WebSphere software

IBM

WebSphere Business Process Management V6.1.2 Production Topologies

Securing, administering, and extending WebSphere Process Server topologies

Incorporating WebSphere Business Services Fabric

Integrating WebSphere Business Monitor

Peter Daly
Martin Keen
Ryan Malynn
Thomas McManus
Karen Poyer
Julia Reder
Mohamed Shamseldin Salem
Kevin Senior
Jeffrey Slone
Vignesh Velusamyravindran

Redbooks

ibm.com/redbooks

International Technical Support Organization

WebSphere Business Process Management V6.1.2 Production Topologies

November 2008

SG24-7665-00

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

First Edition (November 2008)

This edition applies to Version 6.1.2 of WebSphere Process Server, WebSphere Business Monitor, and WebSphere Business Services Fabric.

Contents

Notices

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

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

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

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

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

Any references in this information to non-IBM 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®
AlphaBlox®
alphaWorks®
CICS®
DataPower®
DB2 Universal Database™
DB2®
developerWorks®
FileNet®
HACMP™
IBM®
IMS™
Lotus®
Redbooks®
Redbooks (logo) ®
Tivoli®
WebSphere®
Workplace™
Workplace Messaging®
z/OS®

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.

Snapshot, and the NetApp logo are trademarks or registered trademarks of NetApp, Inc. in the U.S. and other countries.

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.

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

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

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries 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

The IBM® WebSphere® Dynamic Process Edition is a comprehensive set of role-based, SOA-enabled product capabilities providing customers 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: IBM WebSphere Process Server, WebSphere Business Services Fabric, and WebSphere Business Monitor.

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 book also contains a chapter on extending existing production topologies to add components such as additional clusters.

This Redbooks publication also provides practical examples demonstrating how to incorporate WebSphere Business Services Fabric and WebSphere Business Monitor into existing topologies. The book contains extensive examples of working with all of these products in distributed environments. A separate publication covering z/OS® is forthcoming.

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.

Figure 0-1 From left-to-right: Ryan, Peter, Kevin, Mohamed, Vignesh, Julia, Jeff, Karen, Martin, and Tom

Peter Daly is a WebSphere Consultant in the Software Group of IBM UK. Prior to joining IBM he was employed as a UNIX® administrator and programmer in an international research laboratory in France and was a manager of supercomputer systems in the UK. He now specializes in WebSphere Process Server and has contributed to other redbooks. Peter holds Bachelor's Degrees in Physics and Computer Science. His favorite phrase is "Cymru am byth."

Martin Keen is a Senior 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 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.

Ryan Malynn is an Information Developer in the United States. He has 9 years of experience in the software field. He has worked at IBM for almost 8 years. His areas of expertise include content management and security. He has written extensively on WebSphere Application Server and WebSphere Portal Server.

Thomas McManus is a Senior Software Engineer with IBM SWG Business Partner Technical Strategy and Enablement. He has ten 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

Karen Poyer is an intern with ITSO Raleigh for the summer of 2008. She attends Creighton University, in Omaha, Nebraska and will graduate in May 2009 with a Bachelor's Degree in Mathematics and minors in Computer Science and Vocal Music.

Julia Reder is a certified Software IT Specialist with IBM Sales and Distribution; Western Region in San Francisco, CA. She has extensive experience with IBM software technical sales, development, and performance evaluation. Julia holds a Bachelor of Science Degree in Physics and a Master of Arts Degree in Asian Studies.

Mohamed Shamseldin Salem is a Senior IT Specialist with IBM Software Group in Cairo Technology and Development Center (C-TDC) Egypt. He has 5 years working in WebSphere Business Monitor information development, and SWAT teams. Mohamed has extensive experience in installation, configuration, and security for the WebSphere product stack. He provided technical support for WebSphere products in Europe and Africa in critical customer situations. He is a certified software solution developer for WebSphere Business Monitor and WebSphere Portal Server. Mohamed holds a Bachelor's Degree in Computer Engineering from Cairo University, Egypt.

Kevin Senior is an IBM certified IT Specialist working for the Worldwide Technology Practice within IBM Software Services for WebSphere and based out of the IBM Hursley laboratory in the UK. He has 28 years experience at IBM as a Systems Programmer working with IMS™, DB2®, CICS®, and WebSphere brand products on z/OS. Currently, he specializes in WebSphere Portal Server and WebSphere Process Server for z/OS. For ITSO, Kevin coauthored several IBM Redbooks and Redpapers. Although Kevin is English, he now lives in Italy and works throughout Europe.

Jeffrey Slone is a Course Developer and Instructor with WebSphere Education in the US. A twelve-year veteran of IBM, Jeffrey has developed and delivered technical training on many of the WebSphere Business Integration product offerings, including WebSphere Process Server, WebSphere Business Monitor, WebSphere Business Modeler, and WebSphere Integration Developer. He authored the IBM Redpaper, *Lotus Workplace Messaging Administration Guide*, REDP-3860, and was a contributing author for the following IBM Redbooks Publications: *Lotus Workplace 1.1 Products Deployment Guide*, SG24-7087, *Lotus Workplace 2.0.1 Products: Deployment Guide*, SG24-6378, and *IBM Workplace Collaboration Services: Release 2.5 Deployment Guide*, SG24-6777. Jeffrey holds a Master's Degree in Computer Science from Southern Polytechnic State University.

Vignesh Velusamyravindran is an IT Architect in IBM India Software Labs. He has 9 years of experience in the IT field. He works with IBM business partners to architect/develop SOA based applications using WebSphere portfolio. His area of expertise include developing and architecting SOA-based distributed enterprise applications. He holds a Bachelor's Degree in Physics and Master's Degree in Computer Science.

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

Charlie Redlin
IBM Software Group, Application and Integration Middleware Software, WebSphere Software Architect

Matthew Kelm
IBM BPM Customer First Lab

Michele Chilanti
IBM Software Group, Application and Integration Middleware Software, Senior Consultant - WebSphere Services

Travis Nelson
IBM Software Group, Application and Integration Middleware Software, Adapter Competency

Chuck Misuraca
IBM Software Group, AIM Services.

Stephen Gibney
IBM Software Services for WebSphere, UKISA.

Mohamed Saeed
Senior IT Specialist. WebSphere Business Monitor SWAT team, IBM Egypt.

Mohamed Hegazy
Senior IT Specialist, IBM Software Services for WebSphere

Keys Botzum
Senior Technical Staff Member, IBM Software Services for WebSphere

Jens Engelke
Senior IT Specialist, IBM WebSphere Solution Center

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 and concepts

1

Basic concepts and business process management product descriptions

This chapter provides an introduction to the fundamental concepts and technologies that apply when deploying business process management (BPM) solutions. This chapter contains the following sections:

1.1 The IBM BPM Suite

The IBM BPM Suite is a set of collaborative, role-based capabilities that allow the customer to model, simulate, execute, rapidly change, monitor, and optimize business processes. The IBM BPM Suite combines capabilities from across IBM and offers a choice between two foundational offerings, the IBM WebSphere Dynamic Processes Edition and the IBM FileNet® Active Content Edition. A diagram of these offerings and their components is shown in Figure 1-1.

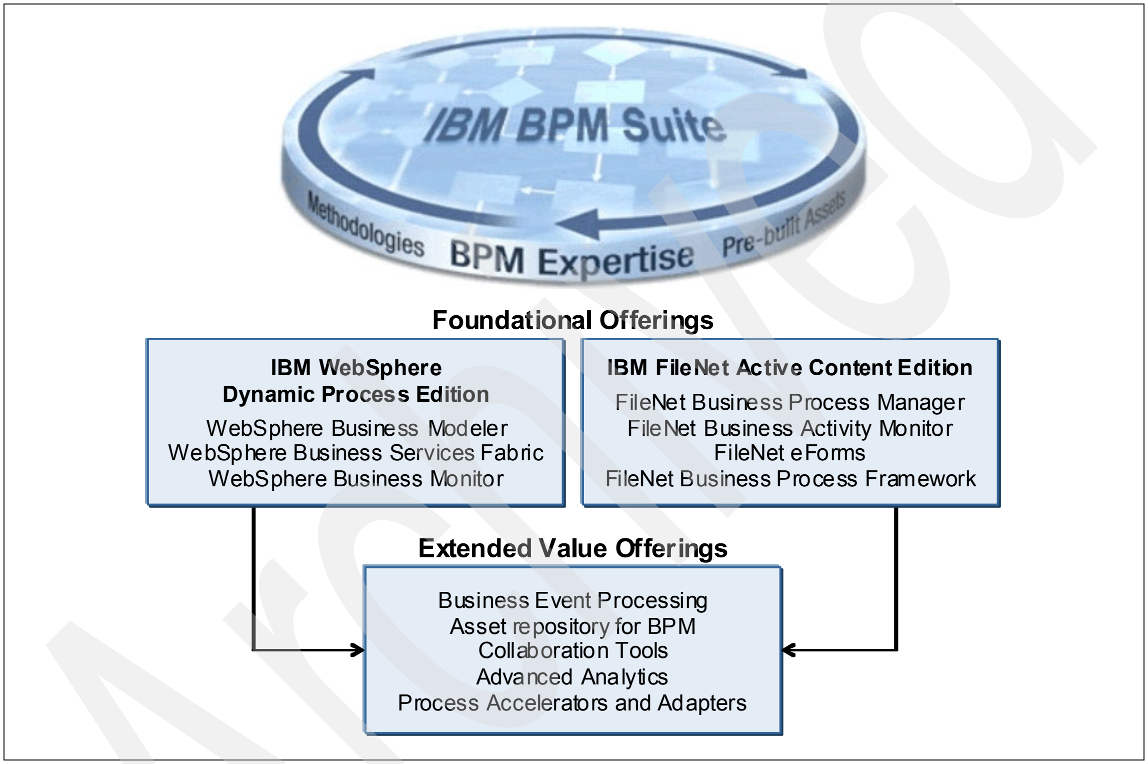


Figure 1-1 The IBM BPM Suite and its two foundation offerings

WebSphere Dynamic Processes Edition utilizes WebSphere Business Modeler, WebSphere Business Services Fabric, and WebSphere Business Monitor, while FileNet Active Content Edition employs FileNet Business Process Manager, FileNet Business Activity Monitor, FileNet eForms, and FileNet Business Process Framework. This Redbooks publication focuses on IBM WebSphere Dynamic Process Edition.

1.2 IBM WebSphere Dynamic Process Edition

IBM WebSphere Dynamic Process Edition is the core offering from the IBM BPM Suite. A comprehensive set of role-based, SOA-enabled product capabilities, it provides customers the ability to optimize processes continuously and adapt them to rapidly changing needs. It includes three products:

- IBM WebSphere Business Modeler Advanced V6.1.2

  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 V6.1.2

  An SOA-based process engine capable of unique dynamic execution of business processes determined on the fly using business service policies and diverse, managed service selection. The WebSphere Business Services Fabric software stack is shown in Figure 1-2.

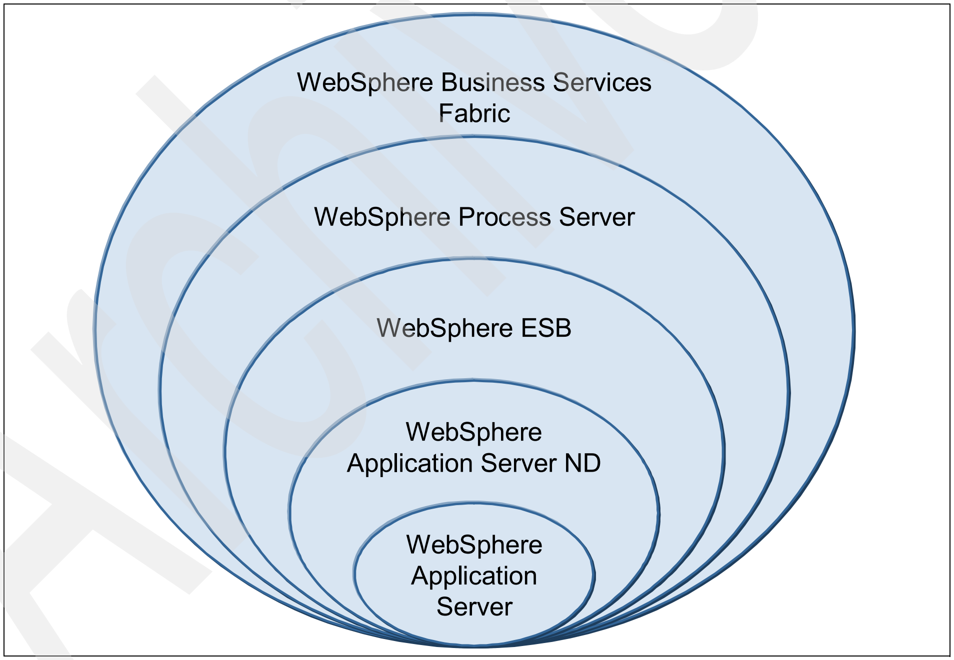


Figure 1-2 WebSphere Business Services Fabric software stack

- IBM WebSphere Business Monitor V6.1.2

  Comprehensive business activity monitoring providing a real-time view of your business processes and operations.

1.2.1 WebSphere Business Modeler

WebSphere Business Modeler offers capabilities for users 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 components:

- 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 far-reaching tools that allow the user 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. New in V6.1.2 is a tool that allows the user to model the business, with drag-and-drop capabilities to help set up simulations. This makes it easy to analyze workloads and bottlenecks. The Business Measures View and the Business Measures Details dialog box allow for better monitoring.

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 business processes.

New in V6.1.2, users of WebSphere Business Modeler Publishing Server can import user-interface form views for human tasks and then review and comment on the forms. Also in V6.1.2, WebSphere Business Modeler Publishing Server can display process models created with Business Process Modeling Notation (BPMN). BPMN is a standard graphical notation for drawing business practice models or using graphical notation.

1.2.2 WebSphere Application Server

WebSphere Application Server is the foundation of the IBM WebSphere software platform and a key building block for SOA. It provides a transaction engine that helps the user build, run, integrate, and manage dynamic applications. WebSphere Application Server allows the user to run services in a reliable, scalable, highly-available environment to ensure business opportunities are not lost due to application downtime.

1.2.3 WebSphere Enterprise Service Bus

WebSphere Enterprise Service Bus (ESB) is the mediation layer that runs on top of the transport layer within WebSphere Application Server. As such, WebSphere ESB 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 ESB uses Service Component Architecture (SCA) as its invocation model, which is why SCA is part of the first layer of elements. Also, Common Event Infrastructure (CEI), the foundation for monitoring business performance, is part of that same layer.

From the ESB definition given by SOA, 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 ESB 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 talk to WebSphere MQ, WebSphere Message Broker, or an adapter.

The modules in charge of performing the operations for WebSphere ESB 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 ESB.

1.2.4 WebSphere Process Server

WebSphere Process Server is an SCA-compliant runtime element that provides a fully converged, standards-based process engine that is 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. Following the principles of SCA, there is 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 human task can be replaced with a business rule without the need to modify the business process.

1.2.5 WebSphere Business Services Fabric

WebSphere Services Fabric (hereinafter called Fabric) is an end-to-end SOA platform to model, assemble, deploy, manage, and govern composite business applications. Fabric includes WebSphere Process Server (runtime environment) and WebSphere Integration Developer (design time tooling). With Fabric, organizations can assemble business-level services into extended, cross-enterprise business processes and solutions.

Fabric uses Composite Business Applications (CBA) which leverage business-level services. CBA's allow for more flexible business solutions, in contrast to the rigid business policies that many subscribers might beforehand have dealt with. Additional features allow the user a good deal of control over business processes, such as incrementally transforming core business processes to increase efficiency and rapidly enabling multichannel service delivery to provide higher service levels at a lower cost.

1.2.6 WebSphere Business Monitor

WebSphere Business Monitor is an integral part of the IBM BPM Suite. It is a comprehensive business activity monitoring solution that provides a near real-time view of business performance.

WebSphere Business Monitor monitors activities or processes by receiving and processing business events, called *common base events*, from business applications. The events that the WebSphere Business Monitor server receives reflect the user's business activity. Information processed from events is stored in the WebSphere Business Monitor database.

In summary, to monitor business operations, WebSphere Business Monitor offers the following functions:

- Captures business-related data that is specified by you requests from business applications 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
- 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

1.3 Basic concepts

This section introduces basic concepts that are used throughout this book. Each concept is described in the following sections:

- "SOA programming model"
- "Business processes" on page 11
- "Composite Business Application" on page 12
- "Business Space" on page 13

1.3.1 SOA programming model

This section introduces two key components of the SOA programming model: Service Component Architecture and Service Data Objects.

Service Component Architecture

SCA is a model for application development that splits the application function from the implementation details. SCA defines *modules* and *components* that are connected using standard interfaces:

- Module

  A module performs or supports a specific business function and can be deployed directly. Modules can be incorporated into many applications, increasing the potential for re-use across the organization. A module is constructed of one or more components.

- Component

  A component is a discrete, reusable unit that provides published interfaces and references other components' interfaces. Components can be implemented with many different technologies, such as Plain Old Java Objects (POJO), Enterprise Java Beans (EJB™), Business Process Execution Language (BPEL), or even a simple scripting language such as Perl.

  Components expose business-level interfaces to the application business logic so that the service can be used or invoked. The interface of a component defines the operations that can be called and the data that is passed, such as input arguments, returned values, and exceptions. Import and export components also have interfaces so that the published service can be invoked.

  All components have interfaces of the Web Services Description Language (WSDL) type. Only Java components support Java-type interfaces. If an SCA component, SCA import, or SCA export has more than one interface, all interfaces must be the same type.

  Components can be called synchronously or asynchronously regardless of whether the implementation is synchronous or asynchronous. The preferred interaction style can be either the same as or different than the implementation. The asynchronous interaction advertises to interface users that it includes at least one operation that can take a significant amount of time to complete. Consequently, the calling service must avoid keeping a transaction open while waiting for the operation to complete and send its response.

Service Data Objects

Service Data Objects (SDO) provide a framework for the design and use of business objects in an SOA. The fundamental concept in the SDO architecture is the *data object*. In fact, the term SDO is often used interchangeably with the term data object. A data object is a data structure that holds primitive data, multi-valued fields (other data objects), or both.

The data object also has references to metadata that provide information about the data found in the data object. In the SDO programming model, data objects are represented by the commonj.sdo.DataObject Java interface definition. This interface includes method definitions that allow clients to obtain and set the properties associated with DataObject.

Another important concept within the SDO architecture is the *data graph*. A data graph is a structure that encapsulates a set of data objects. From the top level data object in the graph, all other data objects can be reached by traversing the references from the root data object. In the SDO programming model, data graphs are represented by the commonj.sdo.DataGraph Java interface definition.

1.3.2 Business processes

This section discusses some of the fundamental concepts relating to business processes.

Business Process Execution Language

The Business Process Execution Language (BPEL or WS-BPEL) is a description language, based on the Extensible Markup Language (XML), that defines business processes and the logic that is required to perform these processes. BPEL code can be executed to provide services on application servers such as WebSphere Process Server.

Long-running and short-running processes

Business process flows are characterized by the length of time that they are expected to run and are classed as either long-running (macro-flow) or short-running (micro-flow) processes. This fundamental classification is aligned to the potential run-time of a process rather than the average run-time. Ultimately, it is decided on the basis of whether a process should be written out to disk or held in memory. Processes that involve human task elements are classified as long-running or macro-flow processes.

Human tasks

Human tasks are components that package up human interaction within the flow of the business process. The components enable manual creation, allocation, escalation, and tracking of process instances.

Business relationships and rules

In business integration scenarios, it is often necessary to access the same data (for example, client records) in various backend systems, for example, an Enterprise Resource Planning (ERP) system and a Customer Relationship Management (CRM) system.

A common issue when keeping business objects synchronized is that different backend systems use different schemas to represent the same objects. Creating and maintaining mappings between these schemas is a complex task. WebSphere Process Server simplifies the process by providing the relationship service to establish mappings between objects in these disparate backend systems. These relationships are then accessed when translating one business object format into another.

WebSphere Process Server also includes the Business Rules Manager (BRM) Web-based runtime tool for the business analyst. Business rules are a means of implementing and enforcing business policy through the externalization of business function. This enables a business environment to become more responsive by allowing process changes to be dynamically applied. The BRM tool allows for updating of business rules as business needs dictate without affecting other SCA services and deployed processes.

Invocation methods

Invocation methods can be split into two logical groups:

- Synchronous

  A synchronous invocation is one in which a client application process makes a call and waits for a response before proceeding. If there is no further work to do, the client application process ends.

- Asynchronous

  An asynchronous invocation is one in which a client application process makes a call and does not wait for a response before proceeding.

  There are different types of asynchronous invocations:

  – Asynchronous one-way invocation

    A client application process makes a call and proceeds. It does not expect to receive a response from the server application.

  – Asynchronous two-way deferred response invocation

    A client application process makes a call and proceeds without waiting for a response. The client application process then intermittently polls for a response message.

  – Asynchronous two-way with callback invocation

    A client application process makes a call and proceeds without waiting for a response. A service sends the response back to the client process upon completion of processing the request.

1.3.3 Composite Business Application

A Composite Business Application (CBA) is a collection of related and integrated business services that provide a specific business solution and support multiple business processes that are built on SOA. A business service is a building block and is designed and constructed for reusability, whereas a CBA is the derivative of a combination of business services.

At first glance, this may appear as just a simple collection of business services. A CBA, however, is a broader more comprehensive view of the business solution. The CBA is more specialized to the business solution and drives the process, channels, roles, and business object model of the overall business solution. The CBA leverages business services to deliver the final solution. A CBA also shares many of the following characteristics that are associated with business services:

- Designed at business level to deliver a specific business outcome
- Uses business policies and metadata to describe and explain service and solution characteristics, such as costs, availability, supported roles, supported channels, standards, and operational capabilities
- Increases straight-through processing by automating white spaces in the process
- Leverages industry models in support of interoperability and common understanding
- Supports multiple consumption channels (Web, B2B, and so on)
- Derived from multiple business services

Embracing a business services platform, such as WebSphere Business Services Fabric, provides designers with the ability to create solutions without tremendous amounts of integration. Business services and processes share more meaning through semantic glossaries, service interface specifications, and platform neutrality.

1.3.4 Business Space

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.

Business Space uses mashup technology, which refers to Web pages which are created by combining Web applications (widgets) to make novel interfaces. Users can customize these business spaces to view runtime business data. Administrators can create new spaces or use the predefined scenarios that are shipped with Business Space.

Business Space is shipped with WebSphere Process Server, WebSphere Business Monitor, and WebSphere Business Modeler Publishing Server. It includes templates for predefined scenarios. Business Space also includes information from WebSphere Business Services Fabric. The relationships between the Business Space Framework and the products in the WebSphere BPM is shown in Figure 1-3 on page 14.

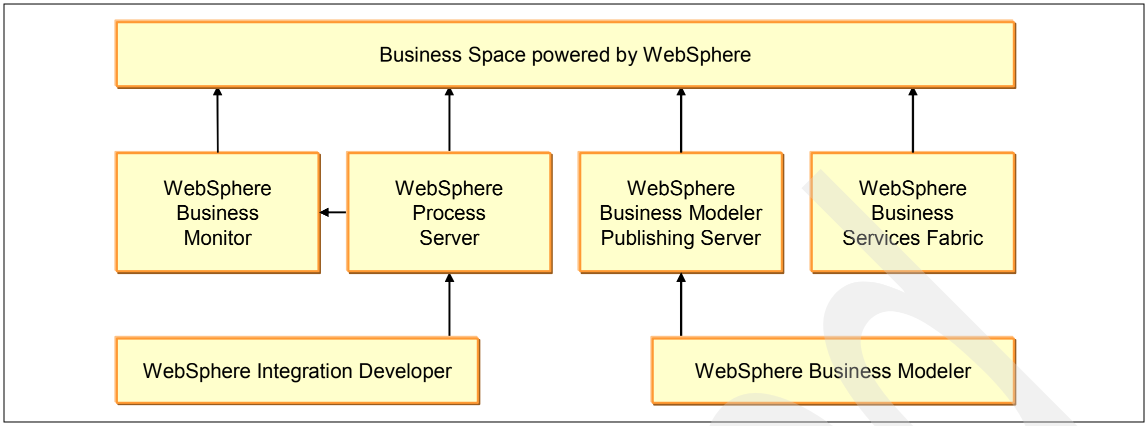


Figure 1-3 Business Space and BPM Products

The Business Space included with WebSphere Process Server contains Human Workflow widgets for business users to view all tasks that have been created. Users can create tasks and view Human Workflow diagrams that show the task's status. Business Space also includes a widget that works with business rules.

1.4 Network deployment concepts

This section defines the following concepts:

- "WebSphere Application Server Network Deployment components" on page 14
- "Clustering" on page 16
- "Load balancing" on page 18
- "Failover" on page 18

1.4.1 WebSphere Application Server Network Deployment components

WebSphere Application Server Network Deployment solutions are built from the following components:

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 that each host 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.

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 can be configured as follows:

- Stand-alone application

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

1.4.2 Clustering

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 utilize 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.

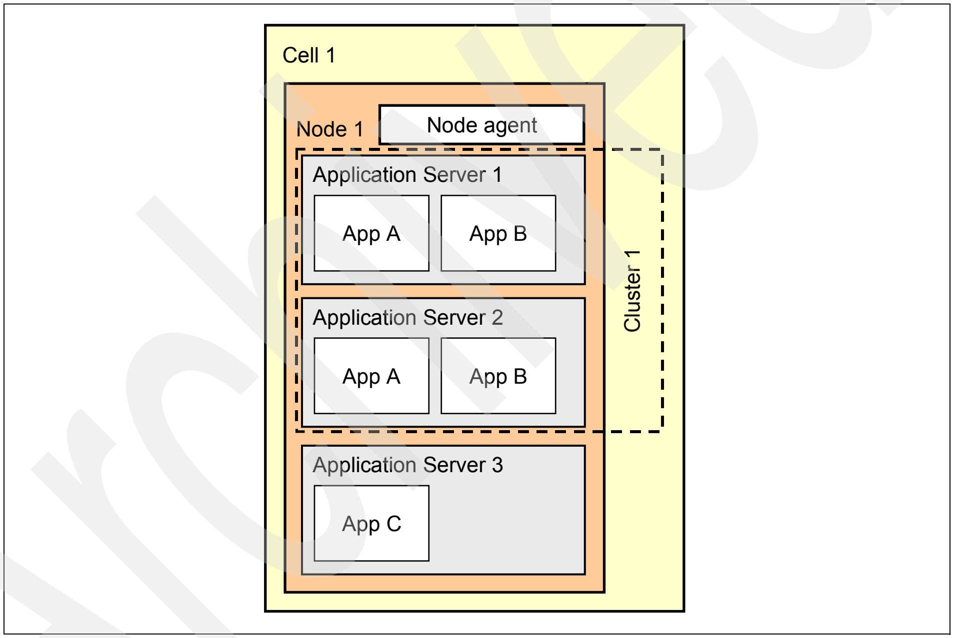


Figure 1-4 A vertically clustered WebSphere environment

Horizontal clustering

In a horizontal cluster, multiple application servers are distributed across nodes in order to utilize more physical resource (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 underpinning of that member's node fails.

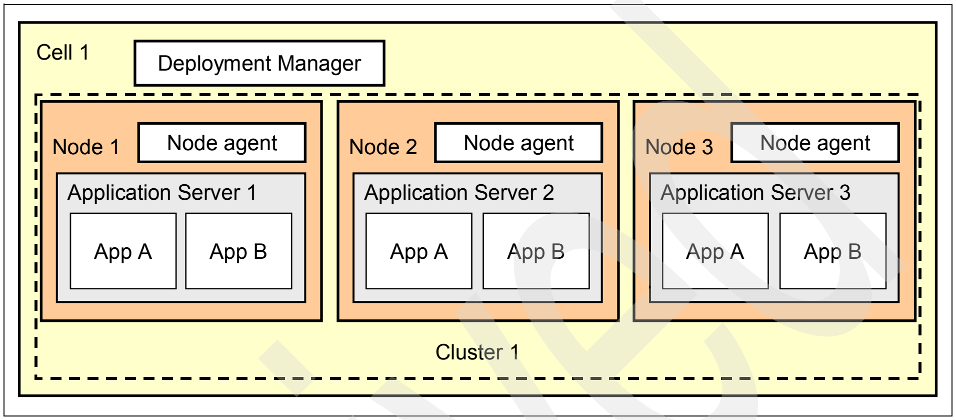


Figure 1-5 A horizontally clustered WebSphere environment

1.4.3 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 being overloaded (Figure 1-6).

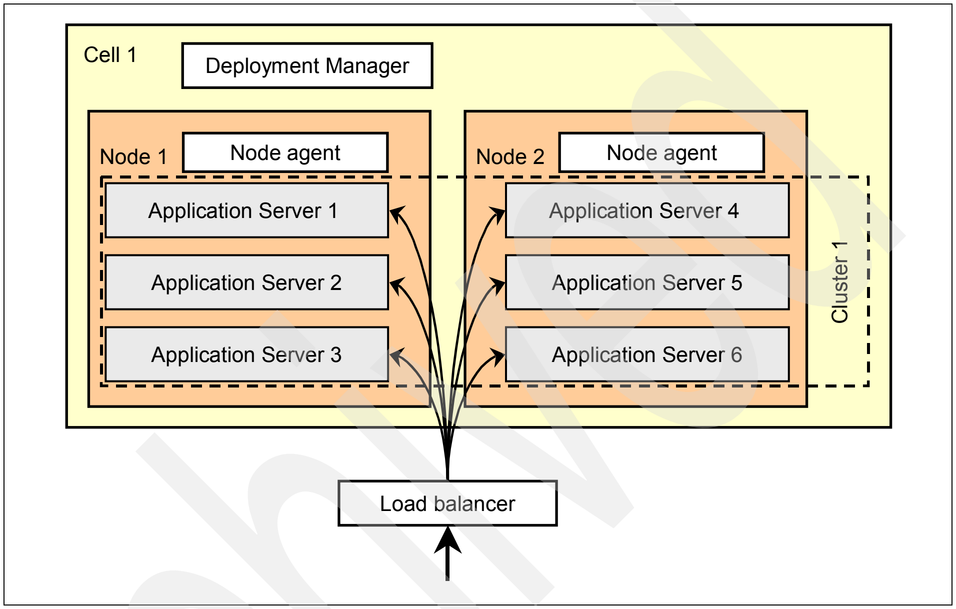


Figure 1-6 A load-balanced WebSphere environment

1.4.4 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 on page 19. 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.

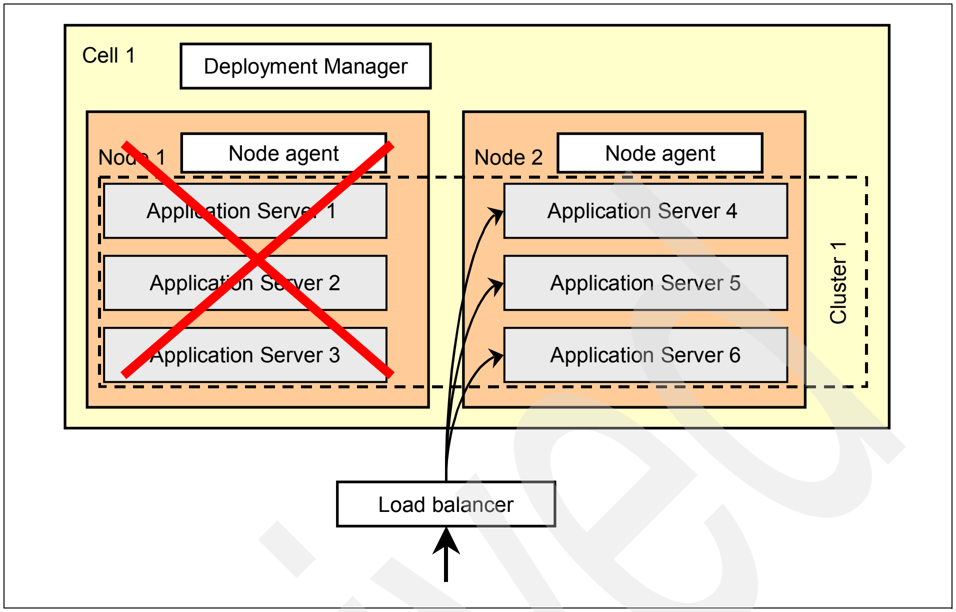


Figure 1-7 Failover in a clustered WebSphere environment

2

Security considerations for BPM

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

2.1 Security in WebSphere Application Server

WebSphere Application Server is the foundation on which WebSphere Process Server is built. This section discusses WebSphere Application Server security considerations that are pertinent to a WebSphere Process Server environment. It contains the following sections:

- "Overview of security provided by WebSphere Application Server" on page 22
- "Application security" on page 23
- "Administrative security" on page 24
- "Java 2 security" on page 28
- "Operating System security" on page 29

2.1.1 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 is broken down to the four components shown in Figure 2-1.

- Application Security
- Administrative Security
- Java 2 Security
- Operating System Security

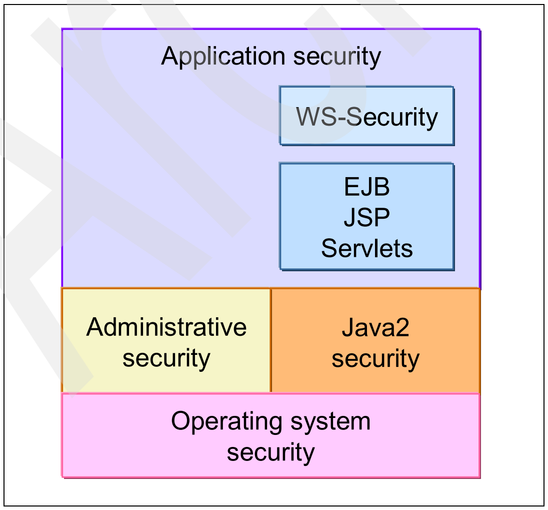


Figure 2-1 WebSphere Application Server security components

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

2.1.2 Application security

Application security provides application isolation and requirements for authenticating users and controlling their access to the applications in your environment.

Application security has to 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 need to 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 right 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 (MDB) and have configured them to use runAs roles. The runAs role is an identity assertion and the MDB will always run as the user you have mapped to the role. An authentication alias is an artifact used to define a user which will be mapped to a runAs role.

It is suggested to require a document from the application team, such as shown in Table 2-1 on page 24, which will list out and describe the roles for the application. Have a column to enter in the actual user or group that you will assign during deployment.

Table 2-1 Sample table of roles and group/user mapping: Retail / Banking application

Description of Role	Security Role	runAs Role	Administrator assigned users or groups
Access to functions for only 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

2.1.3 Administrative security

Administrative security represents the security configuration which 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 local operating system.

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 needs to 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 systems. 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 2-2 on page 26 illustrates a trust association between WebSEAL and WebSphere Process Server that is established using SSL.

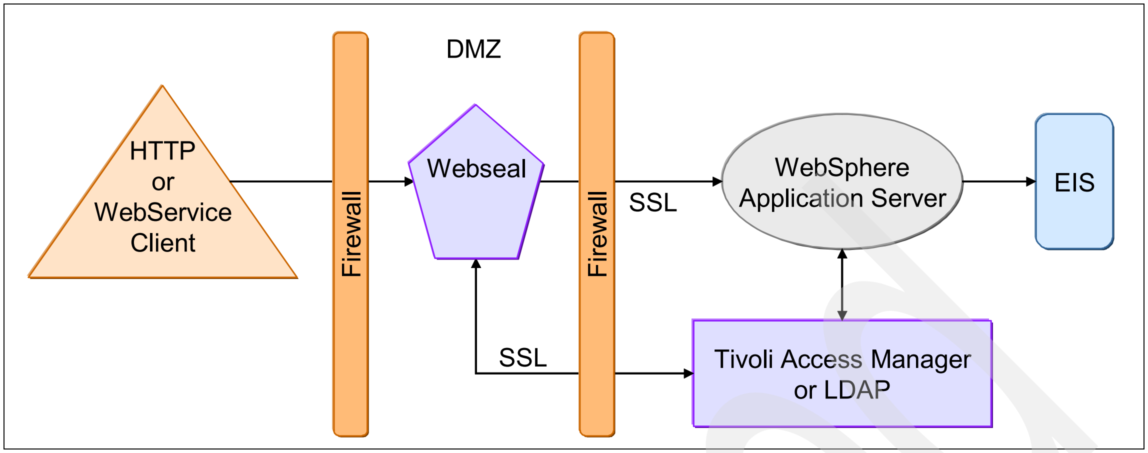


Figure 2-2 Single sign-on

> **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. For example, in Figure 2-2, Enterprise Information System should only have a self-signed certificate in its keystore and only WebSphere Application Server should have Enterprise Information System's signer certificate. This limits Enterprise Information System's client connections.

If the target Enterprise Information System (EIS) 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, pre-configured 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 for only the sender and receiver to 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 guarantying the confidentiality of the data.

- Integrity

  Integrity is the desire guaranteed 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 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 their environment.

Service integration bus

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. In section 7.1.5, “Configuring the Service integration bus” on page 187, you will review the bus security configuration 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 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.

2.1.4 Java 2 security

Java 2 security provides a policy-based, fine-grain 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 need to 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.express.doc/info/exp/ae/csec_rsecmgr2.html

2.1.5 Operating System security

You do not want your operating system compromised. You should install and run WebSphere Application Server as a non root user. This user should be part of a group that has permissions to the file system that that has permissions to the local file system.

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`

2.2 Security for a WebSphere Process Server solution

This section addresses security considerations for WebSphere Process Server. It contains the following sections:

- "Overview of business integration security" on page 29
- "Access control for SCA container" on page 32
- "Access control for Business Process Choreographer container" on page 33
- "Access control for Common Event Infrastructure container" on page 36
- "Securing SCA modules" on page 37
- "People resolution and directories" on page 39

2.2.1 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 2-3 on page 30.

- SCA modules
- SCA runtime

BPC adds BPC runtime component to the application security component of WebSphere Application Server as shown in Figure 2-3

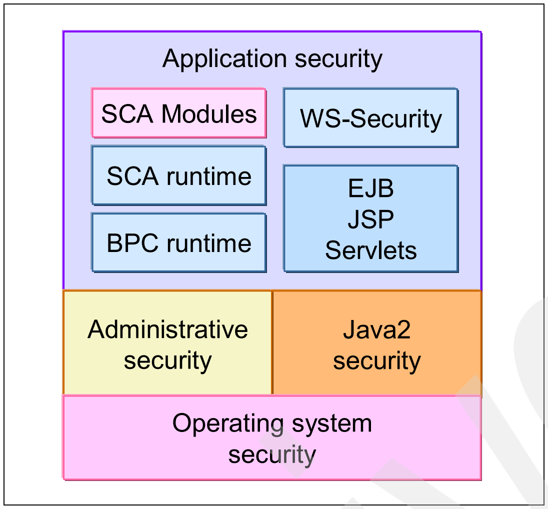


Figure 2-3 WebSphere Process Server security components

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 needs to be protected with operating system security.

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.

- **Security → Business Integration Security**
- **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 you can understand what access you will want to grant to certain groups in your organization. These roles are provided to you in the following sections:

- Section 2.2.2, “Access control for SCA container” on page 32
- Section 2.2.3, “Access control for Business Process Choreographer container” on page 33
- Section 2.2.4, “Access control for Common Event Infrastructure container” on page 36

Once you understand these roles, you will want to build a table just like the one you require from development shown in Section 2.2.2, “Access control for SCA container” on page 32. You will use this table in Section 7.1.7, “Mapping groups to the business integration containers and supporting applications” on page 193.

2.2.2 Access control for SCA container

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

Table 2-2 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	SCA Application Bus ME data source authentication alias	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 2-3.

Table 2-3 Failed Event Manger roles

Application Name	Security Role	Description	Notes
wpsFEMgr_6.1.2	WBIOperator	Everyone	Users assigned to this role have adminstrator privileges. This role is also referred to as the system administrator for Failed Event manager

2.2.3 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 2-4, are set up during creation of the deployment environment.

Table 2-4 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 need to have users and groups assigned (Table 2-5). At minimum, all of the *APIUser roles should be All Authenticated. You may wish to restrict this even more based on what the development staff have created with these APIs.

Table 2-5 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
	WebClientUser	All Authenticated users	
	JMSAPIUser	All Authenticated users	Users assigned to this role can access business flow manager message -driven bean to process asynchronous API calls.

Application Name	Security Role	Default permission	Notes
TaskContainer_<deploymentEnvironment.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 business flow manager and 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
	EscalationUser	All Authenticated users	Users assigned to this role can access human task manager message -driven bean to process asynchronous API calls.
BPCExplorer_<deploymentEnvironment.cluster>	WebClientUser	All authenticated users	Users assigned to this role can use the Business Process Choreographer Explorer
BPCObserver_<deploymentEnvironment.cluster>	ObserverUser	All authenticated users	Users assigned to this role can use the Business Process Choreographer Observer
BusinessSpaceManager	administrator	All authenticated users	Users assigned to this role can administer business space manager

Application Name	Security Role	Default permission	Notes
BusinessRulesManager_<deploymentEnvironment.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

2.2.4 Access control for Common Event Infrastructure container

Common Event Infrastructure (CEI) runtime uses container-managed aliases to authenticate with the bus and datastore. These authentication aliases, shown in Table 2-6, 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 2-6 Common Event Infrastructure Authentication Aliases

Authentication Alias	Description	Notes
CommonEventInfrastructureJMSAuthAlias	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 need to have users and groups assigned shown in Table 2-7 on page 37. For greater detail on uses for each role, refer to the Information Center article *Securing accessing to Common Event Infrastructure functions*, available at the following Web page:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.websphere.cei.612.doc/doc/ccei_security.html

Table 2-7 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

2.2.5 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 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 2-1, access to the one-way invoke method is restricted to users that are members of the role manager.

Example 2-1 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 Developer Works article *WebSphere Process Server security overview*, available at the following Web page:

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

2.2.6 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 on 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 virtual member manager.

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

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.websphere.bpc.612.doc/doc/bpc/cpeopledirectory_provider.html

For more on the overall usage of people directories refer to 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/v6r1mx/topic/com.ibm.websphere.bpc.612.doc/doc/bpc/c6bpel_auth_instance.html

2.3 Access control for WebSphere Business Services Fabric

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

2.3.1 Preparation

When installing and starting WebSphere Business Services Fabric, turn off Java 2 security. Refer to Section 10.4.8, "Configuring security" on page 339 and the Information Center article *Configuring security*, available at the following Web page:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.ws.fabric.install612.doc/fpi/task/t_%20configuring_security.html

2.3.2 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, shown in Table 2-8, are set up during configuration of the environment.

Table 2-8 WebSphere Business Services Fabric authentication aliases

Authentication Alias	Description	Notes
Fabric_DB2_Authalias	Fabric database authentication alias	User name and password for the four WebSphere Business Services Fabric databases. Business Services Repository, Governance manger, Performance manager and messaging engine
Fabric_Bus_AuthAlias	Fabric Bus Authentication Alias	User name and password for the WebSphere Business Services Fabric service integration bus. This username will need to be added to the following locations: ► Fabric bus's connector role, ► Fabric activationSpecs – Hub Event Activation – Hub Request Activation – DAPerfMon Activation ► DAEventConnectionFactory

The WebSphere Business Services Fabric installation pre-populates the six groups to the Fabric Tools application shown in Table 2-9 on page 42. 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 2-9 WebSphere Business Services Fabric security roles: FabricTools andFabric 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 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, 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 a 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.
FabricSubscriberManager	Group provided at installation FabricSubscriberManagers	The FabricSubscriberManager has full access to the Subscriber Manager enabling them to perform all required subscriber management functions.

Security Role	Default permission	Notes
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 needs to log into the WebSphere Business Services Fabric for z/OS using a browser.

2.4 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 2-10, are set up during configuration of the environment.

Table 2-10 WebSphere Business Monitor authentication aliases

Authentication Alias	Description	Notes
MonitorAlphabloxAlias	Authentication for MonitorAlphabloxAlias.	User name and password for AlphaBlox®
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 2-11 indicates the roles and the actions that can be completed for each role:

Table 2-11 WebSphere Business MonitorData Security Roles

Roles	Notes	URL
Business-Manager	This role provides basic read-only access to public (shared) KPIs within a resource group.	/models/*
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	► /models/*/kpis/* ► /models/*
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.	► /models/*/kpis/* ► /models/*
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.	► /models/*/kpis/* ► /models/*/version/*/kpis/* ► /models/*
SuperUser		All URIs without restrictions

The roles referenced in Table 2-12 are used for WebSphere Business Monitor dashboards. These roles encompass access to AlphaBlox, REST APIs and Business Space.

Table 2-12 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

2.5 Additional security considerations

This section addresses the following security considerations:

- Creating a secured link between two cells
- Ideas on to make security administration a little easier

2.5.1 Creating a secured link between two cells

You may find in business integration systems that you 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. 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 with a secured channel. So you will need to configure SSL so that the consuming cell has the signer certificate of the producing cell. If this is bidirectional, then you will need to 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_sslmanageltpakeys.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 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 be looked up from a properties file which will provide more flexibility, but it 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 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. Steps to create 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 2-2.

Example 2-2 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 needs to 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 will need to create the same configuration changes in the WPSCell02 cell. This will allow both cells to send messages back an forth to both SCA modules.

1. Define a foriegnBus in your cell (Example 2-3).

   Example 2-3 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 2-4).

   Example 2-4 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 2-5).

   Example 2-5 Add user to bus connector role

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

4. Create a Service Bus Link (Example 2-6).

Example 2-6 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 2-7).

Example 2-7 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 remote queue (Example 2-8).

Example 2-8 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 local queue () Example 2-9.

Example 2-9 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 elaborate explanation of configuring SCA cross cell review the article *Instructions for configuring SCA cross-cell communications* available from the following Web page:

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

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

2.5.2 Ideas on to make security administration a little easier

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 your 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 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 will need to manage.

If security was simple, the system could be easily compromised. There are considerations that can make the administrator's job easier. However they might reduce your flexibility. There are comprises that you have to weigh 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. For instructions on changing password without outages, refer to section 7.2.4, "Changing the database password" on page 222. These same instructions work for bus runAs role user IDs as well. If you follow this methodology, make sure the second ID is also assigned to the role. (for example, bus connector role, and so forth)

Using groups for infrastructure IDs

You may decide to have a different user for each messaging engine. If so, you need to 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, 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 will reduce the number of locations 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.

2.6 Populating the security registry

In the environment we create in this Redbooks publication, we use Tivoli Directory Server as the LDAP server. The LDAP server can be populated by importing an LDIF file. We have included the LDIF file we used for our environment in the additional materials supplied with this book, in Appendix A, "Additional material" on page 449. The LDIF file we used is ldap_itso.ldif and is supplied in the LDAP_config directory of the additional material. It defines the System group with the users shown in Table 2-13

Table 2-13 This group will have system privileges

Group	User	Password	Description
System	sca	passw0rd	This is the user assigned to the Authentication alias for Common Event Infrastructure, Business Process Choreographer, and Service Component Architecture messaging.
	escalation	passw0rd	This is the user assigned to EscalationUser for the runAs role for the Human Task Manager message-driven bean.
	jmsapi	passw0rd	This is the user assigned to JMSAPIUser for the runAs role for the Business Flow Manager message-driven bean.

The groups in Table 2-14 and Table 2-15 on page 52 are used to administer different aspects of your BPM environment

Table 2-14 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	
	kevin	passw0rd	
	tom	passw0rd	
monadmin	monitor	passw0rd	This group is the WebSphere Business Monitor Administrators. This group needs to be mapped when installing the WebSphere Business Monitor
	mohamed	passw0rd	
fabadmin	fabric	passw0rd	This group is the WebSphere Business Services Fabric Administrators. This group needs to be mapped when installing the WebSphere Business Services Fabric
	vignesh	passw0rd	

Table 2-15 System users

Group	User	Password	Description
wpsusers	ryan	passw0rd	This group is the WebSphere Process Server operators
	mohamed	passw0rd	
	jeff	passw0rd	
monuser	tom	passw0rd	This group is the WebSphere Business Monitor users
	jeff	passw0rd	
	mohamed	passw0rd	
wpscfg	vignesh	passw0rd	This group is the WebSphere Process Server configurators
	peter	passw0rd	
	julia	passw0rd	

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 on 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. This includes the following factors:

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

To make the topology design and implementation process easier, WebSphere Process Server V6.1.2 includes a set of deployment environment patterns that represent the most common production topologies.

Using the deployment patterns to create your environment represents a dramatic improvement over the deployment process in WebSphere Process Server V6.0.2, where the entire installation had to be done manually or with scripts. However, manual deployment (through the administrative console) or a scripted install is still possible in V6.1.2. 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

A number of different components are created and used during WebSphere Process Server deployment environment generation. These components are discussed in this section.

- “Databases” on page 55
- “Service integration buses” on page 56
- “Business Process Choreographer” on page 56
- “WebSphere Process Server applications” on page 57

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 needs to be created prior to starting WebSphere Process Server. The common database persists 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 needs to 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 persists information regarding the Event Service, such as Common Based Events and key performance indicators (KPIs).

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. 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 provides services to run business processes within an application server.

- Human Task Manager

  This component provides services to run human tasks within an application server.

3.2.4 WebSphere Process Server applications

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

- BPC Explorer

  This tool implements a generic user interface for interacting with business processes and human tasks. It is typically used to initiate and test business processes.

- BPC Observer

  This tool creates reports on processes that have been completed. It displays the status of running processes.

- Business rules manager (BRM)

  This tool assists business analysts in browsing and modifying business rule values.

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

CEI is an embeddable technology intended to provide basic event management services to applications that require those services.

For service component event points that you monitor, events can be published to the CEI server and stored in the CEI server database.

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 our customers require. Each pattern centers around the number of WebSphere Process Server clusters and cluster members.

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 as follows:

- 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

  Supporting infrastructure includes the CEI and other infrastructure services used to support your environment, such as the Business Process Choreographer Observer, 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 V6.1.2 are as follows:

- 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 Section 3.3.1, “Single Cluster topology pattern” on page 60.

- 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) and one for the messaging infrastructure (named Messaging). This pattern is discussed in Section3.3.2, “Remote Messaging topology pattern” on page 62.

- 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 Section 3.3.3, “Remote Messaging and Remote Support topology pattern” on page 65.

- Custom deployment environments

  If none of the IBM supplied deployment environment patterns meets your requirements, you may create a custom deployment environment. This pattern is discussed in Section 3.3.4, “Custom topology patterns” on page 67.

Regardless of the type of pattern you use, generating a deployment environment on the administrative console creates an XML-based representation of your topology that can be exported and 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 pre-production environments.

Making changes:

- After generation, you are not allowed to make a change to the deployment environment definition and re-generate the deployment environment. You need to start from the beginning if you need to do so.
- Any changes made to a specific resource after generation (for example, a data source) will not be reflected in the deployment environment descriptor.

There are several methods 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 administrative 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** command line utility.

Option three was used to create the topology used in the lab environment for this Redbooks publication. Regardless of which method you use to create the deployment environment, you can still manage some aspects of the deployment environment through the administrative console. (For example, add more nodes to the deployment environment).

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.

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

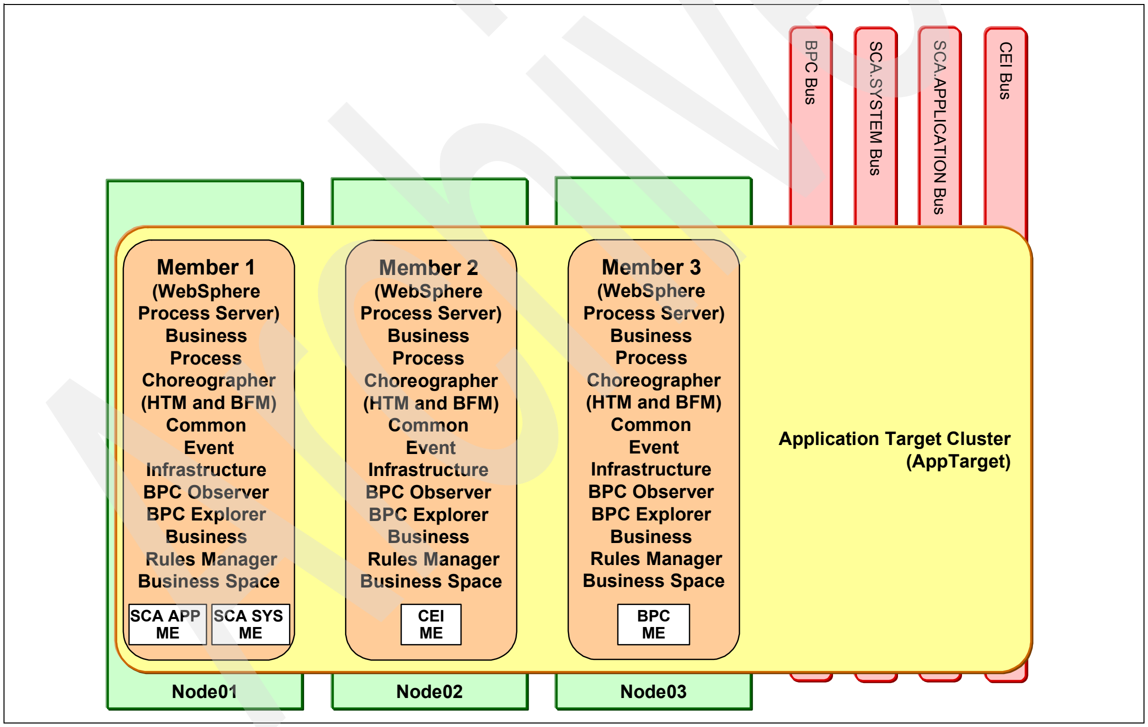


Figure 3-1 Example Single Cluster topology

Note the following aspects of this example:

- All of the components are configured in a single cluster which 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 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 Explorer
  - BPC Observer
  - Business Rules Manager
  - CEI
  - Business Space
- Each cluster member is an application deployment target.
- In Figure 3-1 on page 60, 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. This configuration is discussed in Chapter 8, "Advanced production topologies" on page 229. 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 Single Cluster topology is different than when the messaging engines are in a remote cluster. When the messaging engines and the applications are co-located, the default behavior is for message producers and consumers to always use a local active messaging engine (if one is available). For example, assume you have two applications deployed to each cluster member that need 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.

Read and write local 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 on page 63.

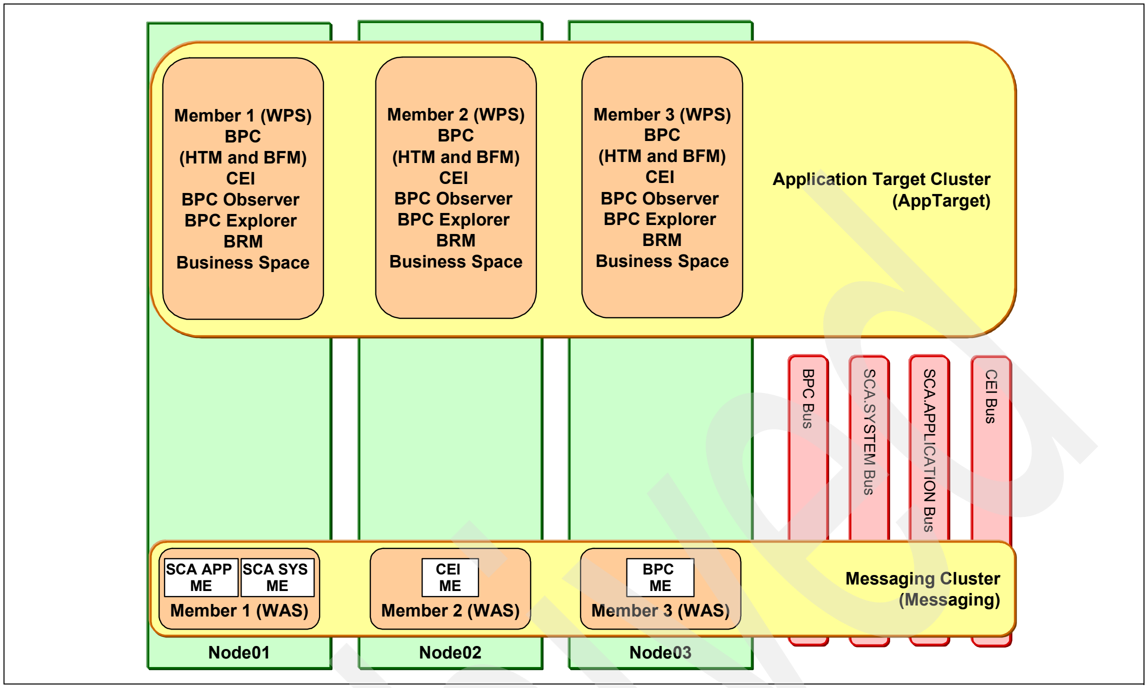


Figure 3-2 Example Remote Messaging topology

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 Explorer
  - BPC Observer
  - Business Rules Manager
  - CEI
  - Business Space

- In Figure 3-2 on page 63, 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. This configuration is discussed in Chapter 8, “Advanced production topologies” on page 229. 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 topology is different than 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.

This behavior creates issues if you attempt to partition the destinations in the remote messaging cluster. 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 member of the Messaging cluster. However, this configuration can create issues for your applications.

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 you partition destinations you lose 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 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 5, “Configuring a Remote Messaging and Remote Support topology” on page 89. A Remote Messaging and Remote Support sample topology is shown in Figure 3-3.

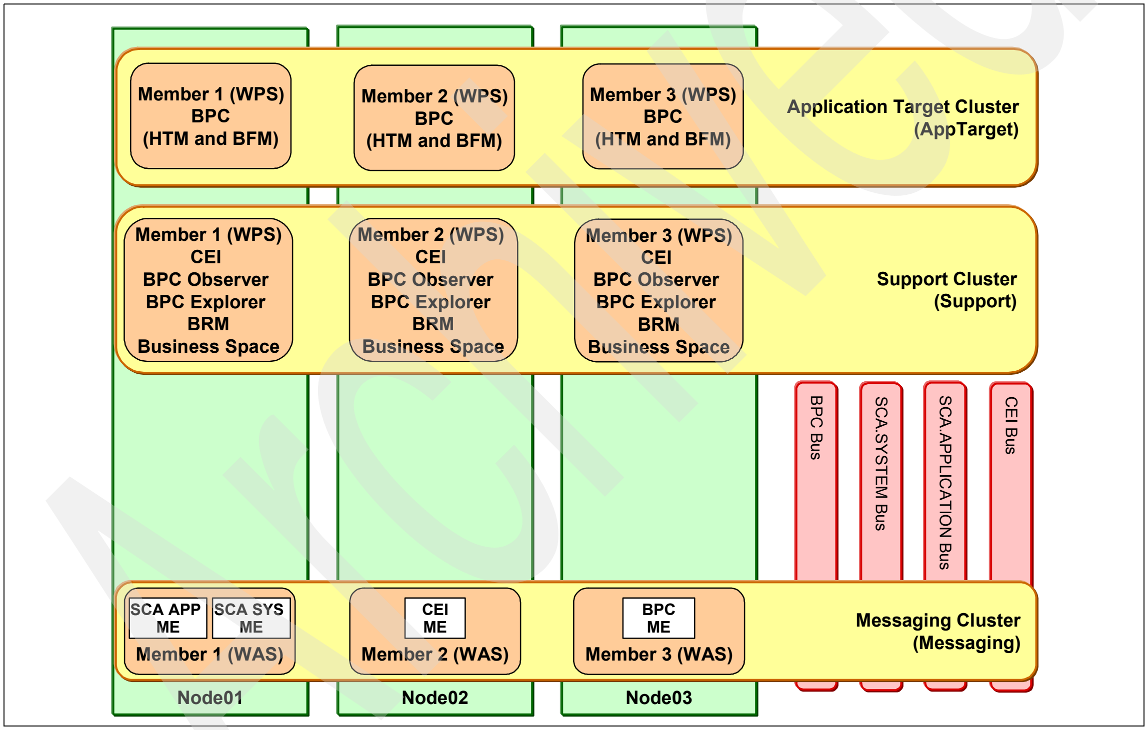


Figure 3-3 Example Remote Messaging and Remote Support topology

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 Explorer
  - BPC Observer
  - 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 65.
  - 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. This configuration is discussed in Chapter 8, "Advanced production topologies" on page 229. 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 you partition destinations you lose 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 is suitable to your needs, you can create a custom topology. As well, you can use the administrative console to manually deploy the environment in any way 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 than the process for using the default topology patterns. Using custom topologies is discussed in Chapter 6, "Configuring a custom topology" on page 165. 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 the BPC Observer

  The BPC Observer client is a reporting tool designed to provide basic statistical information about your business processes using events generated by the CEI. If you have a more robust monitoring tool such as WebSphere Business Monitor, deploying the Observer application may be an unnecessary use of resources.

- 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 8, "Advanced production topologies" on page 229 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, please 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 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 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.

3.4.1 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) is 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 will place 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.

3.4.2 Remote Messaging topology pattern

For environments where there are 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 lessens 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 straight forward 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.

3.4.3 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, does create 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 will also add the administrative burden associated with creating and maintaining policies.

From a scalability standpoint, the Remote Messaging and Remote Support topology provides the most flexibility. Because each of the distinct functions within WebSphere Process Server is divided amongst 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 provides 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.

3.4.4 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.

3.4.5 Condensed topology selection criteria

Table 3-1 on page 72 provides a condensed list of the advantages and disadvantages to each of the topology patterns. Consider the information listed in Table 3-1 on page 72 a quick guide to selecting your production topology.

Table 3-1 Topology selection considerations

Consideration	Single Cluster topology	Remote Messaging topology	Remote Messaging and Remote Support topology
Number of clusters to maintain	1 cluster for all components	1 cluster for applications and for the support infrastructure 1 cluster for messaging	1 cluster for applications 1 cluster for the support infrastructure 1 cluster for messaging
Hardware requirements	Can be implemented on limited hardware	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
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
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
Administrative burden	Relatively small	Requires additional effort	Required most administrative effort
Scalability	Easiest to scale but all components are scaled at the same rate	Messaging cluster scalability is limited (no benefit beyond three servers) All other components are scaled at the same rate	Easiest to scale. All functions are separated Messaging cluster scalability is still limited (no benefit beyond three servers)

3.5 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 applications in the WebSphere business integration product portfolio. 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, 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.

3.5.1 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 3-4 on page 74.

