the core groups window.
7. In the Additional Properties section, click the **Preferred servers** link.
8. In the Core group servers section, select the first preferred server **<HostNameNodeName>/<MessagingClusterMember>** and click **Add**. In our lab, this value was ITSO2Node03/RMS.Messaging.ITSO2Node03.0.
9. In the Core group servers section, select the second preferred server **<HostNameNodeName>/<MessagingClusterMember>** and click **Add**. In our lab, this value was ITSO1Node01/RMS.Messaging.ITSO1Node01.0.
10. In the Core group servers section, select the third preferred server **<HostNameNodeName>/<MessagingClusterMember>** and click **Add**. In our lab, this value was ITSO2Node02/RMS.Messaging.ITSO2Node02.0.

> **Note about the order in which you add servers:** Adding servers in this order forces the messaging engine to start on RMS.Messaging.ITSO2Node03.0. If this cluster member is unavailable, the high availability manager should start the messaging engine on RMS.Messaging.ITSO1Node01.0. If this cluster member is unavailable, the high availability manager should start the messaging engine on RMS.Messaging.ITSO2Node02.0. Because you selected the Preferred servers only option, only the three servers listed can run the BPC messaging engine.

Click **OK**.

11. Click the **Save** link at the top of the window.
12. If you have automatic synchronization enabled, when the synchronization process is complete, you should see the following message:

    `The configuration synchronization complete for cell.`

    Click **OK**. Otherwise, manually synchronize the changes when you are done creating policies.
13. You should be returned to the Policies window and you should see your newly created policy.
14. Once all four policies have been created, perform the following steps:
    a. Stop the clusters.
    b. Stop the deployment manager.
    c. Start the deployment manager.
    d. Restart the node agents.
    e. Restart the clusters.

11.5.5 Verifying the policy configuration

Once the policies have been created and the servers have been restarted, you verify the status of the messaging engines on each server.

According to the policies that you created, the SCA.SYSTEM and SCA.APPLICATION messaging engines should be started on cluster member 1. The CEI messaging engine should be started on cluster member 2 and the BPC messaging engine should be started on cluster member 3. To verify the policy configuration, perform the following steps:

1. On node one, open the `SystemOut.log` file for messaging cluster member 1. In our lab this server was named RMS.Messaging.ITSO1Node01.0. By default, this log is located in `\profiles\<ProfileName>\logs\<NodeName>` (for example, `\profiles\ITSO1\logs\ITSO1Node01.0`).

You will see the following messages in the SystemOut.log file:

```
Messaging engine RMS.Messaging.000-SCA.SYSTEM.ITSO1Cell01.Bus is in
state Started.
Messaging engine RMS.Messaging.000-SCA.APPLICATION.ITSO1Cell01.Bus
is in state Started.
Messaging engine RMS.Messaging.000-CommonEventInfrastructure_Bus is
in state Joined.
Messaging engine RMS.Messaging.000-BPC.ITSO1Cell01.Bus is in state
Joined.
```

2. On node two, open the SystemOut.log file for messaging cluster member 2. In our lab this server was named RMS.Messaging.ITSO2Node02.0. By default, this log is located in \profiles\<ProfileName>\logs\<NodeName> (for example, \profiles\ITSO2\logs\ITSO2Node02.0).

   You will see the following messages in the SystemOut.log file:

```
Messaging engine RMS.Messaging.000-SCA.SYSTEM.ITSO1Cell01.Bus is in
state Joined.
Messaging engine RMS.Messaging.000-SCA.APPLICATION.ITSO1Cell01.Bus
is in state Joined.
Messaging engine RMS.Messaging.000-CommonEventInfrastructure_Bus is
in state Started.
Messaging engine RMS.Messaging.000-BPC.ITSO1Cell01.Bus is in state
Joined.
```

3. On node two, open the SystemOut.log file for messaging cluster member 3. In our lab this server was named RMS.Messaging.ITSO2Node03.0. By default, this log is located in \profiles\<ProfileName>\logs\<NodeName> (for example, \profiles\ITSO2\logs\ITSO2Node03.0).

   You see the following messages in the SystemOut.log file:

```
Messaging engine RMS.Messaging.000-SCA.SYSTEM.ITSO1Cell01.Bus is in
state Joined.
Messaging engine RMS.Messaging.000-SCA.APPLICATION.ITSO1Cell01.Bus
is in state Joined.
Messaging engine RMS.Messaging.000-CommonEventInfrastructure_Bus is
in state Joined.
Messaging engine RMS.Messaging.000-BPC.ITSO1Cell01.Bus is in state
Started.
```

11.6 Discouraged patterns

The patterns discussed above are only a few ways to expand a topology. There are some patterns, however, that are discouraged from use.

11.6.1 One application target with multiple messaging engines

To avoid issues from having too many destinations on a single messaging engine, you may be tempted to increase the number of messaging engines without maintaining the 1-1 ratio with application clusters, as seen in Figure 11-73.

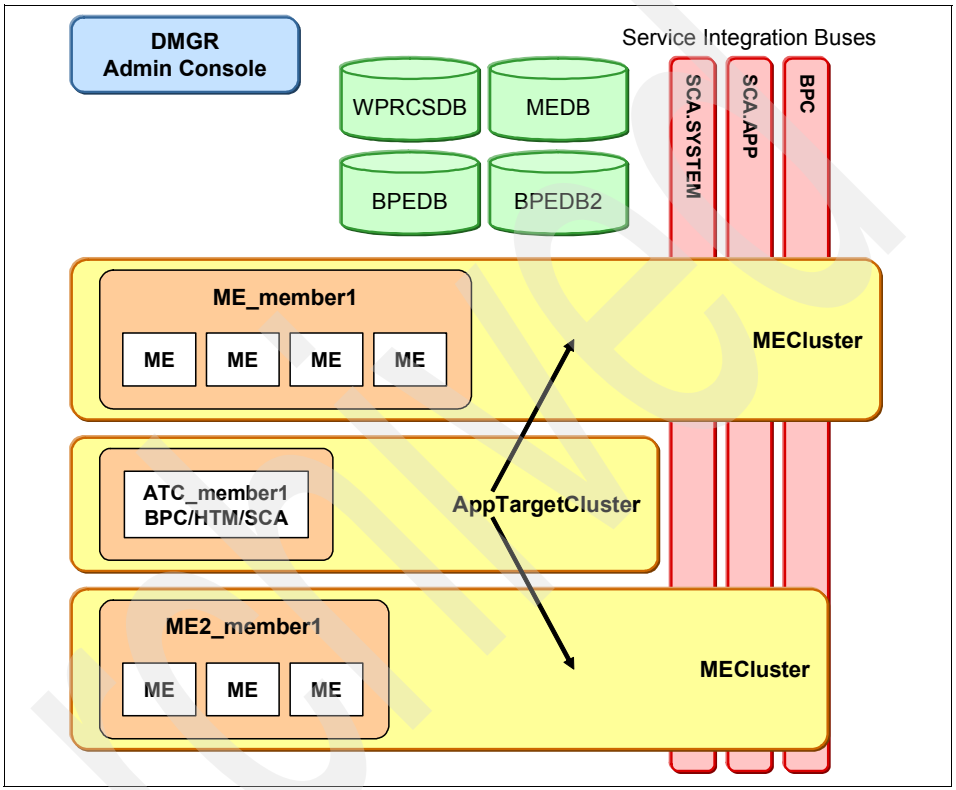


Figure 11-73 Multiple ME clusters with a single application target cluster

This pattern simply does not work. A single process server or ESB deployment target can only configure its destinations on a single member of an SIBus. This means that in a deployment with a single application target but multiple messaging engines, you cannot choose to put half of the destinations on one messaging target.

Even if the possibility arose to force this configuration, you would have distinct issues uninstalling or installing an application (you would have to reconfigure the specification for the location of the destinations each time). This ultimately becomes impossible to manage.

11.6.2 Multiple application targets with a single application target

In this topology, multiple application targets with a single messaging target are configured to share a single ME cluster, as seen in Figure 11-74.

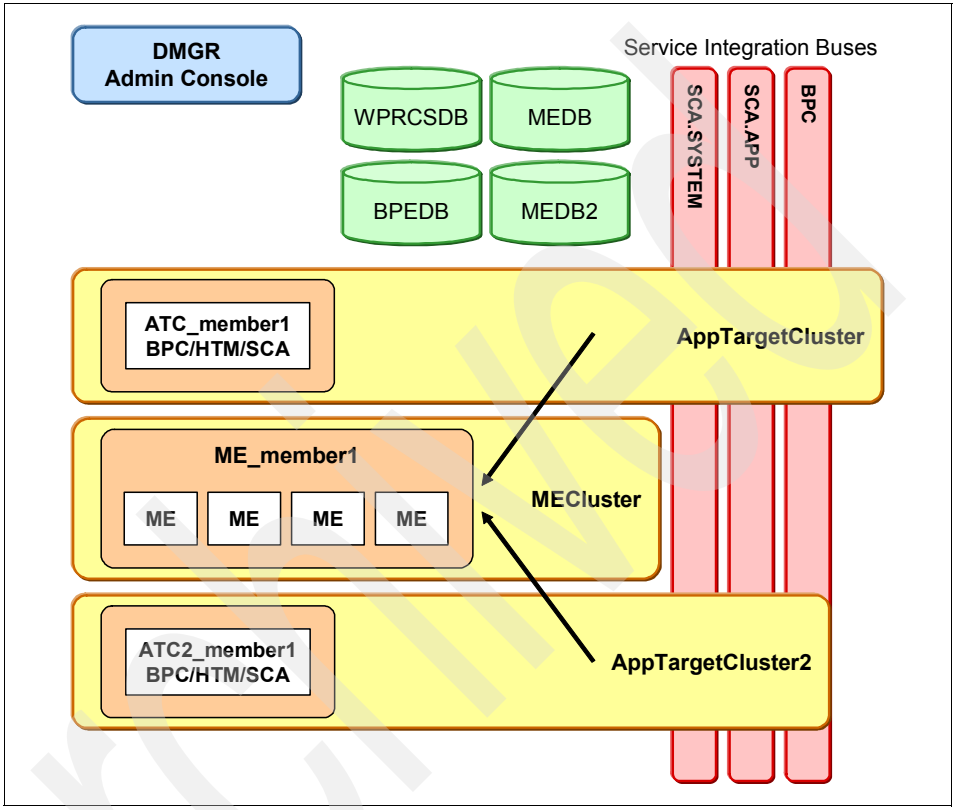


Figure 11-74 Multiple application targets with a single ME cluster

It may be tempting to create this topology for a number of reasons:

- If deployed applications are not making heavy use of the ME, either because they are micro-flows or because the applications use synchronous invocation
- If there is a hardware restriction that limits the number of JVMs

This pattern has limited scope for growth and limited applicability. Consider the following when choosing this topology:

- Will future applications be micro-flows or use synchronous invocation styles?
- Versioning of existing applications will increase the number of destinations on an ME, especially if older versions are left on the system for some time. How will this impact you over time?
- Will sharing an ME be acceptable for isolation requirements?
- How will the configuration of clusters for future applications be determined?

There are additional problems associated with this pattern, including:

- The length of time that the ME cluster will take to start with a large amount of destinations
- The fact that an ME failover affects every application

Note: A more complete discussion on expanding your topology can be found in this developerWorks article:

http://www.ibm.com/developerworks/websphere/library/techarticles/0901_herness/0901_herness.html

12

Monitoring a production topology

This chapter introduces the IBM Tivoli monitoring products that allow you to visually monitor, manage, and control service-oriented architecture (SOA) solutions. This chapter illustrates how IBM Tivoli Monitoring and IBM Tivoli Composite Application Manager (ITCAM) for SOA can be used to gain real-time visibility into the performance aspects of your business. It also shows a variety of performance reports that are provided by the Performance Manager component of WebSphere Business Services Fabric.

To demonstrate the capabilities of these products, the Web services deployed for the vehicle loan application are enabled for service management and viewed through Tivoli Enterprise Portal. Refer to Chapter 2, “Sample business application scenario used in topologies” on page 23, for a discussion of the application.

The chapter is organized into the following sections:

12.1 Prerequisite monitoring software

The following products from the IBM Tivoli portfolio were used to monitor the vehicle loan application:

- IBM Tivoli Composite Application Manager (ITCAM) for SOA V7.1
- IBM Tivoli Monitoring V6.2

12.1.1 ITCAM for SOA V7.1

ITCAM for SOA V7.1 provides monitoring and management of services and mediations in a service-oriented architecture environment. It can monitor, manage, and control the Web services layer of the IT architecture. It can also function in the lower layers of the infrastructure to identify performance problems and the specific services causing such problems. Some of the key features of ITCAM for SOA include:

- Visualization of the service-to-service topology with workspace linking to and from Tivoli Enterprise Portal.
- Monitoring by Web service requester identity. Metric data is aggregated by each service port and operation name pair and can be used to identify who sent the request.
- Expanded support for more application servers including BEA Weblogic, Microsoft .NET Framework V3, and DataPower®.
- Support for the promotion of Service Component Architecture (SCA) mediation primitives in IBM WebSphere Enterprise Services Bus and WebSphere Process Server.

ITCAM for SOA consists of the following logical components:

- A Web services data collector that acts as the JAX-RPC handler and intercepts Web service calls to collect statistical information and write to a log file.
- Tivoli Enterprise Monitoring Agent that collects information from all data collectors on a monitored machine and forwards them to Tivoli Enterprise Monitoring Server.
- An Eclipse-based viewer that processes log files generated by the data collector. It generates visual representations of metrics from the monitored services. The viewer is available as a browser-based or desktop application.
- Mediation SCA tools that enable partial monitoring of SCA within WebSphere Process Server and WebSphere Enterprise Services Bus.

12.1.2 IBM Tivoli Monitoring V6.2

IBM Tivoli Monitoring is the primary product that provides the base infrastructure of management and monitoring. The core components of this product are:

- Tivoli Enterprise Monitoring Server (TEMS)

  The monitoring server is central to the monitoring solution. TEMS is responsible for collecting alerts and performance and availability data from agents. It also tracks the heartbeat request interval for all configured TEMS agents. When installing TEMS, the primary Monitoring Server is configured as a HUB and subsequent servers are configured as remote to the Hub, thus providing greater scalability and centralized collection and analysis of the data.

- Tivoli Enterprise Portal Server (TEPS)

  This is a presentation layer and database repository for all graphical representation of monitoring data. TEPS is used for retrieval, manipulation, analysis, and formatting of data. It manages this access through user workspace consoles. TEPS maintains a persistent connection to the TEMS Hub and can be considered a logical gateway between the TEMS Hub and TEP client.

- Tivoli Enterprise Portal (TEP) clients

  This is the user interface to the monitoring solution. It can be accessed using either a TEP Desktop (Java) client or a TEP browser client that loads a Java applet in a Web browser.

- Tivoli Enterprise Monitoring Agents (TEMA)

  The agents are installed on the target systems (systems requiring monitoring and data collection). The agents gather and distribute metrics to the monitoring servers, including initiating the heartbeat status.

12.1.3 Additional information

Additional information about ITCAM for SOA, IBM Tivoli Monitoring, and other monitoring products can be found in the information centers at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?topic=/com.ibm.itcamsoa.doc/welcome.htm

http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/index.jsp?toc=/com.ibm.itm.doc/toc.xml

12.2 Monitoring the SOA environment

Solution and infrastructure management is a broad discipline. Therefore, the focus of this chapter is appropriately constrained to the services layer of the IBM SOA Foundation Reference Architecture shown in Figure 12-1.

> **Note:** The IBM Tivoli portfolio provides service management capabilities across all layers of the SOA Foundation Reference Architecture. This chapter focuses specifically on ITCAM for SOA in the services layer.

Refer to the following IBM Redbooks publications for a more detailed discussion of the subject:

- *Best Practices for SOA Management*, REDP-4233
- *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240

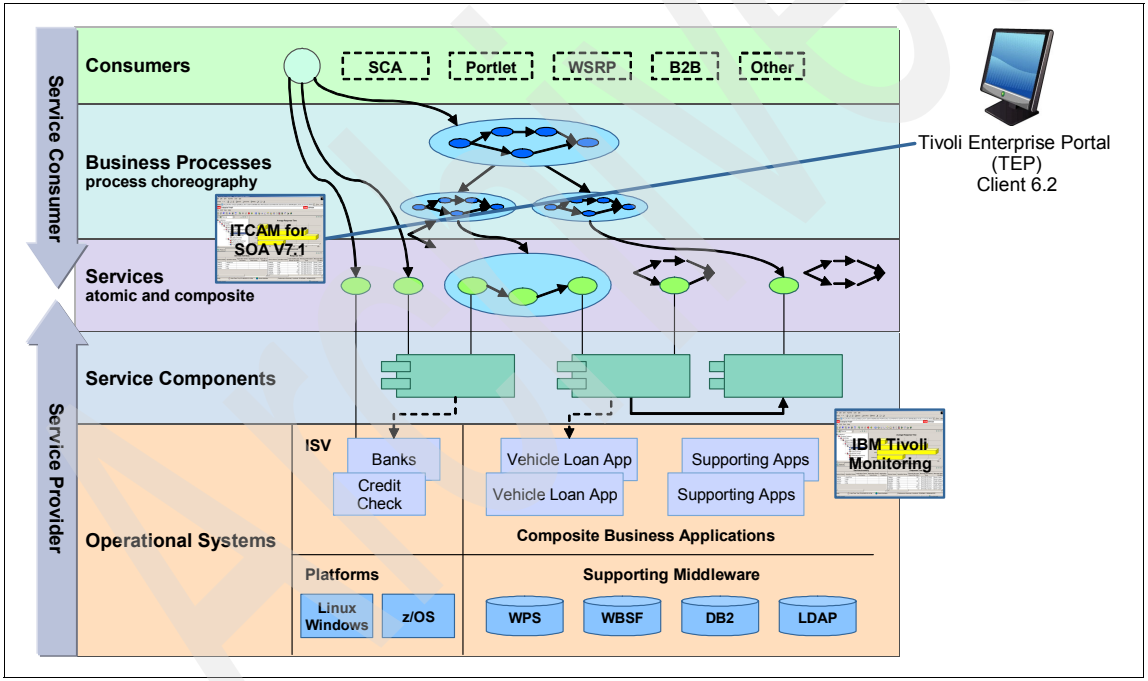


Figure 12-1 Monitoring in the SOA Foundation Reference Architecture

Deployment of a service-oriented solution in the production topology introduces specific requirements to monitoring operational and transactional characteristics of composite applications, business services, and the underlying infrastructure. Services (realized as Web services in the sample application) must be managed

in the same manner as resources in the Information Technology (IT) domain. Services are therefore subject to commonly used operational semantics such as average response time, mean time to failure, and other measurable criteria.

ITSOBank monitoring infrastructure

The ITSOBank deployment and monitoring infrastructure consists of two nodes, as seen in Figure 12-2:

- A clustered application server running the vehicle loan application. This is the managed node and contains the data collector, the Tivoli Enterprise Management Agent V6.2, and ITCAM for SOA.
- A monitoring server running Tivoli Enterprise Monitoring Server V6.2, Tivoli Enterprise Portal Server V6.2, and the desktop version of the Tivoli Enterprise Portal Client V6.2.

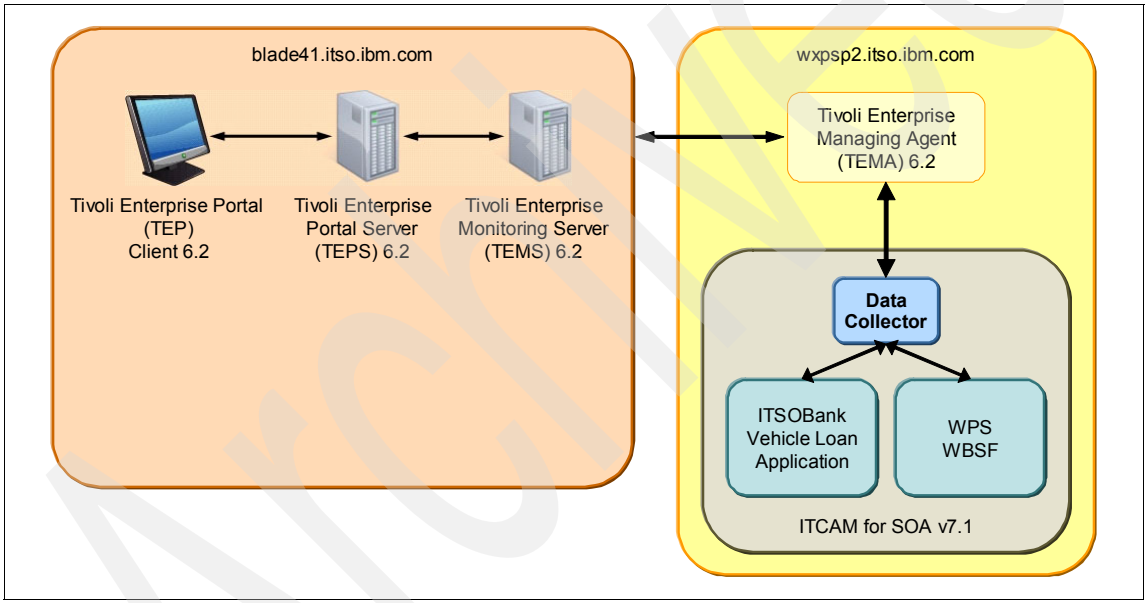


Figure 12-2 ITSOBank monitoring deployment infrastructure

With this configuration, Web service interactions and other resources on the managed node can be monitored in a centralized console. The installed components are listed in Table 12-1.

Table 12-1 Installed monitoring components for ITSOBank

Installation node	Installation components
Monitoring server	► IBM DB2 UDB Enterprise Edition V9.5 ► IBM Tivoli Enterprise Monitoring Server V6.2: – Tivoli Enterprise Monitoring Server (TEMS) – Tivoli Enterprise Portal Server (TEPS) – TEPS Desktop Client – Tivoli Enterprise Monitoring Agents • Tivoli Enterprise Monitoring Agent Framework • Warehouse Proxy • Monitoring Agent for OS • Universal Agent ► Tivoli Monitoring Application Support for ITCAM for SOA
Monitored server	► ITCAM for SOA V7.1 ► Tivoli Enterprise Monitoring Agents: Data Collector ► WebSphere Process Server V6.2 ► WebSphere Business Services Fabric V6.2 ► IBM DB2 UDB Enterprise Edition V9.5
Developer workstation	ITCAM for SOA Tools - Eclipse-based Web services Navigator

12.3 Monitoring the vehicle loan process

The vehicle loan application is implemented as a Composite Business Application (CBA) for deployment on WebSphere Business Services Fabric. A CBA is a collection of related and integrated business services that provide a specific business solution and support multiple business processes built on SOA. To test the CBA we use a BPEL process to invoke the loan process, which causes the execution of the WebSphere Business Services Fabric Dynamic Assembler (DA) SCA component. The DA represents the service endpoints that are associated with business policies. At runtime the DA uses policies and context data in the request to determine the appropriate service endpoints that will be invoked to fulfill the request. The data collector captures messages that flow through these DA components.

In addition to the DA, other elements of interest to the monitoring solution include the database, operating systems, transactions, and many more. Each of these elements can be configured accordingly the ITCAM for SOA concept of *situations*. A situation is a condition in which a set of attributes (measurements)

are tested against a threshold within any filtering rules. The following situations are predefined in ITCAM for SOA:

- Fault: Monitors for Web service faults.
- Message Arrival Critical: Triggered when Web service traffic exceeds a specified threshold.
- Message Arrival Clearing: Clears a previously triggered Message Arrival Critical situation.
- Message Size: Monitors the length (in bytes) of each message during the Web services flow.
- Response Time Warning: Triggered when the round-trip response time (in milliseconds) of a Web service request exceeds a specified threshold.
- Response Time Critical: Same monitoring characteristics as Response Time Warning but with a longer time interval setting to indicate deteriorating response times and prompt for immediate intervention.

ITCAM for SOA also provides facilities to extend these situations or design custom situations.

12.3.1 Enabling data collection for WebSphere Process Server

To enable data collection in WebSphere Process Server run this command:

```
KD4ConfigDC.bat -enable -env <e> <WPS_HOME>
```

Where *e* is the numeric designation for the target application server and *WPS_HOME* is the installation root directory for WebSphere Process Server. For this lab we ran the following command:

```
C:\ibm\ITM\TMAITM6\KD4\bin>KD4configDC.bat -enable -env 1 c:\"Program
Files"\IBM\WebSphere\ProcServer
```

The response from this command is:

```
 Configuration command =
"C:\\ibm\\ITM\\TMAITM6\\"\KD4\bin\configWASDC.BAT -enable c:\"Program
Files"\IBM\WebSphere\ProcServer
                1 file(s) copied.
Return code from configWASDC.bat = 0
```

The following application server environments are available for enablement:

- 1: IBM WebSphere Application Server (including WebSphere ESB and WebSphere Process Server)
- 2: Microsoft ASP.NET

- 3: BEA WebLogic Server
- 4: JBoss® Application Server
- 6: SAP® NetWeaver
- 7: IBM WebSphere Application Server Community Edition
- 8: IBM WebSphere DataPower
- 10: IBM WebSphere Message Broker

12.3.2 Discovering services uses ITCAM for SOA

The IT operator responsible for monitoring the infrastructure usually has a complete list of all services in the runtime environment. However, if such a list is not available one can be obtained from the ITCAM for SOA service inventory. This inventory is compiled by the agents listening in the monitored server and can later be used to create situations.

The high-level tasks for service discovery are:

1. Launch the Tivoli Enterprise Portal Client.
2. Ensure that the Services Management Agent is started.
3. Generate traffic by running the vehicle loan application.
4. Review the services inventory.

Launch the Tivoli Enterprise Portal Client

To to this:

1. Click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Tivoli Enterprise Portal**.
2. Log into the client using the ID `sysadmin`. The default password is `sysadmin`.

Ensure the Services Management Agent is started

To do this:

1. In the Navigator panel expand **Enterprise** → → **Windows Systems**. This reveals a list of nodes participating in the monitoring infrastructure.

2. Expand the monitored server then Services Management Agent (Figure 12-3).

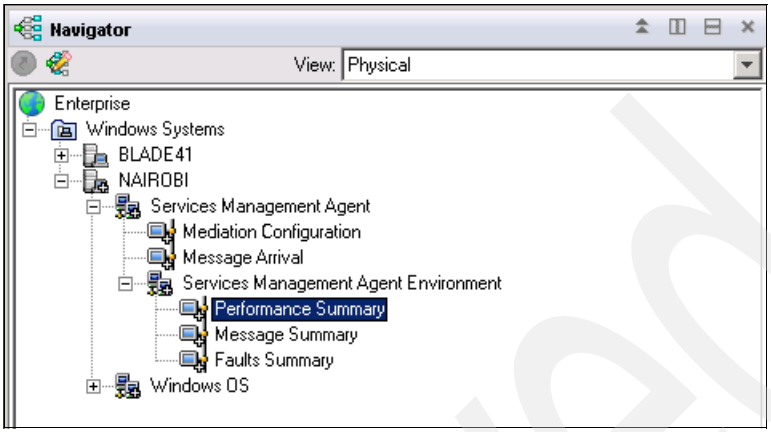


Figure 12-3 Tivoli Enterprise Portal: Navigator panel

Generate traffic by running the vehicle loan application

To generate traffic, invoke the vehicle loan application as discussed in 14.4.6, “Running the vehicle loan process application” on page 430.

Review the services inventory

Once traffic is generated the Services Inventory panel is populated with multiple columns giving details on the various services and operations and operation statistics such as message count, round trip time, elapsed time, and so forth. Figure 12-4 represents a truncated view of the Service Inventory panel.

Services Inventory

Service Type	Service Port Name (Unicode)	Service Port Name Type	Service Port Namespace (Unicode)	Operation Name (Unicode)
Requester	TempPort	WSDL_Port_Name		TempOperation
Requester	CreditCheckImport	SCA_Component_Name	ITSO	checkCustomerCredit
Provider	CreditCheckImport	SCA_Component_Name	ITSO	checkCustomerCredit
Requester	LoanProcessContextInjector	SCA_Component_Name	ITSO	processLoan
Provider	LoanProcessContextInjector	SCA_Component_Name	ITSO	processLoan
Requester	LoanProviderContextExtractor	SCA_Component_Name	ITSO	extractContext
Provider	LoanProviderContextExtractor	SCA_Component_Name	ITSO	extractContext
Requester	LoanProviderDA	SCA_Component_Name	ITSO	provideLoan
Provider	LoanProviderDA	SCA_Component_Name	ITSO	provideLoan
Requester	NewLoanProcess	SCA_Component_Name	ITSO	processLoan
Provider	NewLoanProcess	SCA_Component_Name	ITSO	processLoan
Requester	RatingImport	SCA_Component_Name	ITSO	performRiskRating
Provider	RatingImport	SCA_Component_Name	ITSO	performRiskRating
Requester	TestLoanProcess	SCA_Component_Name	ITSO	processLoan
Provider	TestLoanProcess	SCA_Component_Name	ITSO	processLoan

Figure 12-4 Services discovered in a sample execution of the application

12.3.3 Performance metrics

The Tivoli Enterprise Portal contains workspaces that display performance metrics for each monitored server. The graph in Figure 12-5. shows the average response times for the various Web service operations invoked in the vehicle loan application.

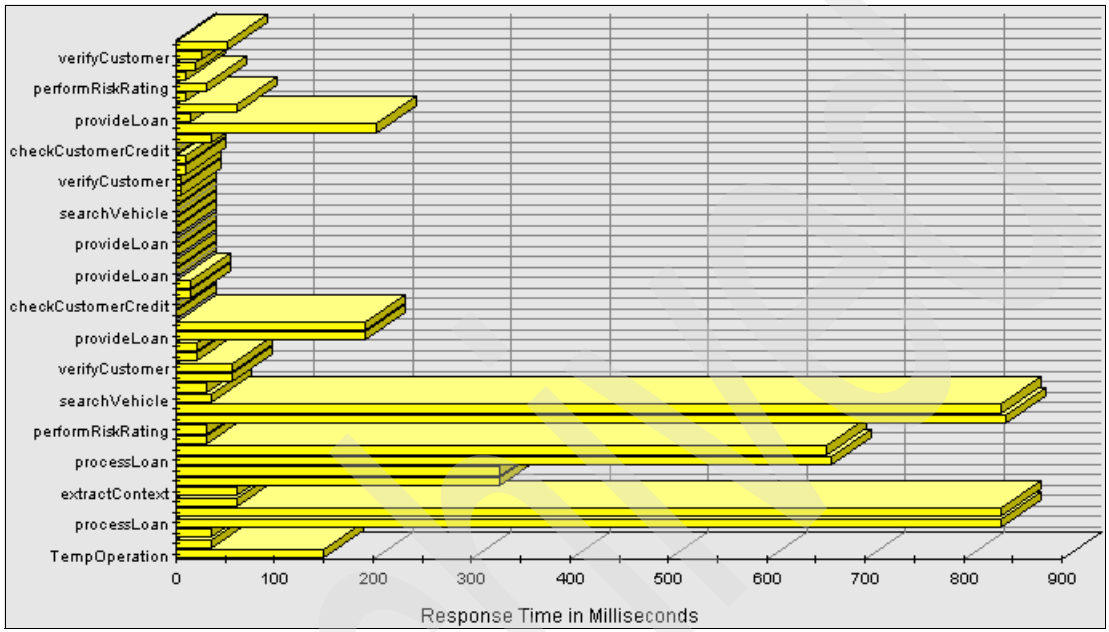


Figure 12-5 Average response time by operation.

By visual inspection it is clear that searchVehicle and processLoan operations take the longest time to execute and should therefore be prime candidates for further performance analysis.

The number of messages exchanged between Web service operations during a specific monitoring period is shown in Figure 12-6. We sent six sample requests to the CBA. Figure 12-6 provides a graphical display of aggregate metrics captured during the execution of the vehicle CBA.

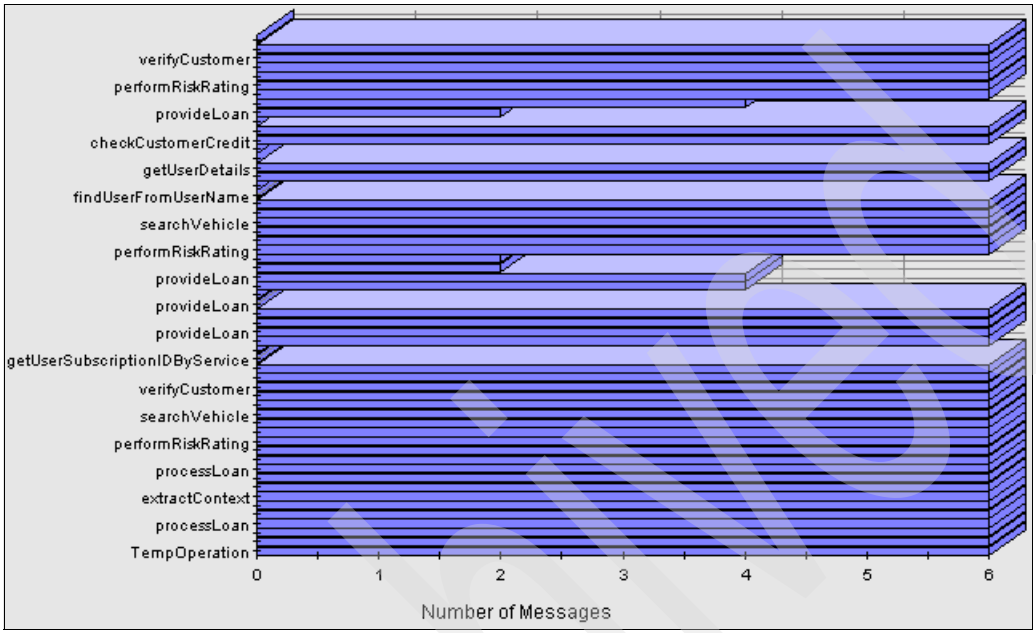


Figure 12-6 Message volume per Web service operation

12.4 WebSphere Business Services Fabric Performance Manager

The Performance Manager component of WebSphere Business Services Fabric provides metrics on business context transactions. Some of its core capabilities include:

- Real-time monitoring of aggregate or individual level traffic including faults, transaction volumes, and availability
- Pre-built visibility and optimization services to monitor business service performance
- The ability to trace individual transactions for audit, security, and compliance requirements

This section describes the three metric reports provided by the Performance Manager:

- Service Invocation Summary report
- Service Performance report
- Service Utilization report

12.4.1 Service Invocation Summary report

The Invocation Summary report displays the response times for Web services and endpoints that are invoked in the context of a single business transaction.

To access the report:

1. In the WebSphere Business Services Fabric administration console, click **Performance Manager** → **Service Invocation Summary**. The Service Invocation Summary page opens, as shown in Figure 12-7.

Service Invocation Selection

View 10 rows at a time

57 rows

Transaction ID	Service Name	Invoked On
ID:wxpsp2_010408-3386-1236718001952-0:9	LoanProcessBS	Mar 10, 2009 3:24:02 PM
ID:wxpsp2_010408-3386-1236718001952-0:11	LoanProcessBS	Mar 10, 2009 3:25:09 PM
ID:wxpsp2_010408-3386-1236718001952-0:13	LoanProcessBS	Mar 10, 2009 3:25:47 PM
ID:wxpsp2_010408-3386-1236718001952-0:15	LoanProcessBS	Mar 10, 2009 3:26:18 PM
ID:wxpsp2_010408-3386-1236718001952-0:17	LoanProcessBS	Mar 10, 2009 3:34:03 PM
ID:wxpsp2_010408-3386-1236718001952-0:19	LoanProcessBS	Mar 10, 2009 3:34:24 PM
ID:wxpsp2_010408-3386-1236718001952-0:21	LoanProcessBS	Mar 10, 2009 3:34:39 PM
ID:wxpsp2_010408-3386-1236718001952-0:23	LoanProcessBS	Mar 10, 2009 3:35:02 PM
ID:wxpsp2_010408-3386-1236718001952-0:25	LoanProcessBS	Mar 10, 2009 4:16:07 PM
ID:wxpsp2_010408-3386-1236718001952-0:27	LoanProcessBS	Mar 10, 2009 4:16:32 PM

Figure 12-7 Service Invocation Summary for the ITSOBank loan application

2. Under the heading Transaction ID, click a transaction to view its respective invocation report.

12.4.2 Service Performance report

This report provides performance analysis using attributes such as the total number of transactions for a business service, average response time of a service or endpoint, and the number of failed transactions.

To access the report, in the WebSphere Business Services Fabric administration console, click **Performance Manager** → **Service Performance**. A sample report for the vehicle loan application opens, as shown in Figure 12-8.

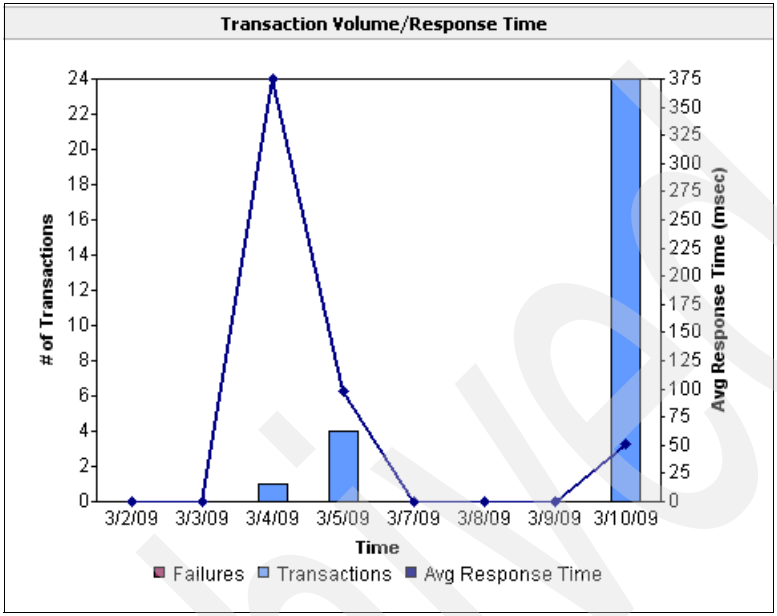


Figure 12-8 Performance report for the high risk loan provider endpoint

12.4.3 Service Utilization report

The Service Utilization report shows the frequency with which an organization has invoked (utilized) a business service in a given time period. To access the report:

1. In the WebSphere Business Services Fabric administration console, click **Performance Manager** → **Service Utilization**.
2. The Service Utilization page opens with a Filter dialog where you can enter filter parameters such as a business service of interest and a utilization time interval.
3. Click **Report** to generate the report.

For more details on the Performance Manager visit the WebSphere Business Services Fabric V6.2 Information Center found at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.ws.fabric.tools.doc/pm/concept/c_pmoverview.html

13

Using Business Space powered by WebSphere and Lotus Forms Client

This chapter describes how to configure and use Business Space powered by WebSphere. Business Space provides a customizable and collaborative environment for monitoring, reviewing, and administering common business processes, such as human task flows, modeling, and performance indicators. It is a browser-based graphical user interface that lets business users interact with content from products in the WebSphere Business Process Management portfolio. The business spaces that you create are collections of related Web content that provide you with insight into your business and the capability to react to changes in it.

Lotus Forms consists of a design tool for creating user interfaces (forms) for human tasks and runtime components for viewing the forms. This chapter demonstrates how Lotus Forms can be integrated with WebSphere Integration Developer to generate a user interface for human tasks for later deployment on WebSphere Process Server. It shows how to use the Forms Designer to create and modify the properties of a form.

This chapter contains the following sections:

- 13.1.4, “Configure Business Space using Integrated Solutions Console” on page 359

13.1 Configuring Business Space powered by WebSphere

This section outlines the steps for setting up the Business Space database and enabling the Business Space service. If you are using the Deployment Environment Configuration wizard, the Business Space service is automatically enabled as part of that process. If you are not using the wizard, you must use the Integrated Solutions Console to complete this step.

13.1.1 Create the Business Space database

First, create a database for use by Business Space. This database can be one of the existing ones or a newly created one. We chose to add the Business Space tables to the BSPCDB database and we will use the schema name BSPACE, but you can also choose your own schema name.

Log in to the DB2 system as the instance owner, then run the following commands:

```
db2 "CREATE DATABASE BSPCDB USING CODESET UTF-8 TERRITORY en-us"
db2 connect to BSPCDB USER db2inst1 using 'dbpass'
db2 connect reset
```

Now that the database is created successfully, you can create the Business Space database tables.

13.1.2 Create the Business Space database tables

To create the tables in Business Space database:

1. In the database machine, log in as the instance owner for the BSPCDB database. You must transfer the file `createTable_BusinessSpace.sql` from the deployment manger to this machine. In the deployment manager this SQL file can be found in `/opt/ibm/WebSphere/ProcServer/dbscripts/BusinessSpace/DB2`.
2. Edit the file `createTable_BusinessSpace.sql` and change the value of @SCHEMA@ to `BSPACE`. Change the value of @TSDIR@ to a suitable location (for example, `/home/db2inst1/db2inst1/NODE0000`) or just remove it altogether. Save your changes.

3. Run the SQL commands against the BSPCDB database:

```
db2 connect to BSPCDB
db2 -tf createTable_BusinessSpace.sql
db2 connect reset
```

Once the database setup is complete, you can log off the database server.

13.1.3 Configure Business Space as part of Deployment Environment wizard

For WebSphere Process Server and WebSphere Enterprise Service Bus runtime environments, the Business Space service and the Representational State Transfer (REST) service for Business Space widgets are automatically configured as part of the Deployment Environment Configuration wizard. You can decide which REST services to configure, as shown in Figure 13-1.

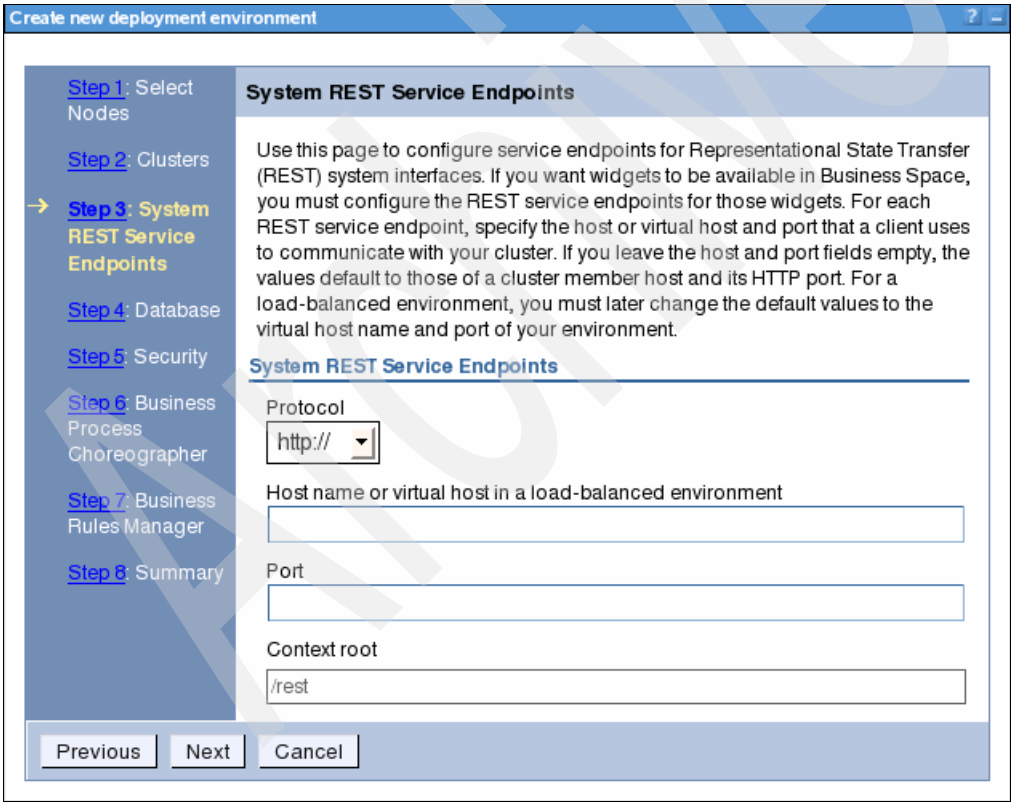


Figure 13-1 System REST Service Endpoints for Business Space window

13.1.4 Configure Business Space using Integrated Solutions Console

If you did not use the Deployment Environment wizard to set up your environment, follow these steps to enable the Business Space service:

1. Ensure that the Integrated Solutions Console is running.
2. In the navigation pane click **Servers** → **Application servers** or **Servers** → **Clusters**.
3. Select the name of your server or cluster target.
4. On the Configuration page, under Business Integration, click **Business Space Configuration**. The Business Space Configuration page appears. If Business Space has already been configured, you can view this page but cannot edit the fields (Figure 13-2).

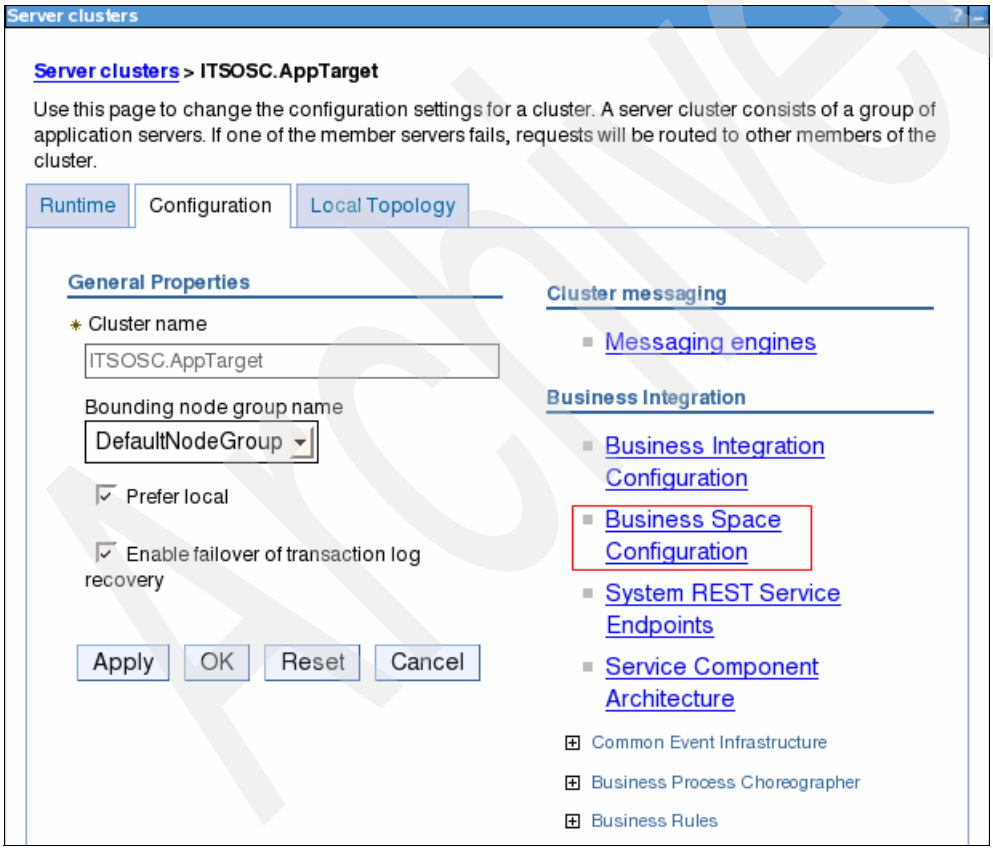


Figure 13-2 Business Space Configuration window

5. Select the **Install Business Space service** check box.

6. In the Database schema name box, type the name of the database schema that you want to use for Business Space, as shown in Figure 13-3.

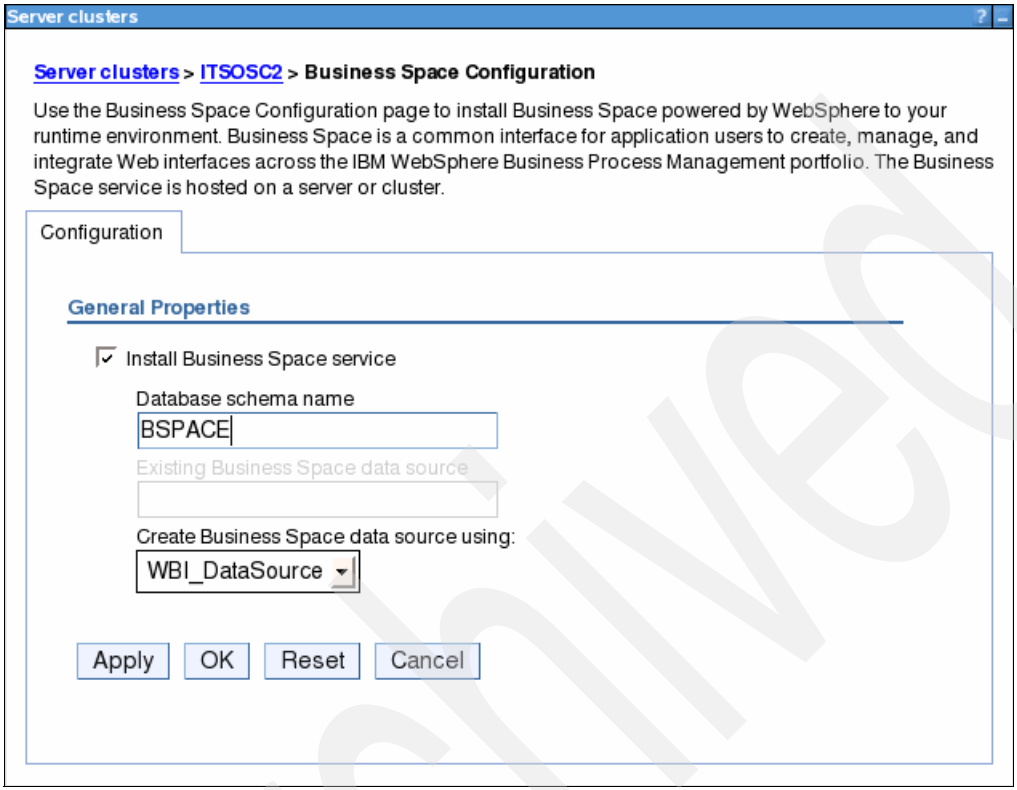


Figure 13-3 Install Business Space service window

7. If no data source is designated in the Existing Business Space data source field, go to **Create Business Space data source using** and select a data source that connects to the database that you want to use with Business Space.

8. Designating a data source under the Create Business Space data source using drop-down box creates a new data source for Business Space with a JNDI name of jdbc/bpm/BusinessSpace that is modelled on the data source that you selected.

> **Note:** If you do not see an existing data source that you want to use, you must cancel the Business Space Configuration page, set up the database and the data source that you want to use, and then restart the Business Space Configuration page to complete the configuration.

9. If you have not yet set up the database and the data source, cancel the Business Space configuration page and follow these steps:
   a. Create the database using the database product software.
   b. Use the administrative console to configure the JDBC provider.
   c. Use the administrative console to create a data source with the JNDI name of jdbc/bpm/BusinessSpace at the server or cluster scope, depending on what you selected in Step 2.
   d. Go back to the Business Space Configuration page to select a data source.

   Click **OK**.
10. Save the configuration.

Business Space Manager

The Business Space can be viewed using a browser with the following URL:

```
http://<hostname>:9080/BusinessSpace
```

On the Welcome to Business Space page, you can use the learning resources to tour Business Space and create your own space. Figure 13-4 shows a sample Business Space Manager window.

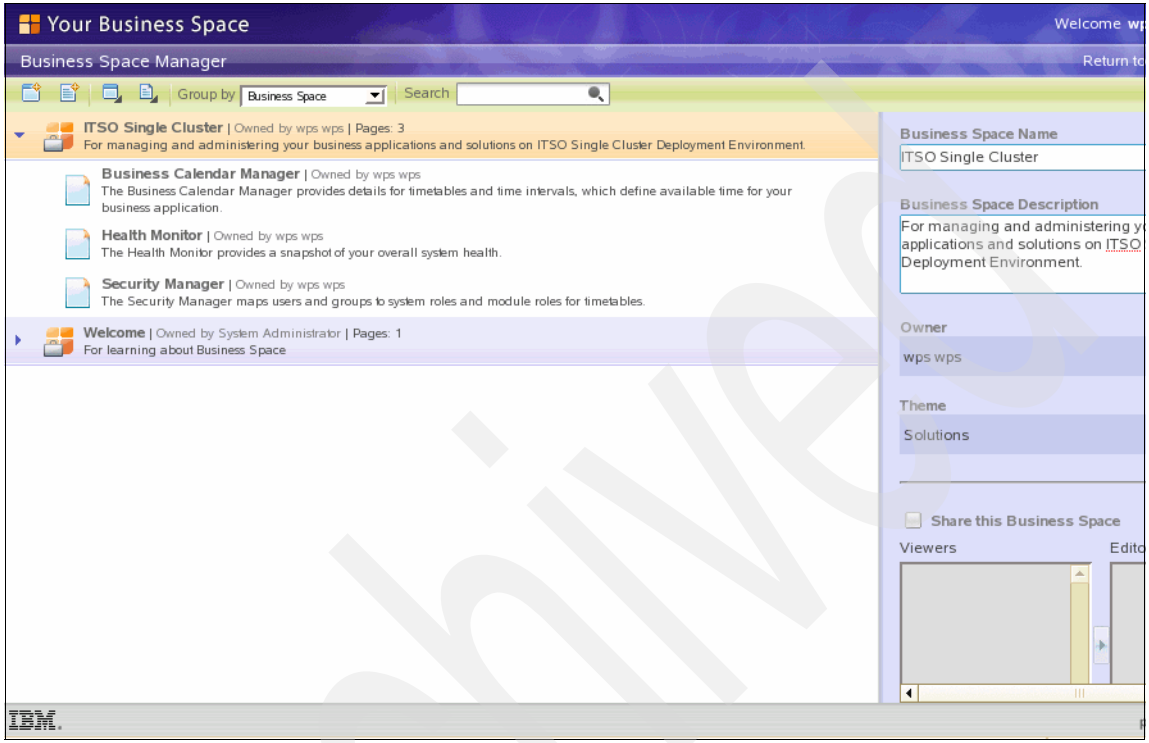


Figure 13-4 Business Space Manager window

For this example, we created a Business Space named *ITSO Single Cluster* from the Solution Management template. This template contains the following widgets:

- Business Calendar Manager
- Health Monitor
- Security Manager

The topology status is displayed as shown in Figure 13-5.

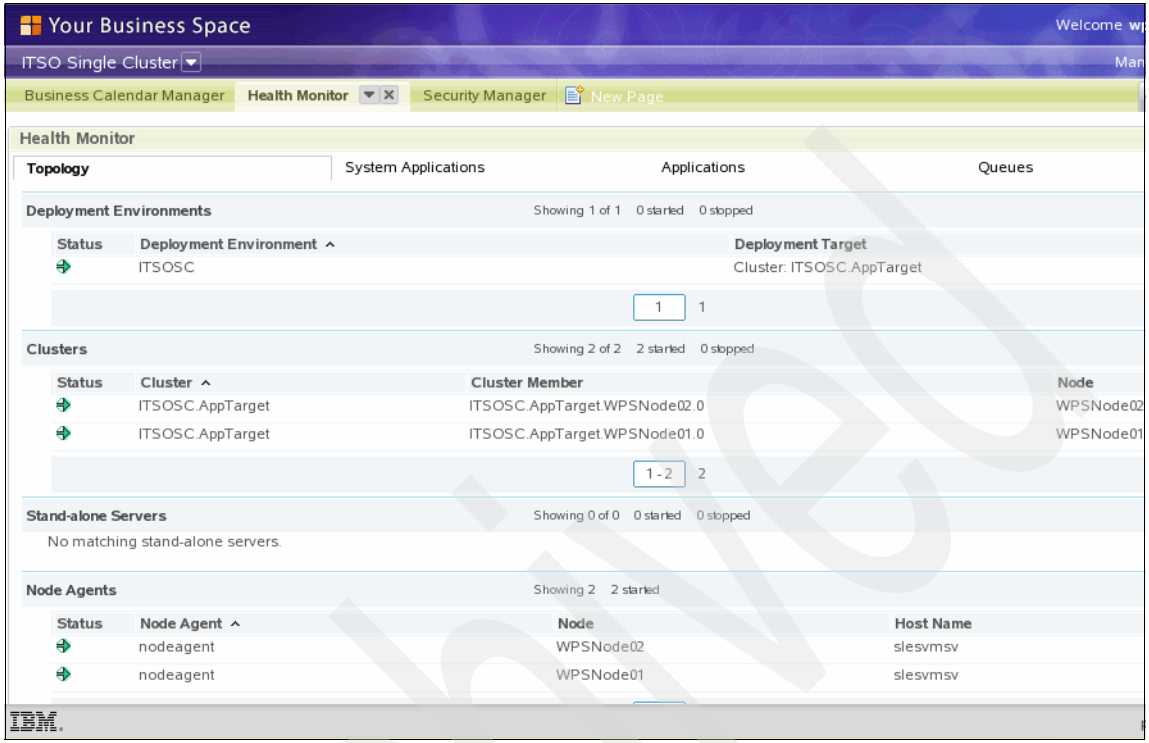


Figure 13-5 Topology from Health Monitor window

13.2 Lotus Forms

This section describes how to install and use Lotus Forms to create a form for a Human Task and then deploy the form to WebSphere Process Server. The instructions assume that you have WebSphere Integration Developer V6.2 installed and have imported the `ITSOLoanProcess_PI.zip` file under the directory `/Scenarios/WPS/HTM`. Lotus Forms consists of the following products:

- Lotus Forms Designer, which is the tool used to create the user interfaces for human tasks.
- The Lotus Forms Viewer and Server API are runtime components. The Lotus Forms Viewer can be used as a standalone application, but is more typically used as a Web browser plug-in to allow Lotus Forms to be displayed within Web pages.
- The Server API is a set of applications that are required by WebSphere Process Server to interact with deployed Lotus Forms.

Note: WebSphere Process Server only function with Lotus Forms 3.0.x.

13.2.1 Install Lotus Forms Designer

The Designer installation is a straightforward installation of the application binaries. The only options available are to install as a standalone application or integrate with an existing IBM package such as IBM WebSphere Integration Developer.

If you install Lotus Forms as a standalone application, you cannot create Human Task forms because Lotus Forms will not be available within WebSphere Integration Developer. In this example, Designer is integrated with Integration Developer.

To create Lotus Forms for a human Task:

1. On the Installation Type window, choose the **Add Designer to an existing package** radio button and click **Next**, as shown in Figure 13-6.

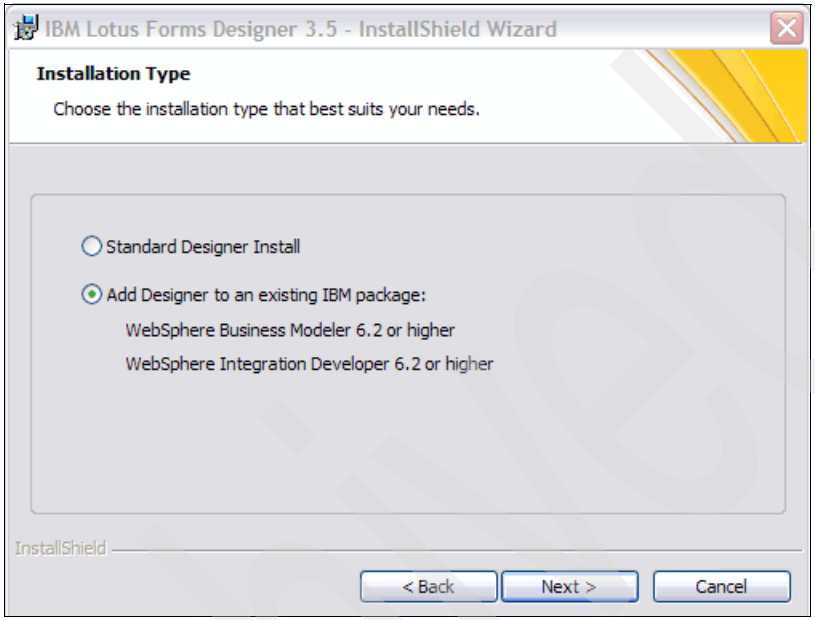


Figure 13-6 Choice of installation type

2. On the Add Designer to an existing IBM package window, click **Change** and navigate to the WebSphere Integration Developer installation directory (Figure 13-7).

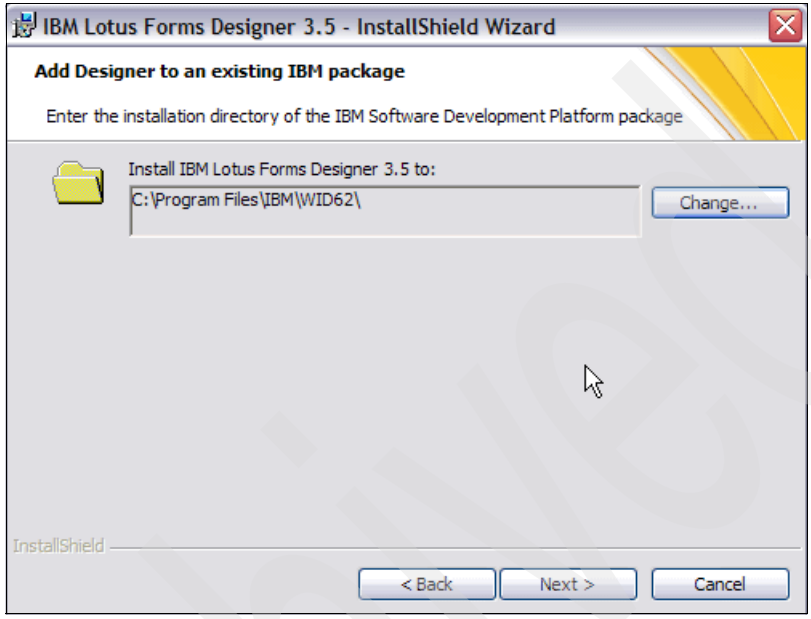


Figure 13-7 Integration with Integration Developer 6.2

Click **Next**.

3. On the Summary window, click **Install** to initiate the installation.

13.2.2 Install Lotus Forms Viewer

The Lotus Forms Viewer is a Windows install that provides a plug-in for browsers to display Lotus Forms. This installation is a standard windows install requesting just the location to install the product.

A standalone viewer is also provided but is not covered in this chapter.

> **Note:** Complete installation instructions for the viewer can be found here:
> `http://publib.boulder.ibm.com/infocenter/forms/v3r5m0/index.jsp?topic=/com.ibm.form.viewer.installing.doc/toc.html`

13.2.3 Install Lotus Forms Server API

The Lotus Forms Server API is used to handle the processing of the Forms at runtime. The Forms can be installed on Windows, AIX®, Solaris™, and Linux.

> **Note:** The Lotus Forms Server API on Solaris requires the libstdc++.so.6 library to be installed.
>
> The Lotus Forms Server API on AIX is required to have Version 8 of the XL C runtime installed. This can be found at:
>
> http://www-01.ibm.com/support/docview.wss?uid=swg21215669
>
> All installations can be run from a graphical installer, but you are required to be running from an X-Windows client do so.

- To install using the graphical installer:
  - For Windows, execute LFServer_35_API_Win32.exe and follow the prompts.
  - For Linux, AIX, and Solaris, execute ./LFServer_350_API_<platform>.bin and follow the prompts.
- To install on Linux, AIX, and Solaris without the graphical interface, execute /LFServer_350_API_<platform>.bin -console.

To perform the installation, perform the following:

1. During installation, choose the Runtime configuration. On the next panel, the installer asks you whether the Server API should be installed into an existing WebSphere Process Server deployment, as seen in Figure 13-8.

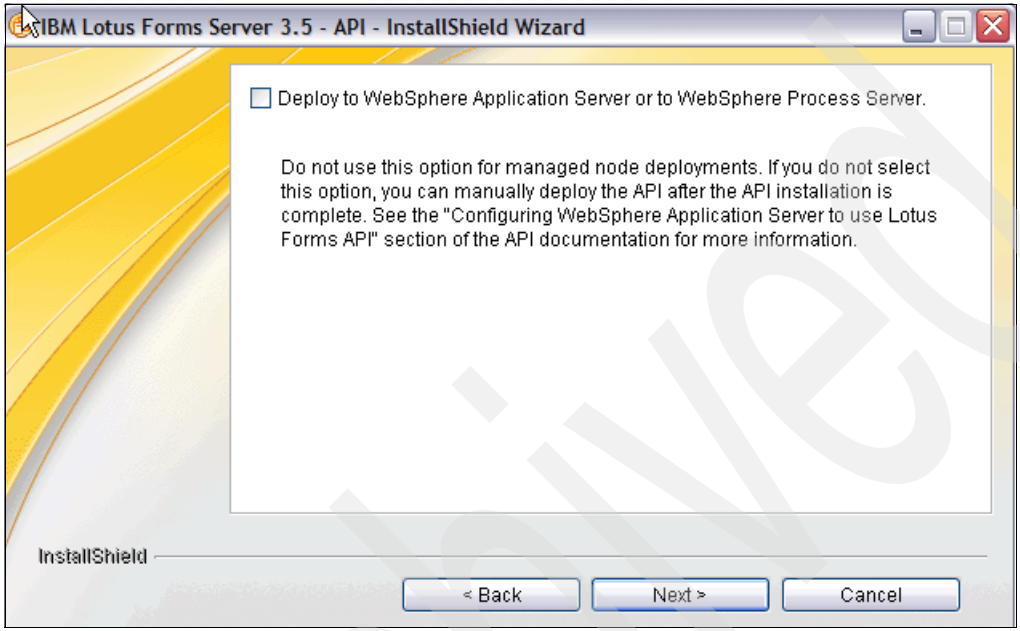


Figure 13-8 API Deployment to an existing WPS deployment

2. This can only be done, via the installer, if the WebSphere Process Server deployment is a single standalone server (Figure 13-9). If the WebSphere Process Server deployment is a clustered environment, the integration must be done manually, and not through the installer.

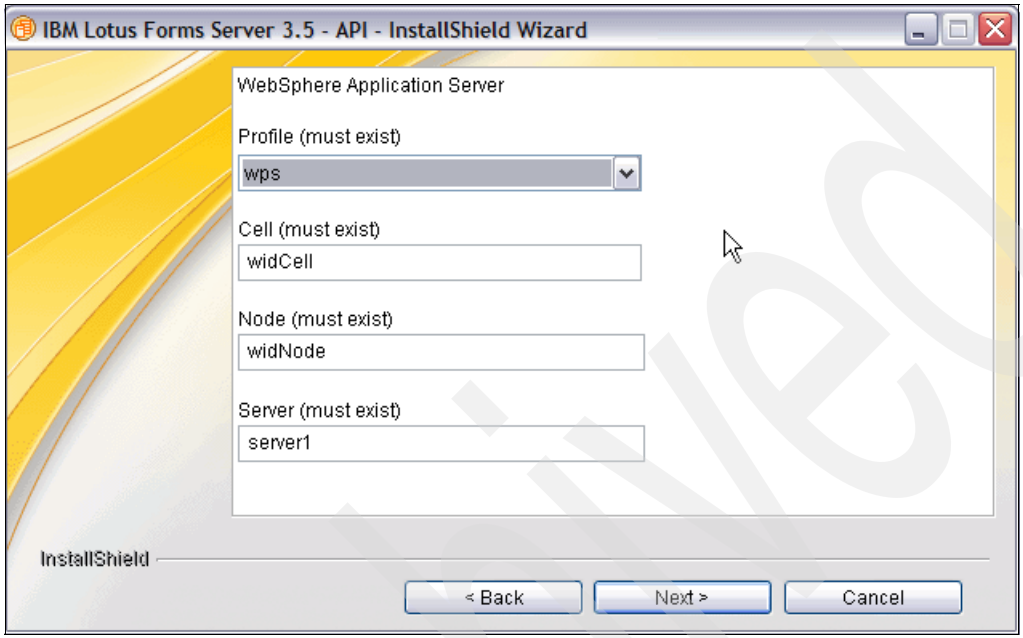


Figure 13-9 Integration into a standalone Process Server deployment

3. Click **Next** to view the Summary panel. Click **Install** to begin the installation.
4. To complete the Lotus Forms integration (in a standalone WebSphere Process Server deployment), use these manual steps. See Step 5 if you are using a clustered deployment.
   - On Windows:
     i. Update the System variable PATH environment variable to include <Server API Location>\Server\3.5\API\redist\msc32.

ii. Amend the `PureEdgeAPI.ini` file (Figure 13-10) to point to the API PureEdge libraries. If you automatically integrated with WPS, this file will already exist in the \WINDOWS directory. Otherwise, this file must be created.

> **Note:** The entry starts with '*=' to indicate the path for all PureEdge API calls. If you need to have multiple APIs installed, enter the version number instead of an asterisk (*). For example, Lotus Forms 3.0 has an API version of 7.5 and Lotus Forms 3.5 has an API version of 7.6.

```
PureEdgeAPI.ini
1  [API]
2  *=C:\IBM\Lotus Forms\Server\3.5\API\redist\msc32\PureEdge\76
```

Figure 13-10 PureEdgeAPI.ini contents

iii. Amend the `prefs.config` file if required. In most situations this file needs no amendment.

> **Note:** If you must amend the `prefs.config` file, instructions on how to do so can be found here:
>
> http://publib.boulder.ibm.com/infocenter/forms/v3r5m0/index.jsp?topic=/com.ibm.form.api.configuring.doc/api_configuring_windows_prefsconfig.html

iv. Register the COM dynamic link library (DLL) by opening a DOS shell or command prompt and executing regsvr32 <path_to_pe_file>.

> **Note:** regsvr32 "c:\Program Files\IBM\Lotus Forms\Server\3.0\API\redist\redist\msc32\pe_com.dll" will register the COM library given a default install location.

– On UNIX:

i. Update the library path for the Server API directories. These directories are `<Forms API Directory>/redist/<platform>` and `<Forms API Directory>/redist/<platform>/PureEdge/76/system`.

ii. Amend the `PureEdgeAPI.ini` file to point to the API PureEdge libraries. If you automatically integrated with WPS, this file will already exist in the `/etc` directory. Otherwise, this file will need to be created.

iii. Amend the `prefs.config` file if required. In most situations this file needs no amendment.

5. To integrate Lotus Forms in a clustered WebSphere Process Server deployment, use these manual steps:

   a. In the WebSphere Application Server Integrated Solutions Console, go to **Environment** → **WebSphere Variables**.

   b. Change the scope to the server that requires Lotus Forms, as shown in Figure 13-11.

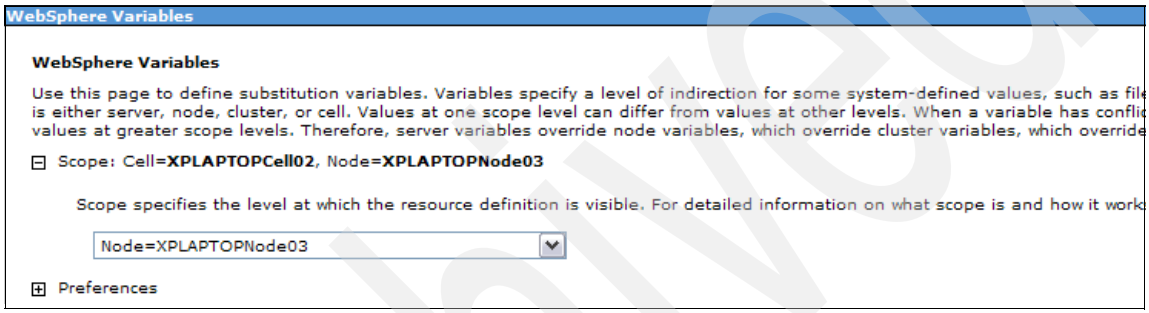


Figure 13-11 Scoped environment variables

   c. Add an environment variable named LFS_API_DIR and set its value to <Server_API_Install>\redist\msc32.

   d. Add an environment variable named LFS_API_LIB_DIR and point it to ${LFS_API_DIR}\PureEdge\76\java\classes.

   e. Add an environment variable named LFS_API_STREAM_LIB_DIR and enter its value as <Server_API_Install>\redist\java.

   f. Repeat steps b–e for each server that will host Lotus Forms.

   g. In the WebSphere Application Server Integrated Solutions Console, open **Servers** → **Application servers** → **<server_name>** → **Java and Process Management** → **Process Definition** → **Environment Entries** → **Custom Properties**, create a PATH property, and give it the value `${LFS_API_DIR};${LFS_API_DIR}/PureEdge/75/system`.

> **Note:** For AIX, the property should be named LIBPATH.
>
> For Solaris and Linux, the property should be named LD_LIBRARY_PATH.

h. In the WebSphere Integrated Solutions Console, open **Environment** → **Shared Libraries** and change the scope by selecting the node where you wish to add the shared libraries entry. This should be the same node that you selected in step b.

i. Click **New** and enter a name of LFS_API_LIB (Classic), as seen in Figure 13-12 or IFS_STREAMING (Streaming API).

j. In the CLASSPATH field add the following files:

- `${LFS_API_LIB_DIR}/pe_api.jar`
- `${LFS_API_LIB_DIR}/pe_api_native.jar`

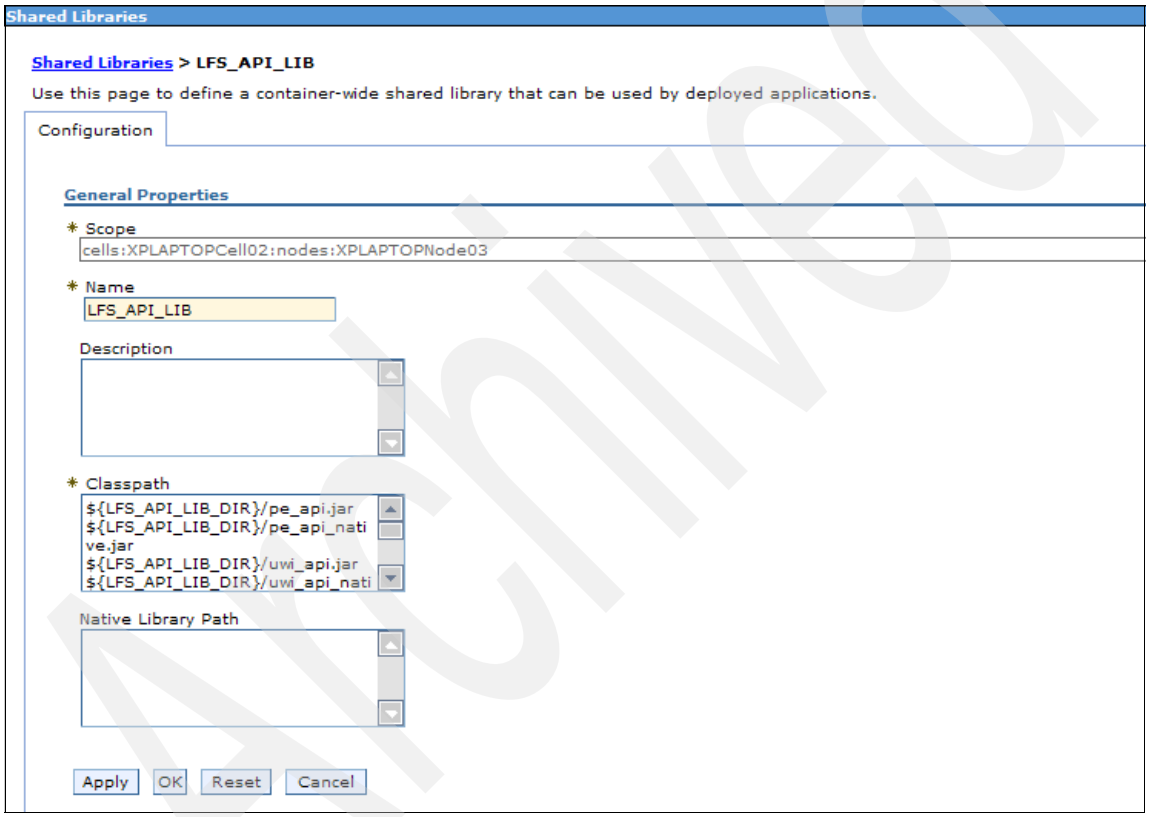


Figure 13-12 Classpath entries

Note: If you are using the Streaming API then just add the single jar file:

`${LFS_API_STREAM_LIB_DIR}/StreamingAPI.jar`

k. In the WebSphere Application Server Integrated Solutions Console, select **Servers** → **Application servers** → **<server_name>** → **Java and Process Management** → **Classloader**. Click **Add**.

l. From the Class Loader Order drop-down menu, select **Classes loaded with application class loader first**. Click **OK**.

m. Click the newly created Class Loader.

n. Click the Shared Library References link and click **Add**.

o. From the drop-down list, select the **LFS_LIB_API** entry.

p. Click **OK** and then click **Save**. Figure 13-13 shows a completed entry.

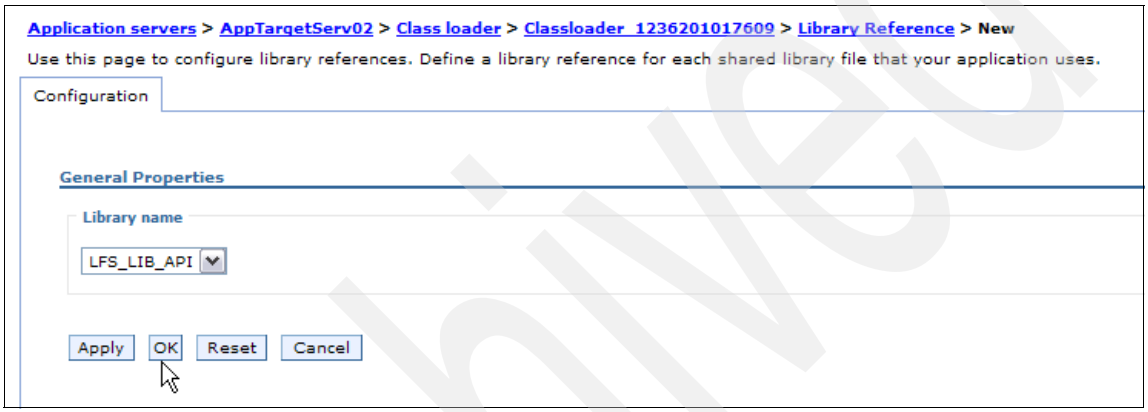


Figure 13-13 Shared libraries added to a class loader

13.2.4 Create a human task form

Once you have Lotus Forms Designer, Viewer, and API installed, you must create a form for a human task and then deploy the form to the WebSphere Process Server. These instructions assume that you have installed WebSphere Integration Developer V6.2 and have imported the PI file `ITSOLoanProcess_PI.zip` under the directory `/Scenarios/WPS/HTM`.

1. To generate a Lotus Form for a human task, open the application and navigate to the required human task (ProcessStarter). Right-click and select **Generate Human Task User Interface** from the context menu (Figure 13-14 on page 374).

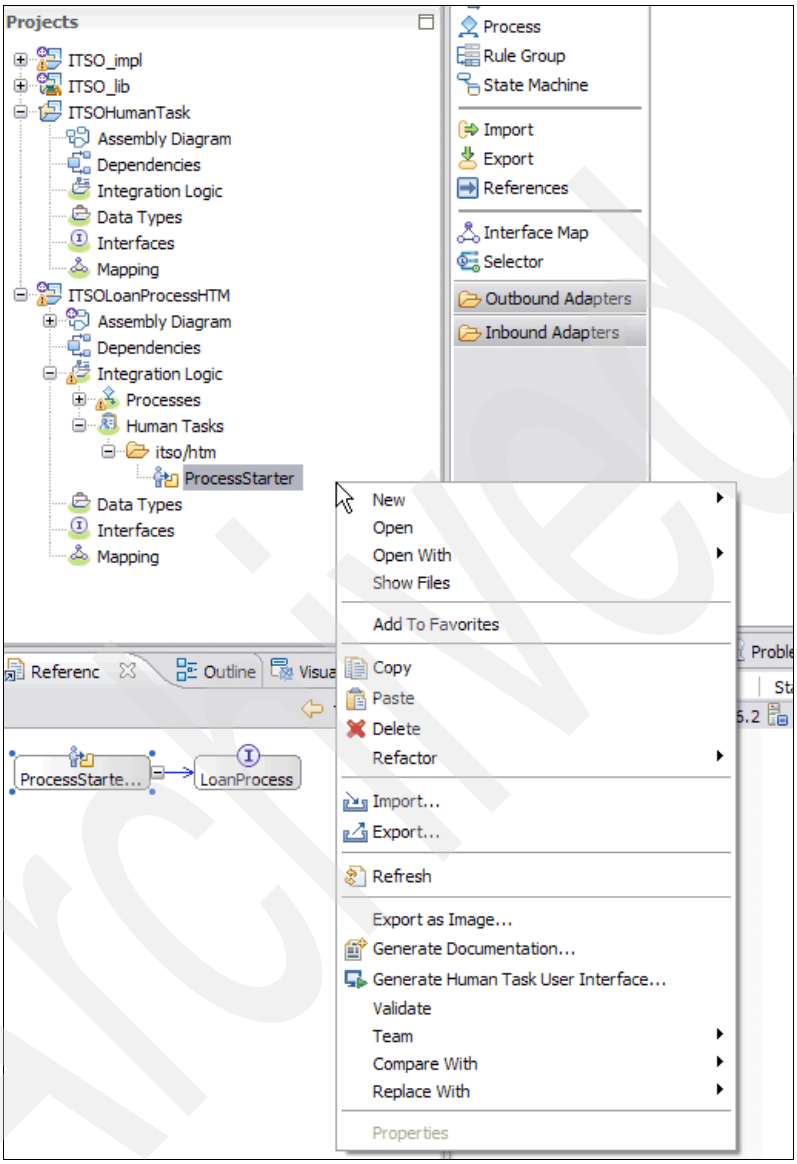


Figure 13-14 Generate a human task user interface

2. A window displays all the human tasks found. Select the **TSOLoanProcessHTM** human task (if it is not already selected) and click **Next** (Figure 13-15).

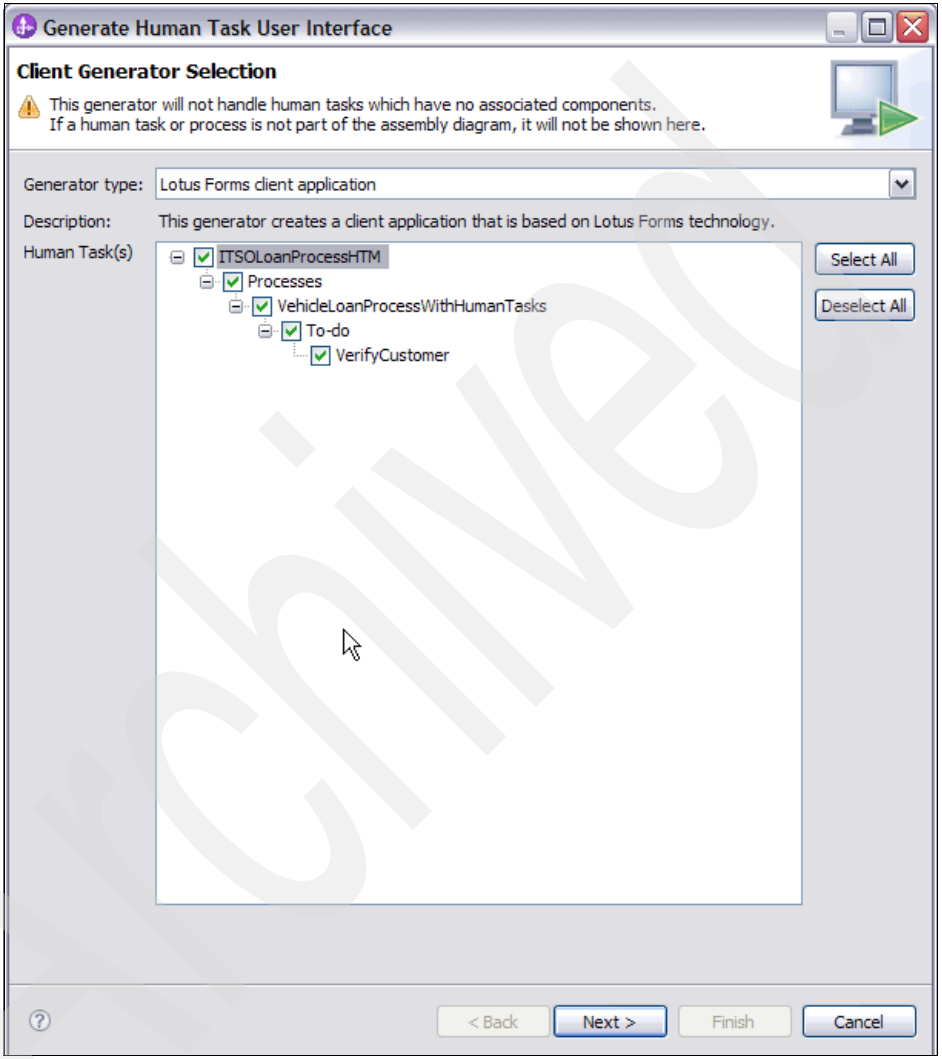


Figure 13-15 Selecting the human task

3. Enter the name of the external Web project (in our example ITSOHTaskUI) for the Lotus Forms UI.

4. Specify whether the module and the UI will reside on the same (local) or different (remote) servers and select the style, as seen in Figure 13-16. Two styles are provided:
   - IBM Style
   - Cool Blue™ Style

   Other style sheets (CSS) can be added to generate different themes. After filling out the options, click **Next**.

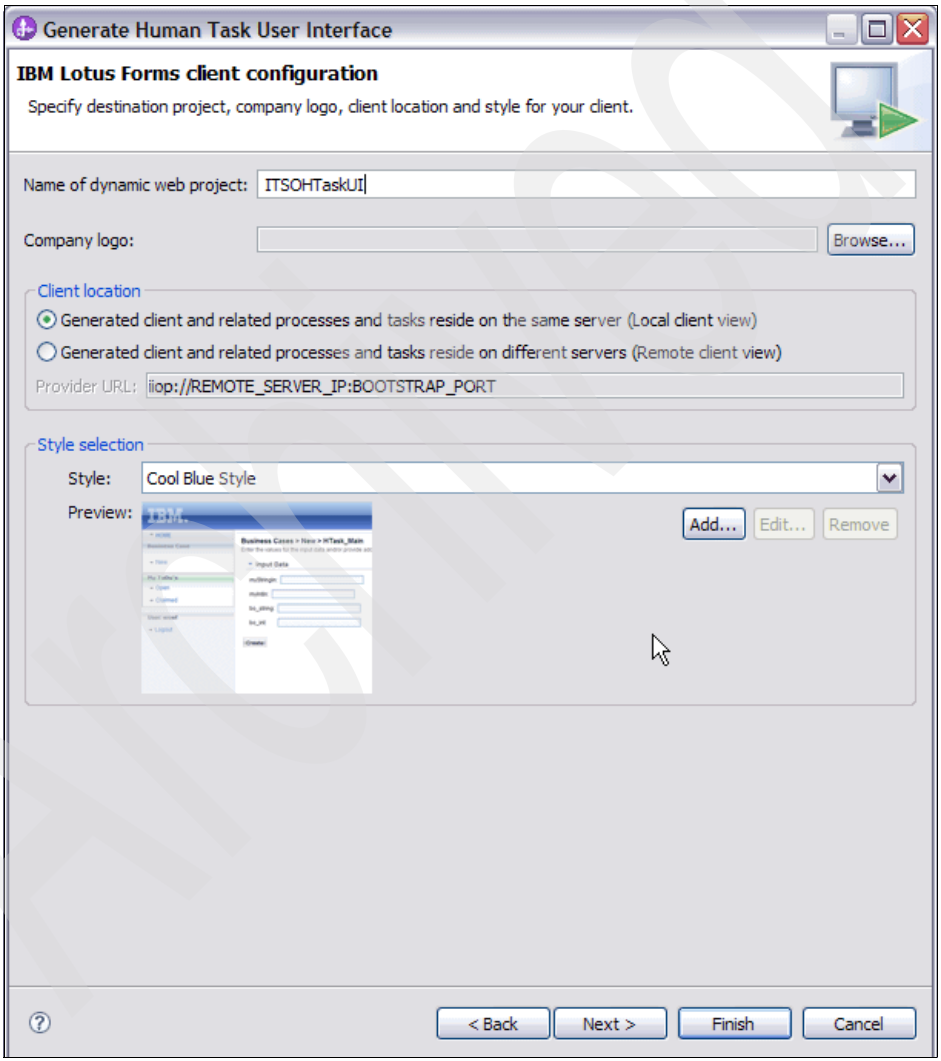


Figure 13-16 Human Task UI generation options

5. Indicate whether you want the new UI to be a Lotus form or a JSP page (Figure 13-17). Click **Finish**.

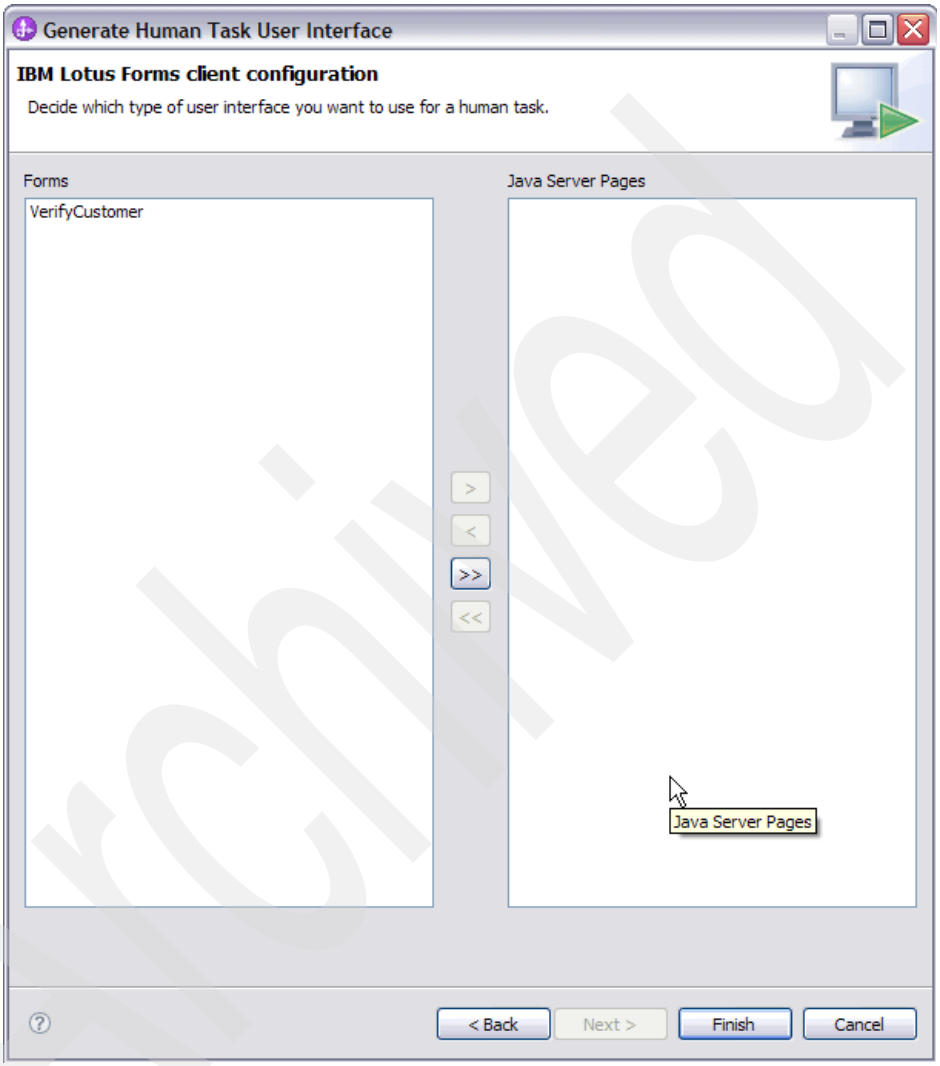


Figure 13-17 Selecting form type, JSP or Form

6. The Lotus Form UI has now been generated but has not yet been associated with the human task. To do this, double-click the human task within your BPEL process. This opens the task window, as seen in Figure 13-18. Click the add definitions icon in the User Interface section and select **Lotus Forms** from the pull-down menu.

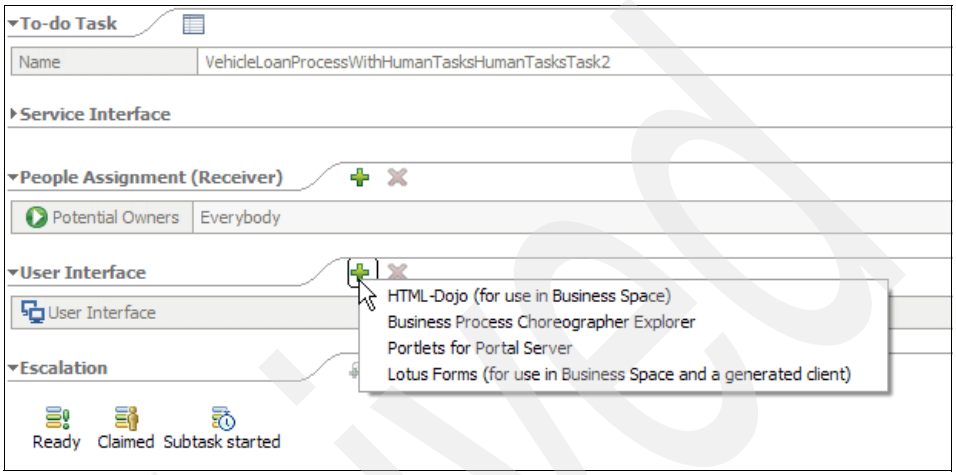


Figure 13-18 Add Lotus Forum UI

7. Click the Lotus Form that was just added and select the **Properties** view. Change Select where to store your Lotus Form from Module to **Web Project**. A new section, as shown in Figure 13-19, will appear, allowing you to browse to the newly created Web project. Open the listed Web project, and open the tree to find the `.xfdl` file, which is the generated Lotus Form. Click **OK** and then **Save**, and the form is ready to be deployed.

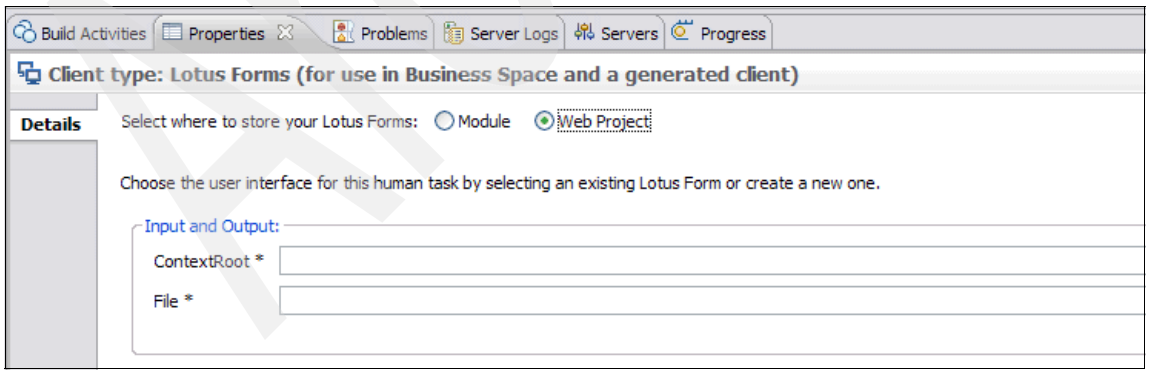


Figure 13-19 Browse to the Web Project

Once deployed, any in-flight process that executes the Human Task will appear in the newly generated Web project. Its default URL will be `http://<hostname>:<defaulthost_port>/<Web Project Name>`, as shown in Figure 13-20.

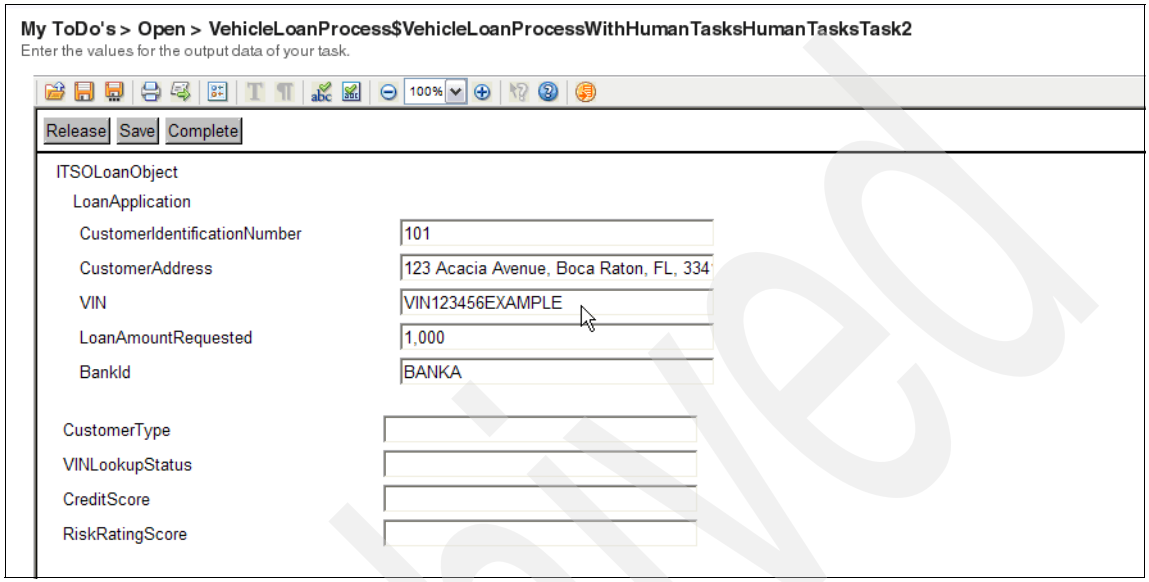


Figure 13-20 Claimed process using Lotus Form

13.2.5 Make simple form adjustments

The generic form is usually not enough for most users for production systems. Lotus Forms Designer gives you the tools to amend every aspect of the generated form.

In this short example, we make three changes to the form:

- Make the input data read-only to stop users from accidentally changing the data in mid-flight.
- Amend the format of the numeric fields to automatically add decimal and group separators.
- Amend the output fields to restrict the numeric fields where necessary.

To make these changes:

1. Once the form has been generated, switch to the Advanced Forms Designer Perspective by clicking **Window** → **Open Perspective** → **Advanced Lotus Forms Designer**.
2. In the Navigator view, expand the **ITSOHTaskUI** project and navigate to **WebContent** → **forms** → **VerifyCustomer.xfdl** and open the file. The `xfdl` file should be displayed in Designer.
3. Click the text field adjacent to the CustomerIdentificationNumber label and select the **Properties** tab. There are a myriad of options for this field, but to make it read-only, we simply change the read-only option to **on**, as seen in Figure 13-21.

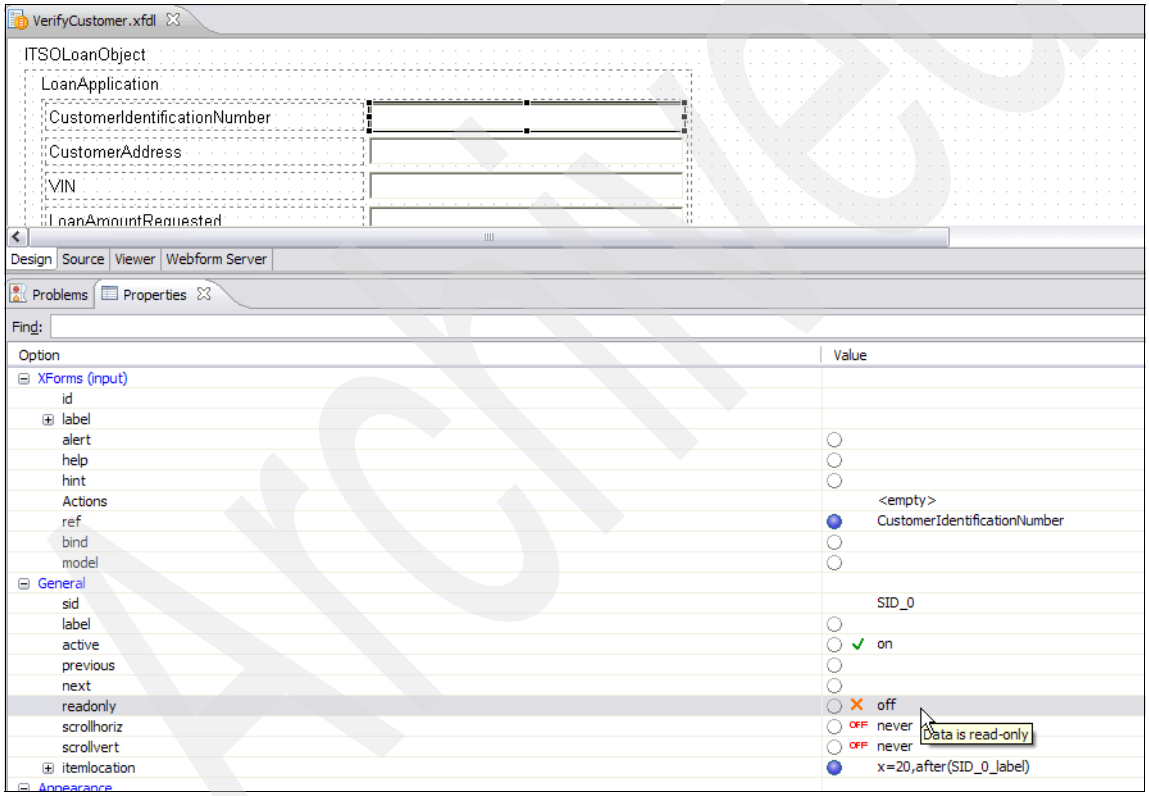


Figure 13-21 Read-only flag for the CustomerIdentification field

4. To restrict the RiskRatingScore to three digits, select **RiskRatingScore** and open the **Properties** tab. Click **Format** → **constraints** → **length** and amend the max value to 3, as seen in Figure 13-22.

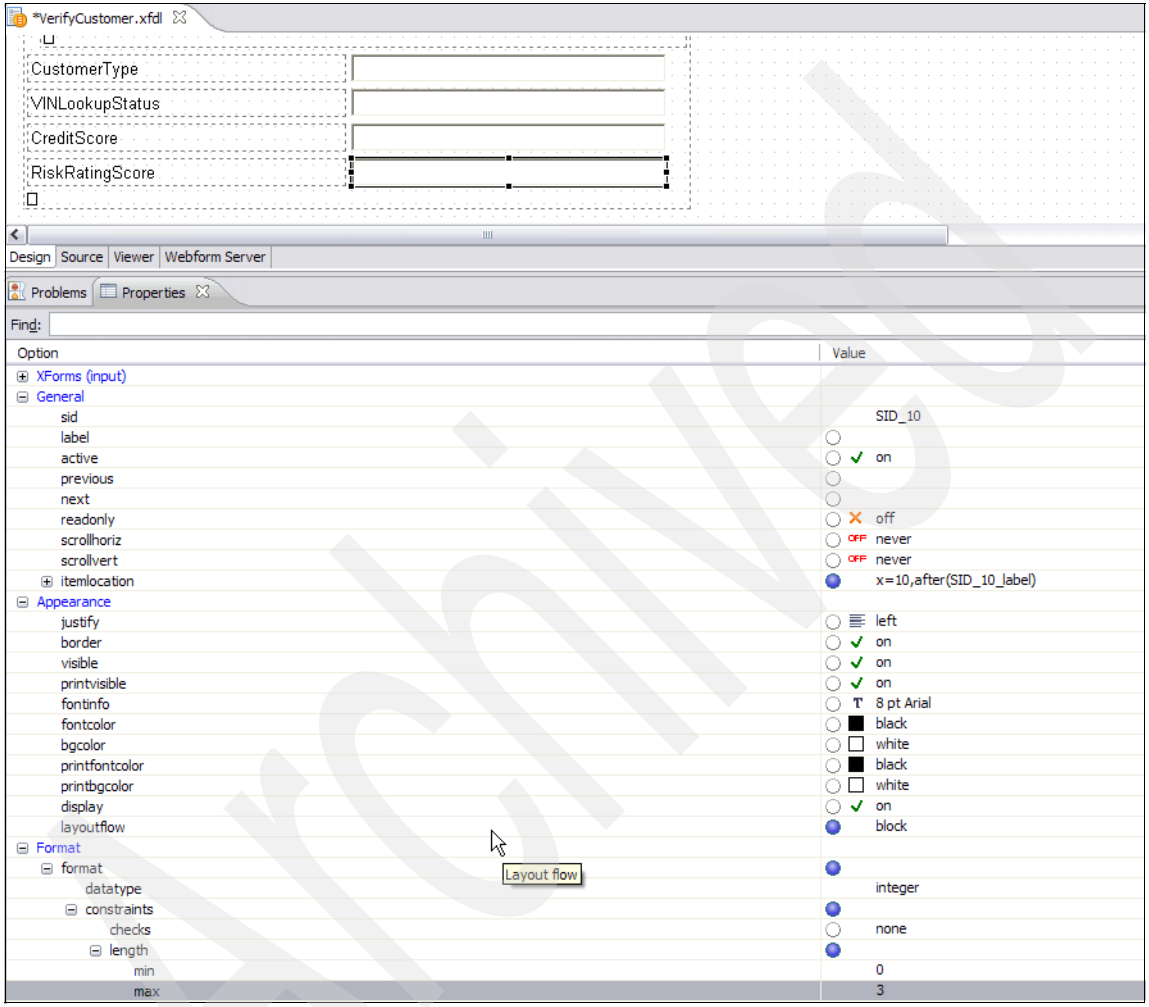


Figure 13-22 Changing the maximum size of a field

5. The last aspect to change is the presentation of the LoanAmountRequested field. You must add a group separator and a decimal separator that will be used if required when displaying the loan amount. To do this click **Format** → **presentation** → **decimalseparator** and enter a comma (,) for the value. Do the same for the groupseparator and enter a value of a comma (,).

6. Save the form and redeploy the application. When a process is in-flight and encounters the human task, the task can be claimed. When viewed using the

URL supplied above, the entries will now have the amended characteristics. Input values are shown in Figure 13-23 and the claimed task with amended values can be seen in Figure 13-24.

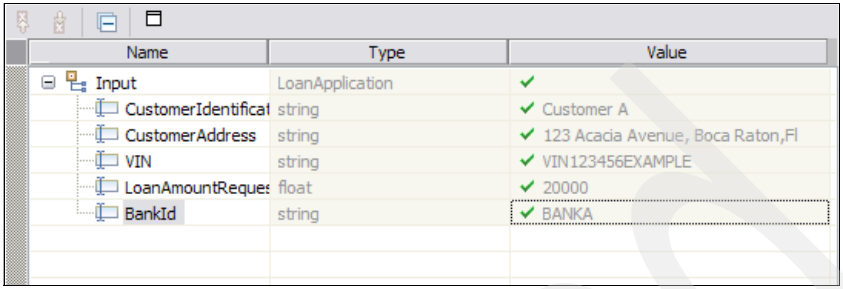


Figure 13-23 Initial input values

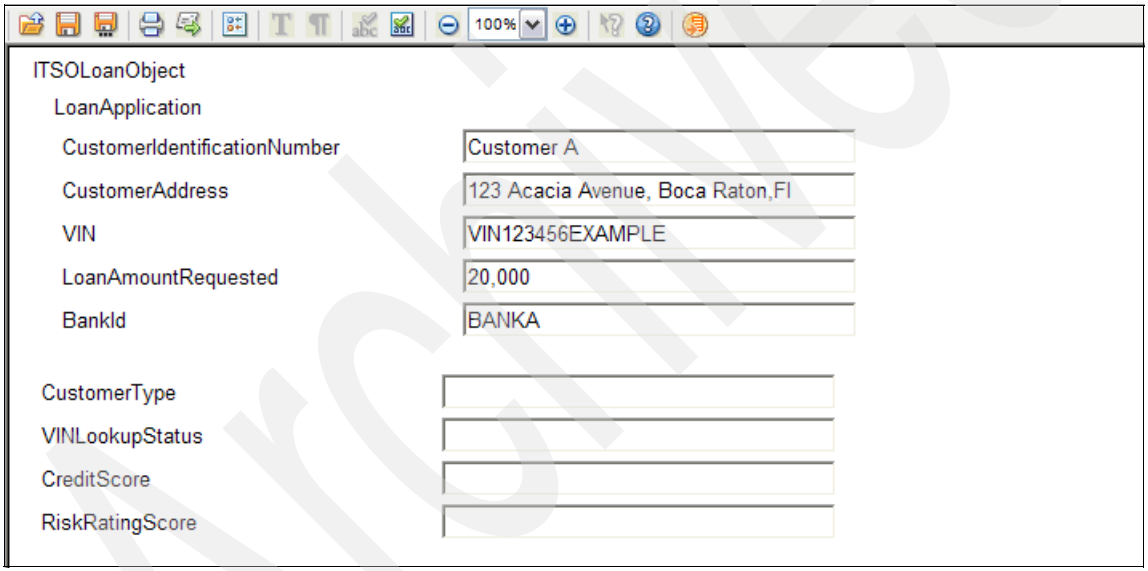


Figure 13-24 Amend presentation and formats in the claimed task

Part 3

Extending the Remote Message and Remote Support topology

14

Incorporating WebSphere Business Services Fabric into a Remote Messaging and Remote Support topology

WebSphere Business Services Fabric can be installed in multiple topologies:

- All components on a single server
- All components into a clustered topology

The clustered topology will achieve a highly available environment with fail over support. This chapter provides detailed instructions on how to incorporate WebSphere Business Services Fabric into a Remote Messaging and Remote Support topology pattern of WebSphere Process Server. It contains the following sections:

- Introduction
- Creating WebSphere Business Services Fabric deployment manager and custom profiles
- Verifying installation and configuration

14.1 Introduction

This chapter provides step-by-step instructions for incorporating WebSphere Business Services Fabric into the WebSphere Process Server Remote Messaging and Remote Support topology pattern. For instructions on how to construct this topology pattern see Chapter 7, “Configuring Remote Messaging and Remote Support” on page 165.

Figure 14-1 shows the Remote Messaging and Remote Support (RMRS) topology pattern, and shows where WebSphere Business Services Fabric components are added to it.

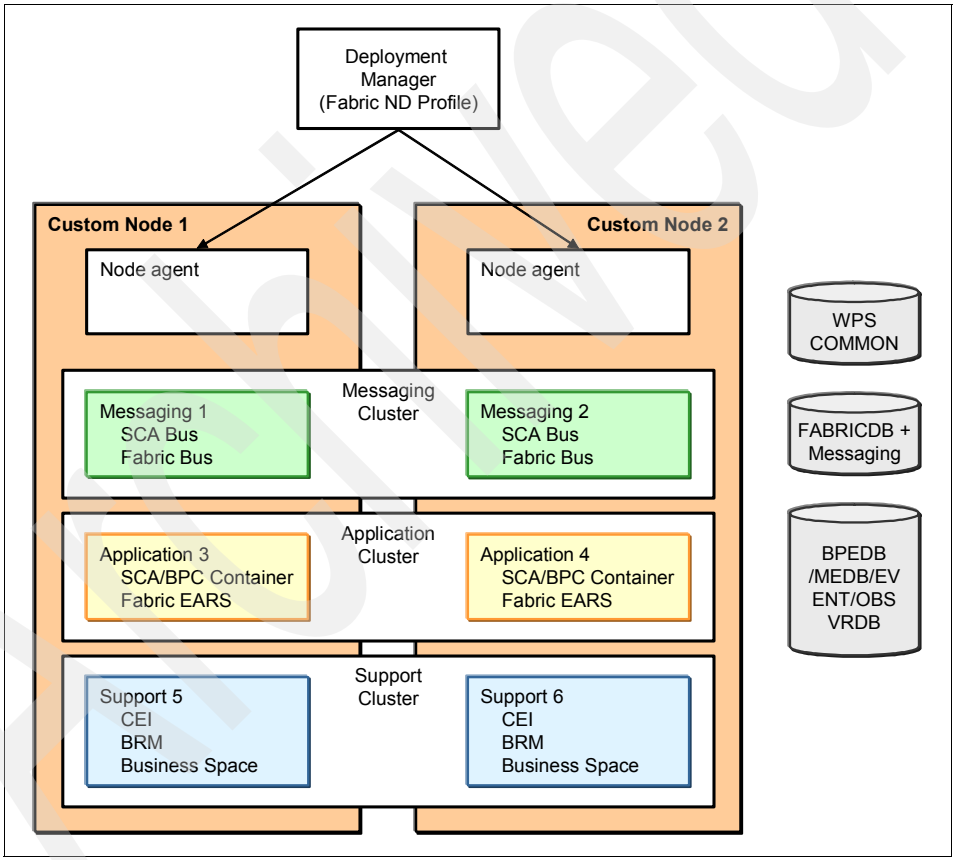


Figure 14-1 Fabric components in an RMRS topology

Fabric adds the following components to the RMRS topology:

- A service integration bus for Fabric is added to the cell.
- A bus member for the new bus is created using the Messaging cluster.
- The Fabric database is added to the topology. The Fabric database contains performance manager data.
- The messaging tables required for Fabric bus are added to the Messaging Engine database (MEDB).
- The Fabric application EAR files (Fabric_Tools, Fabric_Catalog, Fabric_Engine, Fabric_Rest_Service, and Tools_Help) are added to the deployment manager profile.
- The Fabric application EAR files are deployed to the application cluster.
- Fabric events are configured to be emitted to the JMS destinations present in the support cluster.

14.2 Creating WebSphere Business Services Fabric deployment manager and custom profiles

This section contains the following sections:

- Install WebSphere Business Services Fabric
- Install interim fix
- Augment WebSphere DMGR profile with Fabric DMGR profile
- Augment WebSphere custom profiles with Fabric custom profile
- Run the SIB configuration script
- Run the Fabric application deploy script
- WebSphere Business Services Fabric with Business Space
- Apply interim fix 31376
- Verify installation of interim fix 31376
- Post-installation steps
- Configure Fabric events for JMS destinations in Support Cluster

Note: All steps completed before 14.2.10, “Post-installation steps” on page 419, require that the clusters, deployment manager, and nodes be stopped.

Stop clusters, deployment manager, and nodes

In order to install WebSphere Business Services Fabric, you must first stop the WebSphere Process Server clusters, deployment manager, and nodes.

Stop the clusters

Navigate to **Servers** → **Clusters**. Select all the clusters that have a started status. Click **Stop** to stop each cluster. Wait for each cluster to stop before proceeding.

Stop the deployment manager

Navigate to `/opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/bin` and execute the following command:

```
./stopManager.sh
```

If administrative security is configured, then supply a valid user name and password when prompted.

You will receive a message indicating that the deployment manager stop was completed.

Stop the nodes

Navigate to `/opt/ibm/WebSphere/ProcServer/profiles/<profile_name>/bin` and execute the following command:

```
./stopNode.sh
```

You will receive a message indicating that the nodeagent stop was completed. Complete this step for each node in the cell.

14.2.1 Install WebSphere Business Services Fabric

This option is selected because in later steps you will augment the deployment manager and custom profiles with WebSphere Business Services Fabric. You may install WebSphere Business Services Fabric Foundation Pack either through the Launchpad (see “Install using the Launchpad (option 1)” on page 389) or silently (see “Silent installation (option 2)” on page 390). Select one of these options.

Install using the Launchpad (option 1)

To install:

1. Start the launchpad:

   ```
   ./launchpad.sh
   ```

2. Select **IBM WebSphere Business Services Foundation Pack Installation**. Then select the option to launch the installation.
3. Select **English** (or your appropriate language) and click **OK**.
4. At the Welcome window, click **Next**. Agree to the license terms.

5. At the Systems Prerequisites check window, click **Next**.
6. At the WPS Installation Location window, select or enter `/opt/ibm/WebSphere/ProcServer` and click **Next** (Figure 14-2).

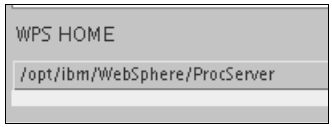


Figure 14-2 Installation path

7. At the Installation Types window, select **Files Only** and click **Next** (Figure 14-3).

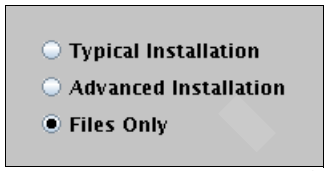


Figure 14-3 Installation Types window

8. At the Choose Install Folder, enter `/opt/ibm/WebSphere/Fabric/FoundationPack` and click **Next** (Figure 14-4).

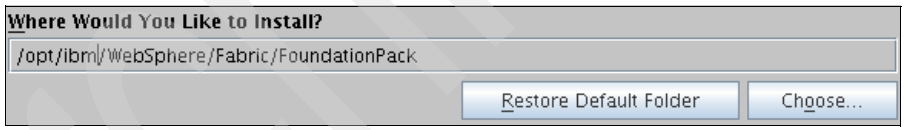


Figure 14-4 Installation path

9. Verify the Pre-Installation Summary and click **Install**.
10. Once the installation process indicates success, click **Done**.
11. Proceed to 14.2.2, “Install interim fix” on page 392. This is a required step.

Silent installation (option 2)

To do this:

1. Navigate to `/installersFP62`.
2. Locate the properties file named `fabric.foundation.properties`. This properties file controls the installation options.

3. Edit the file to make the following changes (Example 14-1):
   - Comment all the options for Advanced Installation and Typical Installation.
   - Set the options for Files Only installation.
   - Set the options for user directory and WPS installation directory.
   - Comment the options for Fabric profile, wps userid, wps password, and JDBC password. These are not needed for this type of installation.

Example 14-1 Modifications to fabric.foundation.properties

```
LICENSE_ACCEPTED=true

USER_INPUT_RESULTS=\"\",\"\",\"Files Only\"
USER_INPUT_RESULTS_1=
USER_INPUT_RESULTS_2=
USER_INPUT_RESULTS_3=Files Only
USER_INPUT_RESULTS_BOOLEAN_1=0
USER_INPUT_RESULTS_BOOLEAN_2=0
USER_INPUT_RESULTS_BOOLEAN_3=1

#USER_INPUT_RESULTS=\"\",\"\",\"Typical Installation\"
#USER_INPUT_RESULTS_1=Typical Installation
#USER_INPUT_RESULTS_2=
#USER_INPUT_RESULTS_3=
#USER_INPUT_RESULTS_BOOLEAN_1=1
#USER_INPUT_RESULTS_BOOLEAN_2=0
#USER_INPUT_RESULTS_BOOLEAN_3=0

#USER_INPUT_RESULTS=\"\",\"\",\"Advanced Installation\"
#USER_INPUT_RESULTS_1=
#USER_INPUT_RESULTS_2=Advanced Installation
#USER_INPUT_RESULTS_3=
#USER_INPUT_RESULTS_BOOLEAN_1=0
#USER_INPUT_RESULTS_BOOLEAN_2=1
#USER_INPUT_RESULTS_BOOLEAN_3=0

USER_INSTALL_DIR=/opt/ibm/WebSphere/Fabric/FoundationPack
WPS_HOME=/opt/ibm/WebSphere/ProcServer
DD_CHOICE_SELECTED=/opt/ibm/WebSphere/ProcServer
#WPS_PROFILE=wbsFabric
#WPS_USER=admin
#WPS_PASSWORD=passw0rd
#JDBC_PASSWORD=passw0rd
```

4. Run the following command:

   ```
   ./install_fabric_lnx -i silent -f fabric.foundation.properties
   ```

   You should see output similar to Figure 14-5, which indicates a successful installation.

```
Preparing to install...
Extracting the JRE from the installer archive...
Unpacking the JRE...
Extracting the installation resources from the installer archive...
Configuring the installer for this system's environment...

Launching installer...

Preparing SILENT Mode Installation...

===============================================================================
IBM WebSphere Business Services Fabric Foundation Pack v6.2(created with Instal
Anywhere by Macrovision)
-------------------------------------------------------------------------------
------------------------




===============================================================================
Installing...
-------------

 [==================|==================|==================|==================]
 [------------------|------------------|------------------|------------------]

Installation Complete.
```

Figure 14-5 Example output from successful installation

5. You can also verify the installation by navigating to the installation root, in this case, `/opt/ibm/WebSphere/Fabric/FoundationPack`. Locate the file named `IBM_WebSphere_Business_Services_Fabric_Foundation_Pack_v6_2_InstallLog.log` and find the Summary section. There should be zero warnings, NonFatalErrors, or FatalErrors (Figure 14-6).

```
Summary
-------

Installation: Successful .

332 Successes
0 Warnings
0 NonFatalErrors
0 FatalErrors
```

Figure 14-6 Sample output in install log

14.2.2 Install interim fix

Interim iFix JR31439 must be applied at this time.

Note: There is an accompanying iFix JR31440 for the Tool Pack. In this topology there is no need to install this particular iFix. However, if you were creating a development environment with WebSphere Integration Developer and the Fabric Tools, then it would be necessary to apply iFix JR31440.

Apply interim fix JR31439

To do this:

1. Download and expand the iFix compressed file for the Foundation Pack. The file name is `WBSF62-JR31349-foundation-pack-ifix-cd-image-multiplatform.zip`.
2. From the `/installers` directory of the expanded iFix, execute the command:

   ```
   ./install_fabriciFix_lnx
   ```

3. At the initial splash window, select an appropriate language. Click **OK** (Figure 14-7).

Figure 14-7 Splash window for iFix

4. At the Introduction window, click **Next**.
5. At the Choose Install Folder window, enter or choose `/opt/ibm/WebSphere/Fabric/FoundationPack` and click **Next**.
6. At the WebSphere Process Server Location window, enter or choose `/opt/ibm/WebSphere/ProcServer` and click **Next**.

7. Review the Pre-Installation Summary window and click **Install**.
8. Verify that you receive a message indicating that iFix 001 was successfully installed to `/opt/ibm/WebSphere/Fabric/FoundationPack`. Click **Done.**

14.2.3 Augment WebSphere DMGR profile with Fabric DMGR profile

To augment the deployment manager profile, you can use either the Profile Management Tool (option 1) or the manage profile command (option 2). The Profile Management Tool is supported only on 32-bit platforms. This section discusses both options.

> **Note:** Verify that DB2 has been started before proceeding.

Augment Dmgr using Profile Management Tool (option 1)

To do this:

1. Start the Profile Management Tool by navigating to `/opt/ibm/WebSphere/ProcServer/bin/ProfileManagement` and executing the following command:

   ```
   ./pmt.sh
   ```

2. Select **Augment an existing profile**.
3. At the Welcome window, click **Next**.
4. Select the profile that you want to augment. In this case, be sure to select a deployment manager profile (**Dmgr01**). Click **Next**.
5. The next dialog presents a list of augments. Select **WebSphere Business Services Fabric** from the list. Click **Next**.
6. Select **Advanced profile augmentation**. Click **Next**.
7. At the Administrative Security window, enter the administrative user name and password for the profile. Click **Next**.

Note: There are manual steps documented to create FABRICDB. See the following Information Center article:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp?topic=/com.ibm.ws.fabric.install.doc/fpi/task/t_setting_up_db_distributed.html

If you are using a remote database server, copy the fabric database script to the remote database server and execute the script manually. The Fabric database script can be found at `<$FABRIC_INSTALL>/configuration/database/db2/db2`.

Modify these files:

- `create_fabric_db_linux.sql`
- `create_fabric_schema.sql`

Execute these scripts:

- db2 -tvf create_fabric_db_linux.sql
- db2 -tvf create_fabric_schema.sql

8. At the Fabric Database Configuration window, enter or make the following selections and click **Next**:
   a. Choose **DB2 Universal** as the database product.
   b. Select **Create New Fabric database**.
   c. Enter `FABRICDB` as the database name.
9. At the Fabric Database Configuration (Part 2) window, enter or make the following selections and click **Next**.
   a. Enter a valid user name to authenticate.
   b. Enter a valid password to authenticate.
   c. Enter `/opt/ibm/WebSphere/ProcServer/universalDriver_wbi/lib` as the location of the JDBC classpath files.
   d. Select **4** as the JDBC driver type.
   e. Enter the host name of the database server.
   f. Enter the server port number (for DB2 on Linux the value is 50001).

   Note: The DB2 server port does not need to be 50001. This setting was used in this publication for demonstration purposes.

10. Review the Profile Augmentation Summary and click **Augment**.

11. Once the profile augmentation is complete, uncheck Launch the First steps console. Click **Finish**.

Augment Dmgr using manageprofiles command (option 2)

To do this:

1. Change directory to `/opt/bin/WebSphere/ProcServer/profiles/Dmgr01/bin`.
2. Run the **`manageprofile`** command by providing following parameters. The WebSphere Business Services Fabric **`manageprofiles`** command uses the same parameters as WebSphere Process Server in addition to Fabric-specific parameters. Table 14-1 provides details for the **`manageprofiles`** command.

Table 14-1 The manageprofiles command .

Parameter	Description
profileName	Name of profile to be augmented
templatePath	Path for dmgr.wsbfabric template
cellName	Cell name of the profile
nodeName	Node name of the profile
adminUserName	WPS admin user name
adminPassword	WPS admin password
fabricDbName	Name of Fabric database
fabricDbType	Specifies the Fabric Database Type Oracle10g for Oracle® DB2_UNIVERSAL for DB2
fabricDbUser	Specifies the user name for accessing the database server
fabricDbPassword	Specifies the password for the user for accessing the database
fabricDbHostName	Specifies the host name for database.

Parameter	Description
fabricDbServerPort	Specifies the port where the TCP/IP service is assigned or the port on which the database is listening
fabricDbDriverType	Specifies the driver type for the database. Support values are: ► 2 for Type 2 drivers ► 4 for Type 4 drivers
fabricDbCreateNew	Specifies whether a new database should be created for Fabric Support values are true or false For Oracle or remote database the value needs to be set to false. The database must exist if the value is set to false.
fabricDbJDBCClasspath	Specifies the JDBC driver path of database

Note: The fabricDbCreateNew = true value is not supported for Oracle databases. The Fabric database must exist in the Oracle database prior to running the **manageprofiles** command.

The following is an example of the manage profile command for augmenting the WebSphere Process Server deployment manager profile with the Fabric Deployment Manager profile:

```
./manageprofiles.sh -augment -profileName Dmgr01 -templatePath
/opt/ibm/WebSphere/ProcServer/profileTemplates/dmgr.wbsfabric
-cellName wpsCell01 –nodeName CellManager01 -adminUserName wps
-adminPassword passw0rd -fabricDbName FABRICDB -fabricDbType
DB2_UNIVERSAL -fabricDbUser db2inst1 -fabricDbPassword passw0rd
-fabricDbHostName localhost -fabricDbServerPort 50001
-fabricDbJDBCClasspath
```

```
/opt/ibm/WebSphere/ProcServer/universalDriver_wbi/lib
-fabricDbDriverType 4 -fabricDbCreateNew true
```

3. After running this command, you should receive the following message:

```
INSTCONFSUCCESS: Profile augmentation succeeded.
```

14.2.4 Augment WebSphere custom profiles with Fabric custom profile

To augment a custom profile, you can use the Profile Managerment Tool (option 1) or the manage profile command (option 2). The Profile Management Tool is supported only on 32-bit platforms. This section discusses both the approaches.

Verify that the deployment manager has been started successfully. If you have multiple custom profiles, you must repeat these steps for each custom profile.

Augment using Profile Management Tool (option 1)

To do this:

1. Start the Profile Management Tool by navigating to /opt/ibm/WebSphere/ProcServer/bin/ProfileManagement and executing the following command:

   ```
   ./pmt.sh
   ```

2. Select **Augment an existing profile**.
3. At the Welcome window, click **Next**.
4. Select the profile that you want to augment. In this case, be sure to select a custom profile (Custom01). Click **Next**.
5. The next dialog presents a list of augments. Select **WebSphere Business Services Fabric** from the list. Click **Next**.
6. Select **Typical profile augmentation**. Click **Next**.
7. At the Federation window, enter the following values and click **Next**:
   a. Enter the host name of the deployment manager.
   b. Enter the SOAP port number (for example, 8879) for the deployment manager.
   c. Enter the user name and password for administrative security on the deployment manager.
8. At the Database Configuration window, enter or make the following selections and click **Next**:
   a. Choose **DB2 Universal** as the database product.

b. Enter `/opt/ibm/WebSphere/ProcServer/universalDriver_wbi/lib` as the database name.

9. Review the Profile Augmentation Summary and click **Augment**.

> **Note:** If the augmentation fails, be sure to check the log file located in `/opt/ibm/WebSphere/ProcServer/logs/manageprofiles`. The log file name is <profile_name>_augment.log, which is different from the log file indicated in the failure message.

10. Once the profile augmentation is complete, uncheck Launch the First steps console. Click **Finish**.

Augment using manageprofiles command (option 2)

To do this:

1. For each of the custom nodes, execute the augmentation process. Change the directory to the appropriate custom profile, for example, `/opt/ibm/WebSphere/ProcServer/profiles/Custom01/bin`.
2. Run the **`manageprofiles`** command by providing following parameters. The WebSphere Business Services Fabric **`manageprofiles`** command uses the same parameters as WebSphere Process Server in addition to Fabric specific parameters. Table 14-2 provides details for the **`manageprofiles`** command.

Table 14-2 Parameters for the manageprofiles command

Parameter	Description
profileName	Name of profile to be augmented
templatePath	Path for managed.wsbfabric template
cellName	Cell name of the profile
nodeName	Node name of the profile
dmgrAdminUserName	Deployment Manager admin user name
dmgrAdminPasssword	Deployment Manager admin password
dmgrHost	Host name where Deployment Manager is running

Parameter	Description
fabricDbType	Specifies Fabric Database Type Oracle10g for Oracle DB2_UNIVERSAL for DB2 Value should be same as fabricDbType value specified during deployment manager augmentation
fabricDbJDBCClasspath	Specifies the JDBC driver path of database

The following is an example of the **manageprofiles** command for a augmenting WebSphere custom profile:

```
./manageprofiles.sh -augment -profileName Custom01 -templatePath
/opt/ibm/WebSphere/ProcServer/profileTemplates/managed.wbsfabric
-nodeName wpsNode01 –dmgrAdminUserName admin -adminPassword passw0rd
dmgrAdminPassword -dmgrHost localhost -dmgrPort 8879 -fabricDbType
DB2_UNIVERSAL -fabricDbJDBCClasspath
/opt/ibm/WebSphere/ProcServer/universalDriver_wbi/lib
```

3. You should receive the following message:

```
INSTCONFSUCCESS: Profile augmentation succeeded.
```

14.2.5 Run the SIB configuration script

This script creates the Fabric bus on the message cluster. The Fabric SIBus can use the same settings as the SCA System Bus, which is configured on the messaging cluster. The script must be run on the deployment manager profile.

Note: Make sure that the deployment manager and any nodes is in a started state. The last steps of the scripts synchronize the deployment manager with the nodes in the cell.

1. The script is in `/opt/ibm/WebSphere/ProcServer/profileTemplates/dmgr.wbsfabric/actions/scripts/cluster` and the file name is `fabricSIBConfig.py`.

2. Run **wsadmin** from /opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/bin with the command:

   ```
   ./wsadmin.sh -lang jython -user wps -password passw0rd -f
   /opt/ibm/WebSphere/ProcServer/profileTemplates/dmgr.wbsfabric/action
   s/scripts/cluster/fabricSIBConfig.py
   ```

3. You will be prompted for several parameters:
   - Enter 2 for the messaging cluster.
   - Press Enter to use the default values of the SCA System Bus settings.
   - Press Enter to use the default value of FABRICME for the Fabric messaging schema.
4. You should receive a message indicating that Fabric.wpsCell01.Bus was created. Changes were synchronized with the nodes in the cluster.
5. Use the administrative console to verify that the Fabric messaging bus was created successfully. Go to **Service integration** → **Service Integration Bus Browser**. The new bus should be in the list of buses.

Verify creation of Fabric bus and resources

If the fabricSIBConfig.py script ran successfully, there are a number of resources created. It is a good practice to verify that these resources do indeed exist and function properly.

1. Navigate to **Service integration** → **Buses**. You should see a bus named Fabric.<cell_name>.Bus in the list (Figure 14-8).

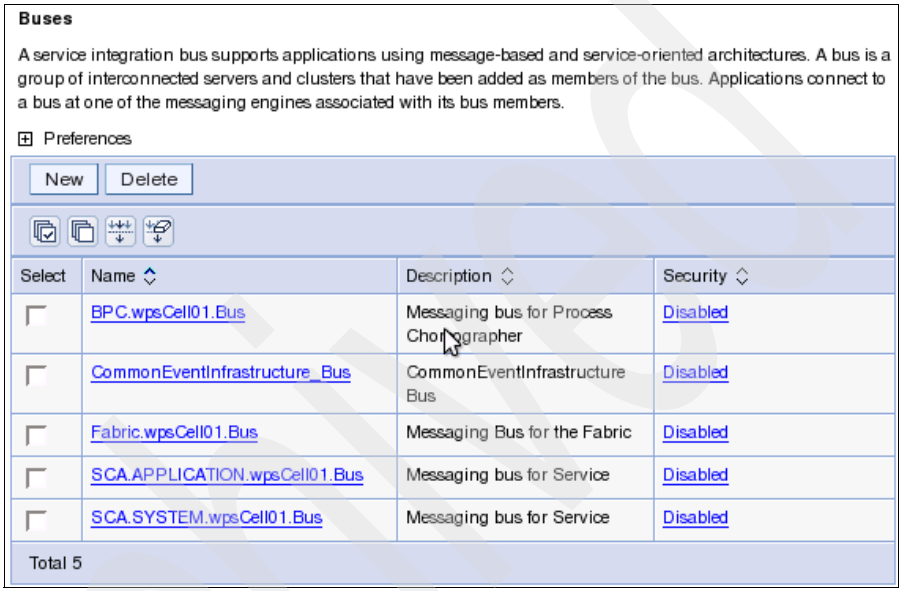


Figure 14-8 List of buses

2. The bus member's target is the messaging cluster that was selected when running the script. Click the bus. Select **Topology** → **Bus members** (Figure 14-9).