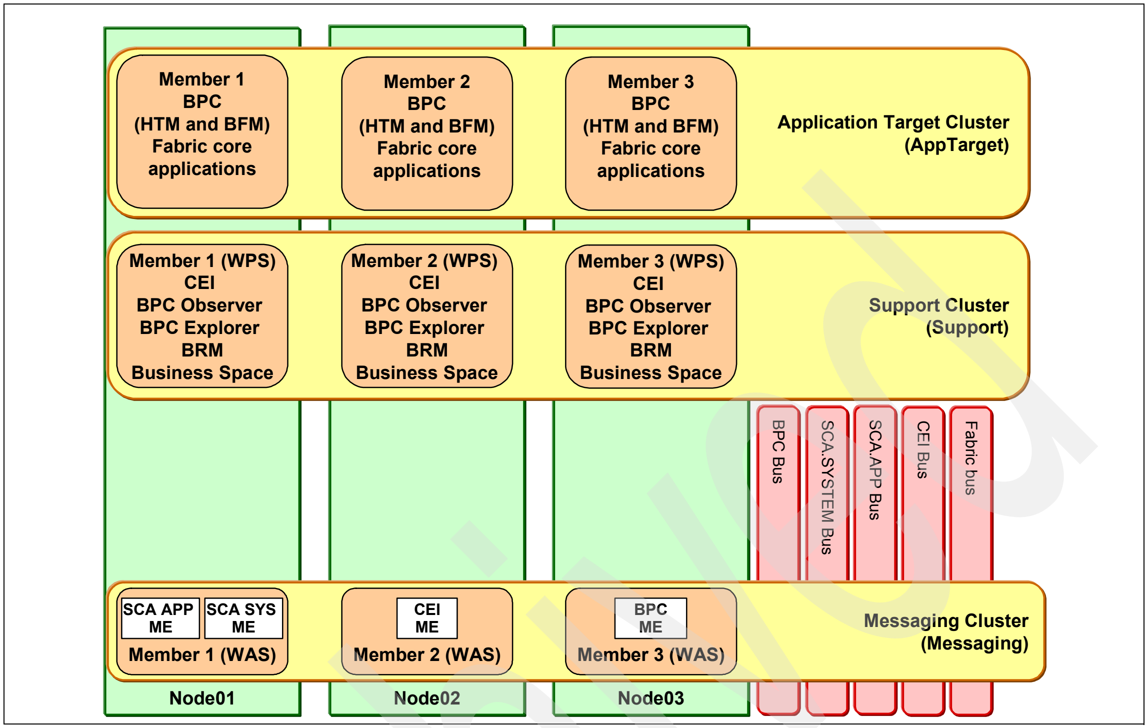


Figure 3-4 WebSphere Business Services Fabric deployment scenario

Chapter 10, “Incorporating WebSphere Business Services Fabric into a production topology” on page 319 provides detailed instructions on how to add WebSphere Business Services Fabric to the WebSphere Process Server Remote Messaging and Remote Support topology.

3.5.2 WebSphere Business Monitor

The recommended production topology for installing WebSphere Business Monitor and WebSphere Process Server is to deploy both products in the same cell. Doing so allows both products to share the CEI (also referred to as local CEI). When WebSphere Business Monitor is added to the WebSphere Process Server Remote Messaging and Remote Support topology, the messaging cluster is a member of the WebSphere Business Monitor bus. This topology is represented in Figure 3-5 on page 75.

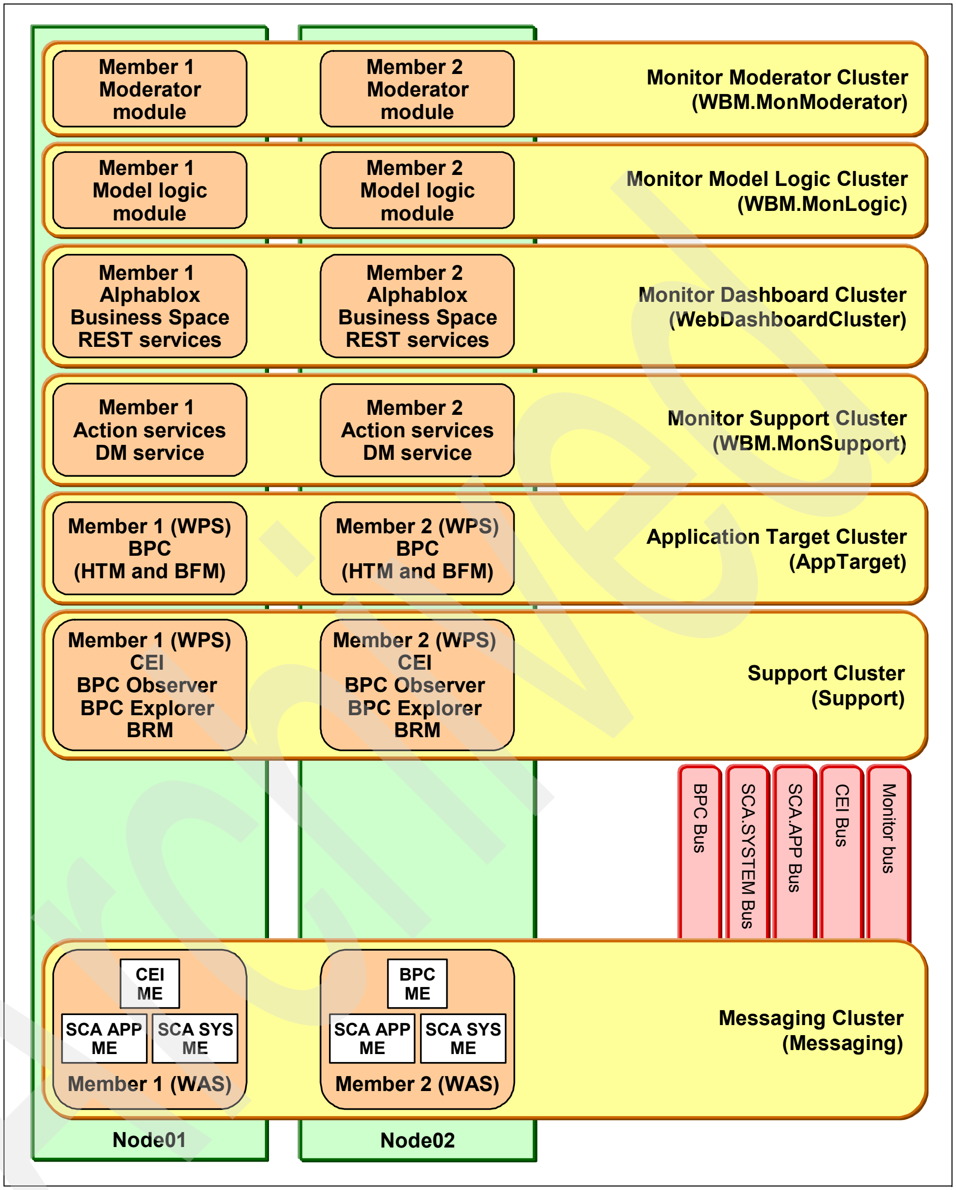


Figure 3-5 WebSphere Business Monitor deployment scenario

For detailed information about configuring WebSphere Business Monitor and WebSphere Process Server in the same cell, see Chapter 11, “Incorporating WebSphere Business Monitor into a production topology” on page 359.

4

Business scenario used in this book

This chapter introduces the business scenario used in this book. The scenario is a vehicle loan processing application at a fictional company called ITSOBank.

This chapter contains the following sections:

4.1 Introduction

The business scenario used in this Redbooks publication is a vehicle loan process of a fictional organization called ITSOBank. The goal of this business process is to collect and analyze a loan applicant's information and provide a suitable loan customized to the customer.

Using WebSphere Process Server or WebSphere Business Services Fabric and WebSphere Integration Developer, in combination with WebSphere Business Modeler and WebSphere Business Monitor. this process can leverage the full cycle of IBM Business Process Management (BPM).

The process is modeled with WebSphere Business Modeler, developed with WebSphere Integration Developer, and deployed to WebSphere Process Server or WebSphere Business Services Fabric. WebSphere Business Monitor monitors system performance indicators and extract business metrics.

4.1.1 Overview of the vehicle loan 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.
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.
5. 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. Table 4-1 on page 79 provides sample interest rates based on the rating score and customer type.

Table 4-1 Interest Rate based on the Rating Score and Customer Type

Rating score	Customer type	Rate of Interest (%) per annum
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

4.2 WebSphere BPM cycle for the vehicle loan process

The vehicle loan process is implemented with the WebSphere BPM product set.

The WebSphere BPM product set provides end-to-end support for the implementation of BPM, including tools for process analysis, definition, execution, monitoring, and administration.

The WebSphere BPM product set used in the vehicle loan process includes the following products:

- WebSphere Business Modeler Advanced Version 6.1.2
- WebSphere Integration Developer Version 6.1.2 and Business Services Composition Studio
- WebSphere Process Server Version 6.1.2
- WebSphere Business Services Fabric Version 6.1.2
- WebSphere Business Monitor Version 6.1.2

Figure 4-1 shows how the WebSphere BPM products work together to provide end-to-end support for the implementation of business process management.

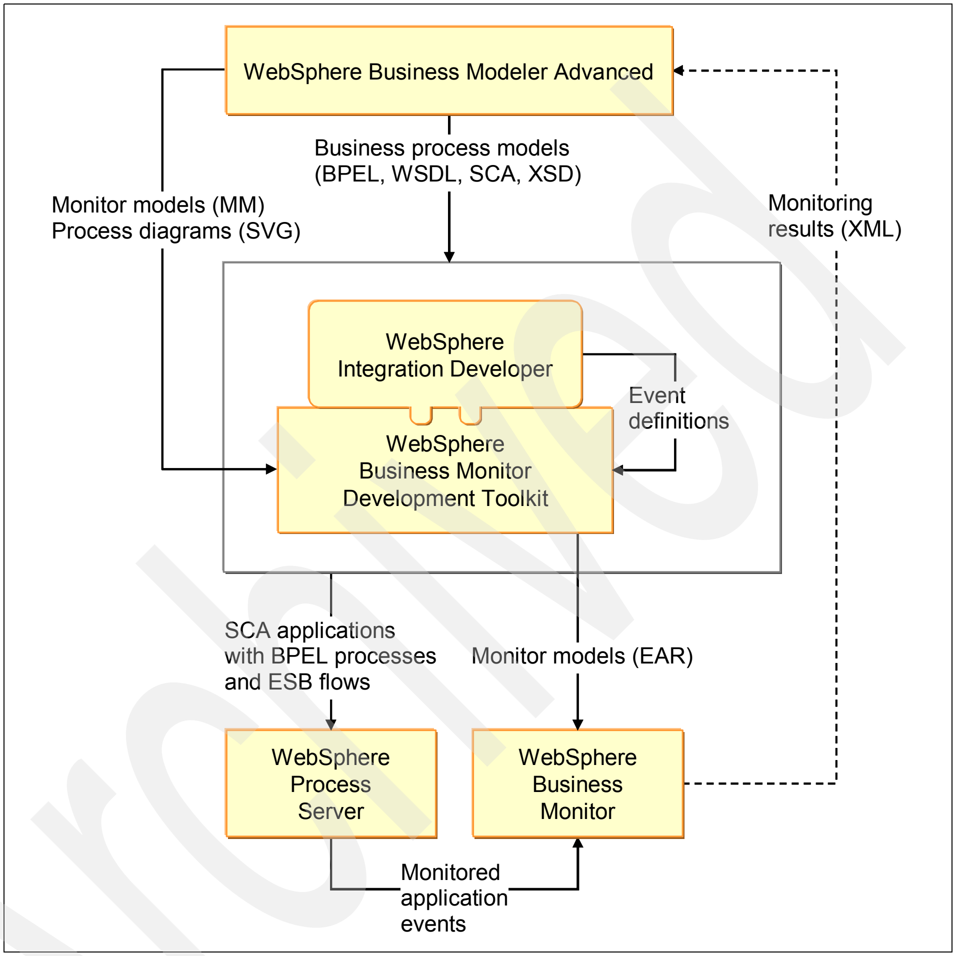


Figure 4-1 WebSphere BPM products interaction

4.3 Vehicle loan process implementations

This section describes how the vehicle loan process is implemented for the following runtimes:

- Vehicle loan process with WebSphere Process Server
- Vehicle loan process with WebSphere Business Services Fabric

The vehicle loan process implementations are included in the additional materials in Appendix A, “Additional material” on page 449.

The additional materials contains the directory \Scenarios. Within this directory are subdirectories for each runtime. Within these subdirectories are an Enterprise Archive (EAR) directory and a project interchange (PI) directory. The EAR directory contains a deployable version of the vehicle loan process for deployment to the appropriate runtime environment. The PI directory contains a project interchange file of the vehicle loan process which can be imported into WebSphere Integration Developer.

4.3.1 Vehicle loan process with WebSphere Process Server

The WebSphere BPM steps for the ITSOBank vehicle loan process is as follows:

1. A business analyst (non-technical person) 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. The vehicle loan process model is shown in Figure 4-2 on page 82.

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

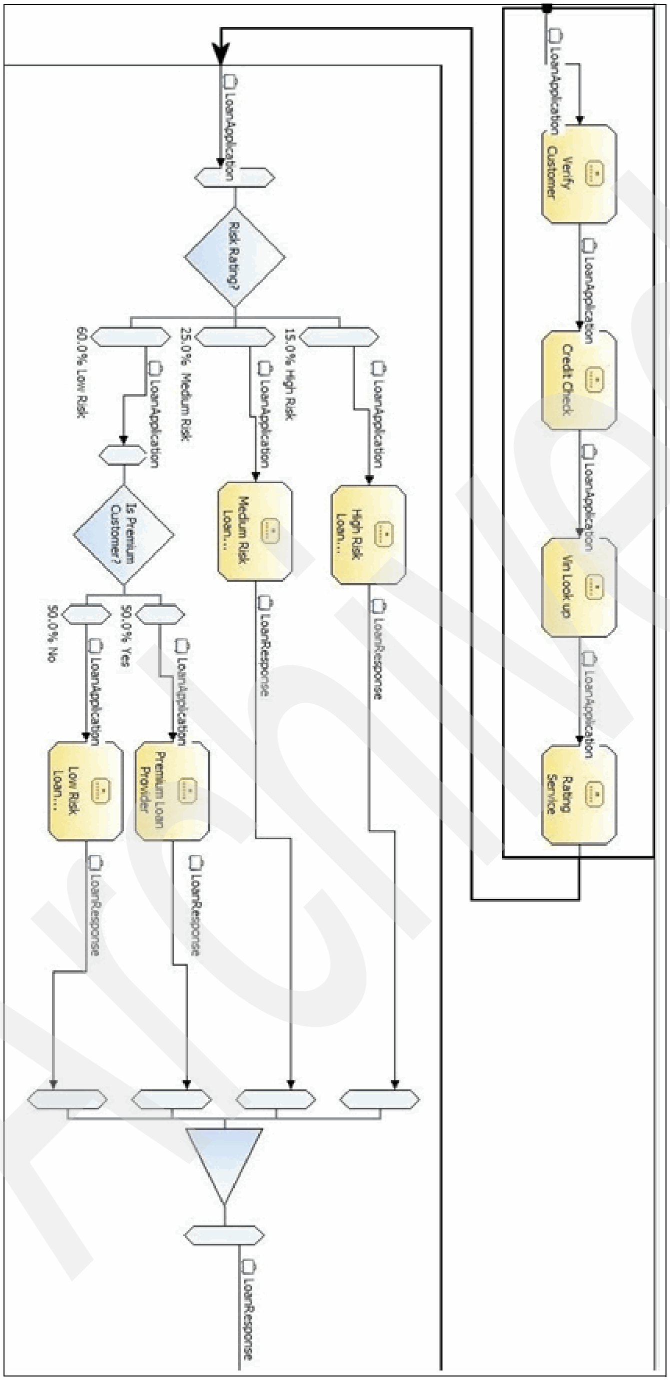


Figure 4-2 Vehicle loan process model

2. 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, VIN Lookup). These components are shown in the Service Component Architecture (SCA) assembly diagram (Figure 4-3).

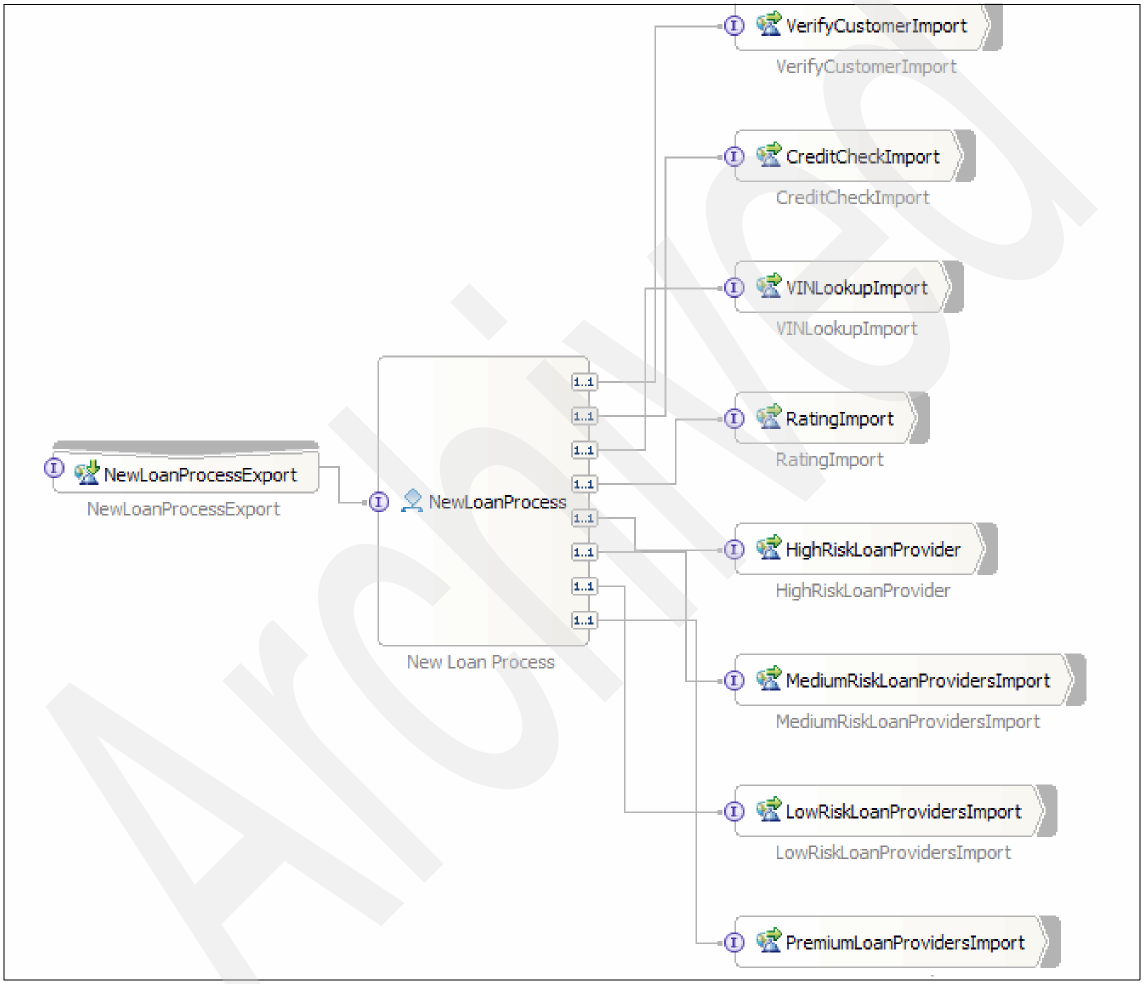


Figure 4-3 Vehicle loan process assembly diagram

3. The integration developer visually composes how the vehicle loan process should execute these reusable service components in a process flow, as shown in Figure 4-4.

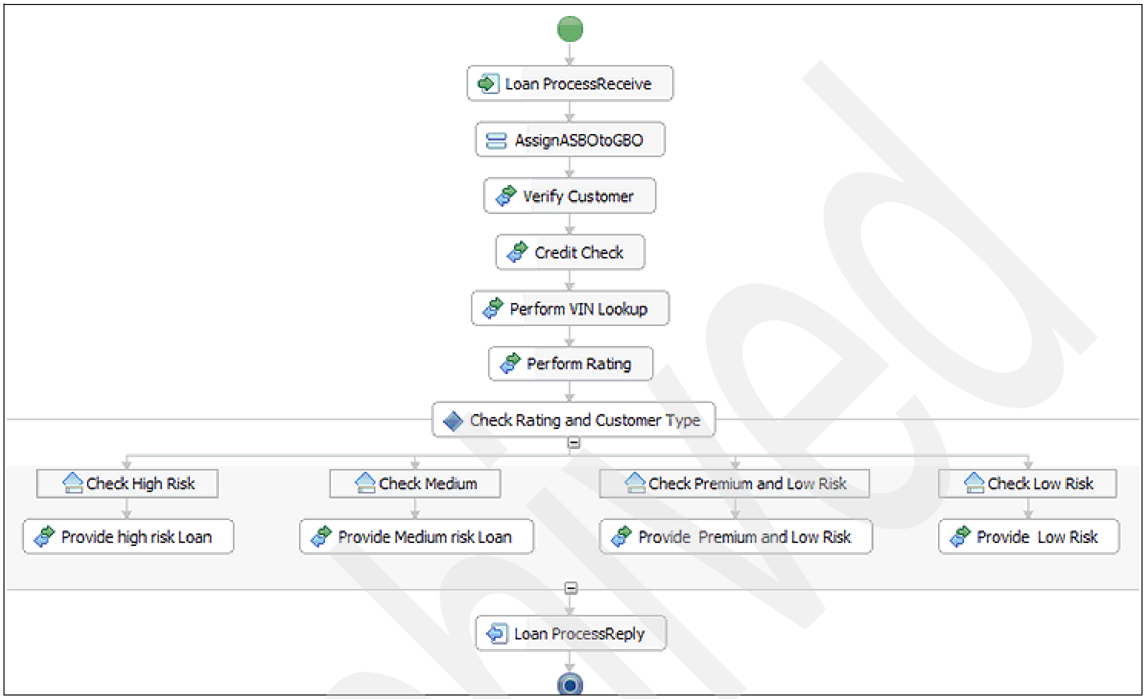


Figure 4-4 Vehicle loan process flow

4. The vehicle loan process application is assembled and packaged into an enterprise archive file for deployment to WebSphere Process Server.

4.3.2 Vehicle loan process with WebSphere Business Services Fabric

ITSOBank has also implemented the vehicle loan process for WebSphere Business Services Fabric. WebSphere Business Services Fabric introduces the concept of *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 consumers need.

In order to implement the vehicle loan process with WebSphere Business Services Fabric, the integration developer will use Business Services Composition Studio 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 4-5. Note the linear nature of this process flow in comparison to Figure 4-4 on page 84.

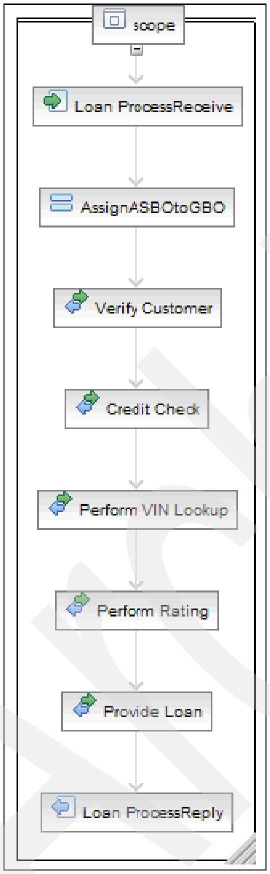


Figure 4-5 Vehicle Loan process for WebSphere Business Services Fabric

At runtime, the Business Services Dynamic Assembler finds the best suited loan provider endpoint for a consumer based on specific business requirements. An endpoint represents a location where a service can be invoked. Assertions are characteristics that describe the capabilities of an endpoint.

In the case of the vehicle loan process, the endpoint assertions that will determine the flow of the process are shown in Table 4-2.

Table 4-2 Endpoint assertions for the vehicle loan process dynamic assembly

Loan provider	Assertion
Low Risk Loan Provider	Rating Score Assertion with value Low
Medium Risk Loan Provider	Rating Score Assertion with value Medium
High Risk Loan Provider	Rating Score Assertion with Value High
Premium Loan Provider	Rating Score Assertion with Value Low and Customer Type Assertion with Value Premium

The vehicle loan process assembly diagram for WebSphere Business Services Fabric is shown in Figure 4-6. Note that it contains a component called LoanProviderDA, which is a dynamic assembler component.

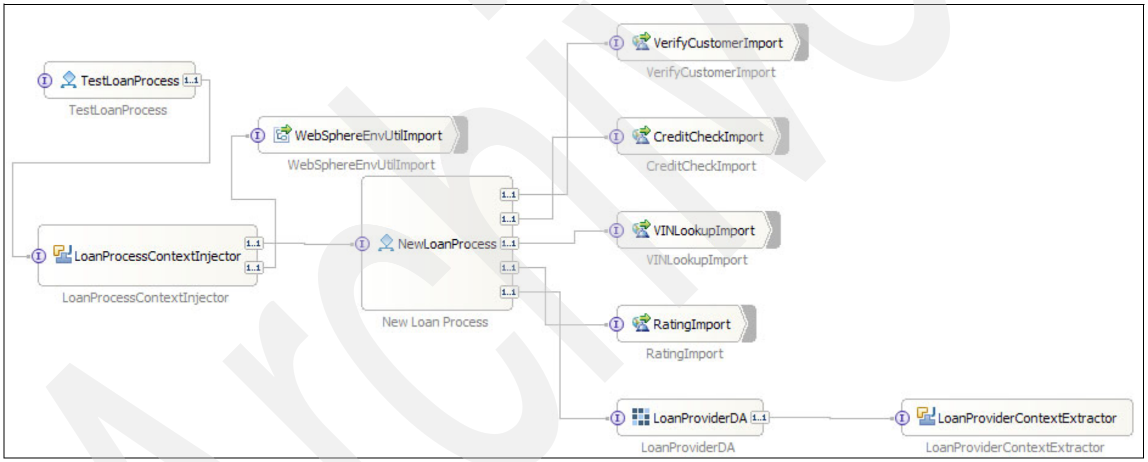


Figure 4-6 Vehicle Loan process Assembly Diagram for WebSphere Business Services Fabric

Part 2

Building production topologies for WebSphere Process Server

5

Configuring a Remote Messaging and Remote Support topology

This chapter provides full instructions for creating a gold topology, which uses the Remote Messaging and Remote Support topology pattern for WebSphere Process Server V6.1.2. In this topology we 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 BPC Observer.

These clusters are configured over two nodes and each node will have a single cluster member. Therefore, there are three clusters of two servers each.

Furthermore, we will create the topology using two distinct methods:

- Through the administrative console and template guided activities
- Using a silent installation using (UNIX) shell scripts.

5.1 Prerequisites to creating the topology

We will configure and deploy a near-production quality Remote Messaging and Remote Support topology. We will include a remote 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). The topology diagram is shown in Figure 5-1. We do not show how to make various components highly available using such technologies as HACMP™ or HADR.

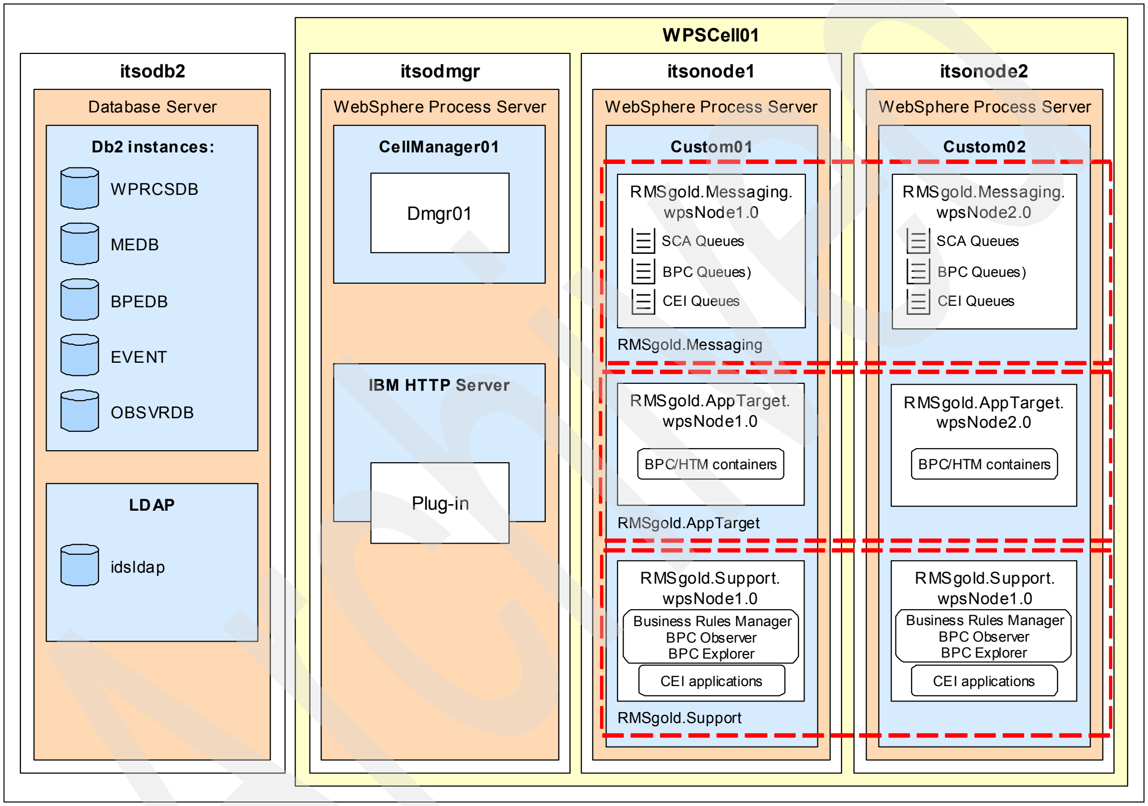


Figure 5-1 The Remote Messaging and Remote Support Topology to be built

A general overview of the stages required are:

- Install and configure DB2
- Install and configure LDAP
- Install WebSphere Process Server base product
- Apply critical fixes
- Create a deployment manager profile
- Create the node profiles
- Create a deployment environment
- Generate the environment
- Test and verify the topology

5.1.1 Software versions

To create a Remote Messaging and Remote Support topology for this chapter we used a number of Linux® systems and the following software installed onto then.

- SUSE® Linux Enterprise Server 10 SP1
- WebSphere Process Server V6.1.2
- IBM DB2 Universal Database™ V9.1
- IBM Tivoli Directory Server V6.0

5.1.2 Software installation

This section contains pointers to installing some of the necessary software.

Install and configure DB2

> **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/t0007312.htm`

The response file we used is given in Example 5-1 on page 92

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= 50001
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
```

Install and configure LDAP

Note: The installation and configuration of LDAP is described in Redbooks publication *Production Topologies for WebSphere Process Server and WebSphere ESB V6*, SG24-7413.

We have included the LDIF file we used for our environment in the additional materials supplied with this book in Appendix A, “Additional material” on page 449.

The LDIF file we used is ldap_itso.ldif and is supplied in the LDAP_config directory of the additional material.

Install the 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/v6r1mx/index.jsp?topic=/com.ibm.websphere.wps.612.doc/doc/iins_rf_wps.html

The response file we used is given in Example 5-2.

Example 5-2 Example WebSphere Process Server V6.1.2 response file

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

Install the update installer

> **Note:** Silent installation of the WebSphere Update Installer is covered in the Information Center article available at the following Web page:
>
> http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tins_updi_install.html

The response file we used is given in Example 5-3.

Example 5-3 Example Update Installer response file

```
-OPT silentInstallLicenseAcceptance="true"
-OPT allowNonRootSilentInstall="true"
-OPT disableOSPrereqChecking="true"
-OPT disableEarlyPrereqChecking="true"
-OPT installLocation="/opt/ibm/WebSphere/UpdateInstaller"
```

Apply critical fixes

> **Note:** We have chosen to patch the product before we create any profiles. Applying these critical fixes can be done either before or after profile creation.

In the base directory of the update installer (see the installLocation parameter in the response file in Example 5-3) create a file called install.txt with the contents shown in Example 5-4.

Extract the critical fixes to the maintenance folder and then run the update command:

```
./update.sh -options install.txt -silent
```

Example 5-4 Example fix installation file

```
-W maintenance.package="/opt/ibm/WebSphere/UpdateInstaller/maintenance"
-OPT disableNonBlockingPrereqChecking="true"
-W product.location="/opt/ibm/WebSphere/ProcServer"
-W update.type="install"
```

Install IBM HTTP Server

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

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

The response file we used is given in Example 5-5.

Example 5-5 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"
```

5.1.3 Create the databases within DB2

In this section we create the databases before we begin the process of creating the topology. We do this because the databases are remote from the WebSphere Process Server farm in production topologies and the work of creating these databases is normally carried out by the database team.

Following common practice, we create a database with one name and a schema within that database with different name. These names are listed in Table 5-1 on page 96.

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.

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, 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 from the deployment manager system (itsodmgr in our environment) to the DB2 system (itsodb2 in our environment). These scripts are located in <install_root>/dbscripts on the deployment manager. For example, in Linux they can be found in /opt/ibm/WebSphere/ProcServer/dbscripts. We suggest you copy the whole folder to the DB2 system under the instance owner.

We have chosen to create the databases before profile creation. Consequently, we need to edit some of the files 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.

There is one other database to create (the CEI Event database) but this must be populated after the topology is generated.

5.1.4 Create the common database

We will now create the common database on the DB2 system using the scripts that we have copied over from the deployment manager. We use the command line interface to DB2.

1. Login to the DB2 system as the instance owner for the common database as given in Table 5-1 on page 96.
2. Change to the dbscripts folder: cd ~/dbscripts/CommonDB/DB2

   We now need to make some adjustments to the following files:

   – configCommonDB.sh
   – createDBTables.sh
   – insertTable_CommonDB.sql
3. 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 96. In our example they are WPRCSDB and db2inst1 respectively. The modified text is shown in Example 5-6.

Example 5-6 Corrected details for the file configCommonDB.sh

```
################################
#  DB_NAME will be replaced
################################
DB_NAME=WPRCSDB

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

4. 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 our example this line becomes db2 set current schema=COMMONDB.

5. Edit the file insertTable_CommonDB.sql and for each of the insert statements change the values as follows:

   – #MajorVersion#: 6
   – #MinorVersion#: 1
   – #RefreshPackLevel#: 2
   – #FixpackLevel#: 0

   Before and after examples are shown in Example 5-7 and Example 5-8 respectively.

   Example 5-7 Changes to the insertTables_CommonDB.sql file: Before

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

   Example 5-8 Changes to the insertTables_CommonDB.sql file: After

   ```
   INSERT INTO SchemaVersionInfo VALUES ('recovery.ejb', 6, 1, 2, 0,
   0);
   ```

6. Create the database, schema and tables with the following command.

   ```
   sh ./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.

7. We can now check the database using the command line interface to DB2.

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

   Sample output is shown in Example 5-9 on page 99.

Example 5-9 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
```

5.1.5 Create the business process choreographer database

When creating the BPC database you can either create a simple database for testing purposes or follow common practice for production topologies, which is to use a dedicated table space and disks for performance. Both 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. You also add a schema name here. An example is shown in Example 5-10.

   Example 5-10 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. Continue to Section 5.1.6, “Create the Business Process Observer database” on page 101)

Creating a higher-performance database

For a higher performing database 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.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 (e.g. BPC)

4. Create the database, table space 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
   ```

5.1.6 Create the Business Process Observer database

You can create the Business Process Observer database in a similar way to the BPC database, a simple one for testing purposes or a higher performance one for production environments. We describe both methods.

Creating a test database

For a simple database, where performance is not important.

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. You also add a schema name here. An example is shown in Example 5-11 after the changes have been made.

   Example 5-11 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. You may now go to the next section (5.1.7, “Generating the messaging engine schemas” on page 102).

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, table space 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
```

5.1.7 Generating the messaging engine schemas

Before we can create the messaging engine schemas we must first generate them on the deployment manager.

1. Login to the deployment manager. We will 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
```

> **Tip:** Before you transfer these files you will need to edit them and remove the lines at the top, which are a log of the command line options used. Remove everything before the “CREATE SCHEMA ...” line, that is, the first 10 lines.

2. Once edited transfer the files to the DB2 system under the db2inst1 users home folder.

5.1.8 Creating the messaging engine database

Transfer the files generated in Section 5.1.7, “Generating the messaging engine schemas” on page 102 to the DB host and db2inst1 user. The creation of the database and schemas will be done on the DB2 host.

Login 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
```

5.1.9 Creating the event database

Login 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
```

5.1.10 Next steps

At this point you have two choices as to how to proceed. You can create the Remote Messaging and Remote Support topology using the windows in the graphical administrative console (Section 5.2, “Installation through the administrative console” on page 104), or you can create the same topology silently, using scripts (Section 5.3, “Installation through scripts silently” on page 137). Security considerations for these options are described in Chapter 2, “Security considerations for BPM” on page 21.

5.2 Installation through the administrative console

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 Section 5.2.1, "Creating a deployment manager profile" on page 104.
2. Create a node (custom) profile on each system and federate into the cell. See Section 5.2.2, "Creating the node profiles" on page 116.
3. Generate a deployment topology. See Section 5.2.3, "Creating a deployment topology" on page 121.
4. Populate the EVENT database. See Section 5.2.4, "Creating the event database tables" on page 133.
5. Adjust some settings that are not correctly created from the generation. See Section 5.2.5, "Checking database connectivity" on page 134.
6. Start, verify and test the topology. See Section 5.2.6, "Completing the topology configuration" on page 136 and Section 5.2.7, "Completing and verifying the configuration" on page 137.

5.2.1 Creating a deployment manager profile

Perform the following steps to create a deployment manager profile.

1. Login to the deployment manager as the root user.
2. Run the profile management tool:

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

   > **Note:** If you get the message 'X connection to <host>:10.0 broken' then you are not working from the UNIX desktop and you will need to run an X server locally.
   >
   > The pmt.sh script for the profile management tool is not available for 64-bit operating systems. If you are using a 64-bit operating system using the manageprofiles.sh script instead.

   A splash window is displayed.
3. At the Welcome to the Profile Management tool window, click **Next**.

4. In the Environment Selection window (Figure 5-2), click **WebSphere Process Server** and click **Next**.

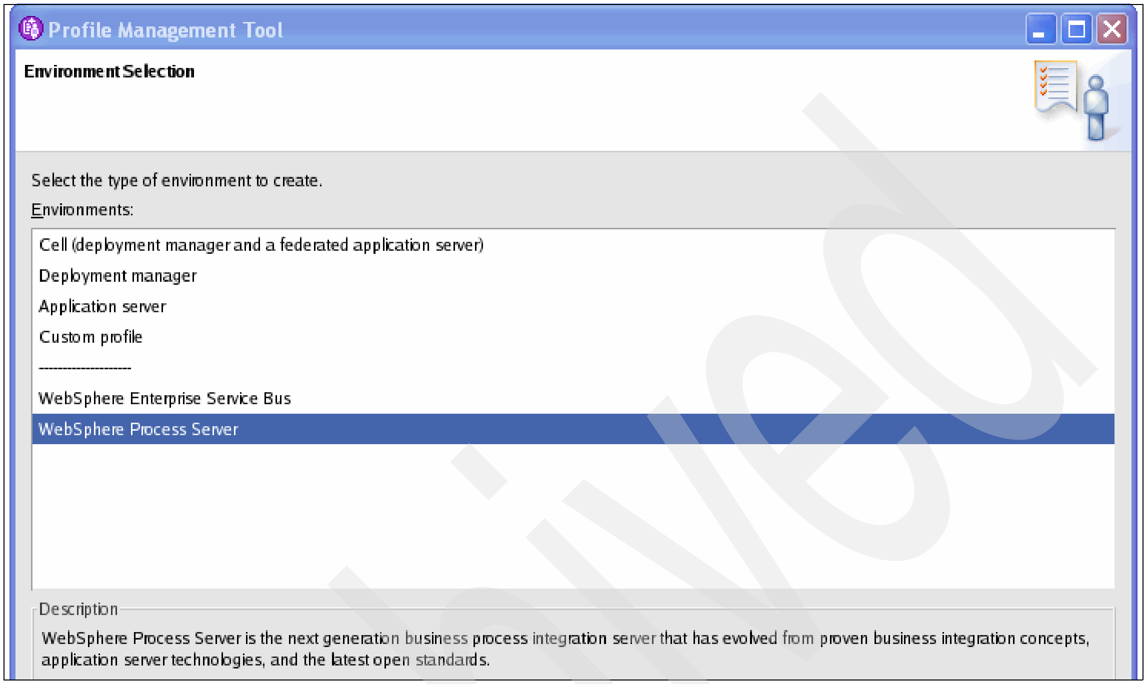


Figure 5-2 The Environment Selection window

5. In the Profile Type Selection window (Figure 5-3), click **Deployment manager profile**, and click **Next**.

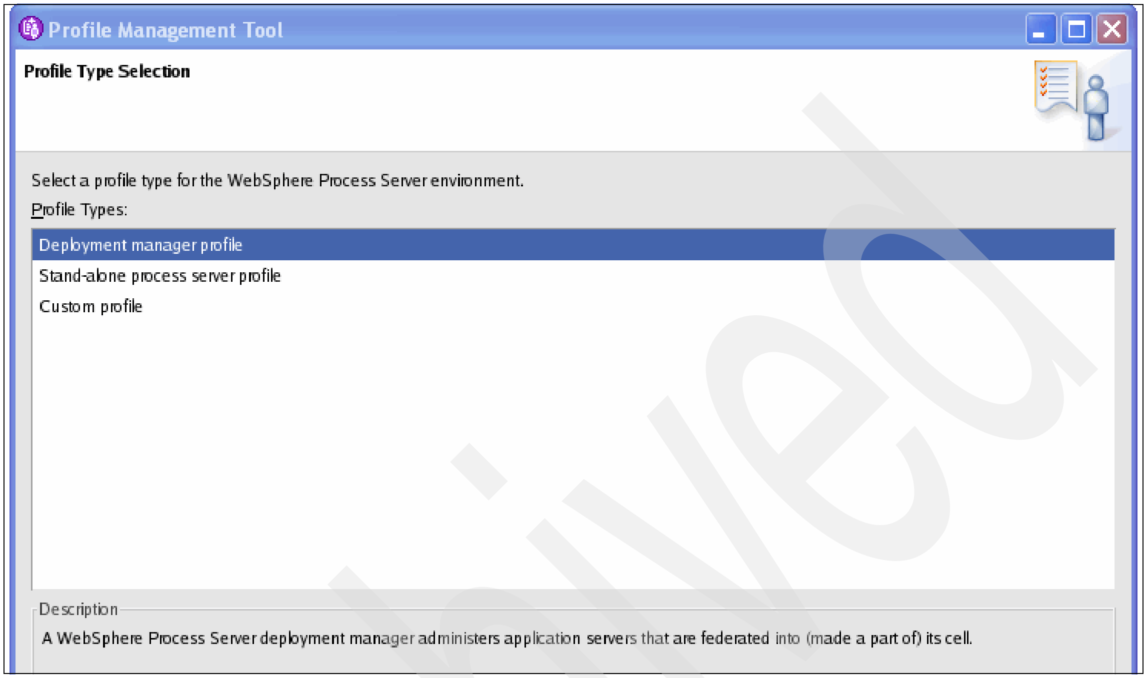


Figure 5-3 The Profile Type Selection window

6. In the Profile Creation Options window (Figure 5-4), click the Advanced profile creation radio button and click **Next**.

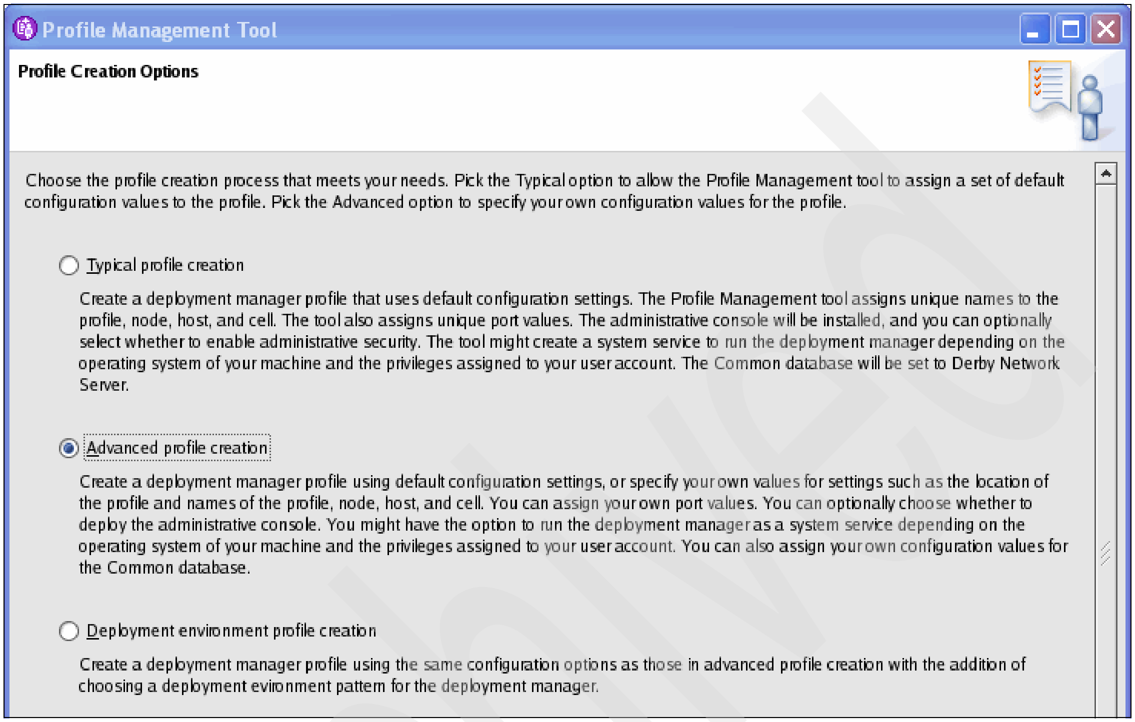


Figure 5-4 The Profile Creation Options window

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

8. In the Profile Name and Location window (Figure 5-5), leave the **Profile name** and **Profile directory** text boxes at their default and click **Next**.

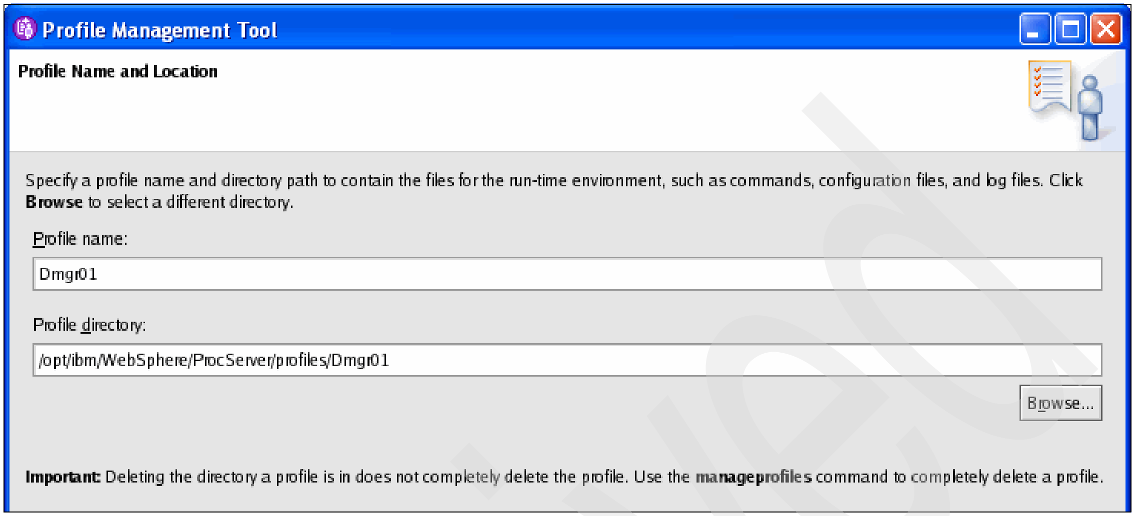


Figure 5-5 The Profile Name and Location window

9. In the Node, Host, and Cell Name window (Figure 5-6), enter CellManager01 in the Node name text box. Leave the Host name text box alone, and enter WPSCell01 in the change the Cell name text box Click **Next**.

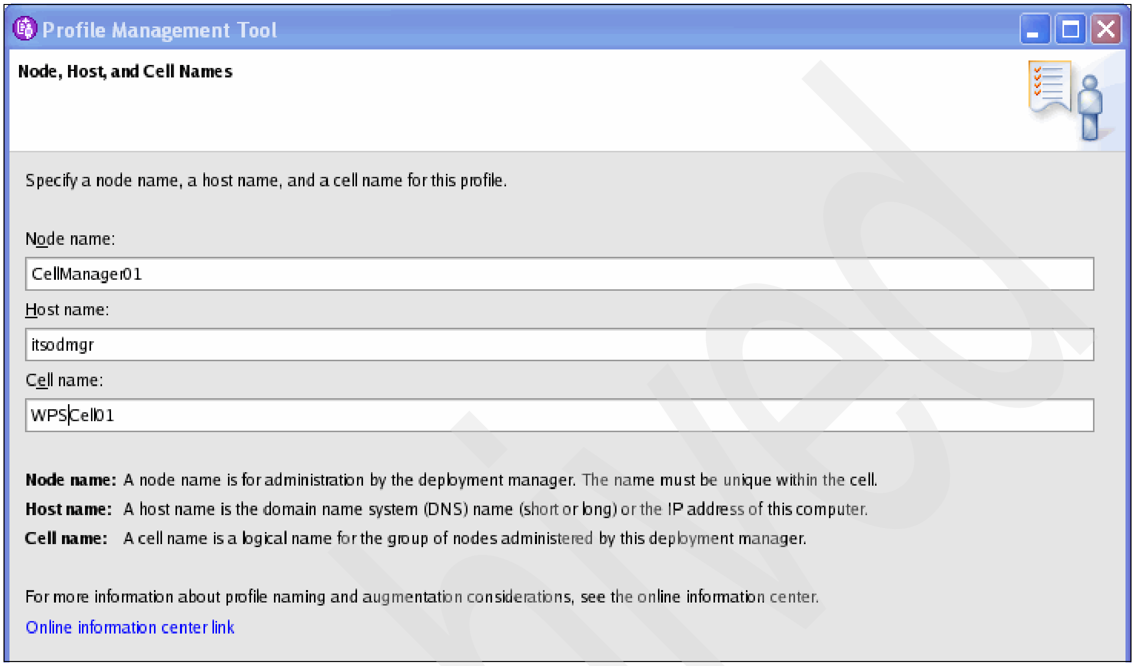


Figure 5-6 The Node, Host and Cell Name window

10. In the Administrative Security window, clear the Enable administrative security check box and click **Next**. We will add security to the topology later.

11. In the Port Values Assignment window, accept the default values. 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 and click **Next**.

13.Configure the following items in the Database Configuration window (Figure 5-7):

a. Select DB2 Universal from the drop-down menu in the Choose a database product text box.

b. Select the Use an existing database radio button (because we have already created the database).

c. Enter the value WPRCSDB in the Database name text box.

d. Select the Delay execution of database scripts for new or existing database check box, and click **Next**.

Figure 5-7 The Database Configuration window

14. Configure the following items in the Database Configuration (Part 2) window (Figure 5-8 on page 112):

   a. Enter the value db2inst1 in the Username text box to authenticate with the database text box from Table 5-1 on page 96.

   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. Ensure the JDBC driver type radio button is set to 4. Type 4 drivers allow for XA recovery and do not require database client software to be installed locally.

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

   f. Enter a value of 50000, for Server port, and click **Next**.

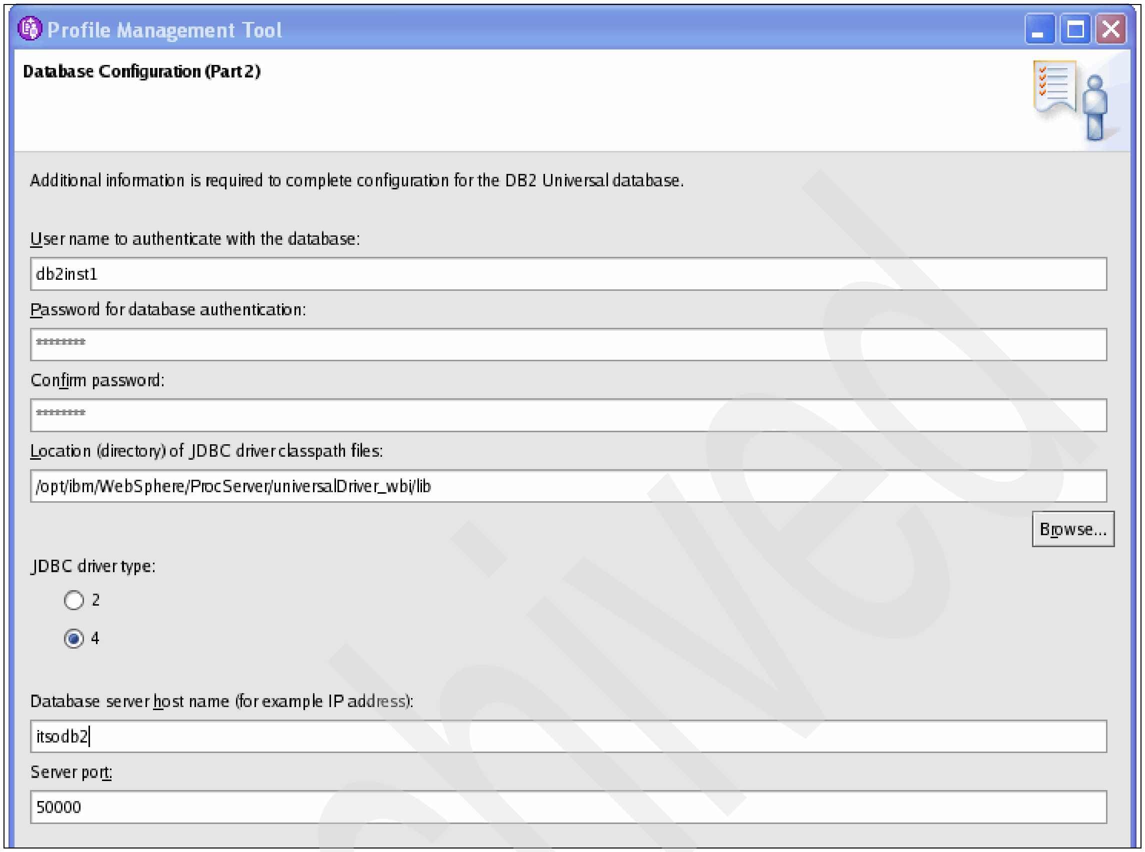


Figure 5-8 The Database Configuration (Part 2) window

15. In the Profile Creation Summary window (Figure 5-9), check the values and click **Create**. This will take some time to complete. A Profile Creation Progress window will be displayed during this process.

Figure 5-9 The Profile Creation Summary window

16. In the Profile Creation Complete window (Figure 5-10), make sure the profile creation was successful, clear the Launch the First steps console check box, and click **Finish**.

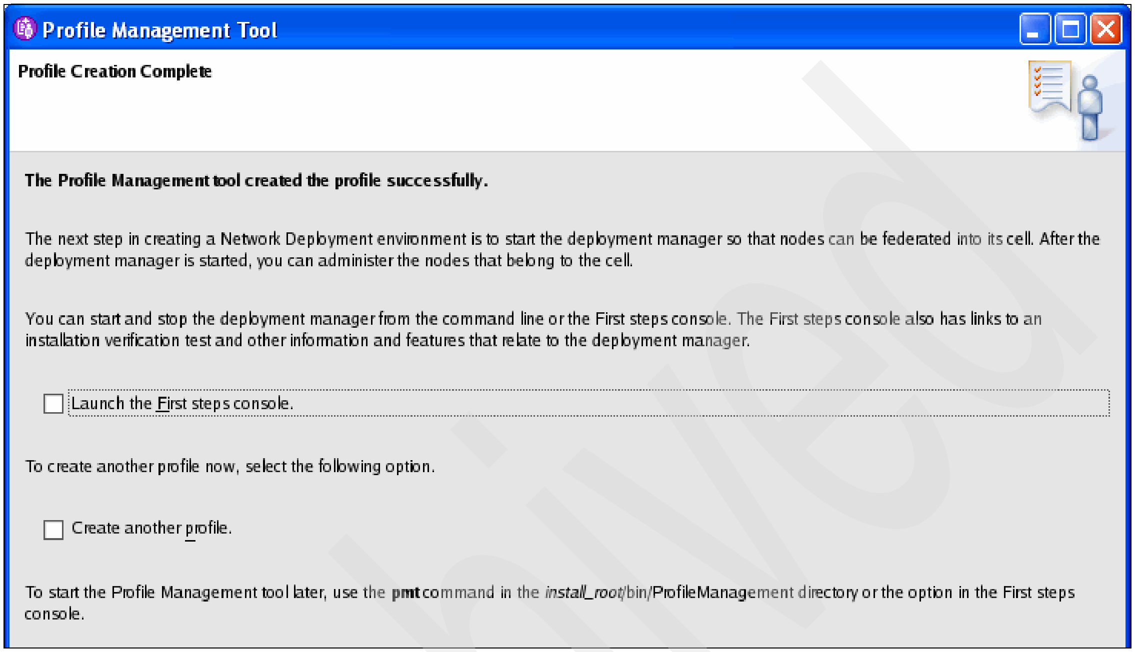


Figure 5-10 The Profile Creation Complete window

17. Start the deployment manager by entering the command:

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

Ensure you can log into the administrative console (Figure 5-11 on page 115) by using the URL http://itsodmgr:9060/ibm/console, where itsodmgr is the host name of the deployment manager, or its IP address.

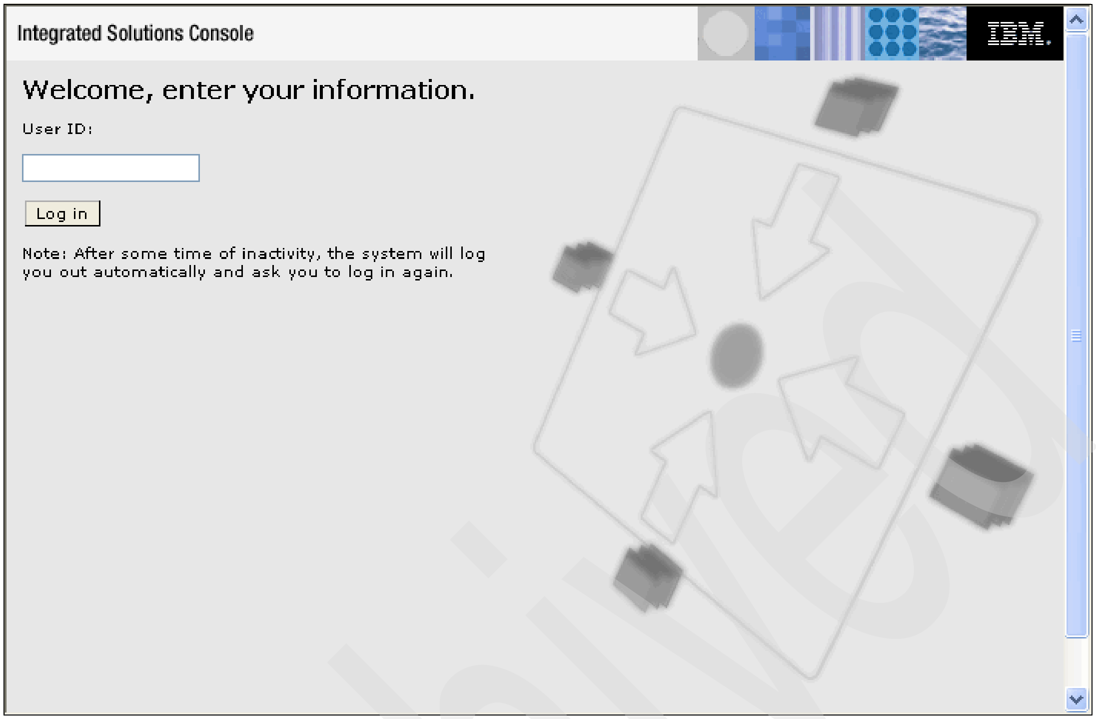


Figure 5-11 The administrative console login window

We need to make some changes for the deployment to be accurate because we have used a schema name of COMMONDB (by default the WPRCSDB 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 in the scope (Cell:WPSCell01), so click this provider.
2. Under **Additional Properties** click **Data sources** and you will see the following two data sources defined:
   - ESBLoggingMediationDataSource
   - WBI_DataSource

   Click the WBI_DataSource.
3. Scroll down and under Authentication alias for XA recovery click the Use component-managed authentication alias radio button, then click **OK**.
4. Click **WBI_DataSource** and under **Additional Properties** click **Custom Properties**.

5. Scroll down the list and click **currentSchema**, and enter the value COMMONDB, then click **OK**.
6. Scroll down the list and click **cliSchema**, and enter the value COMMONDB, then click **OK**.
7. Click **Save** at the top of the page
8. Perform steps 2–6, this time clicking ESBLoggingMediationDataSource. This enables component-managed authentication alias. Use the schema value ESBLOG. Save your changes.
9. Log out of the administrative console and restart the deployment manager using the following commands:

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

5.2.2 Creating the node profiles

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

Before you begin to create the nodes make sure that the deployment manager is running, because we will be federating 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 we only show the different windows here.

1. Login to the first node (itsonode1) as the root user.
2. Run the profile management tool:

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

> **About this installation:** If you get the message ‘X connection to <host>:10.0 broken’ then you are not working from the UNIX desktop and you will need to run an X server locally.
>
> The pmt.sh script for the profile management tool is not available for 64-bit operating systems. If you are using a 64-bit operating system using the manageprofiles.sh script instead.

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

4. Click **WebSphere Process Server** and click **Next**. The Profile Type Selection window is displayed (Figure 5-12).
5. Click **Custom profile** and click **Next**. The Profile Creations Options window is displayed.