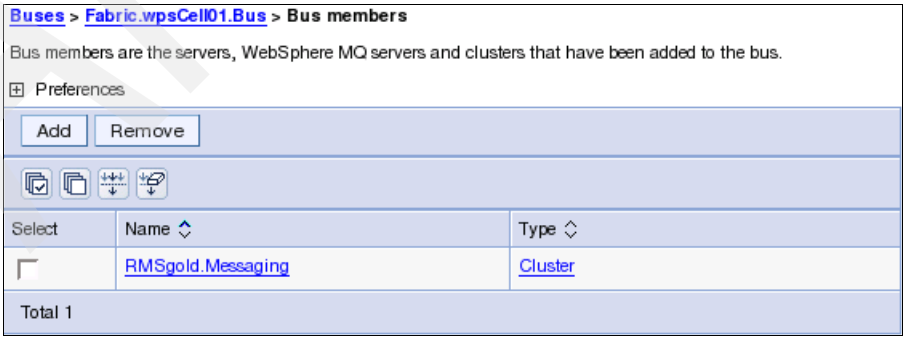


Figure 14-9 Bus member target

3. These topics are available as destinations. Click the bus. Select **Destination resources** → **Destinations** (Figure 14-10).

Figure 14-10 Destinations on the Fabric bus

4. The security settings for the bus were configured when you ran the script. The authentication alias for the Fabric bus is the same as the SCA System bus because you selected the SCA System bus settings when you ran the script. Click the bus. Select **Additional Properties** → **Security**, as shown in Figure 14-11.

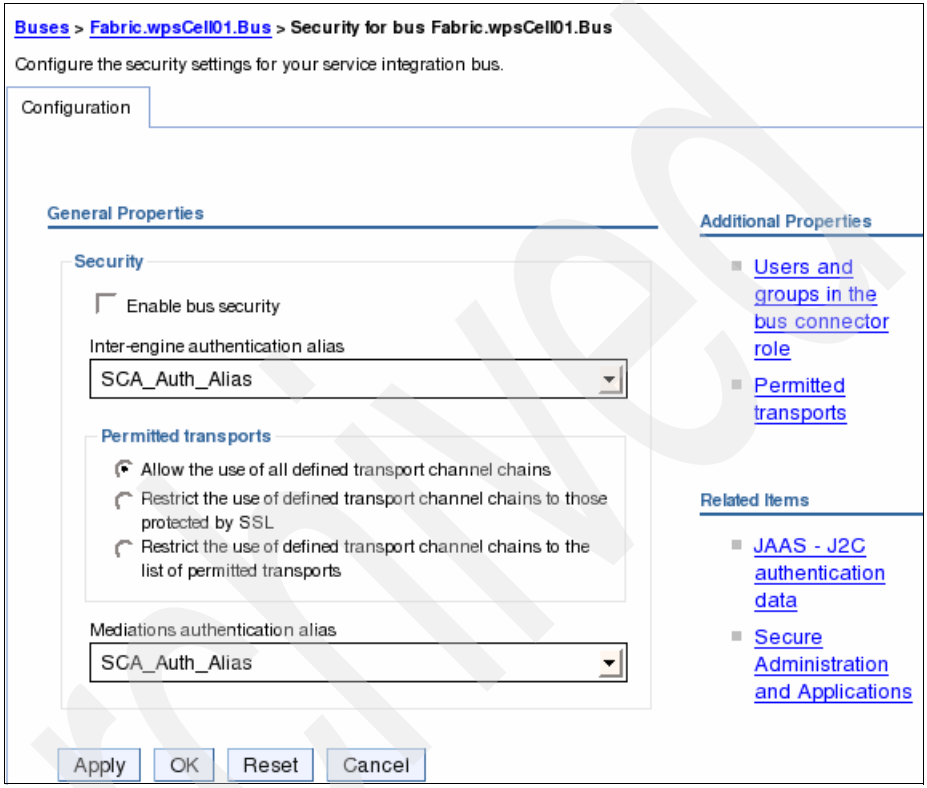


Figure 14-11 Bus security configuration

5. The bus connector role is the same as the SCA System bus because these settings were selected when running the script. Click **Additional Properties** → **Users and groups in the bus connector role**, as shown in Figure 14-12.

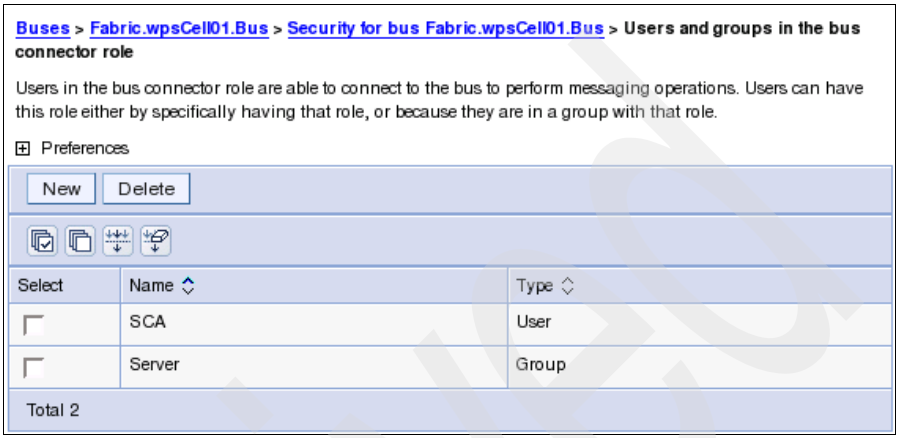


Figure 14-12 Users and groups in the bus connector role for the Fabric bus

6. A messaging engine is configured for the Fabric bus. Click the bus. Select **Topologies** → **Messaging engines** (Figure 14-13).

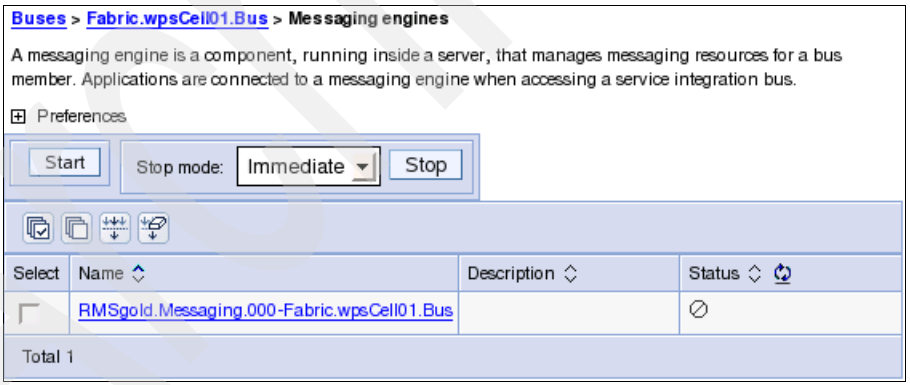


Figure 14-13 Messaging engine

7. The Fabric messaging engine uses the message store type as the SCA System bus. Click the messaging engine. Select **Additional properties** → **Message store**. The properties of the Fabric messaging engine data store was created with the setting shown in Figure 14-14.

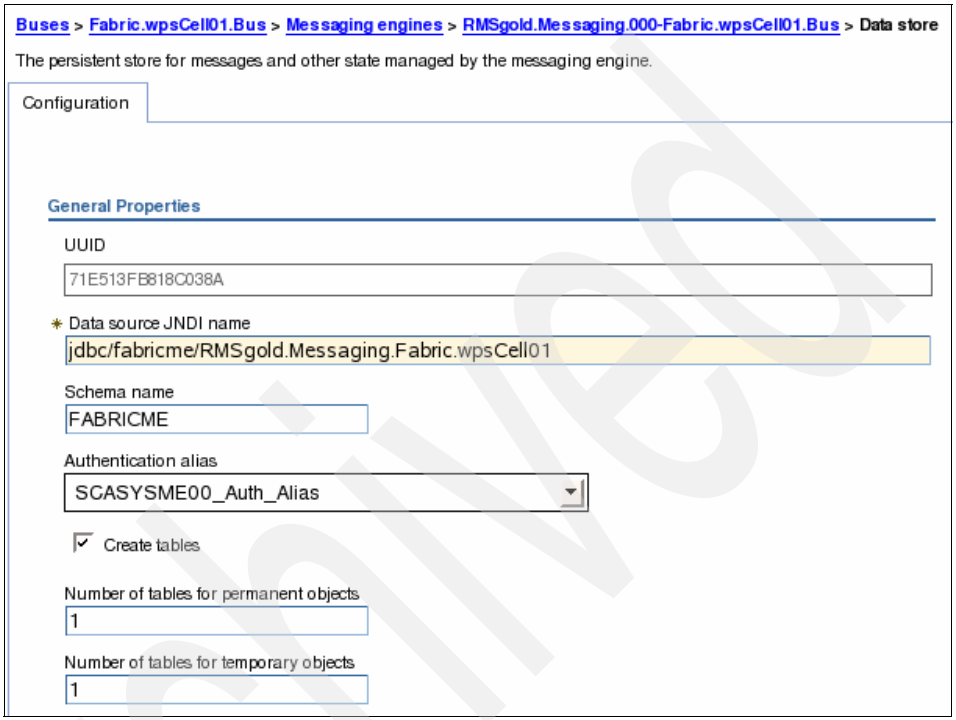


Figure 14-14 Message store configuration

8. A Fabric Bus messaging engine data source was created as well. The properties are identical to that of the SCA messaging engine data source. Navigate to **Resources** → **JDBC** → **Data sources**. Locate the data source shown in Figure 14-15.

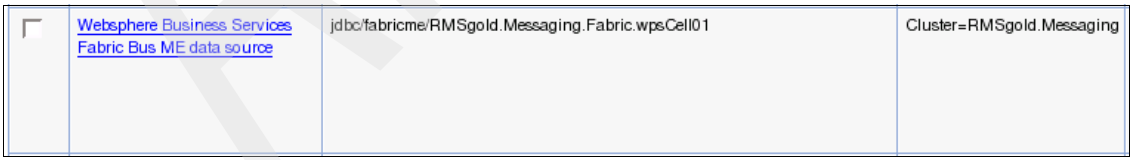


Figure 14-15 Data source for Fabric bus

14.2.6 Run the Fabric application deploy script

This script deploys the Fabric application to the cluster. The App Target cluster must be started before executing this script. In order for the App Target cluster to start without problems, you should start the Messaging and Support clusters in that order first. There are several parameters required by the scripts:

1. The script is in `/opt/ibm/WebSphere/ProcServer/profileTemplates/dmgr.wbsfabric/actions/scripts/cluster` and the file name is `fabricAppDeploy.py`.

   Note: On Linux systems, be sure to set the ulimit to 8192:

   ```
   ulimit -n 8192
   ```

2. Run **`wsadmin`** from `/opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/bin` with the command:

   ```
   ./wsadmin.sh -user wps -password passw0rd -f
   /opt/ibm/WebSphere/ProcServer/profileTemplates/dmgr.wbsfabric/actions/scripts/cluster/fabricAppDeploy.py
   ```

3. You will be prompted for several parameters:
   - Enter `1` for the application cluster.
   - Enter `wsadmin` for the user ID with access to the Fabric administrative console (also known as the Fabric Tools).
   - Enter `/opt/ibm/WebSphere/ProcServer/profileTemplates/dmgr.wbsfabric`.
4. You should receive a message indicating that the Fabric enterprise applications were deployed and that synchronization is complete for the node.

Verify the creation of resources

The Fabric application deploy script completed the following actions:

- Deployed five enterprise applications into the deployment manager profile
- Created JMS resources scoped at the application cluster
- Created several environment resources

To verify that the application deploy script was successful, perform the following:

1. The enterprise applications shown in Figure 14-16 were deployed.

	Fabric_Catalog	
	Fabric_Engine	
	Fabric_REST_Services	
	Fabric_Tools	
	Fabric_Tools_Help	

Figure 14-16 Installed Fabric applications

2. The target for each enterprise application should be mapped to the application cluster that was supplied when running the script. Click one of the applications and check the target. In this example, the status of Fabric Catalog is verified (Figure 14-17).

Enterprise Applications > **Fabric_Catalog** > **Target specific application status**

Use this page to view a mapping of a deployed object, such as an application or module, into a target server or cluster environment. This page displays the status of the enterprise application or module on each server or cluster.

Preferences

Enable Auto Start | Disable Auto Start

Select	Target	Node	Version	Auto Start	Application Status
	RMSgold.AppTarget	Not applicable	Not applicable	Yes	

Total 1

Figure 14-17 Target specific application status

3. Verify the JMS resources created by the script. The first of these resources is the DA Event Connection Factory, which was created to queue connections to the Dynamic Assembler. The target for the connection factory is the application cluster. Select **Resources** → **JMS** → **Connection factories** (Figure 14-18). Scroll to the far right to locate the scope of this resource.

Select	Name	JNDI name
	BFMIF_RMSgold.AppTarget.com.ibm.bpe.api.sca.BFMJMS_FAILEDEVENT_REPLAY_CF	BFMIF_RMSgold.AppTarget/com/ibr
	BFMJMSReplyCF	jms/BFMJMSReplyCF
	BPECF	jms/BPECF
	BPECFC	jms/BPECFC
	DA Event Connection Factory	jms/fabric/DAEventConnectionFacto

Figure 14-18 Connection factories used by Fabric

4. The hub request queue was created to queue requests to the hub, not the Dynamic Assembler. The scope for this queue is the application cluster. Select **Resources** → **JMS** → **Queues** (Figure 14-19).

	HUB Request Queue	jms/fabric/HubRequestQueue

Figure 14-19 Queues used by the Fabric

5. The DA Event topic was created to publish requests to the Dynamic Assembler. The scope for this resource is the application cluster. Select **Resources** → **JMS** → **Topics** (Figure 14-20).

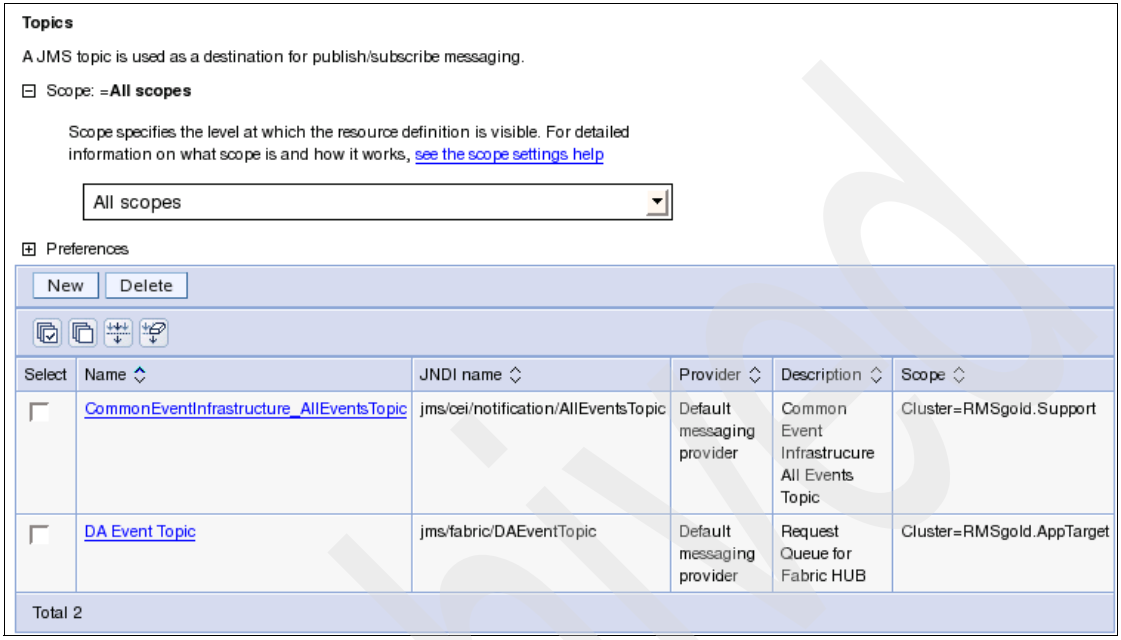


Figure 14-20 Dynamic Assembler topic.

6. Three activation specifications were created. DA_PerfMon_Activation was created to handle the Fabric Performance Monitor events. HUB_Event Activation was created to handle events fired from the hub. HUB_Request_Activation was created to handle requests to the hub. Because security has been configured, the authentication alias for each activation specification has been set to the SCA_Auth_Alias (Figure 14-21). Select **Resources** → **JMS** → **Activation Specifications**.

	DA PerfMon Activation	jms/fabric/DAPerfMonActivation	Default messaging provider	Activation for Fabric Event Topic	Cluster=RMSgold.AppTarget
	HTMInternalActivationSpec	eis/HTMInternalActivationSpec	Default messaging provider		Cluster=RMSgold.AppTarget
	HUB Event Activation	jms/fabric/HubEventActivation	Default messaging provider	Activation for Fabric Event Topic	Cluster=RMSgold.AppTarget
	HUB Request Activation	jms/fabric/HubRequestActivation	Default messaging provider	Request Activation for Fabric HUB	Cluster=RMSgold.AppTarget

Figure 14-21 Activation specifications

7. There are two name space bindings created at the application cluster level. Select **Environment** → **Naming** → **Name Space Bindings** (Figure 14-22).

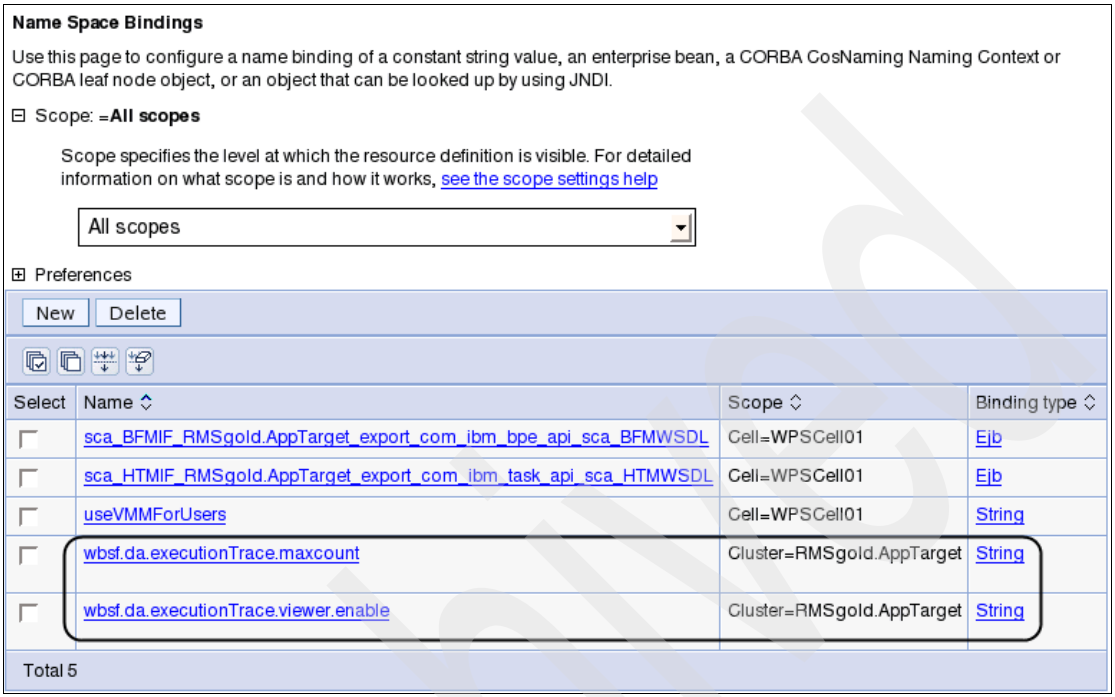


Figure 14-22 New name space bindings

8. A replication domain was created for context replication targeted to the entire domain. Select **Environment** → **Replication domains** (Figure 14-23).

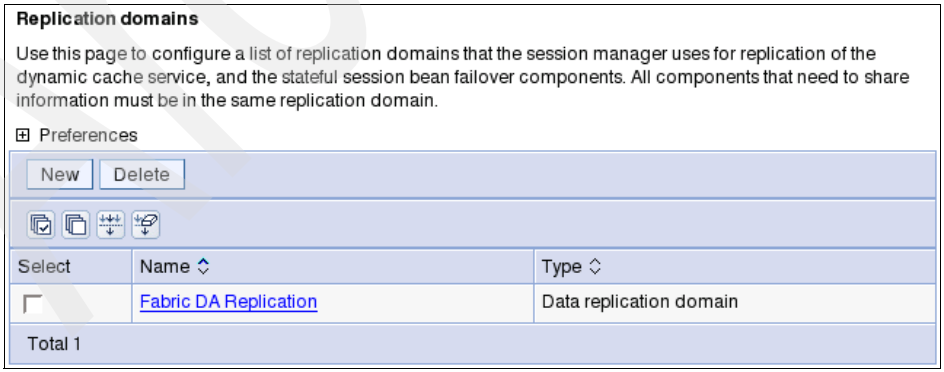


Figure 14-23 New replication domain

9. An object cache instance named Fabric Context Cache is created at the application cluster level. This cache is used by the Fabric applications to store, distribute, and share data. The cache provides better tuning of cache resources. Select **Resources** → **Cache instances** → **Object cache instances** (Figure 14-24).

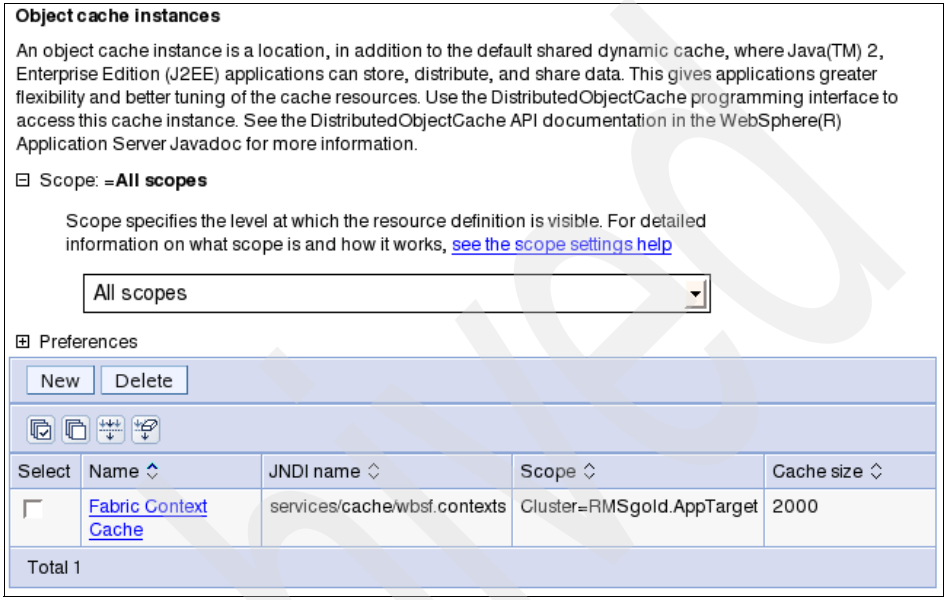


Figure 14-24 New Fabric Context Cache

10. The details for the Fabric Context Cache show that the cache replication is enabled with the replication targeted to the Fabric DA replication domain (Figure on page 414).

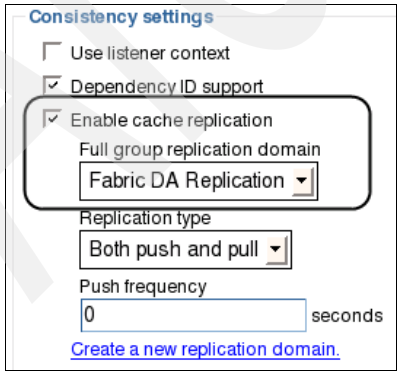


Figure 14-25 Target for the Fabric Context Cache

Note: The Fabric Context Cache can be further tuned using WebSphere Process Server's Integrated Solutions Console.

14.2.7 WebSphere Business Services Fabric with Business Space

This section describes how to set the correct endpoint for the Fabric REST services.

Changing the Fabric Rest endpoints for Business Space

If the Business Space application is deployed in a different cluster from Fabric REST service (services.ear), you must manually edit the `wbsfEndpoints.xml` to provide the correct endpoint for the Fabric Rest URL. Complete this task for the deployment manager and all custom nodes where Fabric is installed:

1. Using the editor of your choice, open `<profile>//BusinessSpace/registryData/wbsfEndpoints.xml`.
2. Locate the line:

   `<tns:url></tns:url>`
3. Change the line to the following values:

   `https://<fabric server>:<secured fabric server port>/fabricrest`

   Where *<fabric server>* is the server host or IP address where the Fabric application is deployed and *<secured fabric server port>* is the secured port (HTTPS).

14.2.8 Apply interim fix 31376

This interim fix provides for the use of Fabric widgets in Business Space. It is important to use the most current update installer from IBM to apply this iFix. Also, verify that all servers, clusters, deployment manager, and node are in a stopped state. None of these should be started when this fix is applied.

1. Copy the interim fix file to `/opt/IBM/WebSphere/UpdateInstaller/maintenance`.
2. Start the WebSphere Update Installer from `/opt/IBM/WebSphere/UpdateInstaller/update.sh`.
3. At the Welcome window, click **Next**.
4. At the Product Selection window, enter or browse to `/opt/ibm/WebSphere/ProcServer` and click **Next**.

5. At the Maintenance Operation Selection window, select **Install maintenance package** and click **Next**.
6. At the Maintenance Package Directory Selection window, enter or select **/opt/IBM/WebSphere/UpateInstaller/maintenance** and click **Next** (Figure 14-26).
7. At the Available Maintenance Package to Install window, be sure the update (`.pak` file) is selected and click **Next** (Figure 14-26).

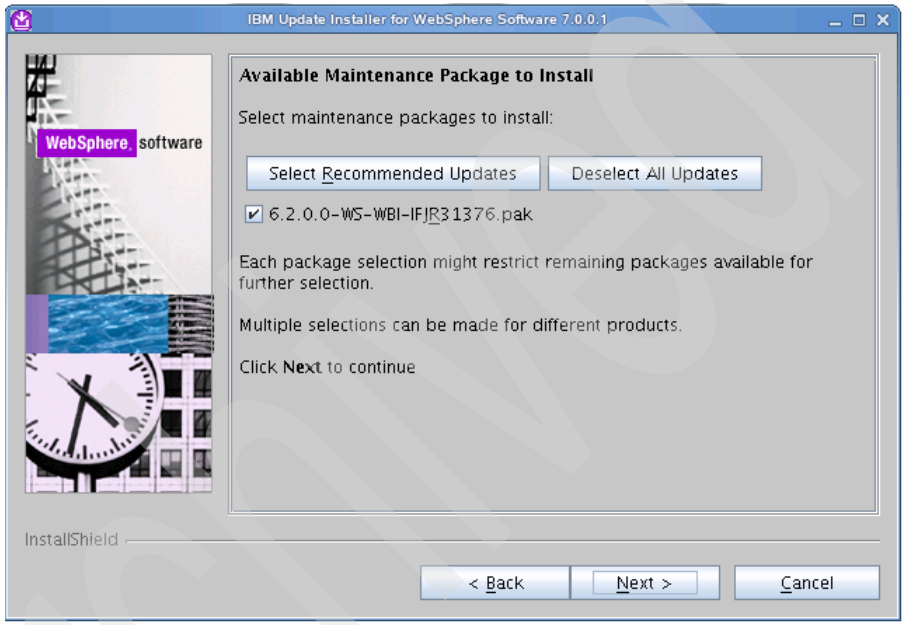


Figure 14-26 Verification window for maintenance package installation

8. At the Installation Summary window, clear the selection to verify permissions to perform the installation. You should be logged in as root. Click **Next**.
9. Once the update installer has completed, you should see a message that reads as follows:

   ```
   Success: The folowing maintenance package was installed:
   6.2.0.1-WS-WBI-IF-JR31376 - Update 6.2.0.0 Bspace IFIX for Fabric
   Widgets
   ```

10. Click **Finish** to close the Update Installer program.

14.2.9 Verify installation of interim fix 31376

To do this:

1. Start the deployment manager, any custom nodes, and clusters.
2. Stop the applications named BusinessSpaceManager and IBM_BSPACE_WIDGETS. Verify that the applications are stopped properly on all of the servers or clusters before proceeding. See Figure 14-27.

Figure 14-27 BusinessSpaceManager and IBM_BSPACE_WIDGETS stopped

3. Select the **BusinessSpaceManager** application and click **Update**. Select the option of updating the entire application.
4. In the path for the application, enter or browse to `/opt/ibm/WebSphere/ProcServer/installableApps/BSpaceManager.ear`. Click **Next**.
5. At Step 1: Select installation options, accept all the defaults and click **Next**.

6. At Step 2: Map modules to servers, verify that the Map modules to servers section shows the correct mapping for the Business Space Manager application. The mapping should be to the clusters where you configured Business Space. The Business Space Manager is mapped to the Support Cluster. Click **Next** (Figure 14-28).

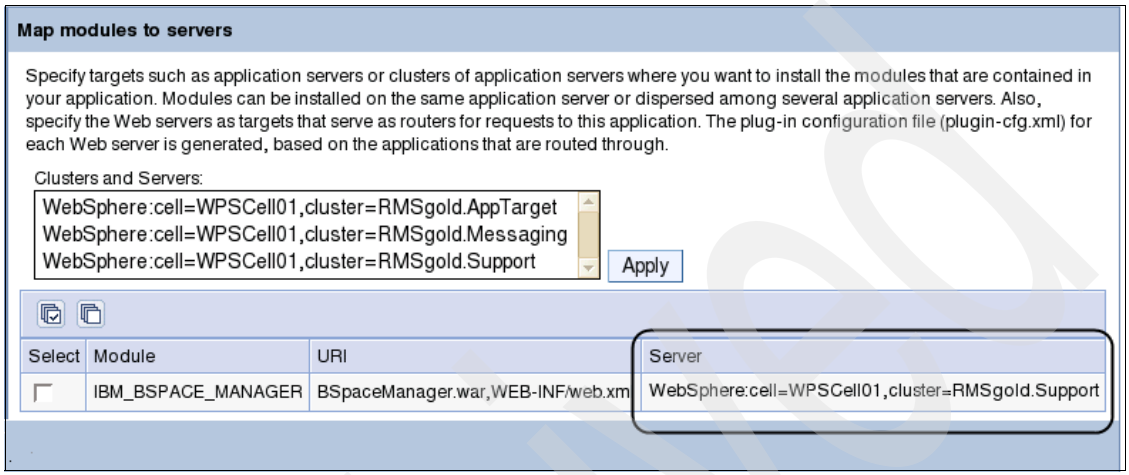


Figure 14-28 Module mapped to support cluster

7. At Step 3: Summary, verify the selections you made and click **Finish**.
8. Make sure that the update completes without errors. Save changes to the configuration.
9. Select the **IBM_BSPACE_WIDGETS** application and click **Update**.
10. In the path for the application, enter or browse to `/opt/ibm/WebSphere/ProcServer/installableApps/BSpaceWidgets.ear`. Click **Next**.
11. At Step 1: Select installation options, accept all the defaults and click **Next**.

12. At Step 2: Map modules to servers, verify that the Map modules to servers section shows the correct mapping for the IBM_BSPACE_WIDGETS application. The mapping should be to the clusters where you configured Business Space. The widgets are mapped to the Support Cluster. Click **Next** (Figure 14-29).

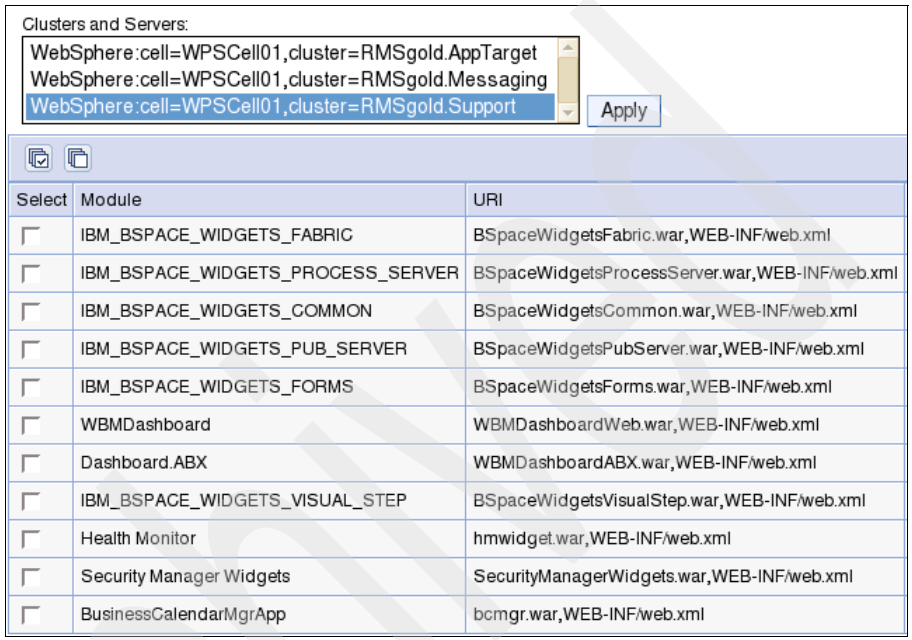


Figure 14-29 Business Space widgets

13. At Step 3: Summary, verify the selections that you made and click **Finish**.
14. Save the application to the master configuration. Allow time for the save process to complete for the 60 MB file. Do not select any links until the save process completes.
15. Synchronize the nodes. Allow time for the data (about 80 MB) to synchronize across all nodes.
16. Start the **BusinessSpaceManager** application and then start **IBM_BSPACE_WIDGETS**. Verify that no errors are returned during application startup.
17. Check that the Business Space console starts properly. Open a browser and enter `http://<host_name>/9080:BusinessSpace`.
18. You should be presented with the login page for the Business Space.

14.2.10 Post-installation steps

This step requires that you start the deployment manager, node agent, and cluster.

Granting user access to Fabric Authoring in Business Space

In order to access Fabric Administration Business Space, the user must be part of the FabricAdministrators group. Perform the following steps:

1. Log in to the WebSphere Process Server admin console and navigate to **Users and Groups** → **Manage Groups**.
2. Click **Create**. Provide `FabricAdministrators` as the group name, then click **Create**.
3. Add the WebSphere Process Server admin or any other ID having administrator rights to the above group, then click **Save**.

14.2.11 Configure Fabric events for JMS destinations in Support Cluster

WebSphere Business Services Fabric events are emitted to the JMS destinations present in the support cluster. WebSphere Business Services Fabric expects destination values in a namespace wbsf-cbe-emitter-factory. This section describes the steps to create and configure the namespace variable.

1. In the Integrated Solutions Console, navigate to **Servers** → **Clusters** and select the Application Cluster.
2. Expand **Common Event Infrastructure** and click **Common Event Infrastructure Destination**.
3. Note the JNDI Name under Event Infrastructure emitter factory JNDI name. In our scenario it is cell/clusters/default.Support/com/ibm/events/configuration/emitter/Default.
4. Navigate to **Environment** → **Naming** → **Name Space Bindings**. In the Scope selection box select **Cluster=default.AppTarget** (select your Application Cluster) and click **New**.
5. Select **String** as the binding type and click **Next**.
6. Specify the following values:
   - Binding Identifier: `wbsf-cbe-emitter-factory`
   - Name in name space: `wbsf-cbe-emitter-factory`

– String value:
`cell/clusters/default.Support/com/ibm/events/configuration/emitter/Default`

Note: The value that must be provided is from step 3.

7. Click **Finish** and save your changes.

14.3 Verifying installation and configuration

This section provides verification points to ensure that Fabric is successfully installed in the RMRS topology. The section provides detailed verification points for each of the steps discussed in 14.2, “Creating WebSphere Business Services Fabric deployment manager and custom profiles” on page 388.

14.3.1 Verify augmentation of deployment manager profile

After the deployment manager profile is augmented, there are several configurations that should be verified.

Verify data source for performance manager

At the cell level, a data source is added for the performance manager. This data source is use to connect to the Fabric database (FABRICDB).

1. From the Integrated Solutions Console, select **Resources** → **JDBC** → **Data sources**.
2. Verify that fabric_pm was added to the end of the list of data sources (Figure 14-30).

	Websphere Business Services Fabric Bus ME data source	jdbc/fabricme/RMSgold.Messaging.Fabric.wpsCell01	Cluster=RMSgold.Messagi
	event	jdbc/cei	Cluster=RMSgold.Support
	fabric pm	jdbc/fabric/pm	Cell=wpsCell01

Figure 14-30 Data source named fabric_pm added during augmentation

Verify J2C authentication data

During augmentation, you supplied a valid user name and password combination to authenticate to the Fabric database. From those inputs, the FABRIC_JDBC_AUTH authentication alias was created and is used as the component managed authentication alias for the fabric_pm data source.

1. From the Integrated Solutions Console, select **Resources** → **JDBC** → **Data sources** → **fabric_pm**.
2. Select **Related items** → **JAAS** → **J2C authentication data**.

3. Verify that FABRIC_JDBC_AUTH is located in the list (Figure 14-31).

	CEIME_RMSgold.Messaging_Auth_Alias	db2inst1	CEI ME data source authentication alias
	CommonEventInfrastructureJMSAuthAlias	SCA	CEI JMS authentication alias
	FABRIC_JDBC_AUTH	db2inst1	
	SCAAPPME00_Auth_Alias	db2inst1	SCA Application Bus ME data source authentication alias

Figure 14-31 FABRIC_JDBC_AUTH authentication alias added during augmentation

Verify the updated WebSphere variable

If you entered a new value for the location of the JDBC class files, then the WebSphere variable named DB2UNIVERSAL_JDBC_PATH will be updated. If you did not change this value, then you do not need to check its value.

1. From the Integrated Solutions Console, select **Environment** → **WebSphere Variables**.
2. Locate DB2UNIVERSAL_JDBC_PATH in the list. Make sure that you find the variable that is scoped to the cell, not the cluster, as shown in Figure 14-32.

DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH		Node=CellManager01
DB2UNIVERSAL_JDBC_DRIVER_PATH	/opt/ibm/WebSphere/ProcServer/universalDriver_wbi/lib	Cell=wpsCell01

Figure 14-32 Variable scoped to the cell level

The value should be /opt/ibm/WebSphere/ProcServer/universalDriver_wbi/lib.

Verify addition of plug-ins and Fabric SCA JAR files

To do this:

1. Verify that the following two plug-ins were copied to /opt/ibm/WebSphere/ProcServer/plugins:
   - com.ibm.ws.repository_6.2.0.XXXX.jar
   - com.ibm.ws.fabric.catalog_6.2.0.jar

2. Verify that the following five Fabric SCA jar files were copied to /opt/ibm/WebSphere/ProcServer/lib/ext, as shown in Figure 14-33.

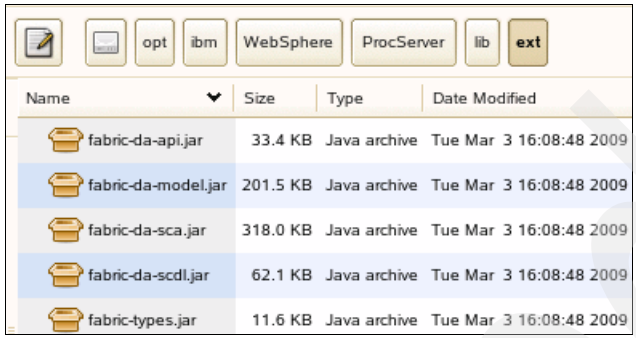


Figure 14-33 Fabric JAR files added during augmentation.

14.3.2 Verify augmentation of the custom profile

If you entered a new value for the location of the JDBC class files, then the WebSphere variable named DB2UNIVERSAL_JDBC_PATH is updated. If you did not change this value, then you do not need to check its value.

1. From the administrative console, select **Environment** → **WebSphere Variables**.
2. Locate DB2UNIVERSAL_JDBC_PATH in the list. Make sure that you find the variable that is scoped to the node. Figure 14-34 illustrates this variable defined to the two nodes in the cell.

Figure 14-34 Updated WebSphere variables

The value should be /opt/ibm/WebSphere/ProcServer/universalDriver_wbi/lib.

14.3.3 Verify execution of Fabric interactive scripts on DMGR profile

The interactive scripts installed Fabric enterprise applications on the Application cluster and configured the Fabric bus on the messaging cluster.

To verify that the Fabric enterprise application is successfully installed:

1. From the Integrated Solutions Console, select **Applications** → **Enterprise Applications**.
2. Verify that the following EARs are deployed and started successfully, as shown in Figure 15-7:
   - Fabric Catalog
   - Fabric Tools
   - Fabric Engine
   - Fabric Tools Help
   - Fabric REST Services

	Fabric_Catalog	
	Fabric_Engine	
	Fabric_REST_Services	
	Fabric_Tools	
	Fabric_Tools_Help	

Figure 14-35 Fabric enterprise applications

3. Type the following URL in the browser. In this example, the URL is http://localhost:9080/fabric.

   ```
   http://<host_name>:<cluster_member_port#>/fabric
   ```

4. You should be presented with the login window for WebSphere Business Services Fabric. Enter the credentials for logging in. In this example, user wps and password passw0rd were used.

After a successful login, you will be presented with the WebSphere Business Service Welcome window, as shown in Figure 14-36.

Figure 14-36 WebSphere Business Service Fabric Welcome window

To verify that the Fabric Bus was successfully started:

1. From the administrative console, select **Service Integration** → **Buses**.
2. Verify that the Fabric.*xxx*.Bus exists (where *xxx* is the cell name). Click **Fabric.xxx.Bus**.
3. On the Fabric.xxx.Bus page, click **Message engines** and check that the message engine (clusterName.000-Fabric.cellName.Bus) exists with a started status.

14.4 Deploying and testing the vehicle loan application

The vehicle loan application described in Chapter 2, “Sample business application scenario used in topologies” on page 23, can be deployed to WebSphere Business Services Fabric.

The sample application described in this section is supplied with the additional material provided with this book. Refer to the \Scenarios sub directory in the additional material supplied with this book. See Appendix A, “Additional material” on page 597, to obtain it.

14.4.1 Update endpoint URLs

The ontology files for the sample application contain endpoint URLs specific to the environment in which they were developed. It is therefore necessary to tailor this information to your deployment environment:

1. Locate the ontology archive `ITSOBankCBAPrj20090303-owl.zip` and extract it into a directory in your workstation.
2. Open the file `http_www.ibm.com_vehicleloan_inst.owl` in your favorite text editor.
3. Search for the string http://itsodmgr and replace the host name itsodmgr with your <*host*>:<*port*> (where <*host*> is the WebSphere Business Services Fabric host and <*port*> is the HTTP listening port. There is no port following the host name itsodmgr in the original file because it used the default HTTP port 80.
4. Save the file.
5. Add the updated ontology file back into the archive `ITSOBankCBAPrj20090303-owl.zip`.

14.4.2 Ontology setup

Perform the following steps to import the Fabric Content Archives into the Governance Manager:

1. Log into the WebSphere Business Services Fabric console by accessing the following URL:

   `http://<hostname>:<port>/fabric`

   For example, the deployment system URL used for this book is:

   `http://blade41.itso.ibm.com:9080/fabric`
2. Navigate to **Governance Manager** → **Import/Export**.

3. Click **Browse** and locate the \Scenarios\Fabric\FCA folder in the zip file provided in the additional material of this book. Choose **OrgUsersAndRoles20090303-owl.zip** and click **Import file**.
4. Repeat these steps to install the remaining ontology files in the following order:
   a. FabricGovernance20090303-owl.zip: Extensions to the core WebSphere Business Services Fabric ontology
   b. ITSOBankOntPrj20090303-owl.zip: Extensions to the assertion ontology
   c. ITSOBankCBAPrj20090303-owl.zip: Extended ontology for the project

14.4.3 Deployment of the enterprise application

Perform the following steps to deploy the Business Services, Business Process Execution Language (BPEL) processes, and their implementation into WebSphere Process Server:

1. Log into the WebSphere Process Server administration console by accessing the following URL:

   `http://<hostname>:<port>/ibm/console`

   For example, for our system the URL is:

   `http://blade41.itso.ibm.com:9061/ibm/console`
2. Navigate to **Applications** → **Install New Application**.
3. Click **Browse** and locate the `\Scenarios\Fabric\EAR\WebSphereEnvUtil.ear`. Click **Next**.
4. At the Select installation options page, accept the default settings and click **Next**.
5. At the Map modules to servers page, accept the default settings and click **Next**.
6. Review the Summary page, then click **Finish**.
7. That completes the installation of the supporting application. Repeat these steps to install the remaining enterprise applications:
   a. ITSO_impl.ear: Implementation of the services invoked by the loan application
   b. ITSO.ear: Technical models of the vehicle loan process in a Service Component Architecture (SCA) module

14.4.4 Integrating the LDAP repository

> **Note:** This security setup assumes that LDAP is installed and configured as discussed in Chapter 10, “Securing a production topology” on page 239.

To ensure that users defined in LDAP are visible in WebSphere Business Services Fabric, the role-to-user mapping for the Fabric_Tools application must be updated as follows:

1. Log into the WebSphere Process Server administration console using administrator credentials.
2. Click **Enterprise Applications** → **Fabric_Tools** → **Security role to user/group mapping**.
3. Select the check box next to the role **FabricBasicUser** and click **Look up users**.

4. You will be presented with a list of users fetched from LDAP and listed on the left side under Available. Add these users to the Selected list by clicking the **>>** button. The end result is shown in Figure 14-37. Click **OK** to complete the mapping of the LDAP user IDs to the role FabricBasicUser.

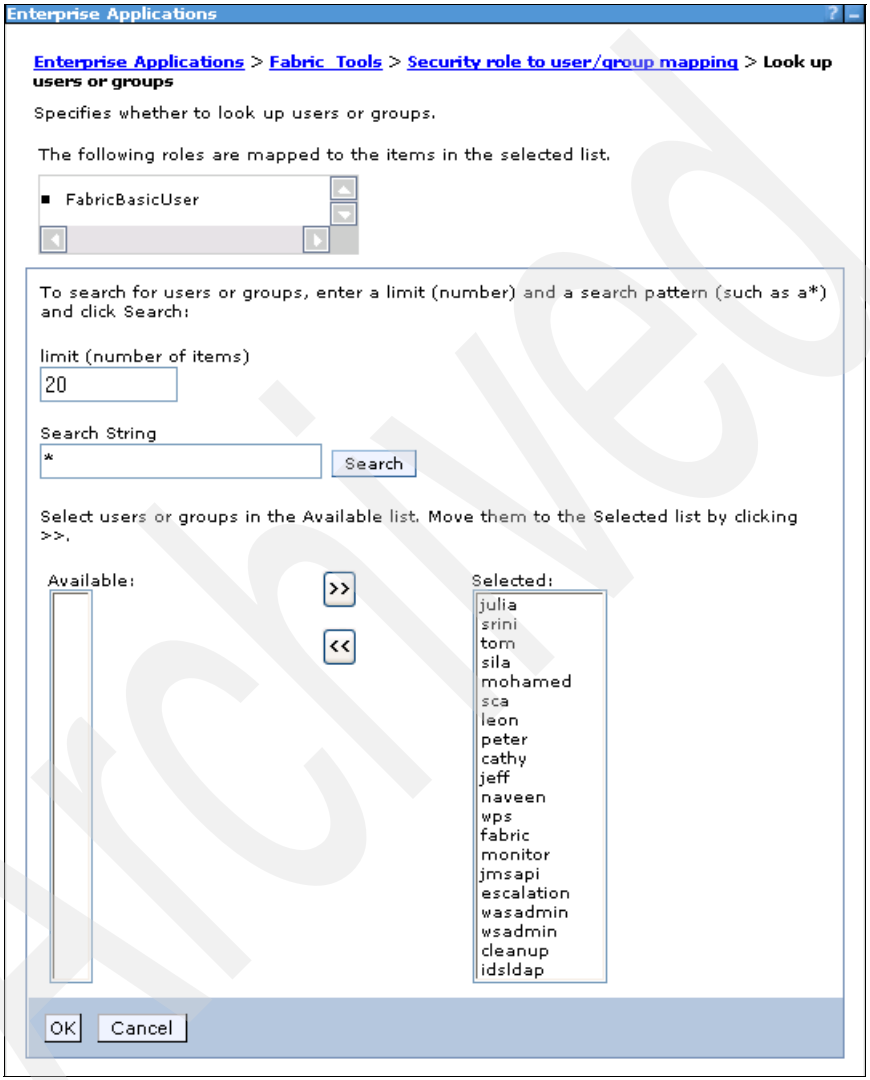


Figure 14-37 Mapping users to the FabricBasicUser role

14.4.5 Manage enrollments and subscriptions

The organization ITSOBankOrg is defined in WebSphere Business Services Fabric to represent the ITSOBank organizational structure. Ensure that the ITSOBank is enrolled for the business services provided by the composite business application:

1. Log into the WebSphere Business Services Fabric console and navigate to the Subscriber Manager twisty in the left pane. Collapse the twisty by clicking the **[+]** icon, then click **Manage Subscribers**.
2. Click **ITSOBankOrg**, then click **Enrollments**. The page displays a list of available business services. Ensure that the check box next to the business service **LoanProcessBS** is checked. If it is not, click the check box then click **Save Enrollments**.
3. Click **Manage Subscriptions** in the left pane. Click the radio button to select **ITSOBank**.
4. In the Search For box enter the first one or two letters of a user's last name. Click **Search**. WebSphere Business Services Fabric will contact the LDAP repository and conduct a wildcard search.
5. From the list of users returned click the target user. Click the icon with the right-facing arrow to add the user to the Selected Users list.
6. Click the check box for **ITSOBankLoanApp**. This action also selects the nested business service named LoanProcessBS and the channel Loan Portal Channel.
7. Click **Subscribe Users** to complete the subscription of the selected user to the LoanProcessBS business service.

14.4.6 Running the vehicle loan process application

The vehicle loan application is implemented as a Business Process Execution Language (BPEL) process. The simplest way to execute this process is by way of the Business Process Choreography (BPC) Explorer tool:

1. Access the BPC facility with the following URL:

   ```
   http://<hostname>:<port>/bpc
   ```

   For example, the BPC used for this book is deployed at:

   ```
   http://blade41.itso.ibm.com:9080/bpc
   ```

2. At the login dialog enter the credentials for the WebSphere Process Server administrator.

3. Click **Process Templates** in the View pane. From the Process Templates list click the check box next to the template **InvokeLoanProcess**, as illustrated in Figure 14-38. This process is a harness that invokes the actual process with the name VehicleLoanProcess.

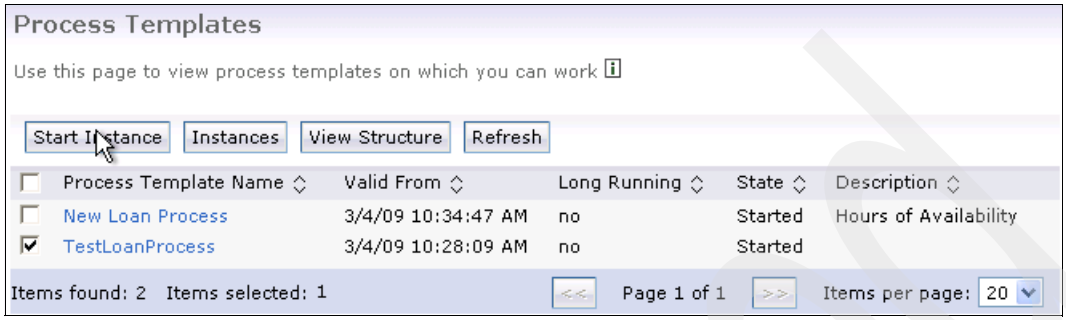


Figure 14-38 Process templates in the BPC

4. Click **Start Instance**. This results in a page with a form to be populated with input parameters for the process, as shown in Figure 14-39.

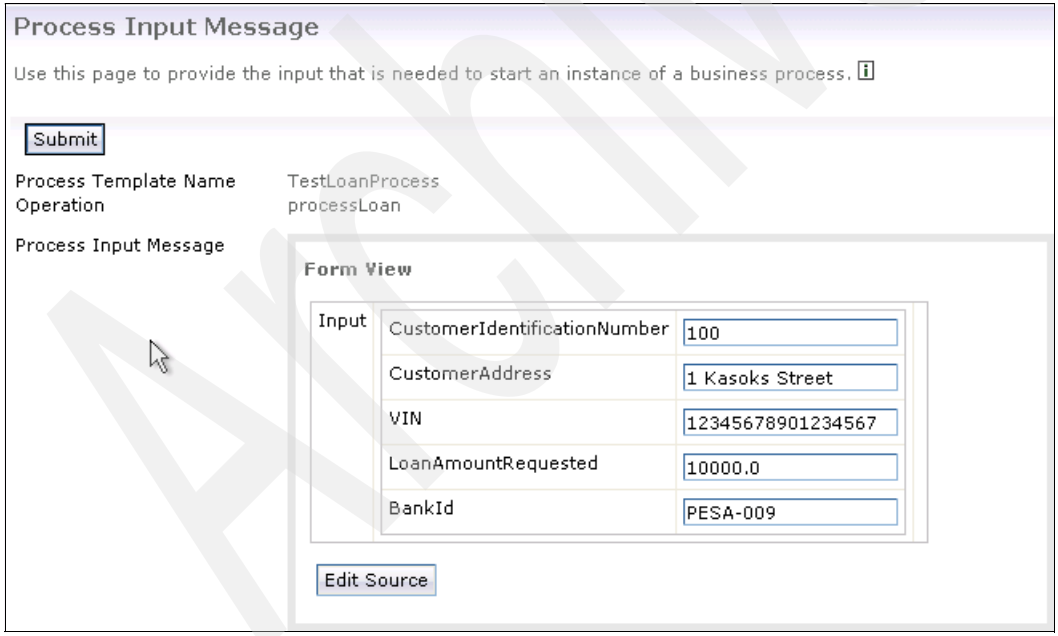


Figure 14-39 Sample input to the ITSOBank loan application

5. Click **Submit**. On successful completion you will get a panel similar to the one shown in Figure 14-40.

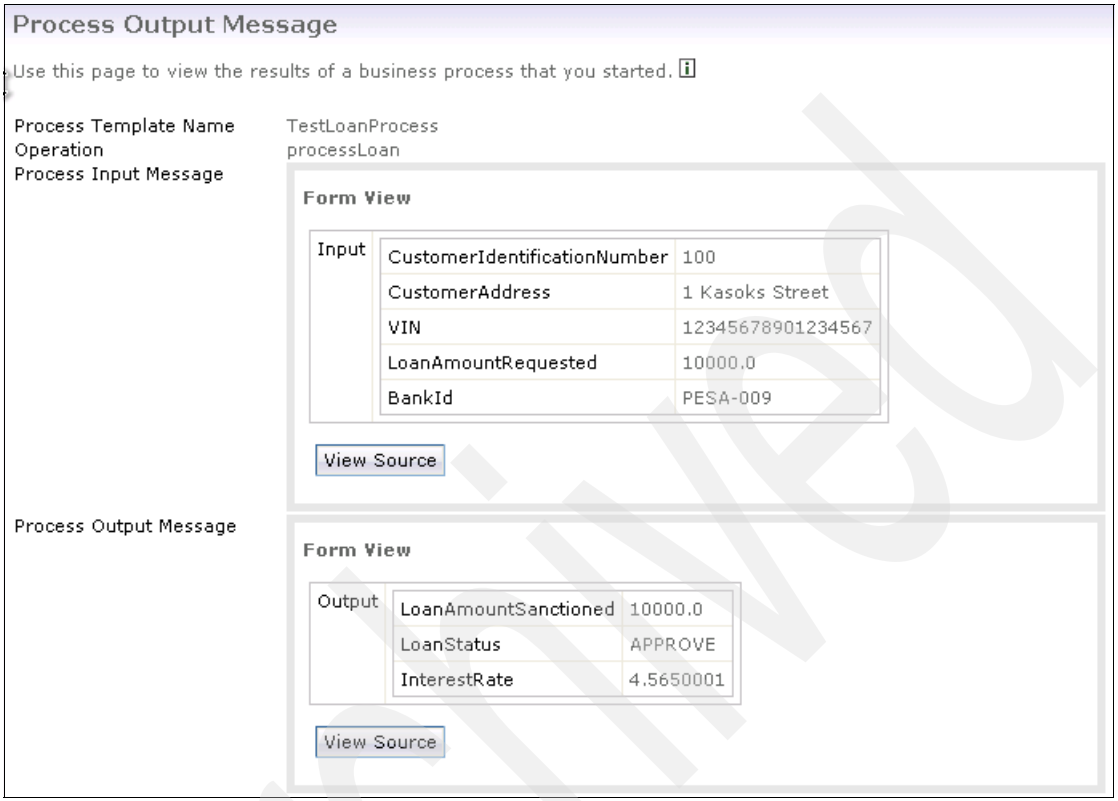


Figure 14-40 Output message from the business process

15

Incorporating WebSphere Business Monitor into a production topology

This chapter guides you through the necessary steps for installing WebSphere Business Monitor in a distributed clustered topology using an existing WebSphere Process Server Remote Messaging and Remote Support topology pattern.

Note: The topology described in this chapter is recommended if you want to extend an existing WebSphere Process Server Remote Messaging and Remote Support topology.

For newly created production topologies based on WebSphere Process Server, we recommend that you use the Four Cluster topology described in Chapter 16, “Creating a WebSphere Dynamic Process Edition production topology” on page 509.

15.1 Overview

WebSphere Business Monitor is a comprehensive business-activity monitoring solution that provides a near real-time view of your business performance. It measures business performance, monitors runtime and completed processes, and reports on business operations by processing events, calculating business metrics, and presenting key performance indicators (KPIs) through business dashboards.

WebSphere Business Monitor also provides capabilities to detect business situations, issues related alerts, and graphically presents business information. When something goes wrong, alerts can be delivered to make the organization aware of potential problems, allowing a directed action to be planned. This helps the organization to identify business problems, correct exceptions, and change processes accordingly.

Figure 15-1 shows the topology implemented in this chapter. Note that it is an extension of the Remote Messaging and Remote Support topology created for WebSphere Process Server.

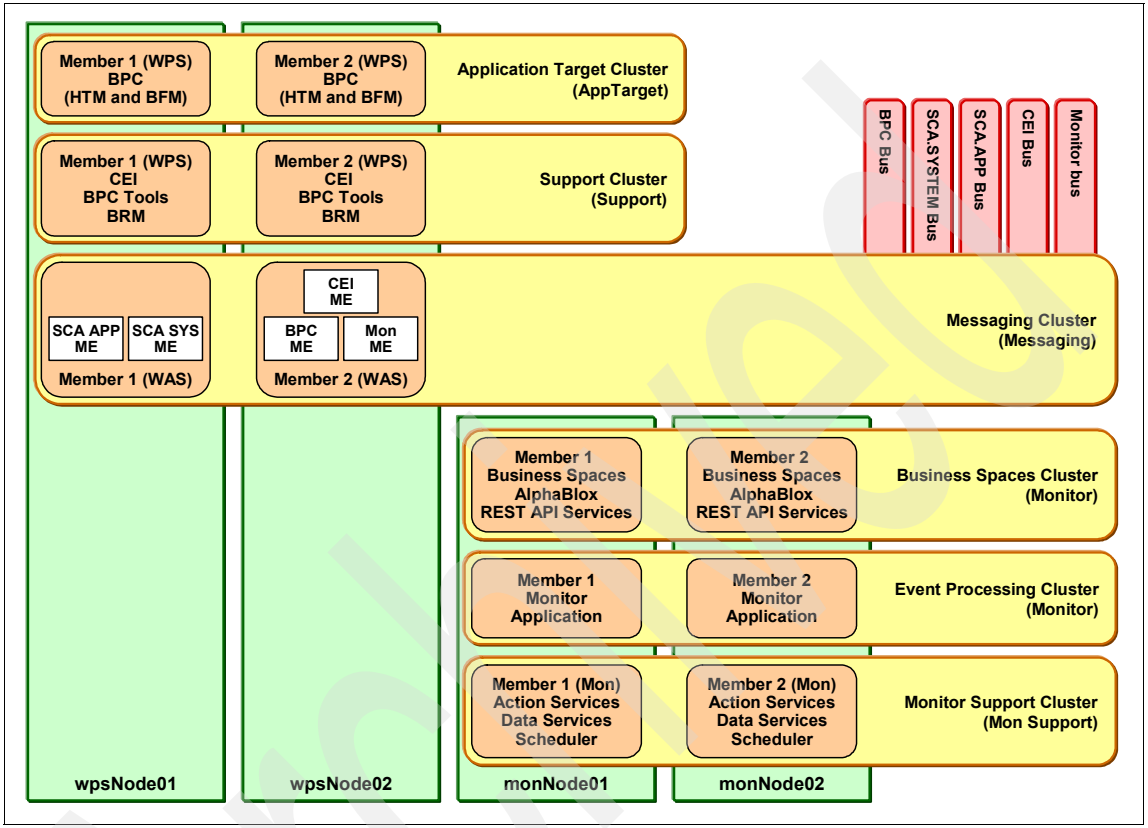


Figure 15-1 WebSphere Business Monitor deployment scenario

15.2 Considerations for WebSphere Business Monitor installation

This section outlines the installation considerations for WebSphere Business Monitor.

15.2.1 Prerequisite software

This chapter describes how to install WebSphere Business Monitor into an existing Remote Messaging and Remote Support topology pattern for

WebSphere Process Server. The topology described in this chapter used the following versions:

- WebSphere Application Server V6.1.0.21
- WebSphere Process Server V6.2
- DB2 UDB ESE V8.2 fix pack 8 OR V9.1 fix pack 5 OR V9.5 fix pack 1
- IBM Tivoli Directory Server V6.0 or V6.1

Note: Detailed step-by-step instructions for building the Remote Messaging and Remote Support topology pattern in WebSphere Process Server are provided in Chapter 7, “Configuring Remote Messaging and Remote Support” on page 165.

The following software should be available before starting installation:

- WebSphere Business Monitor V6.2
- Alphablox V9.5.2

For more information regarding the supported WebSphere Business Monitor products, refer to the following Web page:

http://www-01.ibm.com/software/integration/wbimonitor/requirements/

15.2.2 Installation overview

The installation steps in this chapter were performed in SUSE Linux Enterprise Server 10 SP1. The topology contains the following clusters:

- Monitor Support cluster

  The monitor support cluster was created for WebSphere Business Monitor. It contains the Monitor action services, Monitor emitter service (EmitterRestServices.ear), and the data movement service.

- Business Space cluster

  This cluster was created for WebSphere Business Monitor. It contains the REST service, Alphablox, mobile dashboards (optional), and Business Space. In the original WebSphere Process Server Remote Messaging and Remote Support topology, Business Space was deployed to the Support cluster. We intend to move it to the Business Space cluster in this chapter.

- Event Processing cluster

  The event processing cluster consists of both the Monitor Model Moderator module and the Monitor Model Logic module.

- Messaging cluster

  The messaging cluster is part of the WebSphere Process Server Remote Messaging and Remote Support topology. It contains the service integration buses and messaging engines. This cluster has been extended to include a Monitor service integration bus.

- Support cluster

  The support cluster is part of the WebSphere Process Server Remote Messaging and Remote Support topology. It contains support applications for WebSphere Process Server and the Common Event Infrastructure (CEI).

- Application Target cluster

  The application target cluster is part of the WebSphere Process Server Remote Messaging and Remote Support topology. It contains the Business Process Choreographer (BPC) components.

Note: In this chapter, the WebSphere Business Monitor clusters are installed into the same cell as WebSphere Process Server. This single cell topology ensures that a cross link between a WebSphere Process Server cell and WebSphere Business Monitor cell is not required.

15.3 Installation of WebSphere Business Monitor distributed topology

In the following sections we describe in detail the required steps to install WebSphere Business Monitor cluster, extending existing WebSphere Process Server deployment.

15.3.1 Installing WebSphere Business Monitor software

Install WebSphere Business Monitor in the `/opt/ibm/WebSphere/ProcServer` directory on the Monitor Deployment manger machine.

1. As *root* use the Install Wizard to install WebSphere Business Monitor media to the `/opt/ibm/WebSphere/ProcServer` directory. This adds WebSphere Business Monitor to the WebSphere Process Server currently installed and configured.
2. Verify that the Deployment Manger is stopped.

3. Run <media install root >/WBM/install:
   a. Click **Next**.
   b. Select **I accept both the IBM and the non IBM terms**.
   c. Click **Next** → **Next**.
   d. Select **Advanced Installation**, deselect Information Center, and then click **Next**.
   e. Use the existing version of WebSphere Application Server Network Deployment. The value should be /opt/ibm/WebSphere/ProcServer. Click **Next**.
   f. Select **None** and click **Next**.
   g. Click **Yes** to the pop-up warning about not creating a profile.
   h. Verify your Installation Summary and click **Next**, then click **Finish**.
4. Run versionInfo.sh to verify your installation for the deployment manager, as shown in Example 15-1.

Example 15-1 versionInfo.sh

```
/opt/ibm/WebSphere/ProcServer/bin/versionInfo.sh
Installed Product
Name                        IBM WebSphere Application Server - ND
Version                     6.1.0.21
ID                          ND
Build Level                 cf210844.13
Build Date                  11/6/08

Installed Product
Name                        WebServices Feature Pack
Version                     6.1.0.21
ID                          WEBSERVICES
Build Level                 cf210844.03
Build Date                  11/6/08

Installed Product
Name                        IBM WebSphere Process Server
Version                     6.2.0.0
ID                          WBI
Build Level                 of0847.11
Build Date                  11/26/08

Installed Product
Name                        IBM WebSphere Business Monitor
Version                     6.2.0.0
```

```
ID                          WBM
Build Level                 gm0848.05
Build Date                  12/2/08
```

15.3.2 Creating the WebSphere Business Monitor databases

WebSphere Business Monitor uses a single database for persistence. The default name is MONITOR. We created this database on the same DB2 server used by the existing Remote Messaging and Remote Support topology. To create the WebSphere Business Monitor database:

1. To prepare the database creation scripts for execution:
   a. Locate the script createDatabaseDb2.ddl, found at:

      `/opt/ibm/WebSphere/ProcServer/scripts.wbm/database`

   b. Edit the following variables in the createDatabaseDb2.dll script:
      - $DBNAME$: This variable represents the name of the Monitor database. (Use MONITOR.)
      - $SCHEMA$: This variable represents the name of the Monitor schema. (Use MONITOR.)
      - $TSDIR$: This variable represents the tablespace directory. If $TSDIR$ is omitted from the data file specification of a tablespace, the data file will be created in the Database Manager directory. (Use DEFAULTTS.)
      - $TERRITORY$: This variable represents the locale of the data in the database. (Use EN_US.)

   c. Save and close the file as createDatabaseDb2.alter.ddl.
   d. Make the new file executable:

      ```
      chmod 755 createDatabaseDb2.alter.ddl
      ```

2. Open the DB2 command-line interface and run the createDatabaseDb2.dll script using the following command:

   ```
   db2 -tf createDatabaseDb2.alter.ddl
   ```

> **Note:** The user must be the DB2 instance owner or have the SYSADM privilege to create a new database.

3. Bind the command-line interface to the Monitor database using the following commands:

```
db2 connect to MONITOR
db2 bind /home/db2inst1/sqllib/bnd/@db2cli.lst blocking all grant
public
db2 connect reset
```

4. The result of the bind command should be as shown in Example 15-2.

Example 15-2 Bind command result

```
LINE    MESSAGES FOR db2cli.lst
------------------------------------------------------------------
SQL0061W  The binder is in progress.
SQL0091N  Binding was ended with "0" errors and "0" warnings.
```

15.4 Building WebSphere Business Monitor profiles

This section describes how to prepare and build WebSphere Business Monitor custom profiles.

15.4.1 Preparing to build profiles

In preparation to augment and create the profiles for this configuration, you must collect information in advance. Table 15-1 contains the information that you will need prior to starting the process.

Table 15-1 Information needed before starting profile creation process

Name	Value
Cell security ID	wps
Cell security ID password	passw0rd
Database Name	MONITOR
Database Schema name	MONITOR
Database User Name	db2inst1
Database User Password	passw0rd
Location (Directory) of JDBC driver classpath files on Dmgr host	/opt/ibm/WebSphere/ProcServer/universal.wmb/lib

Name	Value
JDBC Driver type	4
Database host name	svslesvm
Database TCP service port	50001
Deployment Manager host name	svslesvm
DMGR SOAP Port	8879
Monitor Cluster Node Hostname	clusterm
Location (Directory) of JDBC driver classpath files on monitor host	/opt/ibm/WebSphere/ProcServer/universal.wmb/lib

15.4.2 Augment the Deployment Manager

After installing binaries of WebSphere Business Monitor within the same cell of WebSphere Process Server, the next step is to create the WebSphere Business Monitor deployment manager profile. By augmenting the deployment manager profile, both the WebSphere Process Server and WebSphere Business Monitor deployment managers are in one profile. Using this profile, we can control and administer all WebSphere Business Monitor and WebSphere Process server clusters.

1. Run the profile management tool:

   ```
   /opt/ibm/WebSphere/ProcServer/bin/ProfileManagement # ./pmt.sh
   ```

2. Click **Augment and existing profile**, then click **Next**.
3. Select the **Dmgr01** profile. This is your current WebSphere Process Server deployment manager profile. Click **Next**.

4. Select **WebSphere Business Monitor deployment manger**, as shown in Figure 15-2. Click **Next**.

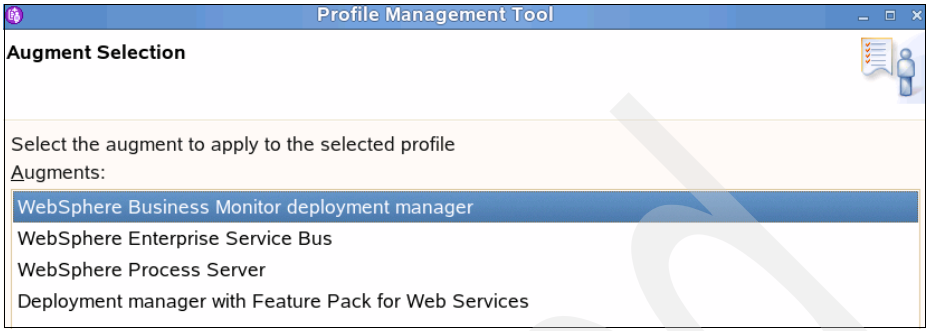


Figure 15-2 Augment profile to WebSphere Business Monitor deployment manager

5. Select **Advanced profile augmentation**. This enables us to manually configure the monitor database and credentials. Click **Next**.
6. Since the WebSphere Process Server environment has security enabled, enter the deployment manager security credentials. These values should be wps for the user name and passw0rd for the password (Figure 15-3). Click **Next**.

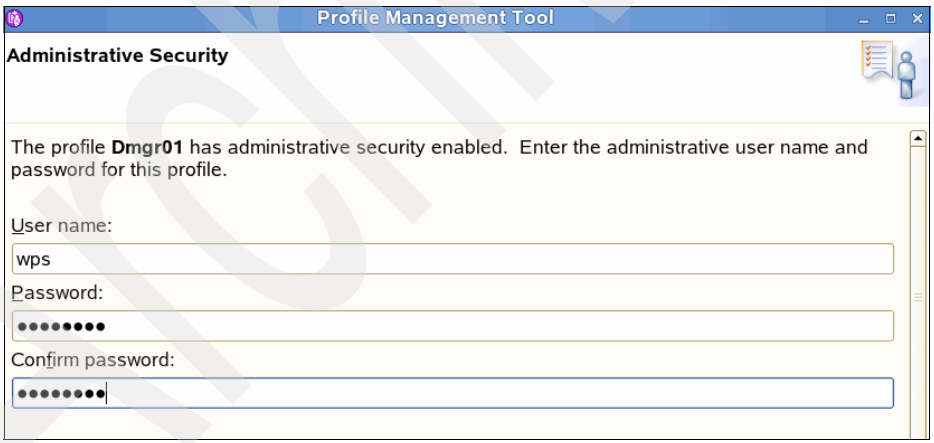


Figure 15-3 Security panel of Profile Management Tool

7. Select the **Use existing database** radio button and confirm that the database name and schema match the values entered in Creating the WebSphere Business Monitor databases. In this book, the values should reflect those show in Figure 15-4. Click **Next**.

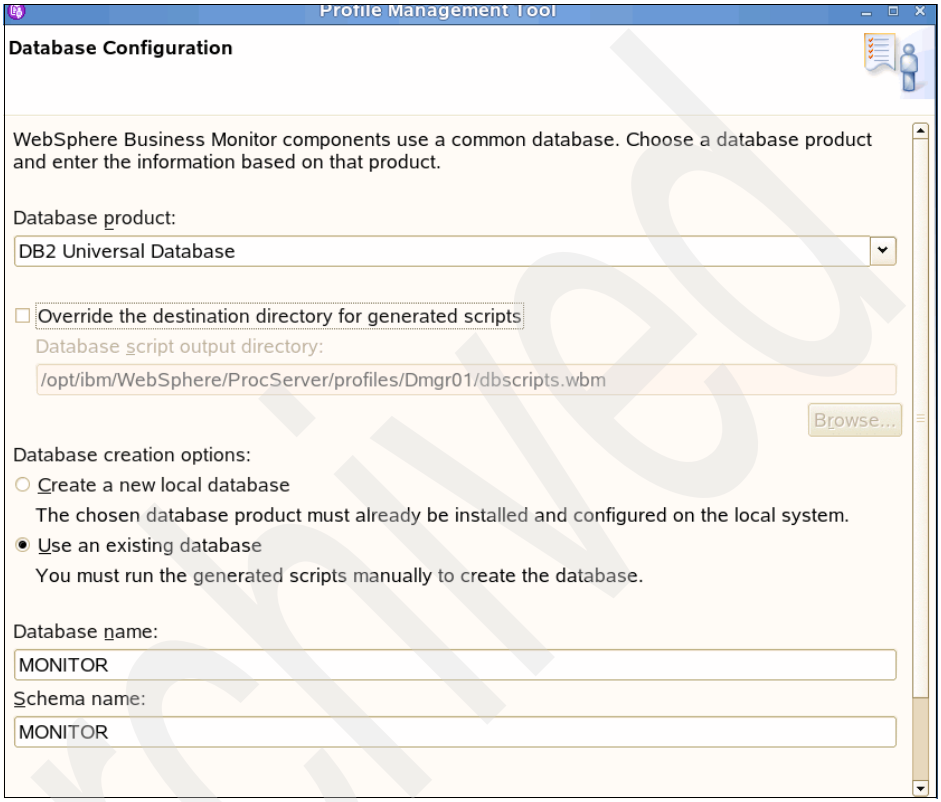


Figure 15-4 Database Configuration panel of PMT tool

8. On Database Configuration panel (Part 2), enter the values in Table 15-2.

Table 15-2 values for Database Config part 2

Name	Value
User Name	db2inst1
Password	passw0rd
Location (Directory) of JDBC driver classpath files	/opt/ibm/WebSphere/ProcServer/universal.wmb/lib
JDBC Driver type	4

Name	Value
Database host name	svslesvm
Database TCP listener port	50001

The panel should resemble Figure 15-5. Click **Next**.

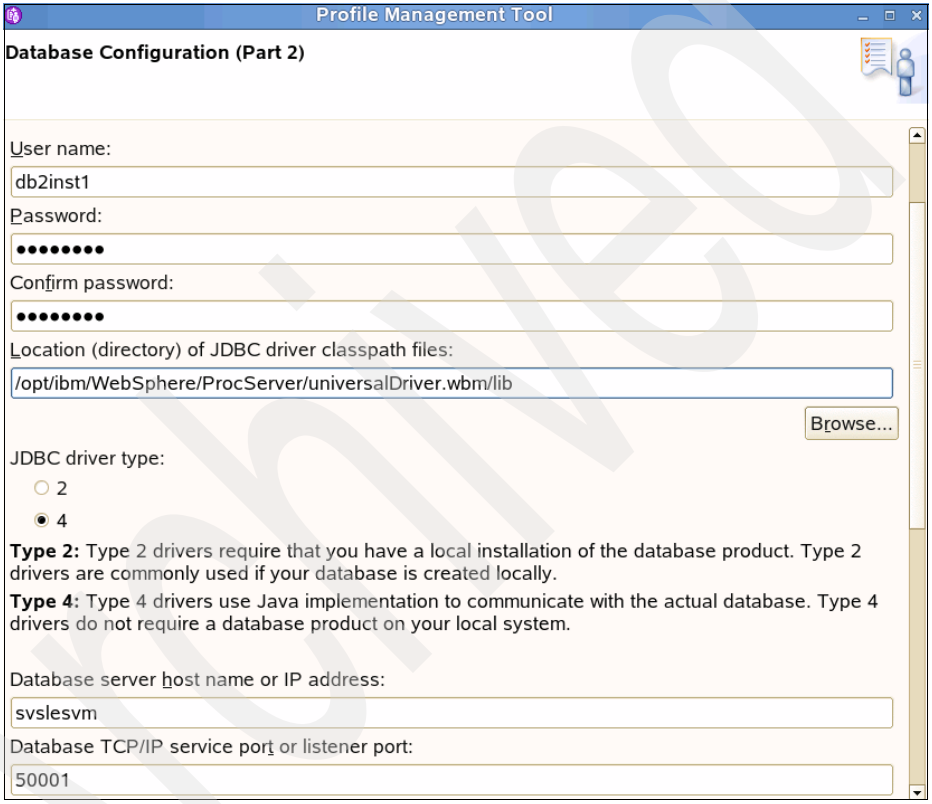


Figure 15-5 Data Configuration (Part 2) panel

9. Review the Profile Augmentation Summary panel. This panel should resemble Figure 15-6. Click **Augment**.

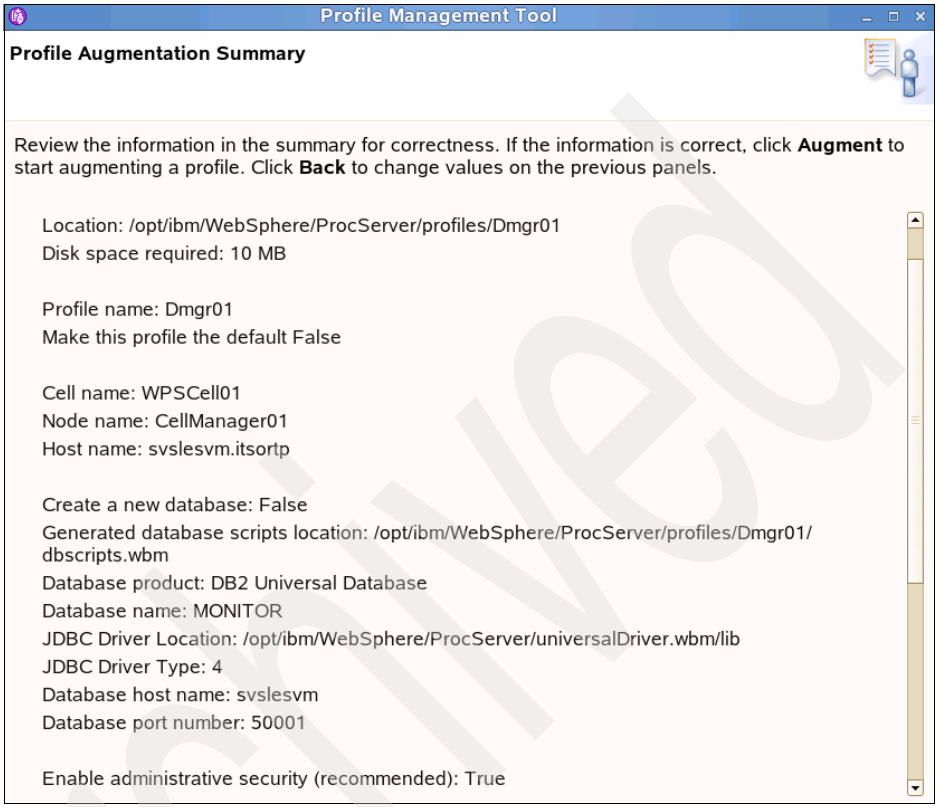


Figure 15-6 Profile Augmentation Summary

10. Verify that the augmentation is successful. The Profile Augmentation Complete Panel should be successful. The panel should resemble Figure 15-7.

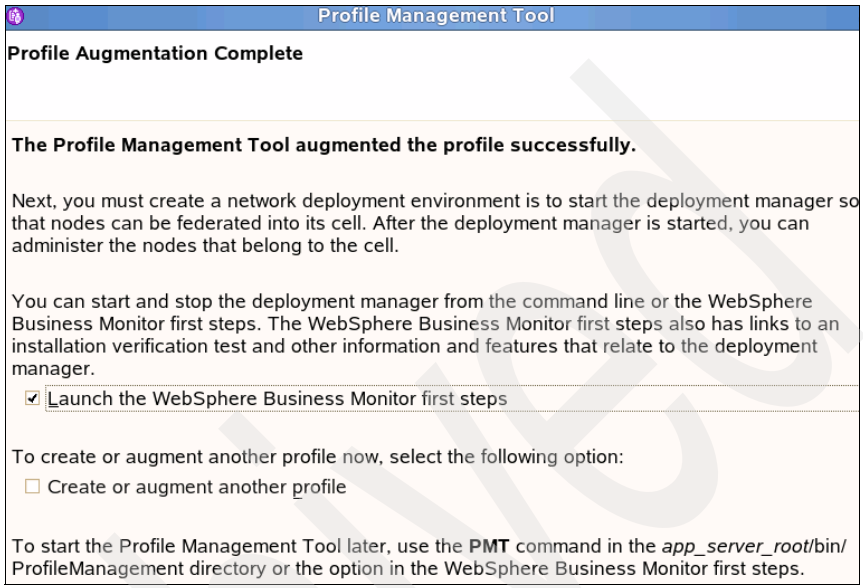


Figure 15-7 Profile Augmentation Complete

11. Keep the **Launch the WebSphere Business Monitor first steps** box checked. Click **Finish**. This launches the First steps panel.

12. Click **Install verification**. This launches a command window to verify the installation/augmentation of the deployment manager.

    a. You are prompted for the user name (wps) and password (passw0rd). The output window produces the text shown in Example 15-3.

Example 15-3 Install verification output

```
Start Monitor Verification Test

Start deployment manager server if it hasn't been started

WASX7209I: Connected to process "dmgr" on node CellManager01
using SOAP connector;  The type of process is: DeploymentManager

  - Installation Verification Utility Testing
    - Verify Monitor Components
      - install.adminConsole.wbm passed
      - install.configactions.wbm passed
```

```
        - install.monServer.wbm passed
        - legal.wbm passed
        - nif.componentmap.wbm.all passed
        - pmt.profileTemplate.wbm passed
        - pmt.plugins.wbm passed
      - Summary: Monitor components verification is successful

    - Basic Monitor Install Verification
      - Verify JDBC Drivers
        - MonitorDBProvider passed
      - Verify Data Sources
        - Monitor_Admin_Database passed
        - Monitor_Database passed
      - Verify Authentication Entries
        - Monitor_JDBC_Alias passed
        - MonitorBusAuth passed
        - MonitorQueueConnectionFactoryAuth passed
      - Verify Other Resources
        - MonitorLifecycleWorkManager passed
        - Action Services ActivationSpec passed
        - Action Services QueueConnFactory passed
      - Summary: Basic monitor installation verification is
successful

    - Verify Database Connection
      - Verify Data Source Connection
        - Monitor_Admin_Database passed
        - Monitor_Database passed
      - Summary: Database testing is successful

    - Summary: Installation verification is successful
```

b. The deployment manager augmentation is compete successfully. Click **Exit** on the First steps panel.

15.4.3 Create WebSphere Business Monitor custom profiles

After installing the cluster members binaries you should create a profile for each cluster member node and federate it to the WebSphere Business Monitor deployment manager. This section describes the required steps for creating node profiles and federating them into the deployment manager.

Notes: Creating and federating clusters members:

- You cannot create custom WebSphere Process Server nodes with WebSphere Business Monitor and use them with the deployment environments wizard.
- It is a mandatory that the timing between any node machine and the deployment manager machine be less than 5 minutes. If the timing is greater than 5 minutes, the federation of the profile to the deployment manager fails.

To create and federate cluster member nodes:

1. Run the profile management tool:

   `/opt/ibm/WebSphere/ProcServer/bin/ProfileManagement/pmt.sh`

2. In the Profile Management Tool window, click **Create**. Click **Next**.
3. Select **WebSphere Business Monitor** and click **Next**.
4. Select **WebSphere Business Monitor custom profile**. Click **Next**.
5. Select the **Advanced profile creation** radio button, as seen in Figure 15-8. Click **Next**.

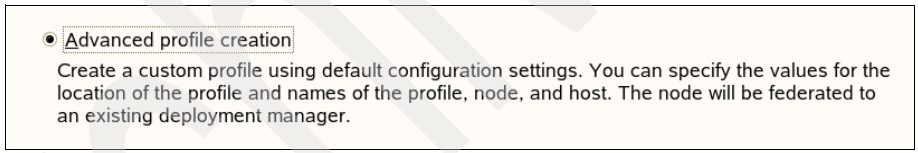


Figure 15-8 Profile creation options

6. Enter a profile name and location. Accept the defaults shown in Figure 15-9. In this panel you might wish to enter a more descriptive name. Click **Next**.

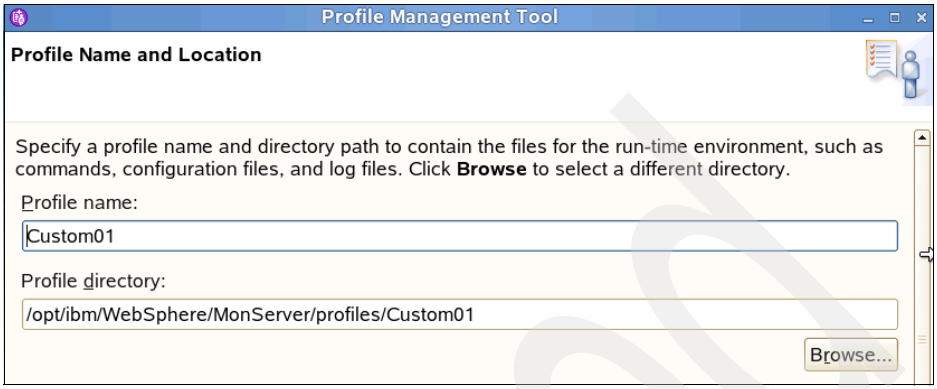


Figure 15-9 Profile Name an Location panel

7. Enter a node name and a host name. Accept the defaults, as shown in Figure 15-10. Click **Next**.

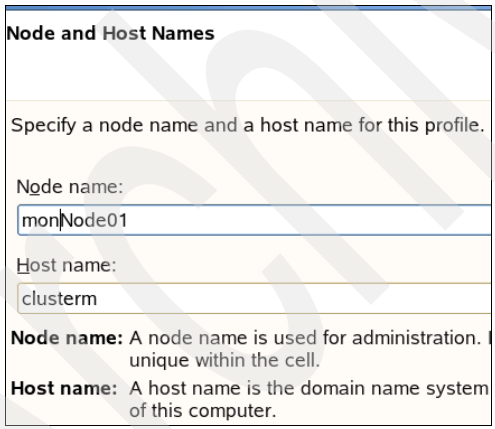


Figure 15-10 Node and Host Names panel

Note: We recommend using the fully qualified domain name as the host name.

8. In the Federation Panel, enter the values shown in Table 15-3.

Table 15-3 Values for Federation Panel

Name	Value
Dmgr Hostname	svslesvm
DMGR SOAP Port	8879
User name	wps
Password	passw0rd

The panel should resemble Figure 15-11. Click **Next**.

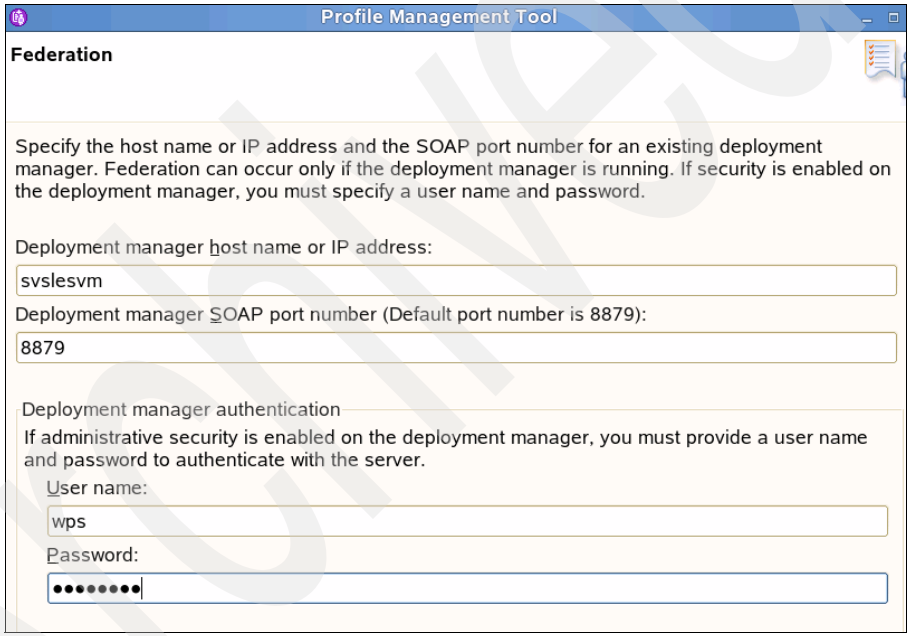


Figure 15-11 Federation Panel

9. This is the Ports panel. Review it, then click **Next**.
10. This is the database Configuration panel. Ensure that the values are correct, then click **Next**.

11. This is the Summary panel, as shown in Figure 15-12. Ensure that the values are correct, then click **Create**.

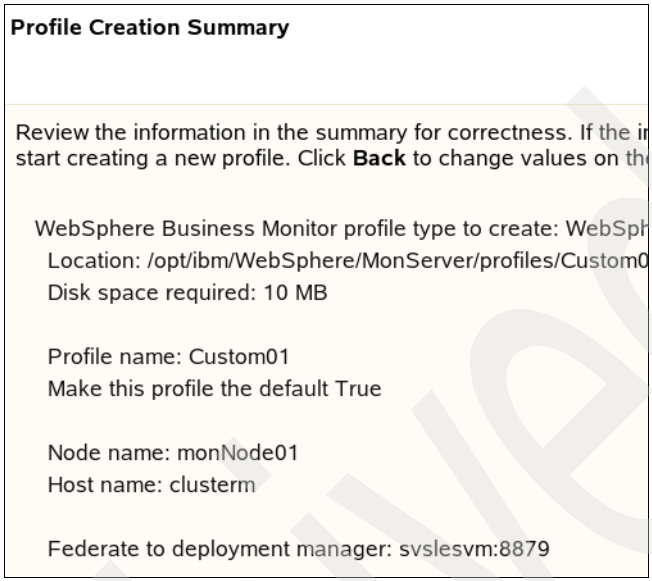


Figure 15-12 Profile Creation Summary

Create a second custom profile. Repeat the previous steps, but make sure to change the profile name and node name in steps 6 and 7.

a. From a terminal window you can issue **`--> ps -elf | grep nodeagent`**. This should produce two Java processes. These are the nodes that you just created and are federated to the deployment manager.

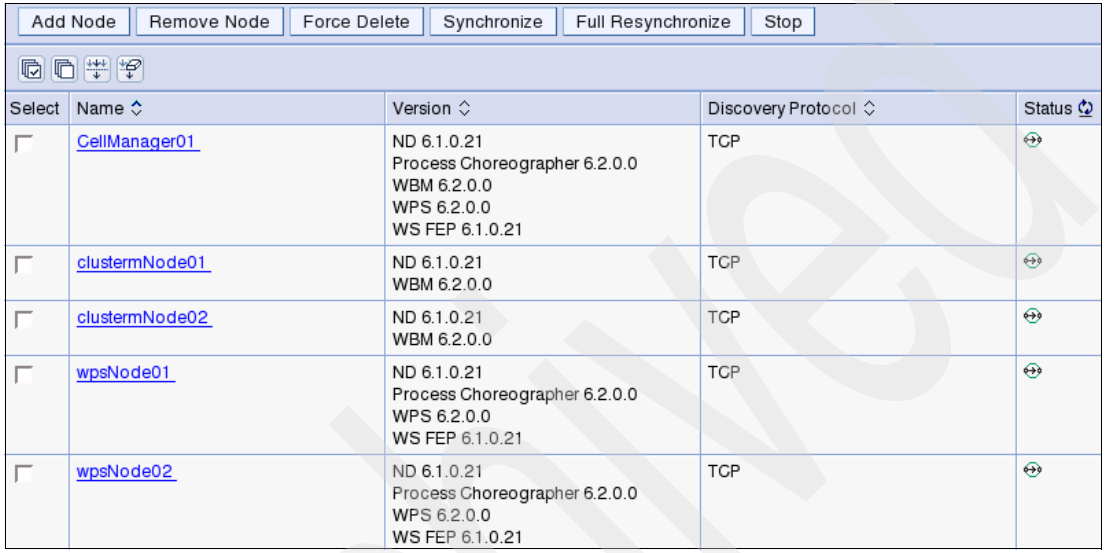
b. Open a Web browser to access the Integrated Solutions Console:

   `http://svslesvm:9060/admin`

c. Enter the username `wps` and password `passw0rd`. Click **login**.

d. Expand **System Administration** and click **Nodes**. Figure 15-13 should show five entries with a status of synchronized:

- CellManager01
- monNode01
- monNode02
- wpsNode01
- wpsNode02



Add Node | Remove Node | Force Delete | Synchronize | Full Resynchronize | Stop

Select	Name	Version	Discovery Protocol	Status
	CellManager01	ND 6.1.0.21 Process Choreographer 6.2.0.0 WBM 6.2.0.0 WPS 6.2.0.0 WS FEP 6.1.0.21	TCP	
	clustermNode01	ND 6.1.0.21 WBM 6.2.0.0	TCP	
	clustermNode02	ND 6.1.0.21 WBM 6.2.0.0	TCP	
	wpsNode01	ND 6.1.0.21 Process Choreographer 6.2.0.0 WPS 6.2.0.0 WS FEP 6.1.0.21	TCP	
	wpsNode02	ND 6.1.0.21 Process Choreographer 6.2.0.0 WPS 6.2.0.0 WS FEP 6.1.0.21	TCP	

Figure 15-13 Nodes Panel Integrated Solutions Console

15.5 Creating the WebSphere Business Monitor clusters

In this section we build clusters for the WebSphere Business Monitor applications to run. There will be three new clusters added to the cell:

- WebSphere Business Monitor support cluster
- Event processing cluster
- Business Space cluster

To create these clusters:

1. Log in to the Integrated Solutions Console as `wps`:

   `http://svsles:9060/admin`

2. Expand **Servers** → **Clusters**.
3. Click **New**.

4. Enter `WBM.Support` for the cluster name. Click **Next** (Figure 15-14).

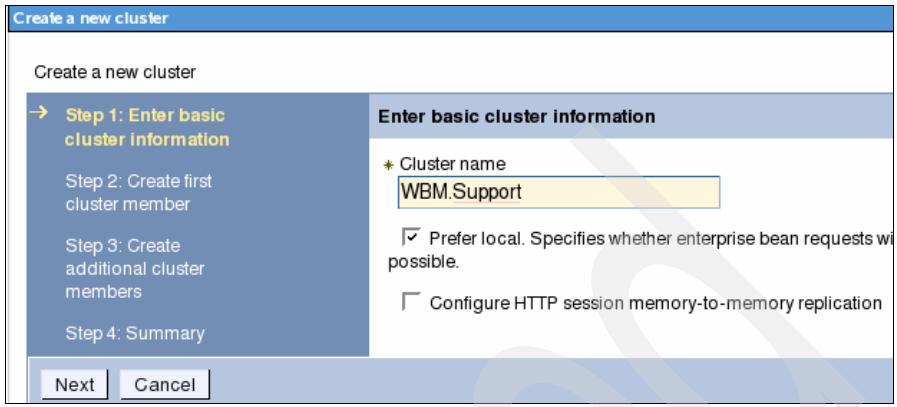


Figure 15-14 New cluster wizard step 1

5. Create a cluster member named `WBMSupportMember01` on the monNode01 using server template default_defaultWBM. Click **Next** (Figure 15-15).

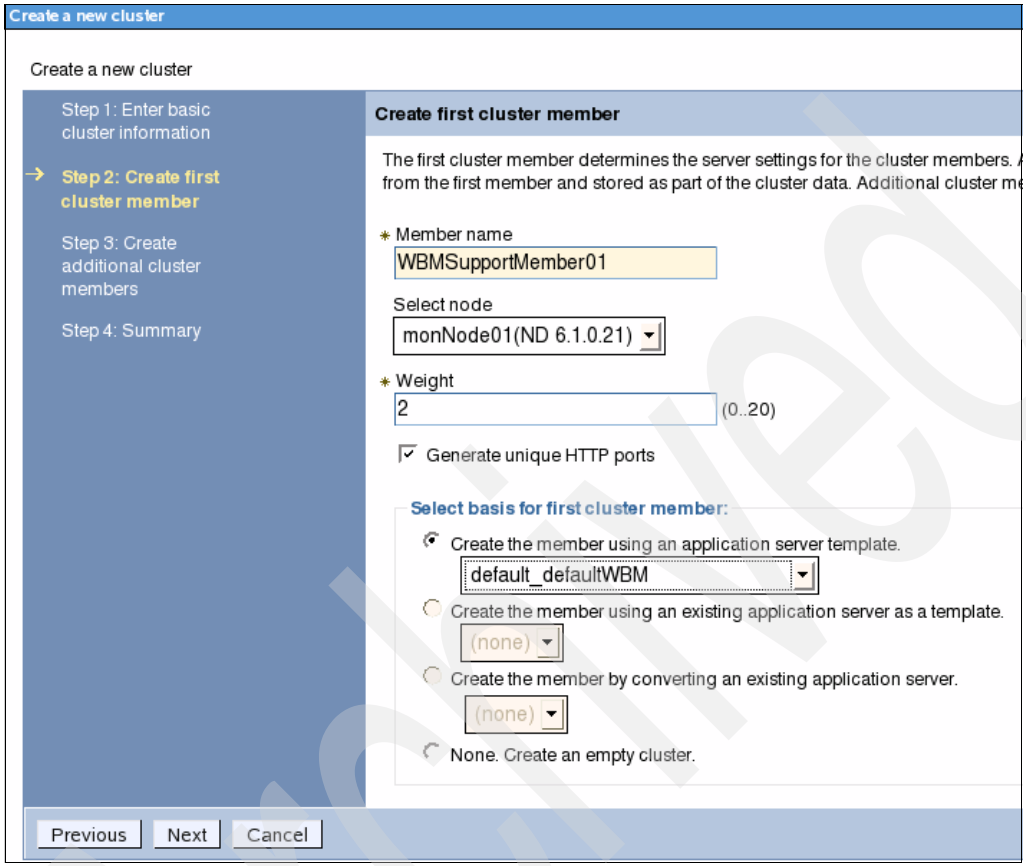


Figure 15-15 New cluster wizard step 2

6. Create a second cluster member using the values in Table 15-4, then click the **Add Member** button. After you have finished creating all of the cluster members click **Next** (Figure 15-16).

Table 15-4 WebSphere Business Monitor support cluster values

Name	Value	SelectNode	Server Template
Cluster Name	WBM.Support		
Member Name	WBMSupportMember01	monNode01	default_defaultWBM
Member Name	WBMSupportMember02	monNode02	default_defaultWBM

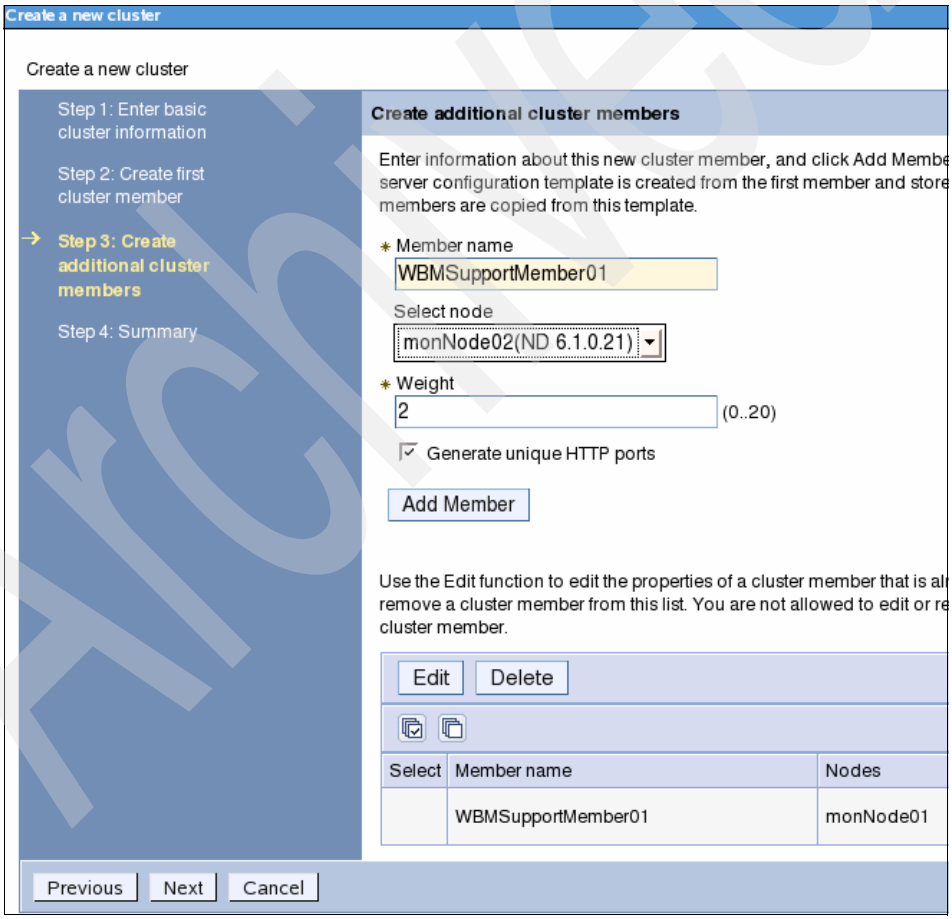


Figure 15-16 New cluster wizard step 4

Note: If you do not click the AddMember button after you enter the second cluster member, your cluster will only have one cluster member. See Figure 15-17.

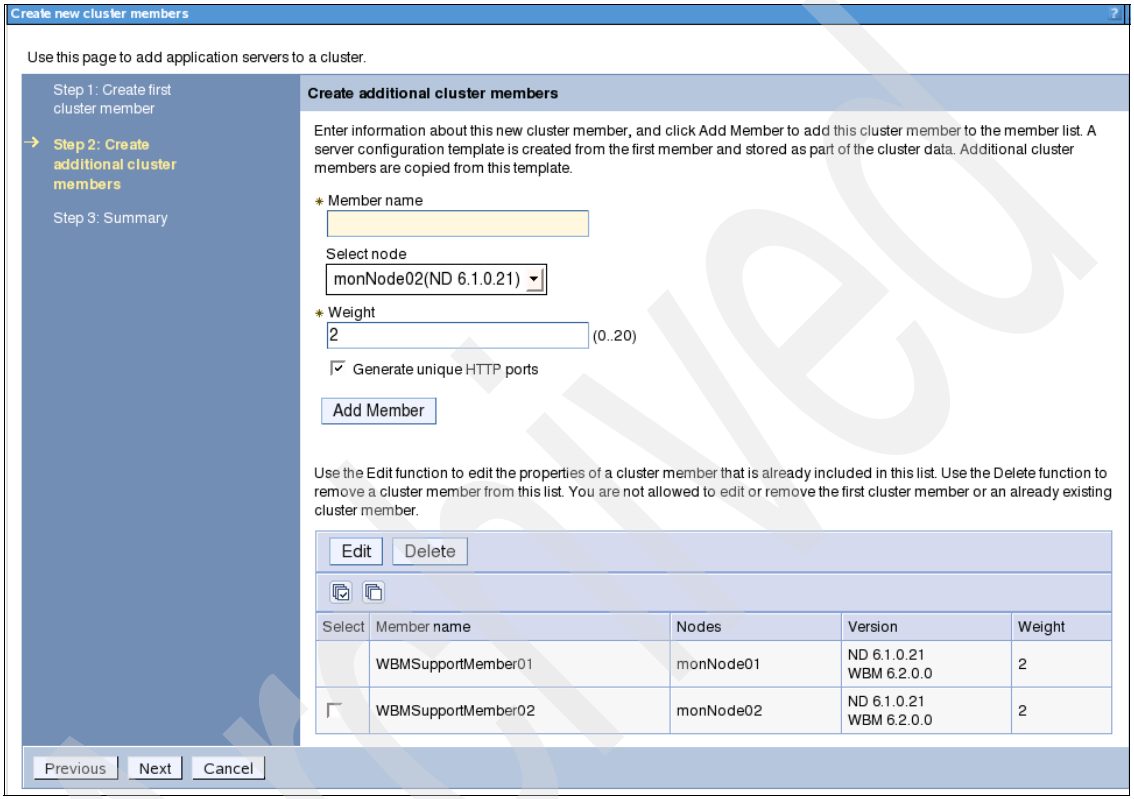


Figure 15-17 New cluster wizard after clicking Add Member

7. Review the summary, then click **Finish**. Save the changes to the master repository (Figure 15-18).

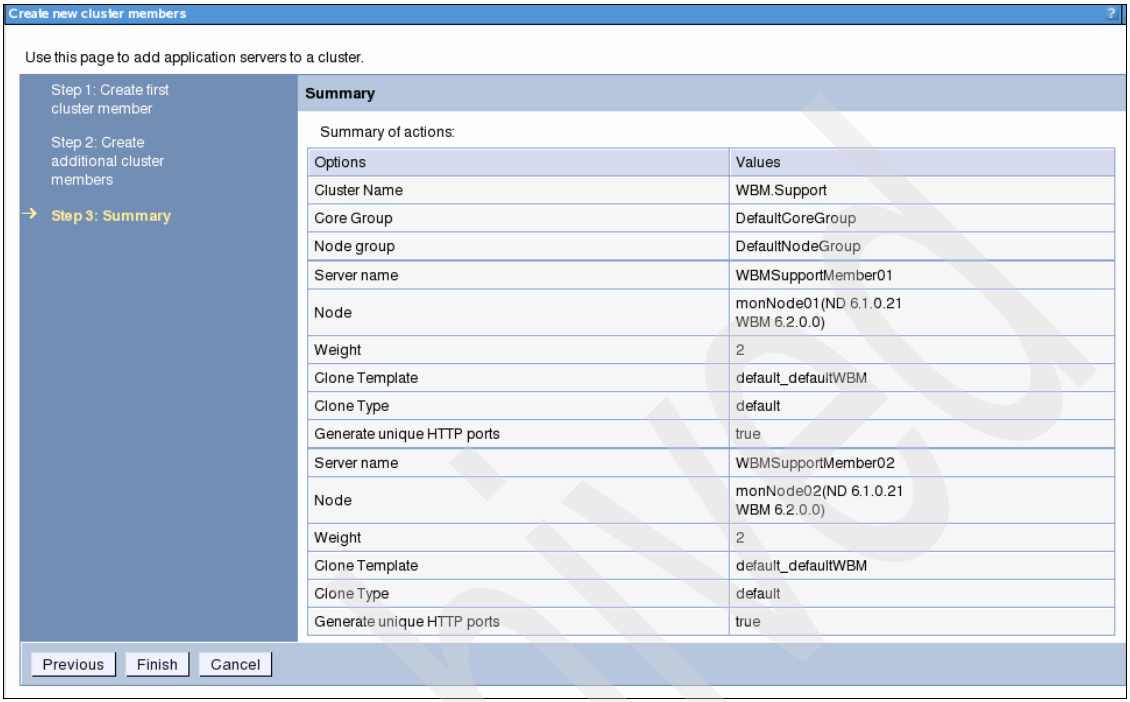


Figure 15-18 Final step of new cluster wizard

8. Repeat the prior steps using the values from Table 15-5 to create the Event Processing Cluster.

Table 15-5 Event processing cluster values

Name	Value	Select node	Server template
Cluster Name	WBM.Event		
Member Name	WBMEventMember01	monNode01	default_defaultWBM
Member Name	WBMEventMember02	monNode02	default_defaultWBM

Repeat the prior steps using the values from Table 15-6 to create the Business Space Cluster.

Table 15-6 Business Space cluster values

Name	Value	Select Node	Server Template
Cluster Name	WBM.BusSpace		
Member Name	WBMBusSpaceMember 01	monNode01	default_defaultWBM
Member Name	WBMBusSpaceMember 01	monNode02	default_defaultWBM

Once the three clusters are completed, the main panel of the Integrated Solutions Console's cluster panel should look like Figure 15-19.

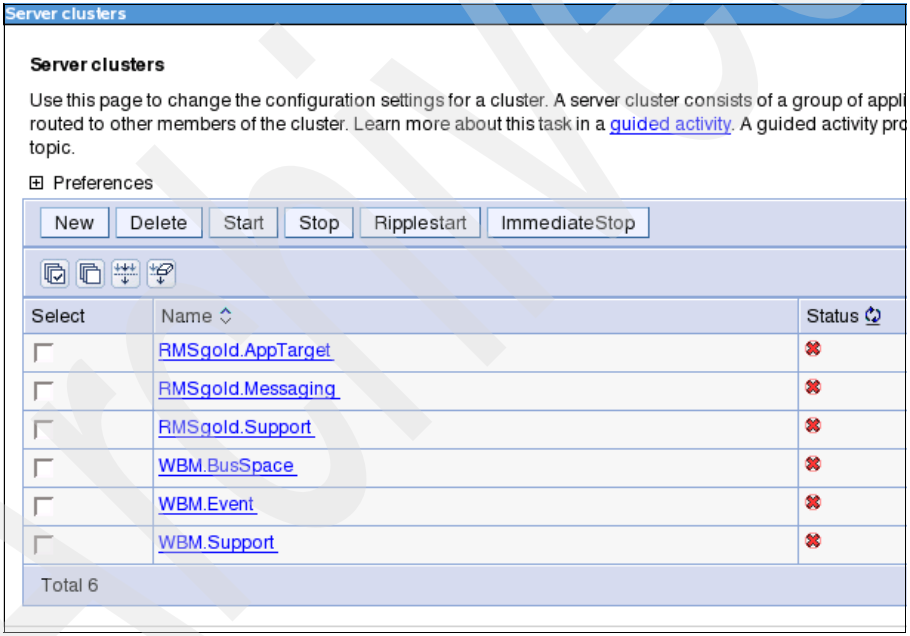


Figure 15-19 Server Clusters panel of Integrated Solutions Console

15.6 Configuring the WebSphere Business Monitor infrastructure

This section provides some configuration steps to perform on the WebSphere Business Monitor infrastructure:

- Enable CEI Server.
- Create a data source for the Monitor Messaging Engine.
- Using the WebSphere Business Monitor configuration panel.

15.6.1 Enable CEI Server

Using the Integrated Solutions Console, enable the CEI Server on the WBM.Support Cluster:

1. Expand **Servers** → **Clusters** → **WBM.Support** → **Common Event Infrastructure** → **Common Event Infrastructure Server**.
2. Check **Enable Common Event Infrastructure server**.
3. Change the Common Event Infrastructure Event Database to Event and deselect **Create Tables**.

4. Select the Remote radio button in the Common Event Infrastructure Bus Member location box and select **Cluster=RMSgold.Messaging**. The panel should resemble Figure 15-20.

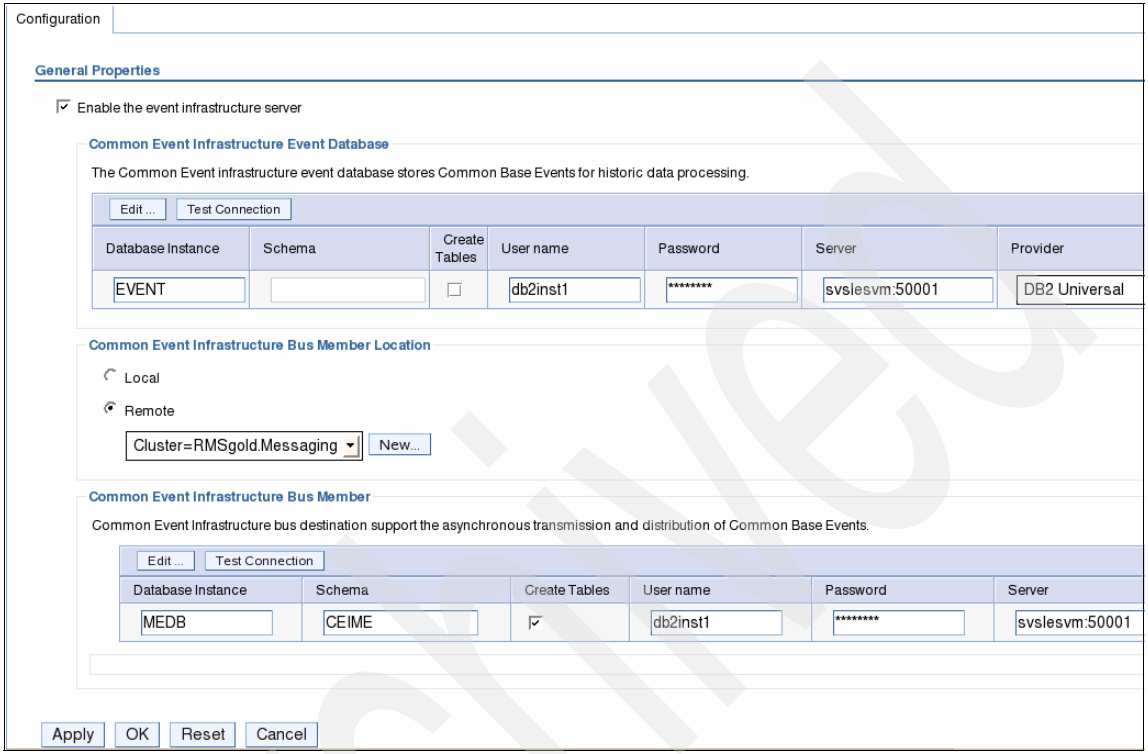


Figure 15-20 Configure Common Event Infrastructure Server

15.6.2 Create a data source for the Monitor Messaging Engine

In the topology, all of the messaging engines will be defined on the RMSGold.Messaging cluster. You must create a data source for the Monitor Messaging Engine data store. In this book, we create a new schema in the MEDB. If you wish to use an existing data source from the other Messaging Engines you can. This would not be a best practice. The best practice is to create its own data source for isolation.

Use the Integrated Solutions Console to create a new data source following these steps:

1. First Create a JAAS - J2C authentication data alias:
   a. Expand **Security** → **Secure administration, applications, and infrastructure** → **authentication** → **Java Authentication and Authorization Service** → **J2C authentication data**.
   b. Click **New**.
   c. Enter the values in Table 15-7 into the new J2C Authentication. Click **OK**.

   Table 15-7 Values to create Monitor ME Authentication Alias - Monitor ME Datastore

Name	Value
Alias	Monitor ME Auth Alias
User ID	db2inst1
Password	passw0rd
Description	Monitor ME datastore

2. Create a new data source:
   a. Expand **Resources** → **JDBC** → **Datasources**.
   b. Change the scope to **Cluster = RMS.Messaging**.
   c. Click **New**.
   d. In the Create new Datasource wizard enter the values in Table 15-8. Click **Next**.

   Table 15-8 DataSource values

Name	Value
Data Source Name	MonitorME Database
JNDI Name	jdbc/monitiorMEDB
Component-managed authentication alias and XA recovery authentication alias	CellManager01/ Monitor ME AuthAlias

   e. Select the **Select an existing JDBC Provider** radio button and **DB2 Universal JDBC Driver Provider** from the drop-down list box. Click **Next**.

f. Enter the values in Table 15-9. Click **Next**.

Table 15-9 Database-specific properties

Name	Value
Database Name	MEDB
Driver type	4
Server name	svslesvm
Port Number	50001

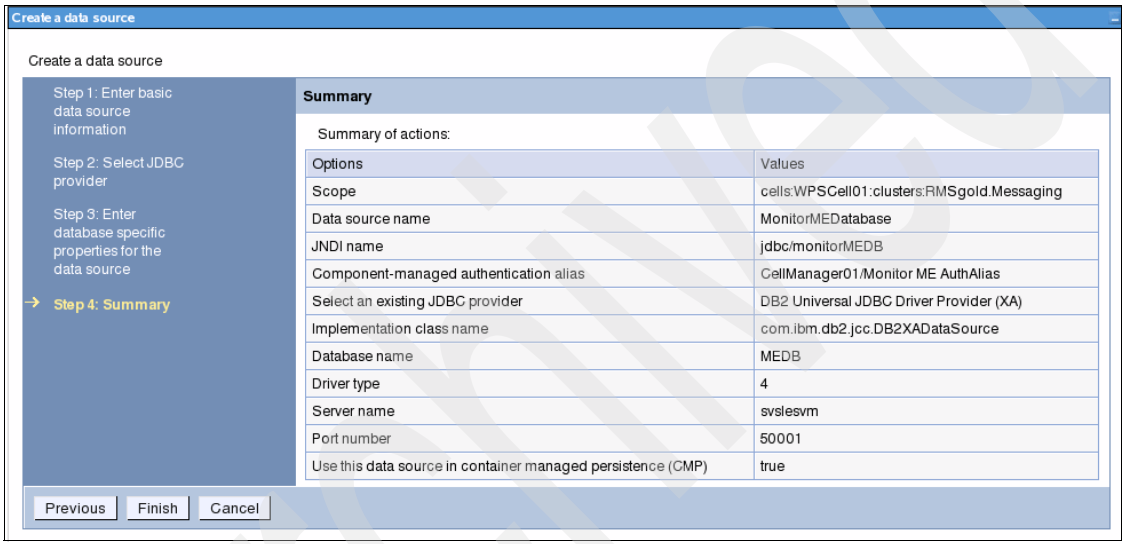


Figure 15-21 Data source Summary panel

g. Review the Summary panel shown in Figure 15-21. Click **Finish**.

h. Restart the Deployment Manager to have it pick up the new J2C Authentication Alias.

i. Restart the WebSphere Process Server nodes to pick up the newly created data source.

j. Use the Test Connection button on the data source page to verify that the datasource is properly working.

15.6.3 Using the WebSphere Business Monitor configuration panel

This is a new edition to WebSphere Business Monitor v6.2. It allows you one location within the Integrated Solutions Console to view and change the WebSphere Business Monitor configuration. The wizard looks like Figure 15-22.

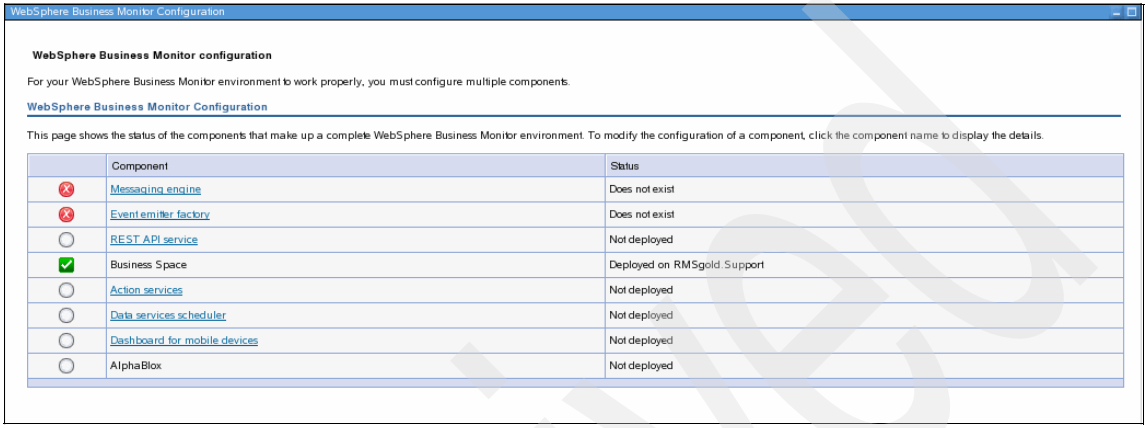


Figure 15-22 WebSphere Business Monitor configuration wizard panel

Configure the Monitor Messaging Engine

To configure:

1. Launch the WebSphere Business Monitor configuration panel:
   a. Log in to the Integrated Solutions Console.
   b. Expand **Servers**.
   c. Click **WebSphere Business Monitor configuration**.

2. Click the **Message engine** link under the component column of the panel. This launches the messaging engine wizard shown in Figure 15-23. Click **Configure the Messaging Engine button**.

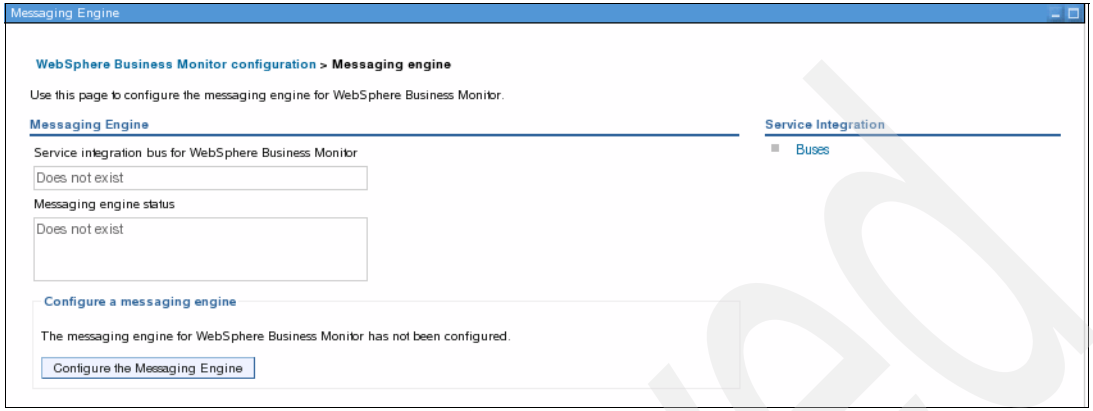


Figure 15-23 Messaging Engine wizard

3. Select the **cluster** radio button and select **RMSGold.Messaging**, as shown in Figure 4. Click **Next**.

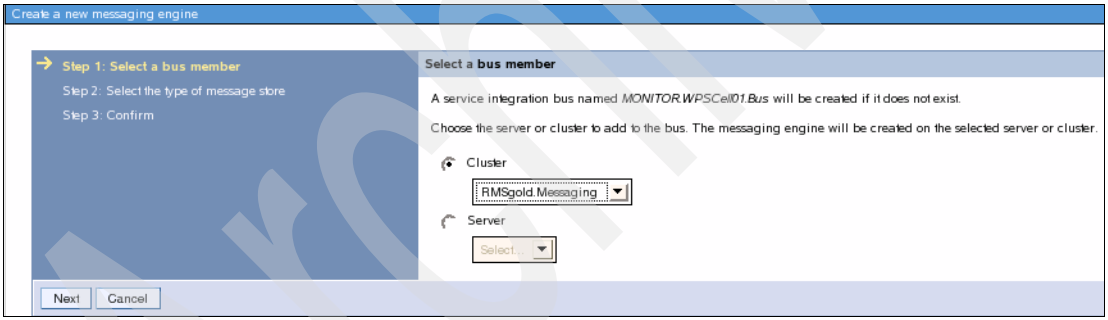


Figure 15-24 Step 1 Messaging Engine Panel

4. Select the **Data store** radio button. Click **Next**.

5. Enter the values provided in Table 15-10. Keep **Create Tables** check box checked.

Table 15-10 Use existing data source values

Name	Value
Data Source JNDI Name	jdbc/monitorMEDB
Schema Name	MONITOR
Authentication Alias	CellManager01/Monitor MEAuthAlias

6. Click the **Next Review** panel, as shown in Figure 7 on page 466, before clicking **Next**.

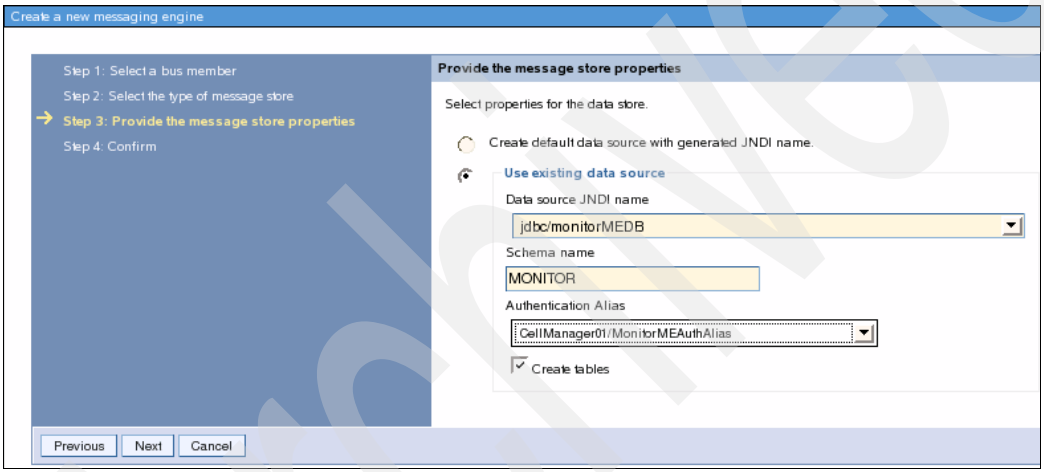


Figure 15-25 Provide Message datastore properties

7. Review the summary shown in Figure 15-26. Click **Finish**.

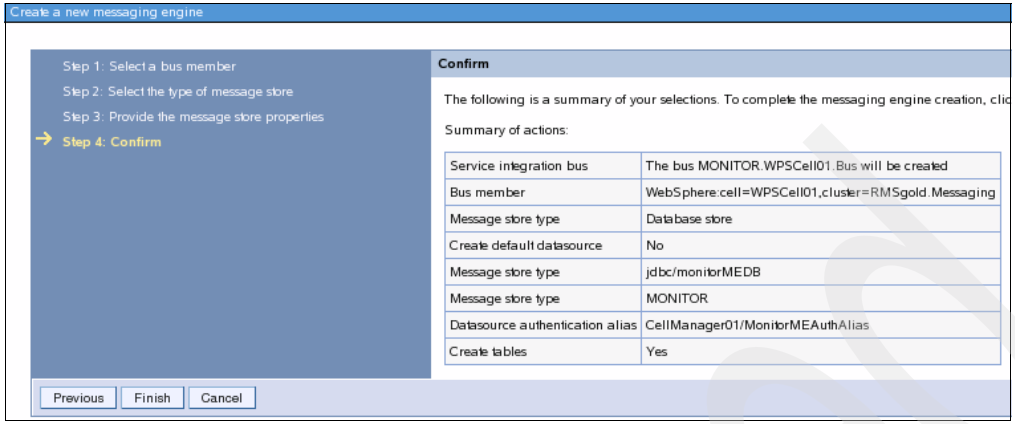


Figure 15-26 Confirm Messaging Engine summary

Create Event Emitter factory

To do this:

1. Launch the WebSphere Business Monitor configuration panel.
   a. Log in to the Integrated Solutions Console.
   b. Expand **Servers**.
   c. Click **WebSphere Business Monitor configuration**.

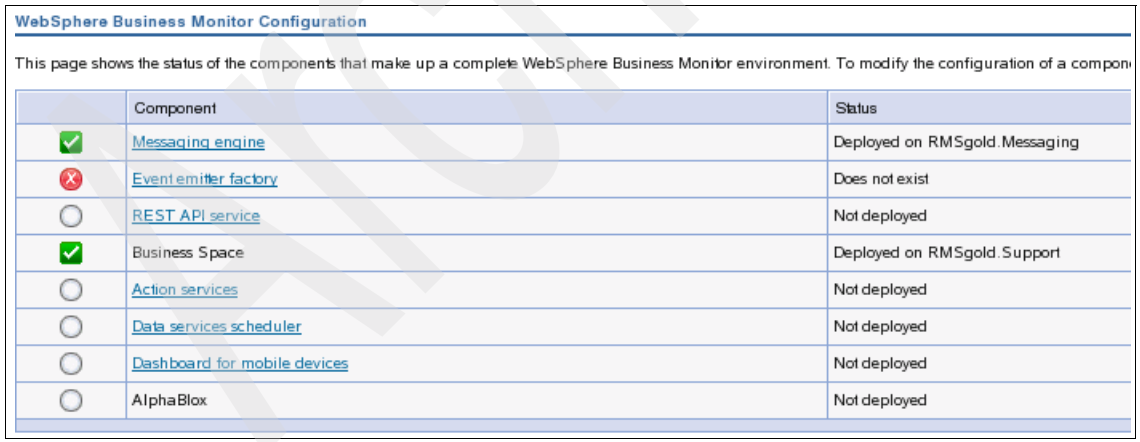


Figure 15-27 WebSphere Business Monitor Configuration panel

2. Click the **Event Emitter Factory** link shown in Figure 15-27 on page 466.
3. Select **WBM.Support** from the Configure an event emitter factory drop-down list box. Click the **Configure an event emitter factory** button shown in Figure 15-28.

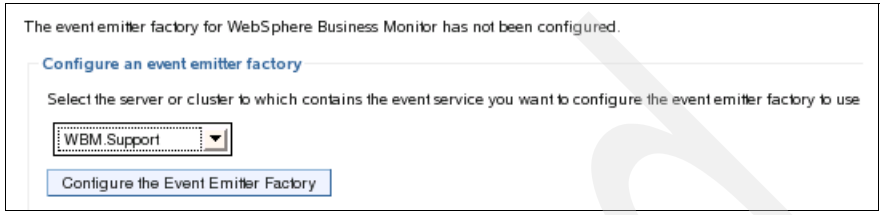


Figure 15-28 Configure Event Emitter Factory

4. You should get a success message that looks like Figure 15-29.

Figure 15-29 Message of success

5. Click the **Event emitter factories** link shown in Figure 15-30.

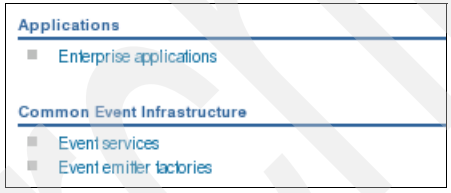


Figure 15-30 Event emitter factories

6. Review the event emitter factories. You should see MonitorEmitterFactory.

Select	Name	Scope	JNDI name	Use New Transactions
	Default Common Event Infrastructure emitter	Cluster=RMSgold.Support	com/ibm/events/configuration/emitter/Default	false
	MonitorEmitterFactory	Cell=WPSCell01	com/ibm/monitor/MonitorEmitterFactory	false

Figure 15-31 Event emitter factory list

15.7 Installing WebSphere Business Monitor support applications

The section describes how to do the following support applications:

- Deploy action services.
- Deploy Data services scheduler.
- Deploy REST API Service.
- Deploy monitor event emitter service (optional).

Note: The monitor event emitter service (EmitterRestServices.ear) is not deployed using a wizard. It is deployed manually. This is optional, as it used to send XSD style events using Monitor API. The event emitter service is not used in the current installation.

15.7.1 Deploy action services

To do this:

1. Launch the WebSphere Business Monitor configuration panel.
   a. Log in to the Integrated Solutions Console.
   b. Expand **Servers**.
   c. Click **WebSphere Business Monitor configuration**.
2. Click the **Action Services** link on the configuration panel, as shown in Figure 15-32.

	Component	Status
✓	Messaging engine	Deployed on RMSgold.Messaging
✓	Event emitter factory	Configured using the event service on WBM.Support
○	REST API service	Not deployed
✓	Business Space	Deployed on RMSgold.Support
○	Action services	Not deployed
○	Data services scheduler	Not deployed
○	Dashboard for mobile devices	Not deployed
○	AlphaBlox	Not deployed

Figure 15-32 WebSphere Business Monitor configuration panel

3. Select **WBM.Support** in the Deploy action services list box. Click the **Deploy Action Services** button shown in Figure 15-33.

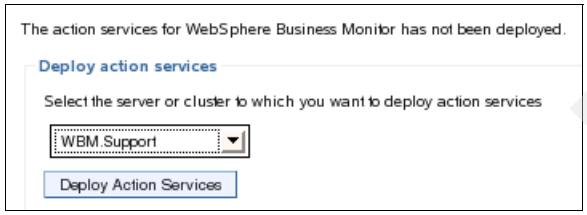


Figure 15-33 Step one panel for deploy action services

This should show you the success message shown in Figure 15-34.

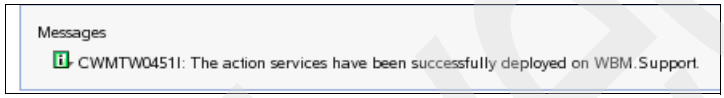


Figure 15-34 Success message for Action Manager

Note: If deployment fails, verify that you have properly configured your /etc/security/limits.conf on your deployment manager machine. The nofile should be configured to * soft nofile 10240 and * hard nofile 10240. These should be set on all machines on which WebSphere Business Monitor is installed.

To enable monitor to send events using action manager functionalities, you must change CSIv2 inbound and outbound configuration. To do the required changes, perform the following instructions for each member in the Monitor model event cluster.

1. Go to **Application servers** → **WBMeventMember01** → **Server security** → **CSIv2 outbound authentication**.
2. In the Basic authentication area, select the **Supported** option.
3. In the Client certificate authentication area, select the **Supported** option.
4. Ensure that the **Identity assertion** option is checked and **Use server trusted identity** is selected.

Perform the following instructions for each member in the Monitor support cluster:

1. Go to **Application servers** → **WBMSupportMember01** → **Server security** → **CSIv2 inbound authentication**.
2. In the Basic authentication area, select the **Supported** option.
3. In the **Client certificate authentication** area, select the **Supported** option.
4. Ensure that the **Identity assertion** option is checked.
5. Ensure that the **Stateful sessions** option is checked.

15.7.2 Deploy Data services scheduler

To do this:

1. Launch the WebSphere Business Monitor configuration panel:
   a. Log in to the Integrated Solutions Console.
   b. Expand **Servers**.
   c. Click **WebSphere Business Monitor configuration**.

2. Click the **Data services scheduler** link on the configuration panel, as shown in Figure 15-35.

	Component	Status
✓	Messaging engine	Deployed on RMSgold.Messaging
✓	Event emitter factory	Configured using the event service on WBM.Support
○	REST API service	Not deployed
✓	Business Space	Deployed on RMSgold.Support
✓	Action services	Deployed on WBM.Support
○	Data services scheduler	Not deployed
○	Dashboard for mobile devices	Not deployed
○	AlphaBlox	Not deployed

Figure 15-35 WebSphere Business Monitor configuration panel

3. Select **WBM.Support** in the Deploy Data services scheduler list box. Click the **Deploy Data Services Scheduler** button, as shown in Figure 15-36.