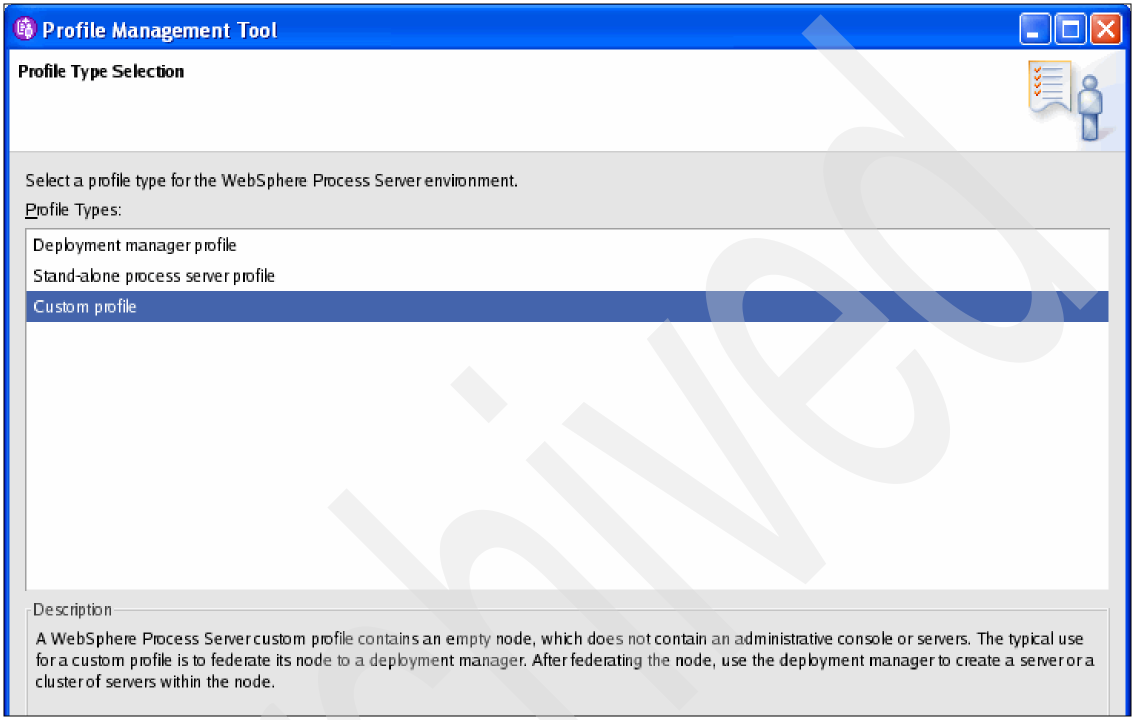


Figure 5-12 The Profile Type Selection window with a custom profile selected

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

9. Enter the host name for the deployment manager (itsodmgr) in the Deployment manager host name or IP address text box (Figure 5-13). Leave all other values at their defaults. You do not need a username and password because we do not enable security at this stage so these values are empty. Click **Next**. The Port Values Assignment window is displayed.

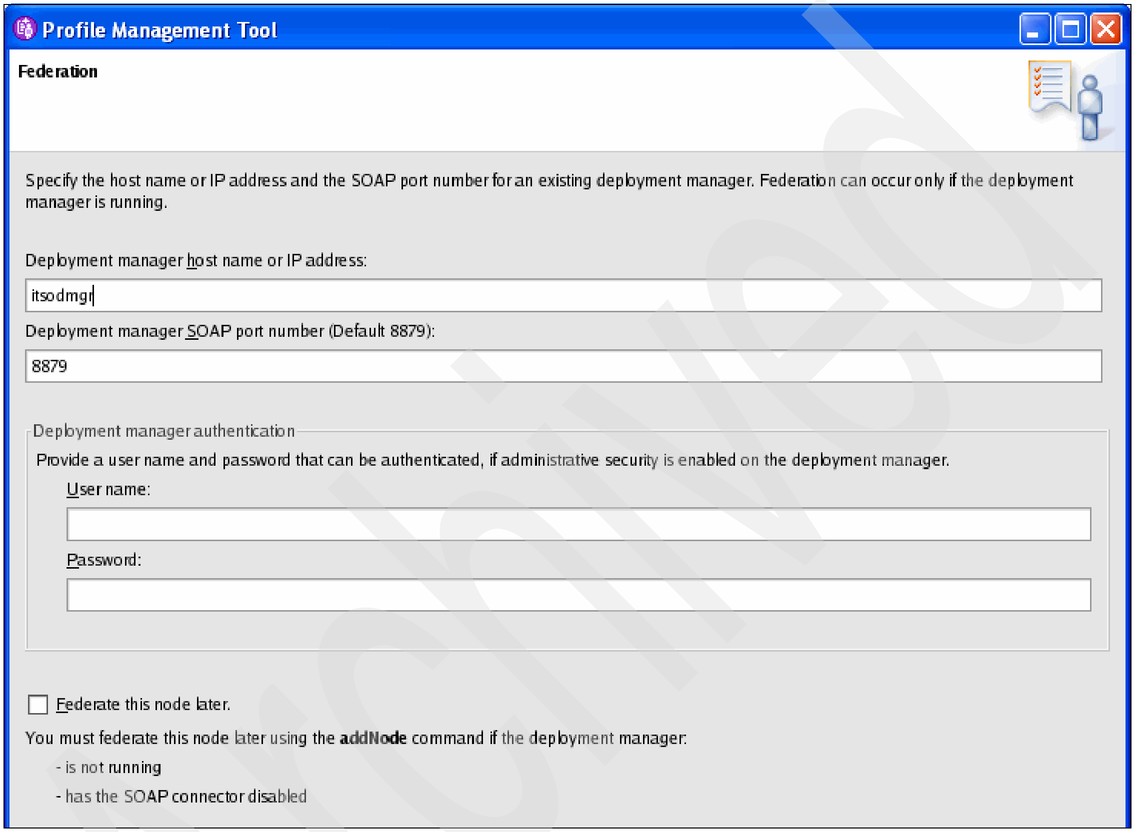


Figure 5-13 The Federation window

10. Accept all the default values and click **Next**. The Database Configuration window is displayed (Figure 5-14).

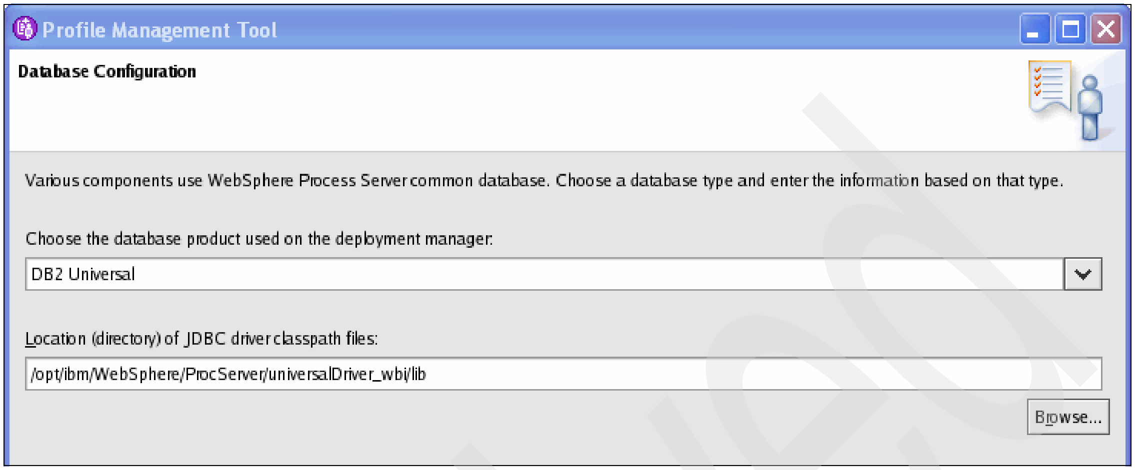


Figure 5-14 The Database Configuration window for a custom profile

11. 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.

12. Check the values and click **Create**. This will take some time to complete. A Profile Creation Progress window will be displayed. When profile creation is complete, the Profile Creation Complete window is displayed.

13. Ensure the profile creation was successful. Clear the Launch the First steps console radio button, and 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 deployment manager must be within 5 minutes of one another

You can now login 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 and Profile directory to end in Custom02 and on the Node and Host name window use wpsNode02 as the Node name, but they should be the only changes.

Creating the node profiles automatically starts the node agent so we can login to the administrative console and check the nodes are available. In the administrative console navigate to **System Administration** → **Node agents** and on the right hand side you should see you newly created nodes running (Figure 5-15).

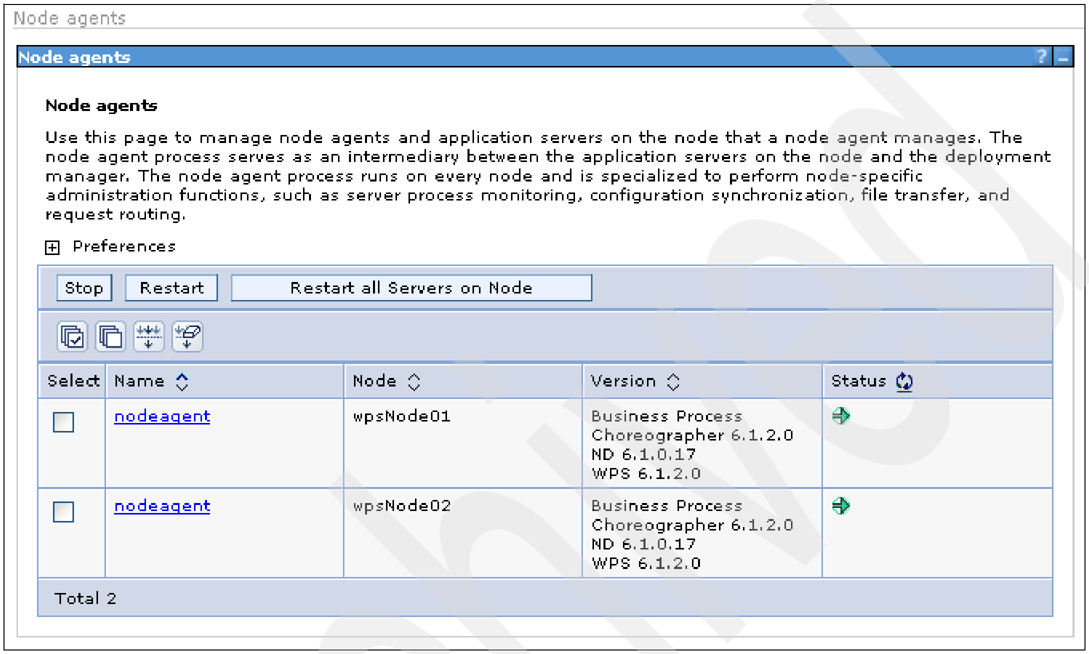


Figure 5-15 The administrative console showing the nodes are running

5.2.3 Creating a deployment topology

We will now create a Remote Messaging and Remote Support topology using the administrative console. Before beginning, ensure that the deployment manager and nodes are running.

1. Login 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 will open (Figure 5-16).

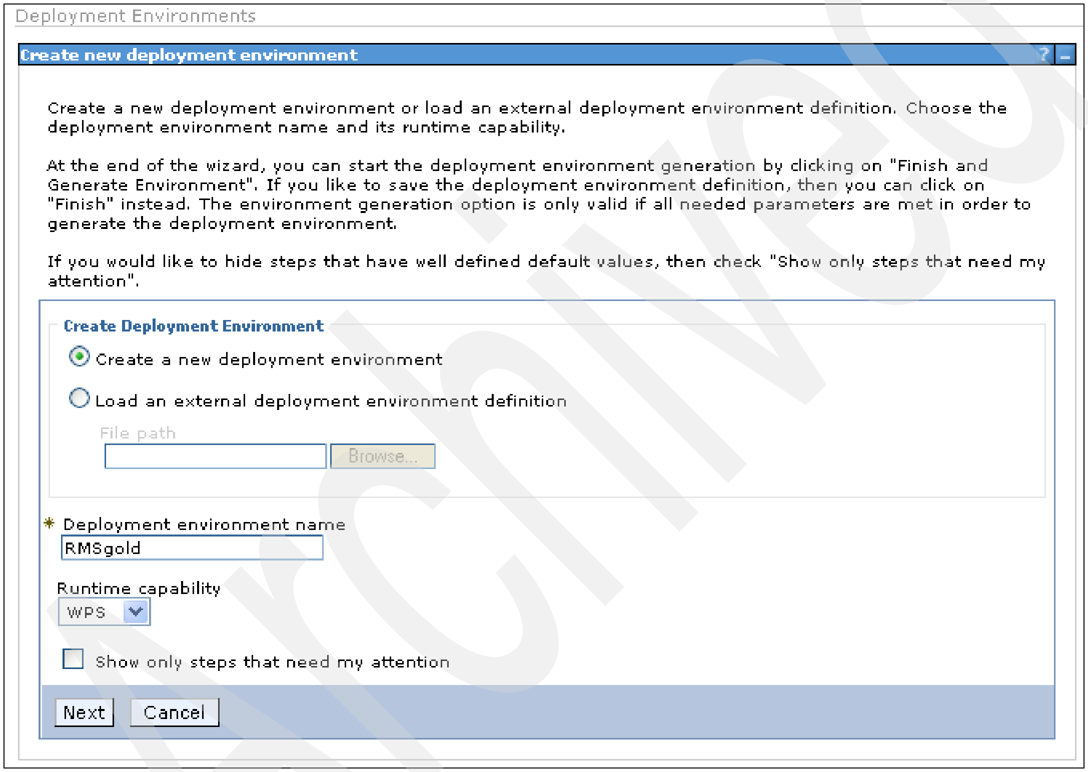


Figure 5-16 The Create new deployment pane

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 will open (Figure 5-17).

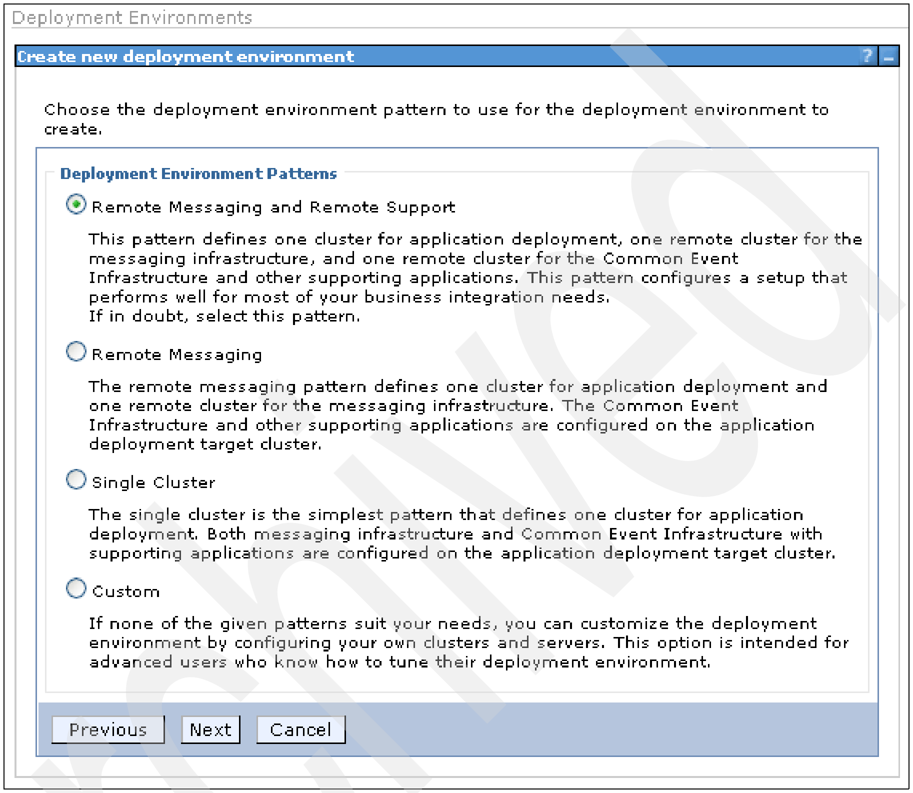


Figure 5-17 The Deployment Environment Patters window

4. Select the Remote Messaging and Remote Support radio button. This is the gold topology. Click **Next**. The Select Nodes window will appear (Figure 5-18).

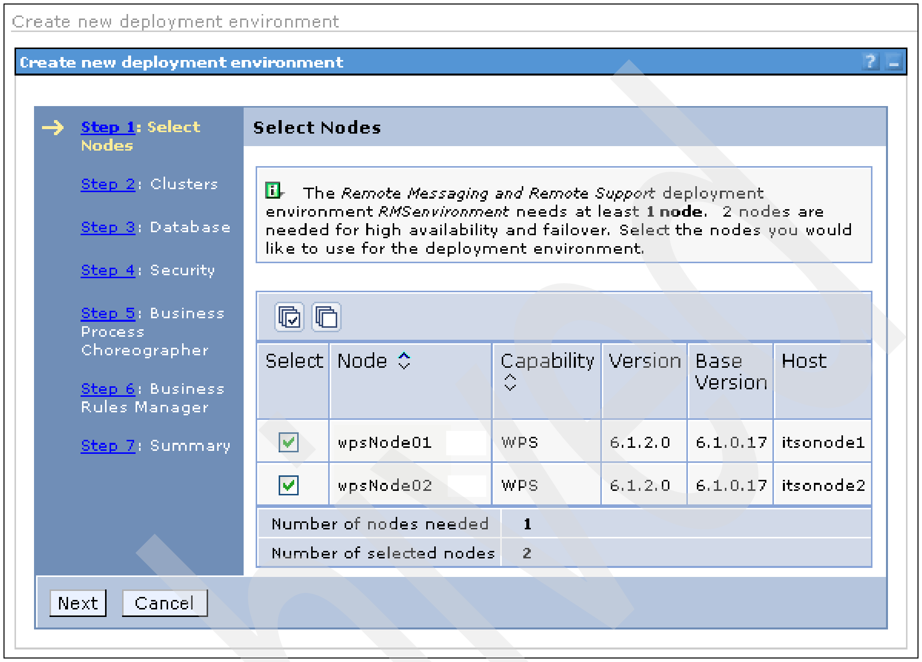


Figure 5-18 The Select Nodes window

You will now see a seven stage process outlined beginning with Select Nodes. Our topology consist of just two nodes so we will use both of then but in a larger environment you can select a sub-set of the entire node list.

5. Click the check box against both nodes and click **Next**. The Clusters window will appear (Figure 5-19).

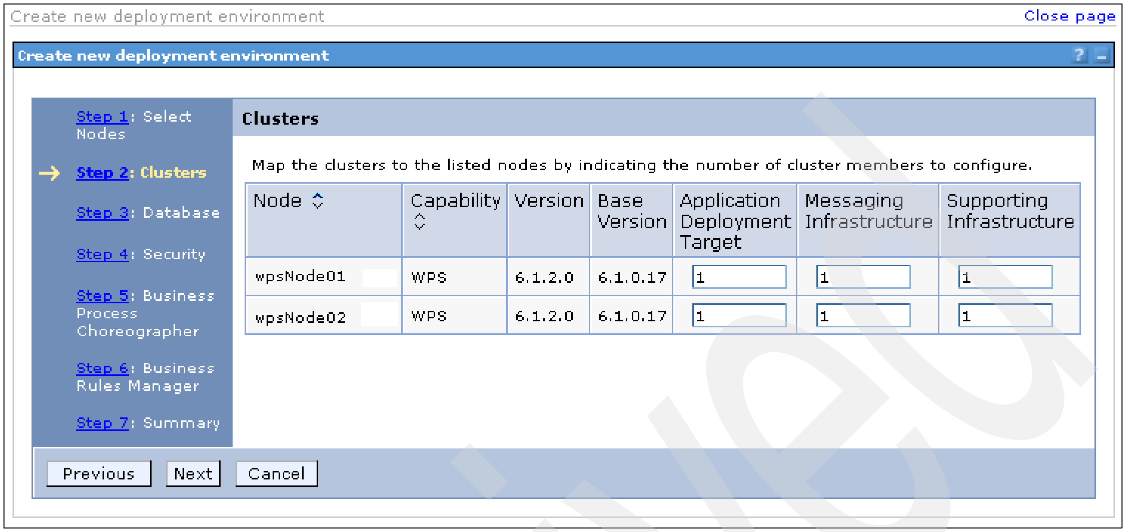


Figure 5-19 The Clusters window, where you can select the distribution of servers within the clusters

The next 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, and Supporting Infrastructure is the cluster for CEI and other services. We will be creating 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 Database window is displayed (Figure 5-20).

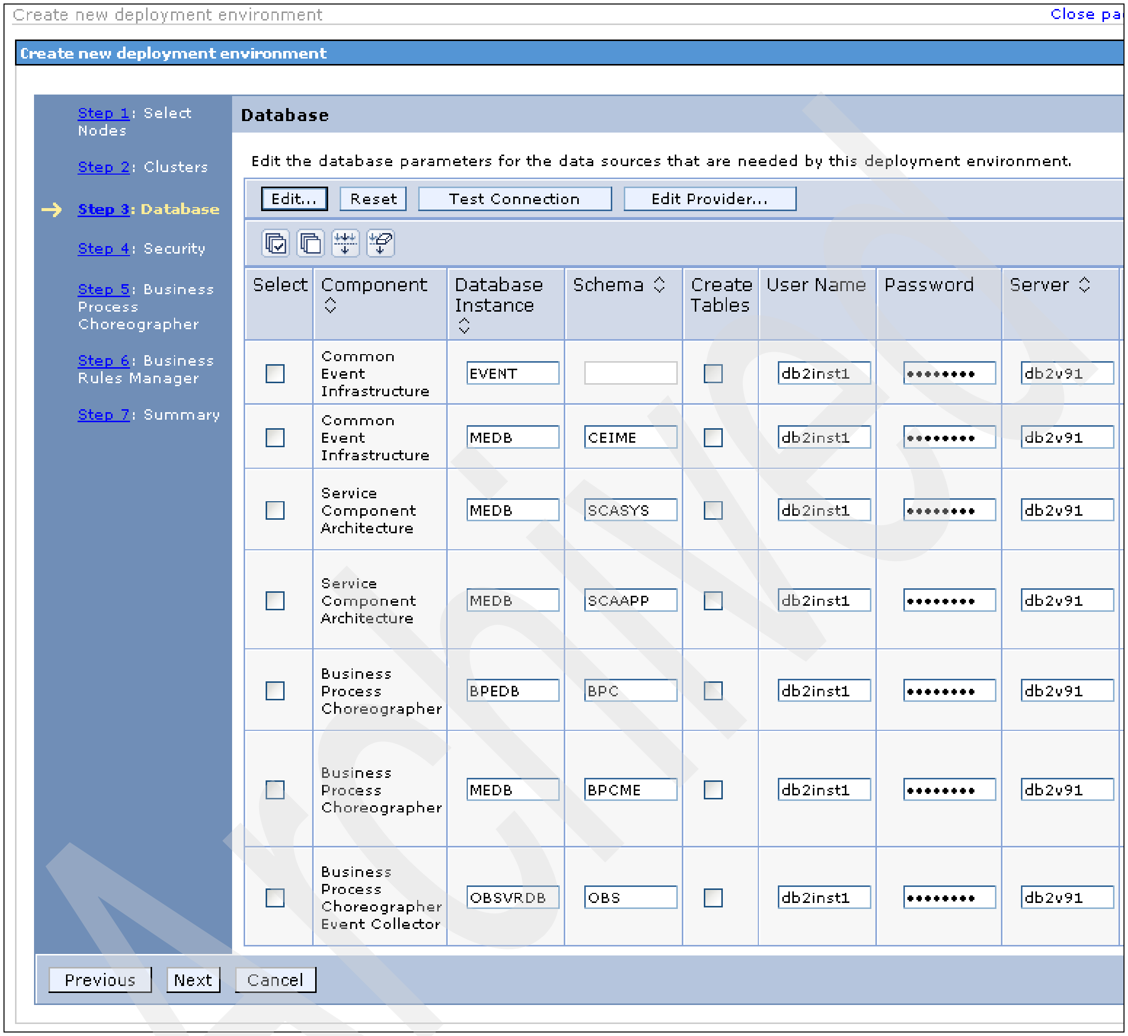


Figure 5-20 The Database window: Care must be taken over the passwords

The Database window is the most complex, and care needs to be taken to edit this table correctly. Refer to Table 5-2 for a description of the fields and how they relate to the databases we created earlier.

Table 5-2 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
OBSVRDB	Business Process Choreographer Event Collector data source	Created earlier with schema OBS

7. Fill in the form with the details shown in Table 5-2 on page 126. Figure 5-20 on page 125 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 (Figure 5-21).

Figure 5-21 The Security window

8. 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 is displayed (Figure 5-22).

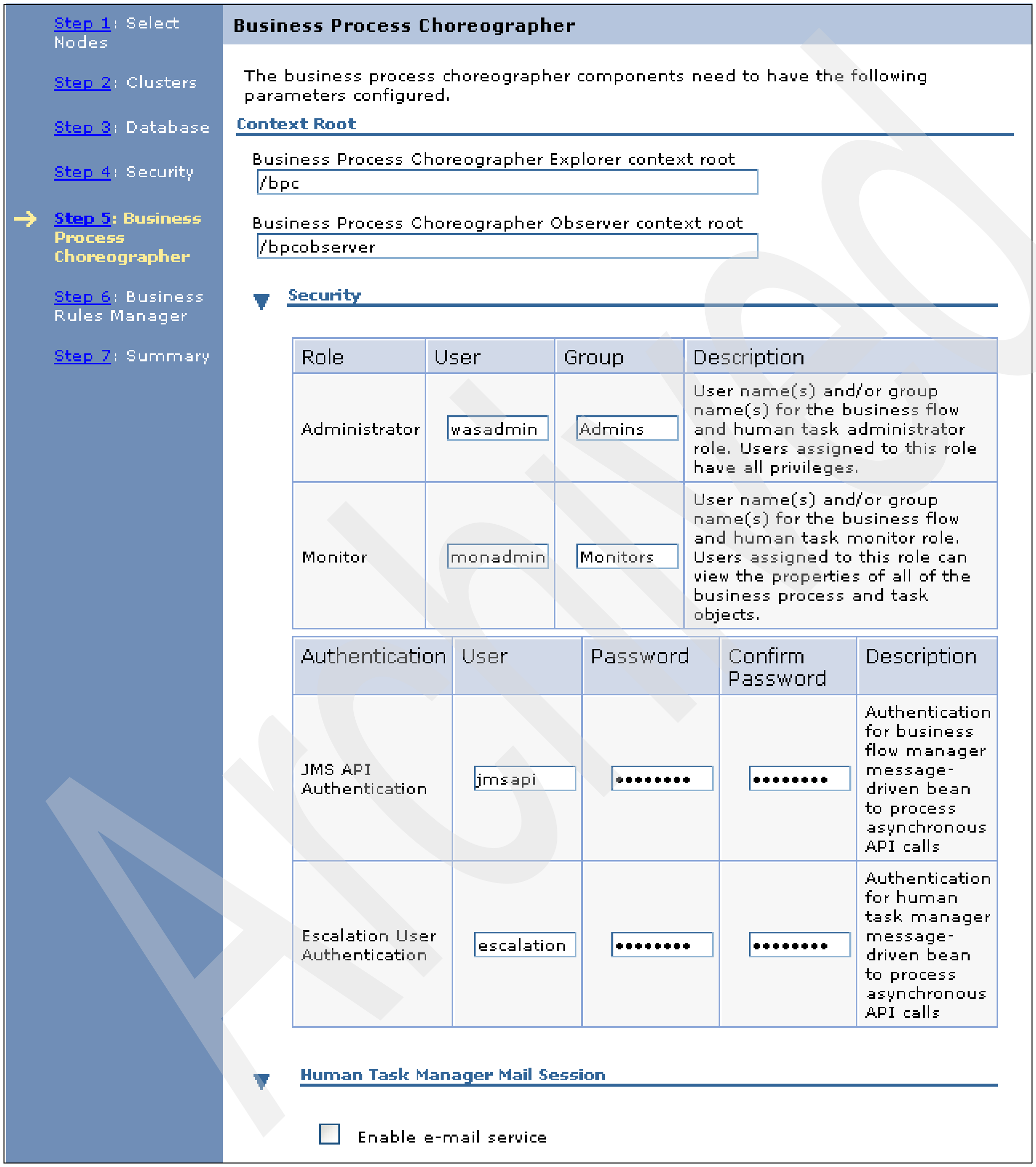


Figure 5-22 The Business Process Choreographer window

9. Perform the following steps to specify some groups and users for authorization:

   a. For the Administrator role, use the following values for 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 User and password:

      - User: jmsapi
      - Password: passw0rd

   d. For the Escalation User authentication, use the following values for User and Password:

      - User:`escalation`
      - Password: `passw0rd`.

      When we enable LDAP, these users and groups must in the LDAP database.

   e. Clear the Enable e-mail service check box 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 is displayed.

10. Click **Next**. The Summary window is displayed.

11. On the Summary panel, check your settings and click Finish.

> **Note:** Do not click **Finish and Generate Environment** since you want to review your settings then generate. Do not try to start the Deployment Environment because the EVENT database does not yet exist and this will cause the deployment to fail.

12. Save the changes. The Deployment Environments window (Figure 5-23) 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 we have a definition of an environment but no resources have yet been created.

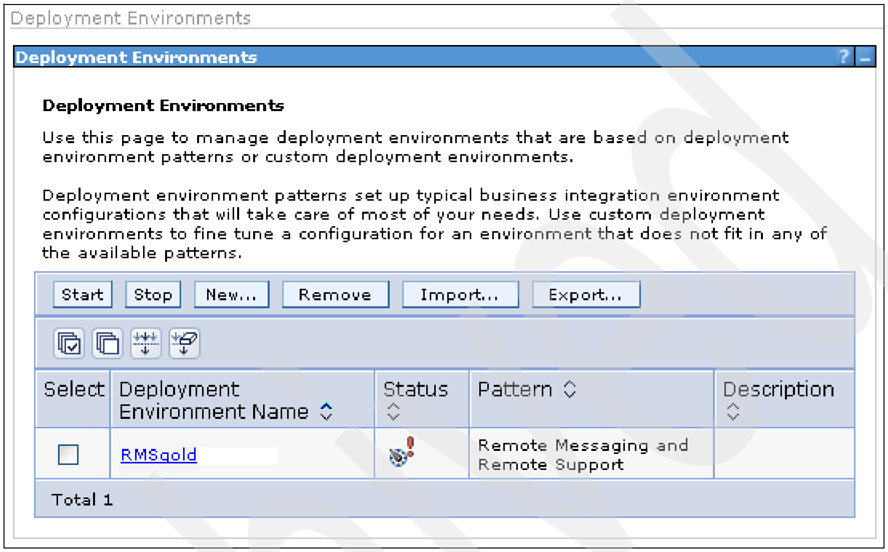


Figure 5-23 The Deployment Environments window

13.Click the **RMSgold** link to display the Configuration window (Figure 5-24). This window shows the status of the three clusters we have defined all of which are currently not configured.

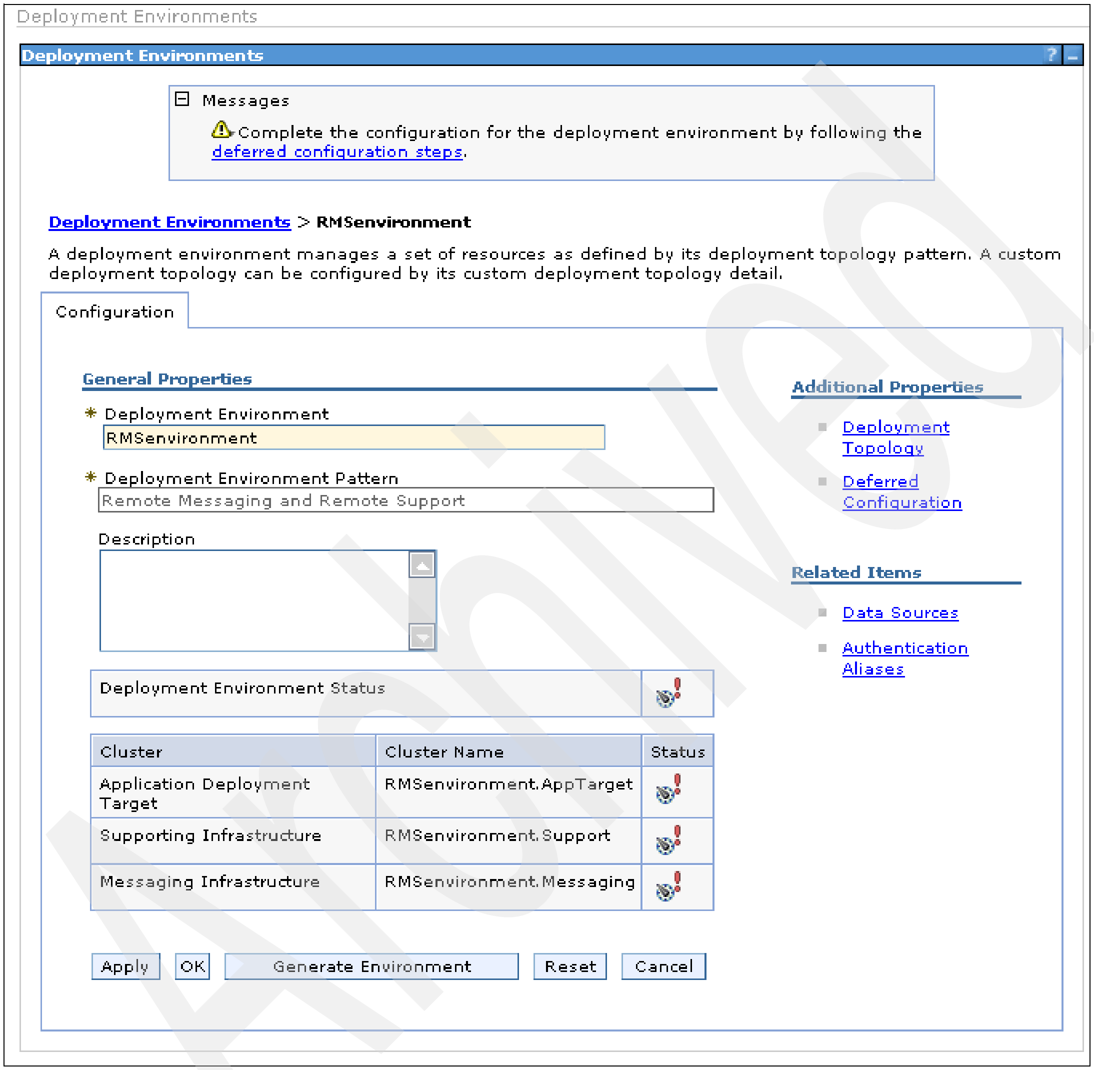


Figure 5-24 The Deployment Environments Configuration window

14.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.

15.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.

16.Click **Generate Environment**. A Configuration Status window will be displayed. When complete, click **Save Changes**. The environment will now have a status of Stopped (Figure 5-25).

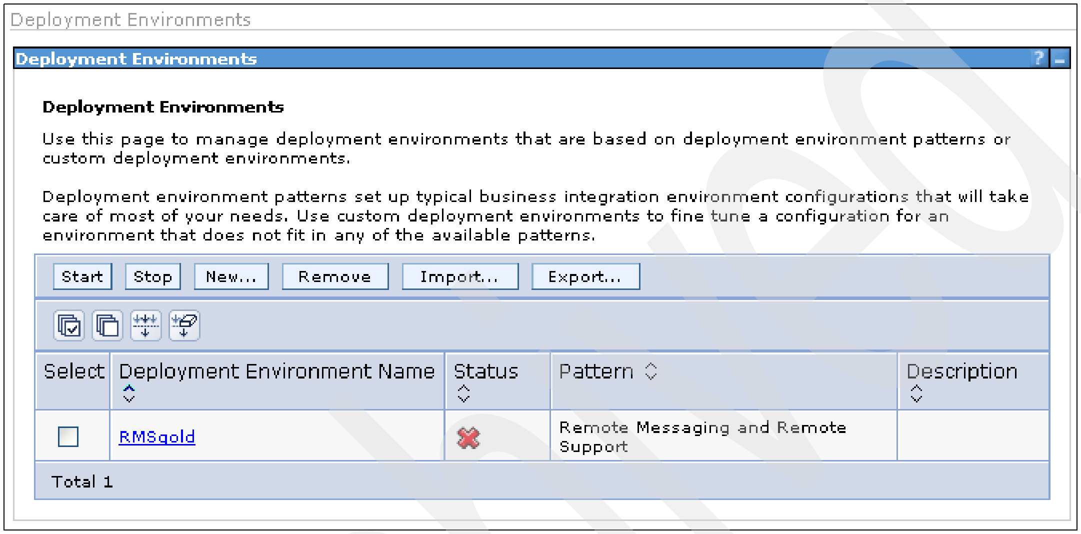


Figure 5-25 The Deployment Environments status window

17.Log out of the administrative console.

5.2.4 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 on the deployment manager under the deployment manager profile.

```
/opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/databases/event/RMSgold.S
upport/dbscripts/db2
```

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

1. Login to the DB2 System as the instance owner
2. Change 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
> ```
>
> 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 the DB2 system.

5.2.5 Checking database connectivity

Before we start the environment we need to check database connectivity.

1. Login to the administrative console and navigate to **Resources** → **JDBC** → **JDBC Providers**. You will see that there are four providers now at different scopes (one for the cell and three for the separate clusters).
2. Click the first one where the scope is Cell=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 they have connectivity.
3. We now need to define some 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 at the scope Cluster=RMSgold.Support.

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

8. Under Additional Properties, click **Data sources**, select the check box (there is only one), and click **Test connection**.
9. Navigate to **Resources** → **JDBC** → **JDBC Providers**. Click the provider at the scope Cluster=RMSgold.AppTarget.

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

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 at the scope Cluster=RMSgold.Messaging.

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

12. Under Additional Properties, click **Data sources**. You will see one data source for each of the schemas we created earlier. In each case, we need to make sure the Authentication alias is correctly set before we test the connection.
13. 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**.
14. 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**.
15. Perform the same actions for the other three data sources using the values shown in Table 5-3.

Table 5-3 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

5.2.6 Completing the topology configuration

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

1. Login to the administrative 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 will immediately change to Started, but you need to wait while the application servers start.
5. Navigate to **Servers** → **Clusters** and you will see the server clusters state is now Partial Start (Figure 5-26).

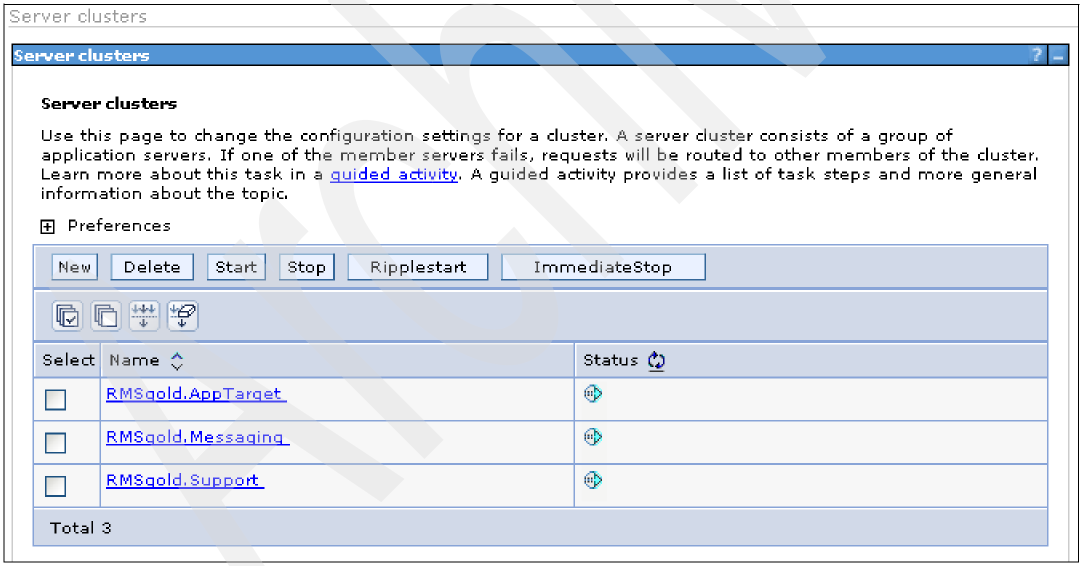


Figure 5-26 The Server Clusters window

5.2.7 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 Section 5.4, “Post-creation configuration and verification” on page 147.

5.3 Installation through scripts silently

This section demonstrates the silent install process. We start from the same point as installation as with the graphical user interface as detailed in Section 5.2, “Installation through the administrative console” on page 104. 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 Section 5.3.1, “Creating a properties file” on page 138.
2. Create a deployment manager profile. See Section 5.3.2, “Creating a deployment manager profile” on page 139.
3. Create the node profiles. See Section 5.3.3, “Creating the node profiles” on page 140.
4. Import and generate a topology definition. See Section 5.3.4, “Importing and generating a topology definition” on page 141.
5. Populate the event database. See Section 5.3.5, “Populating the event database” on page 144.
6. Adjust some settings that are not correctly created from the generation. See Section 5.3.6, “Post-generation topology fixes” on page 144.
7. Start, verify and test the topology. See Section 5.3.7, “Automation of silent installation” on page 146.

5.3.1 Creating a properties file

Many of the values used in silent installation are the same as were used for installing through an administrative console. For example, database name, database user, database password. Therefore, we will start by creating a simple properties file to contain these values. Edit a file called properties.sh with the contents shown in Example 5-12. The values that may need to be changed are dmgrName, dmgrPort, dbHost, dbPort, dbUser, and dbPass.

Example 5-12 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 globals security is enabled we need these evalues

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 nodes.

5.3.2 Creating a deployment manager profile

We can now create a deployment manager profile silently using the script createDmgr.sh. Make sure it is in the same folder as the location of the properties.sh file. This script will take 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, your deployment manager will be created. The output is shown in Example 5-13.

Example 5-13 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 file above assumes there are no port conflicts. It uses the default ports when creating a deployment manager. You can specify different ports using the -startingPort value (e.g. -startingPort 20000) or using the -portsFile option (e.g. -portsFile Myports.props), and list the ports explicitly in the given file.

We now need to make the following changes to the deployment manager:

1. Change the SCA_Auth_Alias password from the default of SCA to our 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.

We perform these steps using a jython script based on the toolkit library. Edit the file called changeDmgr.py. You should edit the SCA_Auth_Alias password (the third parameter to modifyJ2CAuthData) to your needs. This file needs to be placed in the same folder as the toolkit libraries.

Run this script as follows on the deployment manager:

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

The output is shown in Example 5-14.

Example 5-14 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 your run wsadmin.sh, the system will process many JAR files leading to many lines of output. This will only happen once. We do not show this output in the example.

Finally, we need to start the deployment manager:

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

5.3.3 Creating the node profiles

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

Before you begin to create the nodes make sure that the deployment manager is running because we will be federating the nodes as part of the creation process. We can now create the first profile silently using a script. Inspect the createNode.sh file. Make sure it is in the same folder as the location of the properties.sh file.

Note that this script will take two optional parameters: a profile name (default Custom01) and a node name (default wpsNode01). Once you have created this file then you can run it by executing the following command:

```
sh ./createNode.sh Custom01 wspNode01
```

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 5-15.

Example 5-15 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.
```

In the second node, you could 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:

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

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

5.3.4 Importing and generating a topology definition

In this section, we import a topology definition. We use the same topology that we created through the administrative console. We simply used the Export button to save the topology as a file. Before we perform the import, it is worthwhile to look through this file to discuss some of its features. The first few lines are shown in Example 5-16 on page 142.

Example 5-16 The first few lines of the RMSgold.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="6.1.2.0">
  <pattern id="Reference" name="Remote Messaging and Remote Support"
version="6.1.2.0"/>
  <runtime id="WPS" name="WPS" version="6.1.2.0" value="3"/>
  <roles id="ADT" name="AppTarget" version="6.1.2.0" baseRuntime="WPS">
    <deploymentTarget xsi:type="wbitopology:WBICluster"
name="RMSgold.AppTarget"/>
    <nodes name="wpsNode01" runtime="WPS" hostName="itsonode1" port="0"
numOfServers="1"/>
    <nodes name="wpsNode02" runtime="WPS" hostName="itsonode2" port="0"
numOfServers="1"/>
  </roles>
  <roles id="Support" name="Support" version="6.1.2.0"
baseRuntime="WPS">
    <deploymentTarget xsi:type="wbitopology:WBICluster"
name="RMSgold.Support"/>
    <nodes name="wpsNode01" runtime="WPS" hostName="itsonode1" port="0"
numOfServers="1"/>
    <nodes name="wpsNode02" runtime="WPS" hostName="itsonode2" port="0"
numOfServers="1"/>
  </roles>
  <roles id="Messaging" name="Messaging" version="6.1.2.0"
baseRuntime="WAS">
    <deploymentTarget xsi:type="wbitopology:WBICluster"
name="RMSgold.Messaging"/>
    <nodes name="wpsNode01" runtime="WPS" hostName="itsonode1" port="0"
numOfServers="1"/>
    <nodes name="wpsNode02" runtime="WPS" hostName="itsonode2" port="0"
numOfServers="1"/>
  </roles>
```

Note that the second line contains the name of our topology, followed by three stanzas defining the clusters and nodes that support them. This is equivalent to the window shown in Figure 5-19 on page 124. Using this file in a different environment would mean changing the name and hostName in the nodes stanza.

We also have the data sources defined with an example given in Example 5-17. Note that the database user (db2inst1) is embedded in this file in each datasource stanza, so that would need to be changed for a different environment as well.

Example 5-17 Example data source stanza

```
<dataSrc component="WBI_CEI" createTable="false"
dbcomponent="WBI_CEI_EVENT">
     <authAlias name="WPSCell01/RMSgold.Support/EventAuthDataAliasDB2"
userName="db2inst1" password="{xor}Lz4sLChvLTs=" 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="itsodb2" 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=RMSgold.Support"
databaseType="DB2_UNIVERSAL" providerType="DB2 Universal JDBC Driver
Provider" implementationType="XA data source"
dbcomponent="WBI_CEI_EVENT"/>
    </dataSrc>
```

The last few lines of the file show the context root of the BPC Explorer, BPC Observer and the business rules manager.

Now we do the import. The deployment manager must be running for this import to work. Inspect the file called createTopology.sh. You may need to change the fileName value to the location in your environment. This must be an absolute path name.

We run the following command on the deployment manager:

```
cd /opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/bin/wsadmin.sh -lang
jython -conntype SOAP -f createTopology.py
```

There is no output to this action.

5.3.5 Populating the event database

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

```
/opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/databases/event/RMSgold.S
upport/dbscripts/db2
```

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

1. Login to the DB2 system as the instance owner.
2. Change 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. Check the file output.log for any messages.

> **Note:** These DB2 commands may report various informational messages, such as:
>
> ```
> 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.

5.3.6 Post-generation topology fixes

Here we need to create some environment variables, add the ports to default_host, and change some of the data source properties.

Edit the file called postfixTopology.py. You may need to change the location of your DB2 JDBC Driver files.

Run the following command on the deployment manager:

```
cd /opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/bin/wsadmin.sh -lang
jython -conntype SOAP -f postfixTopology.py
```

The output is shown in Example 5-18.

Example 5-18 The output of the postfixTopology.py file

```
====== Create variable DB2UNIVERSAL_JDBC_DRIVER_PATH with value
/opt/ibm/WebSphere/ProcServer/universalDriver_wbi/lib, if it does not
exist ======
Attempting to create the variable DB2UNIVERSAL_JDBC_DRIVER_PATH on
RMSgold.Support
Create/modify variable successful.

====== Create variable DB2UNIVERSAL_JDBC_DRIVER_PATH with value
/opt/ibm/WebSphere/ProcServer/universalDriver_wbi/lib, if it does not
exist ======
Attempting to create the variable DB2UNIVERSAL_JDBC_DRIVER_PATH on
RMSgold.AppTarget
Create/modify variable successful.

====== Create variable DB2UNIVERSAL_JDBC_DRIVER_PATH with value
/opt/ibm/WebSphere/ProcServer/universalDriver_wbi/lib, if it does not
exist ======
Attempting to create the variable DB2UNIVERSAL_JDBC_DRIVER_PATH on
RMSgold.Messaging
Create/modify variable successful.
Changing Auth Alias for Business Process Choreographer ME data source
to BPCME_00_Auth_Alias
Changing Auth Alias for CEI ME data source to
CEIME_RMSgold.Messaging_Auth_Alias
Changing Auth Alias for SCA Application Bus ME data source to
SCAAPPME00_Auth_Alias
Changing Auth Alias for SCA System Bus ME data source to
SCASYSME00_Auth_Alias
Changing Auth Alias for event to WPSDB_Auth_Alias
Changing Auth Alias for Business Process Choreographer data source to
BPCDB_RMSgold.AppTarget_Auth_Alias
Changing Auth Alias for Business Process Choreographer Event Collector
data source to BPCDB_RMSgold.AppTarget_Auth_Alias
Changing Auth Alias for ESBLoggerMediationDataSource to
WPSDB_Auth_Alias
Changing Auth Alias for WBI_DataSource to WPSDB_Auth_Alias
```

We need to add some ports to the virtual hosts and restart the nodes for the changes to take effect. Edit the file called `changeVhost.py` and change the ports if required.

We run this command on the deployment manager:

```
cd /opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/bin/wsadmin.sh -lang
jython -conntype SOAP -f changeVhost.py
```

The output is shown in Example 5-19.

Example 5-19 The output of the changeVhost.py file

```
Adding virtual host *:9081
Adding virtual host *:9082
Restarting
wpsNode01(WebSphere:name=NodeAgent,process=nodeagent,platform=common,no
de=wpsNode01,diagnosticProvider=true,version=6.1.0.17,type=NodeAgent,mb
eanIdentifier=NodeAgent,cell=WPSTestCell01,spec=1.0)
Restarting
wpsNode02(WebSphere:name=NodeAgent,process=nodeagent,platform=common,no
de=wpsNode02,diagnosticProvider=true,version=6.1.0.17,type=NodeAgent,mb
eanIdentifier=NodeAgent,cell=WPSTestCell01,spec=1.0)
```

You can now start the deployment environment.

5.3.7 Automation of silent installation

As can be seen from the previous steps, the GUI is not required for silent installation. It is possible, then, to completely automate the process using UNIX shell scripts.

In the above scenario this would just be a simple concatenation of all the parts together. However, the XML deployment description can be used as a template and parameterized in such a way, that by using a properties file and the template you can be more flexible in your deployments (for example, change the names of the nodes, the database user, and so forth).

5.4 Post-creation configuration and verification

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

5.4.1 Checking database tables

We can check the creation of the database tables using simple DB2 commands as we saw in Section 5.1.4, “Create the common database” on page 97.

1. Login to the database server as the instance owner.
2. Check the messaging engine database first using the command line interface to DB2:

```
db2 connect to MEDB
db2 list tables for schema SCASYS
db2 list tables for schema SCAAPP
db2 list tables for schema BPCME
db2 list tables for schema CEIME
db2 connect reset
```

Sample (truncated) output is shown in Example 5-20.

Example 5-20 db2 list tables for schema SCASYS

```
Table/View                      Schema
SIB000                          SCASYS
SIB001                          SCASYS
SIB002                          SCASYS
SIBCLASSMAP                     SCASYS
SIBKEYS                         SCASYS
SIBLISTING                      SCASYS
SIBOWNER                        SCASYS
SIBXACTS                        SCASYS

8 record(s) selected.
```

Using the same commands we can connect to the database BPEDB and show the tables for the schema BPC, and connect to the database OBSVRDB and show the tables for the schema OBS. Sample (truncated) output is shown for the observer database (Example 5-21).

Example 5-21 db2 list tables for schema OBS

```
Table/View                      Schema
EVENT_ACT_T                     OBS
EVENT_PRC_T                     OBS
INST_ACT_T                      OBS
INST_PRC_T                      OBS
OBSERVER_VERSION                OBS
OPEN_EVENTS_T                   OBS
QUERY_T                         OBS
SLICES_T                        OBS

8 record(s) selected.
```

5.4.2 Adding the Web server to the administrative console

During 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.

You need to copy this script to your deployment manager into 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 will allow 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.

5.4.3 Configuring CEI logging

To configure CEI logging, perform the following steps:

1. In the administrative console, navigate to **Servers** → **Clusters** → **RMSgold.AppTarget**.
2. In the right hand side, under Container Settings, expand the Business Process Choreographer Container Settings section, then click **Business Process Choreographer Containers**.
3. Scroll down the page and expand the State Observers section. Click **Common Event Infrastructure Logging** for either the Business Flow Manager or Human Task Manager check boxes, or both, depending on your requirements.
4. Save and synchronize your changes.

5.4.4 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" on page 150
- "Changing the transaction manager log settings" on page 152
- "Policies for transaction manager peer recovery" on page 152

High availability considerations for the transaction manager

The WebSphere Application Server transaction manager (utilized 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 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, perform the following steps:

1. In the administrative console, expand **Servers** and click the **Clusters** link.
2. Click the check box for the cluster you wish to modify and click **Stop**.
3. Once the cluster is stopped, click the link for the cluster you wish to modify. Figure 5-27 and Figure 5-28 on page 151 show 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. See Figure 5-27.

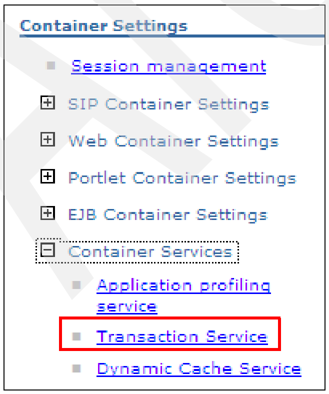


Figure 5-27 Transaction Service link

7. In the General Properties section, enter an appropriate value in the Transaction log directory text box. See Figure 5-28.

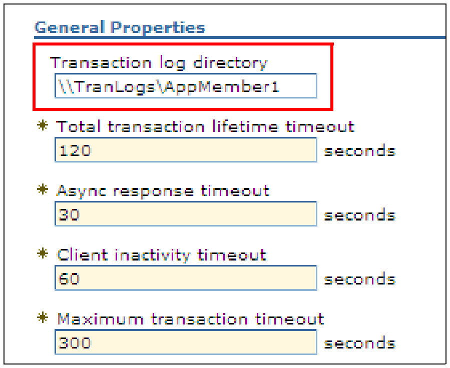


Figure 5-28 Transaction log directory

> **Tip:** If you are using NFS, it is suggested to use 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 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, perform the following steps:

1. In the administrative console, expand **Servers** and click the **Clusters** link.
2. Click the link for the cluster to 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 5-29.

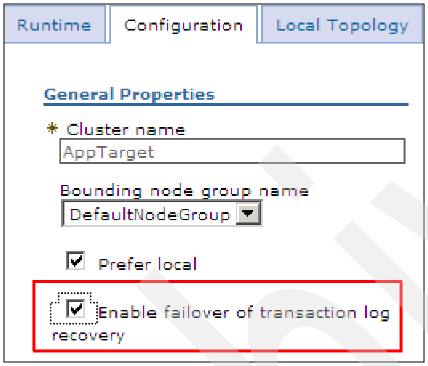


Figure 5-29 Transaction log recovery failover

4. Click **OK**.
5. Save the changes to the master configuration.
6. Wait for automatic synchronization to complete and click **OK**, or manually synchronize the nodes.
7. 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 8, “Advanced production topologies” on page 229.

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, perform the following steps:

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 5-30.

Select	Name	Description	Policy type	Match criteria
☐	Clustered TM Policy	TM One-Of-N Policy	One of N policy	type=WAS_TRANSACTIONS
☐	Default SIBus Policy	SIBus One-Of-N Policy	One of N policy	type=WSAF_SIB

Figure 5-30 Transaction manager policy link

6. Examine the properties of the policy. See Figure 5-31.

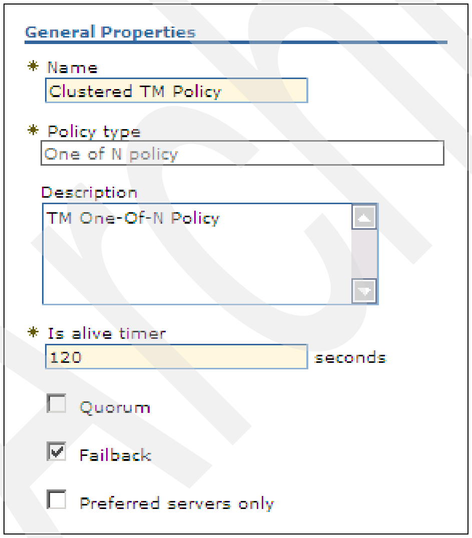


Figure 5-31 Default transaction manager policy

7. Click **Cancel**.

5.4.5 Installing the sample application

This Redbooks publication provides a sample vehicle loan process created for the fictitious company ITSOBank. You can use this vehicle loan process to test the topology you have built in this chapter.

For more information about the vehicle loan process, refer to Chapter 4, “Business scenario used in this book” on page 77. To obtain the additional material supplied with this book, refer to Appendix A, “Additional material” on page 449.

When you have obtained the additional materials, navigate to the Scenarios\WPS\EAR directory. From here, you need to 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. Click the ITSO_implApp.ear radio button, and click **OK**. Click **Next**.
3. On Step 1: Select installation options, click **Next**.
4. On Step 2: Map modules to servers, select **Cluster=AppTarget** and **server=webserver1**, select the ITSO_implWeb check box, and click **Apply**, then click **Next**.
5. On Step 3: Summary click **Finish**.
6. When you see the message “Application ITSO_implApp installed successfully”, click the **save** link, then **OK**.
7. Repeat the process for the ITSOApp.ear file.

You can check 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 5-22.