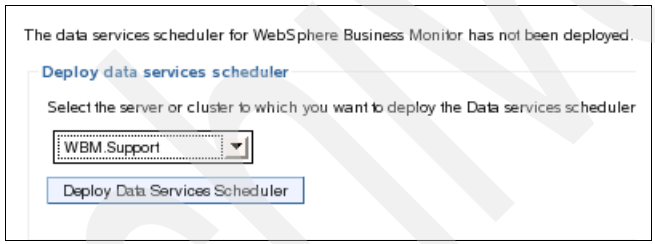


Figure 15-36 Data service scheduler deploy panel

This should show you a success message, as shown in Figure 15-37.

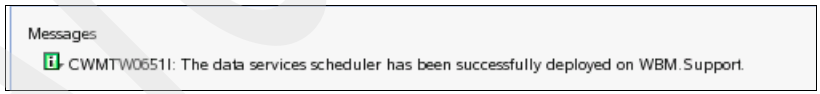


Figure 15-37 Success message for data service scheduler

15.7.3 Deploy REST API Service

To do this:

1. Launch the WebSphere Business Monitor configuration panel:
   a. Log in to the Integrated Solutions Console.
   b. Expand **Servers**.

c. Click **WebSphere Business Monitor configuration**.

2. Click the **REST API service** link on the configuration panel, as shown in Figure 15-38.

	Component	Status
✓	Messaging engine	Deployed on RMSgold.Messaging
✓	Event emitter factory	Configured using the event service on WBM.Support
○	REST API service	Not deployed
✓	Business Space	Deployed on RMSgold.Support
✓	Action services	Deployed on WBM.Support
✓	Data services scheduler	Deployed on WBM.Support
○	Dashboard for mobile devices	Not deployed
○	AlphaBlox	Not deployed

Figure 15-38 WebSphere Business Monitor configuration panel

3. Select **WBM.BusSpace** in the Deploy REST API service list box. Click the **Deploy REST API service** button, as shown in Figure 15-39.

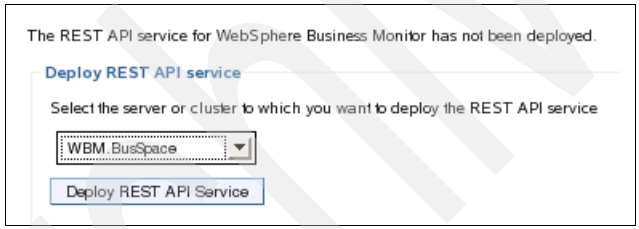


Figure 15-39 REST API service deploy panel

This should show you a success message like that shown in Figure 15-40.

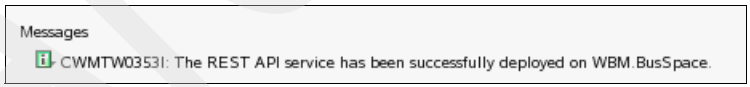
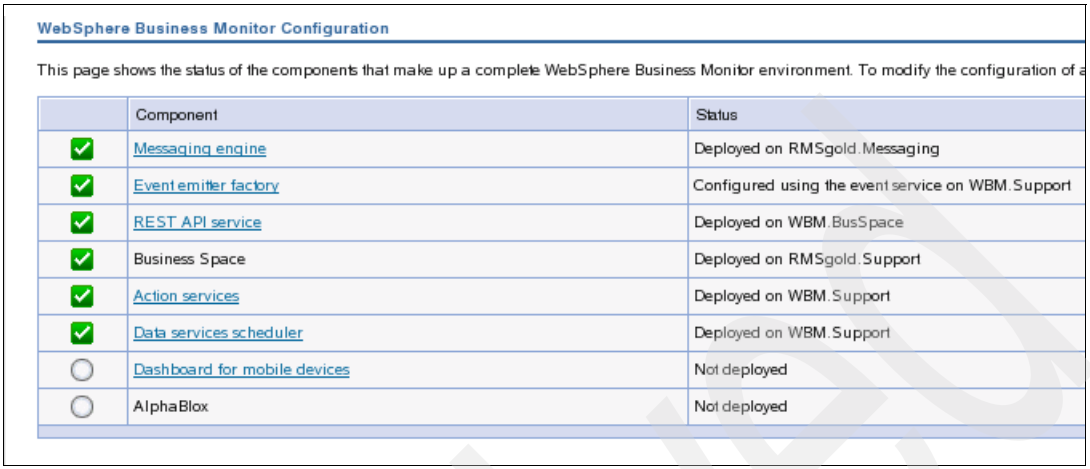


Figure 15-40 Success message for REST API service

The WebSphere Business Monitor configuration panel will look like Figure 15-41.



WebSphere Business Monitor Configuration

This page shows the status of the components that make up a complete WebSphere Business Monitor environment. To modify the configuration of a

	Component	Status
✓	Messaging engine	Deployed on RMSgold.Messaging
✓	Event emitter factory	Configured using the event service on WBM.Support
✓	REST API service	Deployed on WBM.BusSpace
✓	Business Space	Deployed on RMSgold.Support
✓	Action services	Deployed on WBM.Support
✓	Data services scheduler	Deployed on WBM.Support
○	Dashboard for mobile devices	Not deployed
○	AlphaBlox	Not deployed

Figure 15-41 WebSphere Business Monitor configuration panel

Note: These services are deployed, but they are not running. Expand **Applications** → **Enterprise Applications**, then check the service to start. Click the **Start** button

If you are going to provide people with the option of viewing the dashboard via their mobile device then you will need to deploy the Dashboard application for Mobile devices on the business space cluster. You must follow the same steps.

15.8 Business Space considerations

When considering where to deploy Business Space, you must consider the cell configuration:

- Is Business Space already deployed on the RMSgold.Support cluster?
- Has the schema been created and used?
- Will you be using Alphablox dimension and report widgets for Monitor dashboards?

In Table 15-11, the first and last option would leave Business Space on the RMSgold.Support cluster. In the second and third options, you would deploy Business Space Manger to the WBM.BusSpace cluster.

Table 15-11 Business Space deployment decision table

Follow heading	Business Space already installed on RMS.Support cluster	Using Alphablox widgets	Notes
Leave configuration unchanged	Yes.	No.	Skip to configuring widgets.
15.8.1, “Alphablox” on page 475	Yes. If you want to migrate the installation of Business Space to the WBM.BusSpace cluster, follow instructions in 15.8.1, “Alphablox” on page 475, and 15.8.3, “Migrating Business Space from RMSGold.Support cluster to WBM.BusSpace cluster” on page 484.	Yes.	You should install Alphablox prior to installing Business Space Manager to use certain widgets. If you are just installing on WPS clusters, you can install Alphablox on the corresponding WPS cluster and then activate the Monitor widget WAR.

Follow heading	Business Space already installed on RMS.Support cluster	Using Alphablox widgets	Notes
15.8.1, “Alphablox” on page 475, *and* 15.8.2, “Deploying Business Space” on page 483	No.	Yes.	If you are going to use Alphablox, you must install Alphablox first.

15.8.1 Alphablox

You must install Alphablox software on all machines in the cluster where Business Space Manager is to be deployed.

Note: It has been our experience that you should install Alphablox on one cluster member in the cluster and configure that first. Then proceed to install Alphablox on a second cluster member.

Install Alphablox

To install:

1. Stop all Monitor clusters before installing Alphablox.
2. Run the Alphablox installer located in the `<Monitor install>/installableApps.wbm/installer/Alphablox` directory.
3. Provide the following input:
   a. In Choose Locale field select **English**.
   b. Accept the licence agreement.
   c. Enter a destination directory of `/opt/ibm/WebSphere/MonServer/ABX`.
   d. Accept the default server instance name of **AlphabloxAnalytics**.
   e. Select an installation set of **Typical**.
   f. Select the application server to use with Alphablox as **WebSphere**. Ensure that the WebSphere server is stopped at this point.
   g. Enter a WebSphere root directory of `/opt/ibm/WebSphere/MonServer`.
   h. Select the node and server to install Alphablox on the required profile.

   Note: In this topology, the Alphablox application is installed on the WBM.BusSpace cluster.

i. Accept the default values for HTTP Port and SOAP Connector Port.

j. Provide a WebSphere Administrator user name and password.

k. Accept the default values for the Telnet console port and server log file name.

l. Set the Console Message Level to **3**.

m. Accept the SMTP Server and Java Directory default values.

n. Enter `Y` to enable additional drivers for IBM Alphablox.

o. Enter the location of the driver as `/opt/ibm/WebSphere/MonServer/universalDriver.wbm/lib` and confirm that this drive is a DB2 Type 4 driver.

p. Select **DB2** as the database repository.

q. Set the database server to `svslesvm`, the port to `50001`, and the alias to `MONITOR`. This alias is the name of the database that will be used as an Alphablox repository, and can share the same name with the Monitor database.

r. Specify a user name of `db2inst1` and password of `passw0rd`.

s. The installer then runs a database connection test. Check that the test runs successfully. You will see an error that the tables shown in Example 15-4 are not found. They will be created when the Alphablox server is started for the first time.

Example 15-4 Missing alphablox system tables

```
Alphablox system table ABX_OBJECTS not found
Alphablox system table ABX_TYPES not found
Alphablox system table ABX_VERSION not found
Alphablox system table ABX_LOOKUP not found
Alphablox system table ABX_PROPERTY_MAP not found
Alphablox system table ABX_LOOKUP_VALUES not found
```

t. Enter `1` to configure clustering and accept the default values for cluster port number and cluster subnet mask. The cluster port number should be identical for all Alphablox server instances in the cluster.

u. Select the conversion operation **Copy**.

v. Set Move Server Properties to **All**.

w. Select **No** for the User defined DDL schema file.

x. Review the summary shown in Example 15-5 and let the installer complete the remainder of the installation.

Example 15-5 Summary

```
Installation Directory:  /opt/Alphablox
Instance Name:  AlphabloxAnalytics
Application Server:  WebSphere
    WebSphere Home:  /opt/ibm/WebSphere/ProcServer
    WebSphere Product:  IBM WebSphere Application Server - ND
    WebSphere Version:  6.1.0.21
    WebSphere Start File:  setupCmdLine.sh
    WebSphere Cluster Install:  true
    WebSphere Profile: WBMBusSpaceMember01
    WebSphere Cell:  WPSCell01
    WebSphere Node: monNode01
    WebSphere Server: dmgr
    HTTP Request Port:  9080
    SOAP Connector Port:  8879
    SOAP Admin User:  wps
Telnet Console Port:  20023
Server Log File Name:  Server.log
Console Message Level:  INFO
Java Directory:  /opt/ibm/WebSphere/ProcServer/java
Additional Driver Directory:
/opt/ibm/WebSphere/ProcServer/universalDriver.wbm/libDB2 Driver
Type:  4
Drivers:  Enabled: DB2 Type 4, Derby
Repository Type:  Database
    Database Type:  DB2
    Database Server:  svslesvm
    Database Port:  50001
    Database Alias:  MONITOR
    Database User:  db2inst1

CLUSTERING: Enabled:
    Port::  7855
    Subnet Mask::  255.255.255.0

Repository Conversion Utility:
    Operation:  Copy
    Existing Tables:
```

When Alphablox completes its installation, you should see the `IBM Alphablox successfully installed on your system!` message.

4. Start the WebSphere Business Monitor deployment manager and clusters.

Deploy the Alphablox libraries to BusinessSpaceCluster

To do this:

1. From the `/opt/ibm/WebSphere/MonServer/ABX/bin` directory, run:

   ```
   ./DeployWebSphereLibraries.sh -conntype SOAP -username wps -password
   passw0rd
   ```

2. Install libraries to a cluster by selecting the following:
   a. Select option **1** to install libraries.
   b. Select option **1** to install to a cluster.
   c. Select the **BusinessSpaceCluster** cluster since the Business Space Widgets will make use of the libraries.
   d. Verify the cluster members that should have the libraries deployed.
   e. Verify that the libraries were successfully installed.
   f. Exit the deploy script by entering `5` twice.

> **Note:** Check the successful deployment of Alphablox libraries by performing the following instructions:
>
> 1. After restarting the servers, log in to monitor deployment manager admin console.
> 2. Go to **Environment** → **Shared Libraries**.
> 3. There should be a list of Alphablox shared libraries.
>
> If the Alphablox libraries deployment failed or after restarting servers you are not able to log in to the Alphablox admin page you should perform the deployment manually by copying all JAR files in `/AlphabloxInstallationFolder/lib` to the following folder in `/MonitorInstallationFolder/lib/ext`.

Deploy the Alphablox applications

To do this:

1. To finalize the IBM Alphablox installation, two applications should be installed on the Monitor Dashboard cluster:
   – AlphabloxPlatform.ear
   – ApplicationStudio.ear
2. Open the WebSphere Business Monitor Integrated Solutions Console. Go to **Applications** → **Enterprise Applications** and click **Install**.

3. Use the remote file system path setting to browse through the network to locate the `AlphabloxPlatform.ear` file. Browse to `/opt/ibm/WebSphere/ProceServer/ABX/installableApps/AlphabloxPlatform.ear`. Accept defaults. Click **Next**.
4. Accept defaults in Step 1: Select installation options. Click **Next**.
5. In Step 2: Map modules to servers, click the Select all applications icon. Select **WBM.BusSpace cluster** and click **Apply,** as shown in Figure 15-42. Click **Next**.

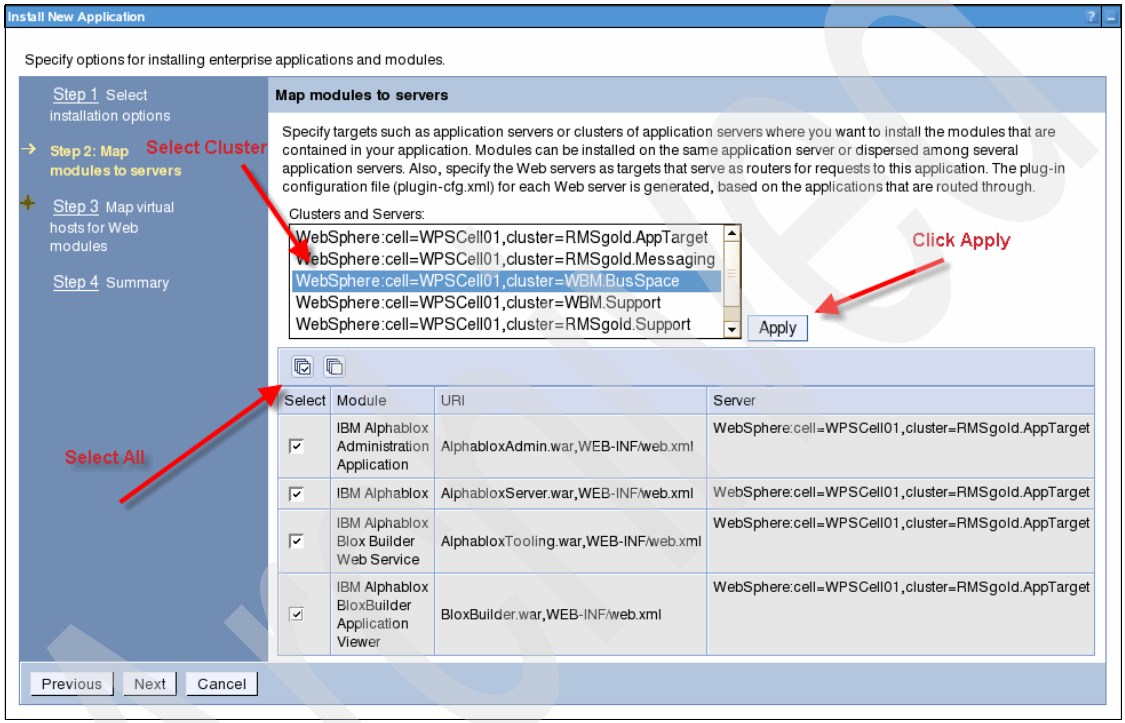


Figure 15-42 Map modules to servers

6. In Step 3: Map Virtual hosts to web modules, keep the defaults and click **Next.**
7. Click **Finish**.

8. You should see the successful message, as shown in Figure 15-43. Click **Save**. Click **OK**.

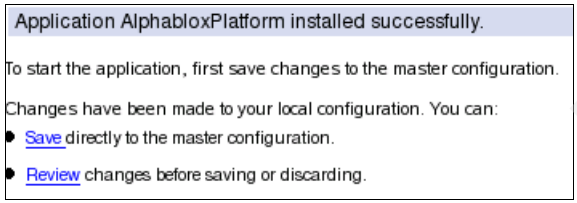


Figure 15-43 Install success message

Repeat these steps to install the AlphabloxStudio application.

Configure the Alphablox applications

After installing Alphablox, you must configure the application to start up last. To do this you must change the starting weight to a high number.1000 is a recommended start weight. Then map roles to users for the applications. To perform the both steps:

1. From the Integrated Solutions Console, expand **Applications** → **Enterprise Applications** → **AlphabloxPlatform**. Click **StartUp Behavior** (Figure 15-44).

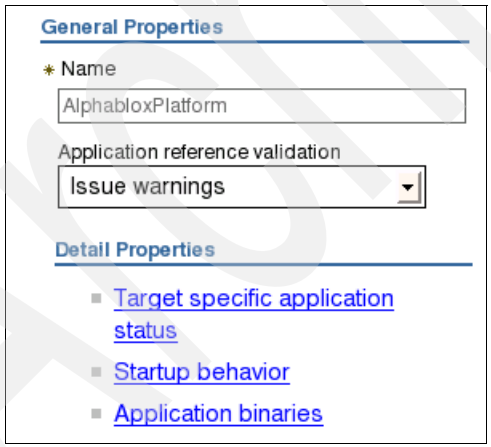


Figure 15-44 Startup behavior

2. Change Startup order from 1 to 1000 (Figure 15-45). Click **OK**.

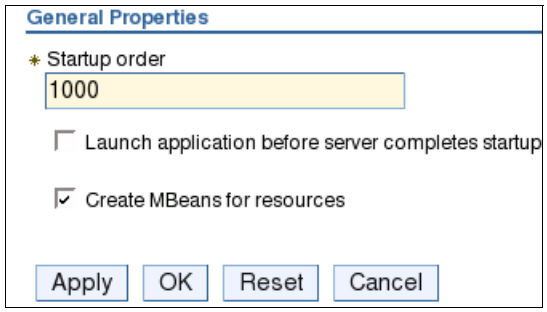


Figure 15-45 General Properties panel for startup behavior

3. Click **Save**, then **OK**.
4. Click the **Security role to user/group mapping** link.
5. For each role, select the role and click the **Look up users** button. Then select the users to map to the selected role, as shown in Figure 15-46.

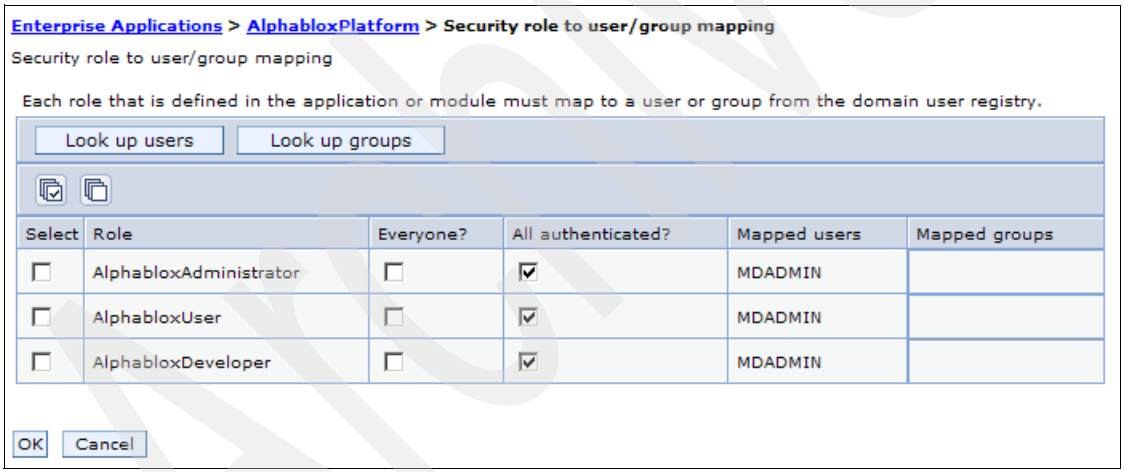


Figure 15-46 Select role to user/group mapping page

6. Click **OK**.
7. Repeat these steps for the AlphabloxStudio application.

Post-installation configuration

For WebSphere vertical clusters, you must perform the following post-installation configuration to properly configure the server-specific JVM™ parameter to

identify the Java Management Extensions (JMX™) communication port that Alphablox should use and set the server log name.

> **Note:** A vertical cluster has cluster members on the same node. A horizontal cluster has cluster members on multiple nodes. You can configure either type of cluster or have a combination of vertical and horizontal clusters.

From the Integrated Solutions Console, expand **Servers** → **Application Servers**. For each vertical node in the cluster, perform the following steps:

1. Click the server's name (for example, server1).
2. In the Server Infrastructure section, select **Java and Process Management** → **Process Definition**.
3. In the Additional Properties section, select **Java Virtual Machine**.
4. In the Generic JVM arguments text box, enter the following arguments, leaving a space between the two:

   ```
   -Dabx.ws.admin.port.override=portNumber
   -Dabx.cluster.log.file.suffix=serverName
   ```

   The port number (portNumber) is usually generated when the server instance is created in the WebSphere server. To determine the port value, select the server's name under **Servers** → **Application Servers** and click **Ports**. The value to use in portNumber above is the port number for the SOAP_CONNECTOR_ADDRESS port name.

   The server name (*serverName*) is the server name displayed under **Servers** → **Application Servers**.
5. Save your changes to the master configuration and then restart the servers in the cluster.

After properly installing DB2 Alphablox, the WebSphere Business Monitor data sources must be using the DB2 Alphablox administration page. Complete the following steps to create the required data sources:

1. Open the DB2 Alphablox Admin Console. In our environment the URL for the DB2 Alphablox Admin Console is:

   `http://clustrm:9084/AlphabloxAdmin`
2. Go to the Administration tab and click **Data Sources**.
3. Click **Create** and perform the following steps:
   a. Enter `MONITOR` in the Data Source Name text box.
   b. Select **Application Server Data Source** from the Adapter list.

c. Enter `jdbc/wbm/MonitorDatabase` in the Application Server Data Source Name text box.

4. Click **Save**.
5. Click **Create** and perform the following steps:
   a. Type `MONITOR_CUBE` in the Data Source Name text box.
   b. Select **Alphablox Cube Server Adapter** from the Adapter list.
6. Click **Save**.
7. Restart the servers.

After creating monitor data sources, you should configure monitor themes for Alphablox. For more information refer to the monitor information center:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.install.doc/install/abx_theme_manual.html

Note: You must restart the server for the created data sources to be reflected on other cluster members.

15.8.2 Deploying Business Space

To do this:

1. Using the Integrated Solutions Console, expand **Servers** → **Clusters** → **WBM.BusSpace** → **Business Integration** → **Business Space Configuration**.
2. Check **Install Business Space service** in the panel.
3. Enter the values provided in Table 15-12. The data source that you select will decide on which database you will create the schema.

Table 15-12 Values for business space service properties

Name	Value
Database Schema name	COMMONDB
Create Business Space data source using	Monitor_Database

4. Click **OK**. Review the message shown in Figure 15-47 prior to clicking **Save**.

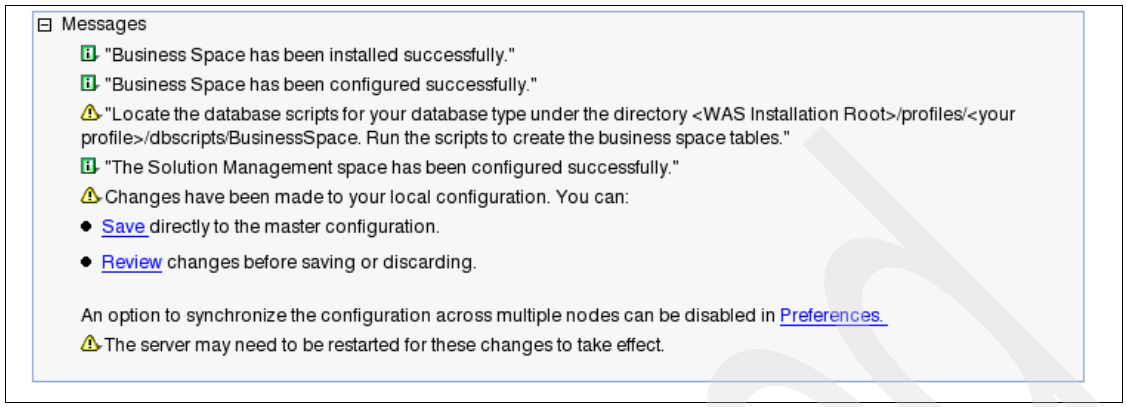


Figure 15-47 Success message from Business Space configuration

Figure 15-47 shows a successful installation of Business Space.

5. Create the tables for the Business Space service. Issue the commands as db2inst1 or anybody with DBADM privileges to create schema objects:

```
>db2 connect to MONITOR
>db2 -tf createTable_BusinessSpace.sql
>db2 reset
```

6. Start the Business Space Manager:
   a. Expand **Applications** → **Enterprise Applications**.
   b. Check **Business Space Manager**.
   c. Click **Start**.

15.8.3 Migrating Business Space from RMSGold.Support cluster to WBM.BusSpace cluster

To do this:

1. Review existing Business Space Manager Security roles. These must be migrated.
   a. Expand **Applications** → **Enterprise Applications Figure 15-48 on page 485 Business Space Manager**.
   b. Click **Security role to user/group mapping**.
   c. Take note of what has already been configured, as shown in Figure 15-48 on page 485. You need these when you deploy to the WBM.Support

cluster. In the panel shown it is configured for All Authenticated. Your configuration will probably have mapped groups.

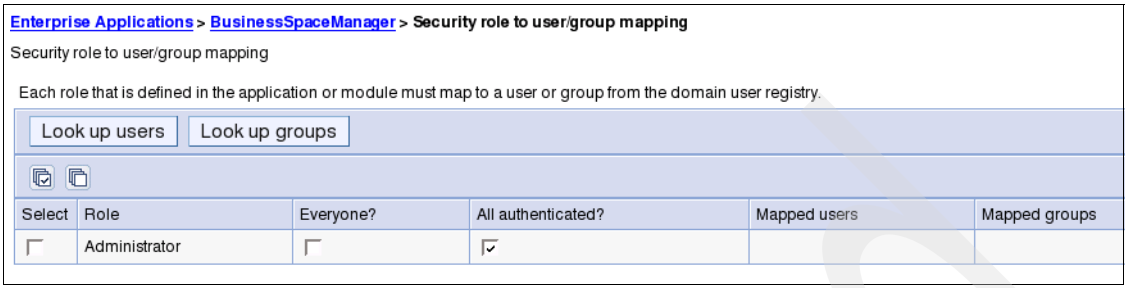


Figure 15-48 Business Space security panel

2. Document the Business Space Manager Datasource and schema for the existing Business Space Manager configuration:
   a. Expand **Resources** → **JDBC** → **Datasources**.
   b. Set the scope to **cluster = RMSgold.support**.
   c. Click **Business Space Manager Datasource**.

d. Make note of the values in the panel shown in Figure 15-49.

> **Note:** The database in the configuration is the main WebSphere Process Server database (WPRCSDB). Your configuration may be different.

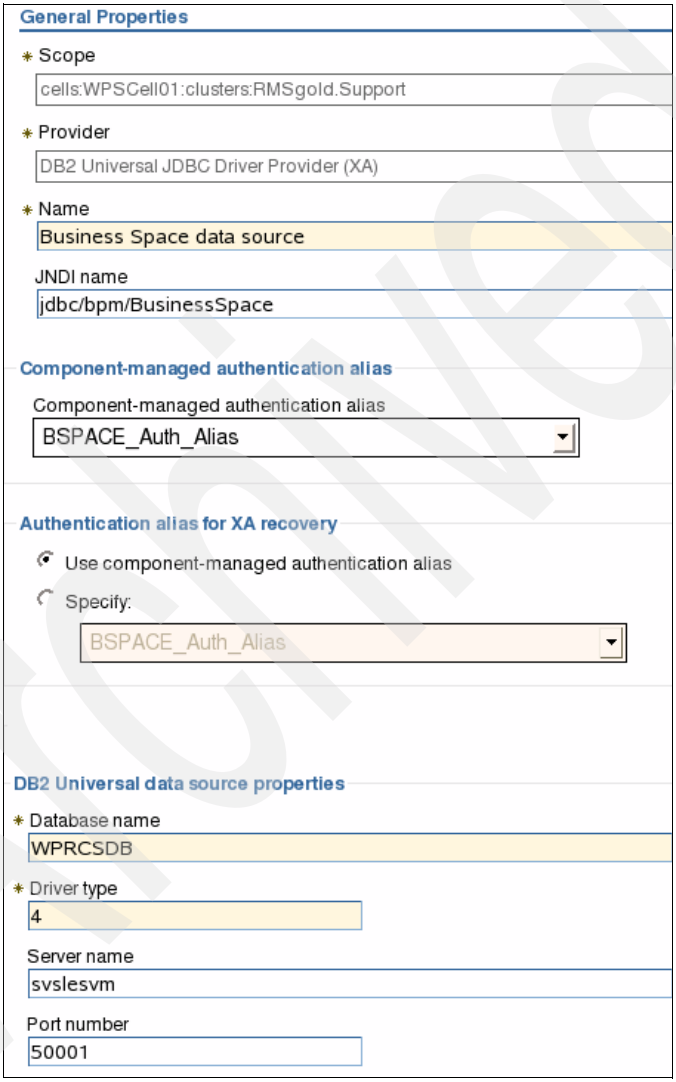


Figure 15-49 Business Space Manager datasource

e. Once you have the database name, you must find the existing schema name. Log in as db2inst1 and issue the following DB2 command to locate the schema. In this case the schema name is COMMONDB.

```
db2inst1:/home/db2inst1> db2 list tables for all | grep -i widget
REGISTERED_WIDGET               COMMONDB        T     2009-02-16 ...
REGISTERED_WIDGET_NLS           COMMONDB        T     2009-02-16 ...
WIDGET                          COMMONDB        T     2009-02-16 ...
```

Note: Another way to find the schema name is to review the generated script createTable_BusinessSpace.sql from the initial deployment under the deployment manager profile.

In this book's configuration the directory is `/opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/dbscripts/BusinessSpace/DB2/WPRCSDB`.

Search the sql file for CREATE TABLE.

3. Stop Business Space Manager:
   a. Expand **Applications** → **Enterprise Applications**.
   b. Check **Business Space Manager**.
   c. Click **Stop**.
4. Uninstall Business Space Manager:
   a. Expand **Applications** → **Enterprise Applications**.
   b. Check **Business Space Manager**.
   c. Click **Uninstall**. Click **Save**.
5. Deploy BusinessSpace Manager:
   a. Using the Integrated Solutions Console, expand **Servers** → **Clusters** → **WBM.BusSpace** → **Business Integration** → **Business Space Configuration**.

b. Check **Install Business Space service** in the panel shown in Figure 15-50.

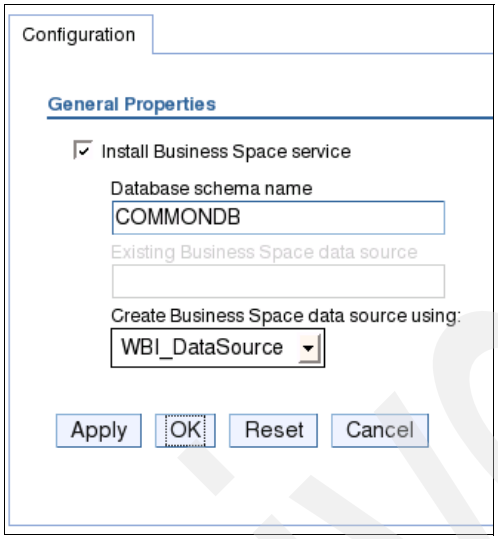


Figure 15-50 Install BusinessSpace service

c. Enter the values provided in Table 15-13. The data source that you select determines on which database you will create the schema. Select the existing datasource and schema.

Table 15-13 Vlaues for business space service properties

Name	Value
Database Schema name	COMMONDB
Create Business Space data source using	WBI_Database

d. Click **OK**. Review the message shown in Figure 15-51 prior to clicking **Save**.

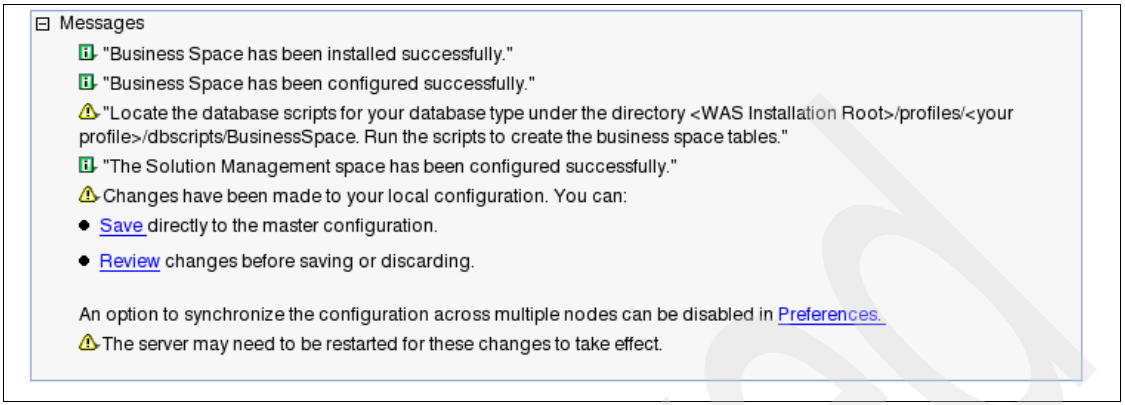


Figure 15-51 Message of successful deployment of Business Space Manger

Since you are using the existing database and schema, you do not have to take the additional steps to create the database tables.

e. After migrating the Business Space to WBM.BusSpace cluster, you must activate monitor ABX widgets by following the instructions in at the following link:

`http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.install.doc/install/alpha_inst.html`

> **Note:** You must migrate the WebSphere Process Server widget endpoint files to the corresponding new location at the WBM.BusSpace cluster after business space migration so that the installed WPS widgets will be available for use.

15.8.4 Configure Business Space for dashboard widgets

After installing Business Space, you should configure monitor widget XML files to enable the dashboards views in Business Space. By default, the dashboard widgets are neither registered nor enabled in Business Space. Using the administrator user, you can add, remove, or update widget XML files. You can edit those XML files, make the required changes, and copy them to the BusinessSpace/registry directory, as indicated below.

The Business Space registration files are located at `<WebSphere_Process_Server_installation>\BusinessSpace\registryData` in all nodes of Business Space cluster.

To configure WebSphere Business Monitor dashboards on Business Space for each cluster member:

1. Create the `BusinessSpace/registryData` directory on all the nodes of the Business Space cluster in the following path:

   `<WebSphere_Process_Server_installation>\profiles\<Monitor_profile_name>\BusinessSpace\registryData`

2. To register WebSphere Business Monitor widgets, create a copy of the `monitorWidgets.xml` file and then edit this file. Locate the element <tns:Widget> for all the widgets that you would like to administer. Add the action attribute to the <tns:Widget> element as shown below:

   – `<tns:Widget action="addUpdate">` (This is the default.)
   – `<tns:Widget action="add">` (Adds a new widget to the registry.)
   – `<tns:Widget action="update">` (Updates the widget to the registry.)
   – `<tns:Widget action= "delete">` (Deletes the widget from the registry.)

   Example 15-6 is an example of an edited `monitorWidgets.xml` file.

Example 15-6 monitorWidgets.xml

```
<!-- START NON-TRANSLATABLE -->
  <tns:Widget action="update">
    <tns:id>{com.ibm.wbimonitor}instances</tns:id>
    <tns:version>1.0.0.0</tns:version>
    <tns:name>Instances</tns:name>
    <tns:type>{com.ibm.bspace}mWidget</tns:type>
    <tns:description>IBM WebSphere Business
Monitor</tns:description>
    <tns:tooltip>Instances</tns:tooltip>
    <tns:categoryId>{com.ibm.wbimonitor}monitor</tns:categoryId>

<tns:widgetEndpointId>{com.ibm.wbimonitor}monitorWidgetRootId</tns:w
idgetEndpointId>
    <tns:viewUrl>_Instances/jsp/html/InstancesView.jsp</tns:viewUrl>
    <tns:editUrl>_Instances/jsp/html/InstancesEdit.jsp</tns:editUrl>
    <tns:helpUrl>dash/help_instances.html</tns:helpUrl>
    <tns:iconUrl>img/Instances.gif</tns:iconUrl>
    <!-- <tns:previewUrl>TBD</tns:previewUrl> -->
    <tns:owner>IBM</tns:owner>
    <tns:email>TBD</tns:email>
    <tns:serviceEndpointRef>
      <tns:name>serviceUrlRoot</tns:name>

<tns:refId>{com.ibm.wbimonitor}monitorServiceRootId</tns:refId>
      <tns:refVersion>1.0.0.0</tns:refVersion>
```

```
    </tns:serviceEndpointRef>
    <tns:localeInfo>
    <!-- END NON-TRANSLATABLE -->
      <tns:locale>en_US</tns:locale>
      <tns:name>Instances</tns:name>
      <tns:description>IBM WebSphere Business
Monitor</tns:description>
      <tns:tooltip>Instances</tns:tooltip>
    <!-- START NON-TRANSLATABLE -->
    </tns:localeInfo>
  </tns:Widget>
  <!-- END NON-TRANSLATABLE -->
```

3. Save the `monitorWidgets.xml` file.
4. Copy the `monitorWidgets.xml` file to the `<WebSphere_Process_Server_installation>\profiles\<Monitor_profile_name>\BusinessSpace\registryData` directory on all the nodes where Business Space is installed.
5. Restart the Business Space cluster.
6. To enable WebSphere Business Monitor widgets, create a copy of the following endpoint registration files:
   - `monitorABXEndpoints.xml`
   - `monitorEndpoints.xml`
7. Edit the two files indicated below in bold. Example 15-7 shows `monitorEndpoints.xml` and Example 15-8 on page 492 shows `monitorABXEndpoints.xml`.

Example 15-7 monitorEndpoints.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- START NON-TRANSLATABLE -->
<tns:BusinessSpaceRegistry
xmlns:tns="http://com.ibm.bspace/BusinessSpaceRegistry"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://com.ibm.bspace/BusinessSpaceRegistry
BusinessSpaceRegistry.xsd ">

  <tns:Endpoint action="addUpdate">
    <tns:id>{com.ibm.wbimonitor}monitorServiceRootId</tns:id>
    <tns:version>1.0.0.0</tns:version>
<tns:description>Location of backing services for Monitor
widgets</tns:description>
  </tns:Endpoint>
```

```
</tns:BusinessSpaceRegistry>
<!-- END NON-TRANSLATABLE -->
```

Example 15-8 monitorABXEndpoints.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- START NON-TRANSLATABLE -->
<tns:BusinessSpaceRegistry
xmlns:tns="http://com.ibm.bspace/BusinessSpaceRegistry"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://com.ibm.bspace/BusinessSpaceRegistry
BusinessSpaceRegistry.xsd ">

  <tns:Endpoint action="addUpdate">
    <tns:id>{com.ibm.wbimonitor}monitorABXServiceRootId</tns:id>
    <tns:version>1.0.0.0</tns:version>
<tns:url>http://rest_services_hostname:port_number/rest/</tns:url>
    <tns:description>Location of backing services for Monitor
widgets</tns:description>
  </tns:Endpoint>

</tns:BusinessSpaceRegistry>
<!-- END NON-TRANSLATABLE -->
```

8. Save the files.
9. Copy the `monitorEndpoints.xml` and `monitorABXEndpoints.xml` files to the `<WebSphere_Process_Server_installation>\profiles\<Monitor_profile_name>\BusinessSpace\registryData` directory on all the nodes where Business Space is installed.
10. Restart the servers.

15.9 Monitor models and WebSphere Business Monitor dashboards

To check the functionality of the WebSphere Business Monitor installation, we recommend installing a sample monitor model to check the event consumption and dashboard capabilities. In this section, we go through the steps of deploying a monitor model and examining dashboard features.

15.9.1 Preparing a business process for monitoring

After creating a business process in WebSphere Integration Developer, you should do the following:

1. Choose the required events to be generated and emitted according to the business situation for each business entity, as shown in Figure 15-52.

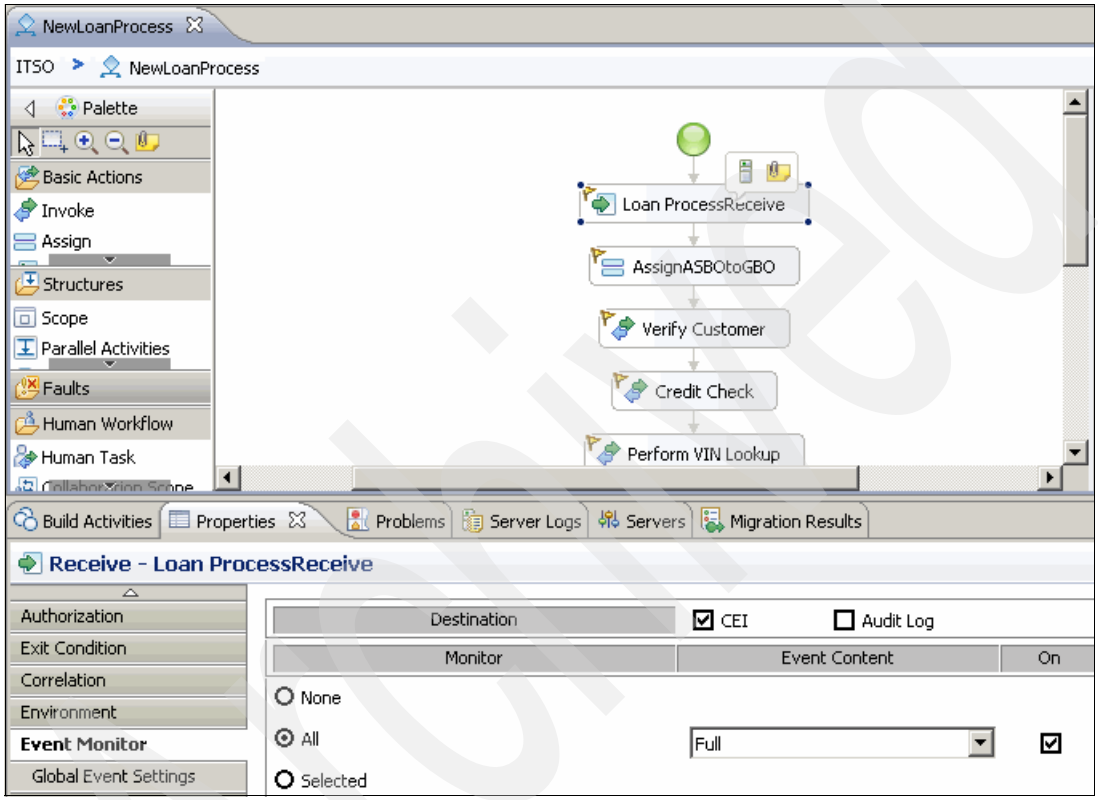


Figure 15-52 Adding events to the BPEL application

2. Generate the corresponding monitor model from the created business process, as shown in Figure 15-53.

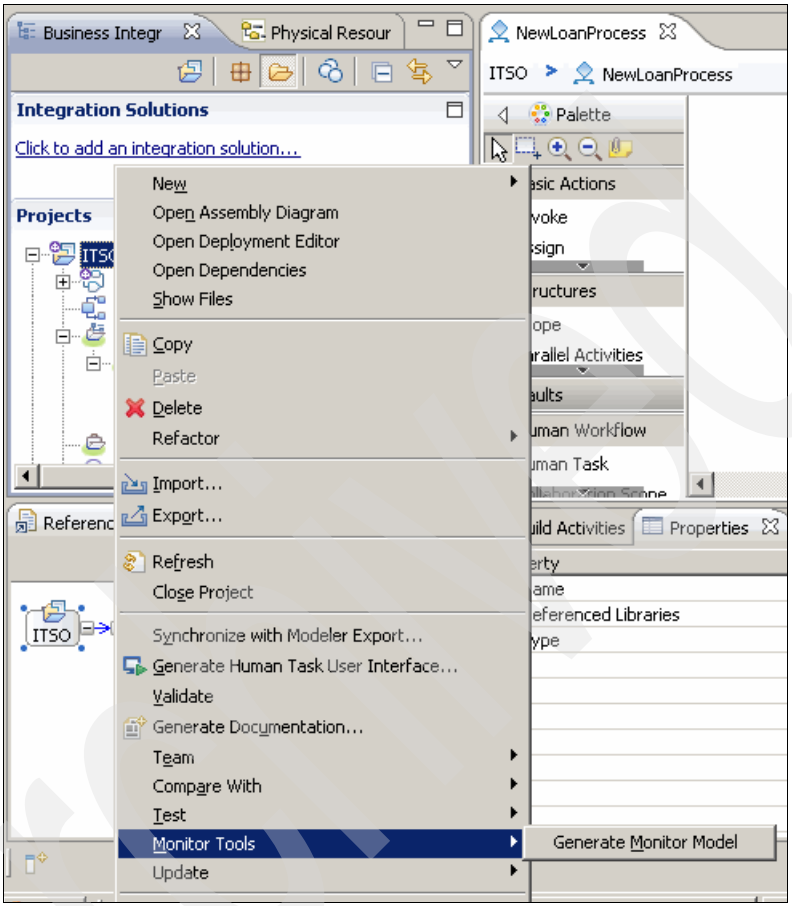


Figure 15-53 Generate the monitor model from the BPEL process

3. During generation, you can select the required events to be monitored from the list of emitted events, as shown in Figure 15-54.

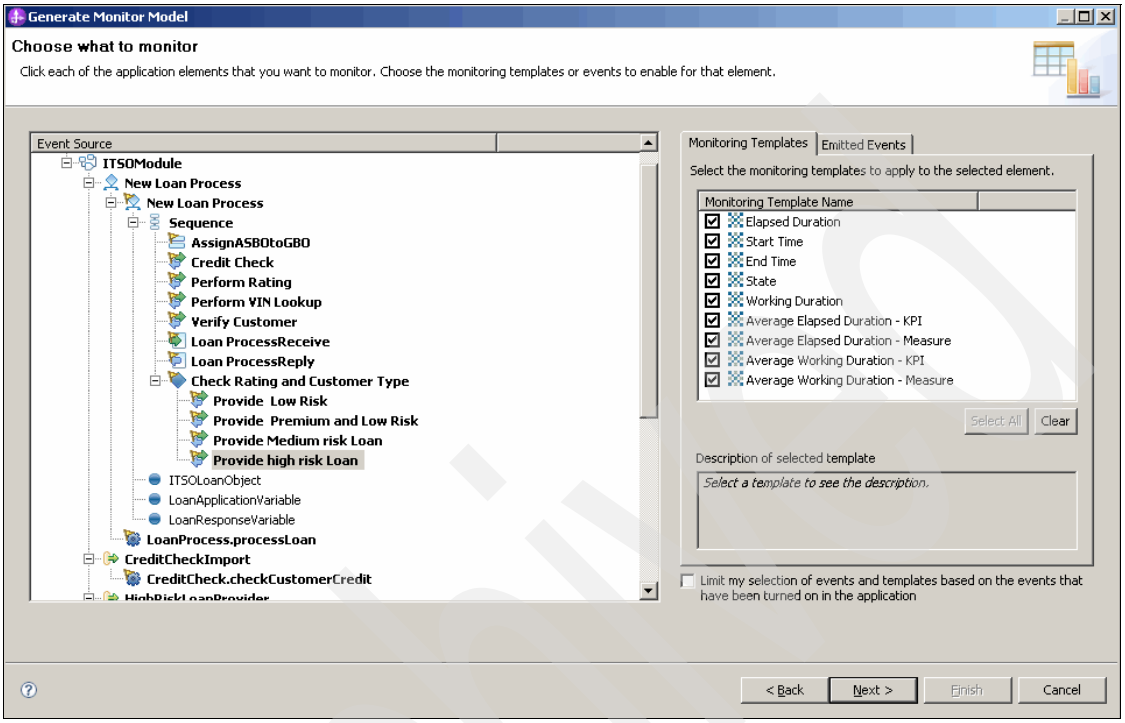


Figure 15-54 Selecting events to be monitored while generating monitor model

15.9.2 Preparing the monitor model in the toolkit

After generating the monitor model, you can modify the model in the Monitoring perspective (the WebSphere Business Monitor toolkit plug-in in WebSphere Integration Developer) to fit the monitoring business requirements. This allows you to create new KPIs, timers, triggers, dimensions, and so forth.

Upon finishing implementation of the monitor model you should generate the corresponding J2EE applications from the monitor model in WebSphere Integration Developer. There will be three projects created:

- Monitor model enterprise application project
- Monitor model logic part project
- Monitor model moderator part project

By exporting the EAR file from any of above projects, you are exporting the monitor model application that you should deploy on WebSphere Business Monitor Server.

15.9.3 Monitor model deployment

The deployment process of the monitoring solution consists of two parts:

- Deploying the monitor model application EAR. This EAR file should be deployed on the WebSphere Business Monitor server on the Model Logic and Model Moderator clusters.
- Deploying the corresponding business application EAR. This should be deployed on the WebSphere Process Server server on the application target cluster.

To deploy a monitor model using a remote CEI configuration:

1. From WebSphere Business Monitor deployment manager Integrated Solutions Console, click **Applications** → **Monitor Models**. Then click **Install**.
2. In the Preparing for application installation panel (as shown in Figure 15-55):
   a. Type the path of the required monitor model ear file.
   b. Select the **Show me all installation options and parameters** option. Click **Next**.

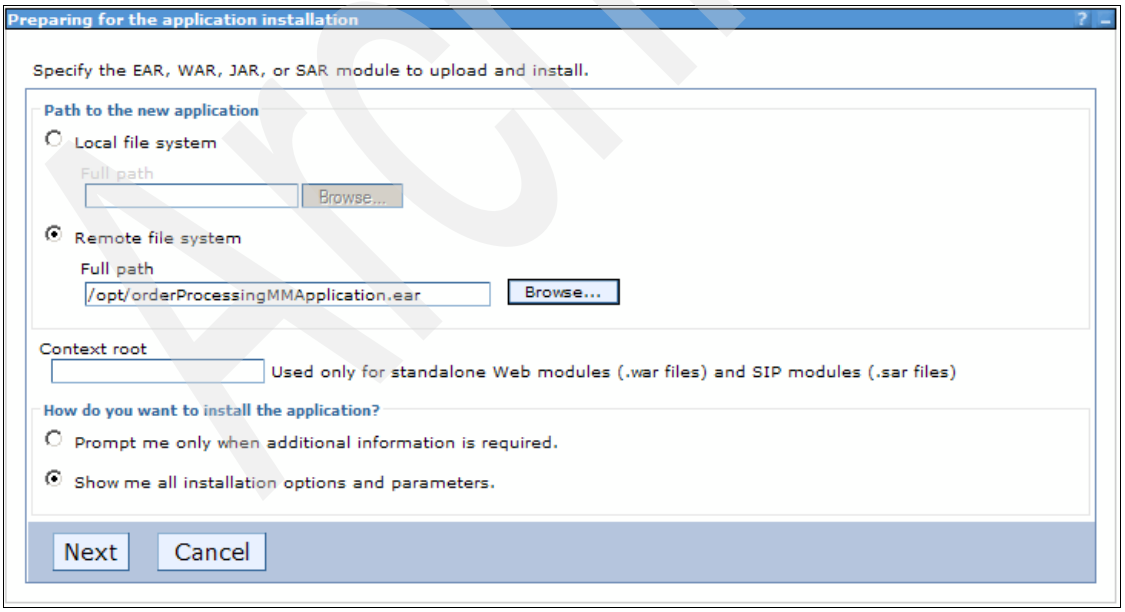


Figure 15-55 Preparing for application installation page

3. In the next panel, click **Next** without any change.
4. In the next panel, click **Continue** without any change. A wizard will be displayed.
5. In Step 1: Select installations options, accept the defaults and click **Next**.
6. In Step 2: Map modules to servers, assign each monitor model module to the corresponding cluster. The MMLogic module of the monitor model will be assigned to the Model Logic cluster and the MMModerator module will be assigned to the Model Moderator cluster. In the current installation, there is only one cluster that processes events (wbm.event cluster). Therefore, you should assign both modules to the wbm.event cluster. To perform this step, select both monitor model modules, select the corresponding cluster, and click **Apply,** as shown in Figure 15-56. Then click **Next**.

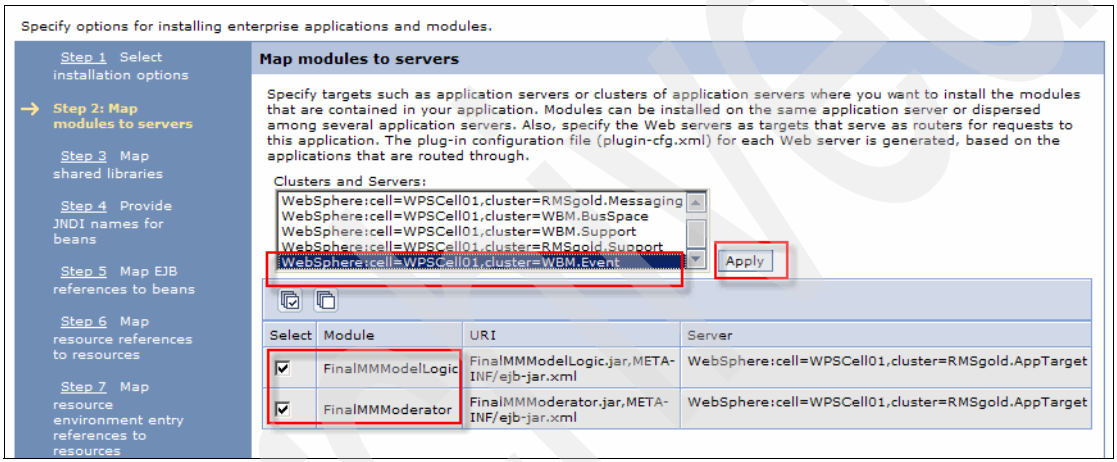


Figure 15-56 Monitor model deployment wizard Step 2: Map modules to servers

7. Keep selecting defaults and click **Next**. If any warning appears, click **Continue**.
8. In Step 12: Select monitor model CEI option, accept the defaults and click **Next**.

Note: In step 12 you can select which event group you want to use and the mode of the monitor model (inactive and active (monitor model queue-based)) after installation, as shown in Figure 15-57 on page 498.

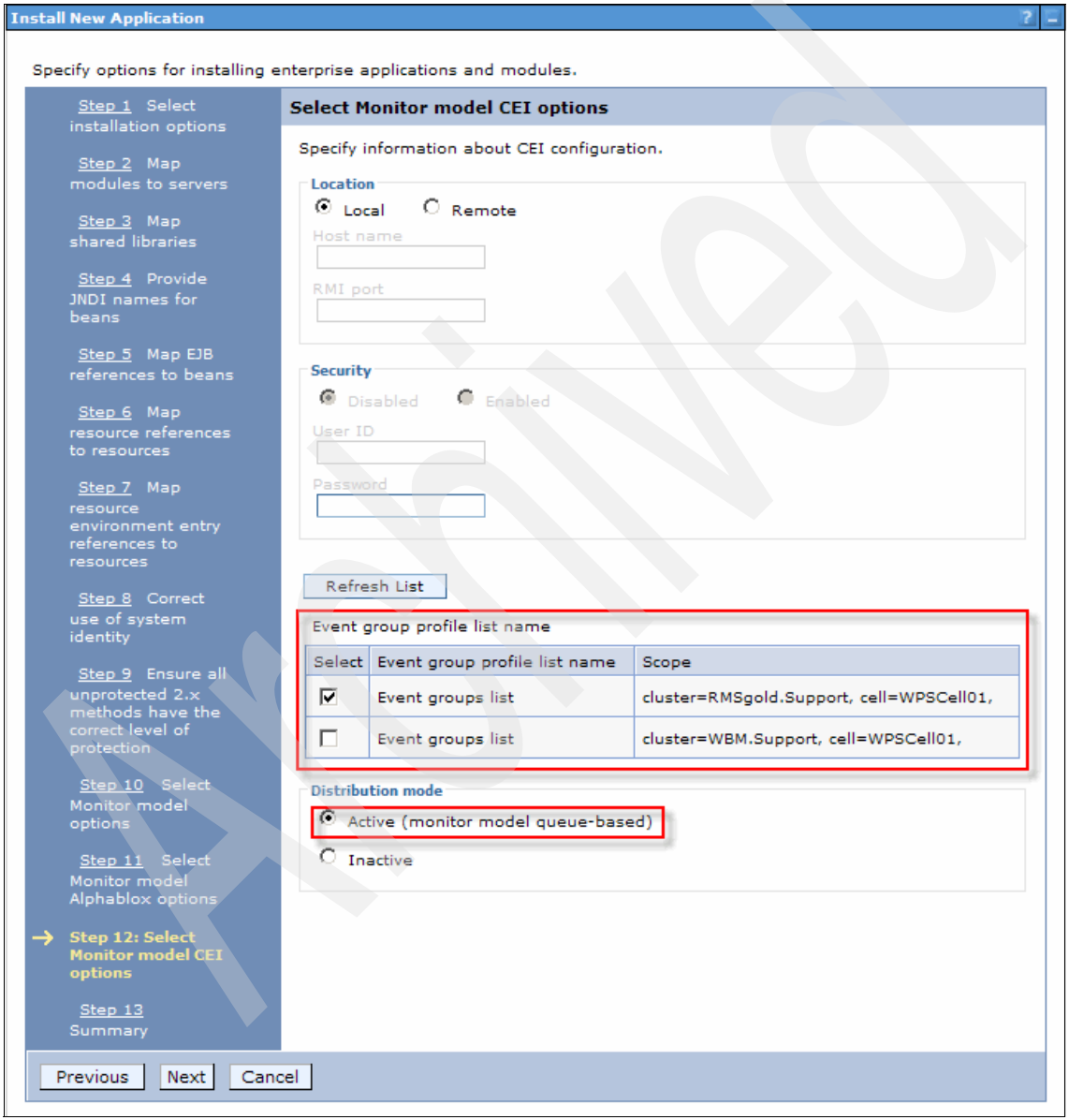


Figure 15-57 Deployment step 12

9. Review the summary and click **Finish**.

For more information about how to deploy a monitor model, refer to the following link for the WebSphere Business Monitor information Center:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.monitor.admin.doc/admin/admin_deploy.html

15.9.4 Inspecting monitor model and business process functionality

After deploying the business process application on the WebSphere Process Server server on the Application Target cluster, you can start to emit events and check the following:

1. The events are emitted correctly from the corresponding deployed BPEL application instance.
2. The corresponding deployed monitor model consumed the events correctly.

To check that the events are emitted correctly from the started instance:

1. Log in to Business Process Choreographer (BPC) Explorer.
2. Click **Process Templates**.
3. Check on the corresponding BPEL process and click the **Start Instance** button, as shown in Figure 15-58.

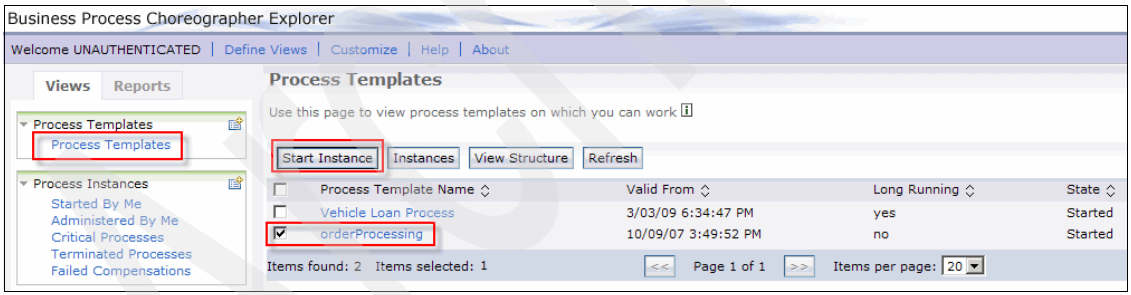


Figure 15-58 BPC: Start a new instance of a BPEL process

4. There should be some events generated and emitted now. You can check the emitted events in **Integration Application** → **Common Base Event Browser** (CBE), as shown in Figure 15-59.

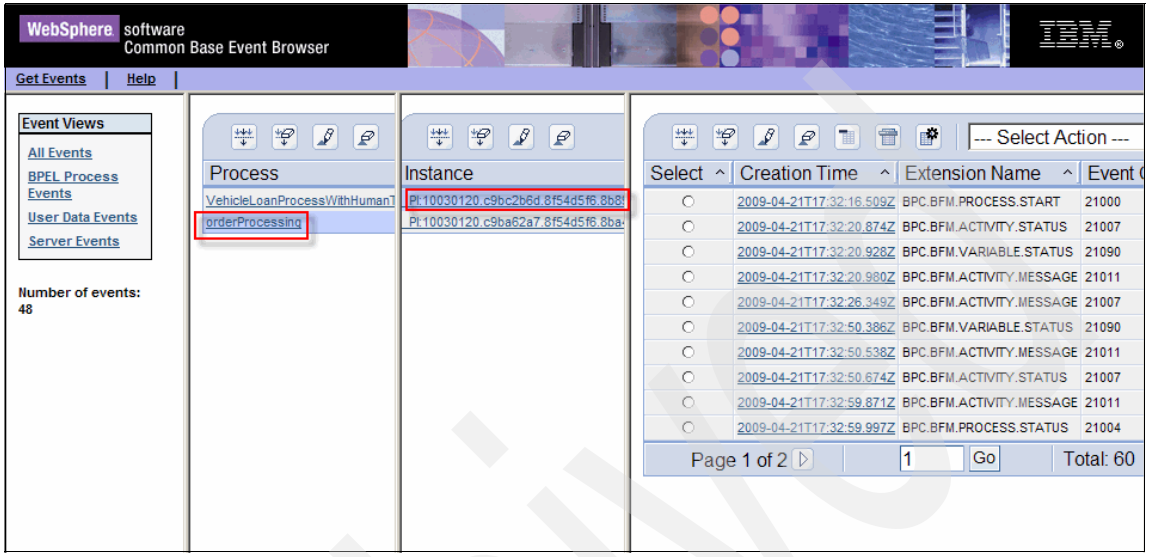


Figure 15-59 CBE browser

To check the consumption of the events by the corresponding monitor model:

1. Emit events using BPC with the corresponding BPEL application.
2. Preview events using the CBE browser.
3. Start the corresponding monitor model.
4. Check the existence of the events using the CBE browser. If the events no longer exist, then the monitor model already consumed them (this will be valid if the datastore is disabled). If the messaging engine datastore is enabled, you must check the event consumption either through dashboard views or by inspecting the corresponding monitor model tables in the Monitor database.

Note: If you are using a remote CEI as an event source, you also must check that events are transmitted correctly to the WebSphere Business Monitor server side. You can perform this by doing the following:

- Check that the SIB link is in running state.
- Stop the corresponding monitor model.
- Emit events at the remote CEI (event source)
- Check the events in the CBE browser at the monitor side.

15.9.5 Monitor dashboards

After deploying the monitor model and ensuring the consistency of the environment and event emission and consumption, you can start to view your dashboards using monitor widgets in Business Space.

You can access the business space dashboards using the following URL:

`http:\\[hostname]:9080\BusinessSpace`

There are eleven dashboard widgets. The following are steps describe how to configure one of these widgets, the configuring instance view widget, as an example:

1. Log in to Business Space.
2. Go to Business Space Manager by clicking **Welcome** → **Manage Business Spaces**.
3. Click the Create new business space icon.
4. Enter the business space name, as shown in Figure 15-60.

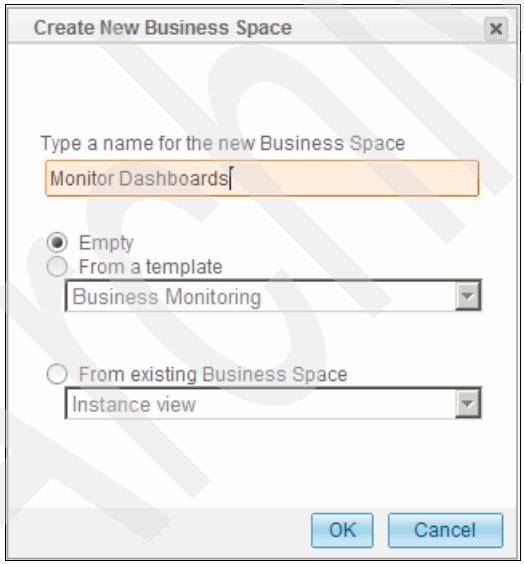


Figure 15-60 New business space

5. Click the Create new page icon.

6. Enter the page name as shown in Figure 15-61.

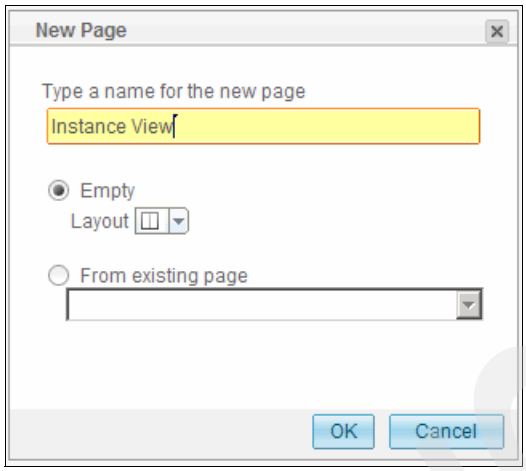


Figure 15-61 New business space page

7. Click the newly created business space page.
8. Click **Add Widgets**, as shown in Figure 15-62.

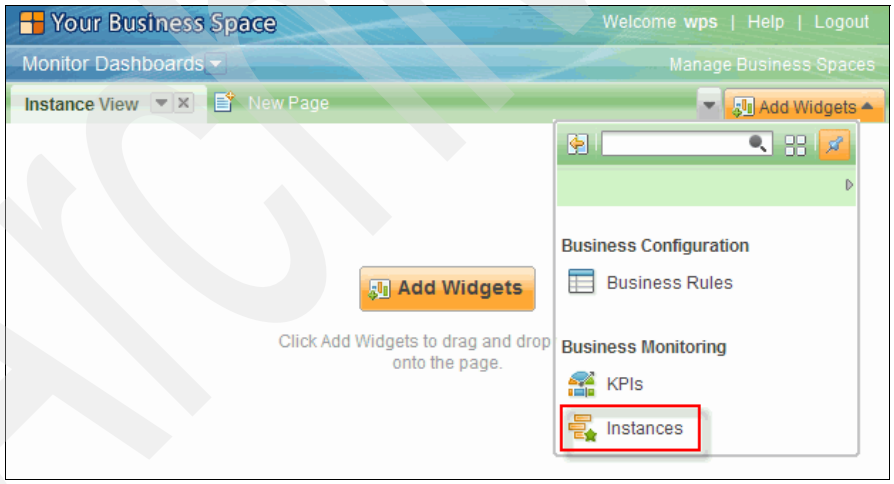


Figure 15-62 Add new widget

9. Select the **Instances** widget and drag an drop it onto the empty page.

10. Click **Configure**, as shown in Figure 15-63, to configure the Instance view.

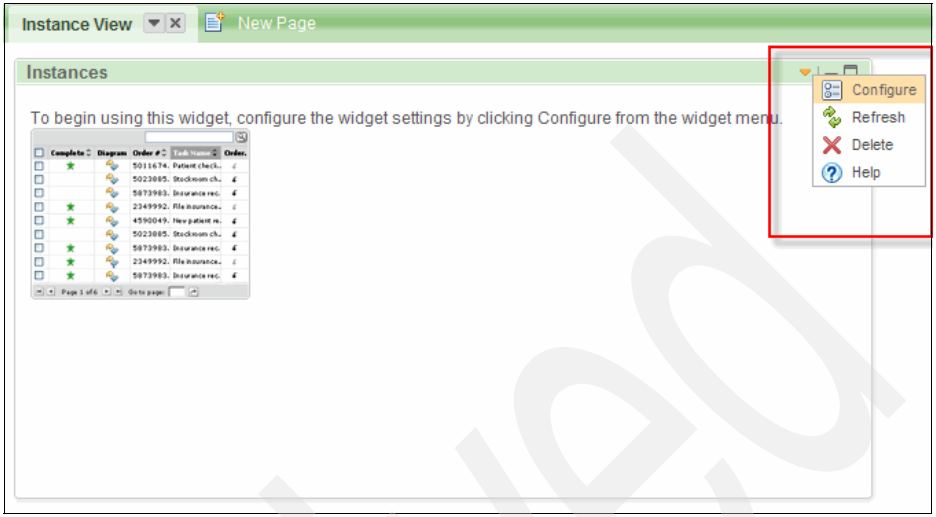


Figure 15-63 Default Instance view

11. In Configuration view, select the monitor model, select the monitor model context, and then select the required columns to be displayed, as shown in Figure 15-64.

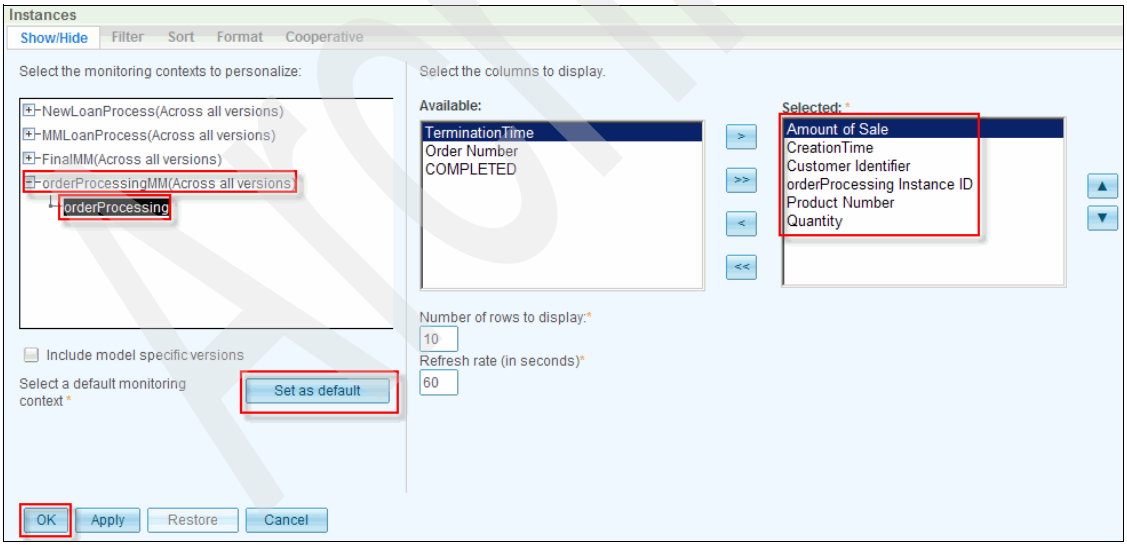


Figure 15-64 Instance view configuration page

> **Note:** If this is the first time that you have configured the instance view, click the **Set as default** button to keep these settings as the default configuration.

The instance view should look like Figure 15-65.

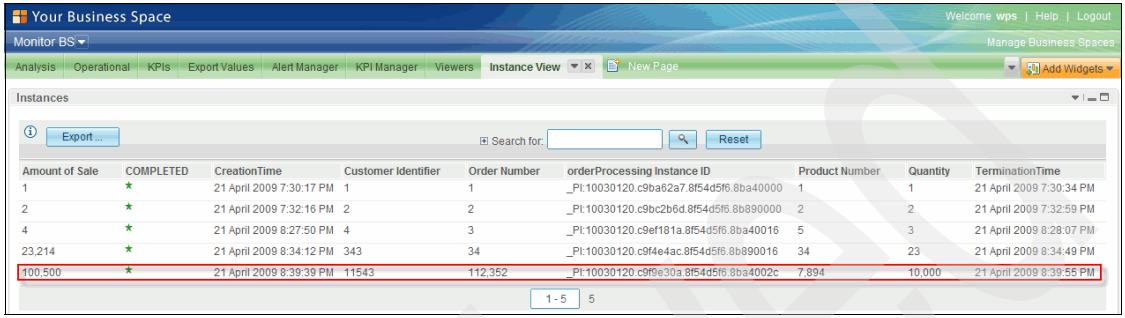


Figure 15-65 Instance view

15.10 Secure WebSphere Business Monitor

When you enable security for WebSphere Business Monitor, you are enabling administrative and application security settings. WebSphere Business Monitor uses many of the security mechanisms provided by the prerequisite products, including WebSphere Application Server.

Configure access to the monitor model resources using Monitor Data Security in the Integrated Solutions Console. For WebSphere Application Server instances that run the WebSphere Business Monitor server, you must configure them to use the federated repository only. They cannot use a local operating system, stand-alone LDAP registry, or stand-alone custom registry directly.

For more information about enabling security of WebSphere Business Monitor refer to the *End to end security lab* available at the following Web page:

`http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.wpi_v6/wbmonitor/6.1/Dashboards.html`

15.11 Maintain WebSphere Business Monitor

In production systems where system high availability is a key requirement, IT monitoring for WebSphere Business Monitor should be planned and maintained

over time. You should plan a manual activity for monitoring operations and performance of WebSphere Business Monitor. This section discusses the key parameters that should be monitored (either manually or using tools) to make sure that WebSphere Business Monitor is running smoothly and that no problems are expected to occur during runtime.

15.11.1 Maintain the WebSphere Business Monitor Server

To maintain a functioning and well-performing system, perform the following tasks:

1. Verify that the Monitor server is running.
2. Verify that the Monitor messaging engine is running.
3. Verify that the Monitor JDBC connections are working correctly.
4. Verify that the Service Integration Bus Link (SIB link) is started if you are using remote Common Event Infrastructure.
5. Verify that messages in monitor models queues are not accumulating.
6. Verify that the sum of all monitor models' queue depths will not exceed the messaging engine maximum number of messages threshold.
7. Check the event consumption rate of WebSphere Business Monitor Server.
8. Verify that WebSphere Business Monitor is not running in error mode (slow or blocked event consumption).

15.11.2 Maintain the WebSphere Business Monitor database

The Monitor database is a key point in the performance of WebSphere Business Monitor. Continuous tuning and maintenance is required. The following is a list of recommended actions:

- Back up the system regularly.
- Check the number of active Monitor Context Instances and determine why they are not being terminated.
- Check the pool size regularly to avoid acquiring locks failures.
- Check the tablespace size regularly.
- Check database sizes regularly to avoid running out of disk space. This includes the Monitor database and messaging engines database.
- Use database tools to refine and tune indexes and tables.

- Run the command RUNSTATS after a significant amount of data has been processed.
- The disk that is used to perform logging to should not be the same disk used to store data.

15.11.3 Performance tuning

The following considerations can positively affect the performance of your system:

- Disable tracing, monitoring, and data store options. Those are used only in problem troubleshooting.
- Disable event logging to the event database (CEI datastore).
- Do not use the default Derby as a database. For high performance, use a database management system such as DB2.
- Enable security only where practical.
- Use appropriate hardware configuration for performance measurement. For example, ThinkPads and desktops are not appropriate for realistic performance evaluations.
- Do not run a production server in development mode or with a development profile.
- Do not use the Unit Test Environment (UTE) for performance measurements.
- Configure for clustering (whenever applicable).
- Configure thread pool sizes appropriately according to the needs.
- For DB2, optimize Buffer Pool Size.
- Set the heap and nursery sizes to manage memory efficiently, and select the appropriate garbage collection policy.
- Set the message consumption patch size according to workload (flow of events).

Part 4

Four Cluster production topology

16

Creating a WebSphere Dynamic Process Edition production topology

This chapter describes step-by-step how to build the WebSphere Dynamic Process Edition production topology. The products included in this topology are:

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

16.1 Overview

> **Note:** In a scenario where WebSphere Process Server and WebSphere Business Monitor only will exist, the Four Cluster topology is the preferred topology to be used. This chapter can be used to build that topology. All that must be done is to skip over any Fabric-specific steps in this chapter.

The WebSphere Dynamic Process Edition production topology consists of four clusters:

- The Messaging cluster will have the required buses:
  - SCA.SYSTEM bus
  - SCA.APPLICATION bus
  - CEI bus
  - BPC bus
  - Monitor bus
  - Fabric bus
- The Application cluster will contain:
  - BPC
  - HTM and BFM
  - BPEL applications
  - Monitor model (Moderator/Application logic)
- The Support cluster will contain:
  - CEI
  - Action Services
  - Data services scheduler
  - Monitor Emitter Service
- The Web cluster will contain:
  - Business Space
  - REST API Services
  - BPC Tools
  - Business Rules Manager
  - Monitor widgets
  - Monitor Alphablox widgets
  - Monitor mobile dashboards
  - Fabric widgets
  - WPS widgets
  - Alphablox
  - Fabric EARs

One of the main reasons for introducing the Web cluster is to recognize that with WebSphere Business Monitor, the event filtering will make heavy usage of the CEI server. It is because of this concern that the Web Cluster is introduced and

applications are moved to that cluster where they will perform better. It is advisable to distribute the applications as shown above.

Business Space, for example, which would have been distributed by default to the Support cluster (when using the Deployment Environments wizard), will now be moved to the Web Cluster.

It is important to also point out that the Deployment Environments wizard cannot be used to generate this topology. The reason is that the Deployment Environments wizard is not able to deal with custom nodes that are augmented with WebSphere Business Monitor.

Note: Even it you alter the deployment environment descriptor to support four clusters, it will fail. The deployment environments wizard only expects three clusters.

This chapter describes the necessary steps to build this production topology manually. Figure 16-1 illustrates the Four Cluster topology.

> **Note:** In Figure 16-1 on page 512, the messaging engines appear to be duplicated, but they are not. The lighter colored member on the right-hand side is there to imply that there is a backup in case of failure. Only one messaging engine would ever be active at a time. It is a common practice to distribute a messaging engine (such as Monitor's messaging engine) to its own member within a cluster for performance reasons. Chapter 3, "Business Process Management production topologies" on page 37, discusses how messaging engines are typically distributed and the considerations that come into play.

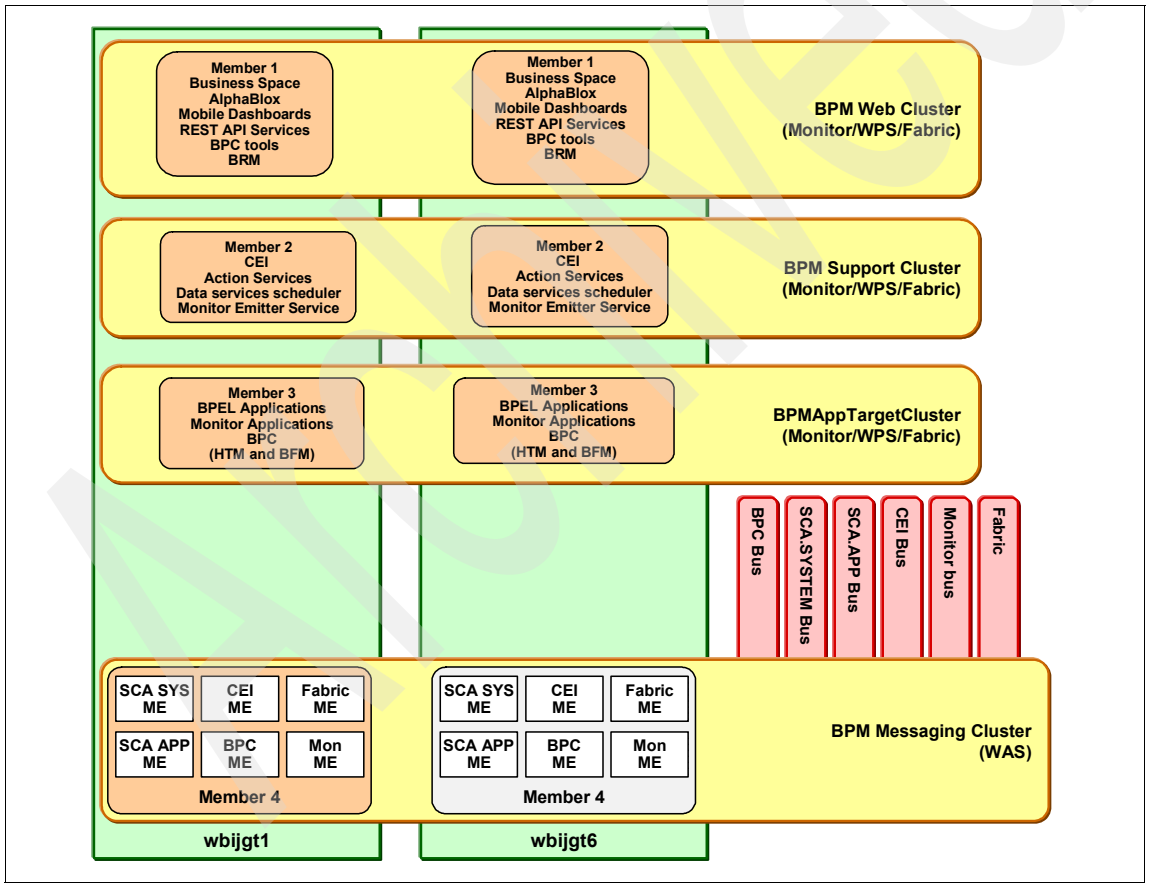


Figure 16-1 Four Cluster topology

Overview of this topology

The previous section described some of the logical layout of this production topology. In this section we discuss some of the physical details. Figure 16-2 represents the physical layout. (Note that only machines wbijgt1 and wbijgt6 will be used in this topology.) Some comments regarding this sample topology are:

- A single cell will be created.
- Four Clusters will be created, each with two members. The cluster members will be spread across two physical AIX systems
- A separate physical AIX system is used for:
  - The deployment manager
  - The HTTP Server (Version 6.1)
  - DB2 Enterprise Server v9.1
- The IBM Tivoli Directory Server V6 is on a standalone machine.

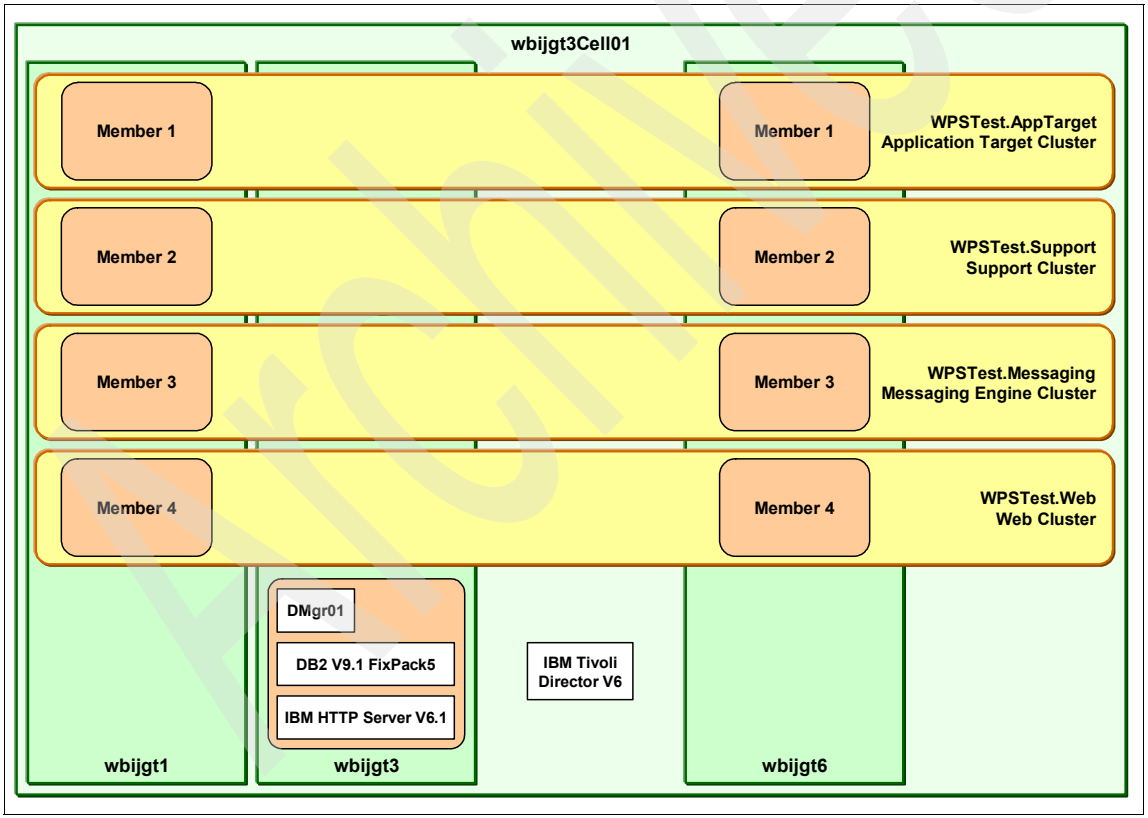


Figure 16-2 WebSphere Dynamic Process Edition sample topology

16.2 Preparing the machines

There are two specific tasks that must be done on any Linux or UNIX system such as the AIX Operating System prior to installing any of the products in this topology.

> **Note:** On any of the Linux or UNIX operating systems, it is very important to set the ulimit before following the steps in this chapter. If this is not done, problems during installations have been known to happen. This can very easily be avoided to save time.

- Check the setting for the maximum number of open files by examining the `etc/security/limits` file on an AIX operating system. There should be a line for each user ID or a default value. Look for *nofiles* and ensure that it is set to 10240. Example 16-1 shows how this value should be set for the root user ID. If root is not the user ID being used for installation, make sure that whatever user ID is being used has nofiles set properly.

  Example 16-1 Example limits of file settings on AIX

  ```
  default:
     fsize = 2097151
     core = 2097151
     cpu = -1
     data = 262144
     rss = 65536
     stack =65536
     nofiles = 2000
  root:
     fsize = -1
     nofiles = 10240
  ```

- If it is a Linux operating system, then the file name to examine is `etc/security/limits.conf`. (See Example 16-2.)

  Example 16-2 Example limits.conf file settings on Linux

  ```
  * hard nofile 10240
  * soft nofile 10240
  ```

- Set the umask value to 022 by issuing the command:

  "`umask 022`"

Sample topology-specific information

Sample topology-specific information is:

- In this sample topology, we assumed that the WebSphere Process Server was installed on a file system mounted as `/monwas`.
- `/monwas/ProcServer` is the root install directory.
- All custom profiles created will be stored in `/monwas/ProcServer/profiles`.
- When WebSphere Business Monitor is installed, it will use the existing level of WebSphere that was installed by WebSphere Process Server.
- The IBM Tivoli Directory Server is located on host name `wbi602a.raleigh.ibm.com`.
- The host name `wbijgt3.rtp.raleigh.ibm.com` will contain:
  - The deployment manager for this cell.
  - The IBM HTTP Server.
  - DB2 v9.5 Enterprise Server: The DB2 port number in this topology will be 55567.

Note: The IBM HTTP Server can be installed at anytime. For information about how to install the IBM HTTP Server, refer to 5.1.3, “Add a Web server” on page 103.

16.3 Installing WebSphere Process Server

The first product that will be installed is the WebSphere Process Server. During this installation, only the product binaries will be installed and no profiles will be created.

Note: APAR JR31348 is required in order to properly be able to install the BPC Explorer in this topology. This APAR fix is available through the IBM Support site. It is also integrated into WPS fix pack v6.2.0.1.

Refer to “Installing WebSphere Process Server base product” on page 102.

16.4 Installing WebSphere Business Monitor

The next product to install is the WebSphere Business Monitor. During this installation, only the product binaries will be installed and no profiles will be

created. Refer to Chapter 15, “Incorporating WebSphere Business Monitor into a production topology” on page 433.

16.5 Installing WebSphere Business Services Fabric

The next product to install is the WebSphere Business Services Fabric. During this installation, only the product binaries will be installed and no profiles will be created. Refer to Chapter 14, “Incorporating WebSphere Business Services Fabric into a Remote Messaging and Remote Support topology” on page 385.

Note: It is important to make sure that the mandatory iFix JR31349 has been installed immediately after the product binaries have been installed. This is described in 14.2.2, “Install interim fix” on page 392.

There is also an interim fix 31376 that is needed for the Fabric widgets. This is described in 14.2.8, “Apply interim fix 31376” on page 414.

16.6 Creating databases

There are several databases that are required by all the products in WebSphere Dynamic Process Edition topology. Table 16-1 identifies the needed databases.

Note: In the topology illustrated in this chapter, separate databases are used. It is possible to combine databases as long as unique schemas are used. However, it is also important to point out that the databases for MONITOR, FABRIC, and Business Space should not be combined with other WebSphere Process Server databases. The reason for this is that as your needs grow, different options exist on how to scale your topology. Some recommendations call for creating a second WPRCSDB database as example.

The MONITOR, Business Space, and FABRIC databases should not be duplicated, but they can be combined.

Table 16-1 WDPE topology databases

Database instance	Description	Comments
WPRCSDB	Process Server Common database	Create with schema COMMONDB.

Database instance	Description	Comments
EVENT	Event server data source	The database will be created now but not the database tables.
MEDB	CEI Messaging engine data source	Create with schema CEIME.
MEDB	SCA System Bus Messaging engine data source	Create with schema name SCASYS.
MEDB	SCA Application Bus Messaging Engine data source	Create with schema SCAAPP.
BPEDB	Business Process Choreographer data source	Create with schema BPC.
MEDB	Business Process Choreographer Messaging Engine data source	Create with schema BPCME.
OBSVRDB	Business Process Choreographer Event Collector data source	Create with schema OBS.
MONITOR	Monitor data source	Create with schema MONITOR.
MONITOR	Business Space data source	The best practice is to create the Business Space tables in the Monitor database. Create with schema IBMBUSSP.
MEDB	Monitor messaging engine data source	Create with schema MONITOR.
FABRICDB	Fabric data source	

Here we list how to create these databases:

- For information about creating the WPRCSDB, MEDB, EVENT, OBSRVRDB, and BPEDB databases refer to 5.2, “Database creation” on page 103.
- Creating WebSphere Business Monitor Database.

  In this topology, we name our database `MONITOR` and use the schema name `MONITOR`. Refer 15.3.2, “Creating the WebSphere Business Monitor databases” on page 439, for information about how to create the database.

  > **Note:** It is worth mentioning here that during installation time, it is possible to generate the database scripts that contain the values (such as schema name) already filled in. The scripts can then be passed to your DBA to be examined and executed.

- Creating the WebSphere Business Services Fabric Database:
  a. Open a command line window.
  b. Type `su - db2inst1`.
  c. Change the directory to where Fabric was installed and change to the directory `<Install Directory>/WebSphere/Fabric/FoundationPack/configuration/database/db2/db2`.
  d. Type `db2 -tvf create_fabric_db_linux.sql`.
- Creating Business Space database tables:
  a. The MONITOR database will be used to store the Business Space database tables. This is a best practice to follow. There is no reason to have a separate database created just for the Business Space tables.
  b. Locate the `createTable_BusinessSpace.sql` file in the `/monwas/WebSphere/ProcServer/dbscripts/BusinessSpace/DB2` directory. This file must be edited and then copied over to the database machine.
     i. Change @SCHEMA@ to some value, such as IBMBUSSP.
     ii. Change @TSDIR@ to a suitable location (for example, an allocated file system space that was assigned by your DBA).
     iii. Save your changes.
  c. Log in to the DB system as the instance owner.
  d. Type:

     ```
     db2 connect to MONITOR user db2inst1 using ‘passwOrd’
     ```

e. Type:

```
db2 -tf createTable_BusinessSpace.sql
```

- The Monitor messaging engine database can be placed in the MEDB along with the other messaging engine tables. Follow these steps:

  a. Log in to the deployment manager and at the command line enter:

  ```
  cd /monwas/ProcServer/bin
  ```

  b. Run this command:

  ```
  ./sibDDLGenerator.sh -system db2 -platform unix -schema MONITOR
  -user db2inst1 -statementend \; > /tmp/monME.ddl
  ```

  c. Log in to the DB2 system as the instance owner and then run these commands:

  ```
  db2 connect to MEDB user db2inst1 using ‘passw0rd’
  db2 -tf monME.ddl
  db2 connect reset
  ```

16.7 Creating the Deployment Manager profile

The next task to perform is the creation of the deployment manager profile.

Note: When creating the deployment manager profile, be sure to enable administrative security and specify a user ID and password. (In our example, we use a user ID *admin* with a password of *admin*, as shown in Figure 16-3.) Experience has shown that it is easier to set up security from the beginning once the deployment manager profile has been created and augmented.

When administrative security is enabled, it uses a default file system repository. This will later be changed to use the LDAP repository provided with this book.

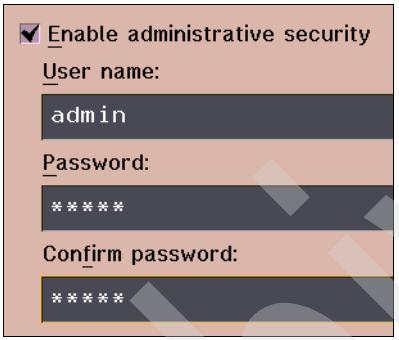


Figure 16-3 Enable administrative security

Create a deployment manager profile for the WebSphere Process Server. The steps to accomplish this task are detailed in 5.3.1, “Deployment manager profile” on page 113.

16.7.1 Augmenting the Deployment Manager profile with WebSphere Business Monitor

The next step is to augment the deployment manager profile with WebSphere Business Monitor:

1. Launch the profile management tool ./pmt.sh.
2. Click **Augment**.
3. Click **Next** → **Next**.
4. Click **Dmgr01** → **Next**.
5. On Select the augment to apply to the selected profile panel, click **WebSphere Business Monitor deployment manager** → **Next**.
6. Click **Advanced profile augmentation** → **Next**.

7. Enter the user ID and password (`admin` and `admin` in our case) and then click **Next**.
8. On the Database Configuration panel (see Figure 16-4):
   a. For Database product select **DB2 Universal Database**.
   b. Set Database creation options to **Use an existing database**.
   c. Specify `MONITOR` for the database name.
   d. Specify `MONITOR` for the schema name.
   e. Click **Next**.

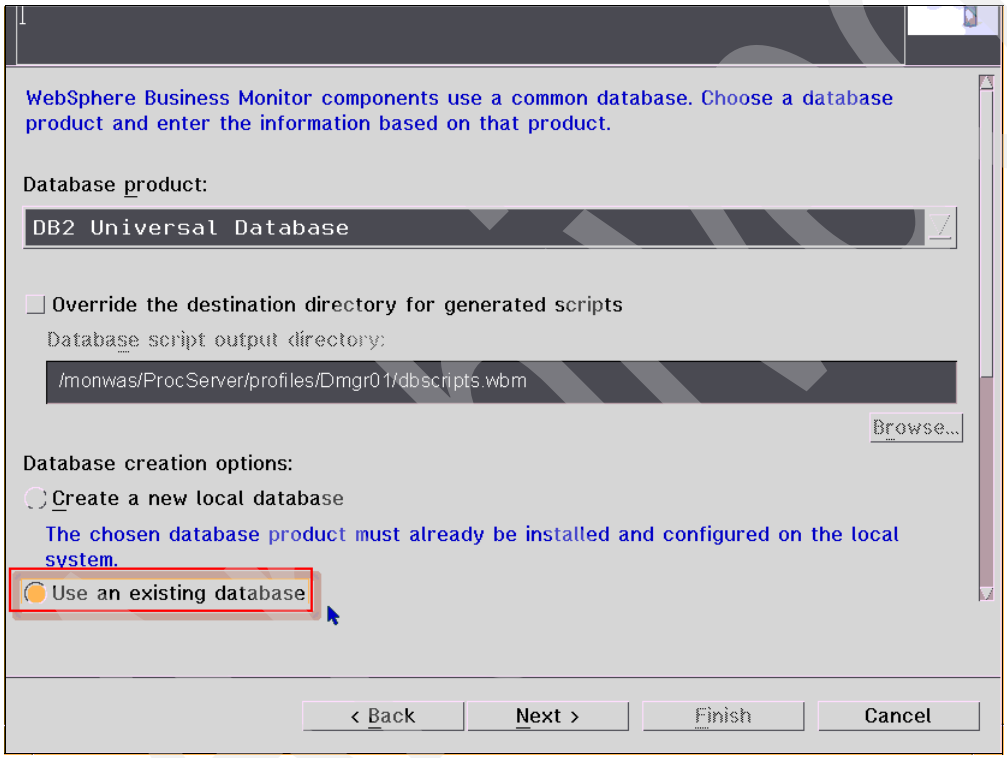


Figure 16-4 WebSphere Business Monitor Database product panel

9. On the panel Database Configuration (Part2), set the following fields:
   a. Set User name to `db2inst1`.
   b. Set Password to `passw0rd`.
   c. Set Confirm password to `passw0rd`.
   d. Set Location (directory) of JDBC driver classpath files to `/monwas/ProcServer/universalDriver.wbm/lib`.

e. Set JDBC driver type to `4`.

f. Set Database server host name or IP Address to `wbijgt3.rtp.raleigh.ibm.com`.

g. Set Database TCP/IP service port or listener port to `55567`. (This is the port used in our topology. Your value may be different.) Click **Next**.

h. On the Profile Augmentation Summary panel, click **Augment**.

i. Once the profile management tool completes, a message should appear indicating that the tool has augmented the profile successfully. At this point, uncheck the Launch the WebSphere Business Monitor first steps check box.

j. Click **Finish**.

16.7.2 Augmenting the Deployment Manager profile with WebSphere Business Services Fabric

The next step is to augment the deployment manager profile with WebSphere Business Services Fabric. This task has been documented in 14.2.3, “Augment WebSphere DMGR profile with Fabric DMGR profile” on page 394.

16.7.3 Starting the Deployment Manager

Once the deployment manager profile has been augmented, it is a good practice to start it:

1. The first indication that there are no problems is that your console shows the text shown in Example 16-3.

Example 16-3 Example console log from deployment manager startup

```
/monwas/ProcServer/bin
# ./startManager.sh
ADMU0116I: Tool information is being logged in file

/monwas/ProcServer/profiles/Dmgr01/logs/dmgr/startServer.log
ADMU0128I: Starting tool with the Dmgr01 profile
ADMU3100I: Reading configuration for server: dmgr
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server dmgr open for e-business; process id is 589846
#
```

2. It is a good practice to check the `SystemErr.log` and `SystemOut.log` files as well.

3. Open a browser and point to the URL:

   `http://wbijgt3.rtp.raleigh.ibm.com:9060/ibm/console`

4. Sign in to the Integration Solutions Console using the user ID `admin` and the password `admin`.

5. Another test that can be performed is to ensure that all of our JDBC resource connections are working properly. Click **Resources** → **JDBC** → **Data sources** and a panel listing our resources should appear, as illustrated in Figure 16-5.

6. Select **All** and click **Test Connection**. All connections should be successful.

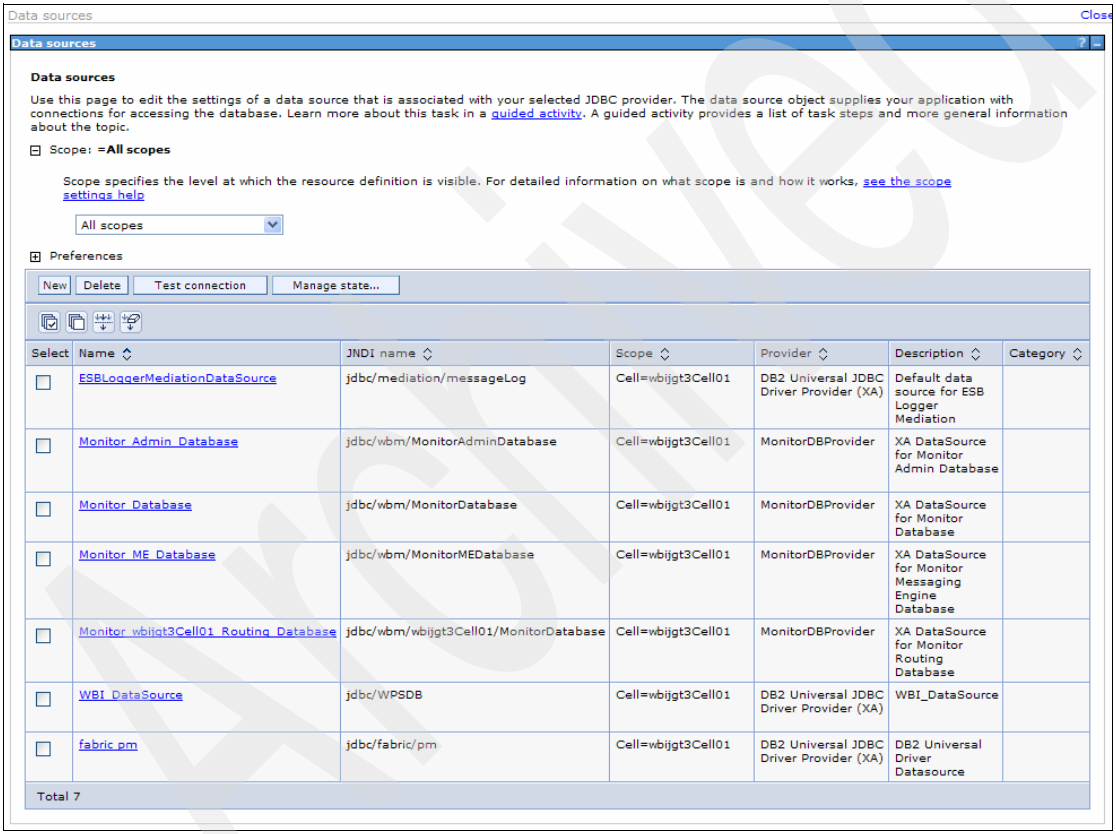


Figure 16-5 List of JDBC resources after creating the Deployment Manager profile

16.7.4 Enabling security to use federated repository

In this sample topology, this would be the ideal point to enable our LDAP security since it is a little easier to make the change now rather than later. To enable the use of federated repository:

1. Start the deployment manager.
2. Sign in to the Integration Solutions Console using the user ID `admin` and the password `admin`.
3. Click **Security** → **Secure administration, applications, and infrastructure**.
4. On the Secure administration, applications, and infrastructure panel, look under User account repository and click **Configure**, as shown in Figure 16-6.

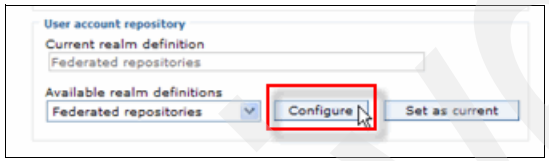


Figure 16-6 Configure button

5. On the Federated repositories panel, under Related Items, click **Manage repositories**, as shown in Figure 16-7.

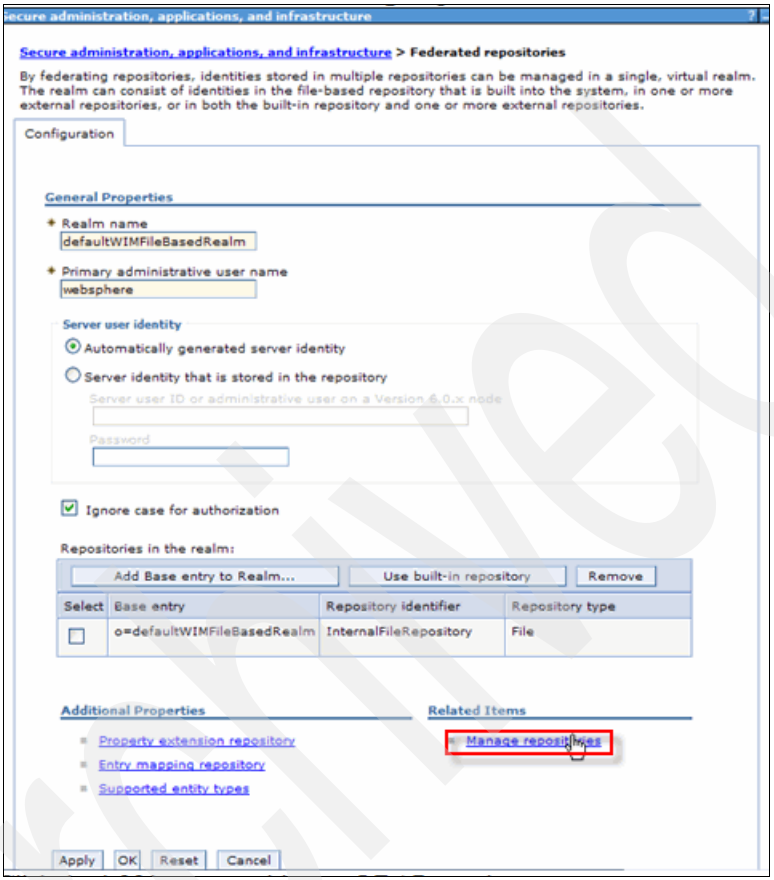


Figure 16-7 Federated repositories configuration panel

6. On the Manage repositories panel, there should be one entry for the file Internal file repository that is used by default. (See Figure 16-8.)

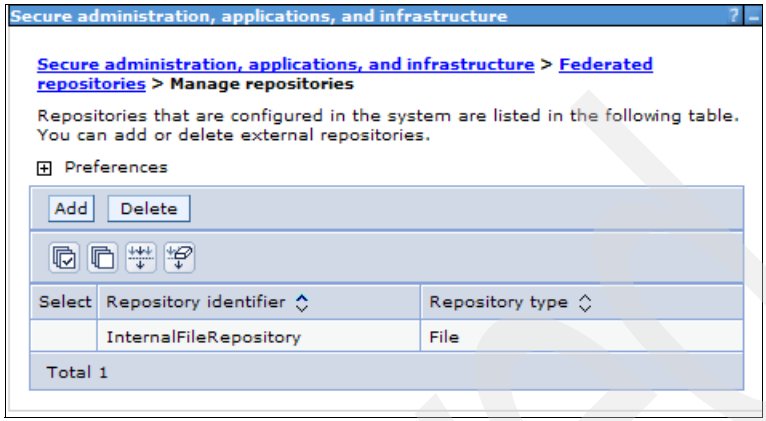


Figure 16-8 Manage repositories panel

7. Click **Add** to create our new federated repository.

8. On the New Repository panel (Figure 16-9), set the following fields:
   a. Set Repository identifier to `wbi602a.raleigh.ibm.com`.
   b. Set Directory type to `IBM Tivoli Directory Server Version 6`.
   c. Set Primary host name to `wbi602a.raleigh.ibm.com`.
   d. The Port value of 389 is correct.
   e. Set Bind distinguished name to `uid=wps,cn=People,O=IBM`.
   f. Set Bind password to `passw0rd`.
   g. Click **OK**.

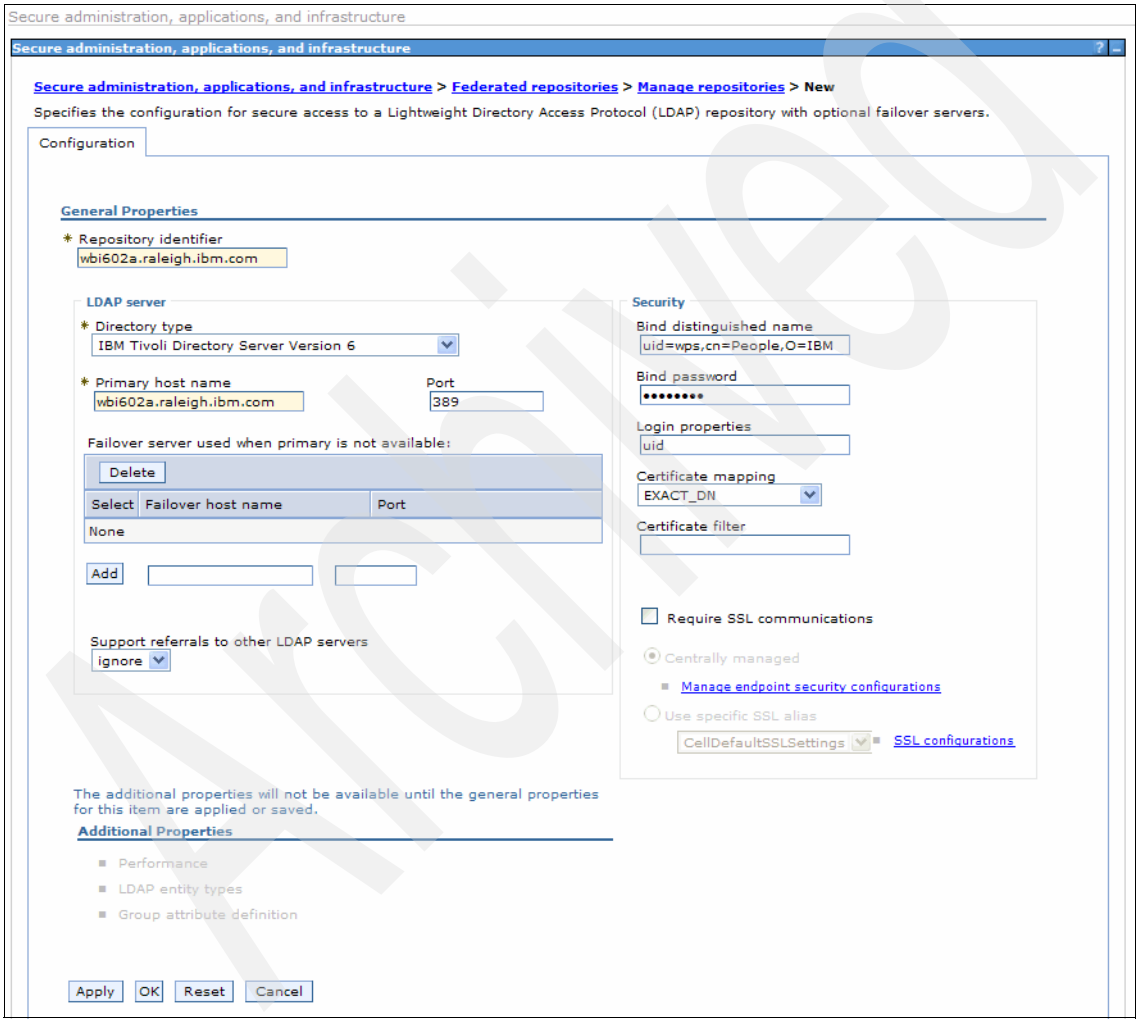


Figure 16-9 New repository configuration panel

   h. Click **Save** to save your configuration changes.

At this point, we should now see two federated repositories listed, as shown in Figure 16-10.

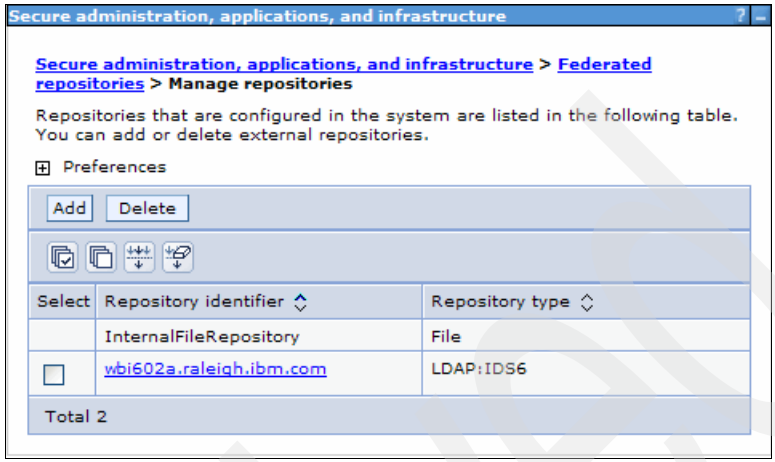


Figure 16-10 Manage repositories panel

9. Click the bread crumb **Federated repositories** to return to this panel.
10. Click **Add Base entry to Realm**, as shown in Figure 16-11.

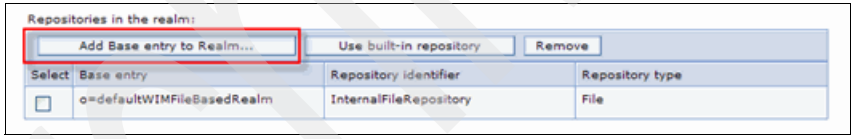


Figure 16-11 Add Base entry to realm

11. On the next panel (Figure 16-12), set the following fields:

   a. Set “Distinguished name of a base entry that uniquely identifies this set of entries in the realm” to `cn=People,O=IBM`.

   b. Set “Distinguished name of a base entry in this repository” to `cn=People,O=IBM`.

   c. Click **OK**.

   d. Click **Save** to save your configuration changes (Figure 16-12).

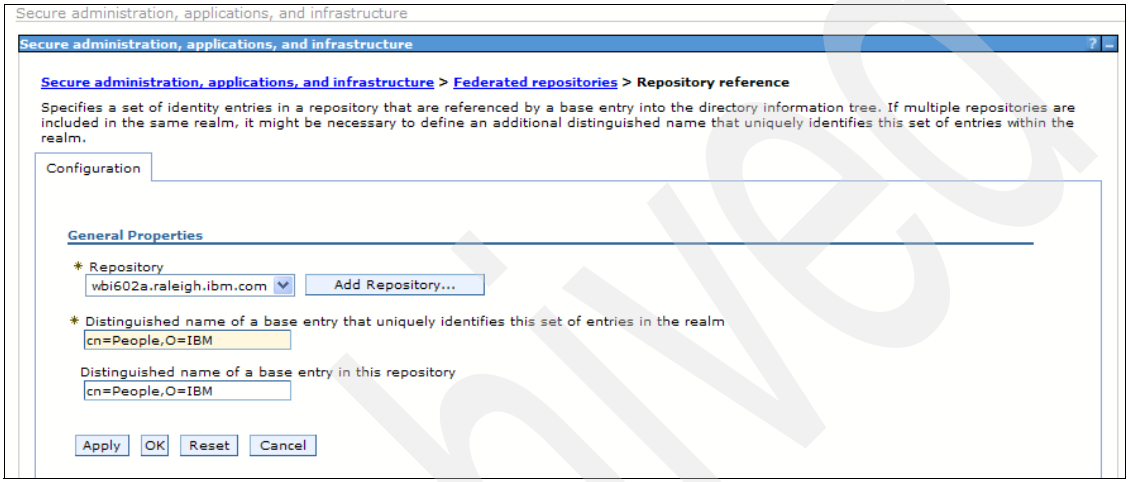


Figure 16-12 Enter information

   e. The next step is to remove the default InternalFileRepository repository. Note that this is an optional step and is not required. To remove this repository, perform the following:

      i. Click the bread crumb to get to the Federated repositories panel.

      ii. Check the check box for the Repository Identifier that has the value **InternalFileRepository**.

      iii. Click **Remove**.

      iv. Click **Save** to save your configuration changes.

12. There is one more resource monitor change to make in order to finish enabling the use of LDAP security. Click the bread crumb to get to the Federated repositories panel. The panel shown in Figure 16-13 should now appear. Change the following:

    a. Set Realm name to be `wbi602a.raleigh.ibm.com`.

    b. Set Primary administrative user name to be `wps`.

    c. Click **OK**.

    d. Click **Save** to save your configuration changes.

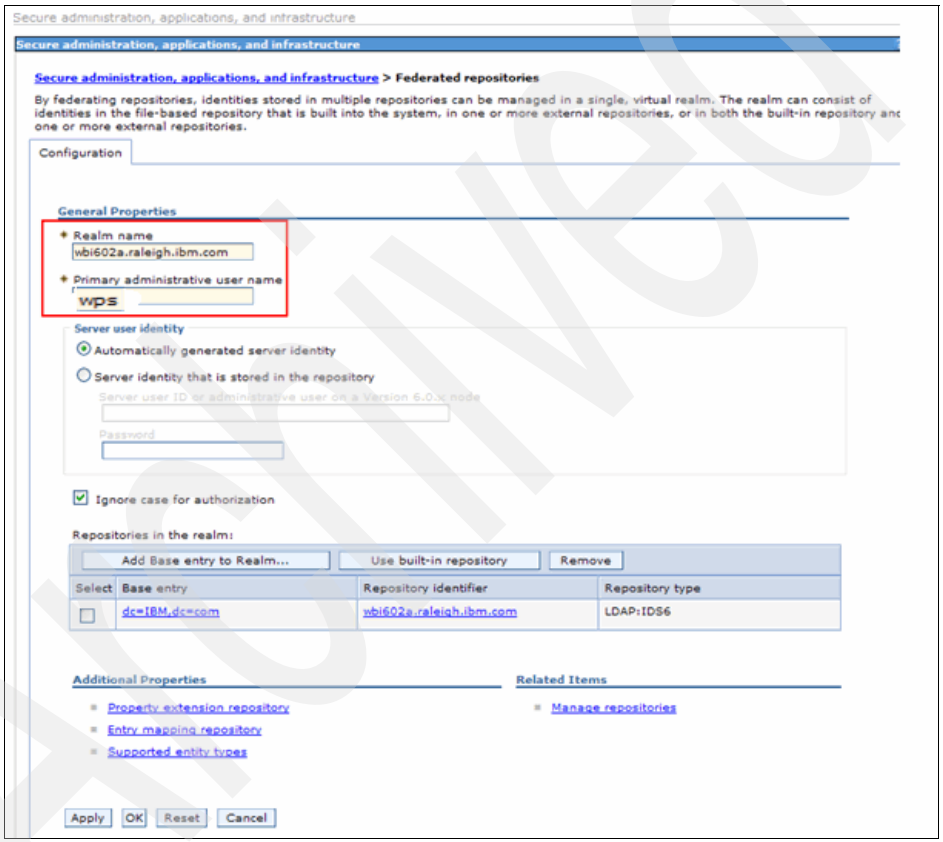


Figure 16-13 Set Realm name

13. To summarize at this point, we have now changed our security such that instead of using a file-based repository, we are now using LDAP. When we first created our deployment manager profile and set administrative security, we specified a user ID of `admin` with a password of `admin`. Now what we have

done is changed such that we will now use user ID `wps` with a password `passw0rd` when signing into the Integrated Solutions Console in the future.

14. Stop the deployment manager and then restart it.
15. When the deployment manager has finished starting, bring up a browser and try to sign in to the Integrated Solutions Console with the new user ID and password.
16. This is also a good time to examine the user IDs that currently exist and are set to admin. They should be modified to use our new LDAP user ID wps. Follow these steps:
    a. Go to the **Integrated Solutions Console**.
    b. Click **Security** → **Secure administration, applications, and infrastructure**.

c. Expand **Java Authentication and Authorization Service** and then click **J2C authentication data**. The list of authentication aliases appears as shown in Figure 16-14.

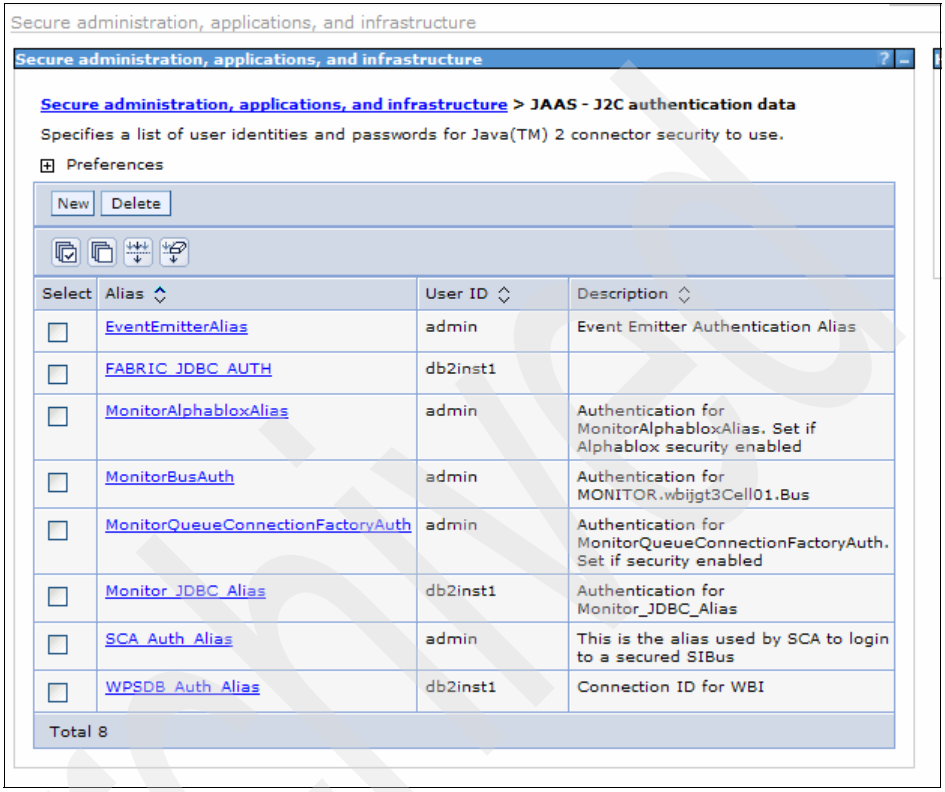


Figure 16-14 List of authentication aliases created after creating a deployment manager profile

d. Examine all user ID values looking for admin and change them to `wps`. The password should be `passw0rd`. (Be careful not to change any JNDI references.)

e. Save all changes made.

16.8 Creating nodes

In this topology there are three physical machines involved. The third machine,wbijg3, hosts the deployment manager and DB2. All machines involved

are running AIX 5.3. In this sample topology, each cluster has two members. Table 16-2 describes the custom profiles and which hosts they will be created on.

> **Note:** The order here is important. You *must* create the WebSphere Process Server custom node first and then augment it with Monitor. There are two reasons for this:
>
> - You cannot augment a federated node with WebSphere Process Server.
> - Monitor forces you to federate when creating a custom node.

Table 16-2 List of custom profiles to create

Profile name	Host name	Comment
Node01	wbijgt1.rtp.raleigh.ibm.com	
Node02	wbijgt6.rtp.raleigh.ibm.com	

For each of the nodes defined in the above table, perform the steps outlined in the 16.8.1, “Creating a Custom WebSphere Process Server node” on page 533, through 16.8.3, “Augmenting with WebSphere Business Services Fabric” on page 536.

16.8.1 Creating a Custom WebSphere Process Server node

Perform the following steps to create a custom profile for WebSphere Process Server.

> **Note:** Be sure that the deployment manager has been started first.

1. Open a window on your AIX system.
2. Type `cd /monwas/ProcServer/bin/ProfileManagement`.
3. Type `./pmt.sh` to start the Profile Management Tool.
4. On the Welcome panel, click **Next**.
5. On the Environment Selection panel, click **WebSphere Process Server**, then click **Next**.
6. On the Profile Type Selection panel, click **Custom profile**, then click **Next**.
7. On the Profile Creation Options panel, click **Advanced Profile Creation** and then click **Next**.

8. On the Profile Name and Location panel:
   a. Set Profile Name to `Node01`.
   b. Set Profile Directory to `/monwas/ProcServer/profiles/Node01`.
   c. Click **Next**.
9. On the Node and Host Names panel:
   a. Set Node name to `Node01`.
   b. The host name should be correct by default.
   c. Click **Next**.
10. On the Federation panel:
   a. Set Deployment manager host name or IP Address to `wbijgt3.rtp.raleigh.ibm.com`.
   b. The Deployment manager SOAP port number default value is correct.
   c. In our sample topology, we are using a federated repository (which is using LDAP).
   a. Set User Name to `wps`.
   b. Set Password to `passw0rd`.
   c. Click **Next**.
11. On the Port Value Assignment panel, leave the values unchanged and click **Next**.
12. In our sample topology, we are using DB2 as a database vendor, so on the Database Configuration panel:
   a. Click **DB2 Universal** on Choose the database product used on the deployment manager.
   b. The Location (directory) of JDBC driver classpath files should be set to `/monwas/ProcServer/universalDriver_wbi/lib`.
   c. Click **Next**.
   d. A Profile Creation Summary panel will appear. Click **Create**.
13. The profile creation will take some time to complete. When the Profile Creation Complete panel comes up, uncheck Launch the First steps console and click **Finish**.

16.8.2 Augmenting with WebSphere Business Monitor

Once the task to create a custom profile for WebSphere Process Server has been completed, it is now time to augment that profile with WebSphere Business Monitor:

1. Type `./pmt.sh` to start the Profile Management Tool.
2. Click **Augment**.
3. Click **Next** on the Welcome Panel.
4. On the Select the profile to augment panel, under Profiles, click the profile that you wish to augment (for example, Node01). Click **Next**.
5. On the Augment Selection panel, under Augments, click **WebSphere Business Monitor custom profile** and then click **Next**.
6. On the following panel, click **Advanced profile augmentation**, then click **Next**.
7. On the Federation panel:
   a. Set Deployment manager host name or IP Address to `wbijgt3.rtp.raleigh.ibm.com`.
   b. The Deployment manager SOAP port number default value is correct.
   c. In our sample topology we use a federated repository (that is using LDAP).
   a. Set User Name to `wps`.
   b. Set Password to `passw0rd`.
   c. Click **Next**.
8. In our sample topology, we use DB2 as a database vendor, so on the Database Configuration panel:
   a. Click **DB2 Universal** on Choose the database product used on the deployment manager.
   b. The Location (directory) of JDBC driver classpath files should be set to `/monwas/ProcServer/universalDriver.wbm/lib`. Click **Next**.
9. On the Profile Augmentation Summary panel, click **Augment**.
10. The profile augmentation step will take some time to complete.
11. A panel should appear indicating that `The Profile Management Tool augmented the profile successfully`.
12. Uncheck Launch the WebSphere Business Monitor first steps and click **Finish**.

16.8.3 Augmenting with WebSphere Business Services Fabric

The next task is to augment the profile with WebSphere Business Services Fabric:

1. Type `./pmt.sh` to start the Profile Management Tool (PMT). (Tip: If you are following these directions, the PMT is most likely still up, and if this is the case, start at the next step.
2. Click **Augment**.
3. Click **Next** on the Welcome Panel.
4. On the Select the profile to augment panel, under Profiles, click the profile that you wish to augment (for example, Node01). Click **Next**.
5. On the Augment Selection panel, under Augments, click **WebSphere Business Services Fabric** and then click **Next**.
6. On the next panel, click **Advanced profile augmentation**, then click **Next**.
7. On the Federation panel:
   a. Set Deployment manager host name or IP Address to `wbijgt3.rtp.raleigh.ibm.com`.
   b. The Deployment manager SOAP port number default value is correct.
   c. In our sample topology, we use a federated repository (which is using LDAP).
   a. Set User Name to `wps`.
   b. Set Password to `passw0rd`.
   c. Click **Next**.
8. In our sample topology, we use DB2 as a database vendor, so on the Database Configuration panel:
   a. Click **DB2 Universal** on Choose the database product used on the deployment manager.
   b. The Location (directory) of JDBC driver classpath files should be set to `/monwas/ProcServer/universalDriver_wbi/lib`.
   c. Click **Next**.
9. On the Profile Augmentation Summary panel, click **Augment**.
10. The profile augmentation step does not take a lot of time to complete.
11. A panel should come up indicating that `The Profile Management Tool augmented the profile successfully`.

16.8.4 Verifying that all nodes appear via WebSphere Admin Console

It is a good idea to verify that all the nodes were created:

1. Sign in to the Integrated Solutions console.
2. Click **System Administration** → **Nodes** (Figure 16-15).
3. A few things to validate include:
   a. Verify that all nodes (two in our topology) are listed.
   b. Verify that each node's status indicates that it is synchronized.
   c. Verify that the version values look like Figure 16-15.

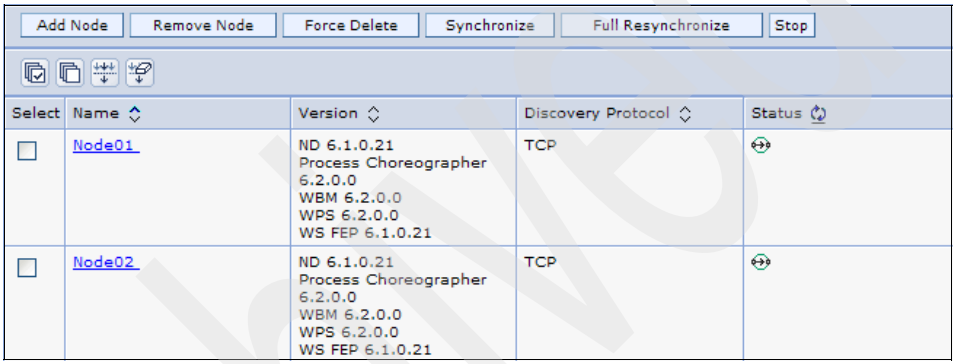


Figure 16-15 WebSphere Admin Console: List of nodes

16.9 Creating clusters

In the WebSphere Dynamic Process Edition Four Cluster topology, there are four clusters that must be created. This section describes the steps necessary to create these four clusters.

16.9.1 Creating the Application Cluster

The first cluster to create is the application cluster. Table 16-3 provides the name of the cluster and what members make up the cluster.

Table 16-3 Application cluster details

Cluster name	Member name	Node	Host name
WPSTest.AppTarget	Member1	Node01	wbijgt1.rtp.raleigh.ibm.com
WPSTestApp.Target	Member1	Node02	wbijgt6.rtp.raleigh.ibm.com

1. Go to the deployment manager console and sign in.
2. Click **Servers** → **Clusters**.
3. Click **New**.
4. The name of this cluster will be WPSTest.AppTarget (Figure 16-16).

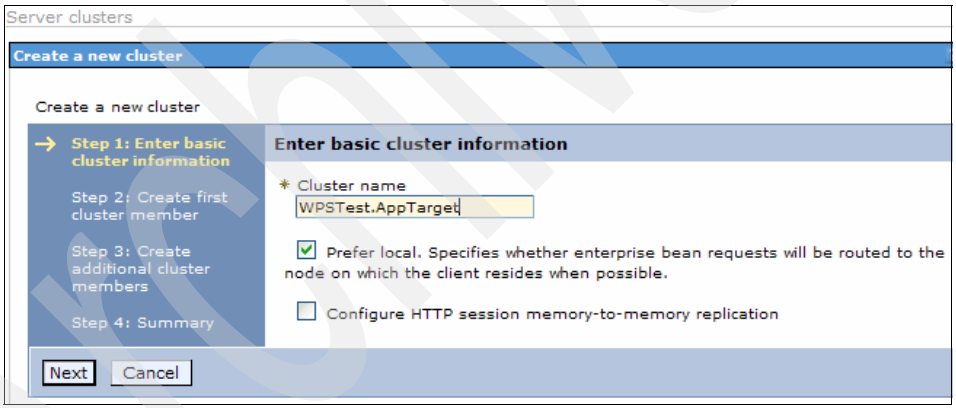


Figure 16-16 Creating the Application Cluster

5. Click **Next** (Figure 16-17).
6. In our sample topology, we will add two members to this cluster, Node1 and Node2:
   a. Set Member Name to `Member1`.
   b. Click **Create the member using an application server template** and select **defaultProcessServer_defaultWBM**. Click **Next**.

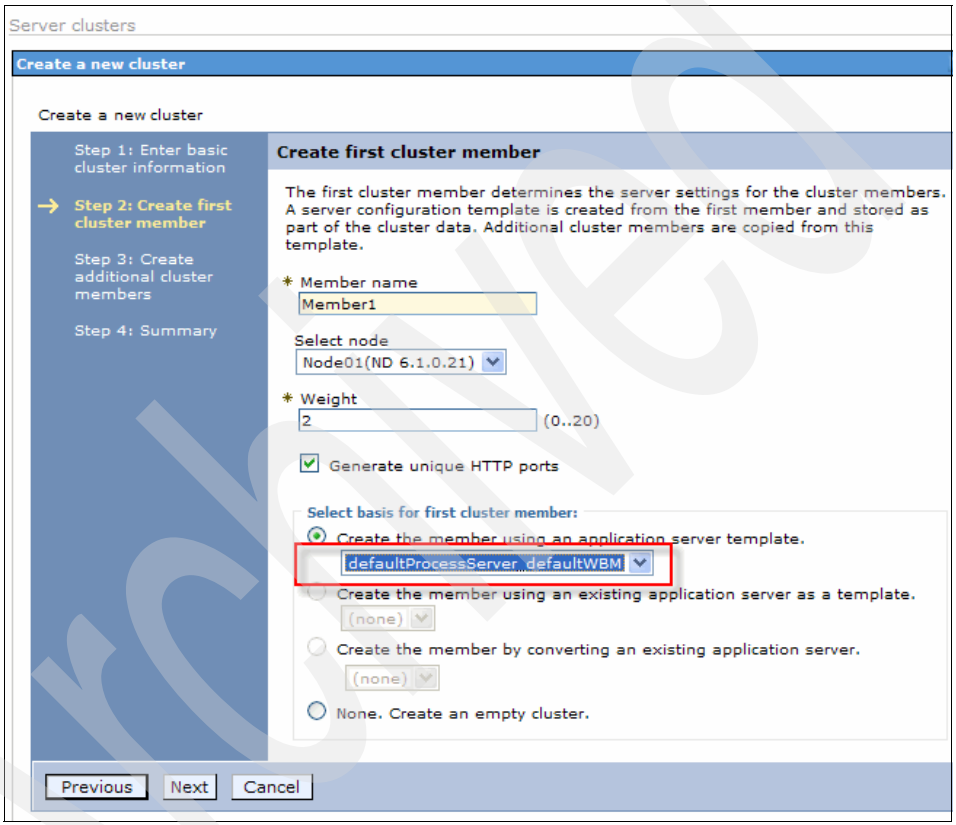


Figure 16-17 Create the member using an application server template

7. To add the additional cluster member:
   a. Set Member Name to `Member1`.
   b. Set Select node to `Node02`.
   c. Click **Add Member**.

8. Click **Next**. The Summary panel should be reviewed to ensure that Node1 and Node2 were the nodes to be added and that they are using the proper clone template (Figure 16-18).