Example 5-22 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. Login to Business Process Choreographer console. In our environment we used the following URL:

   ```
   http://itsodmgr/bpc
   ```

8. Click **My Process Templates**, select **TestLoanProcess** 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 uninstalled 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
```

In the above command, <AppName> is the name of the application you wish to uninstall.

5.4.6 Installing and configuring Business Space powered by WebSphere

Now that we have a working topology we can also install the Business Space application. We will need to create some tables in a database. This database can be one of the existing ones or a newly created one. We will choose to add the Business Space tables to the WPRCSDB database and we will use the same schema name (COMMONDB), but you can also choose your own schema name.

The process to install the Business Space database involves the following three procedures:

- Installing the Business Space database
- Configuring the widget repository endpoints. See page 156.
- Installing the Business Space service. See page 158.

Installing the Business Space database

Perform the following steps to install the Business Space database:

1. In the database machine, login as the instance owner for the WPRCSDB database. You will need to 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 COMMONDB. 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 against the WPRCSDB database:

```
db2 connect to WPRCSDB
db2 -tf createTable_BusinessSpace.sql
db2 connect reset
```

We have now completed the database part, so log off the database server.

Configuring the widget repository endpoints

Before we install the service we need to modify the endpoints to the widget repository because they are configured as relative URLs (for stand alone use) and we have a clustered environment. This work is done on each node.

1. Login to wpsNode01 and on the command line, change to the correct folder with the following command:

```
cd /opt/ibm/WebSphere/ProcServer/BusinessSpace/wps/registryData
```

2. Edit the files wpsEndpoints.xml and bpcEndpoints.xml. Change the relative URL to an absolute one. In each file look for the lines marked with the tags <tns:url>...</tns:url> and add the Web server. For example, in wpsEndpoints.xml the line looks like Example 5-23.

Example 5-23 wpsEndpoints.xml before adding the Web server

```
<tns:url>rest/bpm/brules/v1</tns:url>
```

Change it to appear as shown in Example 5-24:

Example 5-24 wpsEndpoints.xml with the Web server added

```
<tns:url>http://itsodmgr/rest/bpm/brules/v1</tns:url>
```

Save the changes.

3. Create a new folder and copy these files to it:

```
mkdir -p
/opt/ibm/WebSphere/ProcServer/profiles/Custom01/BusinessSpace/regisr
tyData
cp wpsEndpoints.xml bpcEndpoints.xml wpsWidgets.xml
/opt/ibm/WebSphere/ProcServer/profiles/Custom01/BusinessSpace/regisr
tyData
```

4. Repeat these editing and copying steps on the other node.

Now we can install the service itself.

Installing the Business Space service

Perform the following steps to install the Business Space service.

1. Open a browser and login to the administrative console. First, create a data source for the service.
2. Navigate to **Resources** → **Data sources**. Change the Scope to **Cluster=RMSgold.Support** then click **New.**
3. In the Enter basic data source information window (Figure 5-32), enter the following values:
   - Data source name: Business Space
   - JNDI name: jdbc/bpm/BusinessSpace
   - Component-managed authentication alias and XA recovery authentication alias: WPS_Auth_Alias (we are using the WPRCSDB database.)

   Click **Next**.

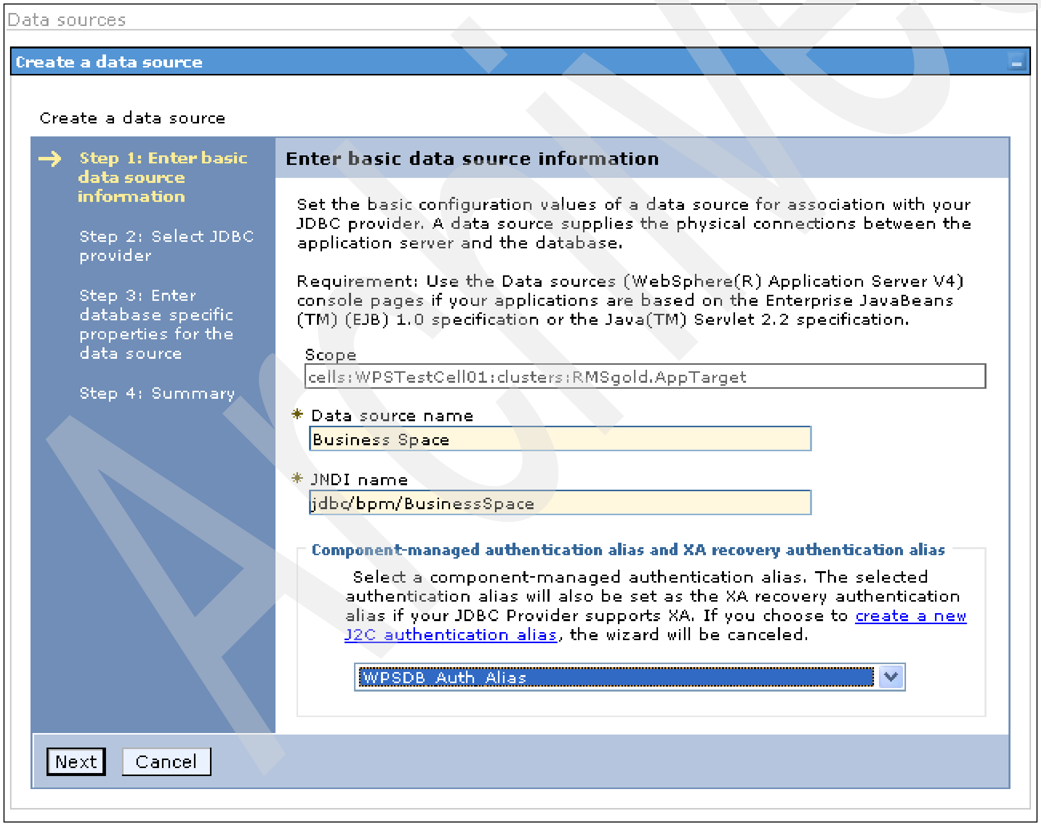


Figure 5-32 The enter basic data source information window

4. In the Select JDBC provider window (Figure 5-33), click Select an existing JDBC Provider and click DB2 Universal JDBC Provider (XA) from the drop-down menu. Click **Next**.

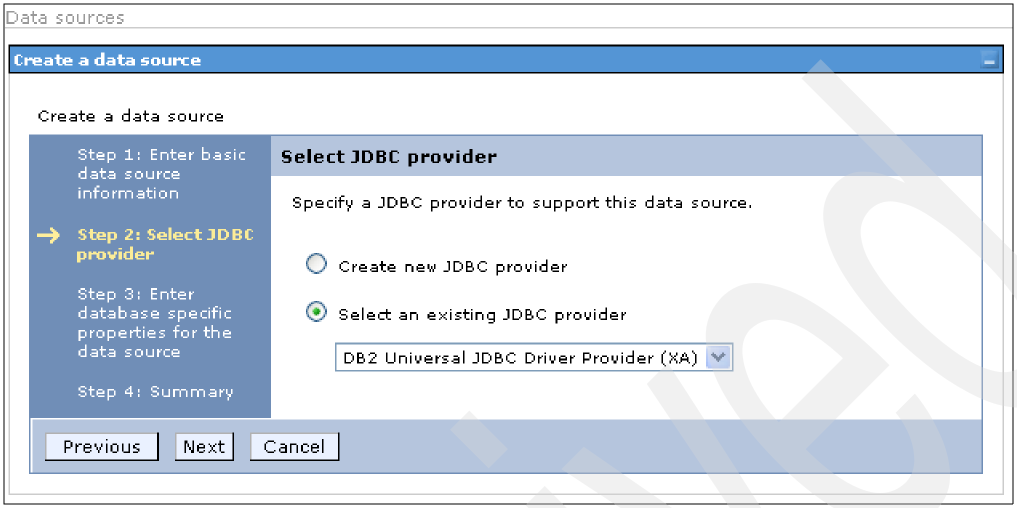


Figure 5-33 The Select JDBC Provider window

5. Enter the following values in the Enter database specific properties for the data source window (Figure 5-34):
   – Database name: WPRCSDB
   – Server name: itsodb2
   – Port number: 50000

   Leave the check box at the bottom selected. Click **Next**.

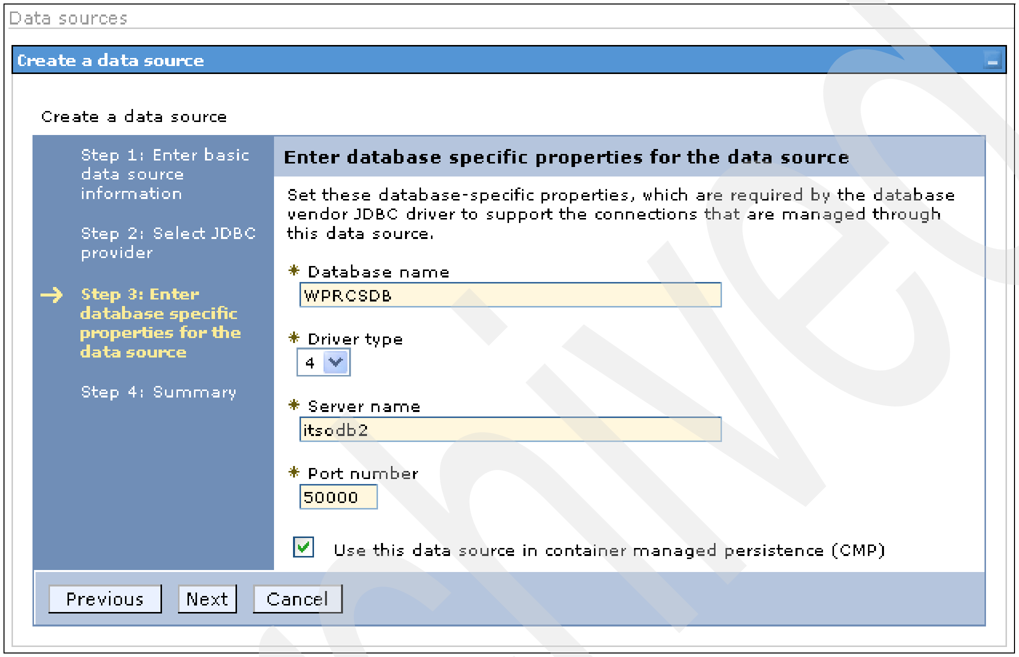


Figure 5-34 The Enter database specific properties for the data source window

6. Click **Finish** and save your changes.
7. Navigate to **Resources** → **Data sources**. Change the Scope to **Cluster=RMSgold.Support** then click the check box next to our new data source and click **Test connection**. You should have a successful result.
8. Navigate to **Servers** → **Clusters** and click **RMSgold.Support**. This is the cluster where we will deploy the Business Space application.
9. Under Business Integration, click the **Business Space Configuration** link.

10. In the General Properties window (Figure 5-35), click the Install Business Space service check box. Use COMMONDB for the Schema name. The Existing Business Space data source should already be filled in with Business Space (our newly created data source). Click **WBI_DataSource** from the Create Business Space data source using list. Click **OK**.

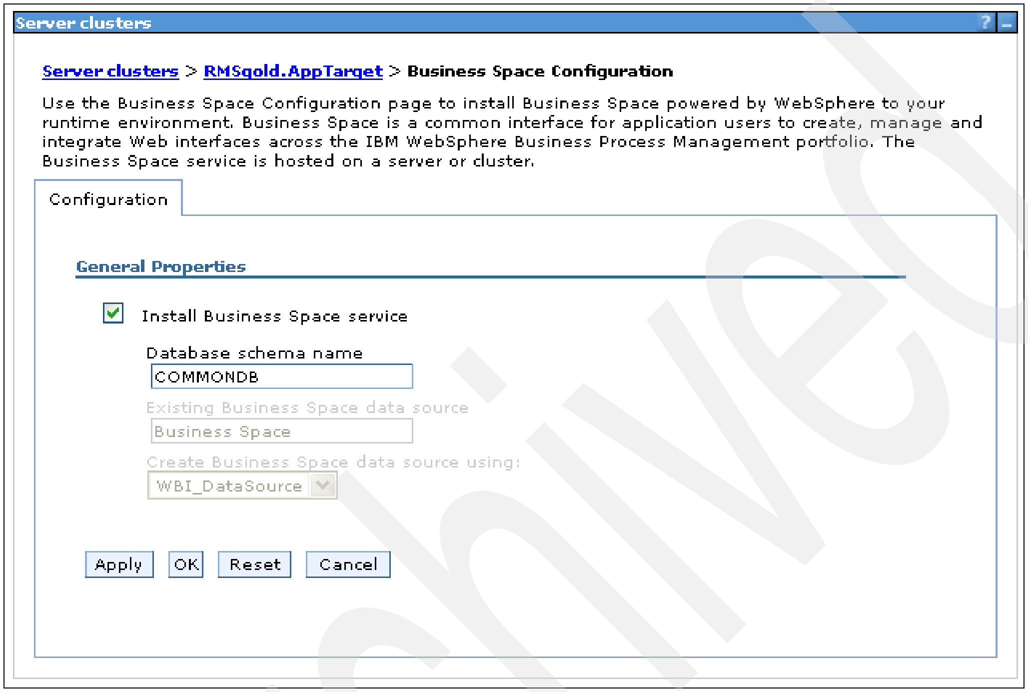


Figure 5-35 The Install Business Space window

11. The service should be installed. Save and synchronize the changes.
12. Start the deployment environment by navigating to **Servers** → **Deployment Environments**. Click the RMSgold check box and click **Start**. Check the SystemOut.log files for any errors connecting to the databases.
13. Run a JACL script to ensure that the business rule widgets are available in Business Space at runtime in a network deployment environment. Login to the deployment manager and run the following commands.

```
cd /opt/ibm/WebSphere/ProcServer/bin
./wsadmin.sh -f installBRRestAPI.jacl -clusterName RMSgold.Support
```

14. Save and synchronize your changes.

15. Map the modules for the new applications to the Web server. Navigate to **Applications** → **Enterprise Applications**. Stop the four new applications:

   – BusinessRuleRestAPI_RMSgold.Support
   – BusinessRulesManager_RMSgold.Support
   – BusinessSpaceManager
   – IBM_BSPACE_WIDGETS

   For each of these applications perform the following steps:

   a. Click the link.
   b. Click **Manage Modules**.
   c. Select all the modules.
   d. Under Clusters and Servers, pick both the Support Cluster and the Web server
   e. Click **Apply**, then **OK**.

   Verify your changes by looking at the Server column in the table. The following information should be listed:

   ```
   WebSphere:cell=WPSCell01,node=testdmgr-node,server=webserver1
   WebSphere:cell=WPSCell01,cluster=RMSgold.Support
   ```

16. Save and synchronize the changes. Now you can start these applications.

17. Propagate the plug-in to the Web server. Navigate to **Servers** → **Web servers**. Select the webserver1 check box and click **Propagate Plug-in**.

You should now be able to run the application at http://itsodmgr/BusinessSpace. Try the user Administrator. Further information about the use of Business Space can be found on developer Works at the following Web page:

http://www.ibm.com/developerworks/websphere/library/techarticles/0807_fasbinder2/0807_fasbinder.html

The Information Center for Business Space can be found at the following Web page:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.websphere.wps.612.doc/doc/tcon_config_bspace.html

5.4.7 Other applications

This section describes how to start the following applications:

- Business Process Choreographer Explorer. See following section.
- Business Process Choreographer Observer. See page 164.
- Business Rules Manager. See page 164.

Business Process Choreographer Explorer

The Business Process Choreographer Explorer can be viewed using a browser with the following URL:

```
http://itsodmgr/bpc
```

A sample window is shown in Figure 5-36.

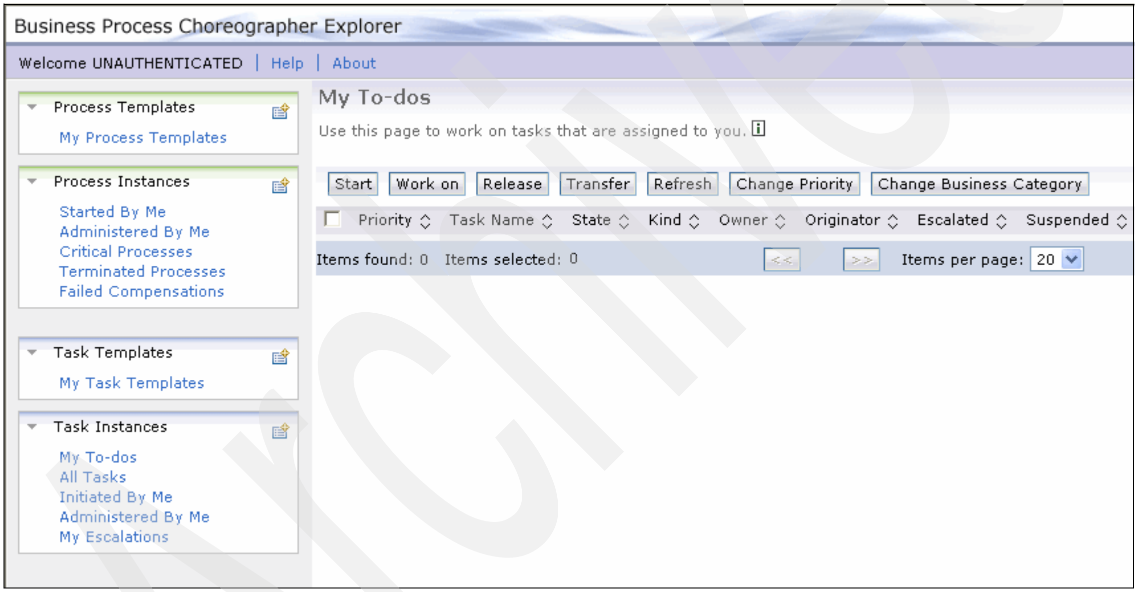


Figure 5-36 The Business Process Choreographer Explorer page

Business Process Choreographer Observer

The Business Process Choreographer Observer can be viewed using the following URL:

```
http://itsodmgr/bpcobserver
```

A sample window is shown in Figure 5-37.

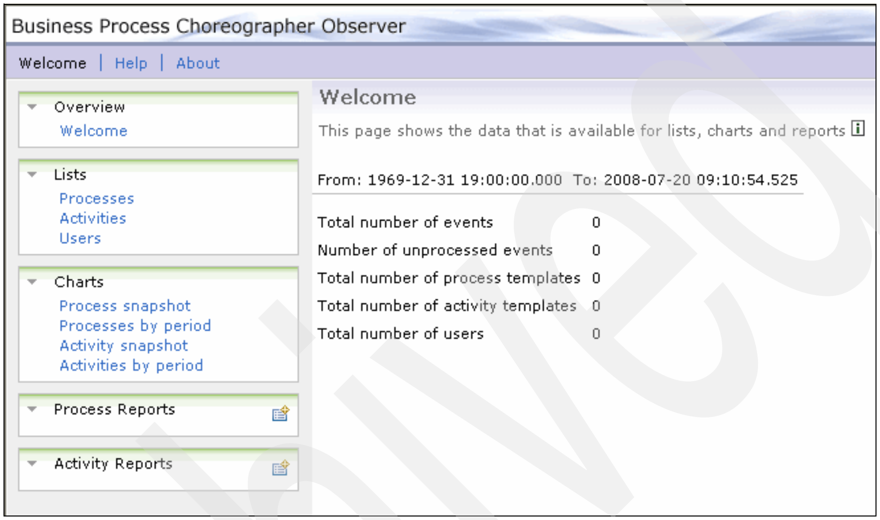


Figure 5-37 The Business Process Choreographer Observer page

Business Rules Manager

The Business Rules Manager can be viewed using the following URL:

```
http://itsodmgr/br
```

A sample window is shown in Figure 5-38.

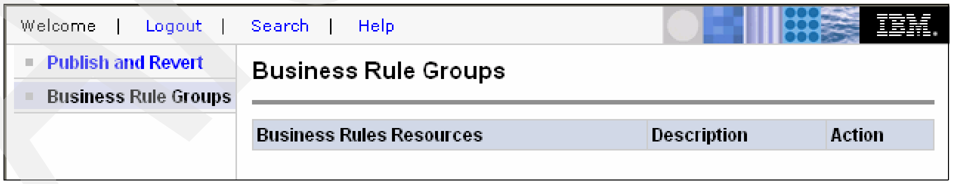


Figure 5-38 The Business Rules Manager page

6

Configuring a custom topology

This chapter provides instructions for creating a custom topology, which includes Remote Messaging only. We will not be using 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.

6.1 Prerequisites to creating the topology

We will start from the same position as the Remote Messaging and Remote Support topology. That is, we assume the base product has been installed but no profiles have been created.

6.1.1 Creating the databases within DB2

We need the following three databases in this topology:

- Common database
- Business Process Choreographer (BPC) database
- Messaging engine database (with three schemas)

We 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 Section 5.1.4, “Create the common database” on page 97 to create the following databases:

- WPRCSDB
- BPEDB
- MEDB

Use the same instructions to create the following schemas:

- BPCME
- SCASYS
- SCAAPP

6.1.2 Creating 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 Section 5.2.1, “Creating a deployment manager profile” on page 104.
- To create a deployment manager profile silently, see Section 5.3.2, “Creating a deployment manager profile” on page 139.

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 Creating the node profiles

There are two options to create node profiles.

- To create node profiles using the profile management tool (the graphical option), see Section 5.2.2, “Creating the node profiles” on page 116.
- To create node profiles silently, see Section 5.3.3, “Creating the node profiles” on page 140.

6.1.4 Creating the 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. Login to the administrative console and navigate to **Servers** → **Clusters**. Click **New**.
2. Enter AppTarget as the name of the cluster and click **Next** (Figure 6-1).

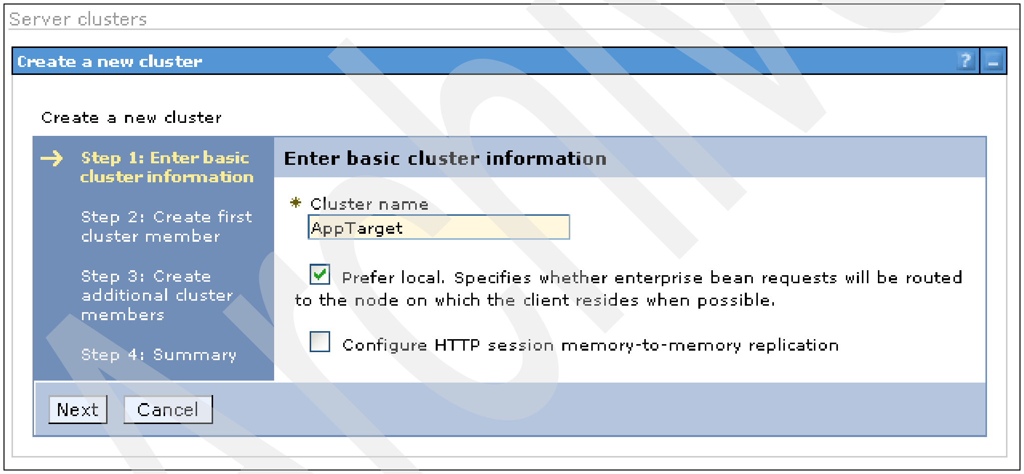


Figure 6-1 The Enter basic cluster information window

3. In the create first cluster member window (Figure 6-2), 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.
   e. Click **Next**.

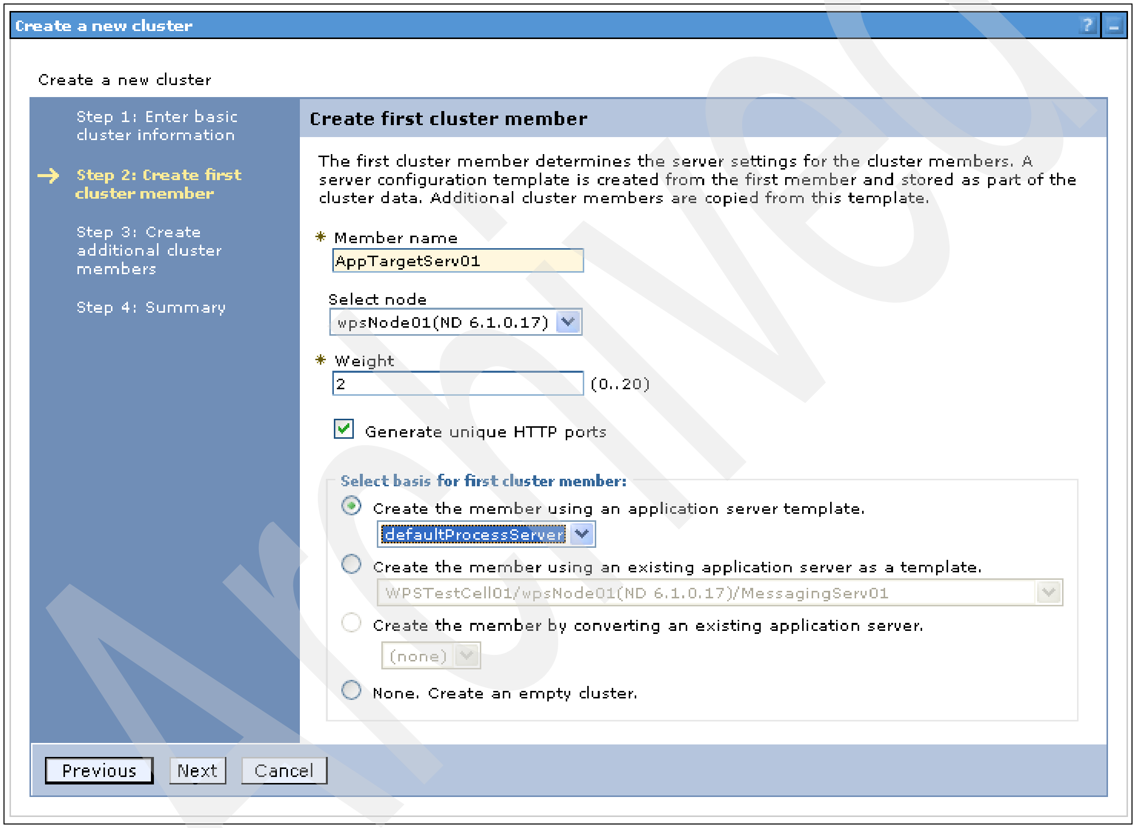


Figure 6-2 The create first cluster member window

4. In The Create additional cluster members window (Figure 6-3), enter AppTargetServ02 for the Member name, select wpsNode02, and click the **Add Member** button. Click **Next**.

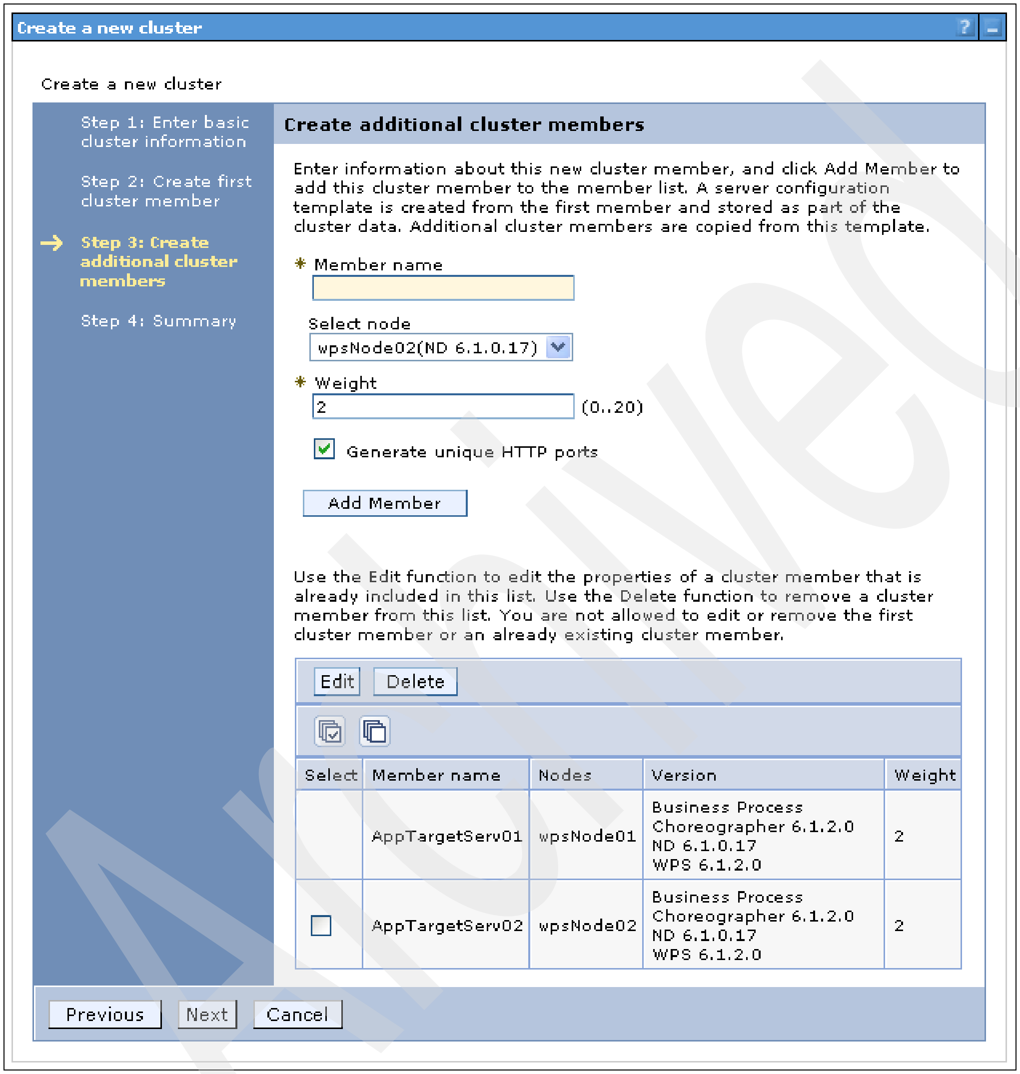


Figure 6-3 The Create additional cluster members window

5. In the Summary window click **Finish**. The new cluster will be created.

Repeat these steps to create a cluster called Messaging and a cluster called Support each with two members. Save and synchronize your changes.

6.2 Using the custom topology wizard

We are now in a position to run the custom topology wizard in the administrative console (Figure 6-4).

1. Login 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 myCluster, and click **Next**.

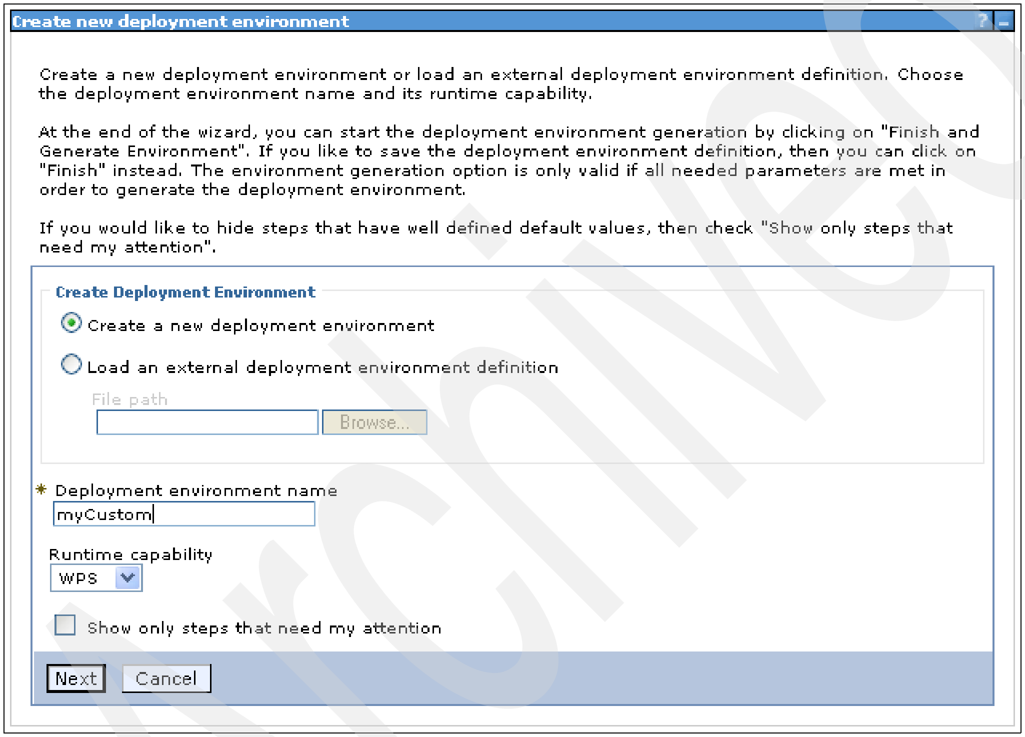


Figure 6-4 The Create new deployment environment window

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 6-5 we have added the AppTarget and Messaging clusters and are about to add the Support 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.

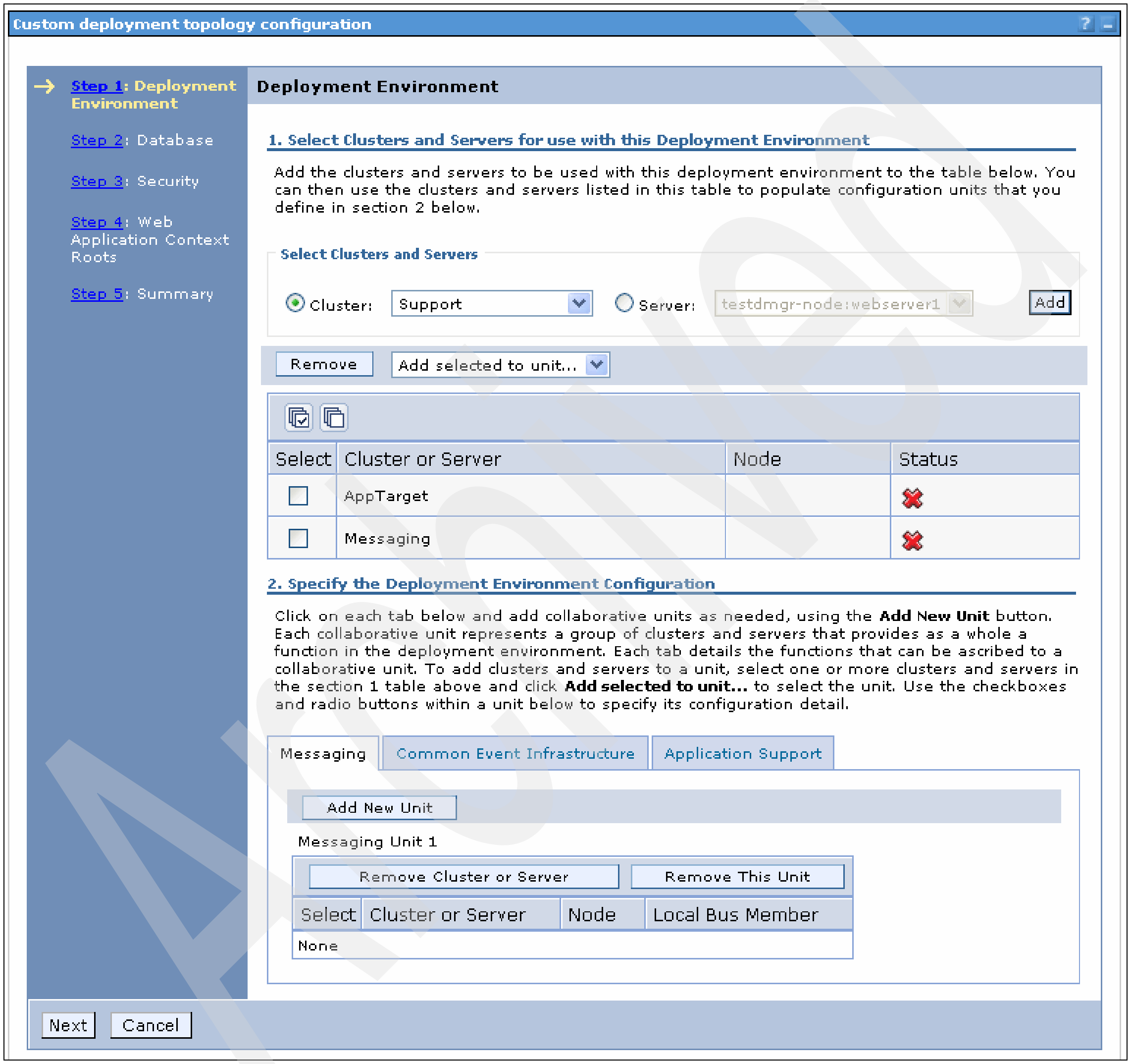


Figure 6-5 Two clusters added and the third about to be added

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.
   c. Click the Local Bus Member radio button for the Messaging cluster as shown in Figure 6-6.

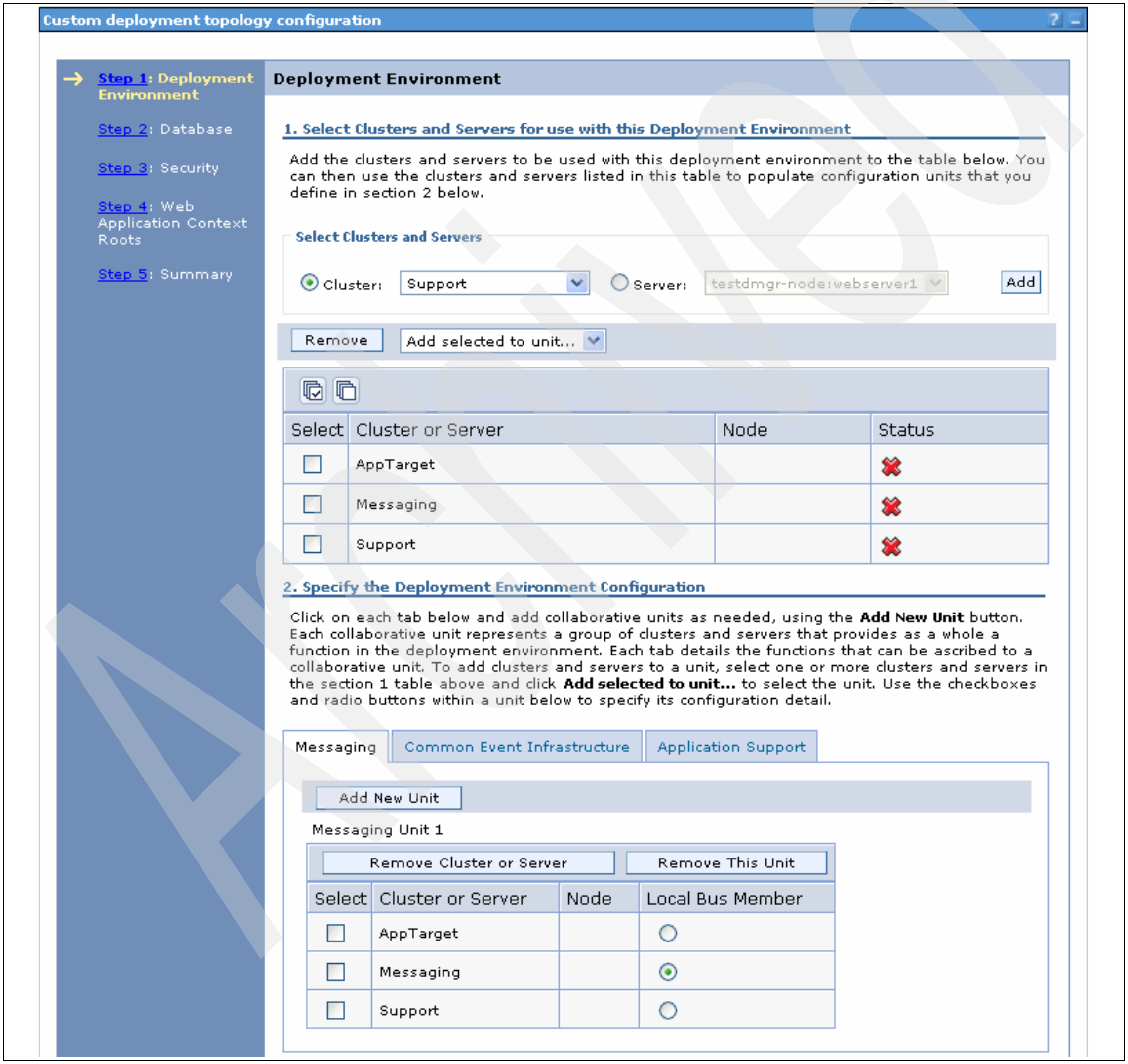


Figure 6-6 All three clusters in the Messaging Unit

7. Click the Common Event Infrastructure tab. Select all three clusters and select **Common Event Infrastructure Unit 1** as the unit.
8. Click the Application Support tab. Select all three clusters and select **Application Support Unit 1** as the unit. The page will refresh.
9. Enable **Service Component Architecture** on the AppTarget and Messaging clusters. Note that this enables further buttons within the window.
10. Enable **Business Process Choreographer Container** on the AppTarget cluster. This will enable further options within the window.
11. Enable **Business Process Choreographer Explorer** and **Business Rules Manager** on the Support Cluster. There is no option for Business Process Event Collector or BPC Observer because we have not used CEI.

    The completed window is shown in Figure 6-7.
12. Click **Next**.

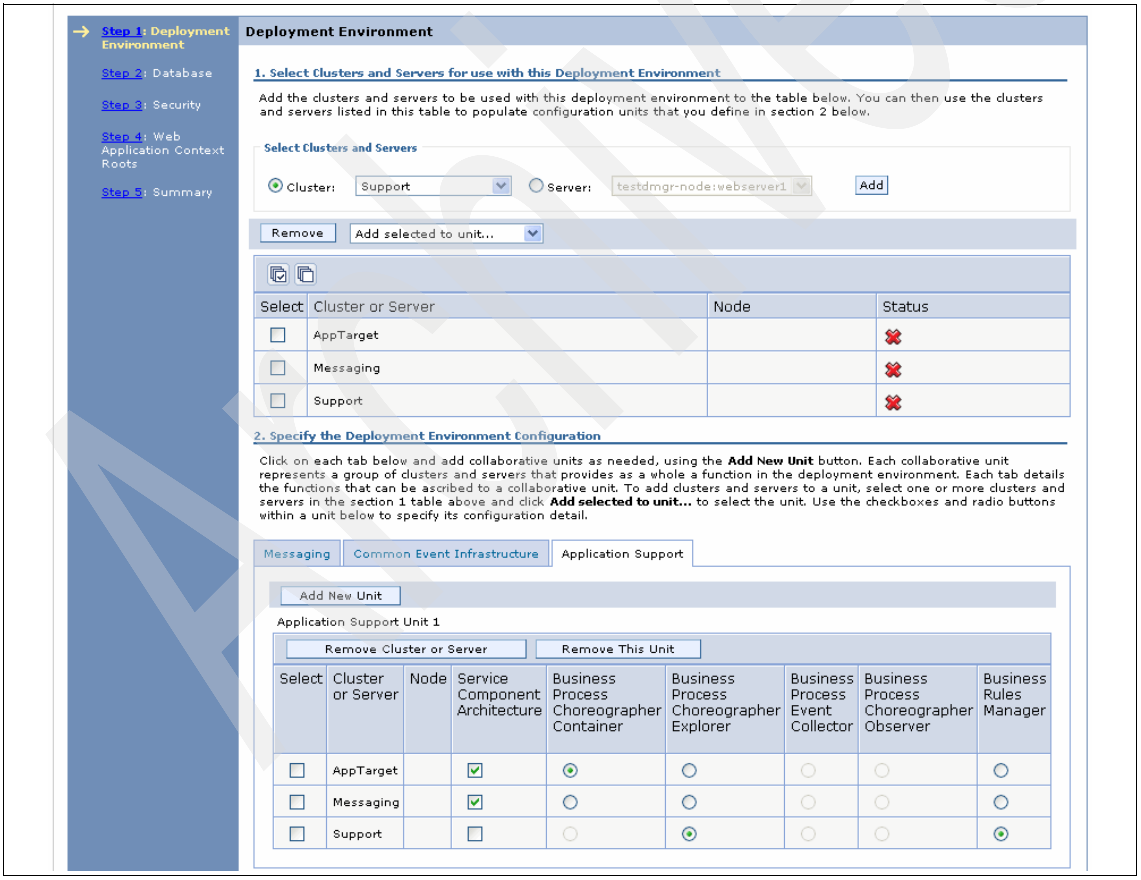


Figure 6-7 Defining the components across the environment

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 all check boxes in the **Create Tables** column are cleared, as shown in Figure 6-8. Click **Next**.

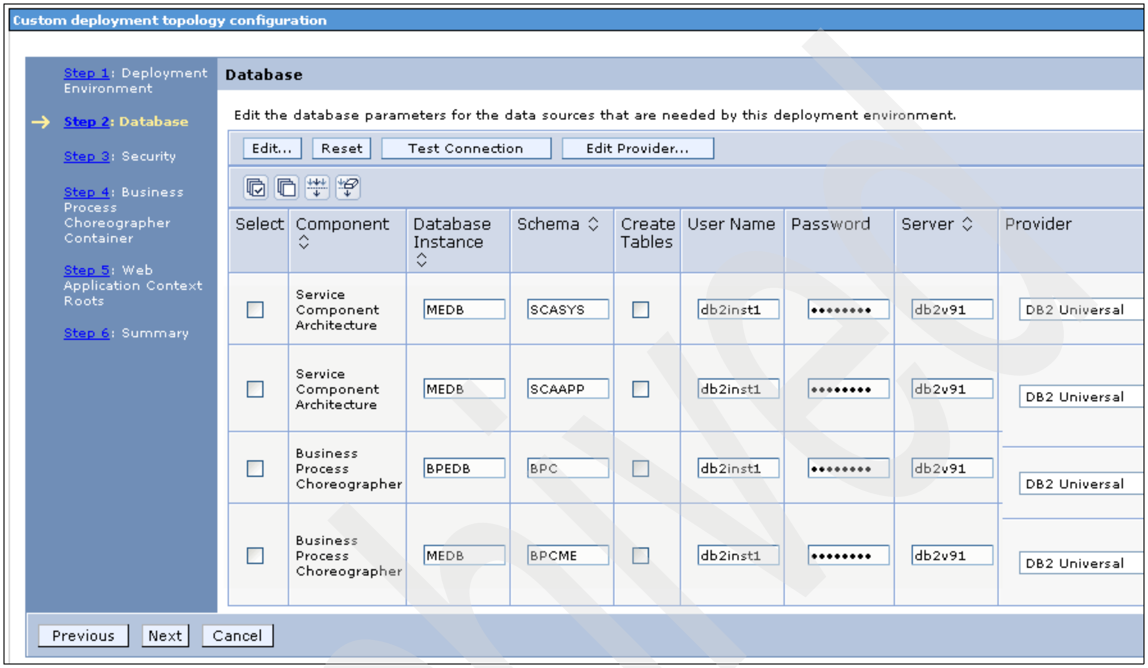


Figure 6-8 The database window

14. Click the **Step 3: Security** link. In the Security window (Figure 6-9), enter the following values:

– User: SCA
– Password: passw0rd

Click **Next**.

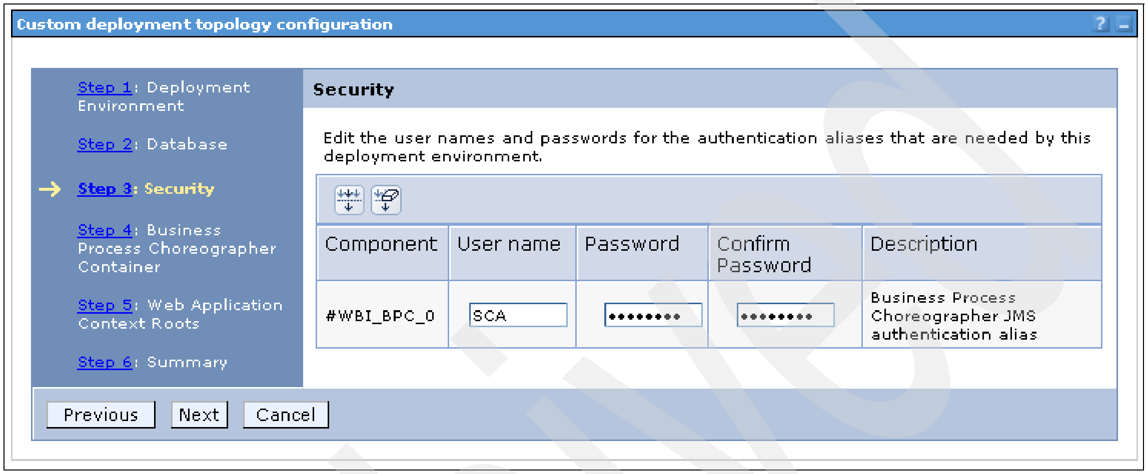


Figure 6-9 The Security window

15. In the Business Process Choreographer Container window (Figure 6-10 on page 176), enter the following details:

   a. For the Administrator role, use the following values for 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 User and Password:
      - User: jmsapi
      - Password: passw0rd

d. For the Escalation User authentication, use the following values for User and Password:

- User: escalation
- Password passw0rd

Later on, when we enable LDAP, these users and groups must in the LDAP database. Clear the Enable e-mail service check box. The completed window is shown in Figure 6-10.

Click **Next**. The Web Application Context Roots window will appear.

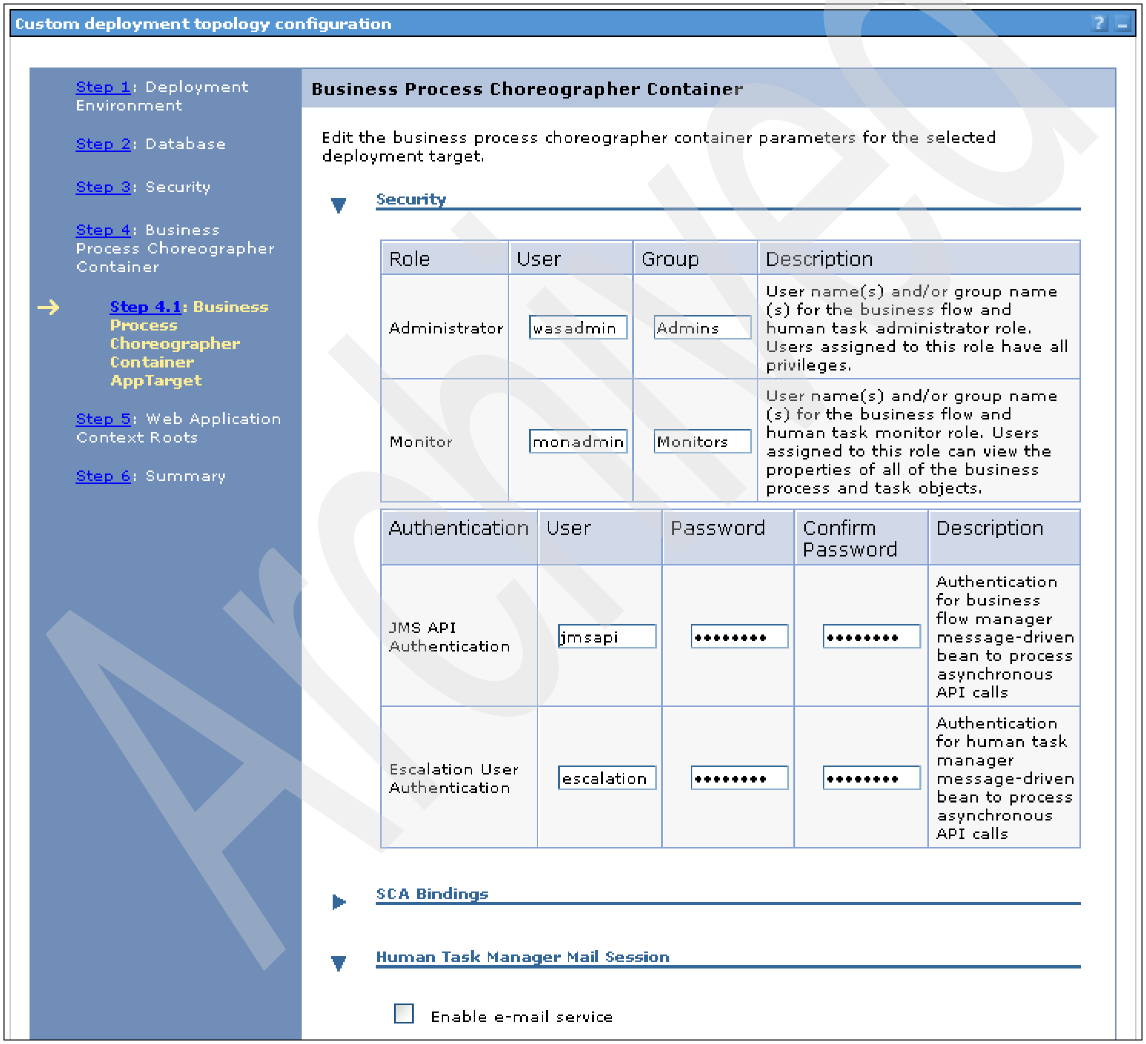


Figure 6-10 The Business Process Choreographer Container window

16. Click **Next**, because there is nothing to change in the Web Application Context Roots window. Note there is no BPC Observer available. The Summary window will appear.

17. Click **Finish**. The Generate a progress window will be displayed as shown in Figure 6-11. Save and synchronize your changes.

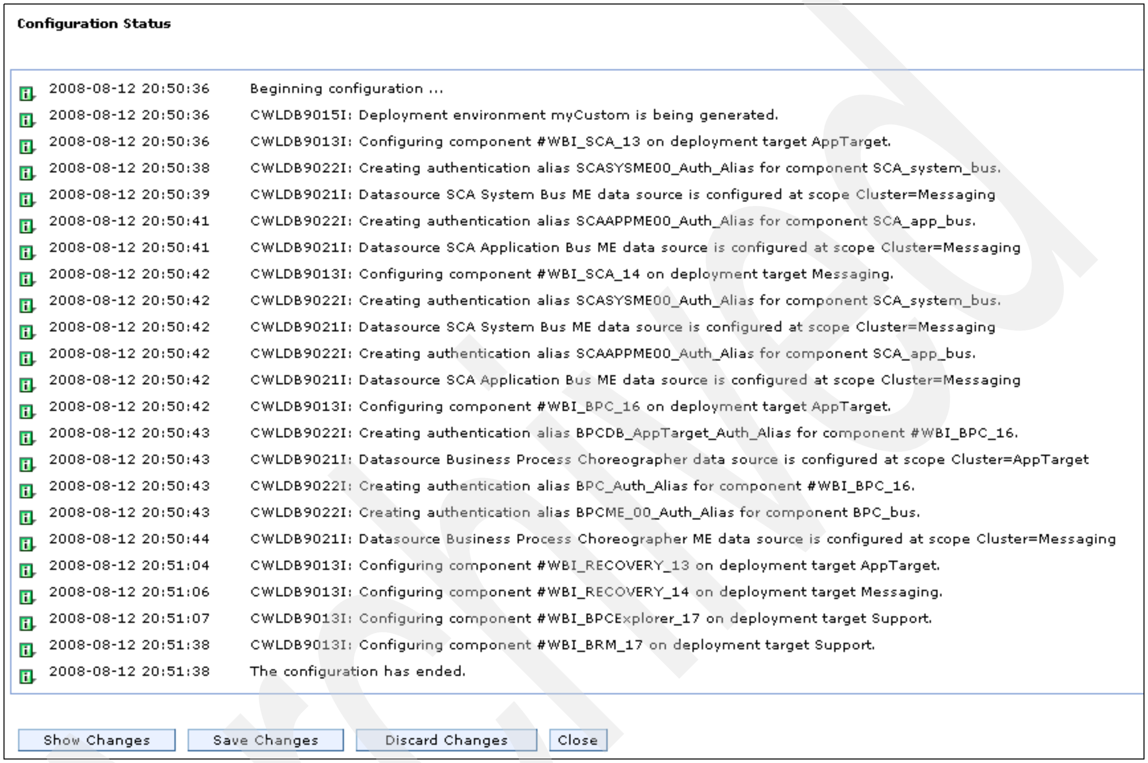


Figure 6-11 The result of the custom deployment

6.2.1 Making required post-creation changes

You have now created a custom deployment topology, but before we start, we need to make some post-creation changes just as we have done for the Remote Messaging and Remote Support topology:

1. Define the WebSphere Variable DB2UNIVERSAL_JDBC_DRIVER_PATH 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: Messaging, Support, then AppTarget.

You can verify some of your topology in 5.4, “Post-creation configuration and verification” on page 147. However, you cannot try the BPC Observer because we do not use CEI in this topology.

7

Securing and administering a production topology

This chapter addresses securing and administering WebSphere Process Server for the Remote Messaging and Remote Support topology pattern. As the primary administrator, you will configure security and learn how to address certain reoccurring operational concerns. These concerns encompass promoting consistent configurations from test to production, changing database passwords in production, maintaining WebSphere Process Server's database tables, and handling failed events.

7.1 Securing a BPM production topology

A Business Process Management (BPM) infrastructure must be properly secured. Out of the box, WebSphere Process Server comes secured with a file registry, SSL enabled and configured with central key management, messaging infrastructure roles assigned, database access configured, and the Integrated Solutions Console secured.

This is a good start. You will need to adapt this configuration to your companies security policies and current infrastructure. This includes encrypting communications with processes external to your WebSphere Process Server cell, configuring your companies user repositories, and mapping groups to administrative roles.

7.1.1 Setting up SSL infrastructure

In WebSphere Process Server v6.1.2. 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 to 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 will need to exchange signer certificates with the other cell.

7.1.2 Choosing the User Account Repository

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. This 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:** Anytime you are using more that one machine, the local operating system user account registry is not supported.

7.1.3 Configuring LDAP

Before starting the LDAP configuration in WebSphere Process Server, be sure that the Tivoli Directory Server is running.

You will need to take steps to configure LDAP as a federated repository.

1. Create the LDAP repository definition.
   a. In the Integrated Solutions Console expand **Security**. Click **Secure administration, applications, and infrastructure**.
   b. Change the Available realm definitions to **Federated repositories**.
   c. Click **Configure**.
   d. Click **Add Base entry to Realm** (Figure 7-1).

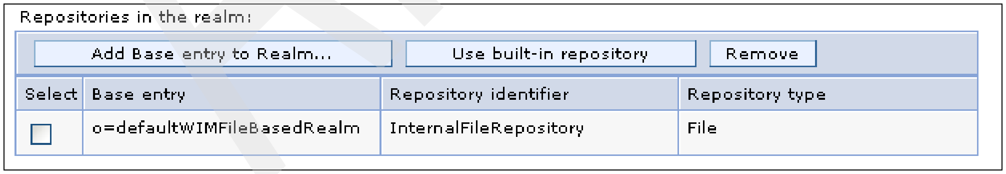


Figure 7-1 Repositories in the realm

e. Click **Add Repository** and add the values in Table 7-1.

Table 7-1 LDAP repository values

Entry	Value
Repository identifier	RMRS Topology LDAP
Directory Type	Tivoli Directory Server Version 6
Primary host name	itsodb2
Login properties	uid

f. Click **OK** and **Save.** Values are shown in Figure 7-2.

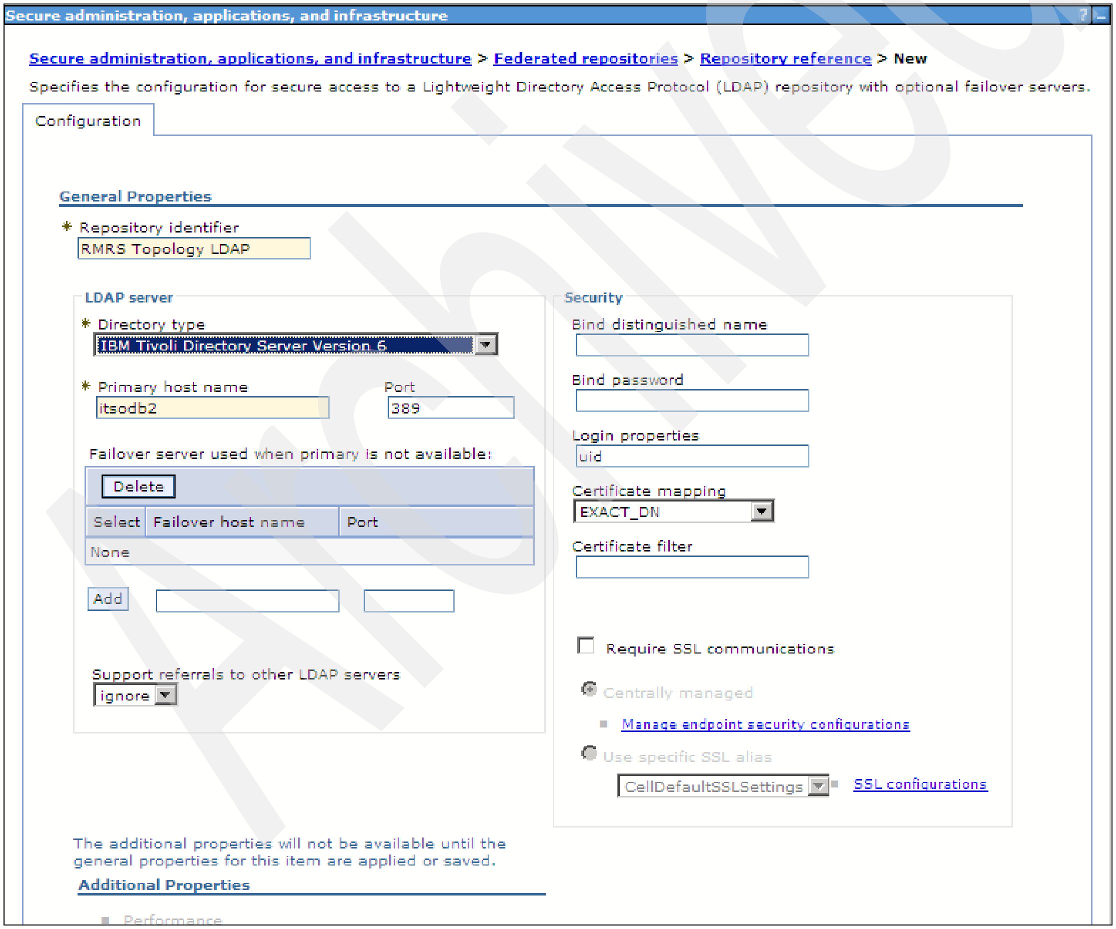


Figure 7-2 LDAP definition window

2. Enter the listed values for the each text box below. See Figure 7-3.
   - Distinguished base entry that uniquely identifies this set of entries in the realm: O=RMRSLDAP.
   - Distinguished name of base entry in this repository: O=IBM

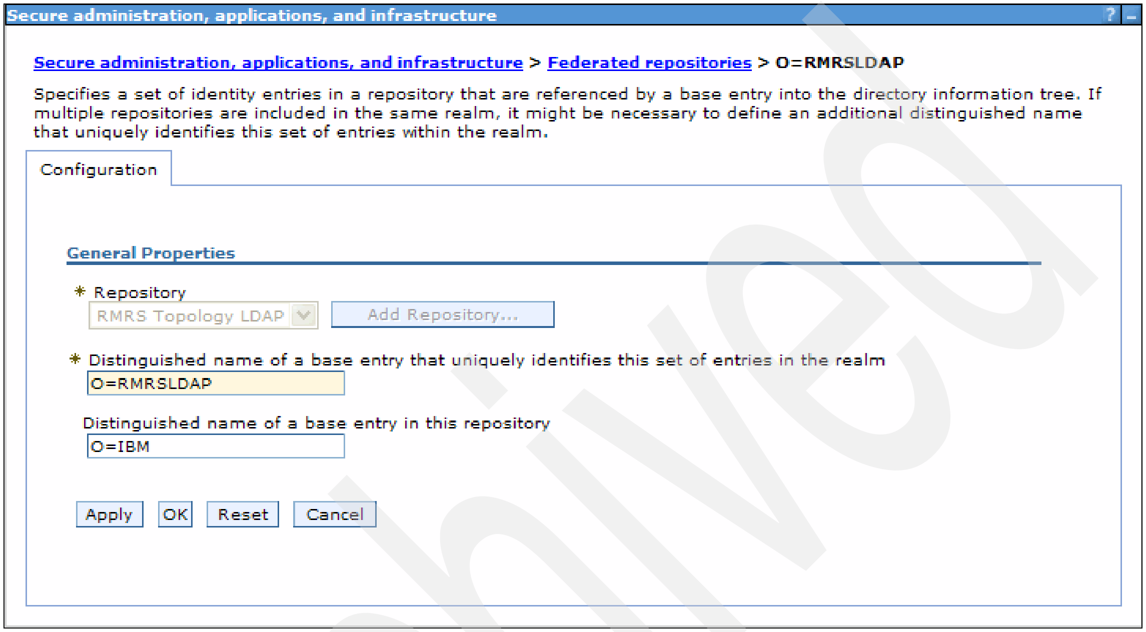


Figure 7-3 Repository Reference

3. Click **OK** and click **Save**.

4. Configure the federated repository.

   a. Select the defaultWIMFiledBasedRealm check box as shown in Figure 7-4.

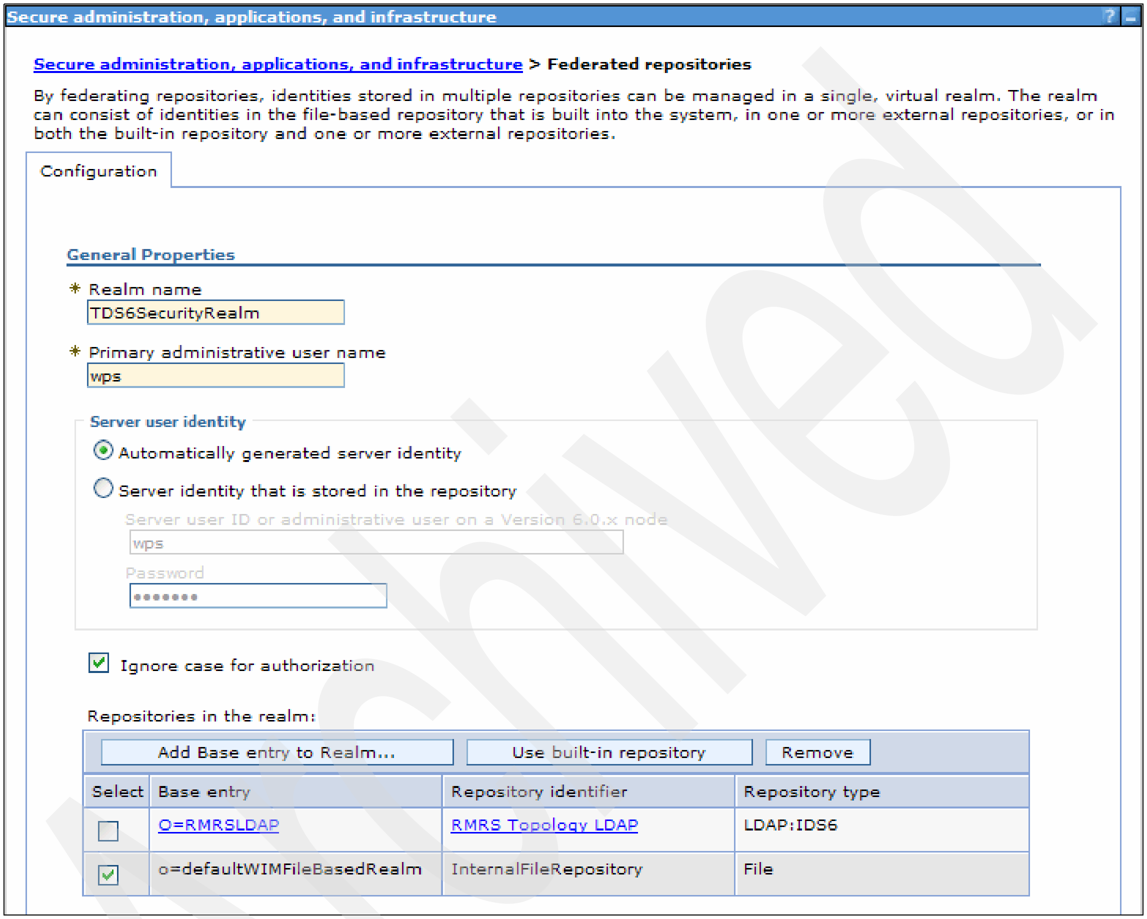


Figure 7-4 Remove defaultWIMFileBasedRealm

   b. Click **Remove**.

   c. Enter the values for the text boxes below:

      - Realm name: TDS6SecurityRealm
      - Primary administrative user name: wps

   d. Select the Automatically generate server identity radio button.

   e. Click **OK** and **Save**.

7.1.4 Enabling administrative security with 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.
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 application are granted custom permissions check box, to debug any initial problems. Make sure your Service Component Architecture (SCA) modules you are deploying are Java 2 security ready. Your window should resemble Figure 7-5.

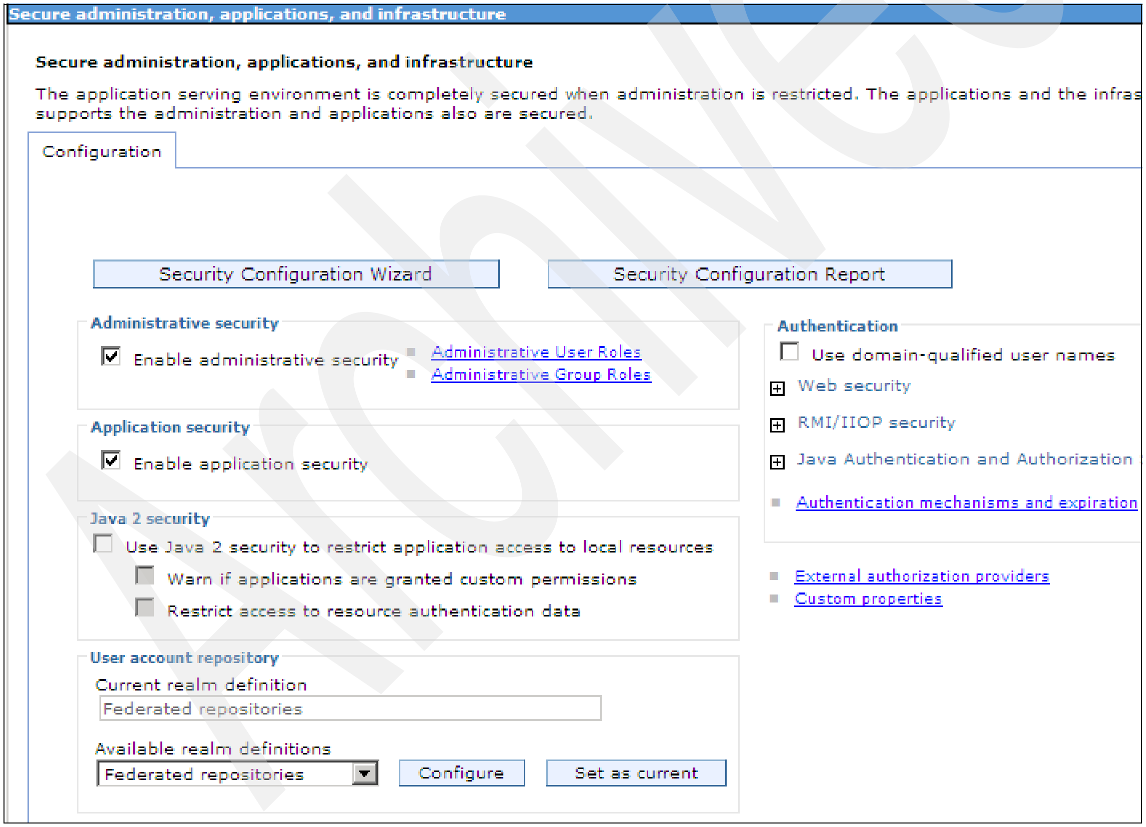


Figure 7-5 Global Security window

6. Click **Apply** and **Save**.

7. Click **System administration** → **S ave Changes to Master Repository**.
   a. Check **Synchronize changes with Nodes**.
   b. 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 which has to be run from the out of synchronization node's profile with the node in a stopped state.

8. Restart your node managers and deployment manager.
9. Perform the following steps to verify your configuration by querying a user from the LDAP repository:
   a. Open the Integrated Solutions Console and login as wps, the Primary Admin ID.
   b. Click **Users and Groups** → **Manage Users**.
   c. Enter tom or another user from your populated repository.
   d. Click **Search**.

   The user tom should return an entry as shown in Figure 7-6.

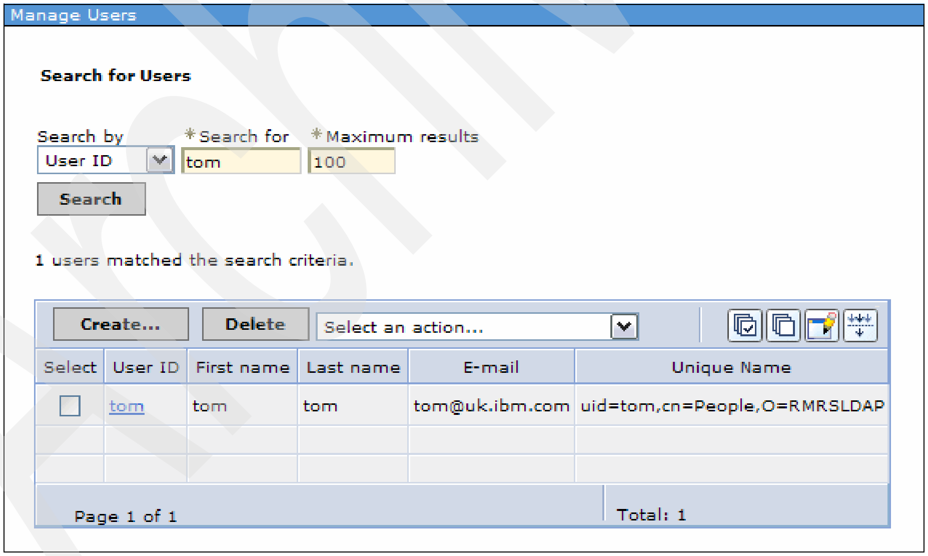


Figure 7-6 Search for Users

> **Note:** It is suggested to map groups to the administrative roles, thus limiting the number of people using the primary admin identity.

7.1.5 Configuring the Service integration bus

Perform the following steps 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 7-7 shows that all of the buses are secured.

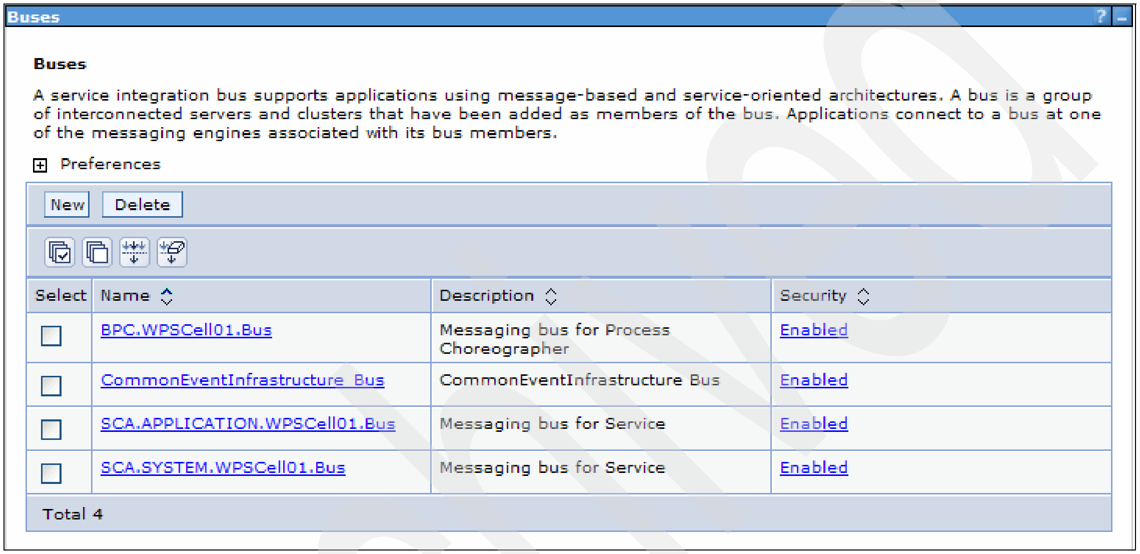


Figure 7-7 Buses window

4. Click **Enabled** for any of the buses to see the security configuration shown in Figure 7-8.

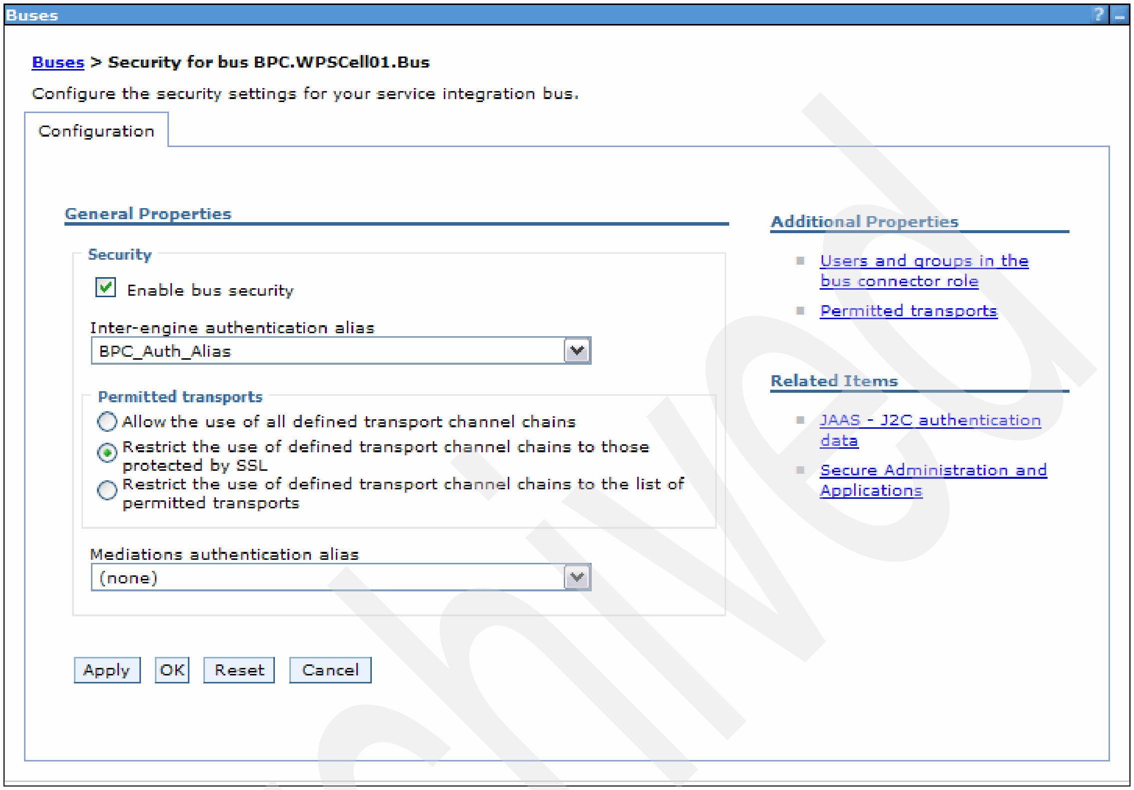


Figure 7-8 BPC Bus security window

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.WPSCell01.Bus → Users and groups in the bus connector role**. This window, shown in Figure 7-9, allows you to add and delete users and groups from this role.

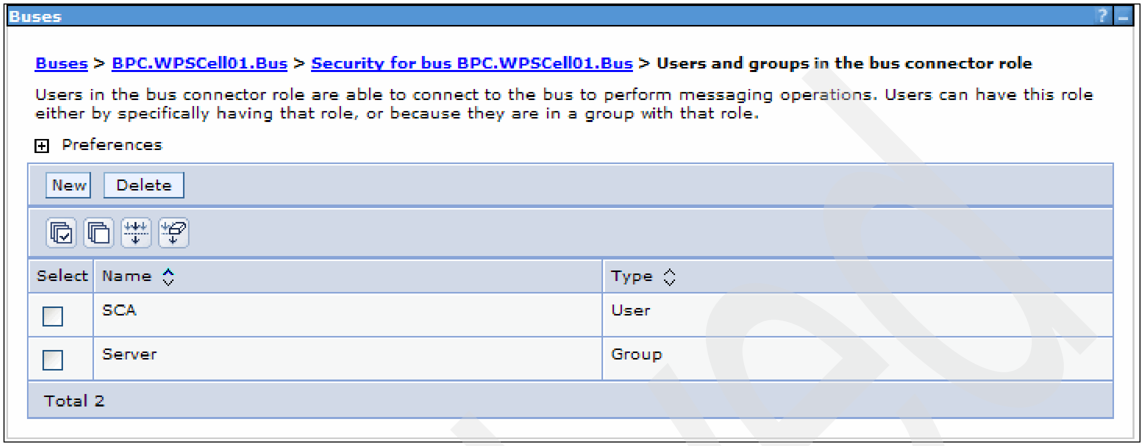


Figure 7-9 Users and groups in the bus connector role

8. Click **Add**. This will launch the Create user or group in the bus connector role window shown in Figure 7-10. In this window, you may grant permissions to an existing user or group.

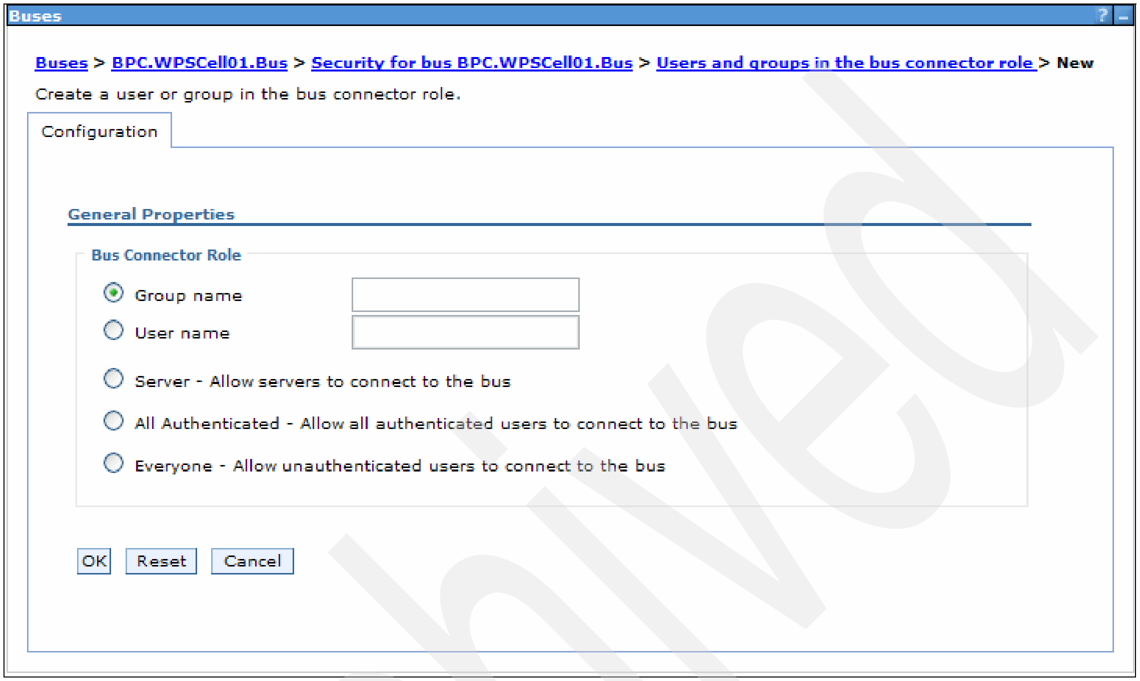


Figure 7-10 Create a user or group in the bus connector role

9. Click Group name or the appropriate radio button.
10. 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 2.5.1, “Creating a secured link between two cells” on page 45.

7.1.6 Map groups to administrative roles

To configure the Administrative User and Group Roles, you need to login as either wps or the server’s primary admin ID. This will give you the authority to map other groups and users to roles. The first role you should assign is the adminsecuritymanager, because this will allow you delegate authority without sharing the primary administrative user name and password.

In this Redbooks publication, you want to map the admins group to be administrators. Users in this group will have nearly full administrator privileges. The only access they will not have is to map users and groups to administrative roles. Using Table 7-2, map the groups to roles.

Table 7-2 Groups for administrative roles

Group	Role	Description
admins	Administrator	All users in the admins group will have access to change anything in the cell, but administer 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

Perform the following steps 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 7-11 on page 192. To assign multiple roles to a particular group, press Ctrl and click the role.
6. Click **OK** and **Save**.

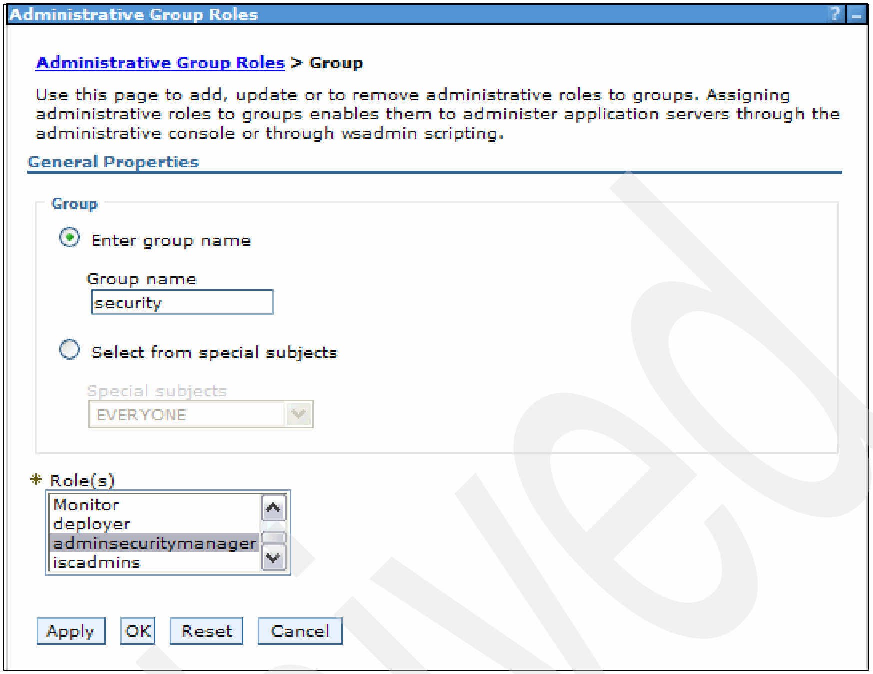


Figure 7-11 Add administrative Group Roles

Repeat for the other two groups in Table 7-2 on page 191. Your administrative groups roles window should look like Figure 7-12.

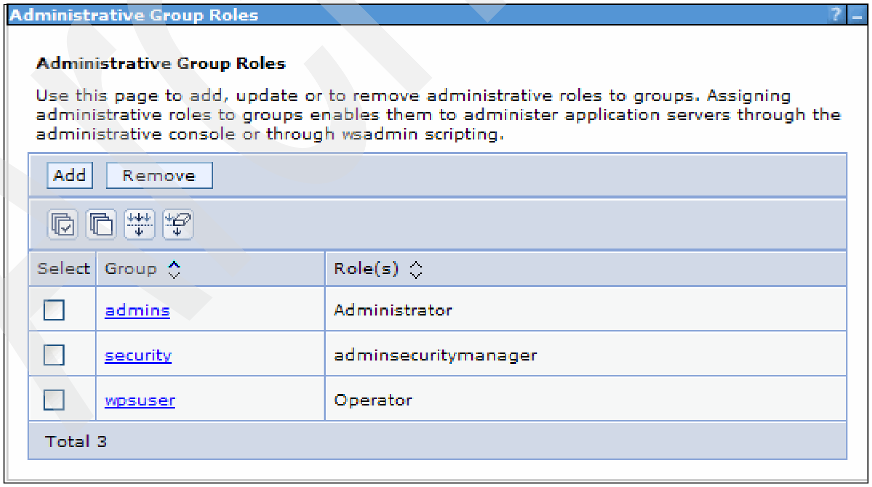


Figure 7-12 Administrative Groups Roles window

7.1.7 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 Section 2.2, “Security for a WebSphere Process Server solution” on page 29. To use the security roles to user/groups mapping feature in the Integrated Solutions Console, you will need to be in a group that is assigned the administrative role of either configurator or administrator. Login to the Integrated Solutions Console.

1. Click **Applications** → **Enterprise Applications** → **BPEContainer RMSGold.AppTarget** → **Security role to user/group mapping**.
2. Check the select box for a role. In this example you will use BPEAPIUser.
3. Click **Look up users** or **Look up groups**. In this example, use **Look up groups**.
4. Either enter ***** or a specific value to the search box.
5. Click **Search**.
6. Highlight wpsuser or your user or group.
7. Click **>>** to add the group to the role**.** The window shown in Figure 7-13 on page 194.

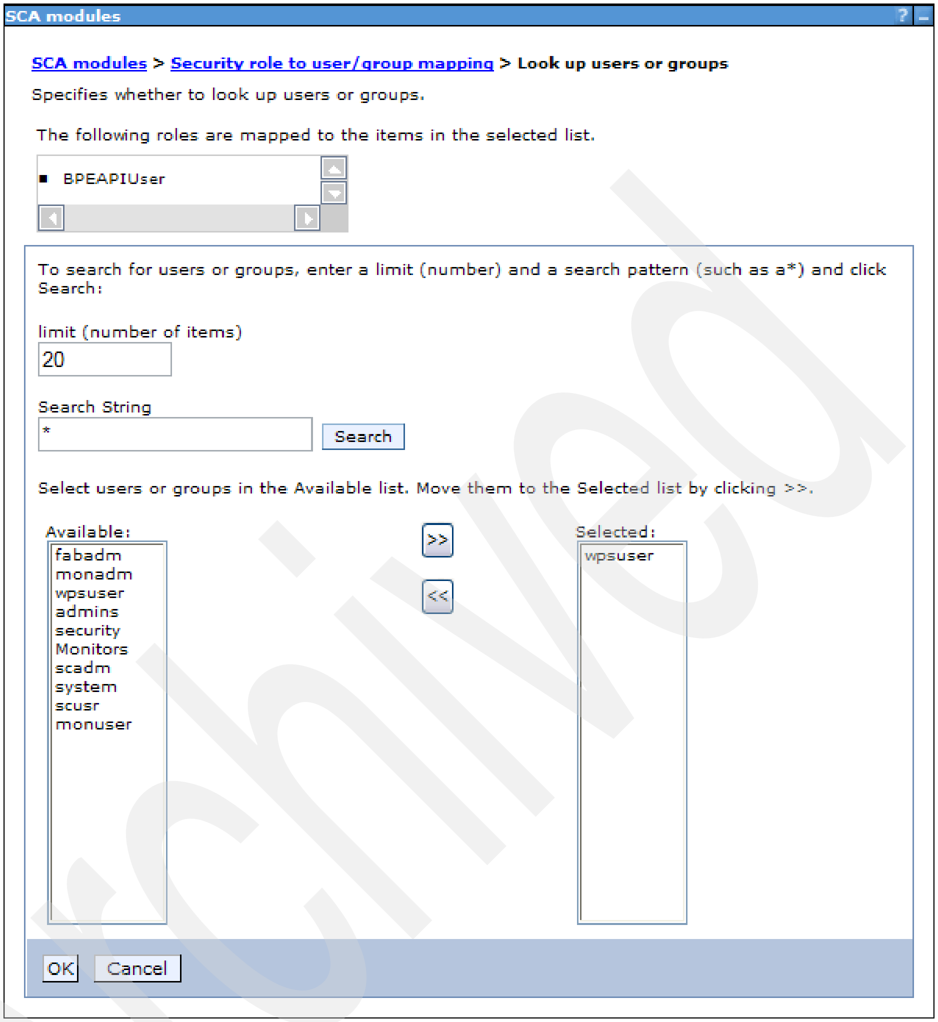


Figure 7-13 Look up users and groups window

8. Click **OK**.

Once this is complete, the group wpsuser has been added to the BPEAPIUser role. These are the same steps you would use for each set of roles below.

> **Important:** In the rest of this section, you will see the default permissions 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 7-3 on page 195 for an example based on our sample ldap.

Table 7-3 Table to use to secure your management applications

Application	Security Role	Administrator assigned users or groups
BPEContainer_<deploymentEnvironment.cluster>	BPESystemAdministrator	admins
	BPESystemMonitor	wpsusers
	BPEAPIUser	wpsusers
	WebClientUser	wpsusers
	JMSAPIUser	jmsapi
TaskContainer_<deploymentEnvironment.cluster>	TaskSystemAdministrator	admins
	TaskSystemMonitor	wpsusers
	TaskAPIUser	wpsusers
	EscalationUser	escalation
BPCExplorer_<deploymentEnvironment.cluster>	WebClientUser	wpsusers
BPCObserver_<deploymentEnvironment.cluster>	ObserverUser	admins
BusinessSpaceManager	administrator	admins
BusinessRulesManager_<deploymentEnvironment.cluster>	BusinessRuleUsers	wpsusers
Event Service	eventAdministrator	admins
	eventConsumer	wpsusers, admins
	eventUpdater	wpsusers, admins
	eventCreator	admins
	catalogAdministrator	admins
	catalogReader	admins

Business Process Choreographer

BPC consists of multiple J2EE Enterprise Application Archives (EAR).

- Two Container EARs
- Five Management application EARs

Mapping roles for the container EARs

Figure 7-14 shows the roles available for the BPEContainer EAR. Perform the following steps to map roles for the BPEContainer EAR.

1. Click **Applications** → **Enterprise Applications** → **BPEContainer_RMSGold.AppTarget** → **Security role to user/group mapping**.
2. Refer to Table 7-3 on page 195 to map your groups to the roles.

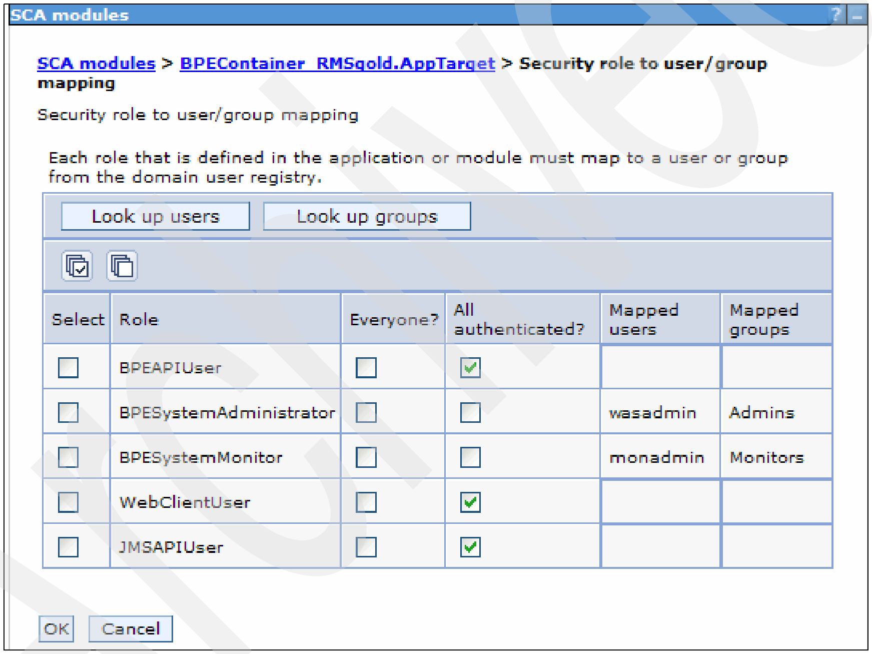


Figure 7-14 BPEContainer roles

> **Note:** You will see that there are users and groups already populated. This occurred during the initial configuration through the wizard

Figure 7-15 shows the roles available for the TaskContainer EAR. Perform the following steps to map roles for the BPEContainer EAR.

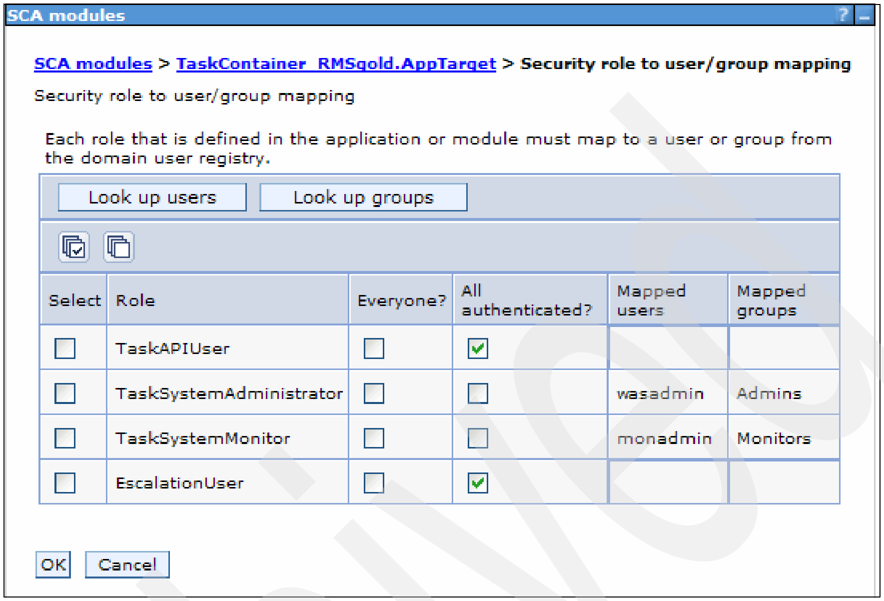


Figure 7-15 Task Container roles

1. Click **Applications** → **Enterprise Applications** → **TaskContainer_RMSGold.AppTarget** → **Security role to user/group mapping**.
2. Refer to Table 7-3 on page 195 to map your groups to the roles.

> **Note:** You can also get to the mapping window for the container application by clicking **Applications** → **SCA Modules** → **TaskContainer RMSGold.AppTarget** → **Security role to user/group mapping**.

Management applications

The BPC Container has five management applications that will give you the flexibility to grant certain groups permissions to certain but not all function. 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 ACLs that you set for your own Business Space, there is internal authorization checking.

  Figure 7-16 shows a single role of administrator. This role has access to administer every single business space in the system, not just their own. Users without this role can only administer their own Business Space.

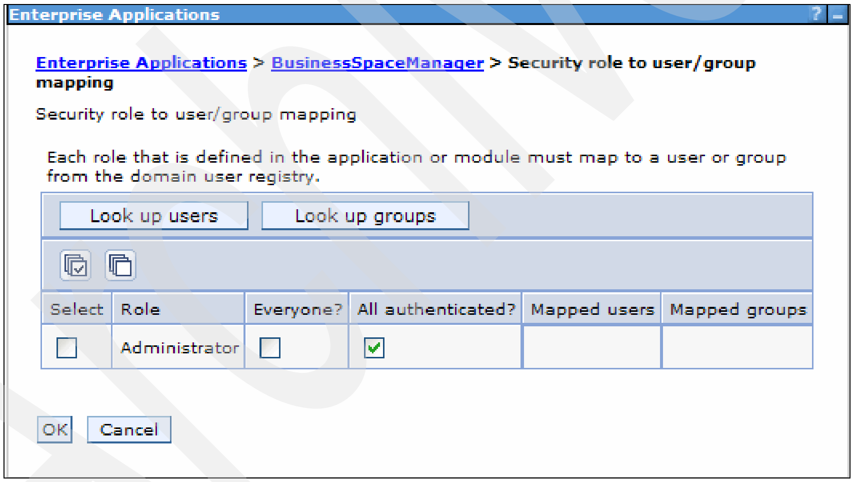


Figure 7-16 Business Spaces roles

1. Click **Applications** → **Enterprise Applications** → **BusinessSpaceManager_RMSGold.AppTarget** → **Security role to user/group mapping**.
2. Refer to Table 7-3 on page 195 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 7-17 shows a single WebClientUser role. Users assigned this role can view and act on only those tasks that have been assigned to you.

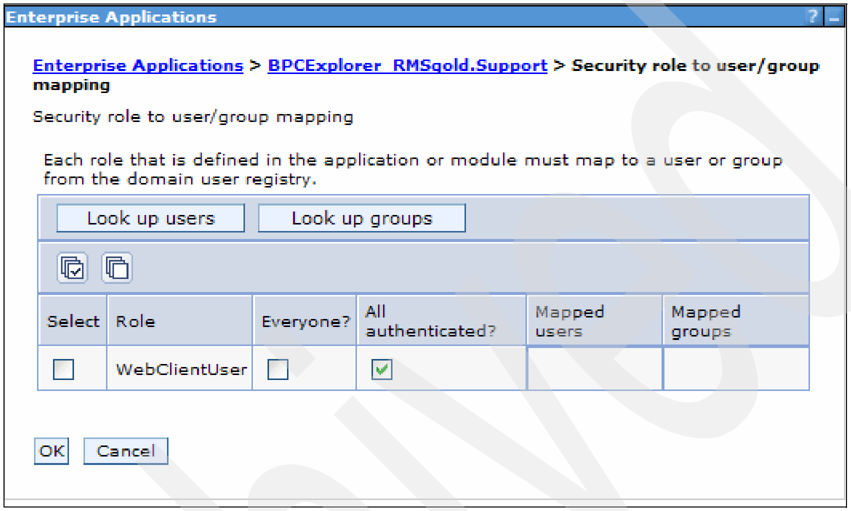


Figure 7-17 BPC Explorer roles

  1. Click **Applications** → **Enterprise Applications** → **BPCExplorer_RMSGold.AppTarget** → **Security role to user/group mapping**.
  2. Refer to Table 7-3 on page 195 to map your groups to the roles.

- BPC Observer

  You can use BPC Observer to create reports on processes that have been completed. You can also use it to view the status of running processes.

  For more information regarding BPC Observer refer to the Information Center article *About Business Process Choreographer Observer*, available at the following Web page:

  http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.websphere.bpc.612.doc/doc/bpc/c2observer.html

Figure 7-18 shows a single ObserverUser role. Users assigned this role will be able to use the application.

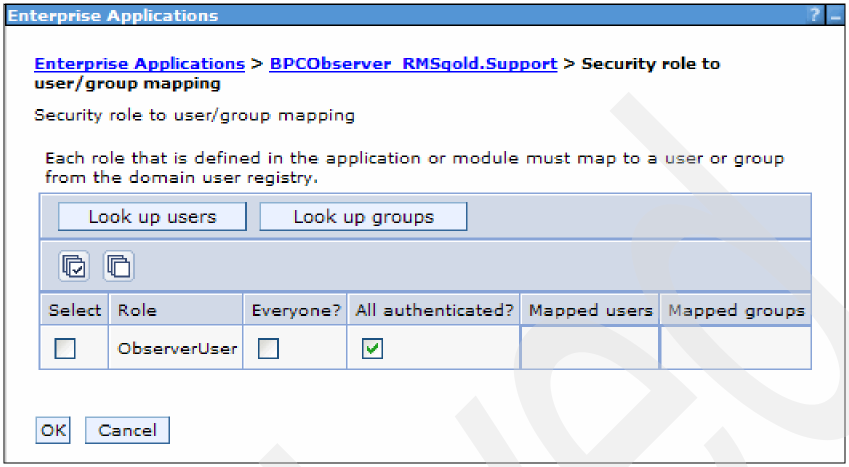


Figure 7-18 BPC Observer roles

1. Click **Applications** → **Enterprise Applications** → **BPCExplorer_RMSGold.AppTarget** → **Security role to user/group mapping**.
2. Refer to Table 7-3 on page 195 to map your groups to the roles.

- Business Rules Manager

  The business rules manager is the main WebSphere Process Server tool that a business analyst uses for rule authoring. For more information of 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/v6r1mx/index.jsp?topic=/com.ibm.websphere.wps.610.doc/doc/cbre_busiru_howbrerworks.html`

  Figure 7-19 on page 201 shows a single Business RulesUser 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 actually 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_RMSGold.AppTarget** → **Security role to user/group mapping**.

2. Refer to Table 7-3 on page 195 to map your groups to the roles.

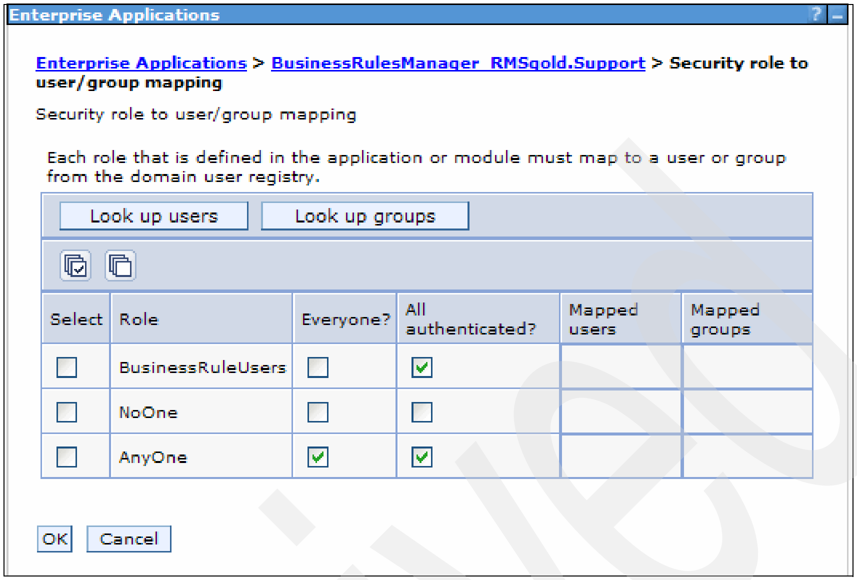


Figure 7-19 Business Rules Manager

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 7-20 shows a single WBIOperator role. Users assigned this role will be able to use the application.

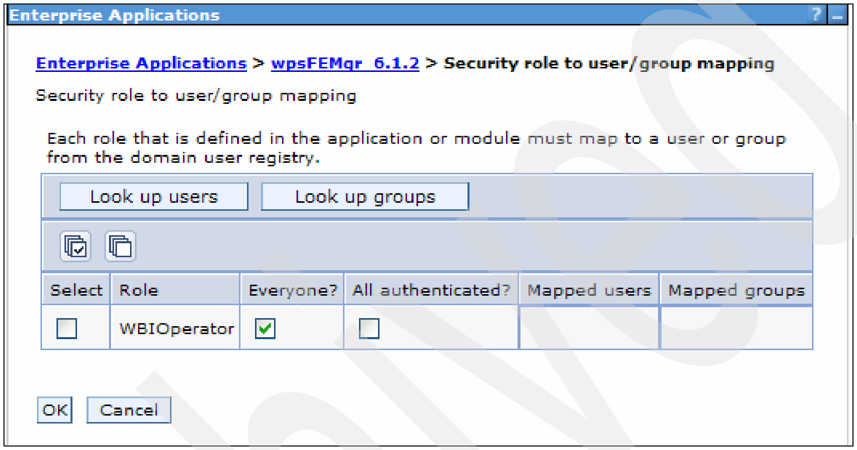


Figure 7-20 Failed Event Manager

1. Click **Applications** → **Enterprise Applications** → **wpsFEMgr_6.1.2** → **Security role to user/group mapping**.
2. Refer to Table 7-3 on page 195 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 7-21 shows multiple roles. Users assigned these roles will gain access to interfaces referenced in Section 2.2.4, “Access control for Common Event Infrastructure container” on page 36.

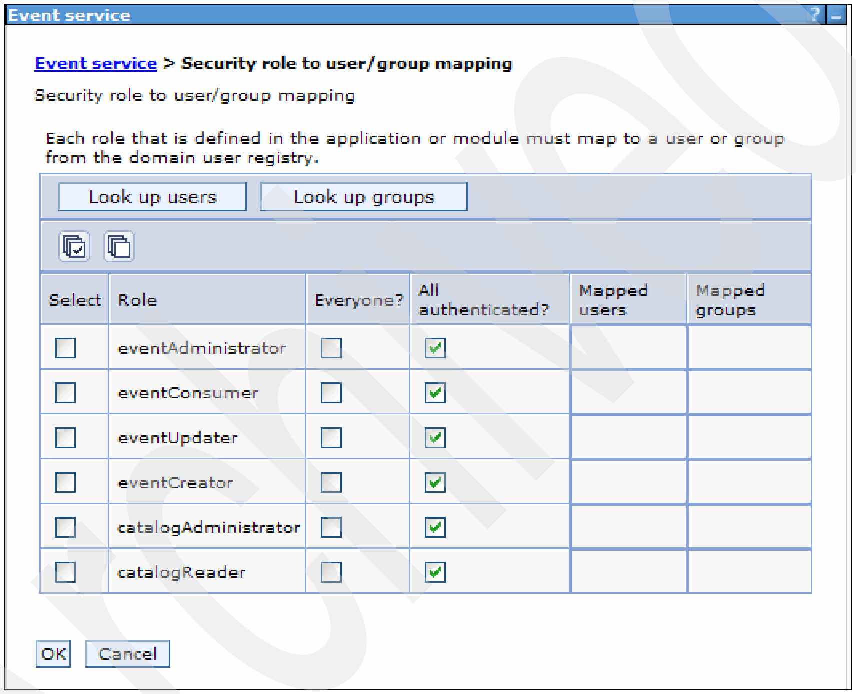


Figure 7-21 Event Service roles

1. Click **Service Integration** → **Common Event Infrastructure** → **Event Service** → **Map security role to user/group mapping**.
2. Refer to Table 7-3 on page 195 to map your groups to the roles.

7.1.8 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 like Table 2-1 on page 24, 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/v6r1mx/index.jsp?topic=/com.ibm.websphere.wps.612.doc/doc/tsec_deploying.html

If the application is communicating with 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.

7.2 Administering a BPM environment

This section provides recommendations for administrators of BPM environments. It contains the following sections:

- “Deployment environments” on page 205
- “Business Process Choreographer” on page 221
- “Common Event Infrastructure” on page 221
- “Changing the database password” on page 222
- “Failed events” on page 224

7.2.1 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 **Server** → **Deployment Environments** to get to the main window (Figure 7-22).

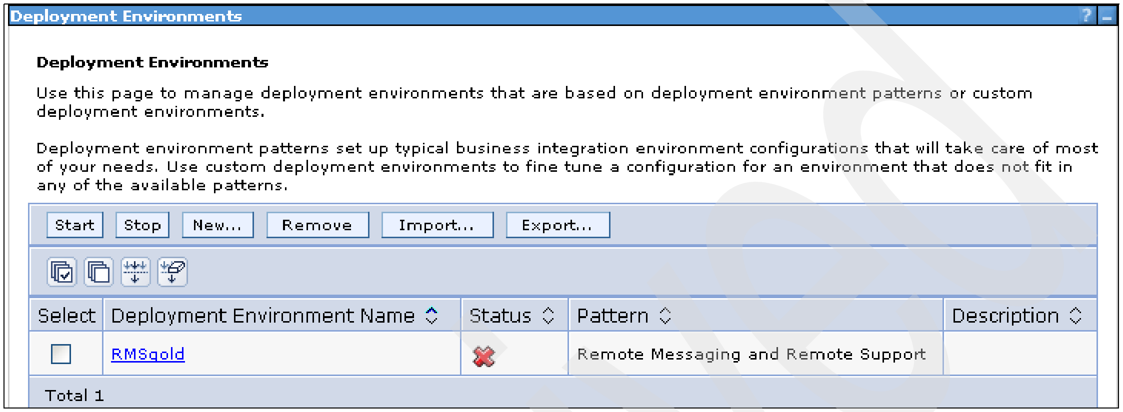


Figure 7-22 Main Deployment Environments window

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 206
- “Starting and stopping deployment environments” on page 208
- “Reviewing and changing deployment environments” on page 210
- “Exporting and importing deployment environments” on page 217

Creating a new deployment environment

In Section 5.2.3, “Creating a deployment topology” on page 121, you built a Remote Messaging and Remote Support environment using the deployment environment’s Create new deployment environment wizard.

To review the wizard:

1. Click the **New** button shown in Figure 7-22 on page 205. The wizard will guide you through naming, deciding, and populating the components in the environment. In the Create new deployment environment window (Figure 7-23), you will provide a deployment environment name and runtime capability (WPS or ESB).

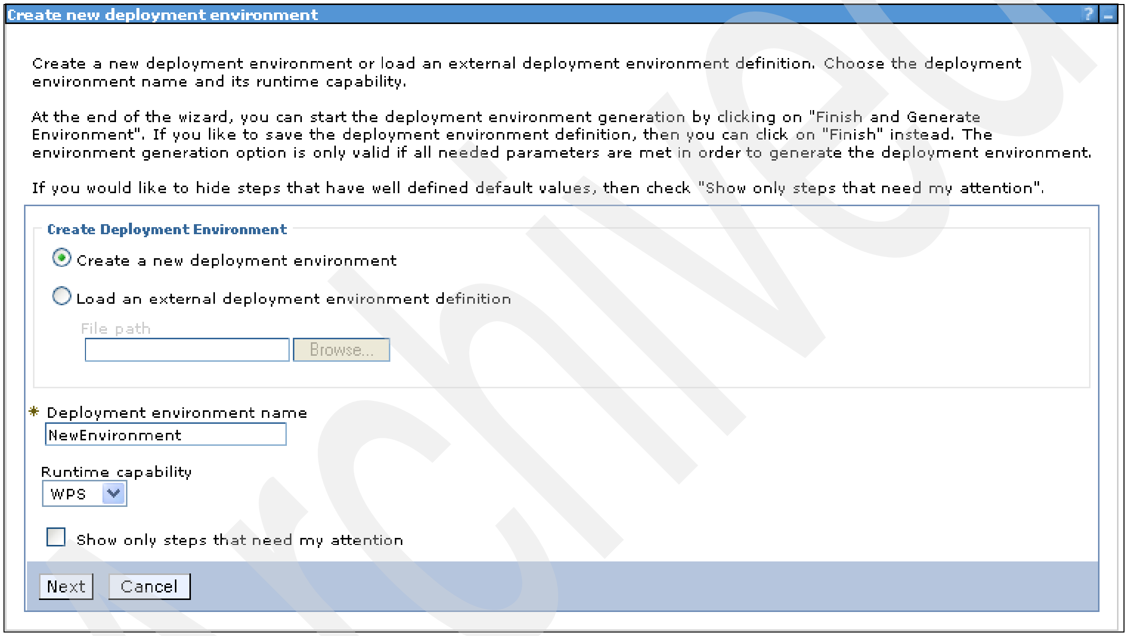


Figure 7-23 Create Deployment Environment

2. Enter New Environment in the Deployment environment name text box

3. Click **Next.** The window in Figure 7-24 will help you decide which pattern to choose.

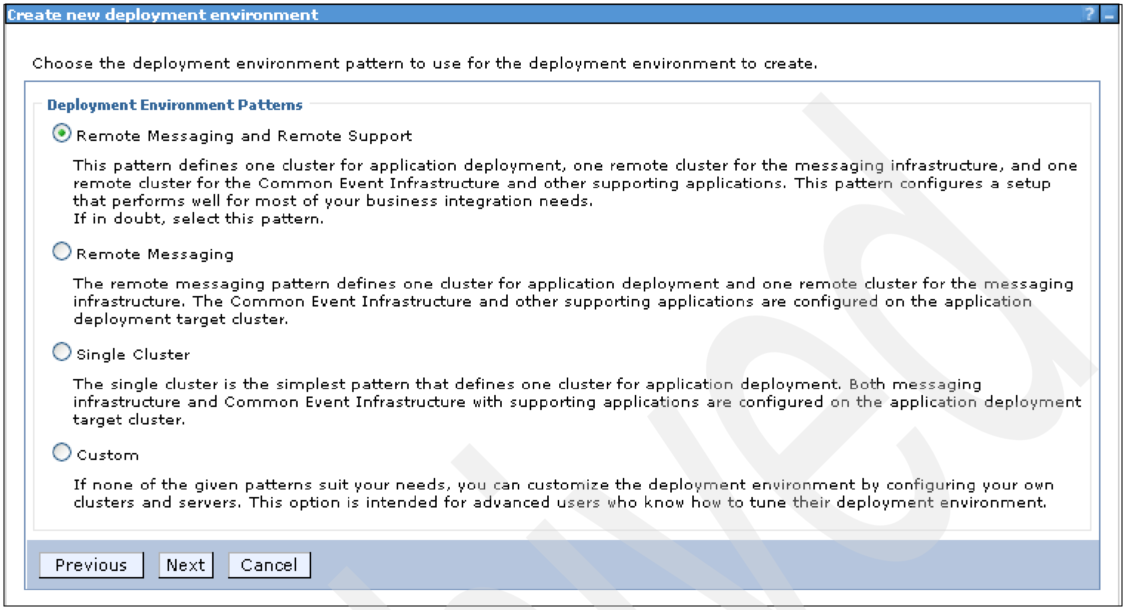


Figure 7-24 Deployment Environment patterns

4. Click the appropriate radio button.

5. Click **Next**.

   The next six steps require filling in the artifacts to describe the environment. Any artifacts that can be discovered by the wizard will be available to you. Figure 7-25 shows the additional steps needed to complete this particular pattern.

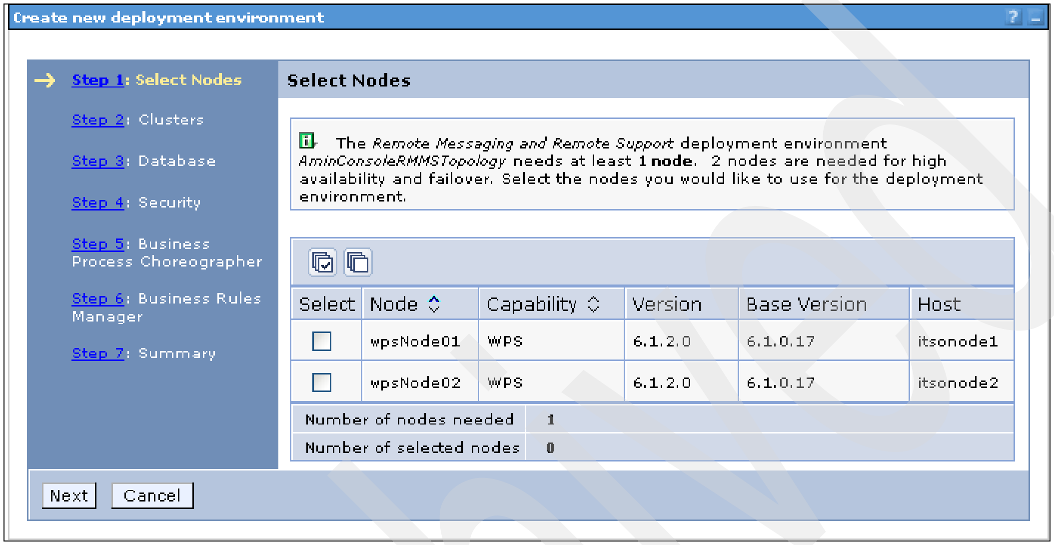


Figure 7-25 Step to complete creating a new pattern

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 pattern is built, it can be centrally managed from the Deployment Environments window of the Integrated Solutions Console.

1. Click **Servers** → **Deployment Environment** to view the configuration information. This window provides the high level status of the environment.
2. From the window shown in Figure 7-26 on page 209, you may start or stop the deployment environment. The current status of the environment is stopped.

   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:** This status indicator will turn green or red immediately, but it may take several minutes for the deployment environment to stop or start.

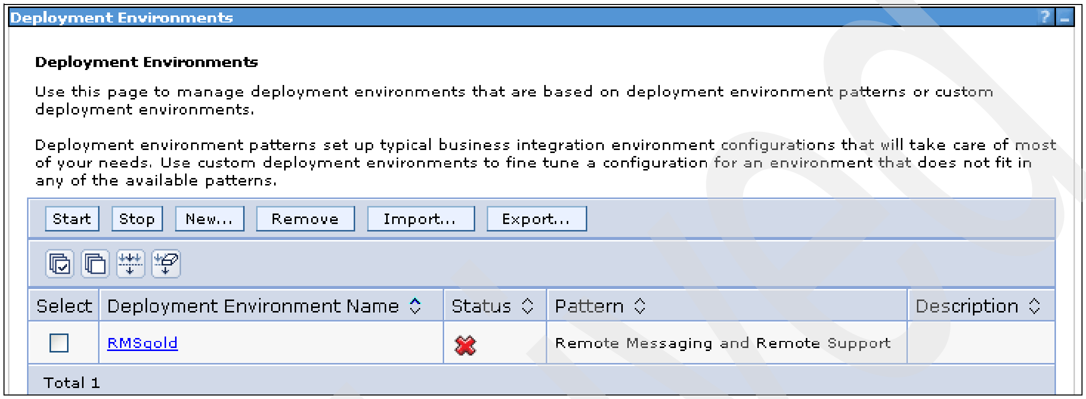


Figure 7-26 Current status is stopped

3. Verify the deployment environment's status in WebSphere Process Server V6.1.2 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:** Deployment environments status discovery is not fully implemented in WebSphere Process Server V6.1.2. This feature is slated for a future release.

Reviewing and changing deployment environments

To see more details of the environment:

1. Click the **RMSgold** link shown in Figure 7-26 on page 209.

   The configuration window shown in Figure 7-27 provides you a way to manage the resources of the deployment topology. These resources are the data sources, authentication aliases, deployment topology, and deferred configuration.

2. Review Figure 7-27. You should notice a warning message to complete deferred configuration.

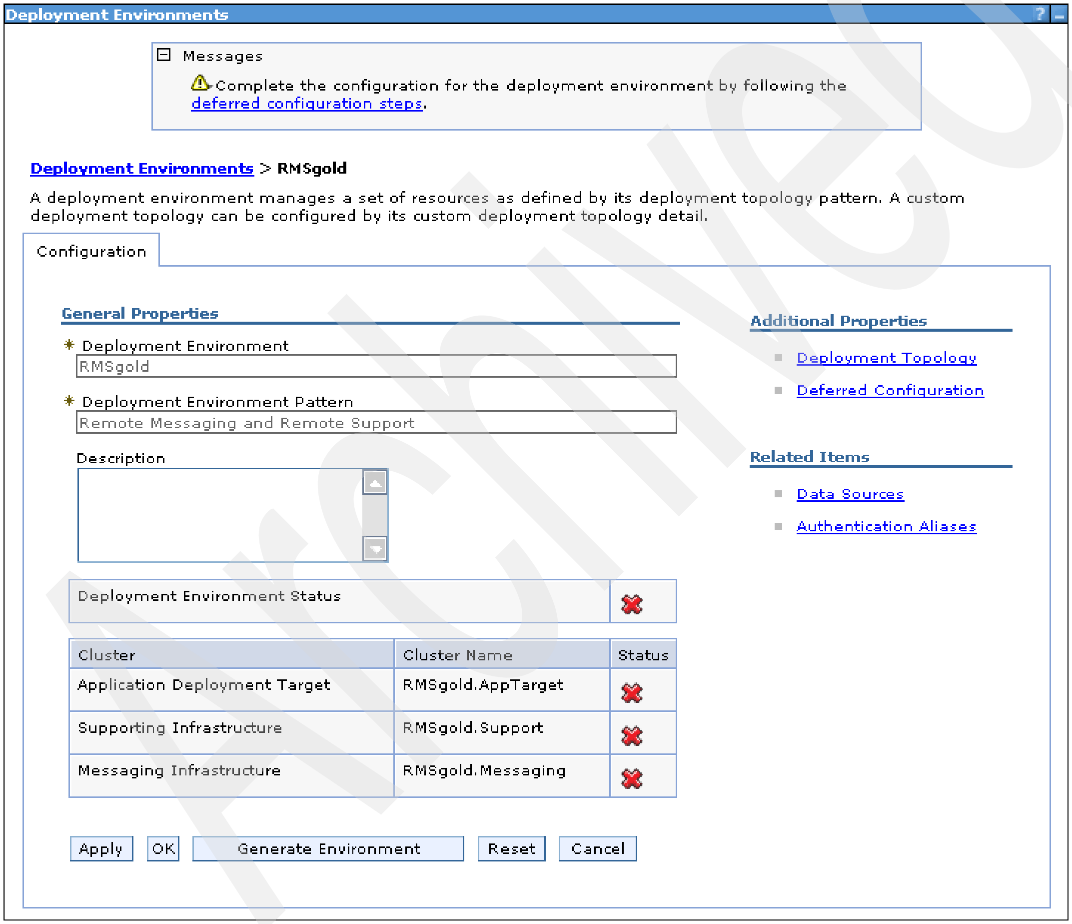


Figure 7-27 RMSGold Configuration window

3. Click the **Deferred Configuration** link and you will see what needs to be configured to properly complete this environment.

   a. A review of the Deferred Compensation window shown in Figure 7-28, indicates that there are six remaining steps to create the database tables. In Chapter 5, “Configuring a Remote Messaging and Remote Support topology” on page 89, you created the databases as a separate step. The deployment environment remembers this and keeps it as a deferred configuration.

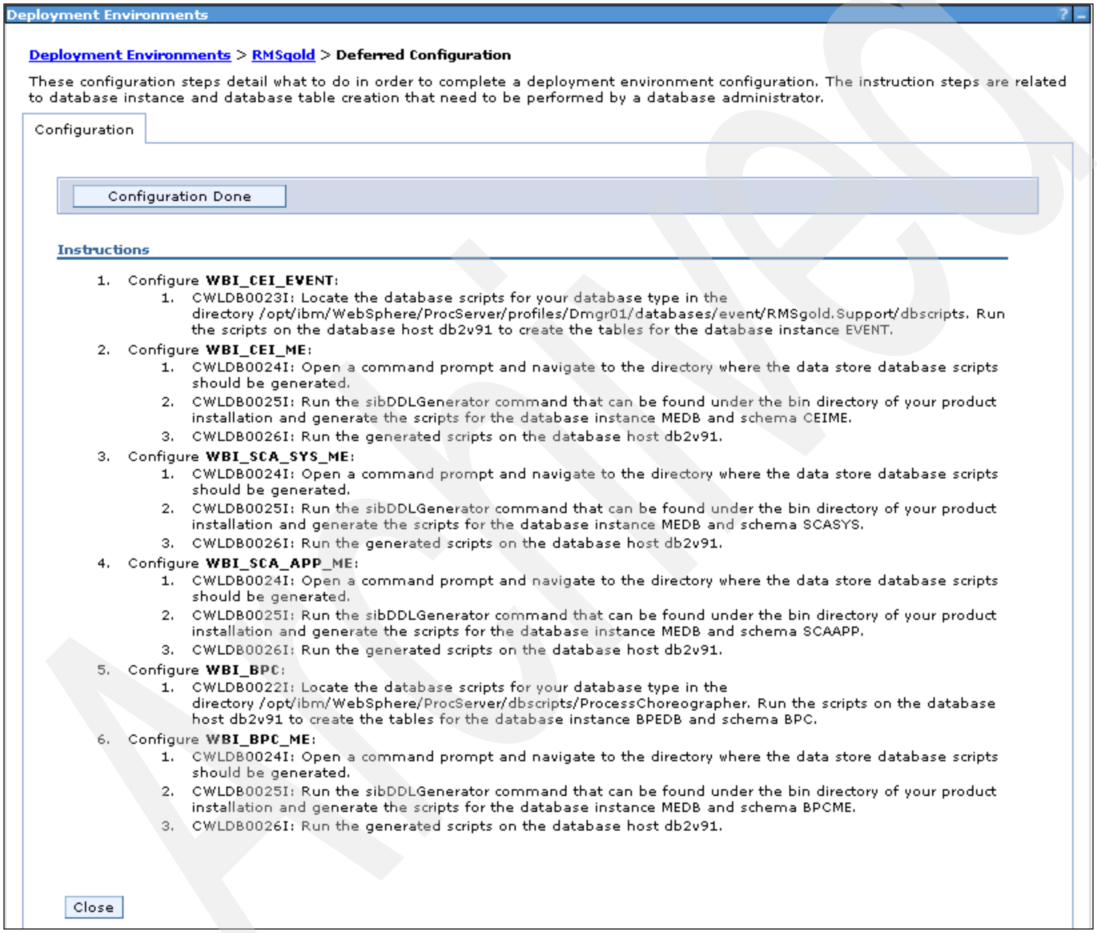


Figure 7-28 Deferred configuration

> **Note:** If these had been generated by the profile manager, these steps would not be considered deferred.

b. Verify that the tables have been created.

c. Click **Configuration Done**. This will remove the warning message and leave an audit message (Figure 7-29).

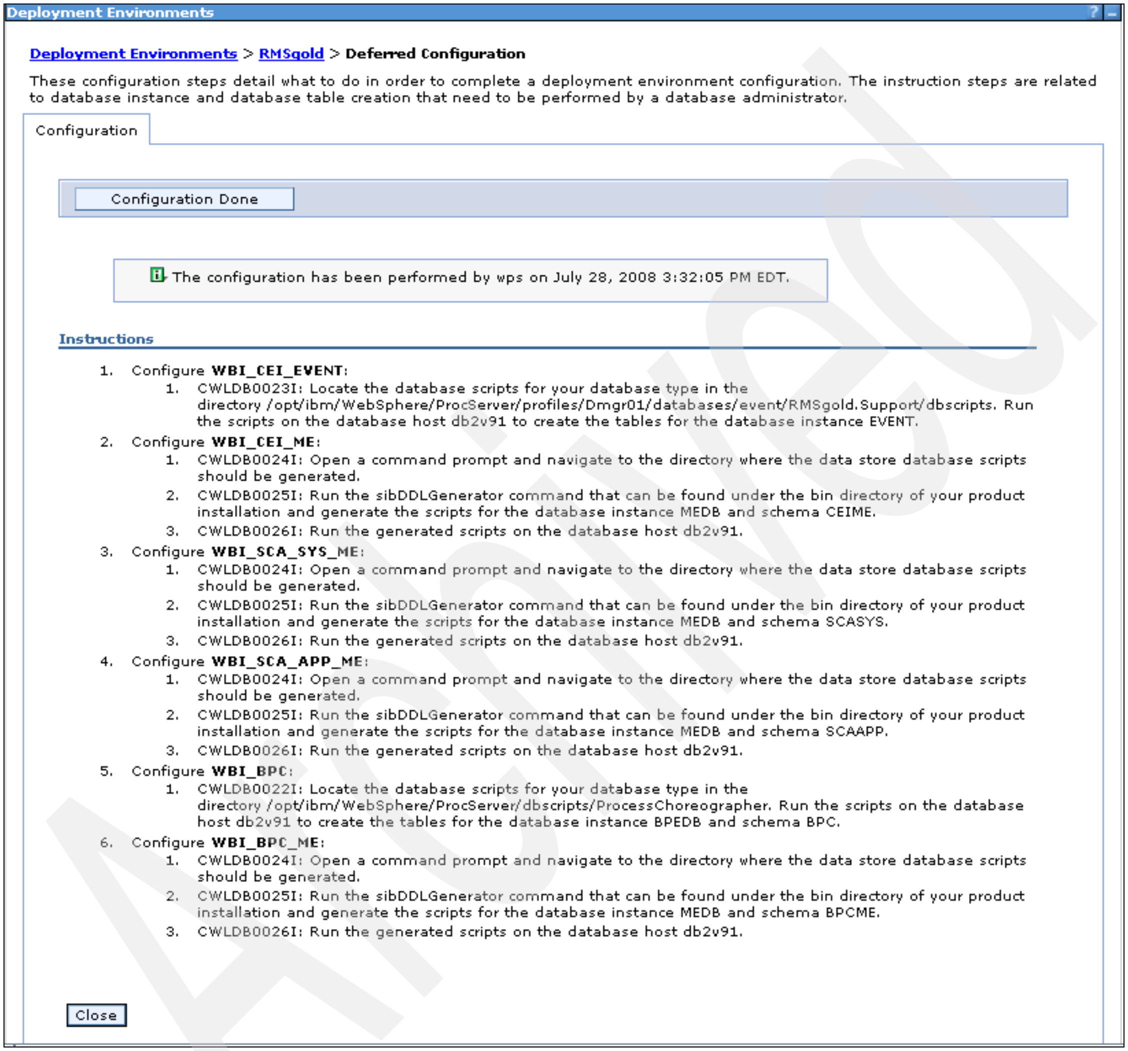


Figure 7-29 Audit message

> **Note:** The instructions in the deferred configuration window do not go away even after clicking **Configuration Done**

4. From the Deployment Environments > RMSGold configuration window (Figure 7-27 on page 210), click **Deployment Topology.** 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 7-30:
      i. Select the check box for the appNode01 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.

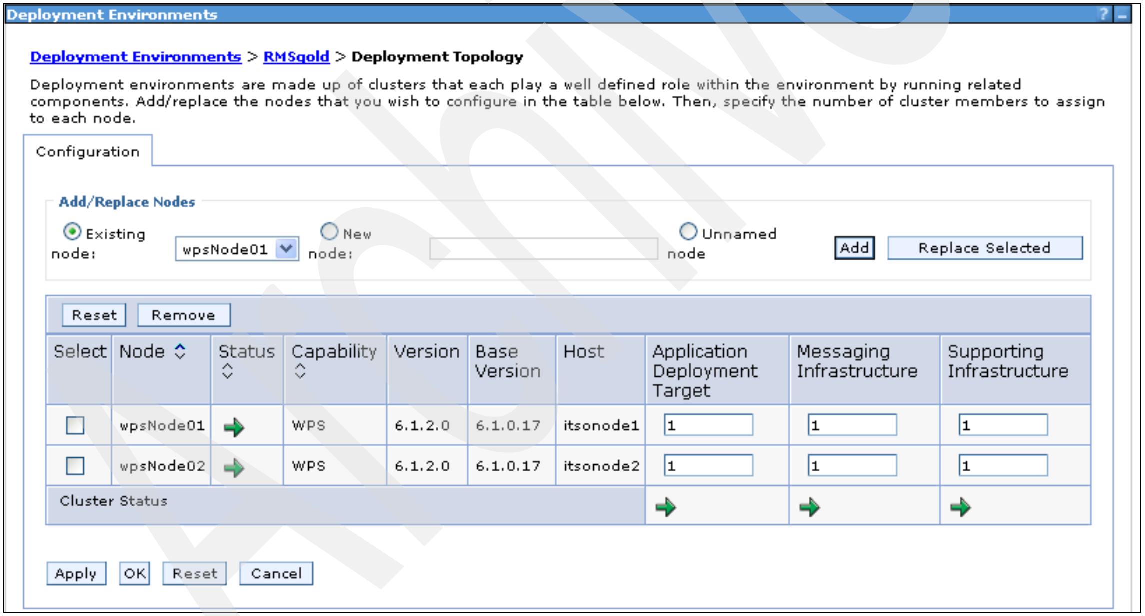


Figure 7-30 Deployment topology window

b. To decrease cluster members, perform the following steps in the window shown in Figure 7-30 on page 213:

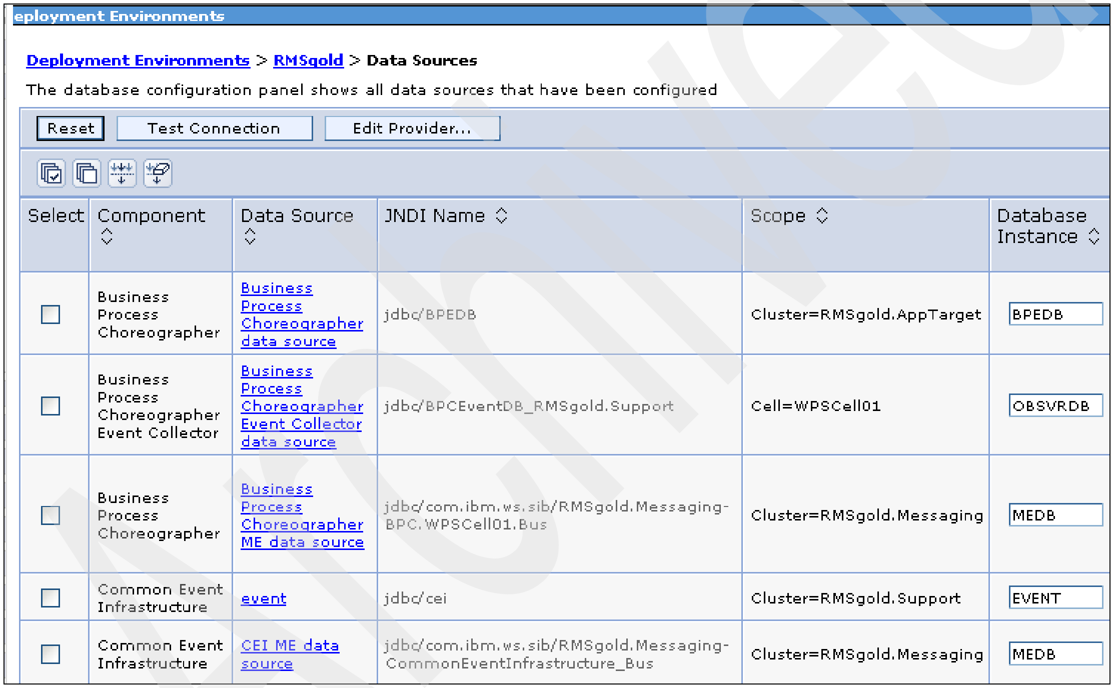
   i. Select a the check box for the AppNode01 row.

   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 see the number of servers has decreased.

5. Click **Data sources** on the right hand side of the window shown in Figure 7-27 on page 210. From the Data sources window, Figure 7-31, you can perform the tasks listed after the figure:



Select	Component	Data Source	JNDI Name	Scope	Database Instance
☐	Business Process Choreographer	Business Process Choreographer data source	jdbc/BPEDB	Cluster=RMSgold.AppTarget	BPEDB
☐	Business Process Choreographer Event Collector	Business Process Choreographer Event Collector data source	jdbc/BPCEventDB_RMSgold.Support	Cell=WPSCell01	OBSVRDB
☐	Business Process Choreographer	Business Process Choreographer ME data source	jdbc/com.ibm.ws.sib/RMSgold.Messaging-BPC.WPSCell01.Bus	Cluster=RMSgold.Messaging	MEDB
☐	Common Event Infrastructure	event	jdbc/cei	Cluster=RMSgold.Support	EVENT
☐	Common Event Infrastructure	CEI ME data source	jdbc/com.ibm.ws.sib/RMSgold.Messaging-CommonEventInfrastructure_Bus	Cluster=RMSgold.Messaging	MEDB

Figure 7-31 JDBC 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 configuration.

> **Note:** At the writing of this Redbooks publication, the Test connection button does not work. This issue will be resolved with APAR JR30145.

– 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 7-32 on page 216. Select the appropriate node and driver paths.

  iv. Click **OK** and click **Save** after changes have been made.

  v. Generate and restart the environment for these changes to be propagated.

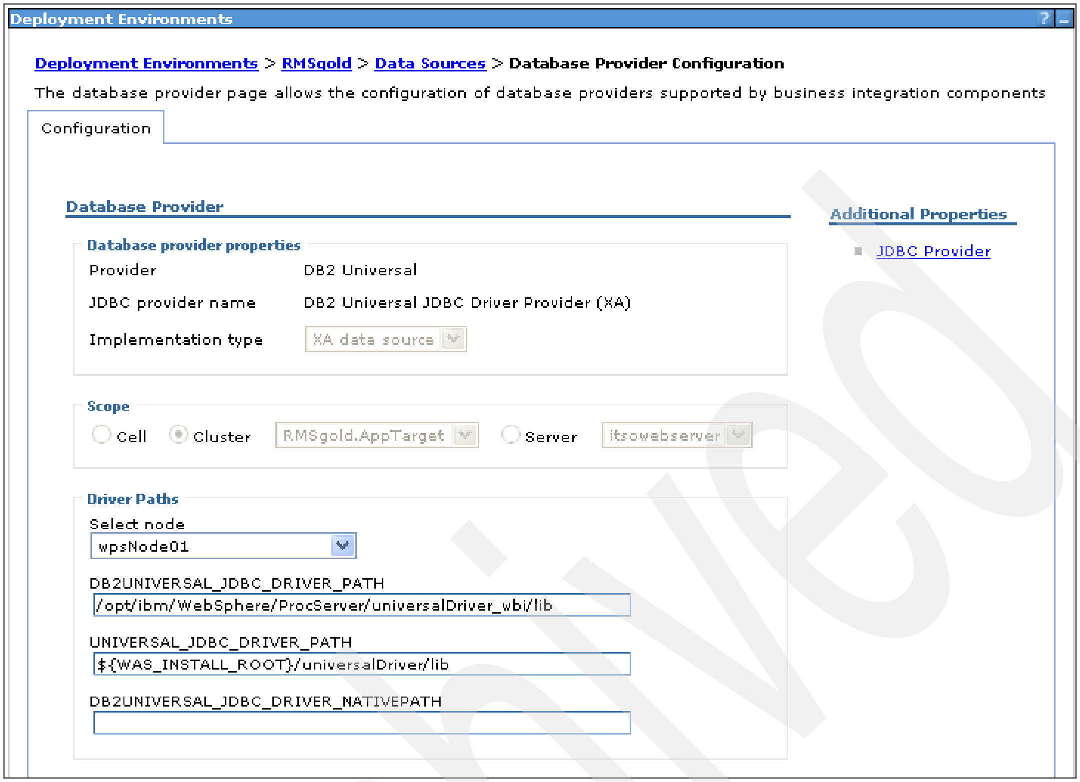


Figure 7-32 Database Provider Configuration window

6. From the Authentication Aliases window (Figure 7-33), you can change the user name and password for all exposed authentication aliases. These changes will be propagated out once you generate the environment.

   To change user name and password, perform the followings steps:

   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.

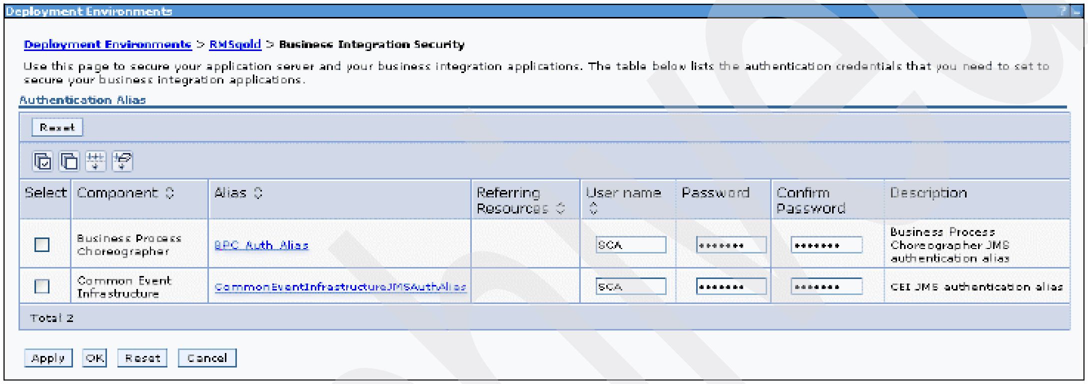


Figure 7-33 Authentication Aliases window

Exporting and importing deployment environments

Exporting your configuration is a helpful tool when you want to promote an environment from system or stress testing to user acceptance testing, then to a production environment. 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, Application Target can have 8 cluster members and Messaging and Support only 2 cluster members.

Once the RMSGold environment is ready to be promoted to RMSUAT, follow these steps to build the RMSUAT environment.

1. Export the deployment environment. This is an XML file.
2. Make a copy of the generated XML file and name it RMSUAT.xml.
3. Review 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 7-1. This is the audit message when you clicked **Configuration Done** in the Deferred Configuration window.

Example 7-1 Second line of generated Deployment Environment export XML file