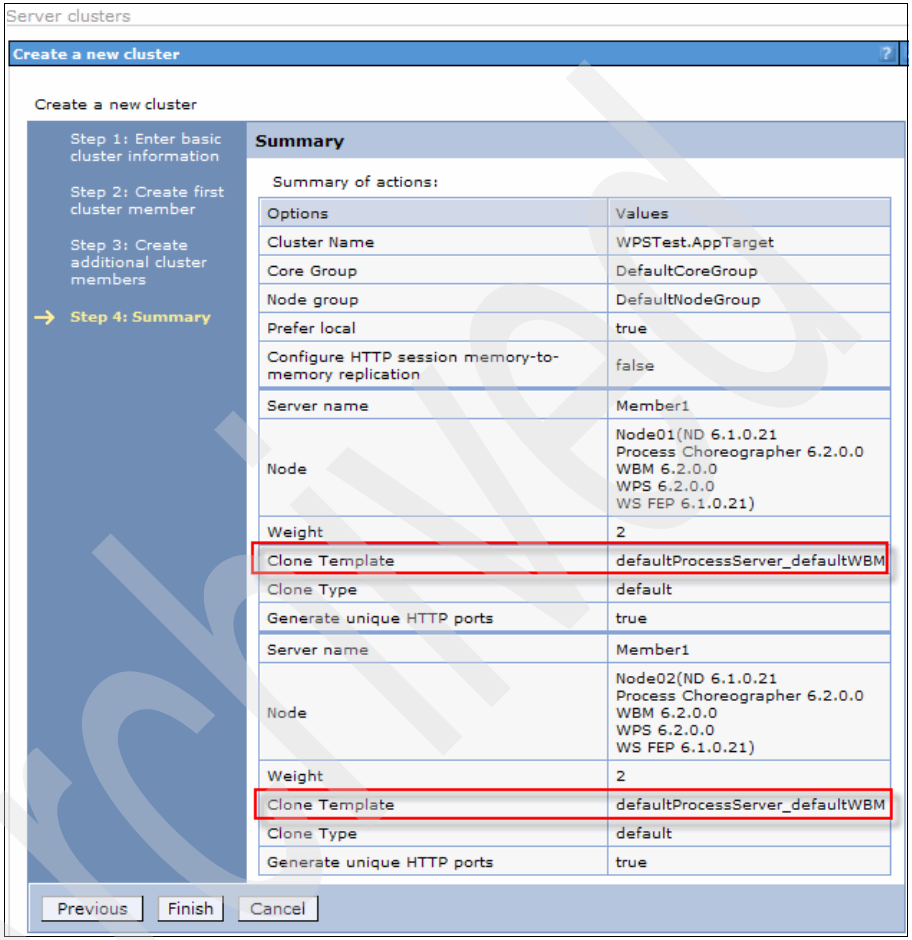


Figure 16-18 Summary panel for the WPSTest.AppTarget cluster creation

9. Click **Finish** to create this cluster.
10. Click **Save** to save your configuration changes.

16.9.2 Creating the Support Cluster

The next cluster to create is our Support Cluster. Follow the same steps that appear in 16.9.1, “Creating the Application Cluster” on page 538, but use WPSTest.Support for the name of the cluster. For illustration purposes, we have

included the summary panel for this cluster creation as a guide. You should compare your values before completing the task of creating this cluster. (See Table 16-4 and Figure 16-19.)

Table 16-4 Names

Cluster name	Member name	Node	Host name
WPSTest.Support	Member2	Node01	wbijgt1.rtp.raleigh.ibm.com
WPSTestSupport	Member2	Node02	wbijgt6.rtp.raleigh.ibm.com

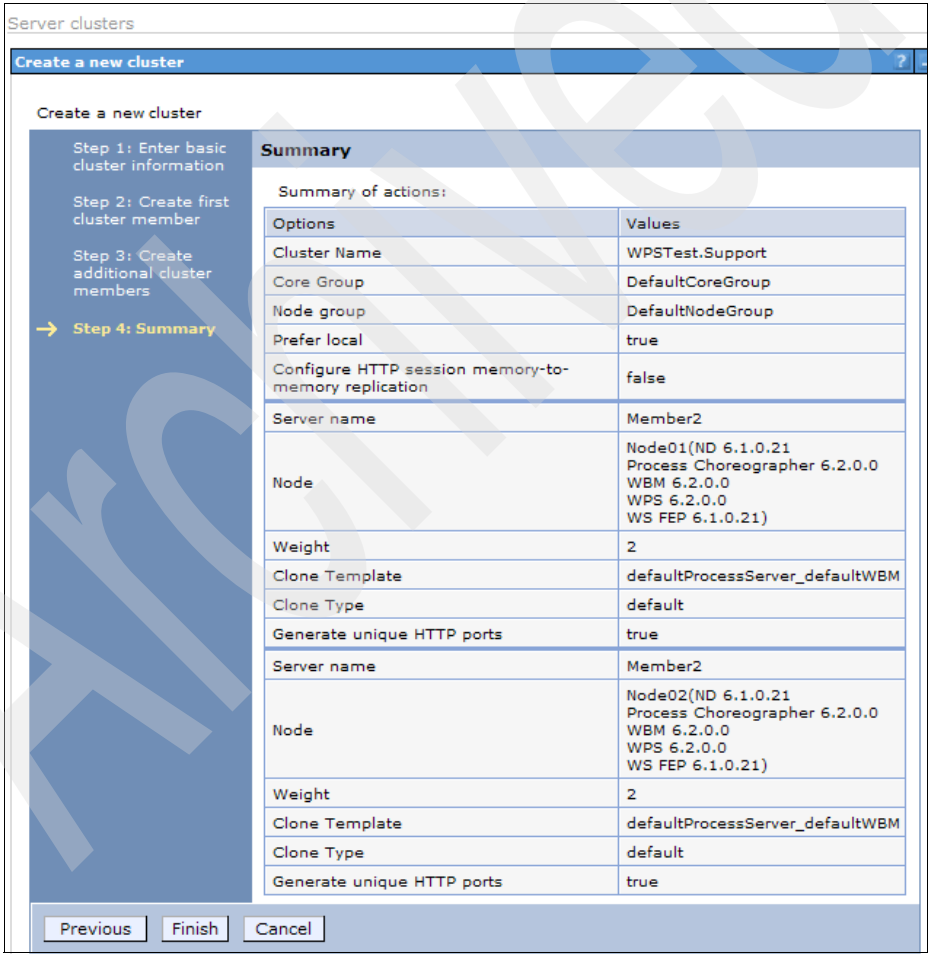


Figure 16-19 Summary panel for WPSTest.Support cluster creation

16.9.3 Creating the Messaging Cluster

The next cluster to create is our Messaging Cluster. Follow the same steps that appear in 16.9.1, “Creating the Application Cluster” on page 538, but use WPSTest.Messaging for the name of the cluster. For illustration purposes, we have included the summary panel for this cluster creation as a guide. (See Table 16-5 and Figure 16-20 on page 543.)

Table 16-5 Support cluster details

Cluster name	Member name	Node	Host name
WPSTest.Messaging	Member3	Node01	wbijgt1.rtp.raleigh.ibm.com
WPSTest.Messaging	Member3	Node02	wbijgt6.rtp.raleigh.ibm.com

Server clusters

Create a new cluster

Create a new cluster

Step 1: Enter basic cluster information

Step 2: Create first cluster member

Step 3: Create additional cluster members

→ **Step 4: Summary**

Summary

Summary of actions:

Options	Values
Cluster Name	WPSTest.Messaging
Core Group	DefaultCoreGroup
Node group	DefaultNodeGroup
Prefer local	true
Configure HTTP session memory-to-memory replication	false
Server name	Member3
Node	Node01(ND 6.1.0.21 Process Choreographer 6.2.0.0 WBM 6.2.0.0 WPS 6.2.0.0 WS FEP 6.1.0.21)
Weight	2
Clone Template	defaultProcessServer_defaultWBM
Clone Type	default
Generate unique HTTP ports	true
Server name	Member3
Node	Node02(ND 6.1.0.21 Process Choreographer 6.2.0.0 WBM 6.2.0.0 WPS 6.2.0.0 WS FEP 6.1.0.21)
Weight	2
Clone Template	defaultProcessServer_defaultWBM
Clone Type	default
Generate unique HTTP ports	true

Previous Finish Cancel

Figure 16-20 Summary panel for WPSTest.Messaging cluster creation

16.9.4 Creating the Web Cluster

The next cluster to create is our Web Cluster. Follow the same steps that appear in 16.9.1, “Creating the Application Cluster” on page 538, but use WPSTest.Web for the name of the cluster. For illustration purposes, we included the summary panel for this cluster creation as a guide (see Table 16-6 and Table 16-6).

Table 16-6 Support cluster details

Cluster name	Member name	Node	Host name
WPSTest.Web	Member4	Node01	wbijgt1.rtp.raleigh.ibm.com
WPSTest.Web	Member4	Node02	wbijgt6.rtp.raleigh.ibm.com

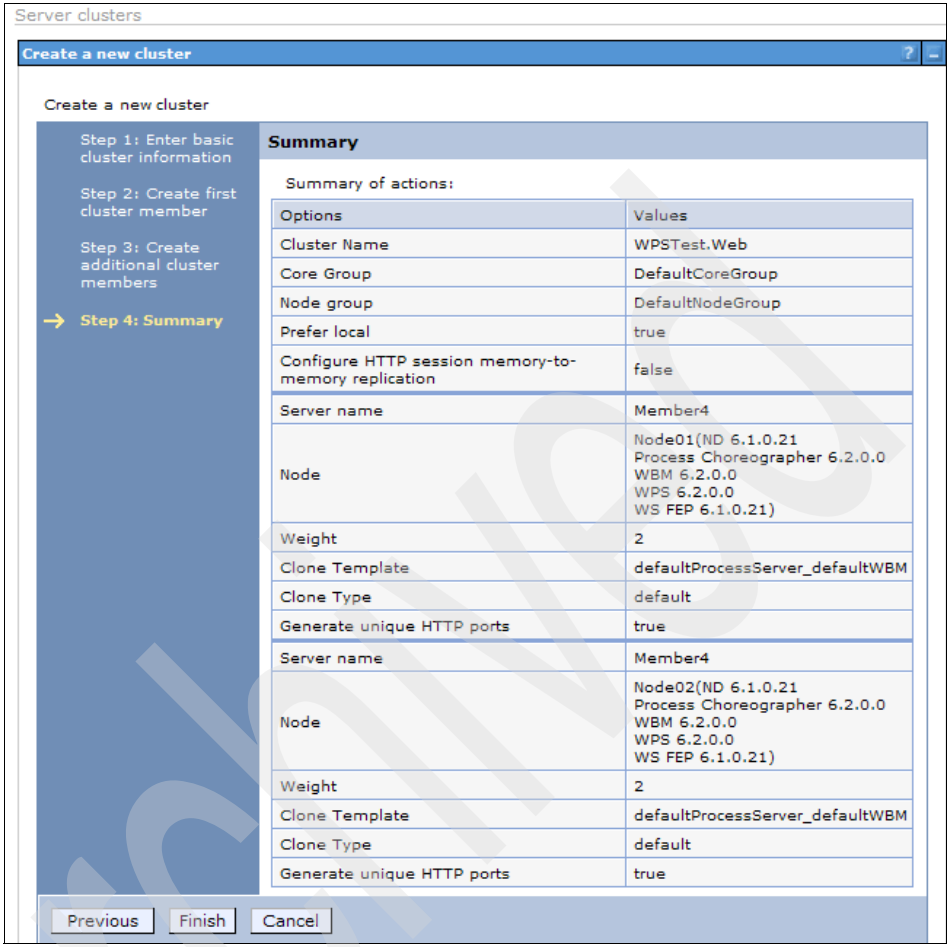


Figure 16-21 Summary panel for WPSTest.Web cluster creation

16.9.5 Verifying clusters

This is a good point at which to ensure that there are no errors anywhere. To do this:

1. Stop all nodes:
   a. Go to the Integrated Solutions Console.
   b. Click **System Administration** → **Node agents**.
   c. Select all nodes and click **Stop**.
2. Stop the deployment manager.

3. Start the deployment manager.
4. Once the deployment manager has come up, start all node agents.
5. Sign in to the Integrated Solutions console.
6. Click **Servers** → **Clusters**.
7. Check all check boxes to select all of our clusters and click **Start**.
8. The amount of time that it will take for the clusters to start will vary.
9. At some point, all the cluster should have started. (See Figure 16-22.)

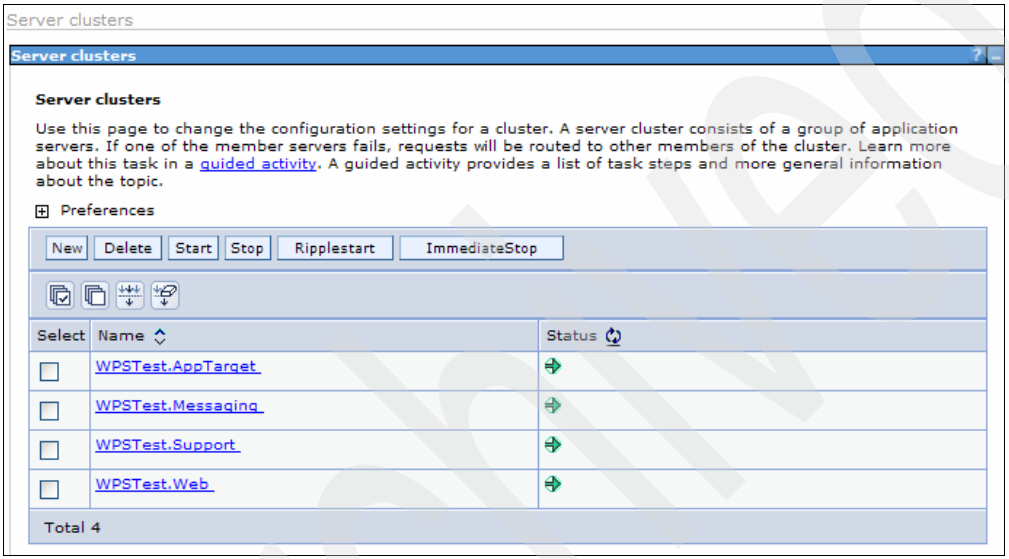


Figure 16-22 Cluster status for our topology

16.10 Configuring and deploying CEI

The Common Event Infrastructure Server will be deployed to the WPSTest.Support cluster. This section details how to complete this task.

1. Sign in to the Integrated Solutions Console.
2. Click **Servers** → **Clusters**.
3. Click **WPSTest.Support**.

4. Expand **Common Event Infrastructure** (Figure 16-23).

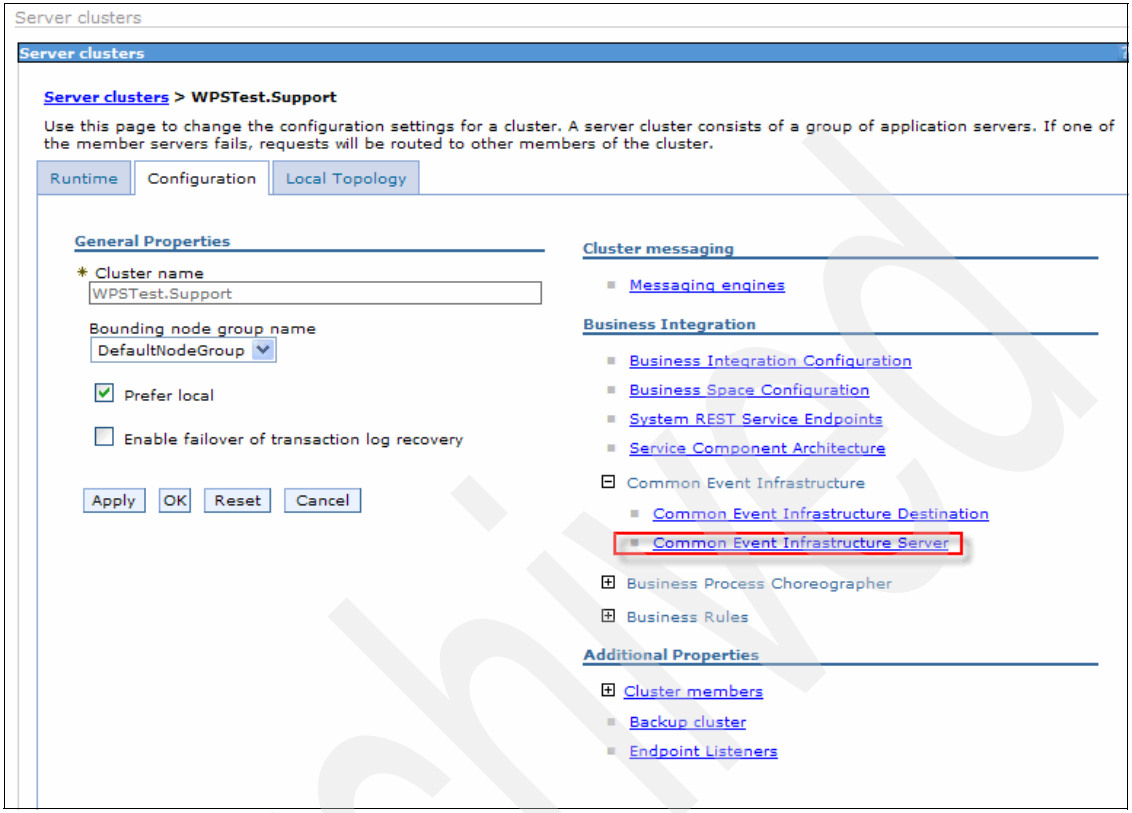


Figure 16-23 Common Event Infrastructure Server configuration

5. Click **Common Event Infrastructure Server**.
6. Check the check box **Enable the Event Infrastructure Server**. Notice that items on this panel are no longer grayed out.

7. Notice the section of the panel Common Event Infrastructure Bus Members Location. Click **Remote** since our bus member will be on the messaging cluster. Click **New** (Figure 16-24).

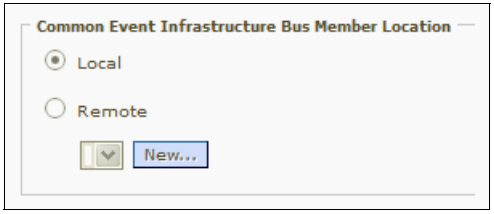


Figure 16-24 Common Event Infrastructure Bus Member location

8. On the Browse Deployment targets panel that pops up, click the radio button for the deployment target that says `Cluster=WPSTest.Messaging` (Figure 16-25).

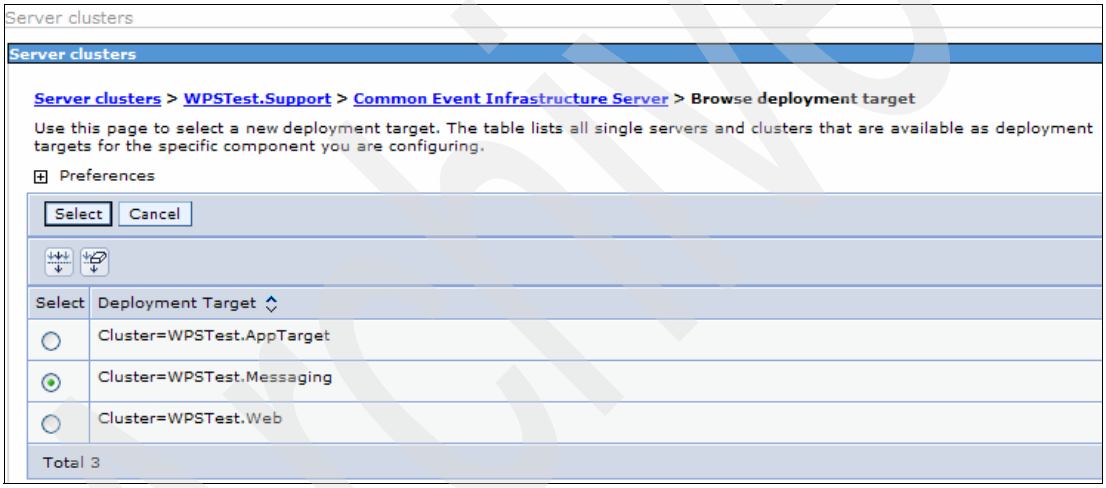


Figure 16-25 Deployment Target selection

9. Click the **Select** radio button. Notice that we are now indicating that our bus member for CEI will be on our messaging cluster (Figure 16-26).

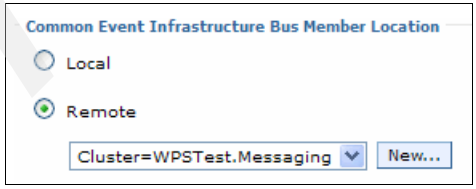


Figure 16-26 Remote selection

10. There are two databases that must be configured for CEI. The first one is the Common Event infrastructure database. Set the database instance to be EVENT. The schema should be left blank. The database has already been created and the database tables have not been created, so leave the **Create Tables** check box checked (Figure 16-27).

11. The second database is the Common Event Infrastructure bus. Set the database instance to MEDB and set the schema to CEIME. Uncheck the check box for **Create Tables** (Figure 16-27).

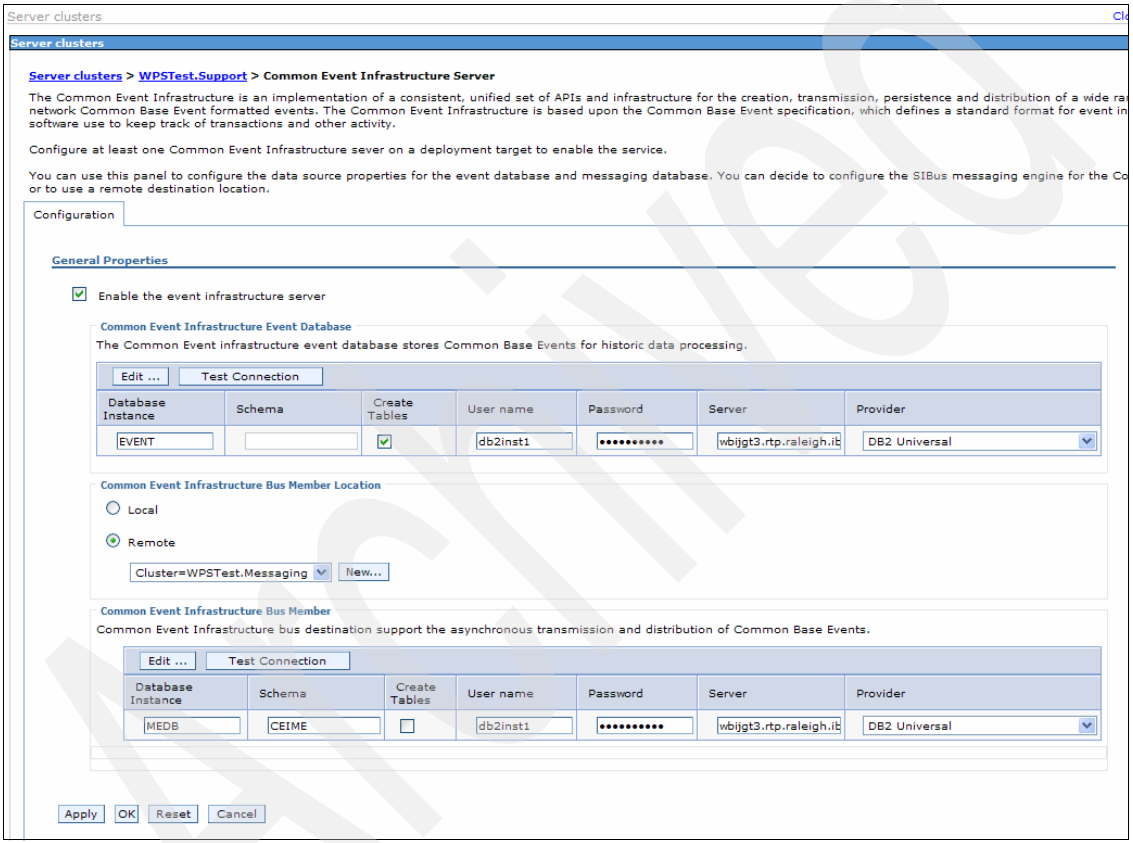


Figure 16-27 CEI Configuration panel

Click **OK**.

12. When the configuration completes, a message indicating that the common event infrastructure server is configured successfully. Click **Save** to save the configuration changes.

13. When the save completes, you will now be on the Server Clusters panel.

14. Click **WPSTest.Support** cluster.

15. Expand **Common Event Infrastructure** → **Common Event Infrastructure Server**. The message shown in Figure 16-28 will appear.

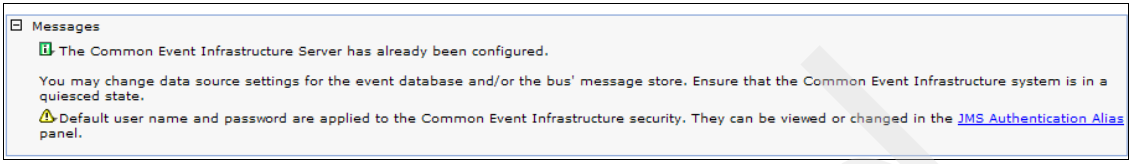


Figure 16-28 CEI is configured message

16. Under Additional properties, click **JMS Authentication Alias**. The default values appear on the panel that appears (Figure 16-29).

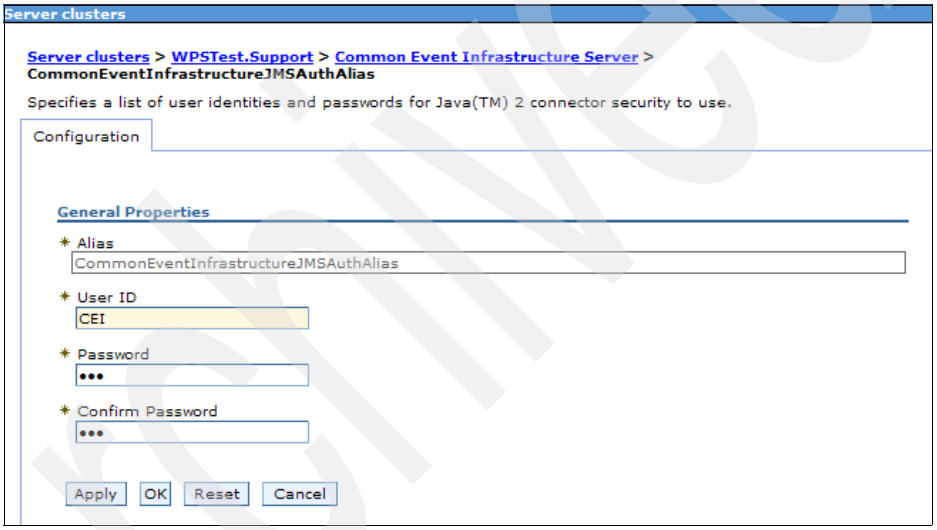


Figure 16-29 Default JMS Authentication alias values

17. Change the default values. In our case, since we are using LDAP security, set the user ID to `wps`, set the password to `passw0rd`, and set Confirm Password to `passw0rd` (Figure 16-30).

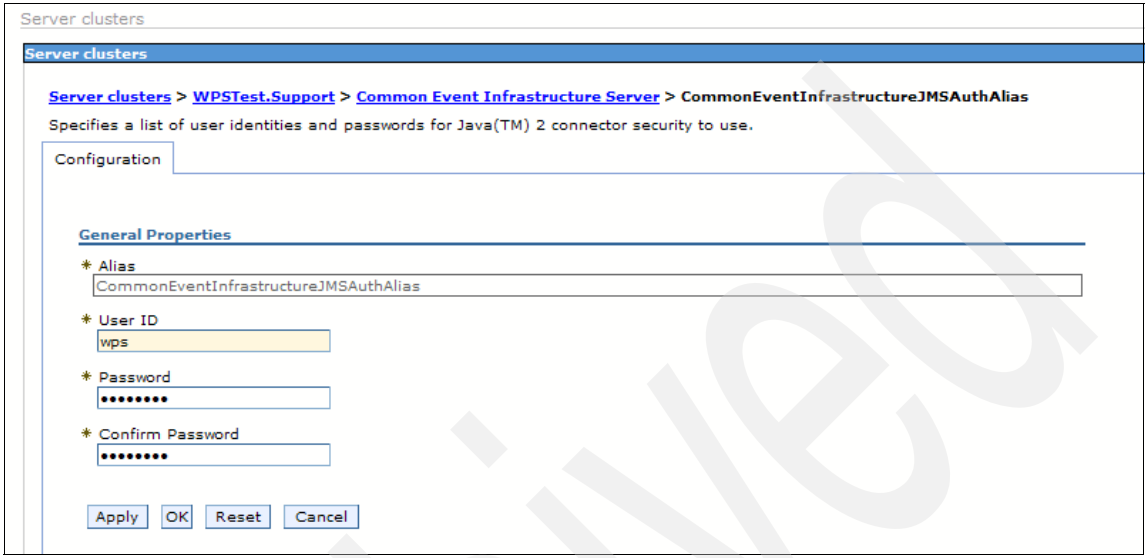


Figure 16-30 Changing default user ID and password for JMS authentication alias

18. Click **OK** and then be sure to save your configuration changes.
19. At this point, there is an extra step that must be performed at the Integrated Solutions Console:
    a. Click **Resources** → **JDBC** → **Data Sources**.
    b. Click the **CEI ME data source** link.
    c. Notice that Component-managed authentication alias and Authentication alias for XA Recovery both indicate None.
        i. For both of the above, change the value to use **CEIME_WPSTest.Messaging_Auth_Alias**. Click **OK**.
        ii. Click **Save** to save your configuration changes.
20. On the Data sources panel, click the check box **CEI ME data source** and click **Test Connection**. Verify that the connection test was successful.
21. From the Integration System Console, click **Servers** → **Clusters** and click the **WPSTest.Messaging** cluster.

22. Under Cluster Messaging, click **Messaging Engines** and verify that there is a messaging engine for CEI, as shown in Figure 16-31.

Figure 16-31 CEI Messaging Engine

16.11 Configuring Service Component Architecture

The Service Component Architecture will be deployed to the WPSTest.Web cluster. This section describes how to complete this task.

1. Sign in to the Integrated Solutions Console.
2. Click **Servers** → **Clusters** and click **WPSTest.AppTarget**.

3. Click **Service Component Architecture** (Figure 16-32).

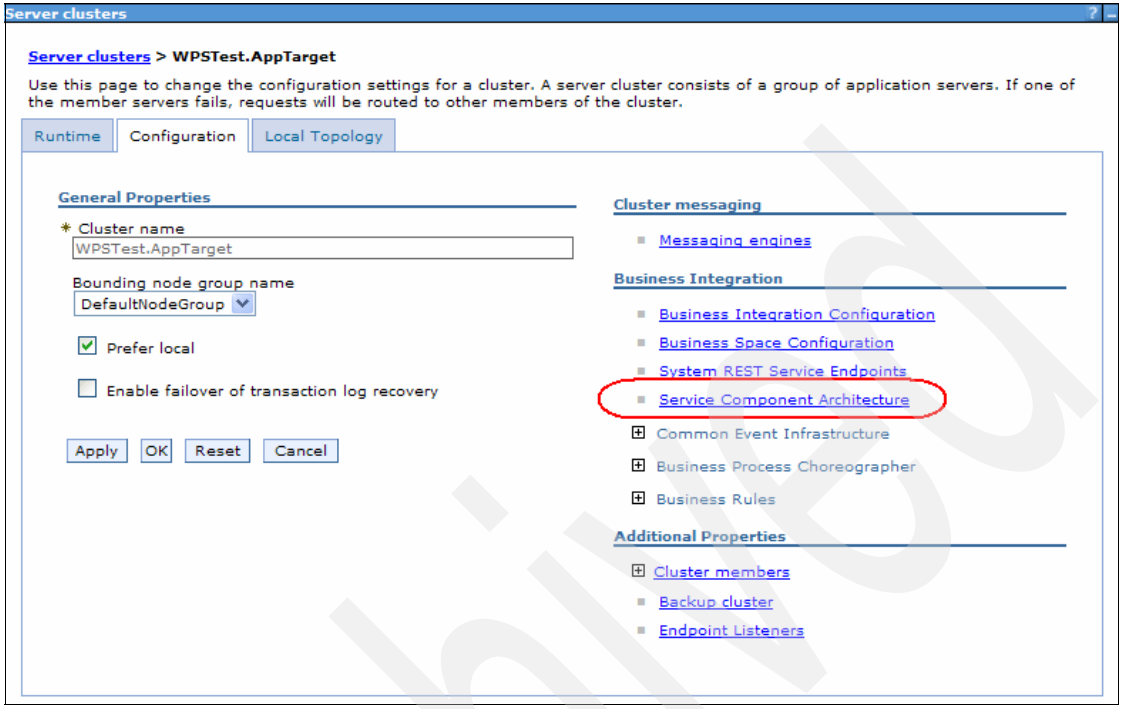


Figure 16-32 Service Component Architecture configuration

4. On the Service Component Architecture configuration panel, check the check box **Support the Service Component Architecture components**.

5. Configure the SCA Bus member to be on the Messaging cluster. Click **Remote**, then click **New** (Figure 16-33).

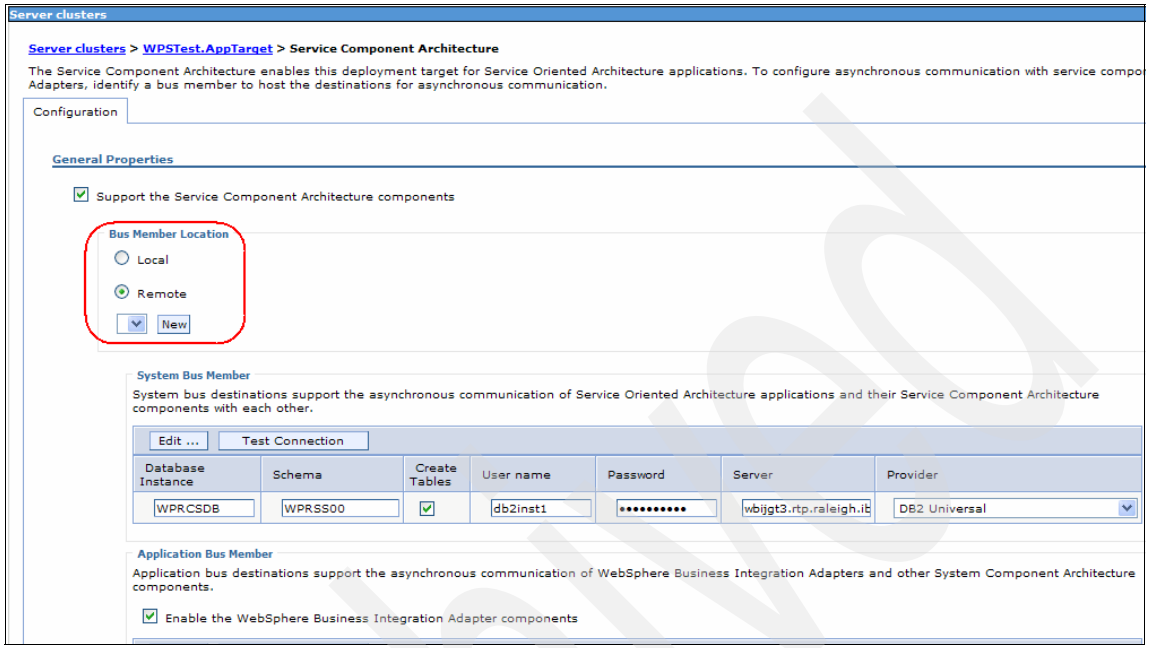


Figure 16-33 Service Component Architecture Configuration panel

6. On the Browse deployment target panel, click the radio button for **WPSTest.Messaging** and then click **Select** (Figure 16-34).

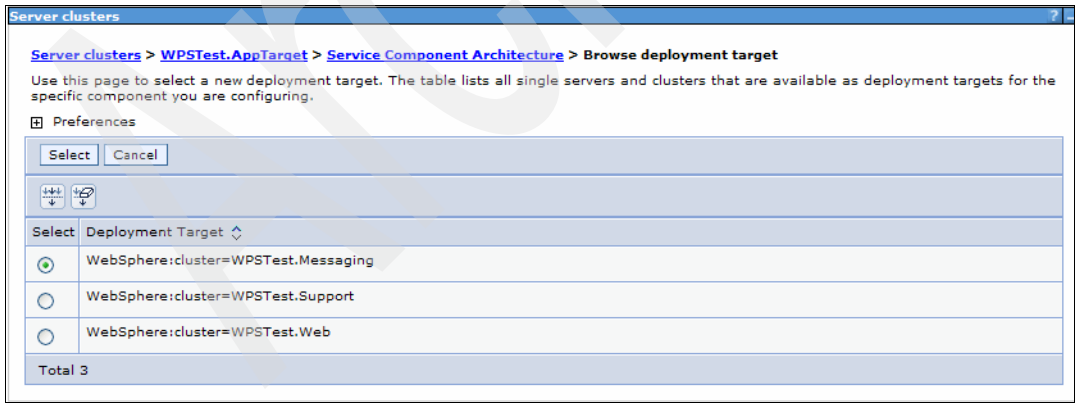


Figure 16-34 Browse Deployment panel

7. In our topology, the SCA configuration will use the previously created tables in the MEDB database. Set the database instance to `MEDB` and the schema to `SCASYS` for the System Bus member, and set the database instance to `MEDB` and the schema to `SCAAPP` for the Application Bus member. Be sure to also uncheck Create Tables for both bus members. Click **OK** (Figure 16-35).

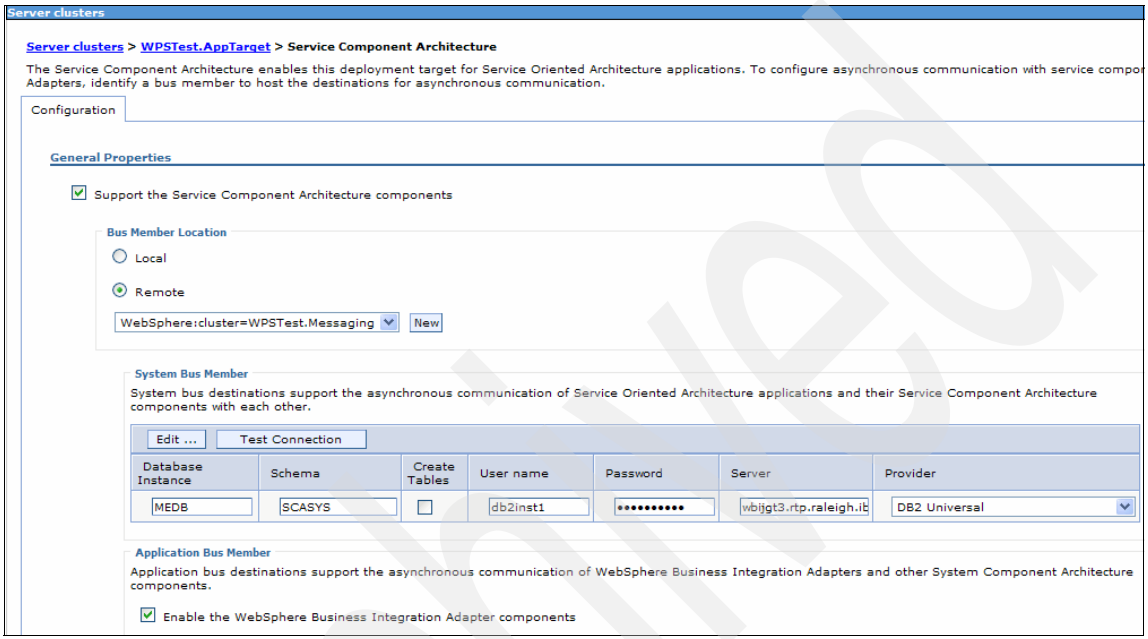


Figure 16-35 Service Component Architecture configured values

8. Once the configuration has completed, you will be asked to save your changes. Click **Save** to save the configuration changes.
9. Once the changes have been saved, you are returned back to the Server Clusters panel. To verify that the Service Component Architecture component has been deployed to the WPSTest.AppTarget cluster, click **WPSTest.AppTarget** → **Service Component Architecture**. A message should appear at the top of the panel indicating that the Service Component Architecture has been configured.

There is one remaining step to perform. If you were to click the **Test Connection** button on either of our datasources, they would fail. The message would indicate something like a `null userid is not supported`. What this essentially means is that the JDBC resource that was created does not have any authentication alias defined. To resolve this:

1. Sign in to the Integrated System Console.
2. Click **Resources** → **JDBC** → **Data Sources**.
3. Click the **Name** link and **SCA System Bus ME data source** link.
4. Notice that the Component-managed authentication alias and the Authentication alias for XA recovery both say None. Change these values to point to `SCASYSME00 Auth Alias`.
5. Click **OK**.
6. Click **SCA Application Bus ME data source**.
7. Notice that the Component-managed authentication alias and the Authentication alias for XA recovery both say None. Change these values to point to SCAAPPME00 Auth Alias. Click **OK**.
8. Save all configuration changes made.
9. Go back to **Resources** → **JDBC** → **Data Sources** and select both **SCA System Bus ME data source** and **SCA Application Bus ME data source.** Click **Test Connection** to verify that the connection is valid.
10. From the Integration System Console, click **Servers** → **Clusters** and click **WPSTest.Messaging** cluster.

11. Click **Messaging Engines** and verify that there is a messaging engine for the SCA System Bus and the SCA Application Bus, as shown in Figure 16-36.

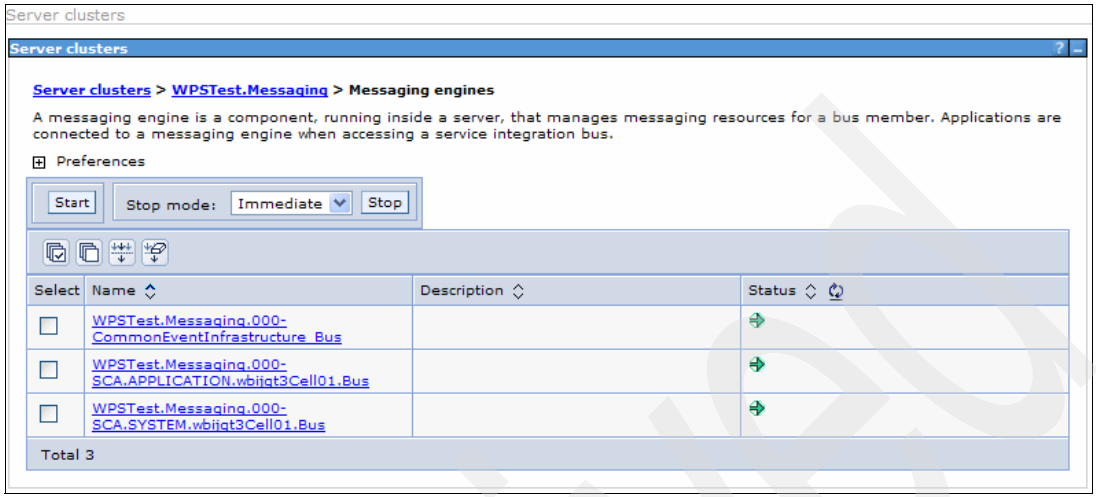


Figure 16-36 SCA Messaging Engines

16.12 Configuring Common Event Destination for Web Cluster

This task is necessary so that the Web Cluster is aware of where the CEI server is. Follow these steps to complete this task.

1. Sign in to the Integrated Solutions Console.
2. Click **Servers** → **Clusters**, then click **WPSTest.Web**.
3. Expand **Common Event Infrastructure**.

4. Click **Common Event Infrastructure Destination** (Figure 16-37).

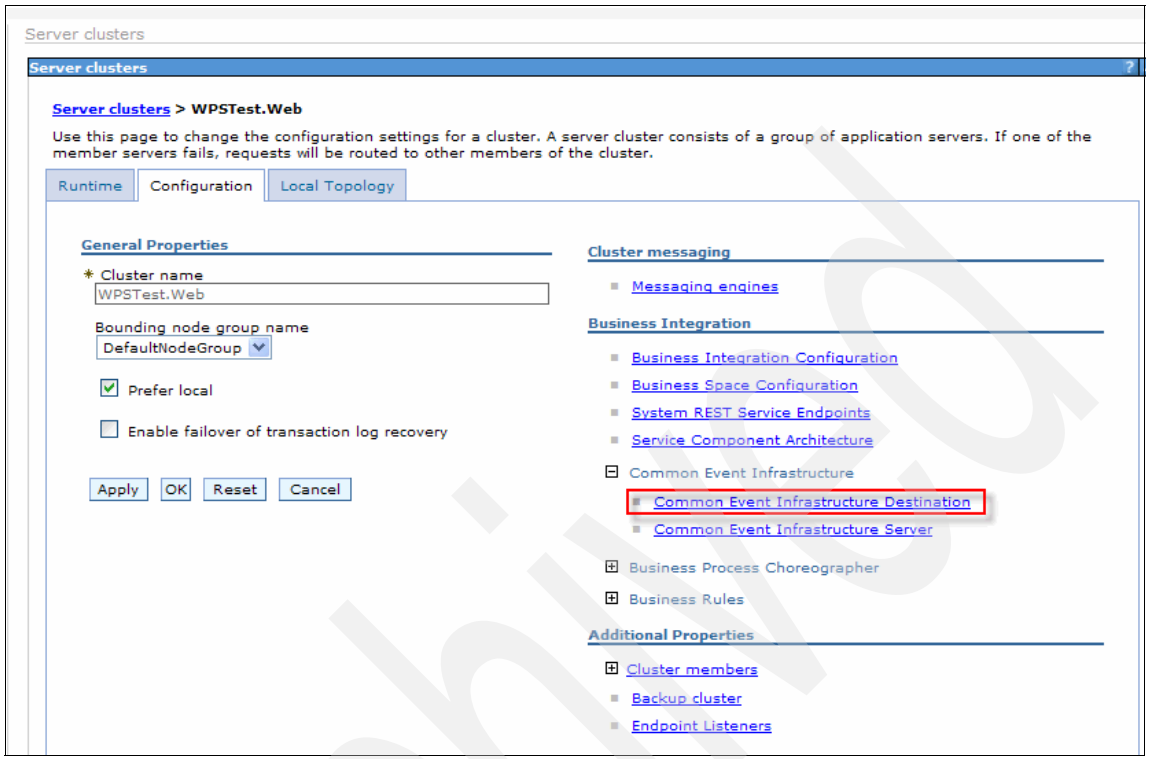


Figure 16-37 Common Event Infrastructure Destination

5. On the Common Event Infrastructure Destination panel, check the check box **Enable service at server startup**. (Note that this may already be checked and that is fine.)

6. Click the first radio button and select **cell/clusters/WPSTest.Support/com/ibm/events/configuration/emitter/Default** (Figure 16-38).

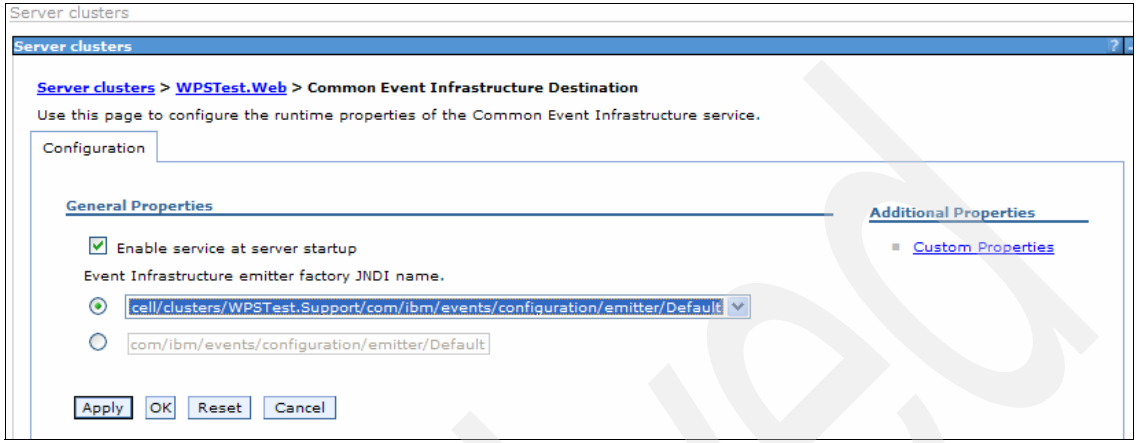


Figure 16-38 Common Event Infrastructure Destination Configuration

Click **OK**.

7. Click **Save** to save your configuration changes.

Configuring Business Process Choreographer

Complete the task of configuring the BPC Explorer by following these steps:

1. Go to the Integrated Solutions Console and log in.
2. In the navigation pane, click **Servers** → **Application Servers** → **Clusters**.
3. Click **WPSTest.AppTarget** cluster.

4. Under Business Integration expand **Business Process Choreographer** (Figure 16-39).

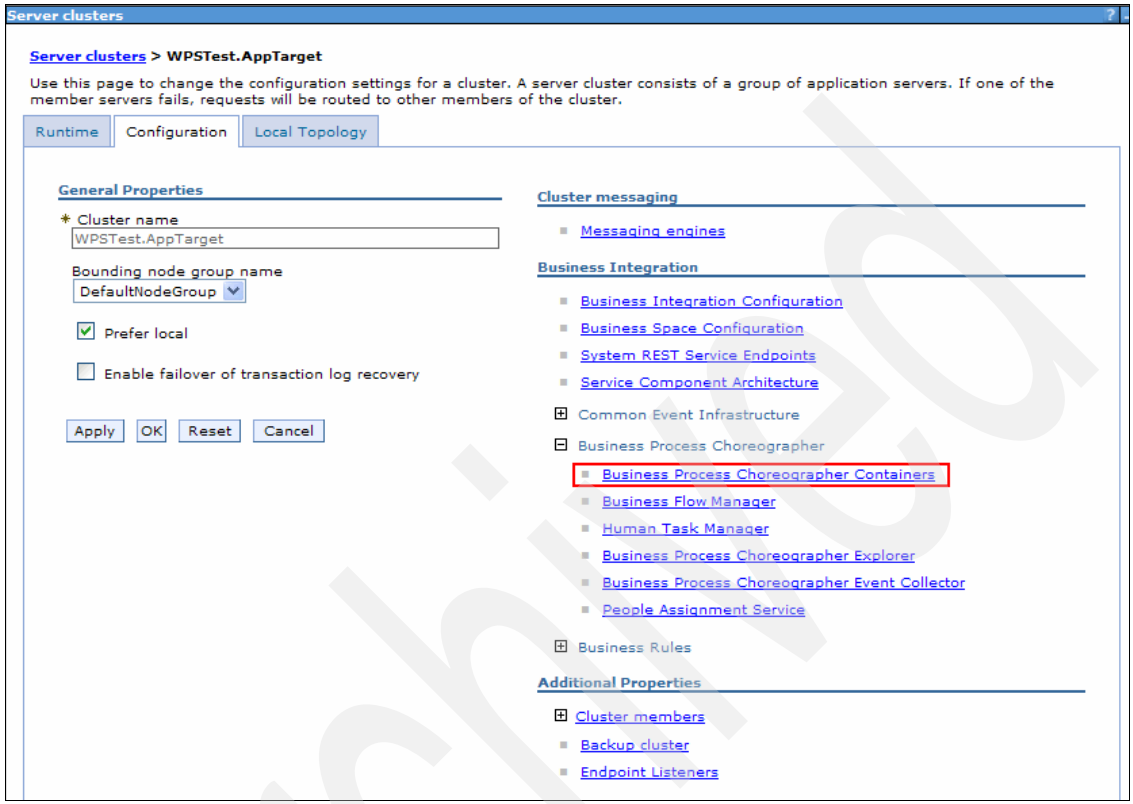


Figure 16-39 Business Process Choreographer

5. Click **Business Process Choreographer Containers**.
6. Notice at the top of the panel a message indicating that the BPC container has not yet been installed.

7. In our topology, we will use the database BPEDB. Set Database instance to BPEDB and set Schema Name to BPC. Since this database has been previously created manually, be sure to uncheck Create Tables (Figure 16-40).

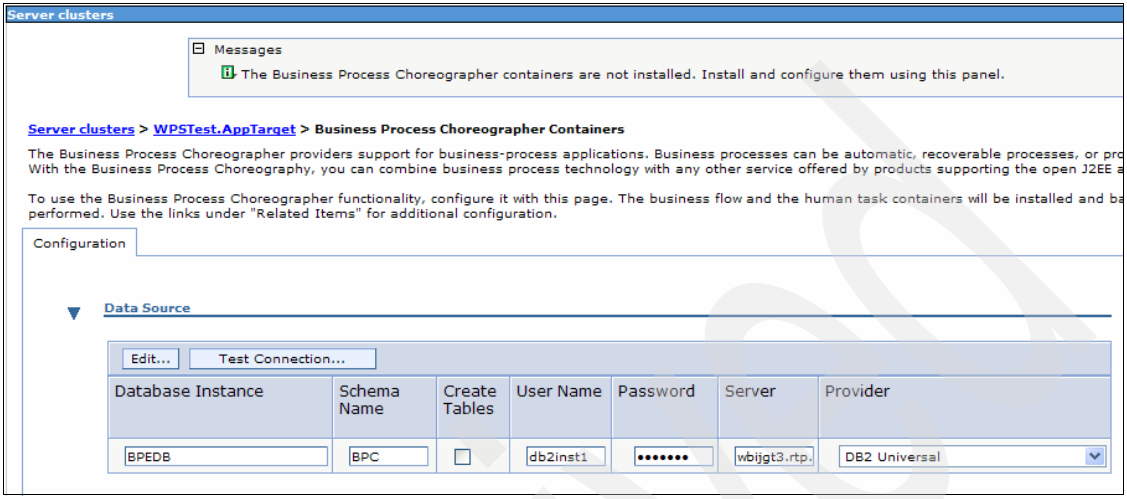


Figure 16-40 BPC Database configuration

8. Under Human Task Manager Mail Session, leave the **Enable e-mail service** box checkmarked (Figure 16-41).

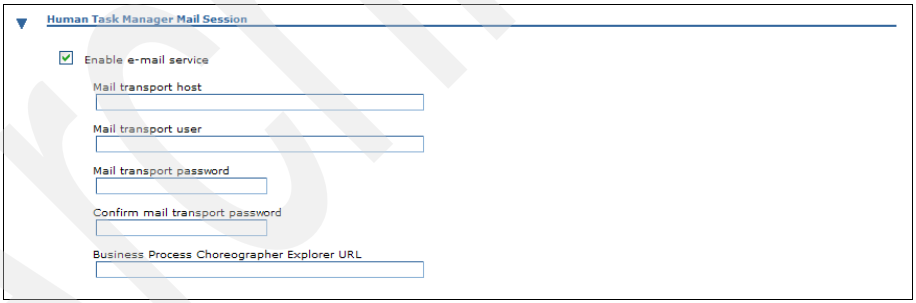


Figure 16-41 BPC Human Task Manager Mail Session configuration

9. Under the Security section, ensure that both the users and passwords are set to websphere (Figure 16-42).

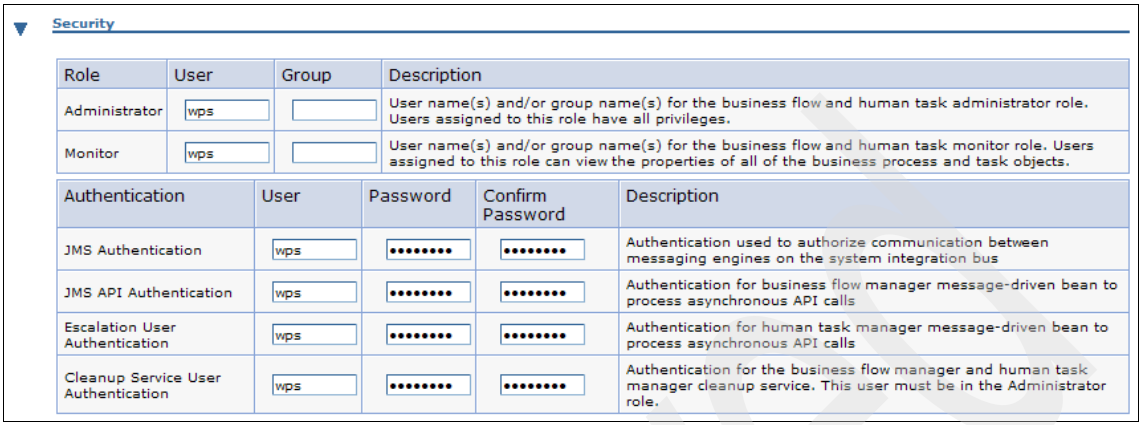


Figure 16-42 Security configuration

10. Under State Observers be sure to check both check boxes, as shown in Figure 16-43.

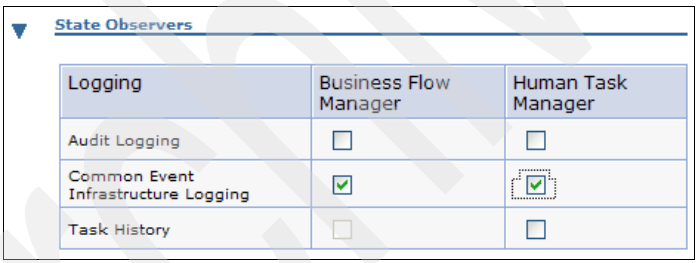


Figure 16-43 State Observers

11. The settings under SCA Bindings can remain as is.

12. Under Buses, un-check the check the box **Use the default configuration**. Instead of the default configuration, we click the **Remote** radio button. Then click **New** (Figure 16-44).

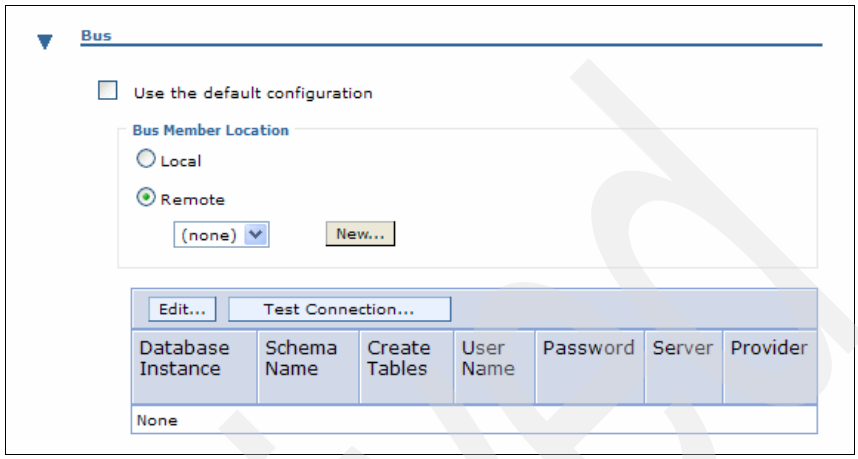


Figure 16-44 BPC Bus Member location

13. On the Browse deployment target panel, click the **WPSTest.Messaging** radio button selection (Figure 16-45).
14. Click **Select**.

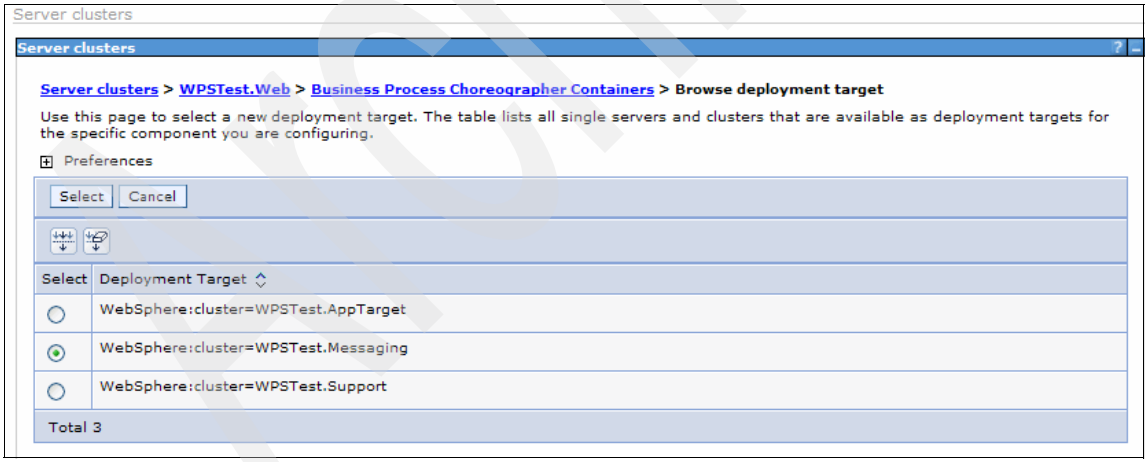


Figure 16-45 Server clusters

15. The configuration should indicate that the bus member is remote and that it is on the Messaging cluster. In our topology, set Database instance to use `MEDB`

and set Schema Name to be `BPCME`. Since this database was created previously, be sure to uncheck Create Tables (Figure 16-46).

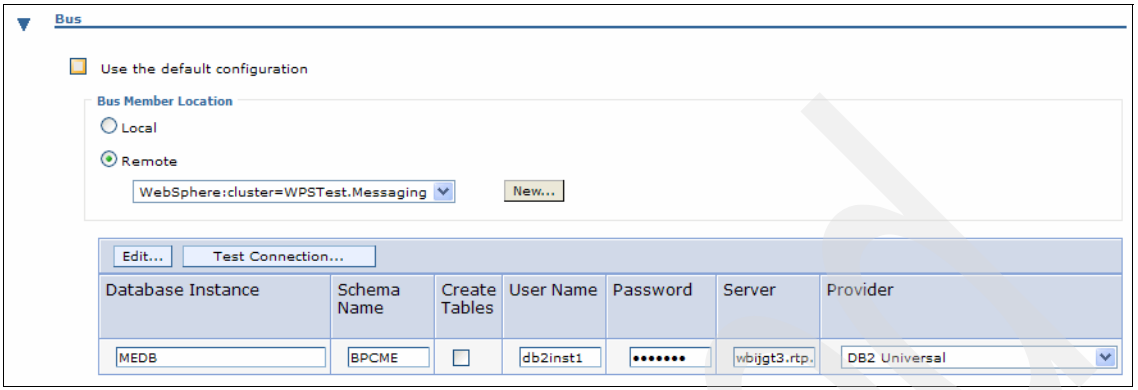


Figure 16-46 Setting database information

16. Click **OK**. Numerous messages will appear on the console. Eventually, a message like will appear `The configuration has ended`. Click **Save Changes** to save the configuration changes (Figure 16-47).

```
2009-05-04 15:07:58   ADMA5005I: The application HTM_PredefinedTasks_V620_WPSTest.AppTarget is configured in the WebSphere Application Server repository.
2009-05-04 15:07:58   ADMA5001I: The application binaries are saved
                      in /monwas/ProcServer/profiles/Dmgr01/wstemp/117946/workspace/cells/wbijgt3Cell01/applications/HTM_PredefinedTasks_V620_WPSTest.AppTarget.ear/
2009-05-04 15:07:58   ADMA5005I: The application HTM_PredefinedTasks_V620_WPSTest.AppTarget is configured in the WebSphere Application Server repository.
2009-05-04 15:07:59   SECJ0400I: Successfuly updated the application HTM_PredefinedTasks_V620_WPSTest.AppTarget with the appContextIDForSecurity information.
2009-05-04 15:07:59   ADMA5005I: The application HTM_PredefinedTasks_V620_WPSTest.AppTarget is configured in the WebSphere Application Server repository.
2009-05-04 15:07:59   CWSCA3013I: Resources for the SCA application "HTM_PredefinedTasks_V620_WPSTest.AppTarget" are being configured.
2009-05-04 15:07:59   CWSCA3023I: The EAR file "HTM_PredefinedTasks_V620_WPSTest.AppTarget.ear" is being loaded for the SCA module.
2009-05-04 15:07:59   CWSCA3017I: Installation task "Resource Task for SCA Messaging Binding and EIS Binding" is running.
2009-05-04 15:07:59   CWSCA3017I: Installation task "Resource Task for SCA Messaging Binding and JMS Binding" is running.
2009-05-04 15:07:59   CWSCA3017I: Installation task "SIBus Destination Resource Task for SCA Asynchronous Invocations" is running.
2009-05-04 15:07:59   CWSCA3017I: Installation task "EJB NamespaceBinding Resource Task for SCAImportBinding" is running.
2009-05-04 15:07:59   CWSCA3017I: Installation task "SIBus Destination Resource Task for SCA SOAP/JMS Invocations" is running.
2009-05-04 15:07:59   CWSCA3017I: Installation task "PolicySet Task for JaxWsImportBinding and JaxWsExportBinding" is running.
2009-05-04 15:07:59   CWSCA3014I: Resources for the SCA application "HTM_PredefinedTasks_V620_WPSTest.AppTarget" have been configured successfully.
2009-05-04 15:07:59   ADMA5011I: The cleanup of the temp directory for application HTM_PredefinedTasks_V620_WPSTest.AppTarget is complete.
2009-05-04 15:07:59   ADMA5013I: Application HTM_PredefinedTasks_V620_WPSTest.AppTarget installed successfully.
2009-05-04 15:08:02   ADMA5016I: Installation of HTM_PredefinedTaskMsg_V620_WPSTest.AppTarget started.
2009-05-04 15:08:02   ADMA5058I: Application and module versions are validated with versions of deployment targets.
2009-05-04 15:08:02   ADMA5005I: The application HTM_PredefinedTaskMsg_V620_WPSTest.AppTarget is configured in the WebSphere Application Server repository.
2009-05-04 15:08:02   CWTCO0008I: The human tasks in application HTM_PredefinedTaskMsg_V620_WPSTest.AppTarget have been successfully configured in the WebSphere
                      configuration repository.
2009-05-04 15:08:02   ADMA5053I: The library references for the installed optional package are created.
2009-05-04 15:08:02   ADMA5005I: The application HTM_PredefinedTaskMsg_V620_WPSTest.AppTarget is configured in the WebSphere Application Server repository.
2009-05-04 15:08:02   ADMA5001I: The application binaries are saved
                      in /monwas/ProcServer/profiles/Dmgr01/wstemp/117946/workspace/cells/wbijgt3Cell01/applications/HTM_PredefinedTaskMsg_V620_WPSTest.AppTarget.e
2009-05-04 15:08:02   ADMA5005I: The application HTM_PredefinedTaskMsg_V620_WPSTest.AppTarget is configured in the WebSphere Application Server repository.
2009-05-04 15:08:02   SECJ0400I: Successfuly updated the application HTM_PredefinedTaskMsg_V620_WPSTest.AppTarget with the appContextIDForSecurity information.
2009-05-04 15:08:02   ADMA5005I: The application HTM_PredefinedTaskMsg_V620_WPSTest.AppTarget is configured in the WebSphere Application Server repository.
2009-05-04 15:08:02   CWSCA3013I: Resources for the SCA application "HTM_PredefinedTaskMsg_V620_WPSTest.AppTarget" are being configured.
2009-05-04 15:08:02   CWSCA3023I: The EAR file "HTM_PredefinedTaskMsg_V620_WPSTest.AppTarget.ear" is being loaded for the SCA module.
2009-05-04 15:08:03   CWSCA3017I: Installation task "Resource Task for SCA Messaging Binding and EIS Binding" is running.
2009-05-04 15:08:03   CWSCA3017I: Installation task "Resource Task for SCA Messaging Binding and JMS Binding" is running.
2009-05-04 15:08:03   CWSCA3017I: Installation task "SIBus Destination Resource Task for SCA Asynchronous Invocations" is running.
2009-05-04 15:08:03   CWSCA3017I: Installation task "EJB NamespaceBinding Resource Task for SCAImportBinding" is running.
2009-05-04 15:08:03   CWSCA3017I: Installation task "SIBus Destination Resource Task for SCA SOAP/JMS Invocations" is running.
2009-05-04 15:08:03   CWSCA3017I: Installation task "PolicySet Task for JaxWsImportBinding and JaxWsExportBinding" is running.
2009-05-04 15:08:03   CWSCA3014I: Resources for the SCA application "HTM_PredefinedTaskMsg_V620_WPSTest.AppTarget" have been configured successfully.
2009-05-04 15:08:03   ADMA5011I: The cleanup of the temp directory for application HTM_PredefinedTaskMsg_V620_WPSTest.AppTarget is complete.
2009-05-04 15:08:03   ADMA5013I: Application HTM_PredefinedTaskMsg_V620_WPSTest.AppTarget installed successfully.
2009-05-04 15:08:04   Component 'Business Process Choreographer' configured successfully on deployment target 'WPSTest.AppTarget'. You need to restart the server/cluster
                      for any changes to take effect.
2009-05-04 15:08:04   The configuration has ended.

Show Changes   Save Changes   Discard Changes   Close
```

Figure 16-47 Save configuration values

17. At this point, it is necessary to correct the data source connected to the Business Process Choreographer ME data source.

    a. Sign in to the Integrated System Console.

    b. Click **Resources** → **JDBC** → **Data Sources**.

    c. Click the **Business Process Choreographer ME data source** link.

d. Notice that the Component-managed authentication alias and the Authentication alias for XA recovery both say None. Change these values to point to BPCME_00_Auth Alias.

e. Click **OK** and then click **Save** to save all configuration changes made.

f. Go back to **Resources** → **JDBC** → **Data Sources** and select the check box **Business Process Choreographer ME data source**. Click **Test Connection** to verify that the connection is valid.