```
<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="6.1.2.0"
deferredConfigTime="2008-07-28T15:32:05.804-0400"
deferredConfigUser="wps">
```

   c. Convert the RMSGold environment naming to RMSUAT. To do so, perform the following steps.
      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 will need to be edited in the authentication alias and service integration bus names. Example 7-2 on page 219 shows the Support Topology's CEI database component.

Example 7-2 Support Topology entry in RMSUAT.xml file

```
<components id="WBI_CEI" name="WBI_CEI" version="6.1.2.0"
topologyRole="Support" baseRuntimeId="WAS" level="1">
    <dataSrc component="WBI_CEI" createTable="false"
dbcomponent="WBI_CEI_EVENT">
      <authAlias name="WPSCell01/RMSUAT.Support/EventAuthDataAliasDB2"
userName="UATInst" password="{xor}Lz4sLChvLTs=" 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. Look through the rest of the file for other values that need to 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** → **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 7-34.

> **Note:** Because the RMSGold name changed to RMSUAT on line 2 of the file, the deployment environment name text box population is not required. If you use the same name the wizard requires a new name.

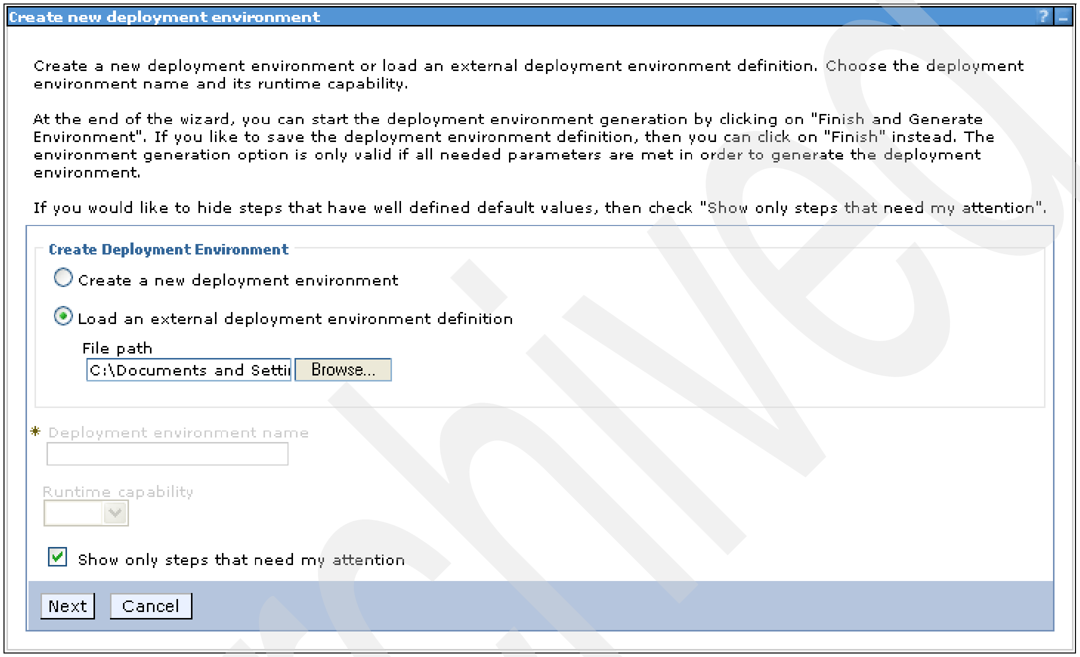


Figure 7-34 Import exported xml file

10. Complete the steps in the Import Wizard as you did when creating a new deployment environment.

Another way is to follow the scripted installation in Section 5.3, "Installation through scripts silently" on page 137, and use the RMSUAT.xml file as the file in Section 5.3.4, "Importing and generating a topology definition" on page 141. This will allow you to quickly replicate your new environment.

7.2.2 Business Process Choreographer

This section describes some administration considerations for the Business Process Choreographer:

Using compensation

If you have processes that are using compensation then you will need to enable this function. This function is enabled by default. To enable the Compensation Service, perform the following steps

1. Click **Servers** → **Application Server** → **RMSGold.AppTarget.wpsNode01.0**.
2. Click **Container Services** → **Compensation Service**.
3. Select **Enable service at server startup**.
4. Adjust options based on your systems needs.
   - Compensation handler retry limit defaults to unlimited retries.
   - Compensation handler retry interval defaults to 30 seconds.

> **Note:** This service is enabled at the server level, not the cluster level.

Process navigation performance tuning

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/v6r1mx/index.jsp?topic=/com.ibm.websphere.bpc.612.doc/doc/bpc/t5tuneint_processnavigation.html

7.2.3 Common Event Infrastructure

In WebSphere Process Server V6.1.2, 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, perform the following steps.

1. Click **Service Integration** → **Common Event Infrastructure** → **Event service** → **Event services** → **Default Common Event Infrastructure event server**.
2. Clear the **Enable event data store** check box as shown in Figure 7-35.

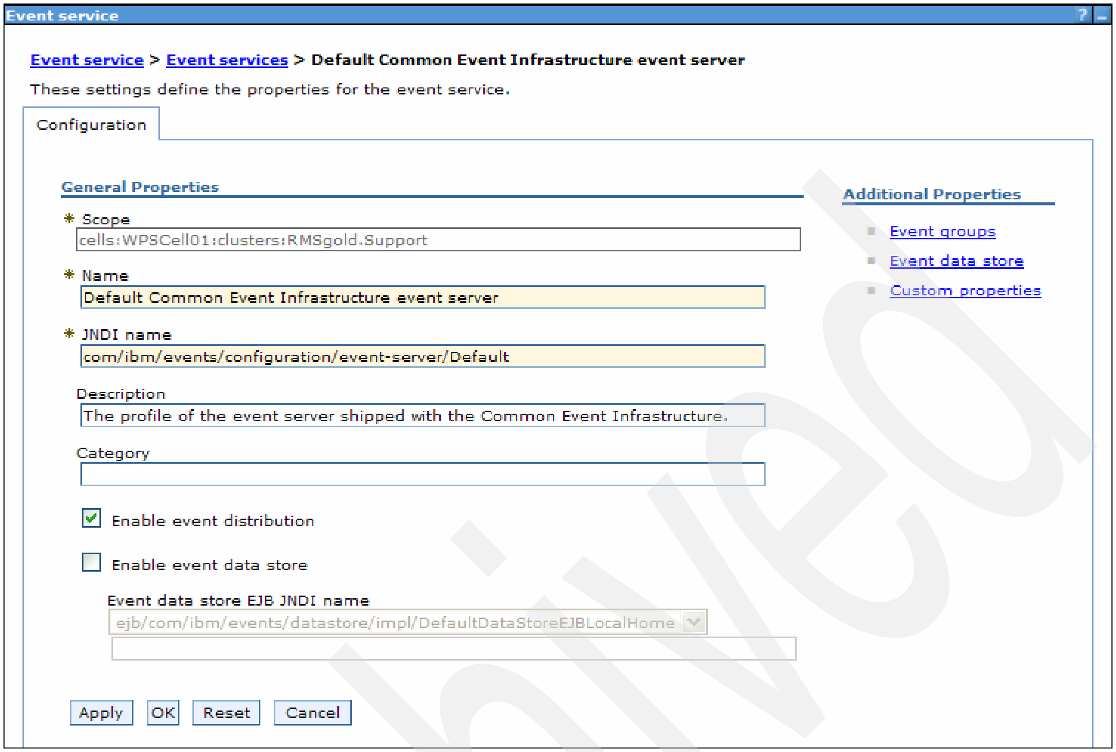


Figure 7-35 Disable event data store

3. Click **OK** and click **Save**.
4. Restart **RMSGold.Support**.

For additional performance best practices, refer to IBM Redpaper *IBM WebSphere Business Process Management V6.1 Performance Tuning*, REDP-4431.

7.2.4 Changing the database password

One common administrative problem is changing a database password to comply with corporate security guidelines. The goal of this section is to give you a method to accomplish this task without a 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 access the same tables with the same privileges. The trick is to stagger the database user ID's 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 thirty 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 clicking the **Ripplestart** button from the **Servers** → **Clusters** window in the Integrated Solutions Console shown in Figure 7-36

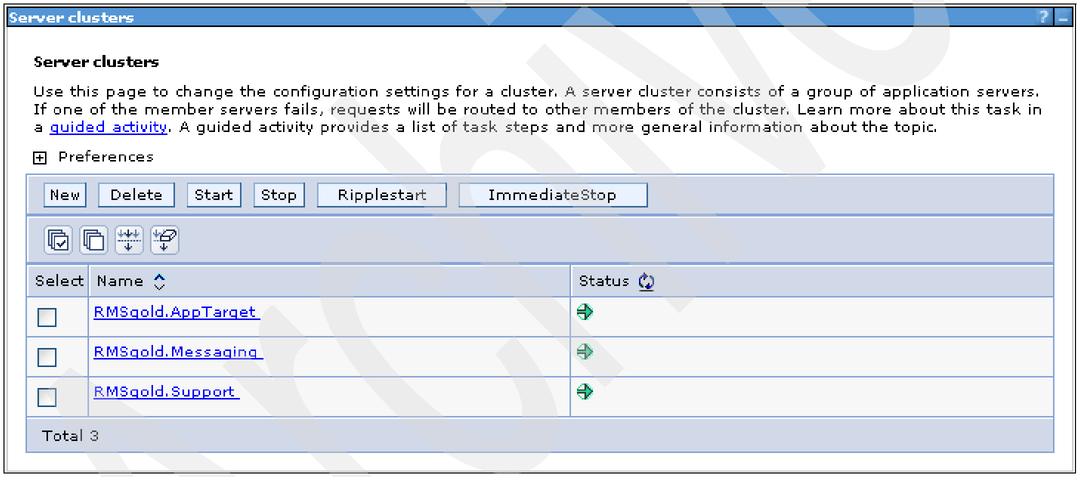


Figure 7-36 Ripplestart cluster to pick up new password

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 only makes one resource manager, queue, or database per transaction, then XA compliance is not be a concern.

7.2.5 Failed events

This section provides guidance of failed events in WebSphere Process Server.

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 developed 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 Service Component Architecture (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 see a greater occurrence of failed events.

How to use the Failed Event Manager

WebSphere Process Server has built into the Integrated Solutions Console a tool called Failed Event Manager. This is a Web based tool that will enable the administrator to submit events to a component that failed to complete.

To launch the Failed Event Manager:

1. Open the Integrated Solutions Console.
2. Click **Integration Applications** → **Failed Event Manager**.

The Failed Event Manager (Figure 7-37) 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.

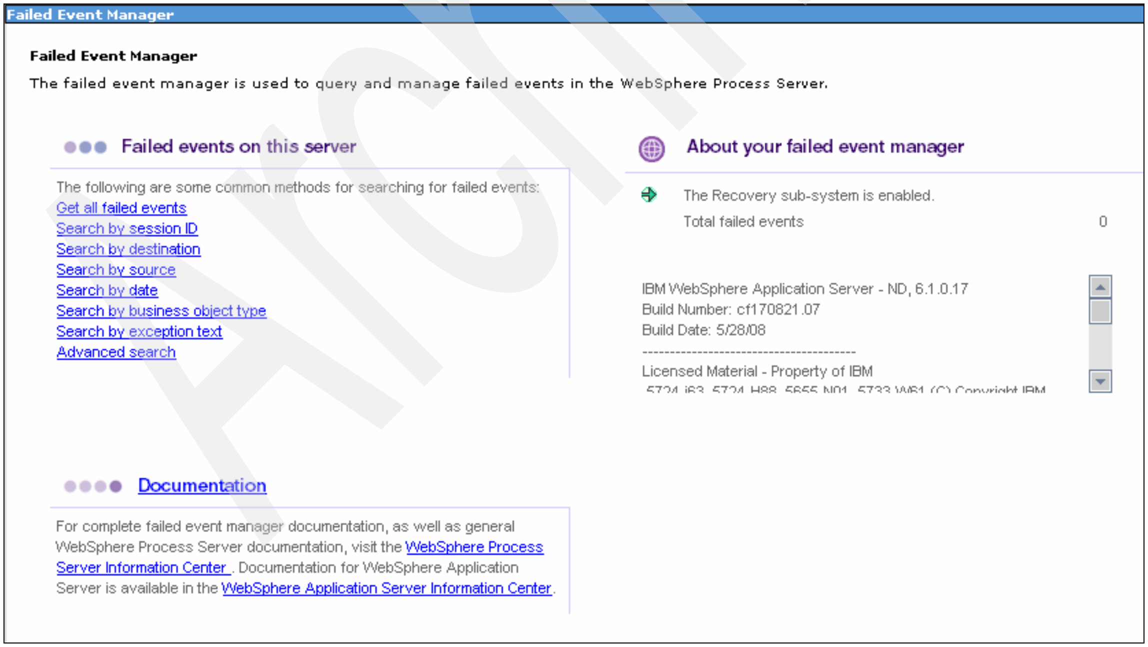


Figure 7-37 Failed Event manager main window

Important: If the About your failed event manager section (shown in Figure 7-37) says that the Recovery sub-system is disabled, verify that the SCA container is started. In this configuration, it is the RMSGold.AppTargetCluster. 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 7-38.

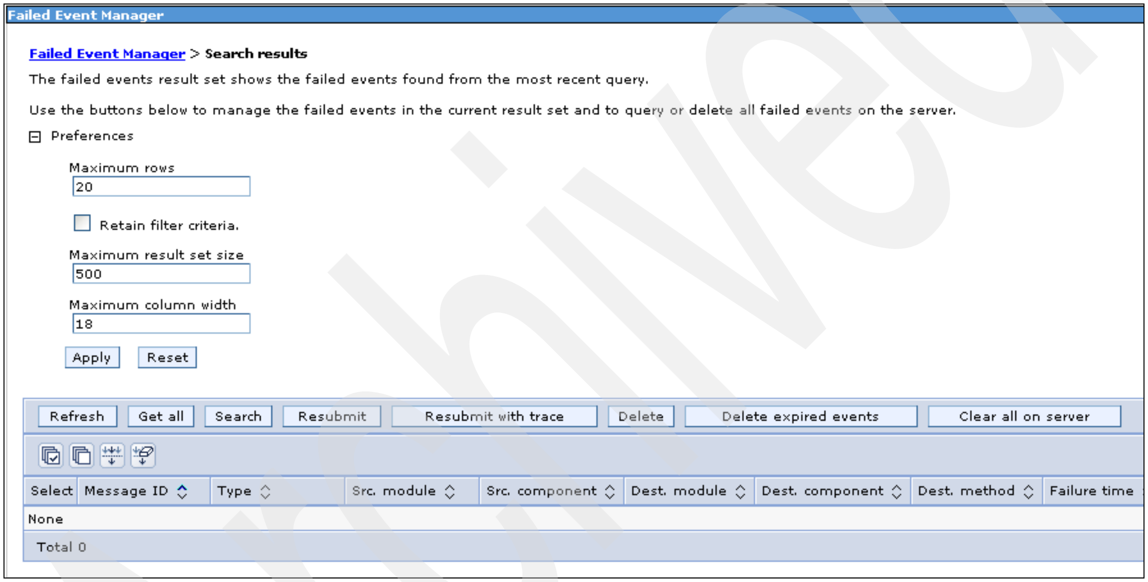


Figure 7-38 Failed Event Manager actions

The Failed Event Manager shows you information about the failed event, so that you can take some action on it. 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 you wish to resubmit.
2. Click **Resubmit**.
3. Click **Refresh**.

This should clear the event. If it still shows up 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 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, then you want to 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.

8

Advanced production topologies

This chapter discusses ways to extend the Remote Messaging and Remote Support topology to provide additional processing capability.

In this chapter, the following topics are discussed:

- Adding additional cluster members to the clusters created during Remote Messaging and Remote Support deployment environment generation
- Adding additional 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.

8.1 Overview of extending the 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. For example, you may find that you need additional processing capability in the application target cluster, or you may find a performance bottleneck in the messaging infrastructure.

Extending the Remote Messaging and Remote Support topology can be done in any of the following ways:

- Extending the application target cluster by adding additional nodes and cluster members
- Extending application processing capabilities by adding an additional application target cluster
- Extending the messaging infrastructure by adding an additional messaging cluster
- 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 additional 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 Windows®-based Remote Messaging and Remote Support topology was created. This topology is represented in Figure 8-1 on page 231.

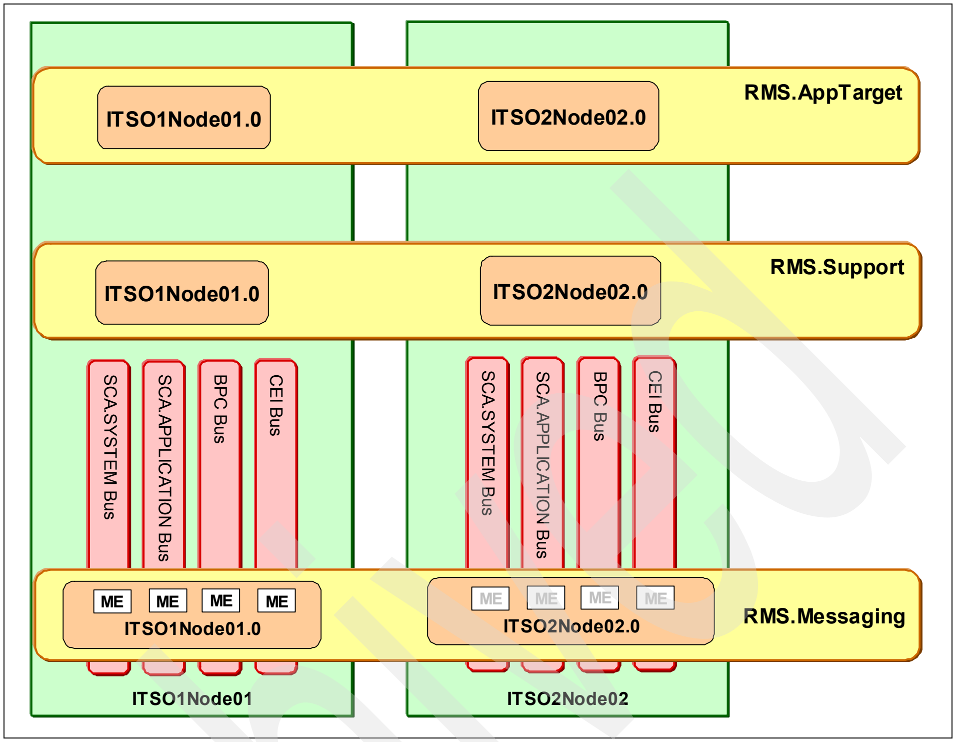


Figure 8-1 Topology created for this chapter

This topology was created using the deployment environments window in the administrative console. This environment contains two machines: ITSO1 and 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 will be expanded by adding additional nodes, servers, and clusters.

8.2 Adding additional nodes and cluster members

If you find a need for adding additional processing capability to the application target cluster in the Remote Messaging and Remote Support topology, you have the option of adding additional 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 you may wish to add additional cluster members to the application target cluster. These include the following reasons:

- To increase application processing capability during peak usage times (for example, increased sales traffic during the fourth quarter or increased accounting traffic at the end of the quarter)
- 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 Machine (JVM™) 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 8-2 on page 233.

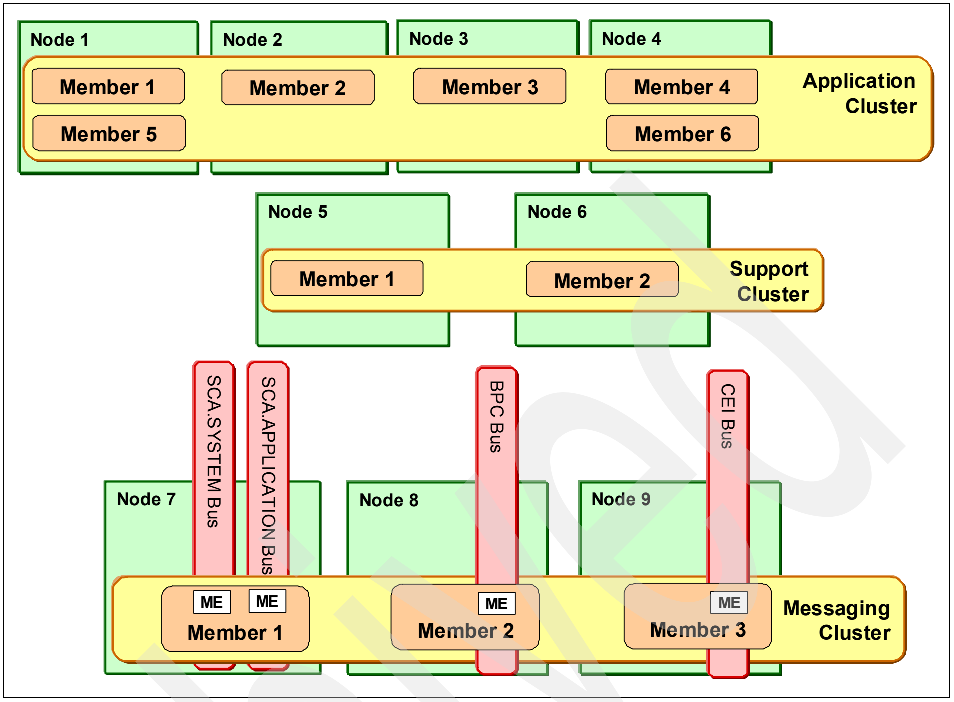


Figure 8-2 Adding additional servers and nodes to the application cluster

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 Section 8.4, “Distributing messaging workload using policies” on page 275). In addition, two of the nodes in the application target cluster were extended by adding additional cluster members to them.

Note the following aspects of this topology:

- Adding additional 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 53.
- 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 additional support cluster members.
- Because adding additional 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 the Redbooks lab, the Remote Messaging and Remote Support topology created in the administrative 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 will house 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 8-3 on page 235.

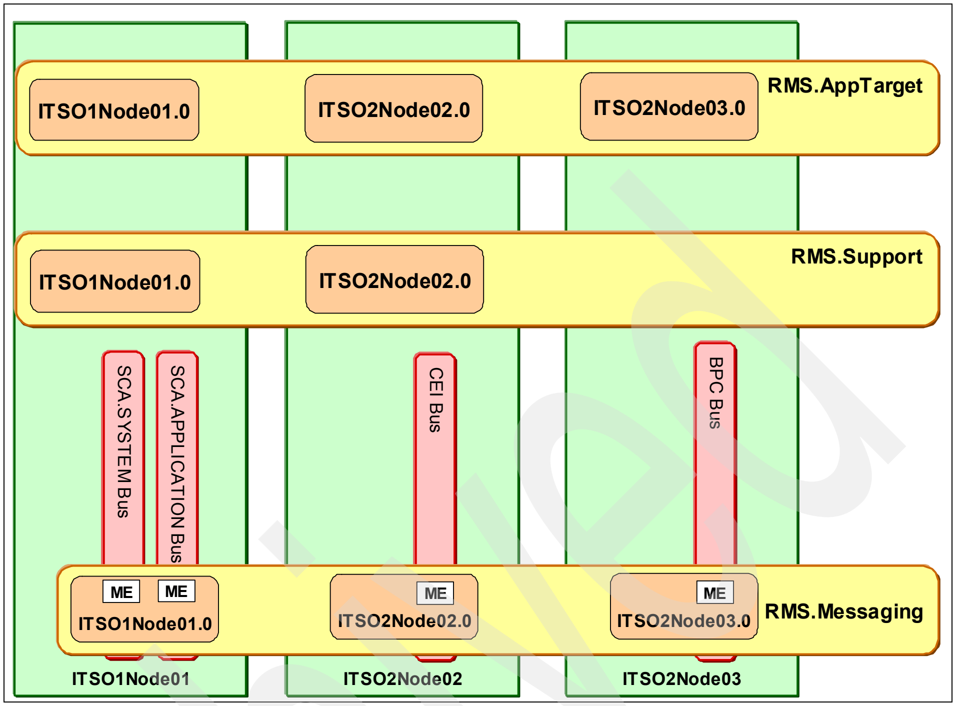


Figure 8-3 Remote Messaging and Remote Support topology with additional nodes and cluster members

Distributing the messaging engines across the messaging cluster members is discussed in Section 8.4, “Distributing messaging workload using policies” on page 275 later in this chapter.

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 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 the Redbooks lab, a new custom profile was created on machine ITSO2.
2. Federate the node.
3. In the WebSphere Process Server administrative console, expand **Servers** and click the **Deployment Environments** link (Figure 8-4).

Figure 8-4 Deployment environments link

4. Click the link for your deployment environment in the Deployment Environment Name column. In the Redbooks lab, the name of the deployment environment was RMS (Figure 8-5).

Select	Deployment Environment Name	Status	Pattern	Description
☐	RMS	✖	Remote Messaging and Remote Support	

Figure 8-5 Deployment Environment Name link

5. In the Additional Properties section, click the **Deployment Topology** link (Figure 8-6).

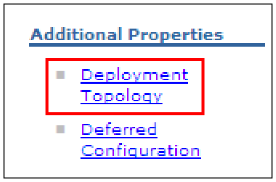


Figure 8-6 Deployment Topology link

6. Click the Existing node radio button and select the newly federated node from the drop-down list. In the Redbooks lab, the newly federated node was named **ITSO2Node03** (Figure 8-7).

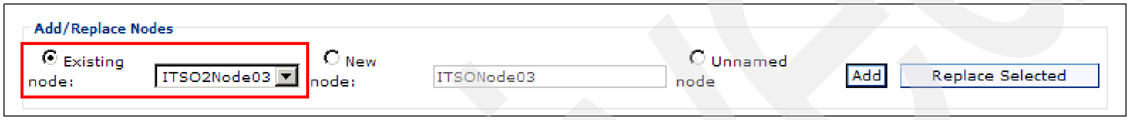
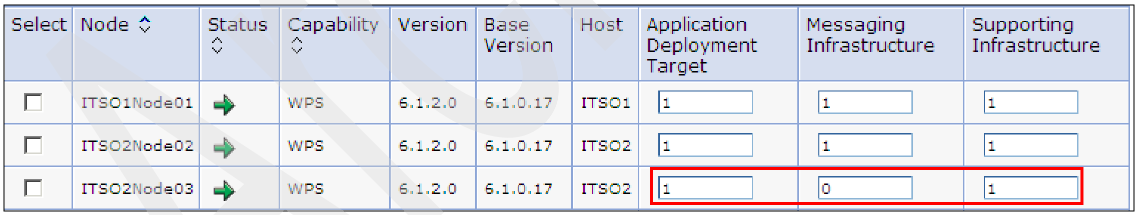


Figure 8-7 Add a new node to the deployment environment

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 will create a single server instance in the RMS.Messaging cluster and a single server instance in the RMS.AppTarget cluster on the new node (Figure 8-8).



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 8-8 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 administrative console navigation pane, expand **Servers** and click the **Clusters** link.

13. Click the link for the application target cluster. In the Redbooks lab, this value was **RMS.AppTarget** (Figure 8-9).

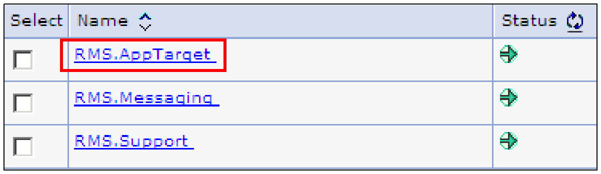


Figure 8-9 Application target cluster link

14. In the Additional properties section, click the **Cluster members** link (Figure 8-10).

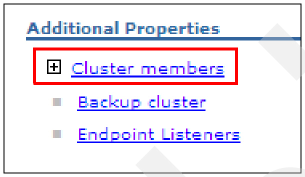


Figure 8-10 Cluster members link

15. In the cluster members table, click the check box for the newly added cluster member and click **Start** (Figure 8-11).

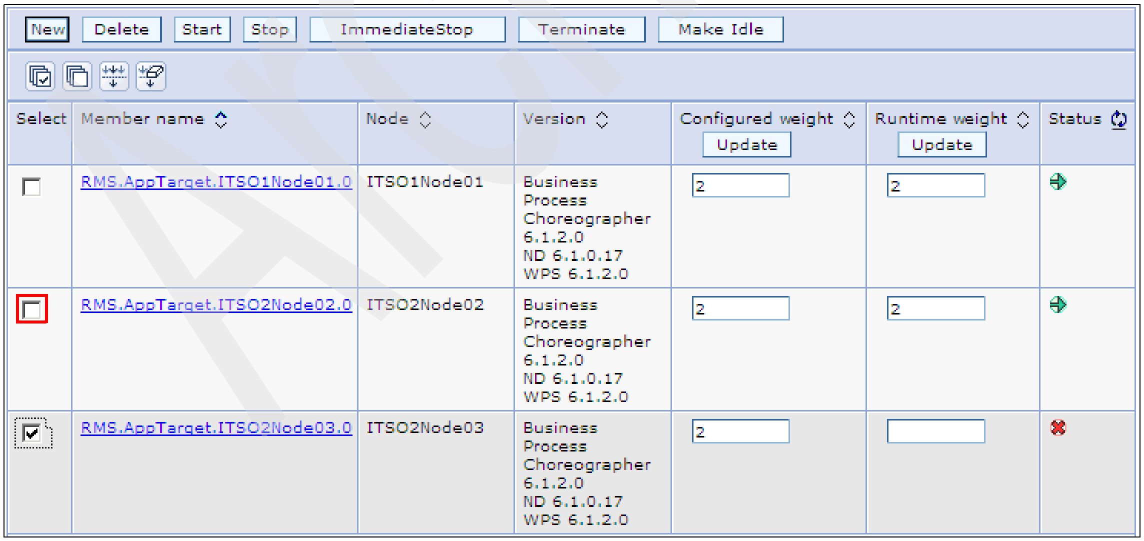


Figure 8-11 Cluster members table

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 administrative 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 8-12.

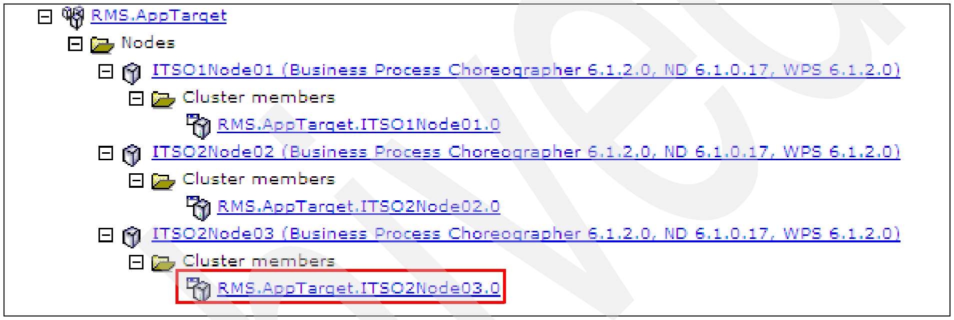


Figure 8-12 Cell topology with additional application cluster member

    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 8-13.

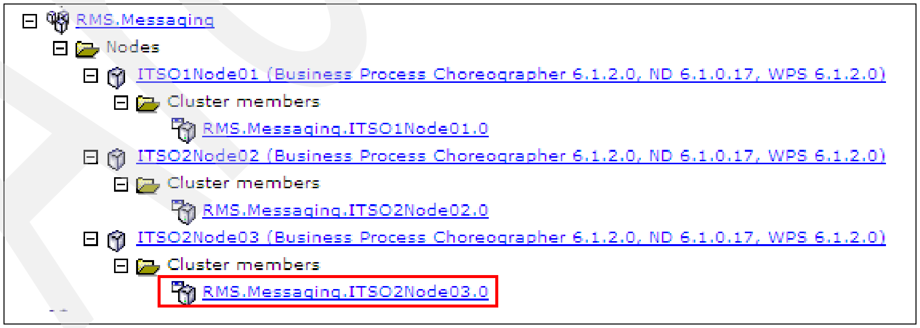


Figure 8-13 Cell topology with additional messaging cluster member

8.3 Adding additional WebSphere Process Server application clusters

There are many reasons you might add an additional WebSphere Process Server application cluster to your cell topology. These include the following reasons:

- 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 will add 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. These include the following limitations:

- 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 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 it will affect your existing performance tuning scenario.
- If you deploy the same application to both clusters, in additional 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 page 275.
- Create an additional messaging cluster. See page 259.

8.3.1 Adding an additional application cluster

Adding an additional application cluster to the Remote Messaging and Remote Support topology is depicted in Figure 8-14.

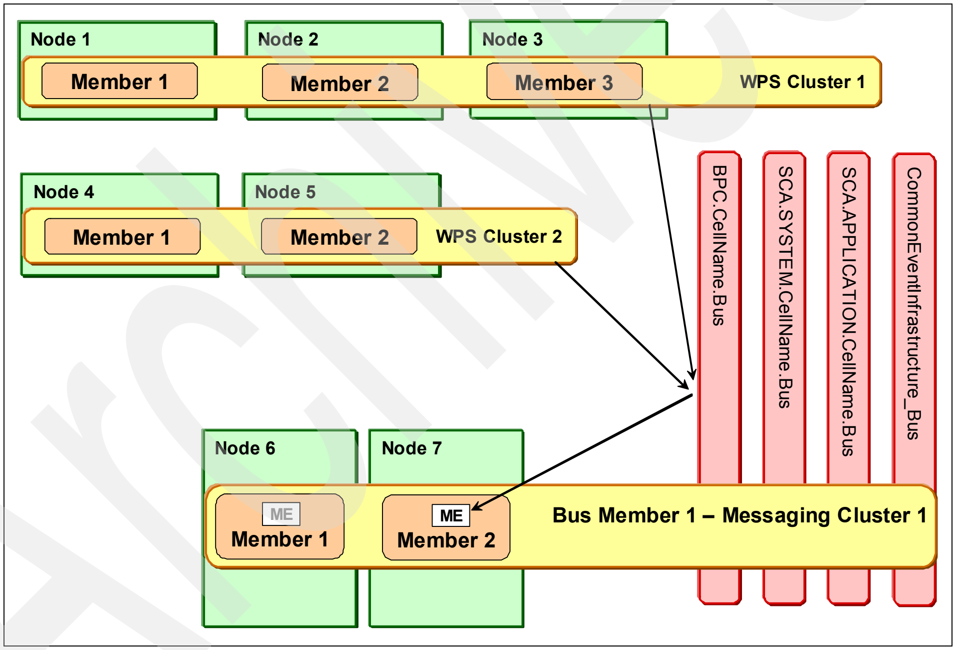


Figure 8-14 Adding an additional application target cluster

This topology depicts 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 will configure 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 will configure 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 will leverage 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 Redbook is represented in Figure 8-15.

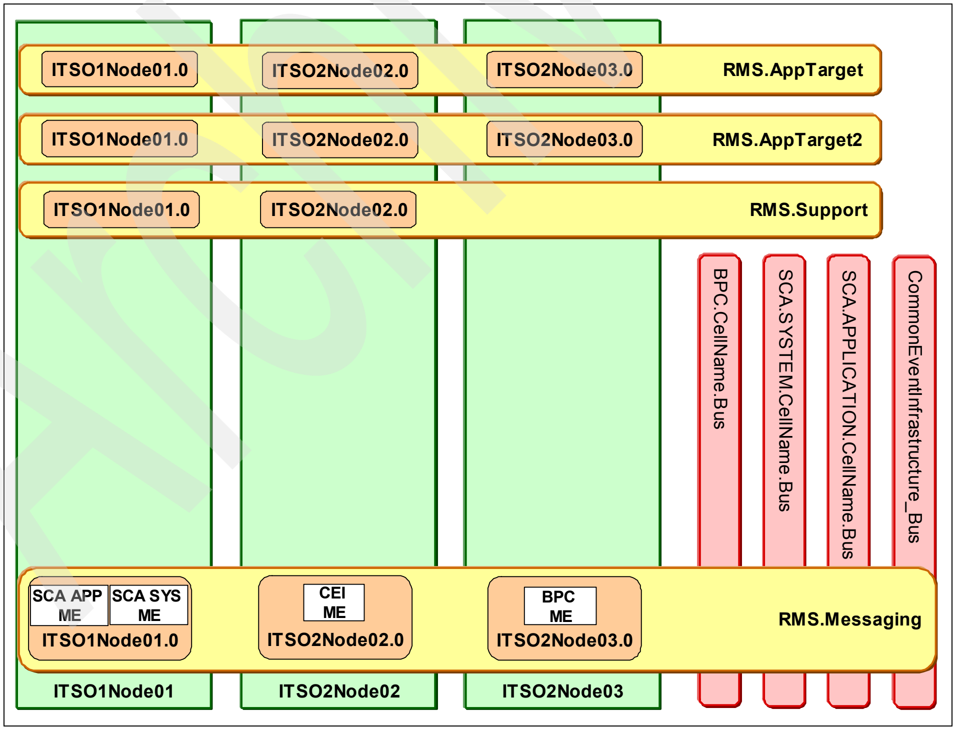


Figure 8-15 Gold topology with second application cluster

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. Creating the second application cluster.
3. Configuring SCA support.
4. Configuring the Business Process Choreographer.
5. 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 you choose ultimately depends upon the database system 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, perform the following steps:

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:
      >
      > ```
      > <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 the Redbooks lab, the name BPEBE02 was used.
   c. Save and close the file.
   d. Edit the <Root>\dbscripts\ProcessChoreographer\DB2\createTablespace.sql file.

e. Replace all instances of @location@ with the name of the DB2 node directory. In the Redbooks lab, the directory C:\DB2\NODE0000 was used (Figure 8-16).

> **Note:** Adjust the directory paths for your operating system. Because the Redbooks 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 8-16 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 the Redbooks 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.
```

k. Issue the following command to close the connection:

```
db2 CONNECT RESET
```

l. Close the DB2 Command window.

Creating the second application cluster

Once the new database is created, the next step is to create the second application target cluster.

To create a second application cluster perform the following steps:

1. In the administrative console navigation pane, expand **Servers** and click the **Clusters** link (Figure 8-17).

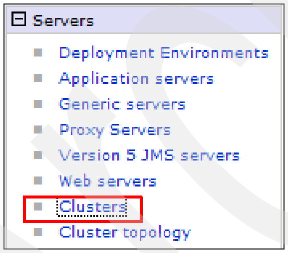


Figure 8-17 Clusters link

2. In the server clusters window, click **New**. This will open the Step 1: Enter basic cluster information window.

3. Enter RMS.AppTarget2 in the Cluster name text box (Figure 8-18).

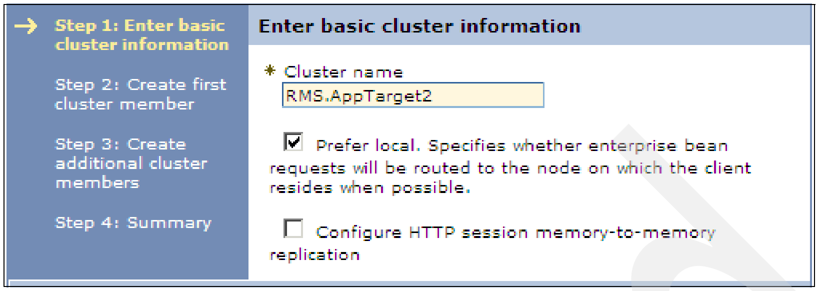


Figure 8-18 New application cluster name

4. Click **Next**. The Step 2: Create first cluster member window opens.
5. Perform the following steps to create the first cluster member (Figure 8-19):
   a. Enter the name of the first cluster member in the Member name text box. In the Redbooks lab, the name RMS.AppTarget2.ITSO1Node01.0 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 the Redbooks 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.

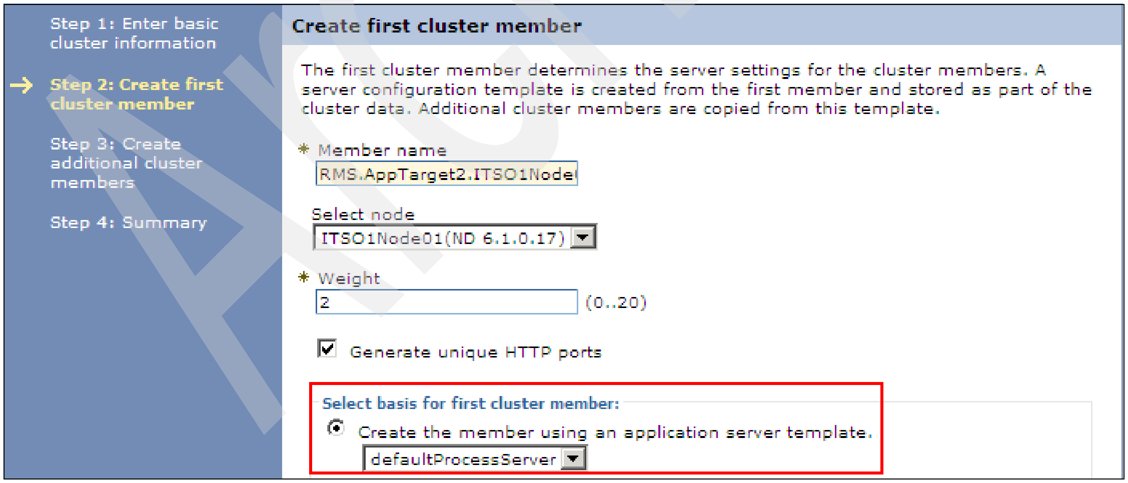


Figure 8-19 Create first cluster member

> **Important:** Because this cluster uses WebSphere Process Server functionality, you must choose the defaultProcessServer template. The default template creates a WebSphere Application Server instance.

6. Click **Next**. The Step 3: Create additional cluster members window opens.
7. Perform the following steps to create additional cluster members (Figure 8-20):
   a. Enter the name of the additional cluster member in the Member name text box. In the Redbooks lab, the name RMS.AppTarget2.ITSO2Node02.0 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 the Redbooks lab, this value was ITSO2Node02.

Figure 8-20 Add additional cluster members

8. Click the **Add Member** button. The name of the additional cluster member should appear in the table.

9. Repeat the previous steps to add any additional cluster members. In the Redbooks lab, a total of three servers were created on three separate nodes (Figure 8-21).

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 8-21 Cluster members table

> **Adding additional members:** Adding additional members during cluster creation is not required. You may create the cluster with one member first and verify the cluster configuration before adding additional members. For demonstration purposes, all cluster members were added at the same time in this example.

10. Click **Next**. The Step 4: Summary window opens.
11. Review your options and click **Finish**.
12. Click the **Save** link at the top of the window (Figure 8-22).

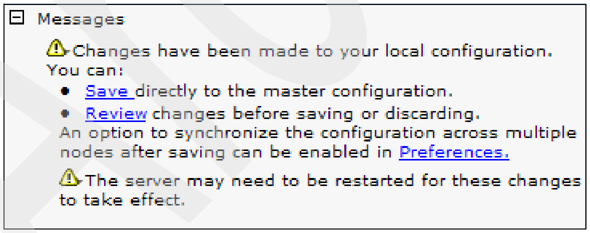


Figure 8-22 Save changes to the master configuration

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 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 administrative console. In addition to using the administrative 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.

- **`configSCAJMSForCluster`**

  Use this command to configure the messaging cluster to support asynchronous communication for SCA applications using the SCA.APPLICATION bus.

To configure CSA support for the second application cluster:

1. In the administrative console navigation pane, expand **Servers** and click the **Clusters** link.
2. In the server clusters window, click the **RMS.AppTarget2** link (Figure 8-23).

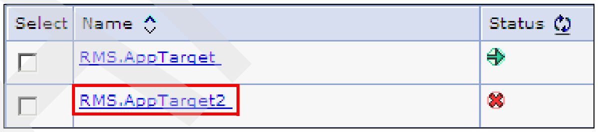


Figure 8-23 RMS.AppTarget2 link

3. In the Business Integration section, click the **Service Component Architecture** link (Figure 8-24).

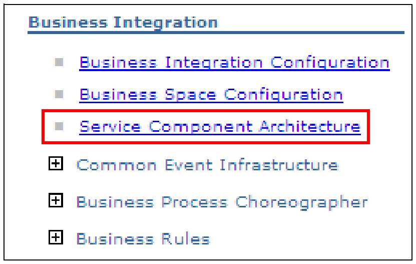


Figure 8-24 Service Component Architecture link

4. In the General Properties section, click the Support the Service Component Architecture components check box (Figure 8-25).

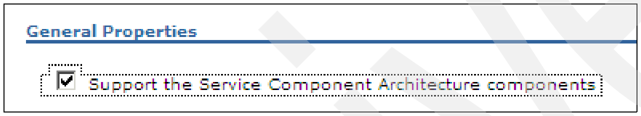


Figure 8-25 SCA support option

5. In the Bus Member Location section, click the Remote radio button and select the **RMS.Messaging** cluster from the drop-down list (Figure 8-26).

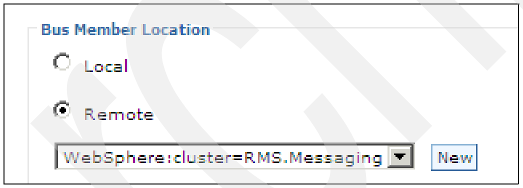


Figure 8-26 Remote bus member

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 8-27 on page 251).

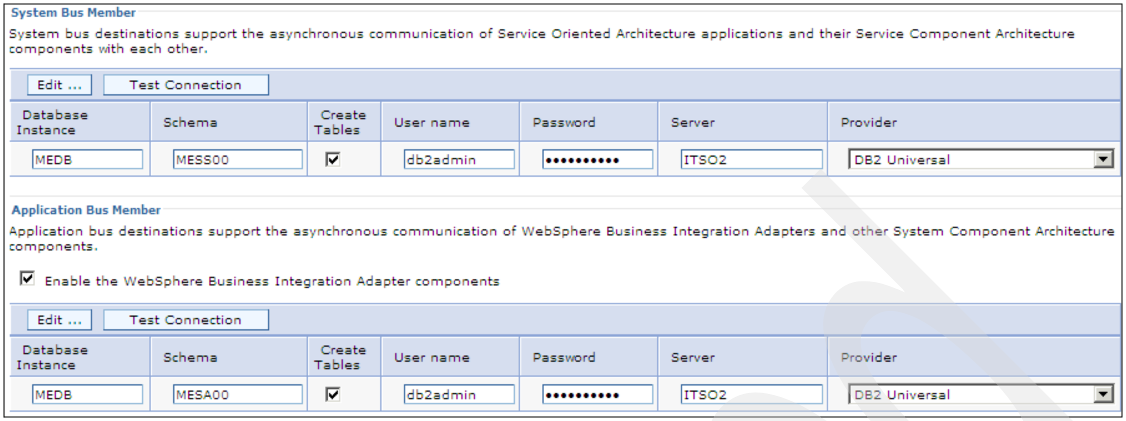


Figure 8-27 SCA.SYSTEM and SCA.APPLICATION bus properties

6. Click **OK**.
7. Click **Save**.
8. 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.

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 administrative console or the bpeconfig.jacl script.

To configure the BPC in the second application target cluster:

1. In the administrative console navigation pane, expand **Servers** and click the **Clusters** link.
2. In the server clusters window, click the **RMS.AppTarget2** link (Figure 8-28).

Figure 8-28 RMS.AppTarget2 link

3. In the Container Settings section, expand **Business Process Choreographer Container Settings** and click the **Business Process Choreographer Containers** link (Figure 8-29).

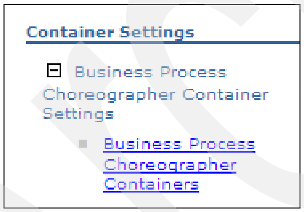


Figure 8-29 Business Process Choreographer Containers link

4. In the Data Source section, perform the following steps (Figure 8-30 on page 253):
   a. In the Database Instance text box, enter the name of the database you configured previously (BPEDB2).
   b. In the Schema Name text box, enter the name of the schema 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 you entered during template-driven deployment (in the Redbooks lab this value was db2admin).

e. Populate the Password text box with the value you entered during template-driven deployment (in the Redbooks lab this value was web1sphere).

f. Populate the Server text box with the host name of the DB2 server (in the Redbooks lab this value was ITSO2).

g. Set the Provider to the appropriate value (in the Redbooks lab the value was DB2 Universal).

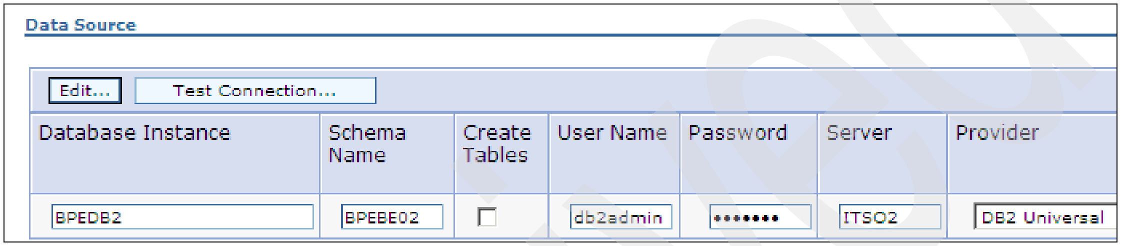


Figure 8-30 BPC data source properties

5. In the Human Task Manager Mail Session section, perform the following steps: (Figure 8-31).

   a. Select the Enable e-mail service check box if your applications use email 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 the Redbooks 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 the Redbooks lab this value was web1sphere)

   e. In the Business Process Choreographer Explorer URL text box, enter the URL for the Explorer client (in the Redbooks lab this value was http://ITSO2/bpc)

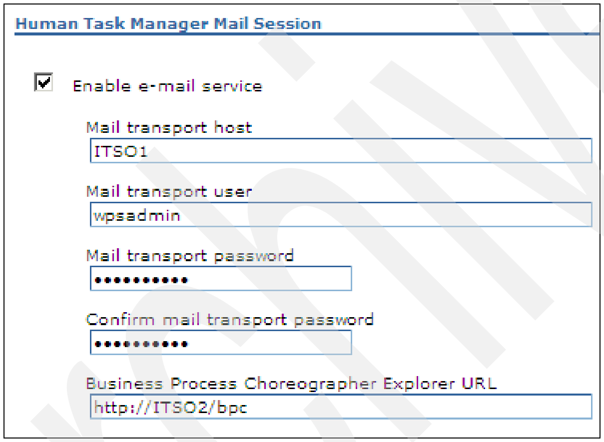


Figure 8-31 Human Task Manager Mail Session properties

6. In the Security section, enter the passwords for the authentication users you configured during template-driven deployment. (For more information about these accounts, see Section 2.2, “Security for a WebSphere Process Server solution” on page 29.) This is shown in Figure 8-32.

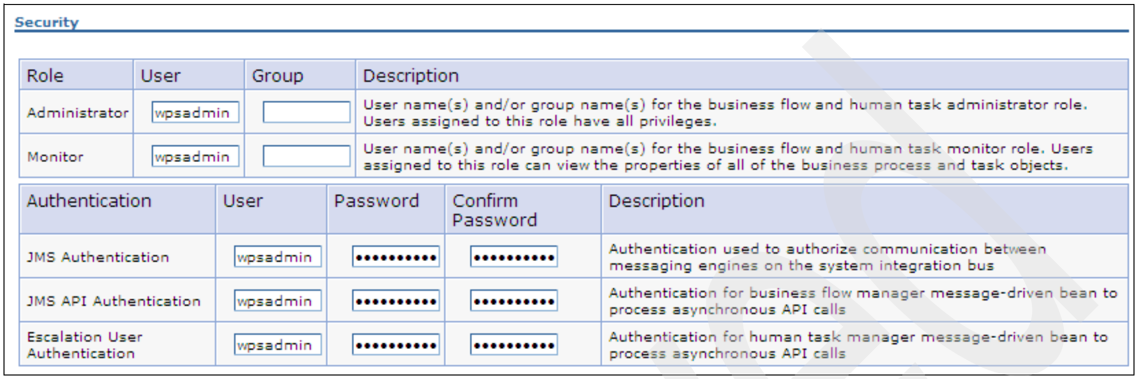


Figure 8-32 Security properties

7. In the State Observers section, if your applications produce CEI events, select the Business Flow Manager or the Human Task Manager (or both if you wish to monitor human tasks and business processes) check boxes for the Common Event Infrastructure Logging row. This is shown in Figure 8-33.

   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.

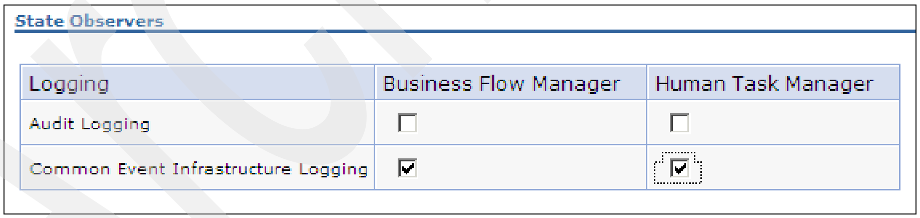


Figure 8-33 State Observers properties

8. In the SCA Bindings section, verify the following information (Figure 8-34).
   – Context Root for the Business Flow Manager: /BFMIF_RMS.AppTarget2
   – Context Root for the Human Task Manager: /HTMIF_RMS.AppTarget2

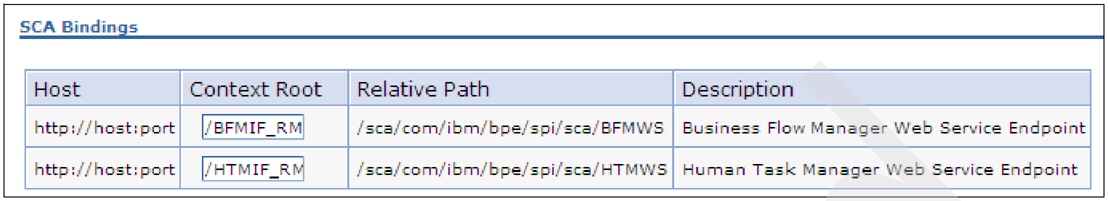


Figure 8-34 SCA Bindings properties

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 8-35.

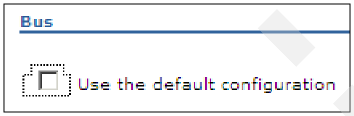


Figure 8-35 Bus properties

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 8-36).

    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.

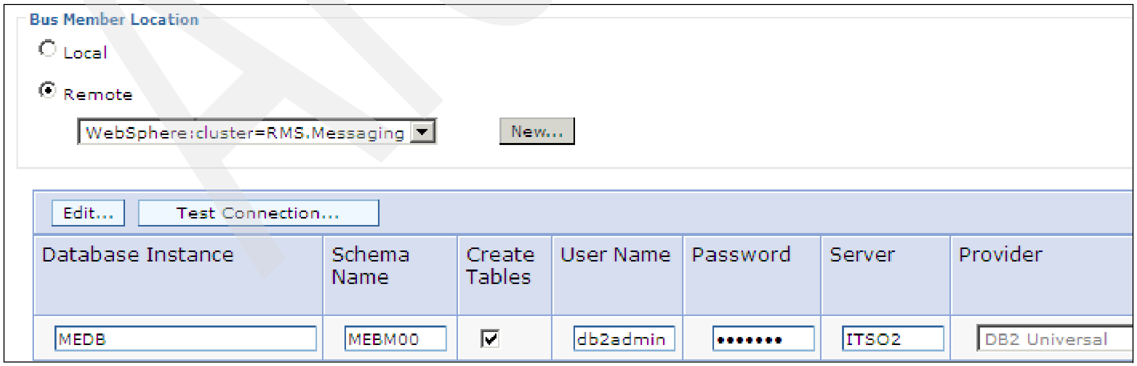


Figure 8-36 Bus Member Location properties

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 you created should also use this destination.

To configure the CEI destination for the additional application cluster, perform the following steps:

1. In the administrative console navigation pane, expand **Servers** and click the **Clusters** link.
2. In the server clusters window, click the **RMS.AppTarget2** link (Figure 8-37).

Figure 8-37 RMS.AppTarget2 link

3. In the Business Integration section, expand **Common Event Infrastructure** and click the **Common Event Infrastructure Destination** link (Figure 8-38).

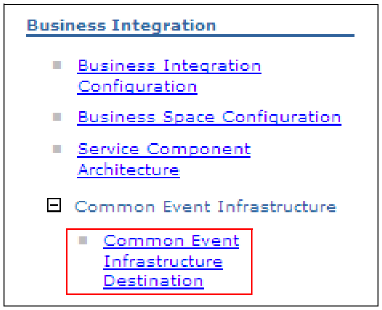


Figure 8-38 CEI Destination link

4. In the General Properties section, select the CEI destination radio button defined at the cell scope. In the Redbooks lab, this value was cell/clusters/RMS.Support/com/ibm/events/configuration/emitter/Default (Figure 8-39).

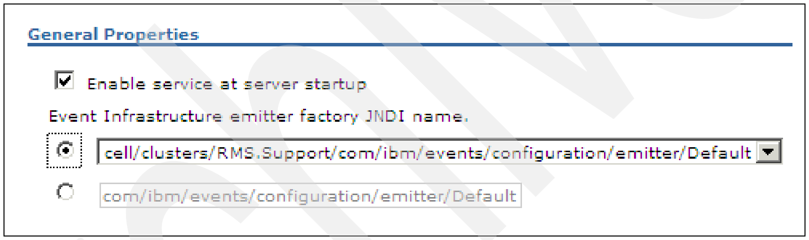


Figure 8-39 Cell-scoped CEI destination

5. Click **OK**.
6. Click the **Save** link to save changes to the master configuration.
7. 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.
8. With the additional application target cluster configured, start the cluster.
9. Verify that the cluster members start without error by checking each member's SystemOut.log for exceptions.

8.3.2 Adding an additional application cluster and an 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 or the messaging infrastructure when there are two application clusters and two messaging clusters is depicted in Figure 8-40 on page 260.

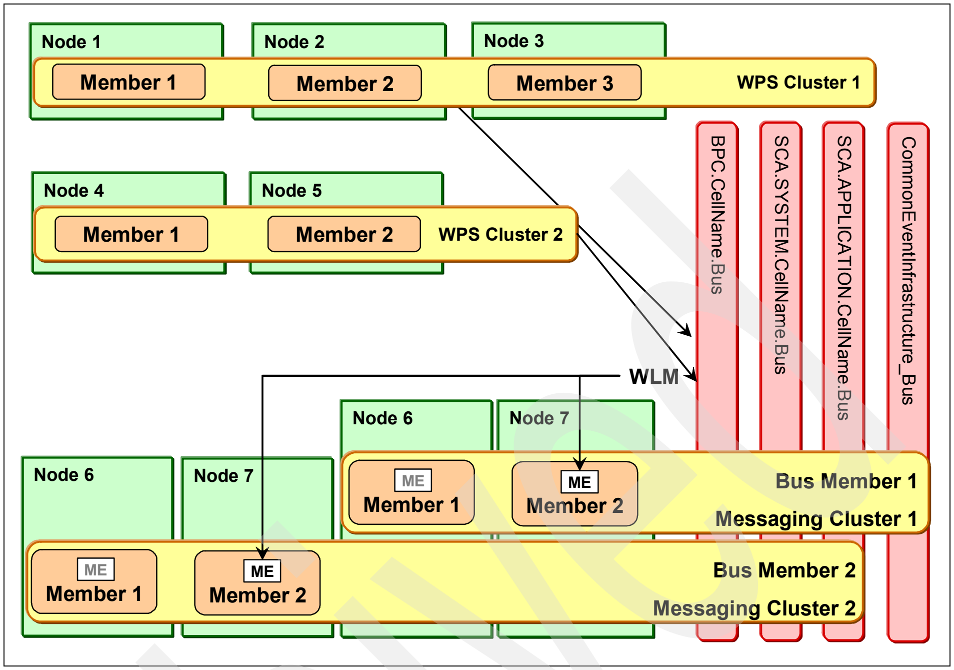


Figure 8-40 Messaging behavior with two application clusters and two messaging clusters

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 8-41 on page 261, 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.

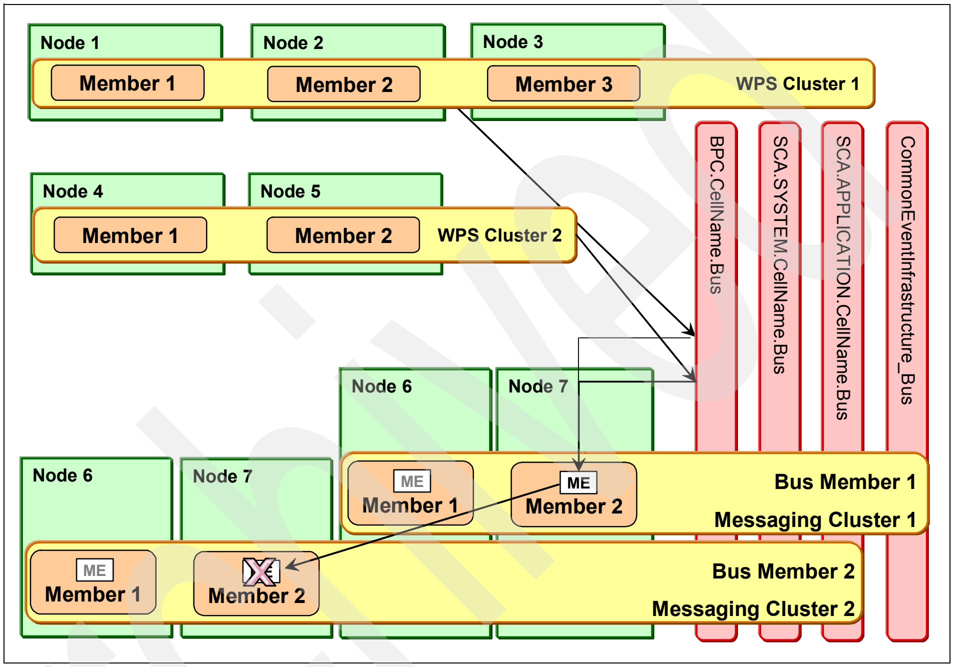


Figure 8-41 Messaging store-and-forward

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 8-42 on page 262.

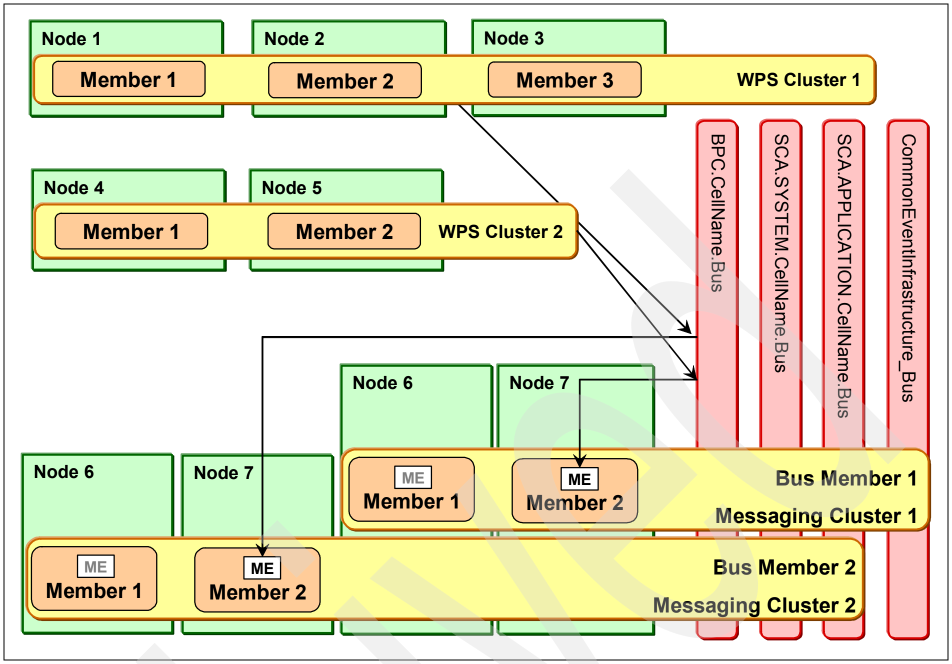


Figure 8-42 Messaging behavior with target significance

In Figure 8-42, 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. Perform the following steps to 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. Perform the following steps to 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 Redbooks publication is represented in Figure 8-43.

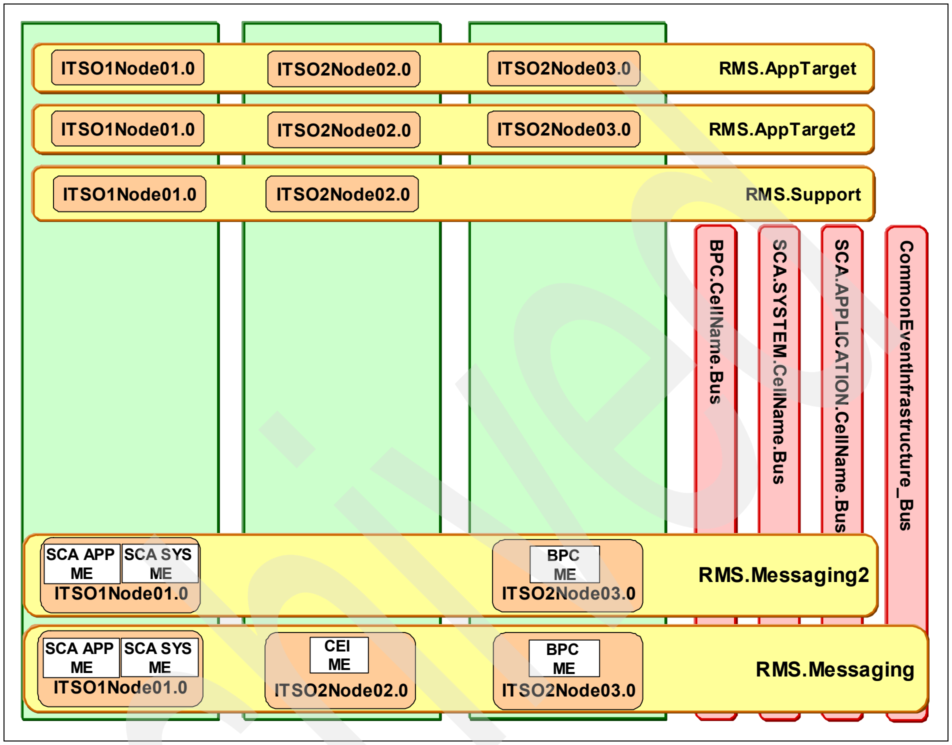


Figure 8-43 Gold topology with two application clusters and messaging clusters

Adding an additional application target cluster

To add an additional application target cluster, follow the steps in 8.3.1, “Adding an additional application cluster” on page 241.

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 265.
2. Creating the second messaging cluster. See Section 266.
3. Configuring SCA support for the additional messaging cluster. See page 269.
4. Configuring target significance. See page 271.

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 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 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 you choose ultimately depends upon the database system 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, perform the following steps:

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.

Creating the second messaging cluster

To create the second messaging cluster in the cell, perform the following steps:

1. In the administrative 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 8-44).

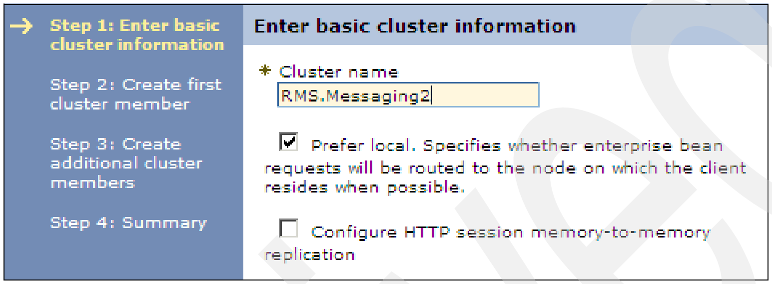


Figure 8-44 New messaging cluster name

4. Click **Next**. The Step 2: Create first cluster member window opens.
5. Perform the following steps to create the first cluster member (Figure 8-45 on page 267):
   a. In the Member name text box, enter the name of the first cluster member (in the Redbooks 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 the Redbooks 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.

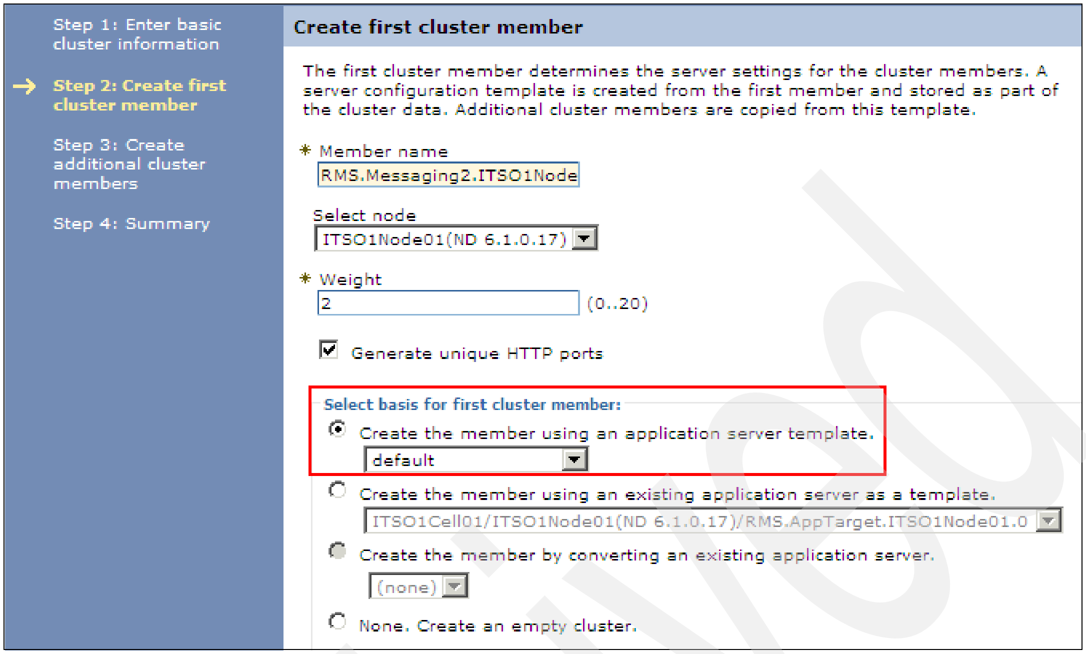


Figure 8-45 Add first member to second messaging cluster

6. Click **Next**. The Step 3: Create additional cluster members window opens.
7. Perform the following steps to create additional cluster members (Figure 8-46 on page 268):
   a. Enter the name of the additional cluster member in the Member name text box. In the Redbooks 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 the Redbooks 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.
   c. In the Select basis for first cluster member section, click the Create the member using an application server template radio button and choose default from the drop-down list.

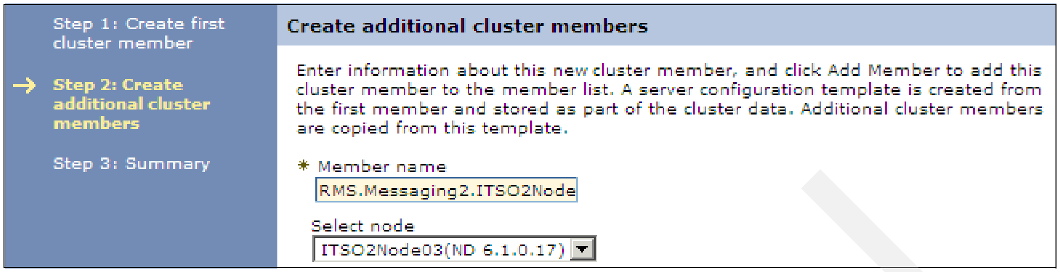


Figure 8-46 Add additional cluster member

8. Click **Add Member**. The name of the additional cluster member should appear in the table below.

> **Note:** Adding additional members during cluster creation is not required. You may create the cluster with one member first and verify the cluster configuration before adding additional members. For demonstration purposes, all cluster members were added at the same time in this example.

9. Click **Next**. The Step 4: Summary window opens.
10. Review your options and click **Finish**.
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.

    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 will need to 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 Section 8.4, “Distributing messaging workload using policies” on page 275.

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 administrative 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, perform the following steps:

1. In the administrative console navigation pane, 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 8-47).

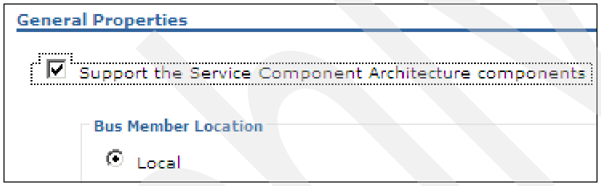


Figure 8-47 Support SCA

Because the current cluster 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 8-48):
   a. Enter MEDB2 in the Database Instance text box.
   b. Enter MESS02 in the Schema text box.
   c. Ensure 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 the following text boxes are populated:
      - User name (in the Redbooks lab, this value was db2admin)
      - Password (in the Redbooks lab, this value was web1sphere)
      - Server (in the Redbooks lab, this value was ITSO2)
   e. Verify DB2 Universal is the selection in the Provider drop-down list.

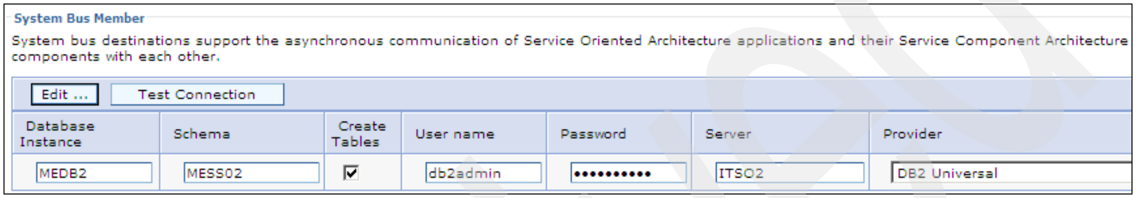


Figure 8-48 System Bus Member properties

6. In the Application Bus Member section, perform the following steps (Figure 8-49):
   a. Enter MEDB2 in the Database Instance text box.
   b. Enter MESA02 in the Schema text box.
   c. Ensure 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 the following text boxes are populated:
      - User name (in the Redbooks lab, this value was db2admin)
      - Password (in the Redbooks lab, this value was web1sphere)
      - Server (in the Redbooks lab, this value was ITSO2)
   e. Verify DB2 Universal is the selection in the Provider drop-down list.

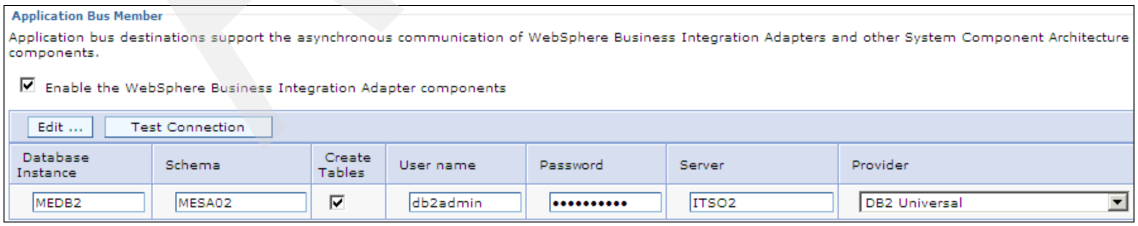


Figure 8-49 Application Bus Member properties

7. Click **OK**.
8. Click the **Save** link at the top of the window to save your changes to the master configuration.
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.

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/

To configure target significance for JMS connection factories and activation specifications, perform the following steps:

1. Download and install the following JMS invocation sample applications:
   - MPGConverterApp.ear
   - JMSInvokerApp.ear
2. Configure the connection factories and activation specifications generated for the JMS invocation sample application for target significance.
   a. In the administrative 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 8-50).

☐	JMSInvoker.MPGConverterProcessExport_CF	JMSInvoker/MPGConverterProcessExport_CF	Default messaging provider
☐	MPGConverter.MPGConverterProcessExport_CF	MPGConverter/MPGConverterProcessExport_CF	Default messaging provider

Figure 8-50 JMS invocation application connection factories

   b. Click the **JMSInvoker.MPGConverterProcessExport_CF** link.

c. In the Connection section, perform the following steps (Figure 8-51):

   i. Ensure 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 the Redbooks lab, this value was SCA.APPLICATION.ITSO1Cell01.Bus.

   ii. In the Target text box, enter the name of the SCA.APPLICATION bus messaging engine you want the application to use.

      In the Redbooks 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.

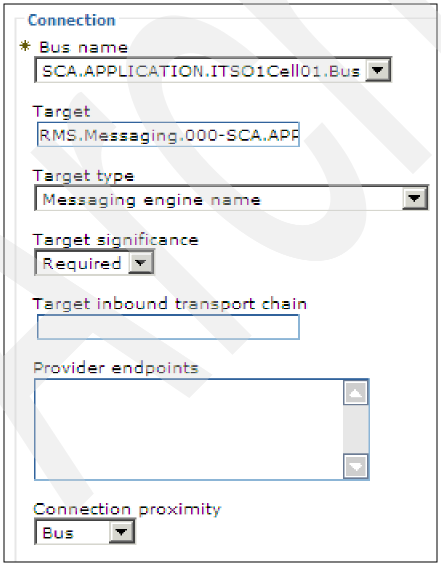


Figure 8-51 Connection factory properties

d. Click **OK**.

e. Click the **Save** link at the top of the window to save your changes to the master configuration.

f. 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.

g. Repeat the previous steps to configure target significance for MPGConverter.MPGConverterProcessExport_CF.

h. In the administrative console, expand **Resources** → **JMS** and click the **Activation specifications** link.

i. You should see two new activation specifications associated with the JMS invocation sample application (Figure 8-52).

☐	JMSInvoker.MPGConverterProcessExport_AS	JMSInvoker/MPGConverterProcessExport_AS	Default messaging provider
☐	MPGConverter.MPGConverterProcessExport_AS	MPGConverter/MPGConverterProcessExport_AS	Default messaging provider

Figure 8-52 JMS invocation application activation specifications

j. Click the **JMSInvoker.MPGConverterProcessExport_AS** link.

k. In the Destination section, perform the following steps (Figure 8-53 on page 274):

i. Ensure 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 the Redbooks lab, this value was SCA.APPLICATION.ITSO1Cell01.Bus.

ii. In the Target text box, enter the name of the SCA.APPLICATION bus messaging engine you want the application to use.

In the Redbooks 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.

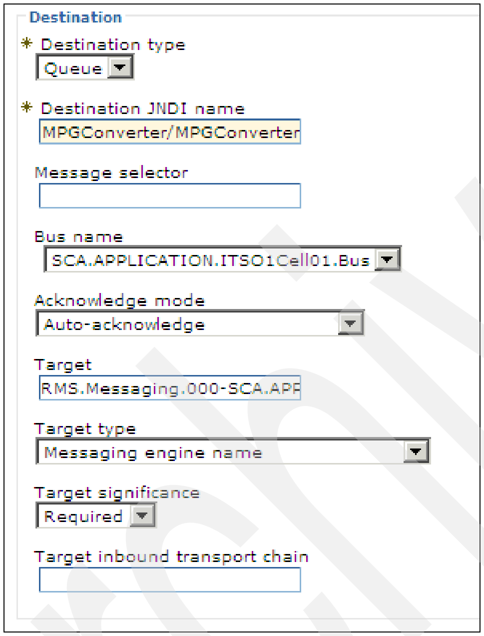


Figure 8-53 Activation specification properties

l. Click **OK**.

m. Click the **Save** link at the top of the window to save your changes to the master configuration.

n. If you have automatic synchronization enabled, when the synchronization process is complete, you should see the following message:

```
The configuration synchronization complete for cell.
```

o. Click **OK**. Otherwise, manually synchronize the changes when you are done.

p. Repeat the previous steps to configure target significance for JMSInvoker.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.

8.4 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 8-54.

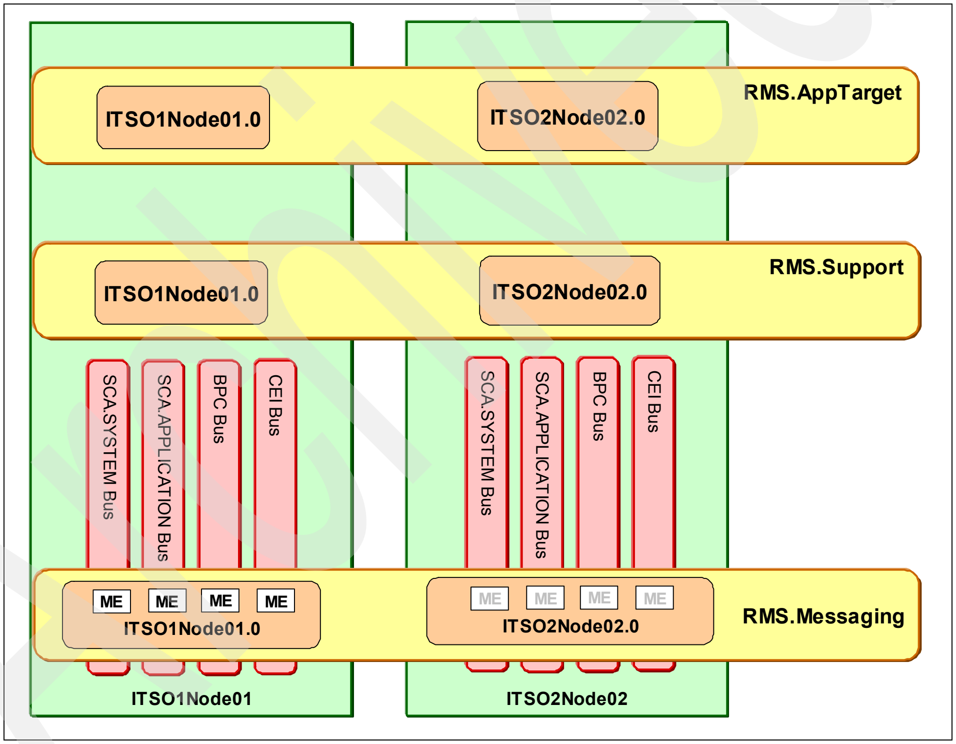


Figure 8-54 Default messaging engine behavior

Here, each of the four messaging engines is in state of “Started” on ITSO1Node01.0. The four messaging engines on ITSO2Node02.0 are all in state “Joined.” 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, and 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 Redbooks 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 8-55 on page 277.

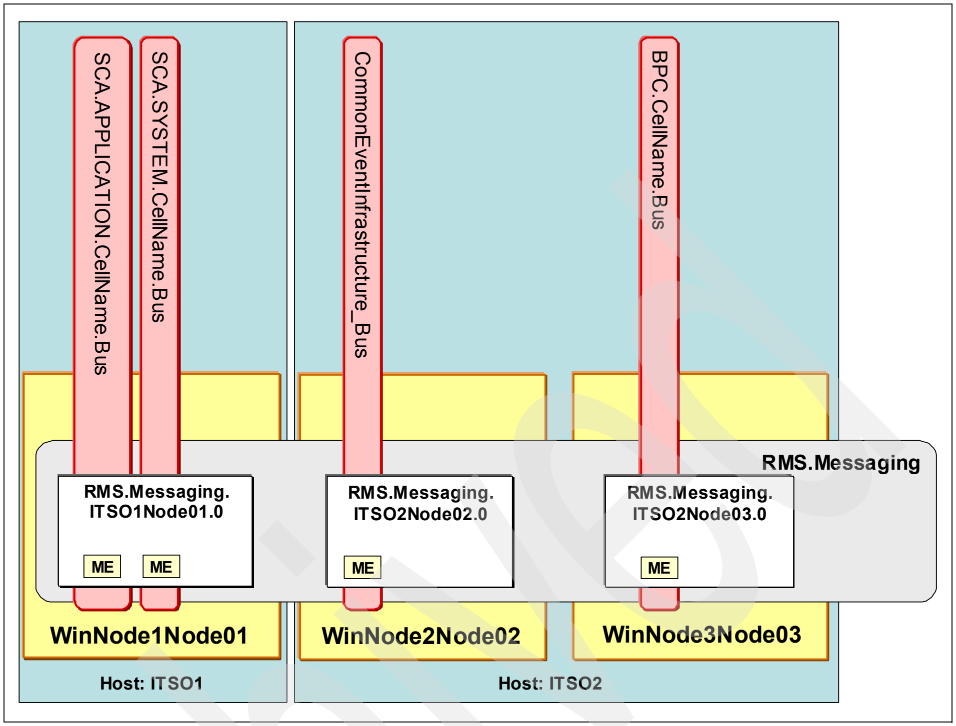


Figure 8-55 Using policies to run messaging engines on separate servers

8.4.1 Create the SCA.SYSTEM messaging engine policy

In order 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 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 Redbooks publication, perform the following steps:

1. In the navigation pane of the administrative console, expand **Servers** → **Core groups** and select the **Core group settings** link (Figure 8-56).

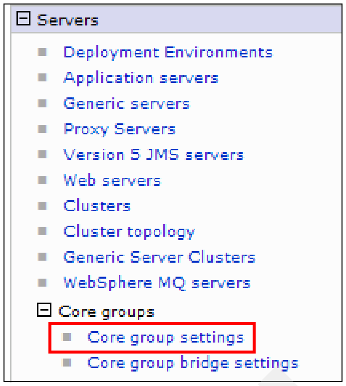


Figure 8-56 Core group settings link

2. Click the **DefaultCoreGroup** link (Figure 8-57).

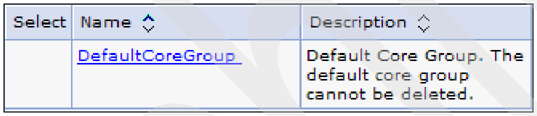


Figure 8-57 DefaultCoreGroup link

3. In the Additional Properties section, click the **Policies** link (Figure 8-58).

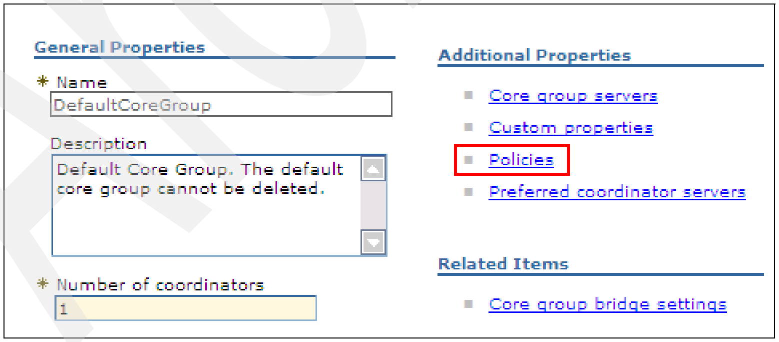


Figure 8-58 Policies link

4. Click the **New** button (Figure 8-59).

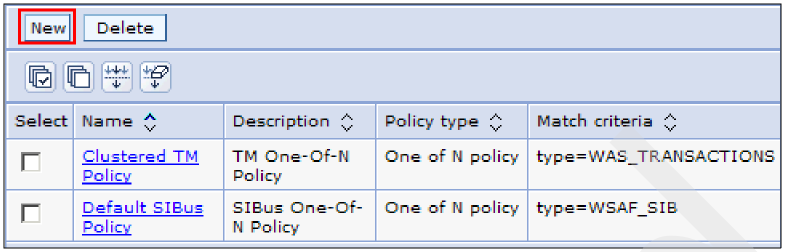


Figure 8-59 Create new policy

5. In the General Properties section, for the Policies text box, select **One of N policy** from the drop-down list (Figure 8-60).

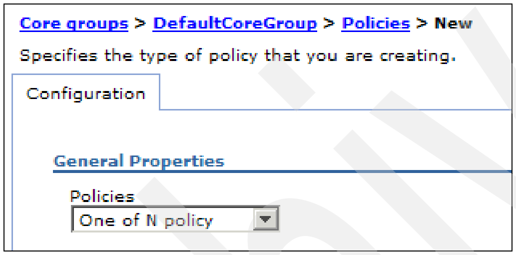


Figure 8-60 Choose policy type

6. Click **Next**.
7. In the General Properties section, perform the following steps (Figure 8-61 on page 280):
   a. For the Name text box, enter SCA_SYS_ME_Policy.
   b. Ensure the Policy type text box is set to One of N 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 assures 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.

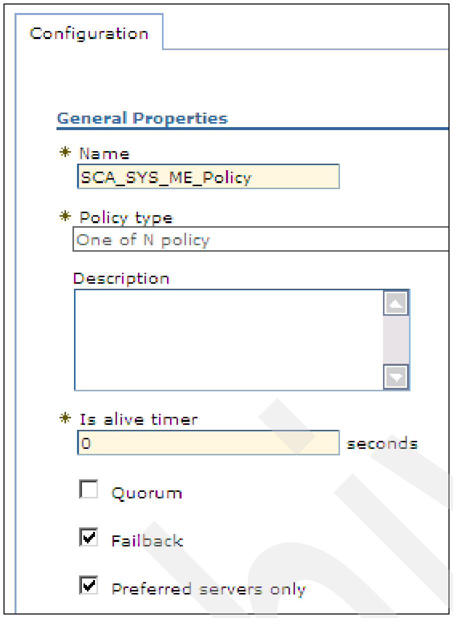


Figure 8-61 New SCA_SYS_ME_Policy

8. Click **OK**. You should be returned to the core groups window with the following message at the top of the window (Figure 8-62):

```
The policy must have at least one match criteria defined.
```

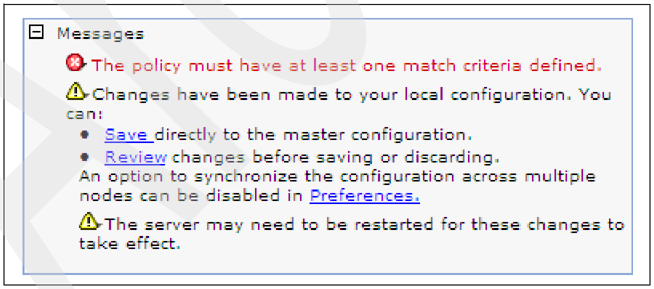


Figure 8-62 Error: No match criteria defined

9. In the Additional Properties section, click the **Match Criteria** link (Figure 8-63).

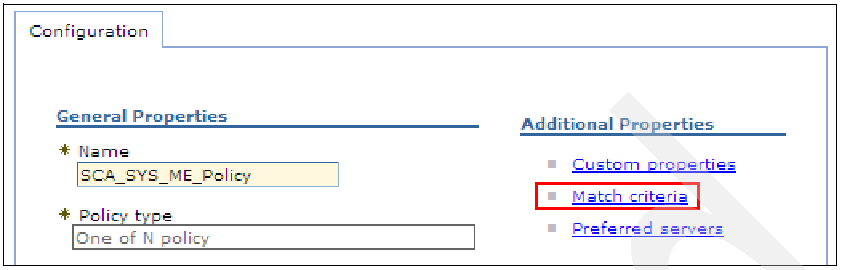


Figure 8-63 Match criteria link

10..Click the **New** button.

11.In the General Properties section, perform the following steps (Figure 8-64):

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.

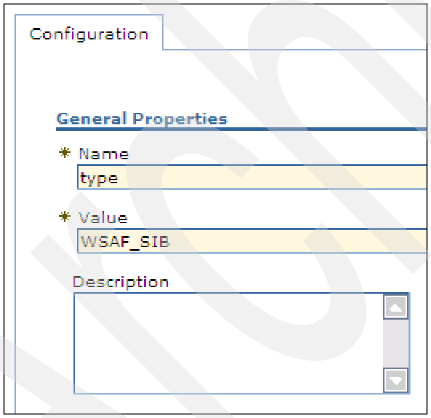


Figure 8-64 Type match criteria

12.Click **OK**.

13.At the Match criteria window, click the **New** button.

14.In the General Properties section, perform the following steps (Figure 8-65):

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 the Redbooks lab, this value was RMS.Messaging.

c. (Optional) Enter a description of the match criteria.

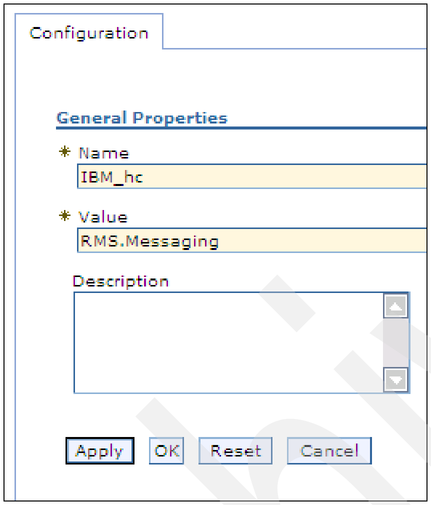


Figure 8-65 Cluster match criteria

15.Click **OK**.

16.When you are returned to the Match criteria window, click the **New** button.

17. In the General Properties section, perform the following steps (Figure 8-66):

   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 the Redbooks lab, this value was SCA.SYSTEM.ITSO1Cell01.Bus.

   c. (Optional) Enter a description of the match criteria.

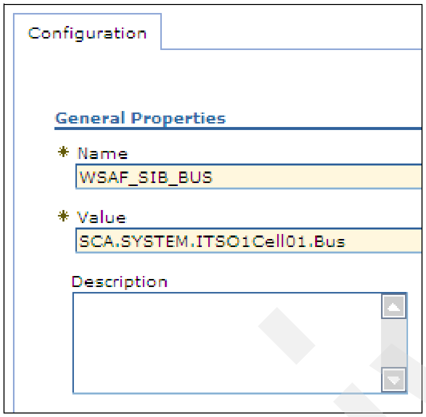


Figure 8-66 Bus match criteria

18. Click **OK**.

19. In the Match criteria window, click the **New** button.

20. In the General Properties section, perform the following steps (Figure 8-67):

   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 the Redbooks lab, this value was RMS.Messaging.000-SCA.SYSTEM.ITSO1Cell01.Bus.

   c. (Optional) Enter a description of the match criteria.

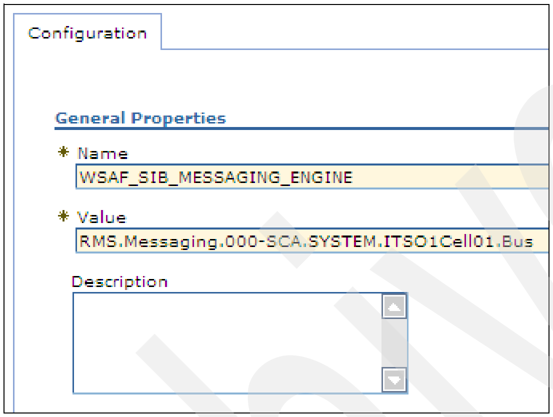


Figure 8-67 Messaging engine match criteria

21. Click **OK**. In the match criteria window, you should see all four of the criteria you created (Figure 8-68).

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 8-68 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.

22. Click the **SCA_SYS_ME_Policy** link in the breadcrumb trail at the top of the core groups window (Figure 8-69).

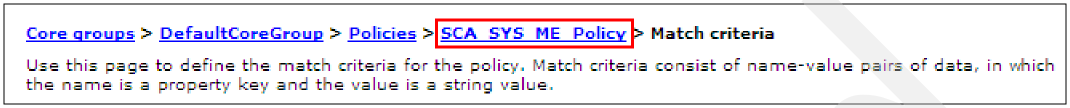


Figure 8-69 Breadcrumb trail

23. In the Additional Properties section, click the **Preferred servers** link.

24. In the Core group servers section, select the first preferred server <HostNameNodeName>/<MessagingClusterMember> and click **Add**. In the Redbooks lab, this value was ITSO1Node01/RMS.Messaging.ITSO1Node01.0. This is shown in Figure 8-70.

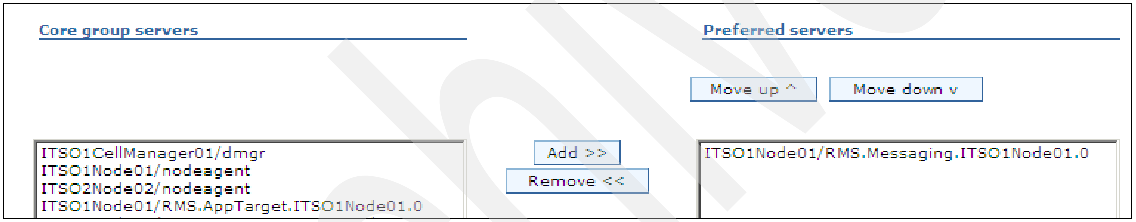


Figure 8-70 Add the first preferred server

25. In the Core group servers section, select the second preferred server <HostNameNodeName>/<MessagingClusterMember> and click **Add**. In the Redbooks lab, this value was ITSO2Node02/RMS.Messaging.ITSO2Node02.0). This is shown in Figure 8-71.

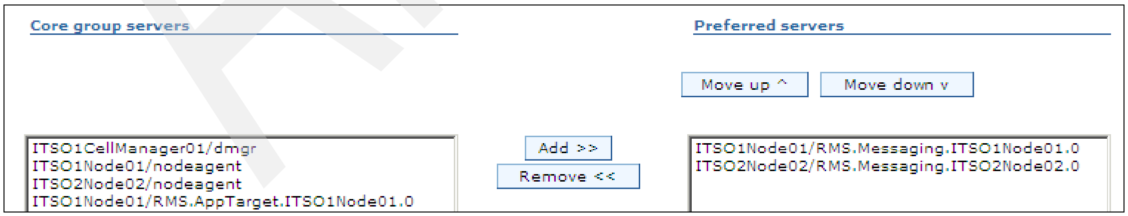


Figure 8-71 Add the second preferred server

26. In the Core group servers section, select the third preferred server <HostNameNodeName>/<MessagingClusterMember> and click **Add** (Figure 8-72). In the Redbooks lab, this value was ITSO2Node03/RMS.Messaging.ITSO2Node03.0.

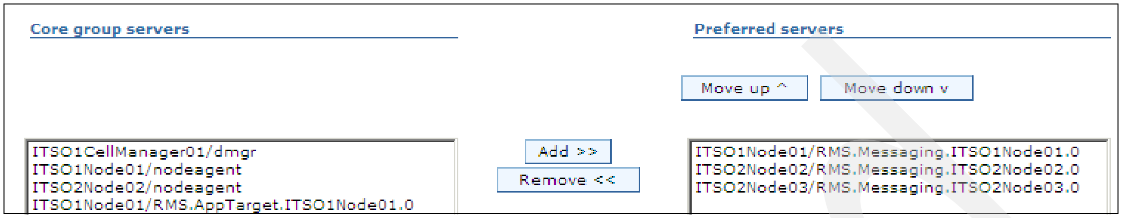


Figure 8-72 Add the third preferred server

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.

27. Click **OK**.

28. Click the **Save** link at the top of the window (Figure 8-73).

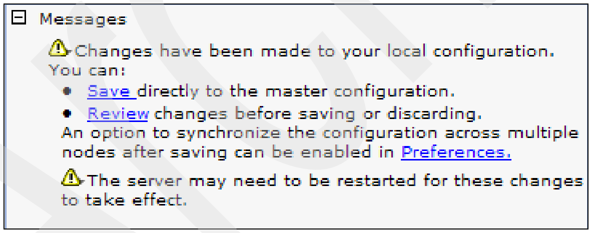


Figure 8-73 Save changes to the master configuration

29. 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.

8.4.2 Create the SCA.APPLICATION messaging engine policy

The second policy 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 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 Redbooks 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.
3. Click **Next**.
4. In the General Properties section, perform the following steps:
   a. For Name, enter SCAAppME000 (ME zero zero zero).
   b. Ensure 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.
5. 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
   ```
6. In the Additional Properties section, click the **Match Criteria** link.
7. Follow the steps in Section 8.4.1, “Create the SCA.SYSTEM messaging engine policy” on page 277, to create the match criteria shown in Table 8-1 on page 288.

Table 8-1 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
WSAF_SIB_MESSAGING_ENGINE	<ClusterName>.000-SCA.APPLICATION.<CellName>.Bus; for example, RMS.Messaging.000-SCA.APPLICATION.ITSO1Cell01.Bus

8. Click the **SCAAppME000** link in the breadcrumb trail at the top of the core groups window.
9. In the Additional Properties section, click the **Preferred servers** link.
10. In the Core group servers section, select the first preferred server <HostNameNodeName>/<MessagingClusterMember> and click **Add**. In the Redbooks lab, this value was ITSO1Node01/RMS.Messaging.ITSO1Node01.0.
11. In the Core group servers section, select the second preferred server <HostNameNodeName>/<MessagingClusterMember> and click **Add**. In the Redbooks lab, this value was ITSO2Node02/RMS.Messaging.ITSO2Node02.0.
12. In the Core group servers section, select the third preferred server <HostNameNodeName>/<MessagingClusterMember> and click **Add**. In the Redbooks 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.

13. Click **OK**.

14. Click the **Save** link at the top of the window.
15. 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.

8.4.3 Creating the Common Event Infrastructure messaging engine policy

The third policy 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 Redbooks 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.
3. Click **Next**.
4. In the General Properties section, perform the following steps:
    a. For Name, enter CEI_ME000 (ME zero zero zero)
    b. Ensure 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.
5. 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
    ```

6. In the Additional Properties section, click the **Match Criteria** link.

7. Follow the steps in Section 8.4.2, “Create the SCA.APPLICATION messaging engine policy” on page 287, to create the match criteria shown in Table 8-2.

Table 8-2 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

8. Click the **CEI_ME000** link in the breadcrumb trail at the top of the core groups window.
9. In the Additional Properties section, click the **Preferred servers** link.
10. In the Core group servers section, select the first preferred server <HostNameNodeName>/<MessagingClusterMember> and click **Add**. In the Redbooks lab, this value was ITSO2Node02/RMS.Messaging.ITSO2Node02.0.
11. In the Core group servers section, select the second preferred server <HostNameNodeName>/<MessagingClusterMember> and click **Add**. In the Redbooks lab, this value was ITSO1Node01/RMS.Messaging.ITSO1Node01.0.
12. In the Core group servers section, select the third preferred server <HostNameNodeName>/<MessagingClusterMember> and click **Add**. In the Redbooks lab, this value was ITSO2Node03/RMS.Messaging.ITSO2Node03.0.

> **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.

13. Click **OK**.

14. Click the **Save** link at the top of the window.
15. .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.
16. You will be returned to the **Policies** window and see your newly created policy.

8.4.4 Creating the Business Process Choreographer messaging engine policy

The fourth policy 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 the Redbooks 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 Redbooks 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.
3. Click **Next**.
4. In the **General Properties** section, perform the following steps:
    a. For Name, enter BPC_ME000 (ME zero zero zero).
    b. Ensure 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.
5. 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
    ```

6. In the Additional Properties section, click the **Match Criteria** link.

7. Follow the steps in Section 8.4.3, “Creating the Common Event Infrastructure messaging engine policy” on page 289, to create the following match criteria as shown in Table 8-3.

Table 8-3 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

8. Click the **BPC_ME000** link in the breadcrumb trail at the top of the core groups window.
9. In the Additional Properties section, click the **Preferred servers** link.
10. In the Core group servers section, select the first preferred server <HostNameNodeName>/<MessagingClusterMember> and click **Add**. In the Redbooks lab, this value was ITSO2Node03/RMS.Messaging.ITSO2Node03.0.
11. In the Core group servers section, select the second preferred server <HostNameNodeName>/<MessagingClusterMember> and click **Add**. In the Redbooks lab, this value was ITSO1Node01/RMS.Messaging.ITSO1Node01.0.
12. .In the Core group servers section, select the third preferred server <HostNameNodeName>/<MessagingClusterMember> and click **Add**. In the Redbooks lab, this value was ITSO2Node02/RMS.Messaging.ITSO2Node02.0.

> **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.

13. Click **OK**.

14. Click the **Save** link at the top of the window.

15. 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.

16. You should be returned to the **Policies** window and you should see your newly created policy.

17. 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.

8.4.5 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 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 the Redbooks 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.

2. You 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.
    ```

3. On node two, open the SystemOut.log file for messaging Cluster Member 2. In the Redbooks 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.
4. 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 Started.
Messaging engine RMS.Messaging.000-BPC.ITSO1Cell01.Bus is in state
Joined.
```

5. On node two, open the SystemOut.log file for messaging Cluster Member 3. In the Redbooks 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.
6. 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.
```

9

Monitoring the production topology

A production topology typically spans across multiple channels and disparate systems to provide integrated services. It is an important requirement for any enterprise to manage and monitor several systems that are involved in building composite applications.

In this chapter, we introduce some of the basics of service-oriented architecture (SOA) management. This chapter contains the following sections:

9.1 Monitoring the SOA environment

A typical composite business application spans across multiple architectural layers, such as business processes, service components, service consumers, and operational IT systems. Managing the SOA infrastructure upon which composite business applications are built needs careful attention to address various challenges. Because enterprise SOA infrastructures have matured over a period of time, the need for management solutions has emerged to address the following issues:

- Understanding the end-to-end flow of business processes.
- Meeting the required Non Functional Requirements (NFRs) and Service Level Agreements (SLAs).
- Ensuring that operational systems that provide integrated services are available as per the SLA requirements.
- Monitoring potential security threats for the SOA infrastructure.
- Understanding the relationship between various participating services in the composite business application.
- Analyzing performance loop holes.

Refer to the following Redbooks publications for a detailed discussion about SOA management and its challenges:

- *Best Practices for SOA Management*, REDP-4233
- *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240

For composite business applications, SOA management requirements span across multiple architectural layers. These layers are best explained by the IBM SOA Foundation Reference Architecture, shown in Figure 9-1 on page 297.

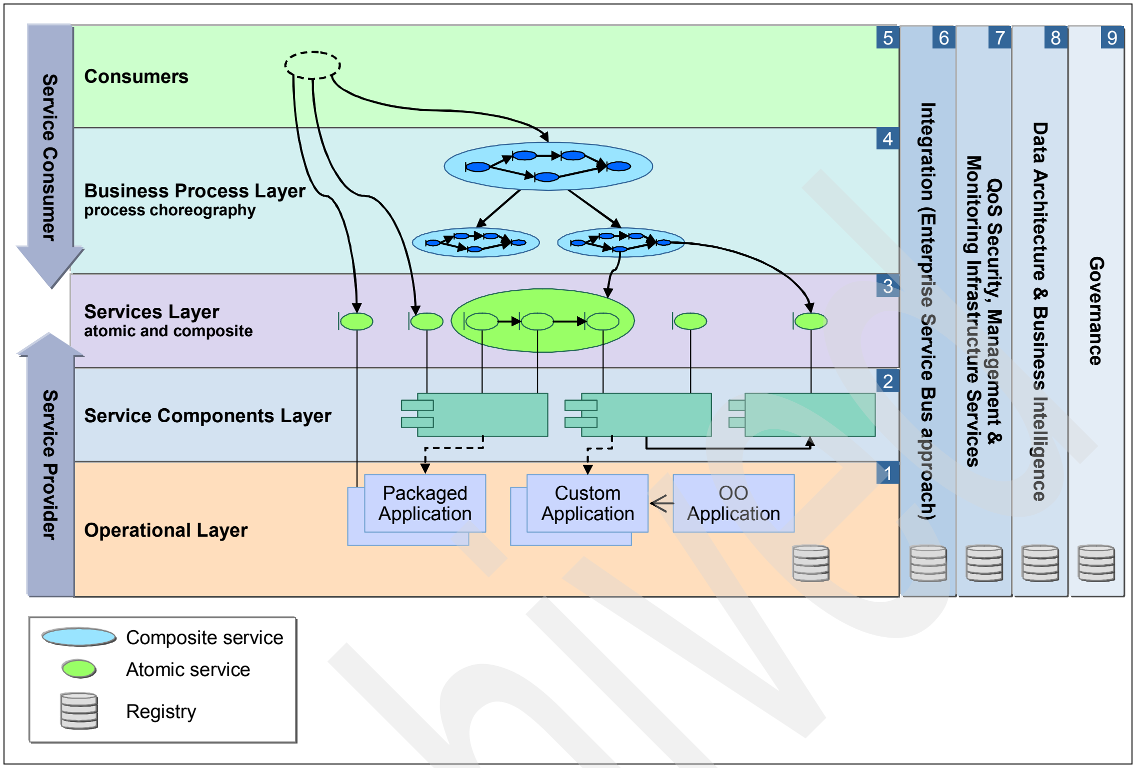


Figure 9-1 Layers in the SOA Foundation Reference Architecture

The most important layer from the perspective of composite business applications is the services layer, which integrates various operational systems, service components, business policies, and service consumers through provisioned channels. You can leverage an SOA management solution that is implemented for business services across the enterprise to achieve the following results:

- Understand how services relate to each other for providing a business functionality.
- Assess the dependency of business services on IT infrastructure and the business process layer.
- Define and refine business related goals.
- Control the message flow for services infrastructure through management mediations, such as filter, log, transform, and route. This is helpful because the composite business application spans across multiple architectural layers.
- Understand the performance of a specific service with transactional metrics of each request.

- Provide relationships between service requests and the implementation artifacts. For example, a JDBC request, Java Beans, SCA Plain Old Java Object (POJO) component, and so on.
- Monitor the health of the operational systems that support the services implementation (for example, an Enterprise Resource Planning (ERP) system, a custom billing system, and so on).
- Provide inputs to solve operational problems (for example, low end-to-end response times, exhausted thread pool, and so on).

It is evident that there is a cost that is associated with implementing and maintaining such SOA management solutions across the enterprise. A centralized SOA management policy set across the enterprise by the SOA governance body is a critical success factor for such solutions.

9.1.1 IBM Tivoli Composite Application Manager for SOA

IBM Tivoli Composite Application Manager (ITCAM) for SOA monitors, manages, and controls business services that run on supported application servers. ITCAM for SOA provides some of the following key features:

- Provides a comprehensive and configurable dashboard for viewing service monitoring data through Tivoli Enterprise Portal Server (TEPS):
  - Service Management Agent Environment shows:
    - Performance Summary
    - Message Summary
    - Fault Summary
  - Service Management Agent shows monitoring agent configuration summary, data collectors, monitoring profiles, and filters.
  - Mediation configuration shows mediation primitive entries for mediation on Service Component Architecture (SCA) components in WebSphere Process Server or WebSphere Enterprise Service Bus.
  - Message arrival view shows the message arrival rate and events based on the message arrival critical situation, which you can use for looking at the throughput rates.
- Provides heterogeneous platform support:
  - Supports Microsoft®.NET, BEA WebLogic, IBM WebSphere Application Server (WAS) JAX-RPC and SCA, IBM DataPower® appliance, WebSphere Application Server Community Edition, and JBoss.
  - CICS Transaction Server environment
  - SAP® NetWeaver environment

- Leverages Tivoli Enterprise Portal situations to check thresholds. ITCAM for SOA provides some predefined situations that you can customize. Some of the predefined situations are:
  - Number of messages received by a service within a time window
  - Size of the messages
  - Faults
- Provides a list of services and operations that are monitored in the environment.
- Leverages Tivoli Enterprise Portal workflow and policy editor for threshold-triggered action sequences.
- Provides basic mediation support with the ability to filter or reject Web services call messages from a particular client or service. It can log request and response messages for analysis.

The building blocks of ITCAM for SOA consists of the following logical components:

- Data Collector

  This building block collects the services data that is appropriate to the environment in a non-intrusive fashion.
- Enterprise Monitoring Agent

  This building block works as a data consolidator. It is responsible for collecting data from data collectors and forwarding them to Tivoli Monitoring Server.
- Web Services Navigator

  This building block is an Eclipse-based navigator that can process the collected log files and provide visual representations of various characteristics of the monitored data.
- Mediation Primitives

  This building block allows control of mediation primitives within the WebSphere Enterprise Bus and WebSphere Process Server.

To know more about the ITCAM product family, refer to the ITCAM for SOA Infocenter at the following Web page:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?topic=/com.ibm.itcamsoa.doc/welcome.htm

Or read the Redbooks publication *IBM Tivoli Composite Application Manager Family Installation, Configuration, and Basic Usage*, SG24-7151.