18. Go to click **Servers** → **Application Servers** → **Clusters**.

19. Select the **WPSTest.AppTarget** cluster.

20. Click **Stop** to stop this cluster. The amount of time that it will take to stop varies. Once it has stopped, select the *WPSTest.Web* cluster and click **Start**.

21. To further verify that there are no problems, go back to the Integrated System Console.

22. Click **Servers** → **Clusters**, then click **WPSTest.AppTarget**.

23. Under **Business Integration** expand **Business Process Choreographer**.

24. Click **Business Process Choreographer Containers**. At the top of the page there should be two messages indicating that both the Business Flow Manager and the Human Task Manager are currently installed.

25. Click **Resources** → **JDBC** → **Data sources**. For both data sources, click the **Test Connection** button to ensure that connections can be successfully made.

26. Stop the WPSTest.Messaging and WPSTest.AppTarget clusters and then restart them.

27. Click **Servers** → **Clusters**, then click **WPSTest.Messaging**.

28. Click **Messaging Engines** and verify that there is a messaging engine for the BPC, as shown in Figure 16-48.

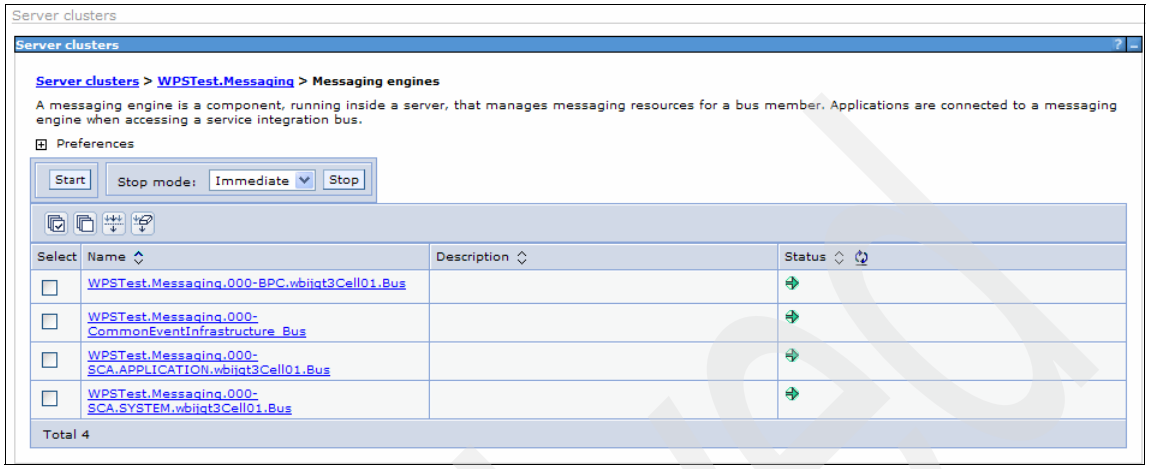


Figure 16-48 BPC Messaging Engine

16.13 Configuring BPC Explorer

Complete the task of configuring the BPC Explorer by following these steps:

1. Go to the Integrated Solutions Console and log in.
2. In the navigation pane, click **Servers** → **Application Servers** → **Clusters**.
3. Click the **WPSTest.Web** cluster.

4. Under Business Integration click **Business Process Choreographer** (Figure 16-49).

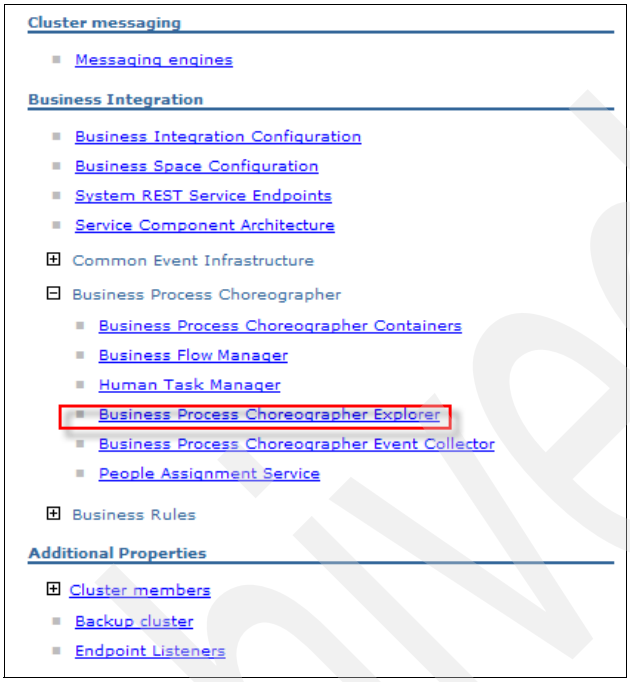


Figure 16-49 Business Process Choreographer Explorer configuration

5. Click **Business Process Choreographer Explorer**.
6. On the BPC Explorer configuration panel, click **Add**.

7. Change the context root to `/bpc` (Figure 16-50).

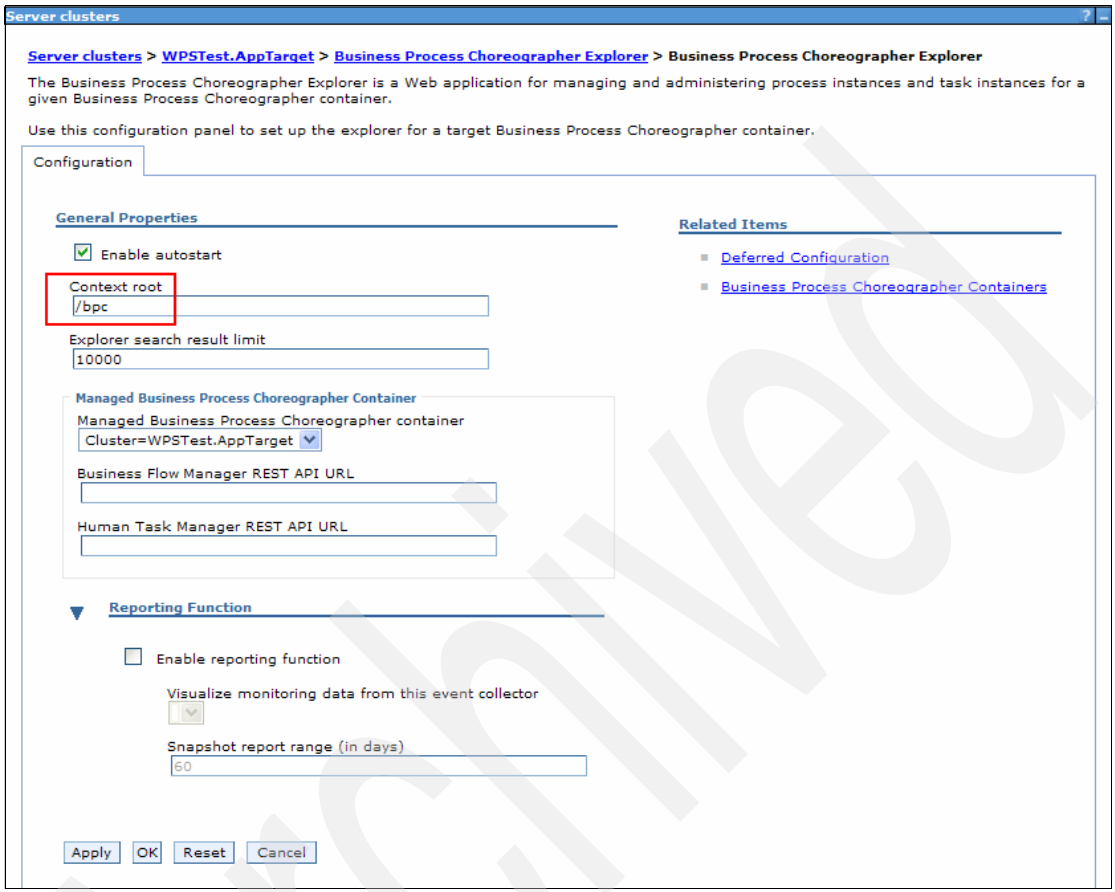


Figure 16-50 BPC Explorer configuration panel

Click **OK**.

8. Click **Save Changes** to save your configuration changes.

9. On the following panel, click **Start** to start the BPC Explorer application and make sure that the Application Status shows that it has started (Figure 16-51).

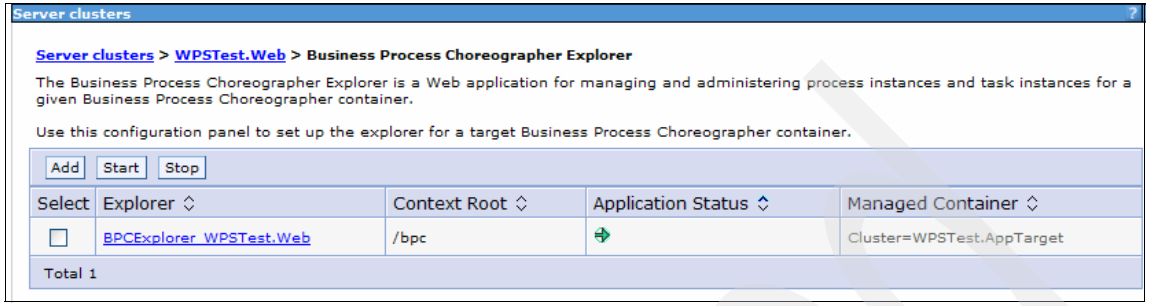


Figure 16-51 Business Process Choreographer Explorer Application Status

16.14 Configuring Business Rules Manager

Complete the task of configuring the Business Rules Manager by following these steps:

1. Go to the Integrated Solutions Console and log in.
2. In the navigation pane, click **Servers** → **Application Servers** → **Clusters**.
3. Click the **WPSTest.Web** cluster.

4. Under Business Integration expand **Business Rules** (Figure 16-52).

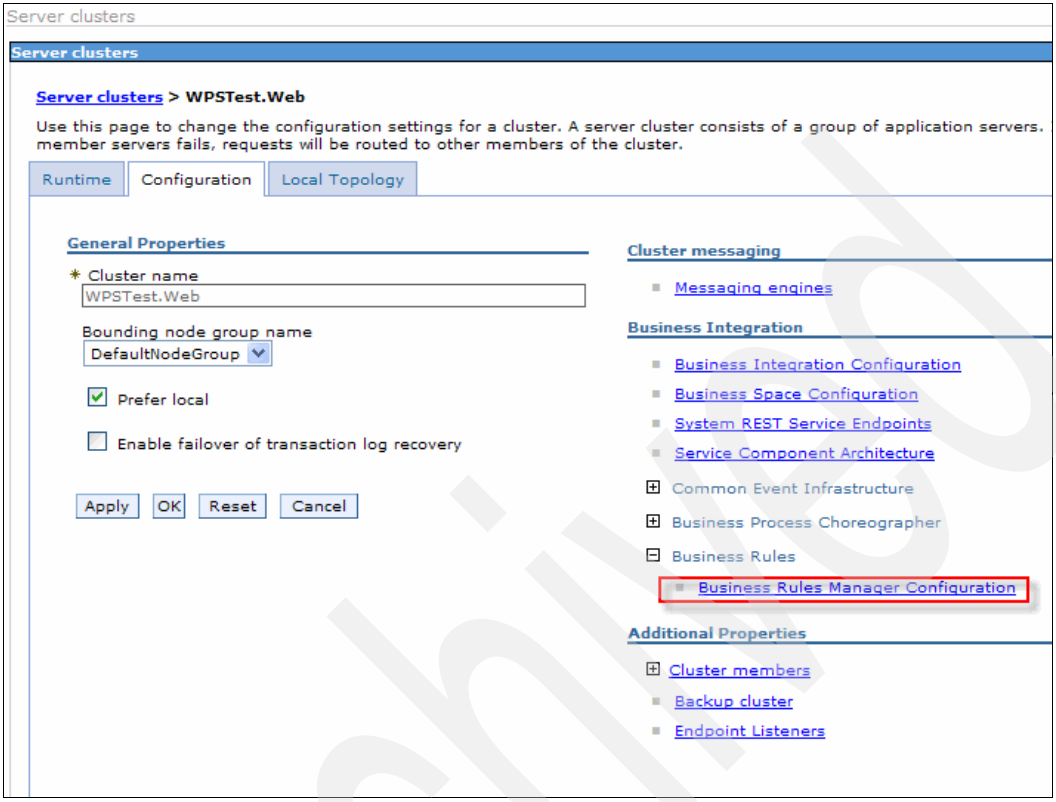


Figure 16-52 Business Rules Manager Configuration

5. Click **Business Rules Manager Configuration**.

6. Check the **Install business rules manager** check box (Figure 16-53).

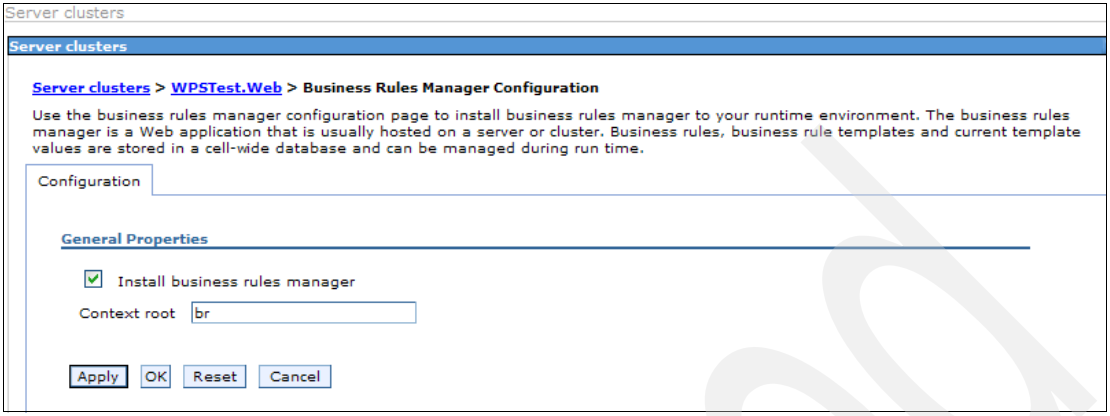


Figure 16-53 Business Rules Manager configuration panel

7. Leave the context root set to br. Click **OK.**
8. Click **Save** to save your configuration changes.

16.15 Configuring Business Space

Business Space must be configured manually. To do this:

1. Go to the Integrated Solutions Console and log in.
2. In the navigation pane, click **Servers** → **Application Servers** → **Clusters**.
3. Click the **WPSTest.Web** cluster.

4. Under Business Integration, click **Business Space Configuration** (Figure 16-54).

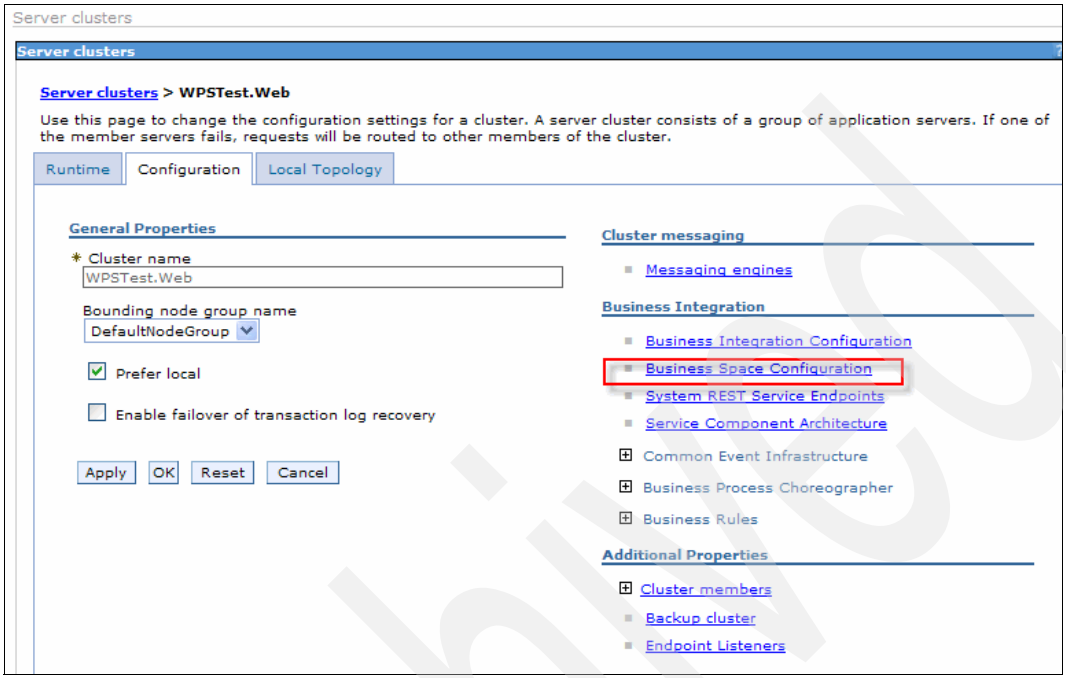


Figure 16-54 Business Space configuration

5. On the Business Space Configuration panel (Figure 16-55):
   a. Check the **Install Business Space service** check box.

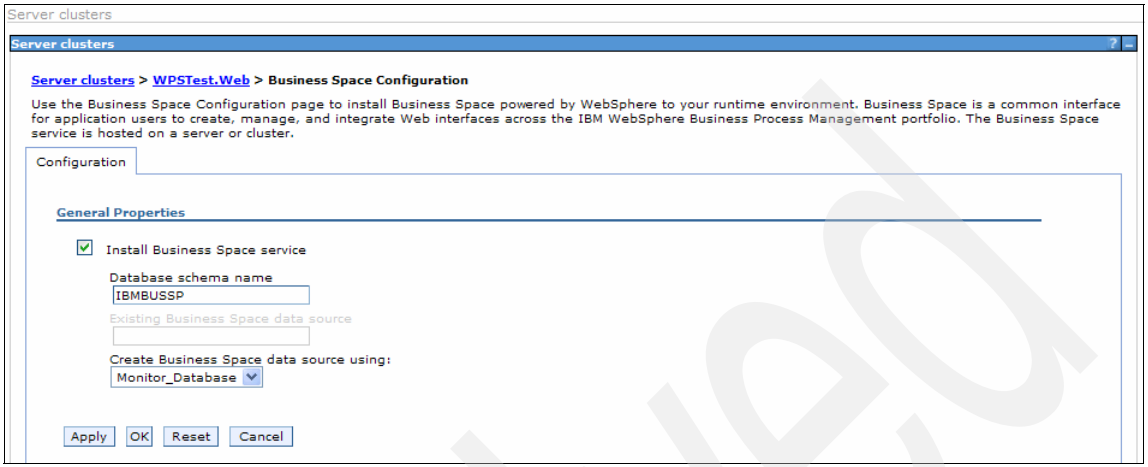


Figure 16-55 Business Space Configuration

   b. In the Database schema name field, type `IBMBUSSP`.
   c. In the Create Business Space data source using drop-down box, select **Monitor_database**. Click **OK**.
   d. Save all changes (Figure 16-56). It is safe to ignore the warning about creating the database tables for Business Space since this step was previously performed.

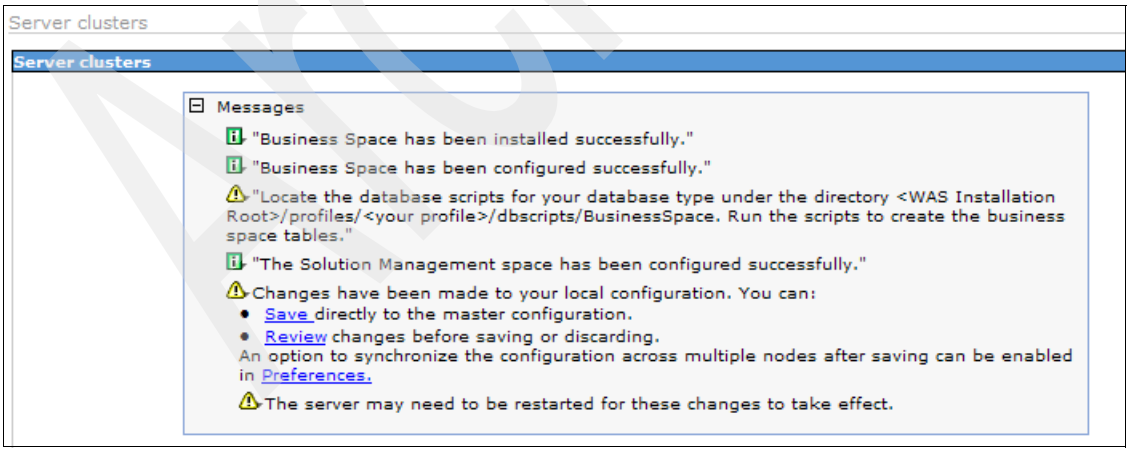


Figure 16-56 Server clusters

16.16 Using WebSphere Business Monitor Configuration Wizard

> **Note:** It is a good idea to make certain that the ulimit setting is correct. Refer to 16.2, “Preparing the machines” on page 514.

There is a new feature in WebSphere Business Monitor for v6.2, which is called the WebSphere Business Monitor configuration wizard. It makes the task of configuring topologies easier. Complete the following steps:

1. Go to the Integrated Solutions Console and log in.
2. In the navigation pane, click **Servers** → **WebSphere Business Monitor configuration**.
3. A checklist style interface indicates what components have or have not been configured (Figure 16-57).

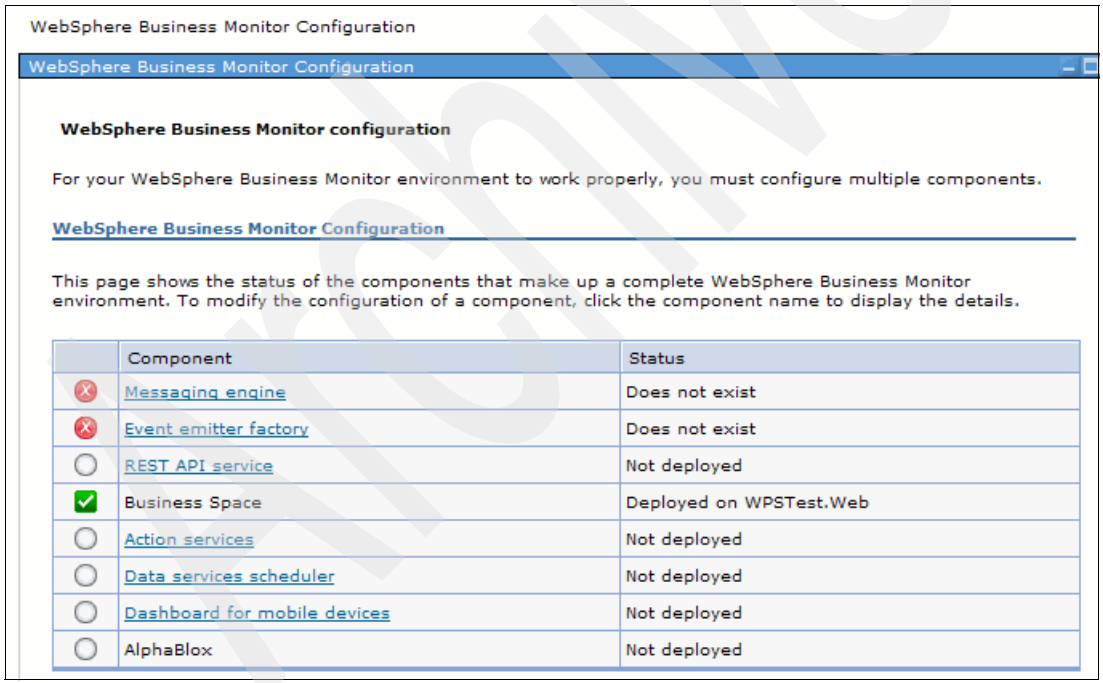


Figure 16-57 WebSphere Business Monitor configuration

16.16.1 Configuring the Messaging Engine

To do this:

1. On the WebSphere Business Monitor configuration panel, click **Messaging Engine**.
2. Click **Configure the Message Engine**.
3. On Select a bus member, click **Cluster**, and then select **WPSTest.Messaging**. Click **Next**.
4. On Select the type of messaging store, click **Data store**, then click **Next**.
5. On Provide the message store properties, click **Use existing data source**.
6. Set Data source JNDI Name to `jdbc/wbm/MonitorMEDatabase`.
7. Leave the schema name that is the default.
8. Uncheck the check mark for Create Tables.
9. The authentication alias should be set to `Monitor_JDBC_Alias`. Click **Next**.
10. A summary panel appears next. Click **Finish** (Figure 16-58).

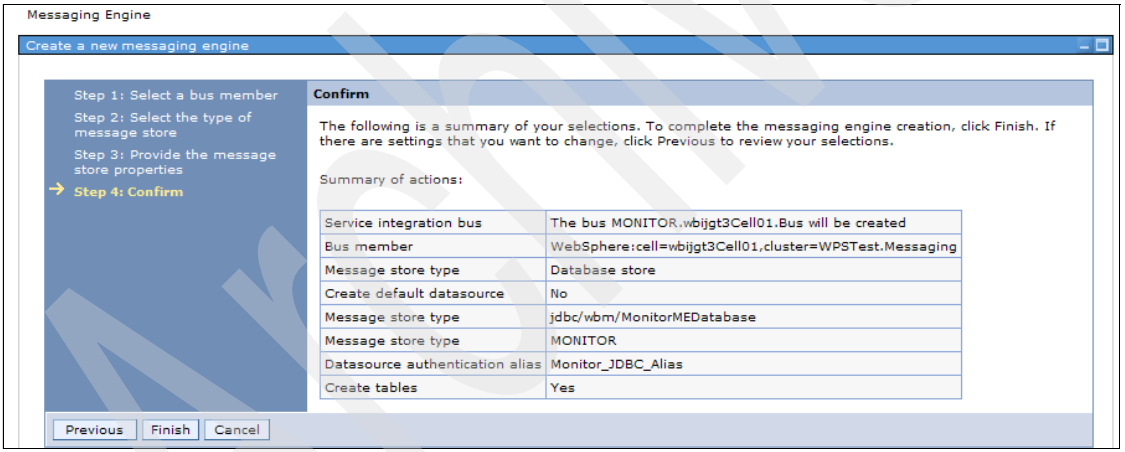


Figure 16-58 Monitor Messaging Engine configuration summary

11. Click the bread crumb **WebSphere Business Monitor configuration** and continue on into the next section.
12. Notice that there is a green check mark next to the Messaging Engine component. The status field should also indicate that it was deployed to our messaging cluster.

13. At this point, the `jdbc/wbm/MonitorMEDatabase` is pointing to the MONITOR database. However, earlier in this chapter, we created the Monitor messaging engine tables in the ME database. To change this jdbc resource:
    a. Go to the Integrated System console.
    b. Click **Resources** → **JDBC** → **Data sources**.
    c. Click **Monitor ME Database**.
    d. Change Database name to `MEDB`.
    e. Save all changes.
    f. Click **Servers** → **WebSphere Business Monitor configuration** and continue with the next section.

16.16.2 Configuring the Event Emitter Factory

To do this:

1. Click **Event Emitter Factory**.
2. Select the **WPSTest.Support** for the cluster to be used.
3. Click **Configure the Event Emitter Factory**.
4. When the event emitter factory has been configured, a message will appear.
5. Click the bread crumb **WebSphere Business Monitor configuration**.
6. Notice that there is a green check mark next to the Event Emitter Factory.

16.16.3 Configuring the REST API service

To do this:

1. Click **REST API service**.
2. Select **WPSTest.Web** for the cluster to deploy to.
3. Click **Deploy REST API Service**.
4. When this configuration task completes, a message will appear.
5. Click the bread crumb **WebSphere Business Monitor configuration**.
6. Notice that there is a green check mark next to REST API Service.

16.16.4 Configuring Action Services

To do this:

1. Click **Action Services**.
2. Select **WPSTest.Support** for the cluster to deploy to.
3. Click **Deploy Action Services**.
4. When this configuration task completes, a message will appear.
5. Click the bread crumb **WebSphere Business Monitor configuration**.
6. Notice that there is a green check mark next to Action Services.

16.16.5 Configuring the Data Services Scheduler

To do this:

1. Click **Data services scheduler**.
2. Select **WPSTest.Support** for the cluster to deploy to.
3. Click **Deploy Data Services Scheduler**.
4. When this configuration task completes, a message will appear.
5. Click the bread crumb **WebSphere Business Monitor configuration**.
6. Notice that there is a green check mark next to Data Services Scheduler.

16.16.6 Configuring the Dashboard for mobile devices

To do this:

1. Click **Dashboards for Mobile devices**.
2. Select **WPSTest.Web** for the cluster to deploy to.
3. Click **Deploy Dashboard for Mobile Devices**.
4. When this configuration task completes, a message will appear.
5. Click the bread crumb **WebSphere Business Monitor configuration**.
6. Notice that there is a green check mark next to Dashboard for Mobile Devices.

Once all the configuration tasks outlined in this section have been completed, the WebSphere Business Monitor configuration should look as illustrated in Figure 16-59.

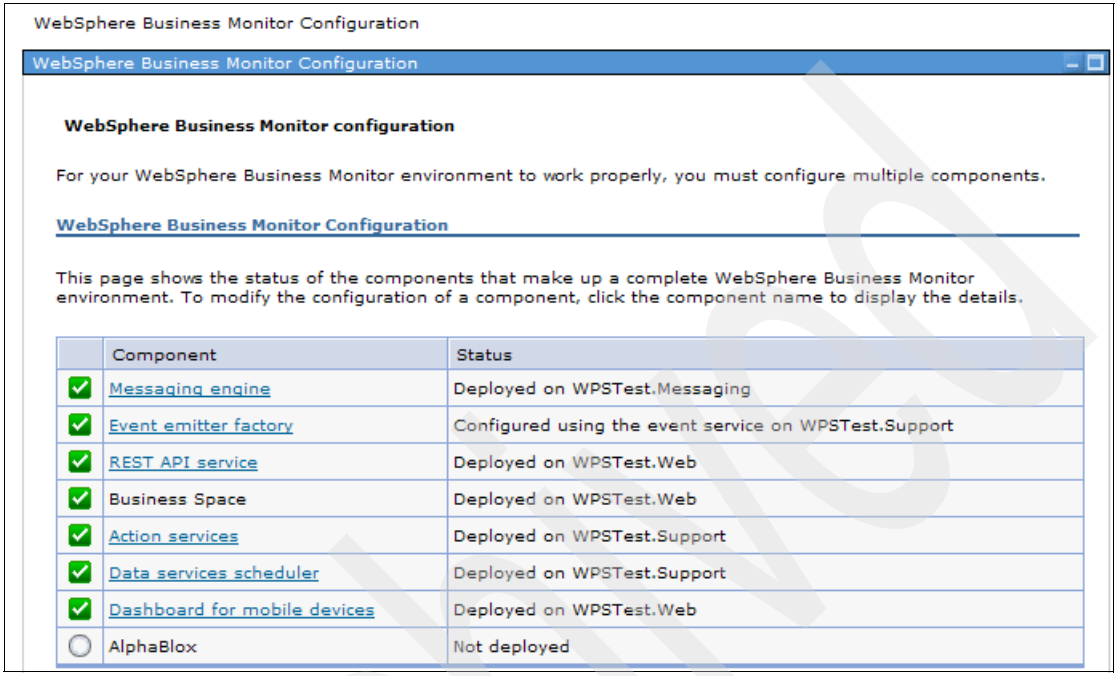


Figure 16-59 WebSphere Business Monitor configuration status

One final comment to make is that the Alphablox component is grayed out. This is because at this point in time, the Alphablox product has not yet been installed and configured.

One additional question check is to ensure that the Monitor Messaging Engine was created. To verify this, complete the following steps:

1. Go to the Integrated Solutions Console.
2. Click **Servers** → **Clusters**.
3. Click **WPSTest.Messaging**.

4. Click **Messaging Engines** and verify that the Monitor messaging engine appears as shown in Figure 16-60.

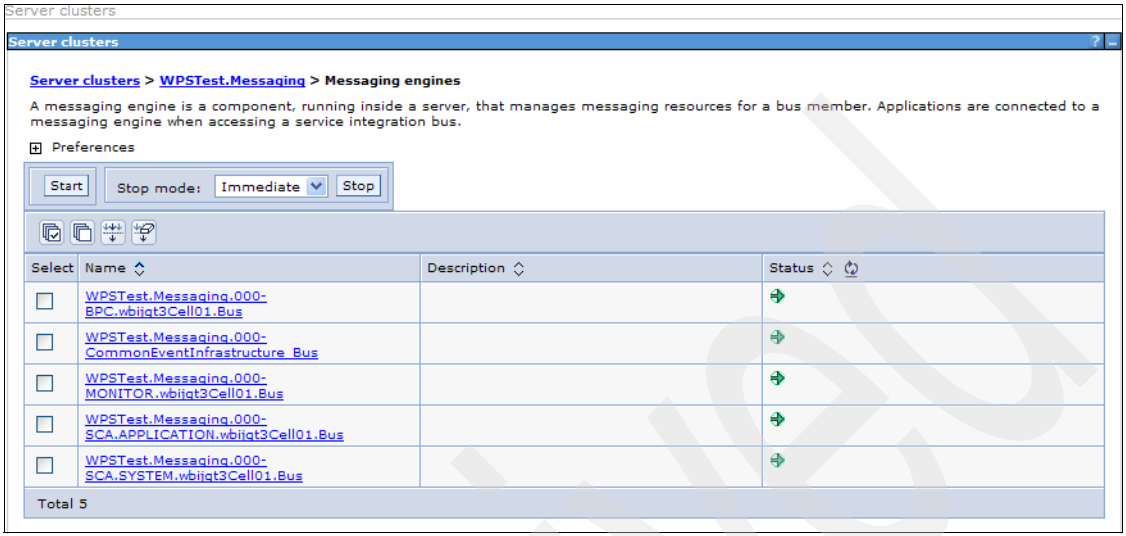


Figure 16-60 Monitor Messaging Engine

16.17 Installing Monitor's Emitter Service

WebSphere Business Monitor provides an API to emit events to WebSphere Business Monitor that is implemented using Representational State Transfer (REST) Services. You can use the API to send XSD style events. This is the application to support this feature in WebSphere Business Monitor and this must be manually installed in V6.2.0. To install this application:

1. Go to the Integrated System Console.
2. Click **Applications** → **Install New Applications**.
3. Click the radio button **remote file system** and then click **Browse**.
   a. Point to your cell manager (wbijgt3CellManager01 in our sample topology).
   b. On the remote file system we installed WebSphere Process Server in the directory `/monwas/ProcServer`. Under this directory is a directory named `installableApps.wbm`. It is within this directory location that you will find a file named `EmitterRestServices.ear`.
   c. Click **EmitterRestServices.ear**.
   d. Click **OK** and then click **Next**.

e. Click **Next** again to get to step 2.

f. At this step, map the module to the WPSTest.Support cluster, as shown in Figure 16-1 on page 512.

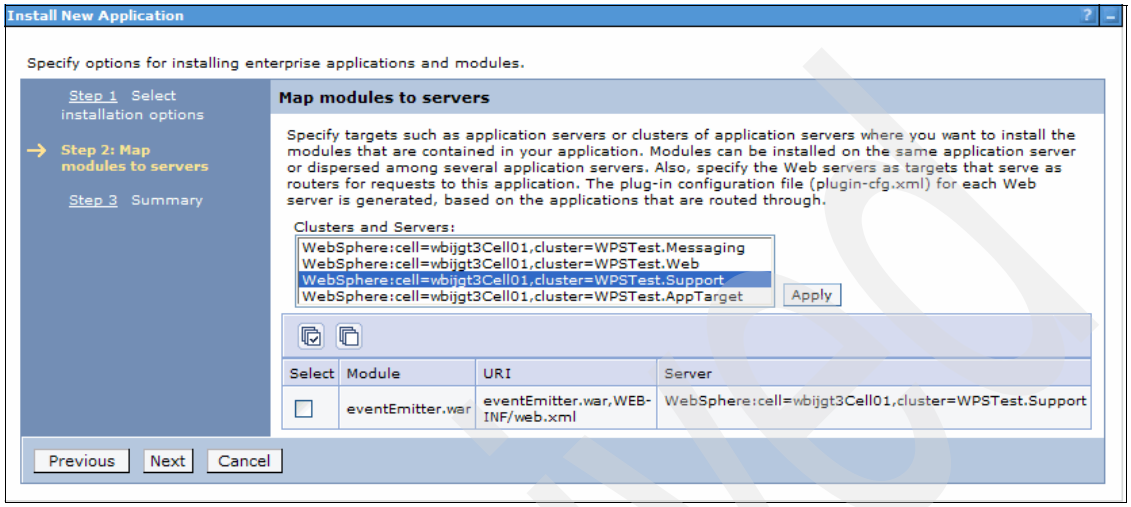


Figure 16-61 Mapping the Monitor Event Emitter Service to the Support cluster

4. Save all changes at this point.

16.18 Verifying Monitor Applications

It is a good idea at this point in time to make certain that there are no problems and that all the Monitor applications start properly:

1. Go to the Integrated System Console.
2. Click **Servers** → **Clusters**.
3. Stop the WPSTest.Web and the WPSTest.Support clusters. Then restart them.

> **Note:** If you wish to ensure that not just the Monitor applications, but all the (Process Server related) applications are starting properly, we recommend that you stop and restart all clusters now.

4. Click **Enterprise Applications** and make certain that the Monitor applications have started, as shown in Figure 16-62.

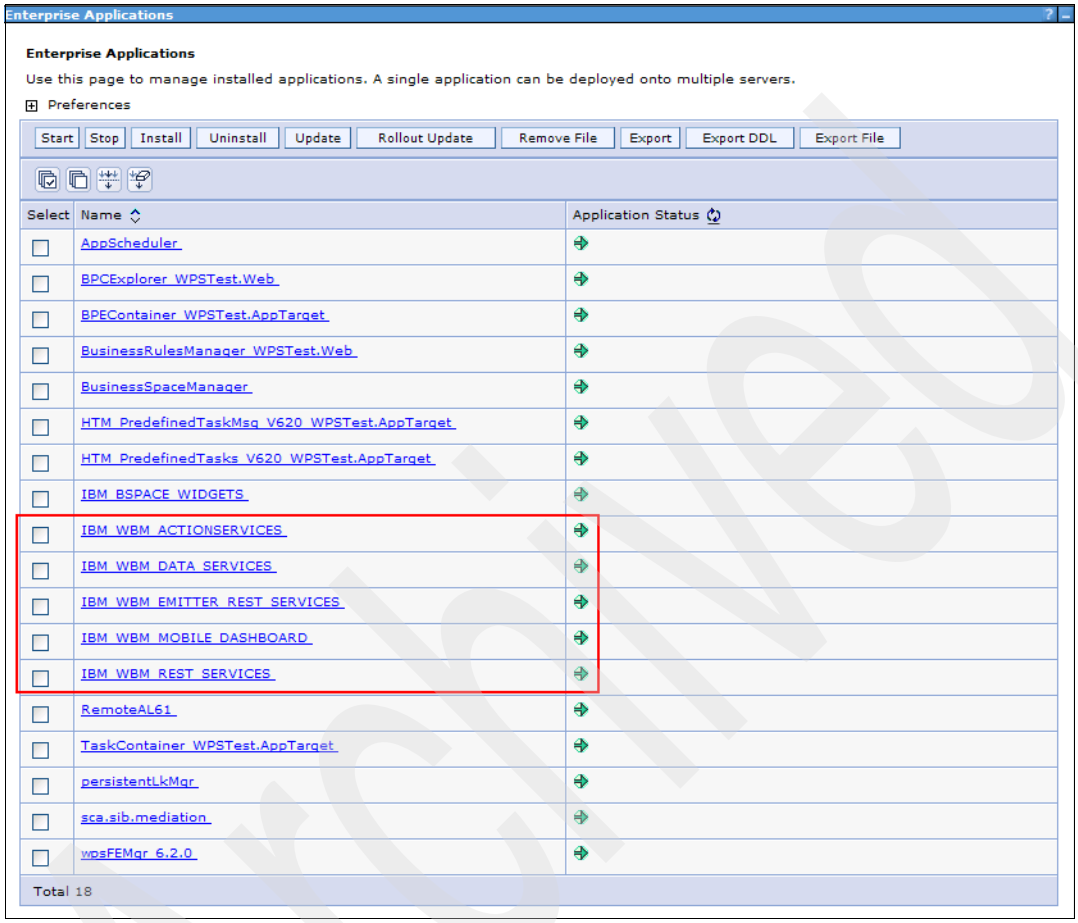


Figure 16-62 Application status

16.19 Configuring Fabric's Messaging Engine

The next task to perform is to run the scripts provided by WebSphere Business Services Fabric to create the messaging engine used by Fabric. Complete the following steps.

> **Note:** The deployment manager must be started as well as the node agents prior to running the following steps.

1. Open a command window on the AIX system where the deployment manager is running. In our topology, this is wbijgt3.rtp.raleigh.ibm.com.
2. Type `cd /monwas/ProcServer/profiles/Dmgr01/bin`.
3. Type:

   ```
   ./wsadmin.sh -lang jython -f
   /monwas/ProcServer/profileTemplates/dmgr.wbsfabric/actions/scripts/c
   luster/fabricSIBConfig.py
   ```

4. A pop-up window will appear prompting you for a user ID and password. Use `wps` for the user ID and `passw0rd` for the password. Click **OK**.
5. You will be asked to select the messaging engine cluster. Be sure to type the number associated with the WPSTest.Messaging cluster.
6. Press Enter for the next question. (The Fabric messaging engine will use the SCA Authentication Alias.)
7. Press Enter for the next question to accept the default schema name for the Fabric database.
8. When the script has finished running, control will be returned to the command prompt. Your console output should look similar to Example 16-4.

Example 16-4 Sample console output

```
# cd /monwas/ProcServer/profiles/Dmgr01/bin
# ./wsadmin.sh -lang jython -f
/monwas/ProcServer/profileTemplates/dmgr.wbsfabric/actions/scripts/c
luster/fabricSIBConfig.py
WASX7209I: Connected to process "dmgr" on node wbijgt3CellManager01
using SOAP connector;  The type of process is: DeploymentManager
The following are clusters are configured
1->WPSTest.AppTarget
2->WPSTest.Messaging
3->WPSTest.Support
4->WPSTest.Web
Select the messaging cluster.Enter the number. 2
```

```
Fabric Messaging engine can use the SCA System Bus settings for
security and ME database.Hit enter to use default value or enter N
to supply the values.
Enter the Fabric Messaging schema name.Hit enter to use default
value of FABRICME for Messaging schema or enter N to supply the
value.
Creating Fabric Bus :- Fabric.wbijgt3Cell01.Bus
Created Fabric Bus.
Synchronizing changes with all the nodes in cluster.
Done with synchronization for node Node01
Done with synchronization for node Node02
#
```

9. Go to the Integrated System Console and sign in:
   a. Click **Servers** → **Clusters**.
   b. Click **WPSTest.Messaging**, click **Stop** to stop the cluster, then select it again and click **Start**.
10. When the cluster has finished starting, click **WPSTest.Messaging**.
11. Click **Messaging Engines** and verify that the Fabric messaging engine appears as shown in Figure 16-63.

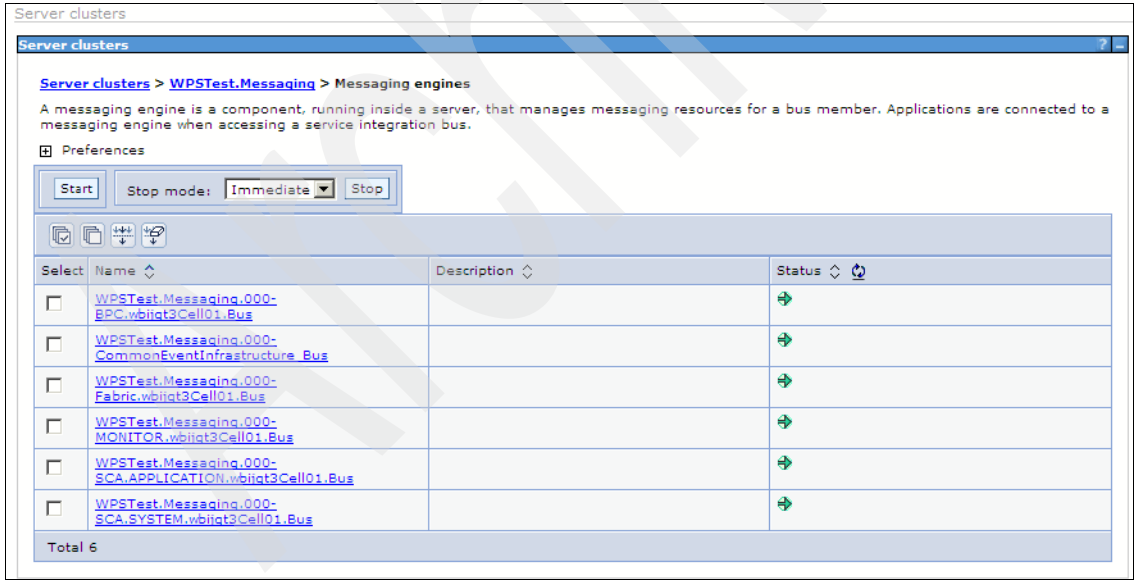


Figure 16-63 Fabric Messaging Engine

16.20 Configuring Fabric applications

The next task to perform is to run the scripts provided by WebSphere Business Services Fabric to deploy the Fabric applications to the Web cluster. Complete the following steps.

> **Note:** The deployment manager must be started as well as the node agents prior to running the following steps.

1. Open a command window on the AIX system where the deployment manager is running. In our topology, this is wbijgt3.rtp.raleigh.ibm.com.
2. Type `cd /monwas/ProcServer/profiles/Dmgr01/bin`.
3. Type:

   ```
   ./wsadmin.sh -lang jython -f
   /monwas/ProcServer/profileTemplates/dmgr.wbsfabric/actions/scripts/c
   luster/fabricAppDeploy.py
   ```
4. A pop-up window will appear prompting you for a user ID and password. Use `wps` for the user ID and `passw0rd` for the password. Click **OK**.
5. You will be asked to select the cluster to which to deploy the Fabric applications. Be sure to type the number associated with the WPSTest.Web cluster. The next question asks you to enter the user ID that would have access to the Fabric Administration console. Type ws`admin` and press Enter.
6. The next question asks you to enter the directory of the dmgr.fabric profile template. Type `/monwas/ProcServer/profileTemplates/WBSFabric` and press Enter.
7. Numerous messages will be displayed in the console window. Keep a watch out for any error information.
8. When the script has finished processing, control will be returned to the command prompt.
9. Sign in to the Integrated Systems Console.
10. Click **Applications** → **Enterprise Applications**.

11. Check the check box for each of the fabric applications shown in Figure 12 and then click **Start**.

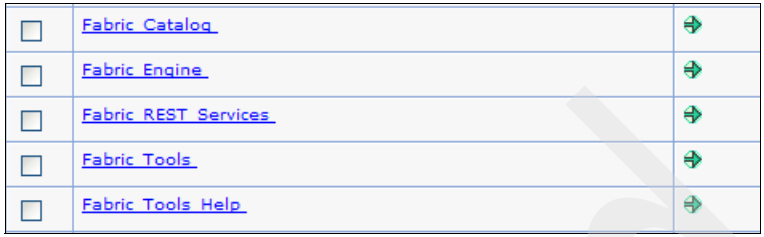


Figure 16-64 Fabric applications

12. Make sure that the fabric applications status turns to started. The amount of time it will take to start varies.

16.21 Configuring the IBM HTTP Server

The installation of the IBM HTTP Server is covered in 5.1.3, “Add a Web server” on page 103.

There is a very good tech note called “Guide to properly setting up SSL within the IBM HTTP Server,” available at the following URL:

http://www-01.ibm.com/support/docview.wss?uid=swg21179559

There is a very good tech note that discusses the details about how to exchange certificates, available at:

http://www-01.ibm.com/support/docview.wss?uid=swg21264477

Since security is enabled in the topology, SSL must also be enabled on the IBM HTTP Server. This essentially means adding the following lines to the configuration. See Example 16-5 for what was added to the HTTP configuration.

Example 16-5 HTTP configuration

```
LoadModule was_ap20_module
/monwas/IBM/HTTPServer/Plugins/bin/mod_was_ap20_http.so
WebSpherePluginConfig
/monwas/IBM/HTTPServer/Plugins/config/webserver1/plugin-cfg.xml
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so

Listen 443
<VirtualHost wbijgt3.rtp.raleigh.ibm.com:443>
ServerName wbijgt3.rtp.raleigh.ibm.com
```

```
SSLEnable
SSLClientAuth none
Keyfile "/monwas/IBM/HTTPServer/Plugins/config/webserver1/plugin-key.kdb"
SSLStashfile "/monwas/IBM/HTTPServer/Plugins/config/webserver1/plugin-key.sth"
ErrorLog "/monwas/IBM/HTTPServer/logs/sslerror.log"
TransferLog "/monwas/IBM/HTTPServer/logs/sslaccesserror.log"
</VirtualHost>
SSLDisable
```

16.22 Configuring the REST endpoints

Once the IBM HTTP Server has been configured to use SSL, there is one more change to make:

1. Go to the Integrated Solutions Console and log in.
2. In the navigation pane, click **Servers** → **Clusters**
3. Click the **WPSTest.Web** cluster.

4. On the right-hand side of the panel, under Business Integration, click **System REST Service Endpoints**. The panel that comes up is illustrated in Figure 16-65.

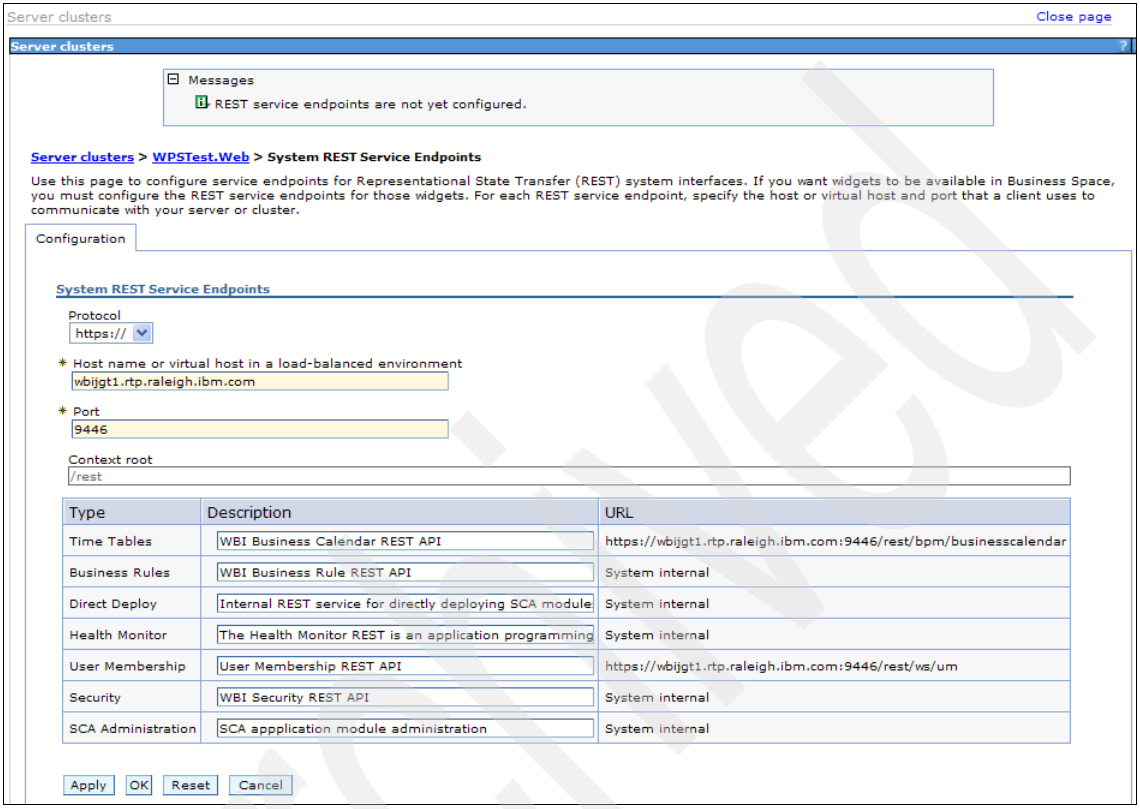


Figure 16-65 System REST Service Endpoints configuration

5. Change the Host name or virtual host in a load balanced environment to `wbijgt3.rtp.raleigh.ibm.com`. (This is the host name of your HTTP Server.)
6. Change the port to 443. Click **OK**.
7. Save all configuration changes.

8. If you repeat steps 1–4 above, you should see an indication that the REST services endpoints are configured on the Web cluster, as Figure 16-66 illustrates.

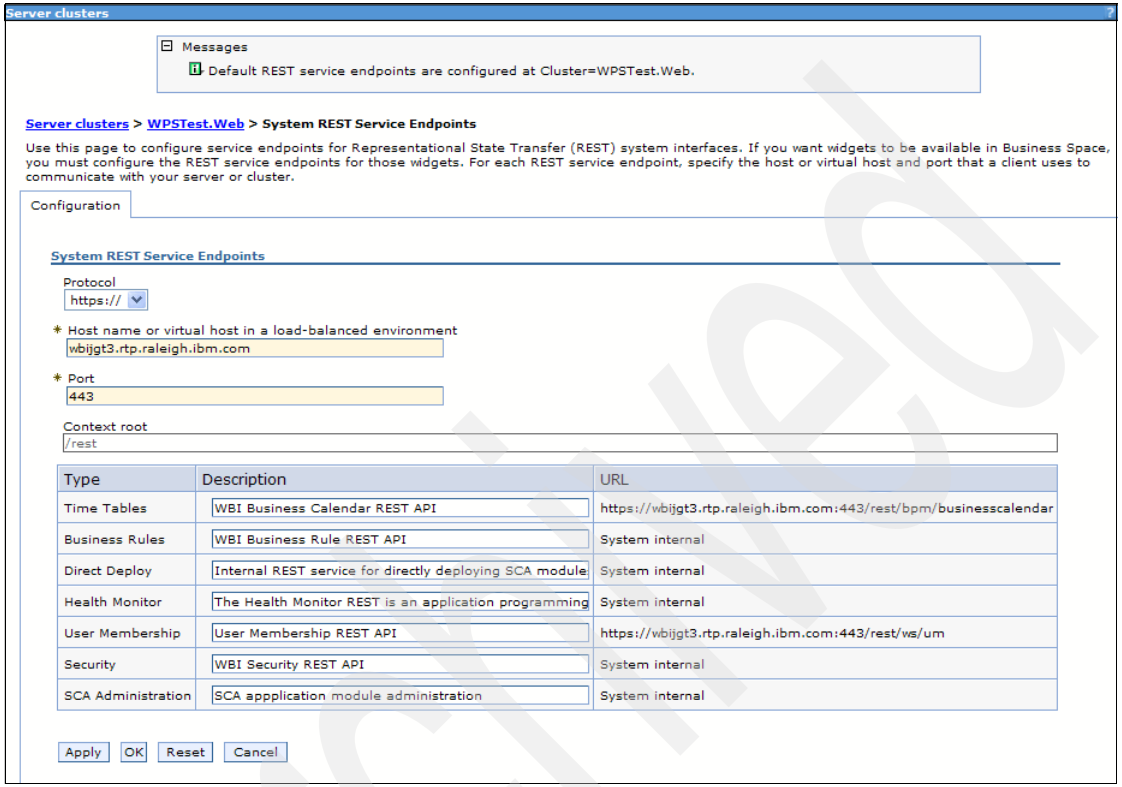


Figure 16-66 REST service endpoints

16.23 Business Space

In this production topology, Business Space has been deployed to the Web cluster. This chapter illustrates the necessary configuration for HTTP server and REST endpoints.

16.23.1 Widget Endpoint configuration

For any widgets that you wish to enable, there is an XML file containing the endpoint information. The file naming convention is such that the file name ends with `Endpoints`. In this topology, Example 16-6 shows how the monitorEndpoints.xml would be modified to support our use of the HTTP server. Notice that the https is specified and that our host name is pointing to our HTTP Server. It is not necessary to specify the port number, but if one were specified, it would be port 443 given how the HTTP server was configured.

Example 16-6 Widget endpoint configuration

```
<tns:BusinessSpaceRegistry
xmlns:tns="http://com.ibm.bspace/BusinessSpaceRegistry"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://com.ibm.bspace/BusinessSpaceRegistry
BusinessSpaceRegistry.xsd ">

  <tns:Endpoint>
    <tns:id>{com.ibm.wbimonitor}monitorServiceRootId</tns:id>
    <tns:type>{com.ibm.wbimonitor}monitorServiceRootId</tns:type>
    <tns:version>1.0.0.0</tns:version>
    <tns:url>https://wbijgt3.rtp.raleigh.ibm.com/rest/</tns:url>
    <tns:description>Location of backing services for Monitor
widgets</tns:description>
  </tns:Endpoint>

</tns:BusinessSpaceRegistry>
```

To configure:

1. In this topology, the `/monwas/ProcServer/BusinessSpace/registryData` directory contains the following endpoint files:
   - `bcmEndpoints.xml`
   - `bpcEndpoints.xml`
   - `fabricEndpoints.xml`
   - `hmEndpoints.xml`
   - `monitorABXEndpoints.xml`
   - `monitorEndpoints.xml`
   - `pubserverEndpoints.xml`
   - `smEndpoints.xml`
   - `visualStepEndpoints.xml`
   - `wpsEndpoints.xml`
   - `wsumEndpoints.xml`

2. For the endpoints that you wish to enable, just edit the endpoint xml file, as illustrated in Example 16-6 above.
3. Once the endpoint files have been modified, they must be copied to each of our nodes. Specifically, the endpoint files must be copied to the following locations:
   - `/monwas/ProcServer/profiles/Node01/BusinessSpace/registryData`
   - `/monwas/ProcServer/profiles/Node02/BusinessSpace/registryData`
4. Once the files have been copied, stop and restart the Web Cluster in order to have the changes take effect.

16.23.2 Logging in to Business Space

Once the HTTP Server has been configured and the REST Services Endpoints have been configured to point to our HTTP server, the next test is to ensure that we can sign in to Business Space. In our topology, we point our browser to the URL `https://wbijgt3.rtp.raleigh.ibm.com/BusinessSpace`, and the window shown in Figure 16-67 should appear.

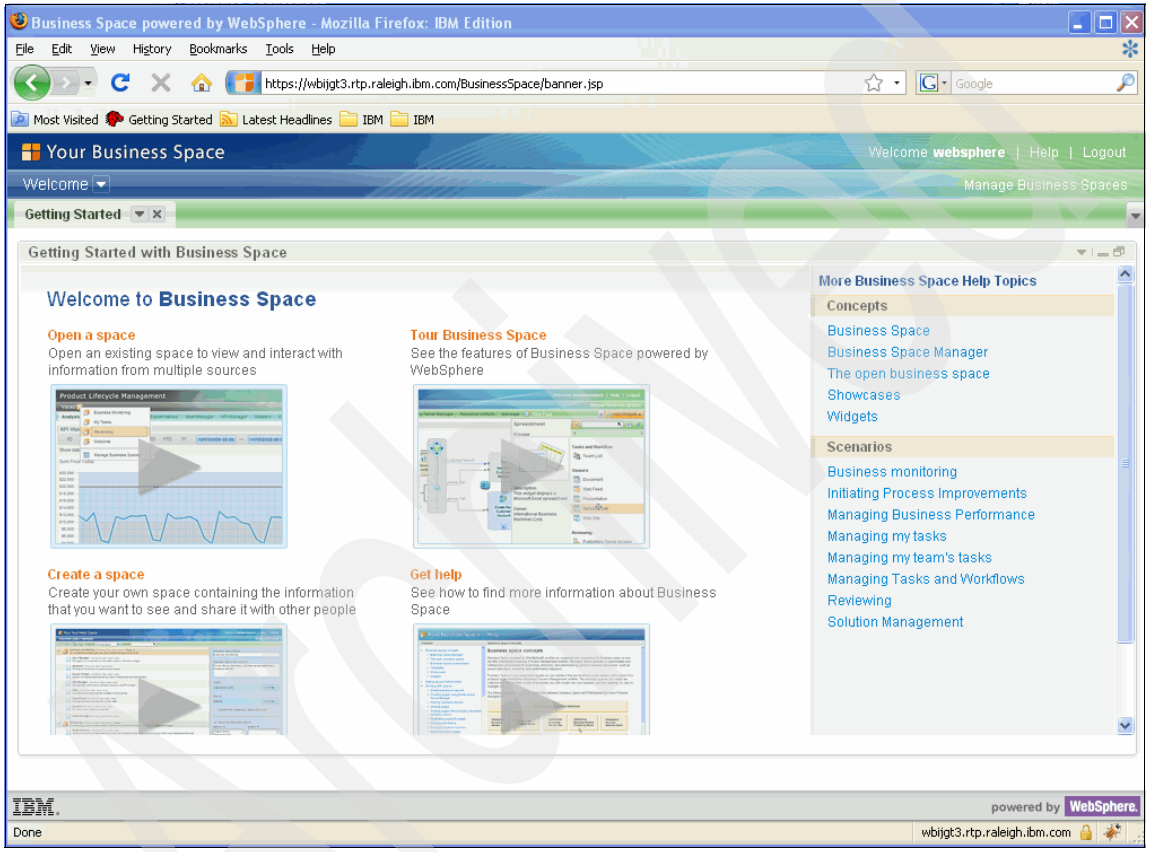


Figure 16-67 Business Space

16.24 Installing and configuring Alphablox

Refer to 15.8.1, “Alphablox” on page 475 to install Alphablox.

Tip: It has been our experience that you should install Alphablox on one cluster member in the WPSTest.Web cluster and configure that first. Then proceed to install Alphablox on a second cluster member.

In order for WebSphere Business Monitor and WebSphere Business Services Fabric to work properly, you must get a new Alphablox 9.5.2 build 28. Refer to the following URL for information about how to obtain this:

http://www-01.ibm.com/support/docview.wss?rs=802&context=SSSRR3&dc=D400&uid=swg24022453&loc=en_US&cs=utf-8&lang=en

Part 5

Appendixes

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 publication Web server. Point your Web browser to:

ftp://www.redbooks.ibm.com/redbooks/SG247732

Alternatively, you can go to the IBM Redbooks publication Web site at:

ibm.com/redbooks

Select **Additional materials** and open the directory that corresponds with the IBM Redbooks publication form number, SG247732.

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.

Abbreviations and acronyms

BPC	Business Process Choreographer
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Modeling Notation
BRM	Business rules manager
CBA	Composite Business Application
CEI	Common Event Infrastructure
CORBA	Common Object Request Broker Architecture
DA	Dynamic Assembler
EAR	Enterprise Archive
EIS	Enterprise Information System
HA	high availability
HTM	Human Task Manager
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines Corporation
ISC	Integrated Solutions Console
IT	Information Technology
ITCAM	IBM Tivoli Composite Application Manager
ITSO	International Technical Support Organization
J2EE	Java 2 Platform, Enterprise Edition
JMS	Java Message Service
JNDI	Java Naming and Directory Interface
JSP	JavaServer Pages
JVM	Java Virtual Machine
KPI	key performance indicator
LB	load balancing
LTPA	Lightweight Third Party Authentication
MDB	Message-Driven Beans
QoS	Quality of Service
REST	REpresentational State Transfer
RMRS	Remote Messaging and Remote Support
SCA	Service Component Architecture
SCDL	Service Component Definition Language
SDO	Service Data Object
SIP	Session Initiation Protocol
SOA	Service-Oriented Architecture
SPNEGO	Simple and Protected GSS-API Negotiation Mechanism
SSL	Secure Sockets Layer
SSO	single sign-on
SWA	SOAP with Attachments
TAI	Trust Association Interceptors
TEMA	Tivoli Enterprise Monitoring Agents
TEMS	Tivoli Enterprise Monitoring Server
TEPS	Tivoli Enterprise Portal Server
TLS	Transport Layer Security
UTE	Unit Test Environment
VIN	Vehicle Identification Number

WPSRS	WebSphere Process Server Recovery Service

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 publications” on page 601. Note that some of the documents referenced here may be available in softcopy only.

- *z/OS: WebSphere Business Process Management V6.2 Production Topologies, SG24-7733*
- *IBM WebSphere Application Server V6.1 Security Handbook*, SG24-6316
- *WebSphere Application Server Network Deployment V6: High Availability Solutions*, SG24-6688
- *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240
- *Production Topologies for WebSphere Process Server and WebSphere ESB V6*, SG24-7413
- *WebSphere Business Process Management V6.1.2 Production Topologies*, SG24-7665
- *Best Practices for SOA Management*, REDP-4233
- *WebSphere Application Server V6.1: JMS Problem Determination*, REDP-4330
- *IBM WebSphere Business Process Management V6.1 Performance Tuning*, REDP-4431

How to get Redbooks publications

You can search for, view, or download Redbooks publications, Redpapers publications, Technotes, draft publications, and Additional materials, as well as order hardcopy Redbooks publication, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

WebSphere Business Process Management V6.2 Production Topologies

WebSphere Business Process Management V6.2 Production Topologies

Building and extending WebSphere Process Server topologies

Incorporating WebSphere Business Services Fabric

Integrating WebSphere Business Monitor

This IBM Redbooks publication addresses the configuration, administration, and security of the key runtime environments in WebSphere Dynamic Process Edition V6.2.

Part 1 of this publication introduces production topology concepts and terminology, and provides security considerations.

In Part 2, through a series of step-by-step instructions you will learn how to select and create a production topology environment based on WebSphere Process Server deployment environment patterns. You will learn how to secure this environment and administer it. This part also contains chapters on extending these topologies, monitoring them with IBM Tivoli Monitoring, and accessing them with Business Space powered by WebSphere and Lotus Forms Client.

Part 3 extends the Remote Messaging and Remote Support production topology for WebSphere Process Server, describing step-by-step how to add WebSphere Business Services Fabric and WebSphere Business Monitor to that topology.

Finally, Part 4 describes how to build a complete WebSphere Dynamic Process Edition production topology from the ground up using the new Four Cluster production topology.

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

SG24-7732-00 ISBN 073843292X