9.1.2 ITCAM for SOA and Business Process Management

We now have some ideas about what ITCAM for SOA is and how it fits into the enterprise-hosting infrastructure for providing management and monitoring functionalities. Let us now look at the integration of IBM WebSphere Dynamic Process Edition with ITCAM for SOA.

Figure 9-2 shows the components of how a composite business application environment interacts with the Tivoli Enterprise Monitoring Server infrastructure.

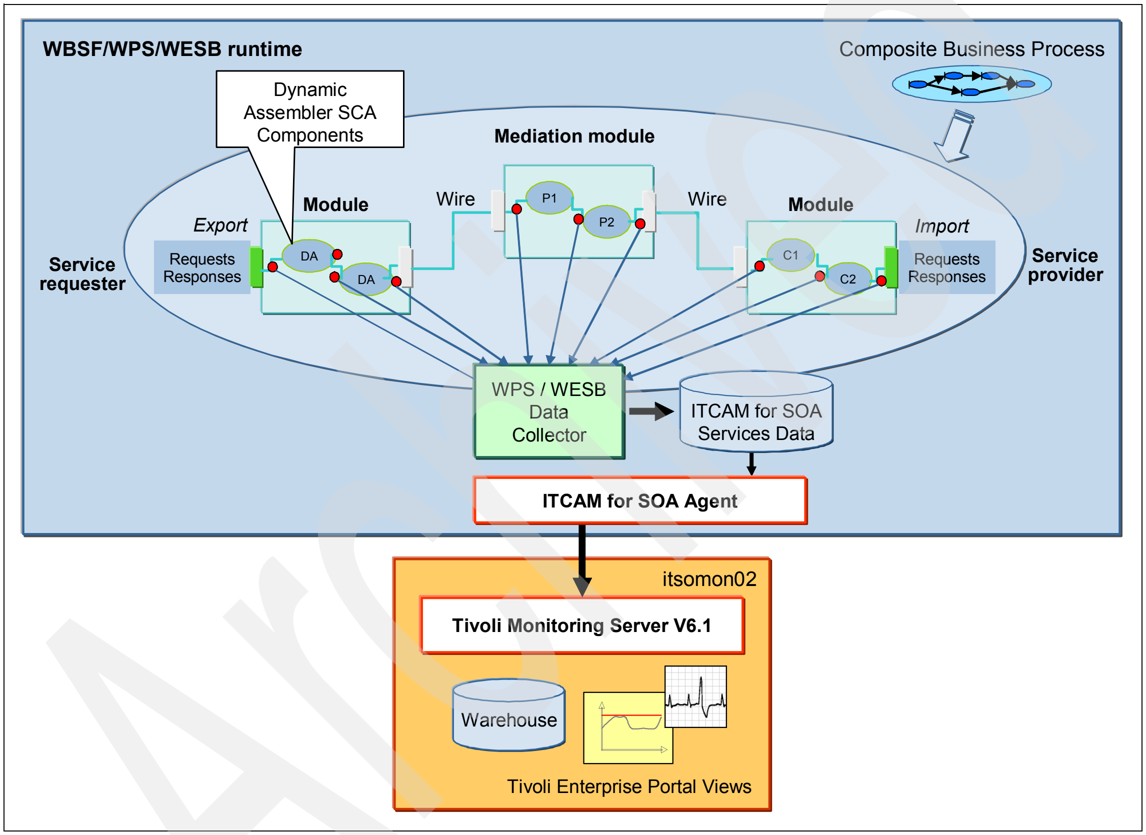


Figure 9-2 Composite business application environment with Tivoli Enterprise Monitoring server

Integration between IBM WebSphere Dynamic Process Edition and ITCAM for SOA is achieved through support for underlying SCA components. SCA modules, mediation modules, and Dynamic Assembler (DA) are SCA components. The data collector implemented for WebSphere Process Server SCA components captures the messages that flow through these special types of SCA components. The Tivoli Enterprise Monitoring Server captures, collates, and transforms this data into different views and other types of monitoring data (Web services data for example).

As the DA represents the service endpoints that are associated with business policies, monitoring DA components is an effective and efficient way to obtain an end-to-end view of the business process flow. Therefore, it is not sufficient to just capture monitoring information for DA invocations for a business process. We suggest that you also monitor your infrastructure and make use of different types of data capturing techniques, such as Web services, transactions, database, operating systems, and so on, to get a holistic picture of the operational environment.

9.2 Monitoring the infrastructure

This section introduces two product offerings for monitoring your infrastructure.

- “ITCAM for WebSphere”
- “IBM Tivoli Monitoring” on page 304

9.2.1 ITCAM for WebSphere

ITCAM for WebSphere enables you to analyze the health of the WebSphere Application Server and the transactions that are invoked in it. It is able to trace the transaction execution to the detailed method-level information, and connects transactions that spawn from one application server and invokes services from other application servers, including mainframe applications in IMS or CICS.

ITCAM for WebSphere provides a flexible level of monitoring, from an non-intrusive production ready monitor, to a detailed deep-dive tracing for problems of locking or even memory leaks. ITCAM for WebSphere provides a separate interactive Web console and also allows monitoring data to be displayed on the Tivoli Enterprise Portal.

An overview of the architecture of ITCAM for WebSphere is provided in Figure 9-3 on page 302.

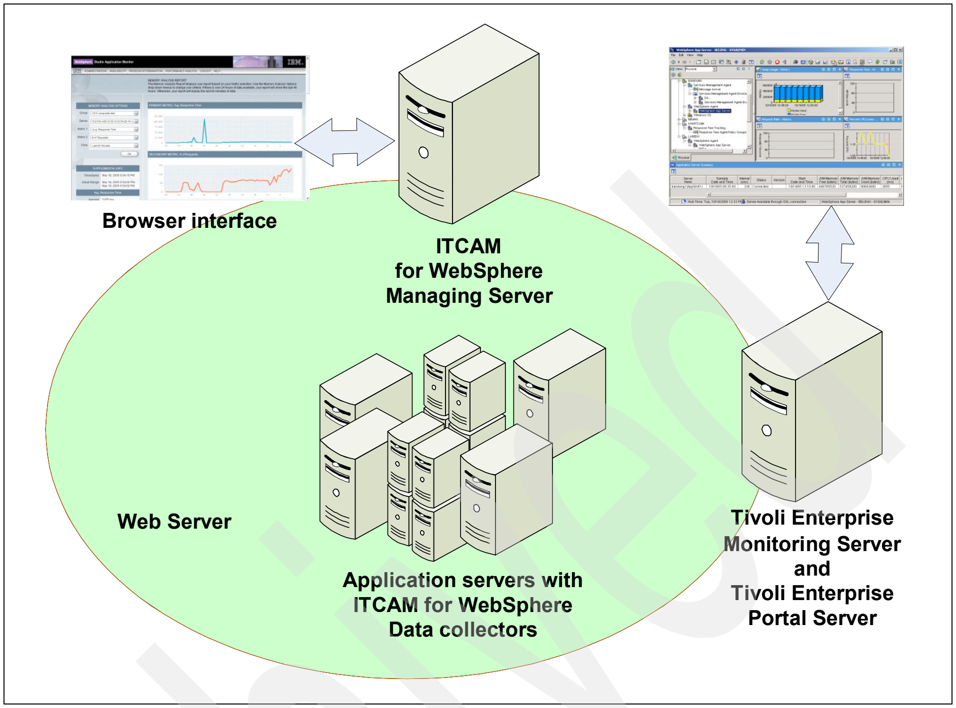


Figure 9-3 ITCAM for WebSphere architecture

ITCAM for WebSphere is a solution that is primarily aimed at second-line support to perform diagnosis of J2EE-based applications and services. ITCAM for WebSphere provides data collectors that allow you to collect data shown in Table 9-1 on page 303.

Table 9-1 Data content classification

Classification	Data
Command and control data	Configuring and unconfiguring data collector
User actions related to threads	► Starting and stopping JVM threads ► Changing thread priorities ► Getting thread priorities and thread status ► Requesting drill information to see cookies, and so on ► Generating thread dumps ► Getting thread stack traces
System information	► Application server information ► Operating system platform information ► JVM information
Application information	► All the applications installed on the monitored application server ► Application binaries and location information ► Thread pool information related to Java Message Service (JMS), Java 2 Enterprise Edition (J2EE) Connector Architecture (JCA), Java Transaction API (JTA), Servlet, Enterprise JavaBeans (EJB), and so on ► Data source information
Performance data	All performance monitor interface (PMI) data
Transport data	► Object Request Broker (ORB) data ► SOAP ports
Memory information	► Obtaining JVM Heap Snapshot™ data ► Performing memory leak analysis ► Performing heap dump

To know more about the ITCAM product family, refer to the ITCAM for WebSphere Infocenter at the following Web page:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?toc=/com.ibm.itcamwas.doc/welcome.html

Or read the Redbooks publication *Large-Scale Implementation of IBM Tivoli Composite Application Manager for WebSphere and Response Time Tracking*, REDP-4162.

9.2.2 IBM Tivoli Monitoring

IBM Tivoli Monitoring is an enterprise class monitoring solution. It has been designed to provide access to information that is crucial to daily operations. This includes the availability and performance of components, applications, and services within your enterprise infrastructure. IBM Tivoli Monitoring uses several layers to provide a monitoring framework.

Tivoli Monitoring Services

Tivoli Monitoring Services is the framework for IBM Tivoli Monitoring and comprises all components as well as describing how they interact. Those components include (but are not limited to):

- Tivoli Enterprise Monitoring Server
- Tivoli Enterprise Monitoring Agent
- Tivoli Enterprise Portal Server
- Tivoli Enterprise Portal

Figure 9-4 illustrates the IBM Tivoli Monitoring components and how the interact together.

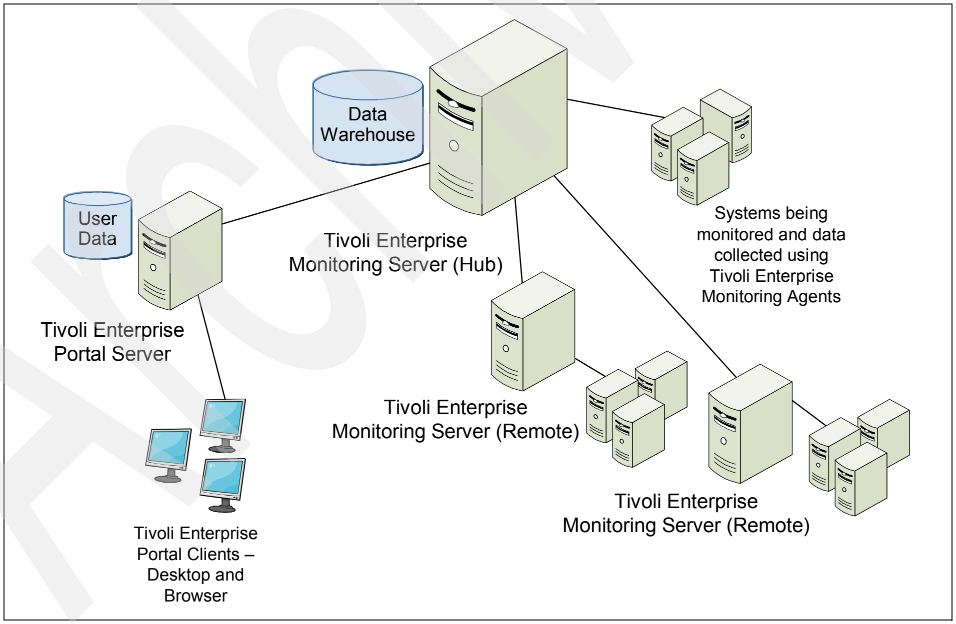


Figure 9-4 IBM Tivoli Monitoring components

Tivoli Enterprise Monitoring Server

Tivoli Enterprise Monitoring Server is the central repository of data that comes from the Tivoli Enterprise Monitoring Agents. It stores the definitions for conditions that indicate a problem with a particular resource and controls the security for your monitoring solution. Each enterprise monitoring solution must contain one hub Tivoli Enterprise Monitoring Server and can include multiple remote Teams, which are used to provide scalability in large installations.

Tivoli Enterprise Monitoring Agent

Tivoli Enterprise Monitoring Agents are data collectors within your monitoring solution. They are installed to gather data from one or more systems that you need to monitor. The data that is collected is sent to a central repository, the Tivoli Data Warehouse. Tivoli Enterprise Monitoring Agents collects information about the attributes of a particular managed system. Examples of agents are:

- Operating System Agent
- Universal Agent

  The Tivoli Universal Agent is a monitoring agent you can configure to monitor any data you require, it is used to gather data from sources no supported by other agents.
- Application Agents

  These collect data from databases, WebSphere Application Server, WebSphere MQ, and so forth.

Tivoli Enterprise Portal Server

TEPS functions as a repository for all user data, such as the user IDs and user access control for the monitoring data. This means the data each user will be able to access and how it is displayed. The TEPS connects to the hub Tivoli Enterprise Monitoring Server and provides a consistent look and feel for the users.

Tivoli Enterprise Portal

TEP is the consolidated user interface for IBM Tivoli Monitoring and is used to connect to the TEPS. The TEP can be launched from a browser or can be installed as a client application on a workstation.

In Figure 9-5 we show a typical view of TEP as a client.

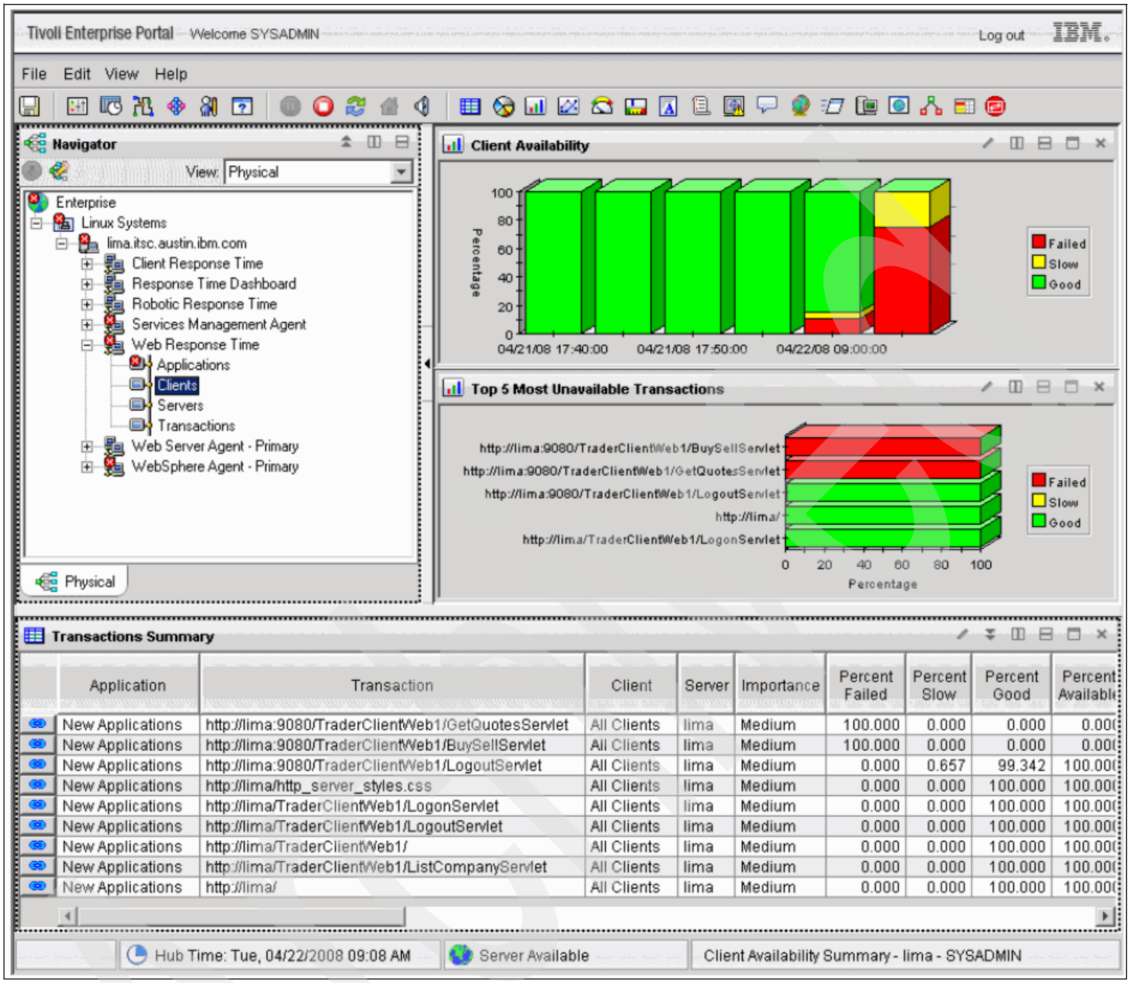


Figure 9-5 The Tivoli Enterprise Portal client

ITCAM for WebSphere integrates into the TEPS.

The IBM Tivoli Monitoring infocenter can be found at the following Web page:

http://publib.boulder.ibm.com/infocenter/tivhelp/v3r1/index.jsp

9.3 Other useful monitoring tools

This section introduces other useful tools for monitoring your BPM infrastructure. It discusses the following:

- "Service Integration Bus Explorer"
- "Service Integration Bus Performance Tool" on page 311
- "Performance Monitoring Infrastructure" on page 313
- "Diagnostic Tool for Java Garbage Collector" on page 314

9.3.1 Service Integration Bus Explorer

The Service Integration Bus Explorer (SIB Explorer) is a stand-alone tool that allows more natural navigation and monitoring of the messaging components of a service integration bus. This tool can display the resources available on the bus and their states and allow limited management of the bus. It is written in Java and communicates directly with the MBean interfaces on the bus.

Note: The SIB Explorer is available to download from alphaWorks® at the following Web page:

`http://www.alphaworks.ibm.com/tech/sibexplorer`

It requires the Standard Widget Toolkit libraries available from the following Web page:

`http://www.eclipse.org/swt`

Installing SIB Explorer on the WebSphere Process Server

We installed the SIB Explorer on the WebSphere Process Server deployment manager using the SWT toolkit version 3.3.2. Perform the following steps to install the SIB Explorer:

1. Download and extract the SWT libraries from Eclipse into a directory (for instance, /usr/local/swt).
2. Download and extract the SI Bus Explorer client code into a directory (e.g. /usr/local/sib).
3. Edit the file env.sh in /usr/local/sib. The file is self-documented but we set the following values:
   - WAS: /opt/ibm/WebSphere/ProcServer
   - SWTJARS: /usr/local/swt
   - CUR: /usr/local/sib

4. Edit the sibexplorer.sh file. and modify the code in Example 9-1 to appear as shown in Example 9-2. This change will allow us to place this file anywhere in the file system

Example 9-1

```
. ./env.sh
```

Example 9-2

```
. $(dirname $0)/env.sh
```

5. Use the code in Example 9-3 to copy the files env.sh and sibexplorer.sh to /usr/local/bin so they are available to everybody.

Example 9-3

```
cp env.sh sibexplorer.sh /usr/local/bin
```

6. Use the code in Example 9-4 to make sibexplorer.sh executable by everyone.

Example 9-4

```
chmod a+x /usr/local/bin/sibexplorer.sh
```

7. Run the SIB Explorer by entering the **`sibexplorer.sh`** command. You will need to have X windows running to see the output.

   When the tool is first run, the window shown in Figure 9-6 appears. Click **Yes**.

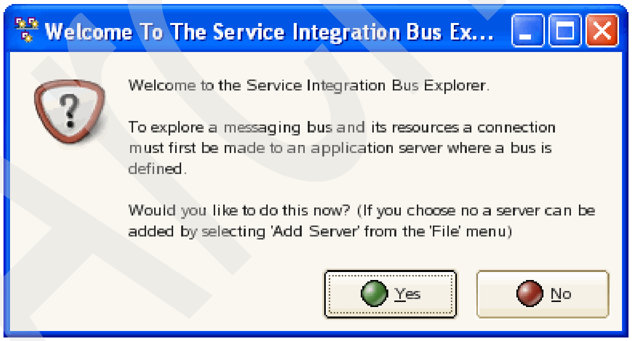


Figure 9-6 The SIB Explorer welcome window

8. In the Server Connection Properties window (Figure 9-7), enter the following values.
   - Host name: localhost
   - Port: 8879

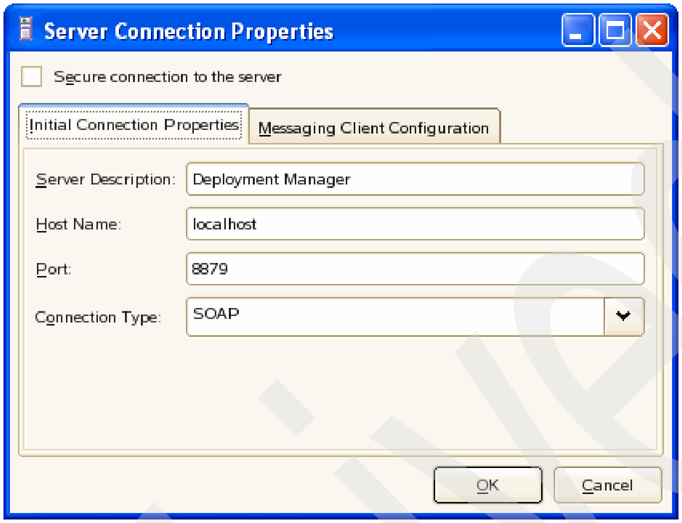


Figure 9-7 The Server Connection Properties window

Click **OK.** The main window will then be displayed.

Opening the new server we have just defined allows us to examine the service integration buses as shown in Figure 9-8 on page 310. Note that in this window we can see the current depth of the queues.

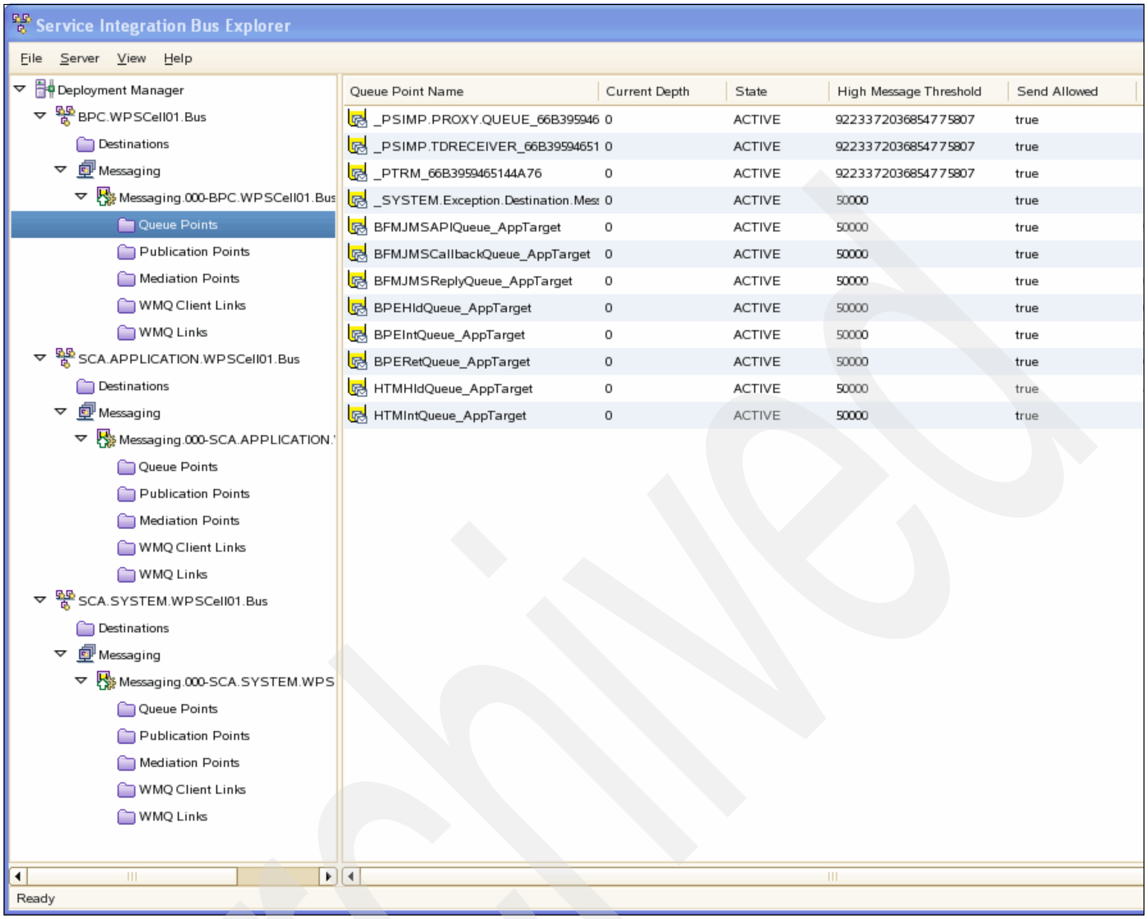


Figure 9-8 The main SIB Explorer window

9. [Optional] If the security of the monitored application servers and messaging engines are enabled, you have to configure SIB Explorer security tabs. The security settings are set in the Server Connection dialog box. In order to set the required security configuration, complete the following steps:

   a. Right-click a server in the left pane and select **Properties**.

   b. Check the **Secure connection** option to the server at the top of the dialog box, and select the **Admin Credentials** tab.

   c. Enter the user name and password for the user who has administrative permissions.

   d. Select the **SSL Options** tab. Enter the location of **SSL key store** and **SSL trust store** along with their corresponding passwords.

e. If security is enabled with the default settings, the server uses the following files:

- DummyServerKeyFile.jks
- DummyServerTrustFile.jks
- DummyClientKeyFile.jks
- DummyClientTrustFile.jks

These files are located in Installation_Directory/profiles/<profile name>/etc.

> **Note:** It may be necessary to copy these files to the machine running the SIB Explorer. The password for these dummy files is WebAS. It is recommended to generate their new key files to ensure that servers is completely secure. In this case, use the corresponding new key files.

9.3.2 Service Integration Bus Performance Tool

The Service Integration Bus Performance Tool is a stand-alone tool that is designed to provide detailed information about messaging components performance and key statistics. Running the tool with default settings provides the following information (automatically updated every two seconds):

- Current message production and consumption rate for every queue and topic on the system (including temporary queues)
- Rate of message production and consumption broken down by reliability level for each queue and topic
- Number of producers/publishers pairs and consumers/subscribers pairs which are attached to each queue or topic
- Number of messages for each queue
- Number of threads in each thread pool (for example: MDBs, ORB, Web container)
- Number of bytes written and received by each application server and message engines
- Detailed information about Java heap size and current percentage of free heap.

> **Note:** The Service Integration Bus performance tool is available to download from alphaWorks at the following Web site:
>
> http://www.alphaworks.ibm.com/tech/sibperf/download

Installing the Service Integration Bus Performance Tool

We installed the Service Integration Bus Performance Tool on the deployment manager using the SWT toolkit version 3.3.2. Perform the following steps to install the Service Integration Bus Performance Tool.

1. Download and extract the SWT libraries from Eclipse into a directory (for example, /usr/local/swt).
2. Download and extract the Service Integration Bus Performance client code into a directory (e.g. /usr/local/sib).
3. Edit the file env.sh in /usr/local/sib. The file is self-documented but we set the following values:
   - WAS_HOME: /opt/ibm/WebSphere/ProcServer
   - SWTJARS: /usr/local/swt
   - SIBPerfDir: /usr/local/sib
4. Edit the sibperf.sh file. and modify the code in Example 9-5 to appear as shown in Example 9-6. This change will allow us to place this file anywhere in the file system.

   Example 9-5

   ```
   . ./env.sh
   ```

   Example 9-6

   ```
   . $(dirname $0)/env.sh
   ```

5. Use the code in Example 9-7 to copy the files env.sh and sibperf.sh to /usr/local/bin so they are available to everybody.

   Example 9-7

   ```
   cp env.sh sibperf.sh /usr/local/bin
   ```

6. Use the code in Example 9-8 to make sibperf.sh executable by everyone.

   Example 9-8

   ```
   chmod a+x /usr/local/bin/sibperf.sh
   ```

7. In the Integrated Solutions Console, perform the following steps to enable Performance Monitoring Infrastructure (PMI) statistics:

   a. Navigate to **Monitoring and Tuning** → **Performance Monitoring Infrastructure** → **server**.

   b. Select the **Enable Performance Monitoring Infrastructure** check box.

   c. Click **Apply** and **Save**.

   d. Restart the WebSphere Process Server.

   > **Note:** If the check box is already selected, no action is required.

8. You can now run the tool by entering the `sibperf.sh` command. You will need to have X Windows running to see the output.

9.3.3 Performance Monitoring Infrastructure

Performance Monitoring Infrastructure (PMI) is the core monitoring infrastructure for WebSphere Application Server and the WebSphere family products. It is a component of the WebSphere Application Server product. The performance data provided by WebSphere PMI helps to monitor and tune the application server performance.

PMI provides a comprehensive set of data that explains the runtime and application resource behavior. For example, PMI provides database connection pool size, servlet response time, Enterprise JavaBeans (EJB) method response time, Java Virtual Machine (JVM) garbage collection time, and CPU usage

Using PMI data, the performance bottlenecks in the application servers can be identified and fixed. For instance, one of the PMI statistics in the Java DataBase Connectivity (JDBC) connection pool is the number of statements discarded from prepared statement cache. This statistic can be used to adjust the prepared statement cache size to minimize the discards and to improve the database query performance. PMI data can be monitored and analyzed by Tivoli Performance Viewer (TPV) and other tools. TPV is a graphical viewer for PMI data that ships with WebSphere Application Server.

Enabling PMI

Perform the following steps to enable PMI from the WebSphere Process Server Integrated Solutions Console.

1. Navigate to **Servers** → **Application Servers**.
2. Click the application server in which you need to enable PMI.
3. Click the Configuration tab.
4. Click **Performance Monitoring Infrastructure (PMI)** under the Performance heading.
5. Select the Enable Performance Monitoring Infrastructure (PMI) check box.
6. (Optional) Choose a statistic set that needs to be monitored under Currently Monitored Statistic Set.
7. Click **OK**.
8. Click **Save**.
9. Restart the application serve for the changes to take effect.

> **Note:** For more information about the PMI tool, go to the WebSphere Application Server Information Center, at the following Web site:
>
> `http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.express.iseries.doc/info/iseriesexp/ae/cprf_pmidata.html`

9.3.4 Diagnostic Tool for Java Garbage Collector

The Diagnostic Tool for Java Garbage Collector examines the characteristics of the garbage collection for an application running under an IBM JVM. It reads from the output of the verbose garbage collection, and produces different kinds of textual and graphical visualizations of garbage collection status.

The tool result data could be stored in a file containing the garbage collection informations using the verbosegc switch as argument for the JVM. The Diagnostic Tool for Java Garbage Collector is specifically designed for looking at the garbage collector activity of a heavily-accessed enterprise application hosted on an application server.

An analysis of data reflecting the activity of the garbage collector in Java enterprise or stand-alone applications is critical to optimizing tasks running under a JVM.

For example, the following issues must be considered in optimization of parameters for Java applications and prevention of bottlenecks:

- Frequency of the garbage collection cycle
- Time spent in different phases of the garbage collection
- Quantities of heap memory involved in the process
- Characteristics of allocation failure characteristics, from which the garbage collection originate
- Unwanted presence of stack overflows

> **Note:** The Diagnostic Tool for Java Garbage Collector is available to download from alphaWorks at the following Web site:
>
> `http://www.alphaworks.ibm.com/tech/gcdiag/download`

Installing Diagnostic Tool for Java Garbage Collector

To install the Diagnostic Tool for Java Garbage Collector, perform the following steps:

1. Download the following libraries:
   - JfreeChart-1.0.0-rc1.jar
   - jcommon-1.0.0-rc1.jar

   > **Note:** Download the two libraries from SourceForge.net (packaged in one file: jfreechart-1.0.0-rc1.zip) then extract the two JAR files.

2. Extract the contents of GCCollector.zip in a directory.
3. Place jfreeChart-1.0.0-rc1.jar and jcommon-1.0.0-rc1.jar in the lib directory that was created when you extracted GCCollector.zip.
4. To run the Diagnostic Tool for Java Garbage Collector, execute the following command (set the current directory as the GCCollector installation folder).

   ```
   javaw -Xmx300m -classpath
   lib/jfreechart-1.0.0-rc1.jar;lib/jcommon-1.0.0-rc1.jar -jar
   lib/GCCollector.jar
   ```

Part 3

Extending production topologies

10

Incorporating WebSphere Business Services Fabric into a production topology

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.

In addition, this chapter discusses how to enable the logging of events in WebSphere Business Services Fabric for consumption by monitoring applications such as WebSphere Business Monitor.

This chapter contains the following sections:

10.1 Introduction

WebSphere Business Services Fabric is a comprehensive SOA based offering to deliver dynamic Service Oriented Applications that leverage existing IT assets and deliver business value incrementally. For detailed information about WebSphere Business Services Fabric concepts, architecture, and the development of dynamic SOA applications, refer to the Redbooks publication *Getting Started with IBM WebSphere Business Services Fabric V6.1*, SG24-7614.

This chapter focuses on incorporating WebSphere Business Services Fabric into an existing WebSphere Process Server production topology. This chapter provides step-by-step instructions to 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, refer to Chapter 5, “Configuring a Remote Messaging and Remote Support topology” on page 89.

Figure 10-1 on page 321 shows the Remote Messaging and Remote Support topology pattern, and shows where WebSphere Business Services Fabric components are added to it.

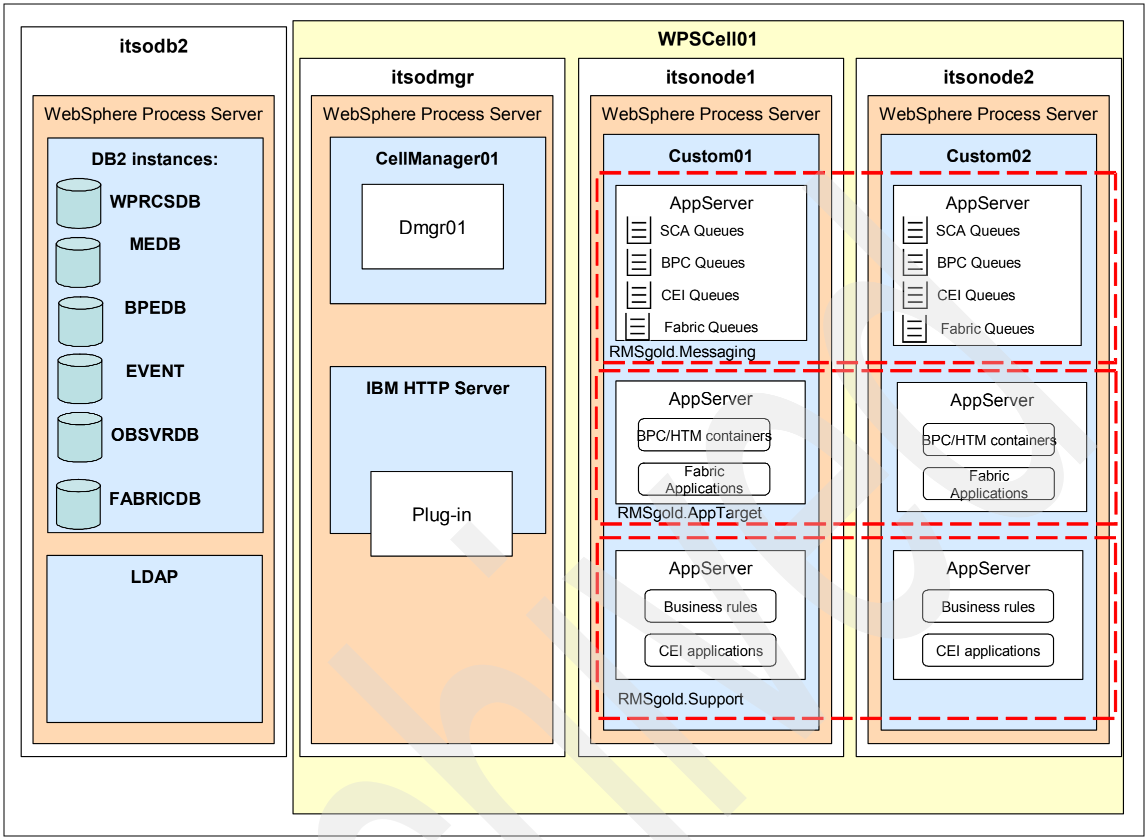


Figure 10-1 Remote Messaging and Remote Support Topology with WebSphere Business Services Fabric

The following WebSphere Business Services Fabric related components are added to the Remote Messaging and Remote Support topology pattern:

- A DB2 database, FABRICDB, is added to the DB2 server.
- A service integration bus, named fabricbus, is added to the messaging cluster (RMSgold.Messaging in our example).
- The WebSphere Business Services Fabric core application EAR files are added to the application cluster (RMSgold.AppTarget in our example).
- WebSphere Business Services Fabric events are emitted to the JMS destinations present in the support cluster (RMSgold.Support in our example).

10.2 Installing Fabric in a clustered environment

At the time of writing, there is no interactive installer to install WebSphere Business Services Fabric in a clustered environment. WebSphere Business Services Fabric needs to be installed and configured manually in the clustered environment. This section describes the steps necessary to install WebSphere Business Services Fabric into a clustered environment. The process to install WebSphere Business Services Fabric in a clustered environment has been broken down into the following two separate procedures:

- “Unloading the Fabric Foundation Pack” on page 322
- “Copying the Fabric artifacts” on page 324

10.2.1 Software versions

The following software and operating systems are used in this chapter:

- SUSE Linux Enterprise Server 10 service pack 1

> **Note:** WebSphere Business Services Fabric V6.1.2 supports SUSE Linux Enterprise Server on the IBM Power family of processors only (not Intel® processors). For a full list of supported platforms, refer to the following Web page:
>
> http://www-01.ibm.com/support/docview.wss?rs=36&uid=swg27012795

- WebSphere Process Server V6.1.2
- WebSphere Business Services Fabric V6.1.2
- IBM DB2 Universal Database V9.1
- IBM Tivoli Directory Server V6.0

10.2.2 Unloading the Fabric Foundation Pack

The first part of the process to install WebSphere Business Services Fabric requires unloading the Fabric Foundation Pack. This section describes the steps to unload the WebSphere Business Services Fabric product binary.

1. Login to the deployment manager machine as a root user.
2. Run the **unzip <fabricbinary.zip>** command to extract the WebSphere Business Services Fabric binaries, where fabricbinary.zip is a WebSphere Business Services Fabric V6.1.2 installable binary.
3. Go to installers directory and run the **./install_fabric_lnx** command.

4. The Installation Wizard (Figure 10-2) will launch. Select **English** and click **OK**.

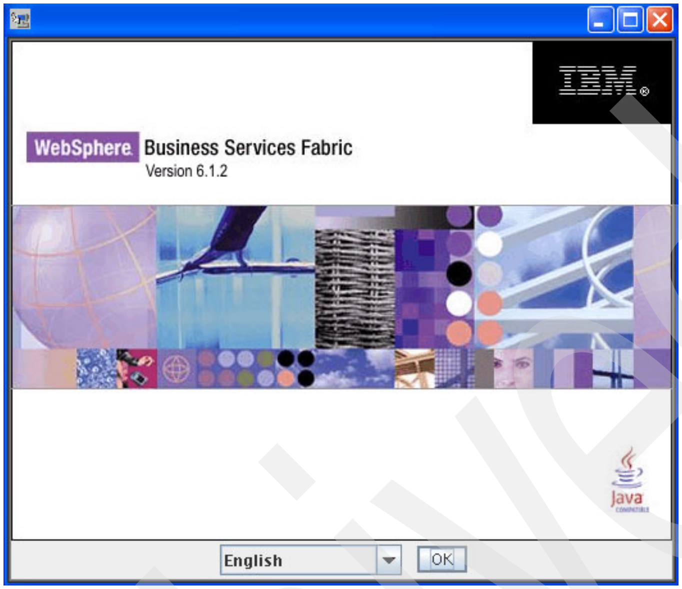


Figure 10-2 Language Selection window

5. Click **Next** in the Welcome window.
6. In the License Agreement window, select **I accept the terms in license agreement** and click **Next**.
7. In the Install Set text box, select **Files Only** (Figure 10-3). Click **Next**.

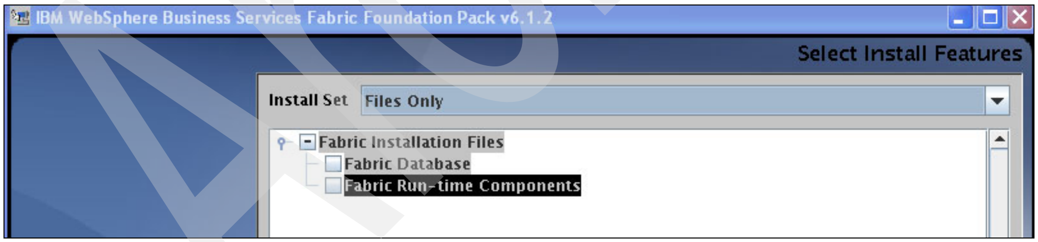


Figure 10-3 Select Install Features window

8. Select the directory (we used /opt/IBM/WebSphere/Fabric/FoundationPack) to install the installation files and click **Next**.
9. In the Pre-Installation Summary window, click **Install**.
10. In the Install Complete window, click **Done**.

10.2.3 Copying the Fabric artifacts

The next part of the process to install WebSphere Business Services Fabric requires copying the Fabric artifacts. To install WebSphere Business Services Fabric manually, several JAR files and product version files must be copied to the WebSphere Process Server nodes from the deployment manager node where we unloaded the WebSphere Business Services Fabric binary. The steps that follow detail how to copy the files to all the WebSphere Process Server nodes.

1. Copy the following Fabric JAR files from <WBSF_HOME>/runtime (/opt/IBM/WebSphere/Fabric/FoundationPack/runtime in our environment) to <WPS_HOME>/lib/ext (/opt/ibm/WebSphere/ProcServer/lib/ext) in machines where WebSphere Process Server is installed, including the deployment manager node.
   - fabric-types.jar
   - fabric-da-scdl.jar
   - fabric-da-sca.jar
   - fabric-da-model.jar
   - fabric-da-api.jar
2. Copy the following product version information files from <WBSF_HOME>/modified/configuration/Runtime (/opt/IBM/WebSphere/Fabric/FoundationPack/modified/configuration/Runtime) to <WPS_HOME>/properties/version (/opt/ibm/WebSphere/ProcServer/properties/version) in machines where WebSphere Process Server is installed:
   - WBSF.product
   - WBSFengine.component

10.3 Creating the Fabric database and schema

WebSphere Business Services Fabric requires one database. It provides the script to create the database with the recommended settings. Perform the following steps to create the database and its tables.

1. Copy the following sql files from <WBSF_HOME>/configuration/Database/DB2/Multiplatforms (/opt/IBM/WebSphere/Fabric/FoundationPack/configuration/Database/DB2/Multiplatforms) to the DB2 machine:
   - create_fabric_db_linux.sql
   - create_fabric_schema.sql
2. Login to the DB2 machine as DB2 administrator (User ID: db2inst1 in our environment).

3. Open the create_fabric_db_linux.sql file and perform the following steps, if necessary:

   a. If the db2 is not installed in the default location, change the database location. In our scenario we have changed the database location to /home/db2inst1(Figure 10-4).

   b. If the table space location is not the default one, change it. In our scenario we have changed the table space location to /home/db2inst1 (Figure 10-4).

> **Note:** Edit the table space location values for all the tablespaces including SYSTEM CATALOG, USER and TEMPORARY.

finst1@db2v91:.../db2scripts_fabric

```
CREATE DATABASE fabricdb ON '/home/db2inst1'
  USING CODESET UTF-8 TERRITORY US COLLATE
  USING
    SYSTEM CATALOG TABLESPACE
      MANAGED BY SYSTEM USING ('/home/db2inst1/db2/fabric/fabric-sys-auto')
      EXTENTSIZE 16 PREFETCHSIZE 64 OVERHEAD 13.17 TRANSFERRATE 0.20
    USER TABLESPACE
      MANAGED BY DATABASE USING(FILE '/home/db2inst1/db2/fabricdb/fabric_USR_DAT
A/fabric-user-4k-c1' 100000)
      EXTENTSIZE 16 PREFETCHSIZE 64 OVERHEAD 13.17 TRANSFERRATE 0.20
    TEMPORARY TABLESPACE
      MANAGED BY SYSTEM USING ('/home/db2inst1/db2/fabric/fabric-temp-auto')
      EXTENTSIZE 16 PREFETCHSIZE 64 OVERHEAD 13.17 TRANSFERRATE 0.20;

CONNECT TO FABRICDB;
```

Figure 10-4 Editing Database and table space location

4. Open the create_fabric_schema.sql file and change the username and password in the connect string. In our scenario the username is db2inst1 and the password is passw0rd (Figure 10-5).

```
CONNECT RESET;


----Updating DB and DBM config

connect to fabricdb user db2inst1 using 'passw0rd' ;

update dbm cfg using DISCOVER_INST disable ;
update dbm cfg using BACKBUFSZ 2048;
update dbm cfg using RESTBUFSZ 2048;
```

Figure 10-5 Editing database username and password

5. Run the following commands:

```
db2 -tvf create_fabric_db_linux.sql
db2 -tvf create_fabric_schema.sql
```

6. Verify the database and schema creation by performing the following steps:
   a. Enter the **`db2`** command to launch the DB2 command prompt.
   b. Enter the **`connect to fabricdb`** command in the DB2 command prompt to connect to the newly created Fabric database.
   c. Enter the **`List Tables for schema db2inst1`** command. It should list 23 tables for the schema db2inst1 in the fabricdb database.

10.4 Configuring WebSphere Process Server cluster resources

This section describes how to configure resources within the cluster scope. It contains the following sections:

- "Setting WebSphere environment variables" on page 327
- "Creating J2C authentication for the Fabric database" on page 328
- "Creating and configuring the data sources" on page 329
- "Creating and configuring the service integration bus" on page 332
- "Creating destinations in the service integration bus" on page 334
- "Configuring the JMS provider" on page 336
- "Configuring the mail provider" on page 338
- "Configuring security" on page 339
- "Configuring distributed cache" on page 340
- "Configuring a namespace variable for CEI" on page 342
- "Installing the Fabric EAR files" on page 342
- "Troubleshooting WebSphere Business Services Fabric installation" on page 345
- "Granting access to the Fabric Tools Console" on page 345

10.4.1 Setting WebSphere environment variables

The data sources of WebSphere Business Services Fabric use the WebSphere environment variable DB2UNIVERSAL_JDBC_DRIVER_PATH to communicate to the fabricdb database in the DB2 database. This section describes the steps to set the WebSphere environment variable.

1. Logon to the deployment manager console using the administrator user (we used wps). The URL in our environment for the deployment manager console is http://itsodmgr:9060/ibm/console. Refer to Section 2.6, "Populating the security registry" on page 50 for the user list and passwords.
2. In the Integrated Solutions Console, navigate to **Environment** → **WebSphere Variables** and in the Scope column, select the cluster **Cluster = RMSgold.AppTarget**.
3. In the Name column, locate DB2UNIVERSAL_JDBC_DRIVER_PATH. Ensure the entry in the value column is the path to the DB2 client JDBC JAR files. See Figure 10-6. The value is set while creating the WebSphere Process Server clusters. If the value is not there, set the value to DB2 client JDBC JAR file path and save the changes.

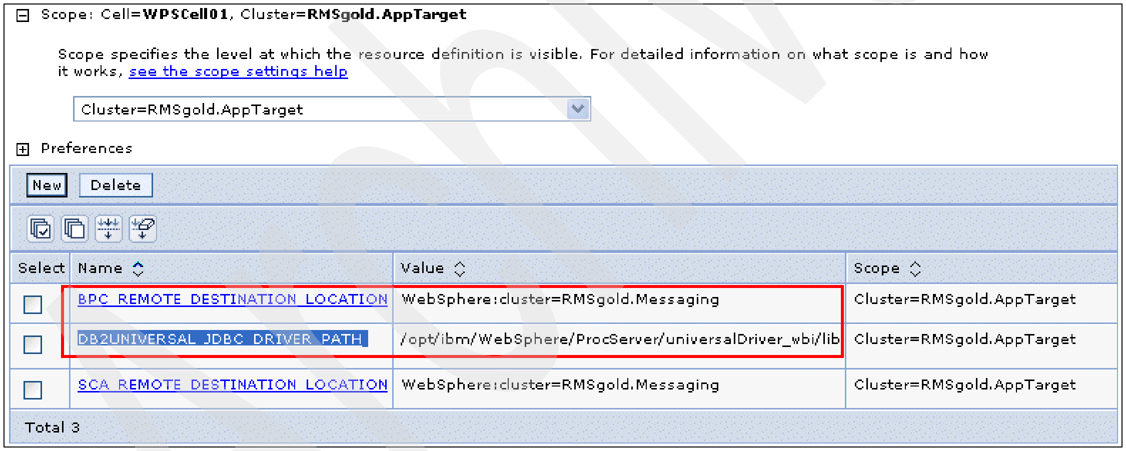


Figure 10-6 WebSphere Variables Cluster scope window

4. Repeat the above steps for clusters RMSgold.Messaging and RMSgold.Support.

> **Note:** The DB2 client JDBC jar files db2jcc.jar, db2jcc_license_cu.jar must be available in the path mentioned in the DB2UNIVERSAL_JDBC_DRIVER_PATH in all the WebSphere Process Server nodes.

10.4.2 Creating J2C authentication for the Fabric database

This procedure details creating the J2C authentication alias used by WebSphere Business Services Fabric data sources to connect the fabricdb database.

1. In the Integrated Solutions Console, navigate to **Security** → **Secure administration, applications, and infrastructure**.
2. Under Authentication, click **Java Authentication and Authorization Service**.
3. Click **J2C authentication data**.
4. Click **New** and enter the following values given in the Table 10-1. Enter the User ID and Password according to your environment.

Table 10-1 J2C authentication data

Field	Values
Alias	Fabric_DB2_Authalias
User ID	db2inst1
Password	passw0rd
Description	Fabric database authentication alias

5. Click **OK** in the Authentication alias creation window (Figure 10-7).

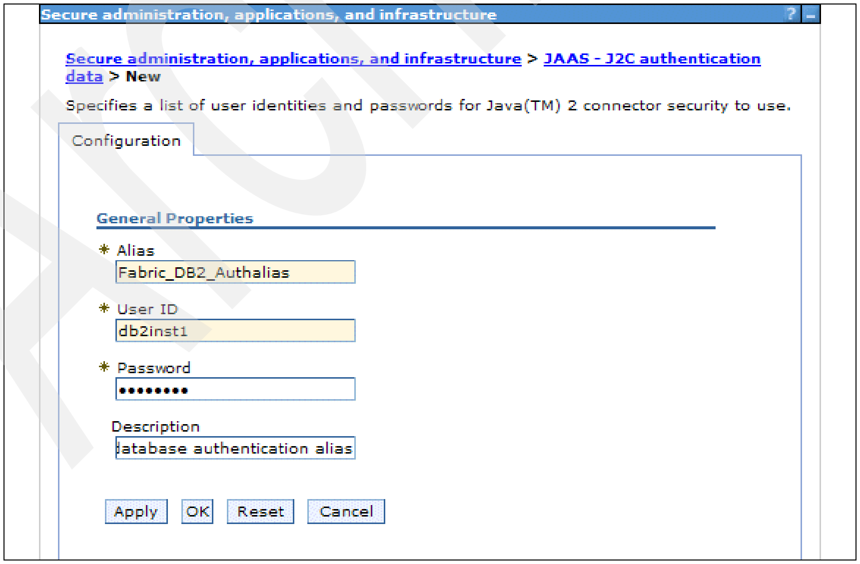


Figure 10-7 Fabric database Authentication details

6. Click **Save** to save in the master configuration.
7. Click **OK** after the nodes are synchronized.
8. Navigate to **System administration** → **Console Preferences**. Ensure the **Synchronize changes with nodes** option is selected. This ensures the changes in the deployment manager are synchronized across the nodes.

10.4.3 Creating and configuring the data sources

Data sources are resources that include information about how to connect to databases. WebSphere Business Services Fabric requires three data sources for the fabricdb and one data source for the MEDB. This section describes the steps to create the required data sources.

1. In the Integrated Solutions Console, navigate to **Resources** → **JDBC** and click **Data Sources**. In the Scope column, select the cluster **Cluster= RMSgold.AppTarget**.
2. Click **New**.
3. Enter fabric bsr in the Data Source name text box.
4. Enter jdbc/fabric/bsr in the JNDI name text box.
5. Select **CellManager01/Fabric_DB2_Authalias** in the Component-managed authentication alias and XA recovery authentication alias drop-down menu (Figure 10-8). Click **Next**.

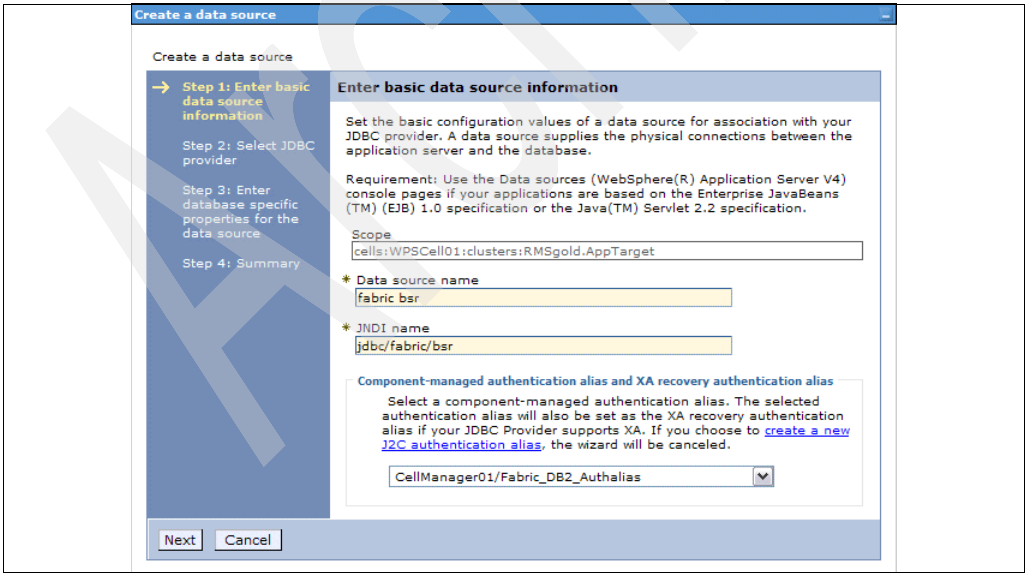


Figure 10-8 Fabric Data source creation Step 1

6. Select the Select an existing JDBC provider radio box. Select **DB2 Universal JDBC Driver provider (XA)** in the drop-down menu as shown in Figure 10-9 and click **Next**.

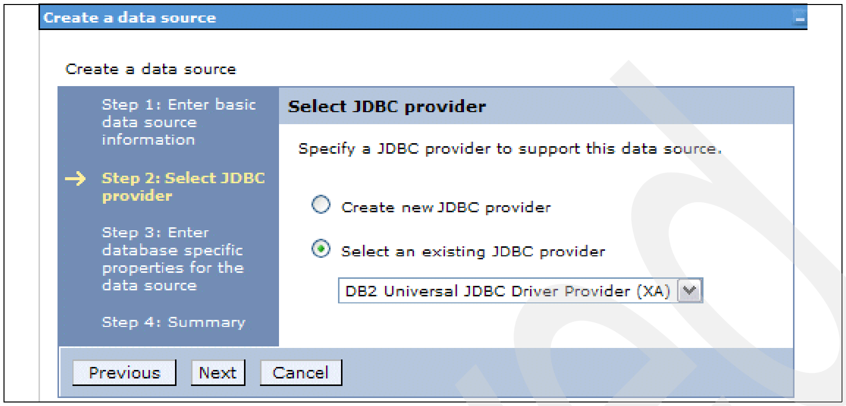


Figure 10-9 Fabric Data source creation Step 2

7. Specify the following information as shown in Figure 10-10.
   – Database name: fabricdb
   – Driver type: 4
   – Server name: db29v
   – Port number: 50000

   Clear the Use this data source in container managed persistence (CMP) check box and click **Next**.

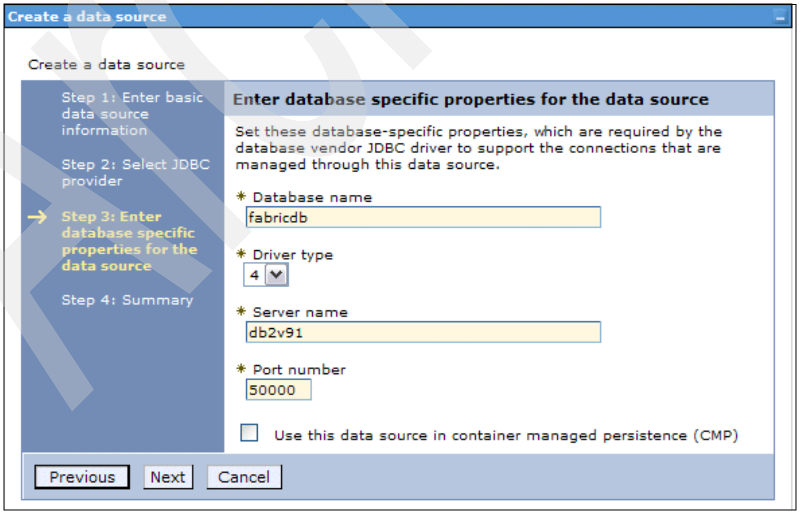


Figure 10-10 Fabric Data source creation Step 3

8. Click **Finish** and click **Save**.
9. Click **OK** after the nodes are synchronized.
10. Create additional data sources using the values specified in Table 10-2. For all the data sources, use the following values:
    – Server name: db29v1
    – Port number: 50000

Table 10-2 Data Sources

Scope	Data source name	JNDI name	Database name	Usage
RMSgold.AppTarget	fabric gm	jdbc/fabric/gm	FABRICDB	Used by Fabric governance manager
RMSgold.AppTarge	fabric pm	jdbc/fabric/pm	FABRICDB	Used by Fabric performance manager
RMSgold.Messaging	fabric me	jdbc/fabric/me	MEDB	Used by Fabric messaging engine

11. Navigate to **Resources** → **JDBC** and click **Data Sources**. In the Scope column, select **All scopes** and select all four of the Fabric data sources you have created. Click **Test connection** and note the status of the test. A successful connection to all the data sources is shown in Figure 10-11. If your test is not successful, validate the values you entered for each of the data sources.

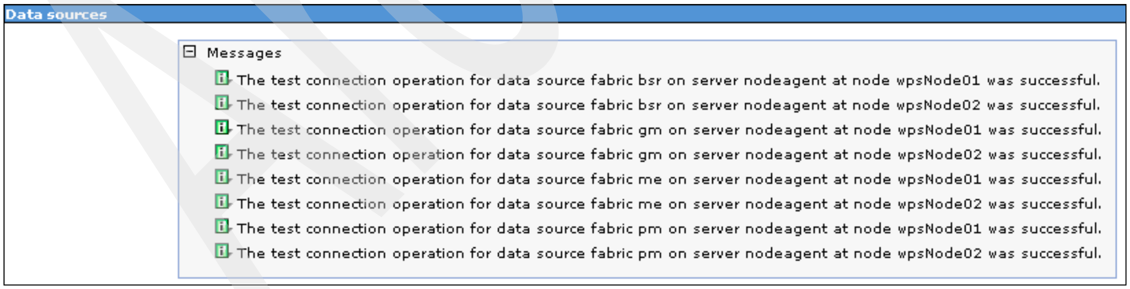


Figure 10-11 Test Connection Message display

10.4.4 Creating and configuring the service integration bus

A service integration bus supports applications using message-based and service-oriented architectures. A bus is a group of one or more interconnected servers that are members of the bus. Applications connect to a bus at one of the messaging engines associated with its bus members. WebSphere Business Services Fabric uses a service integration bus called fabricbus.This section provides the steps to create and configure the fabricbus.

1. In the Integration Solutions Console, navigate to **Service integration** → **Buses** and click **New**.
2. Specify the name of the new bus as fabricbus (Figure 10-12).

Figure 10-12 Fabric bus creation Step 1

3. Make sure the Bus Security check box is not selected and click **Next**.
4. Click **Finish** and click **Save to the master configuration**.
5. Click **OK** after the synchronization of the nodes.
6. Navigate to **Service integration** → **Buses** → **fabricbus** and click **Bus members**.
7. Click **Add** in the Bus Members window.

8. Select RMSgold.Messaging in the Cluster drop-down menu, as shown in Figure 10-13. Click **Next**.

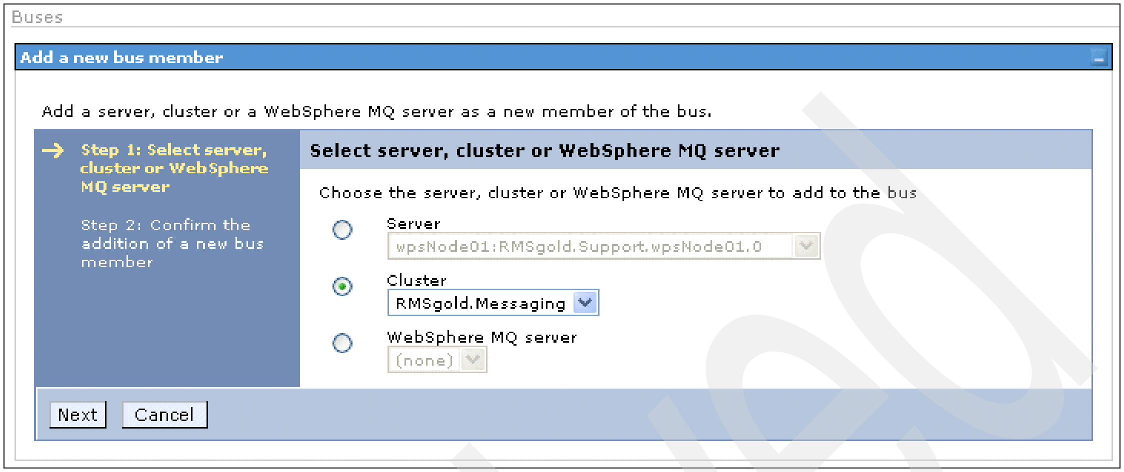


Figure 10-13 Adding Messaging cluster as a Bus Member

9. Select the Data store radio button as shown in Figure 10-14. Click **Next**.

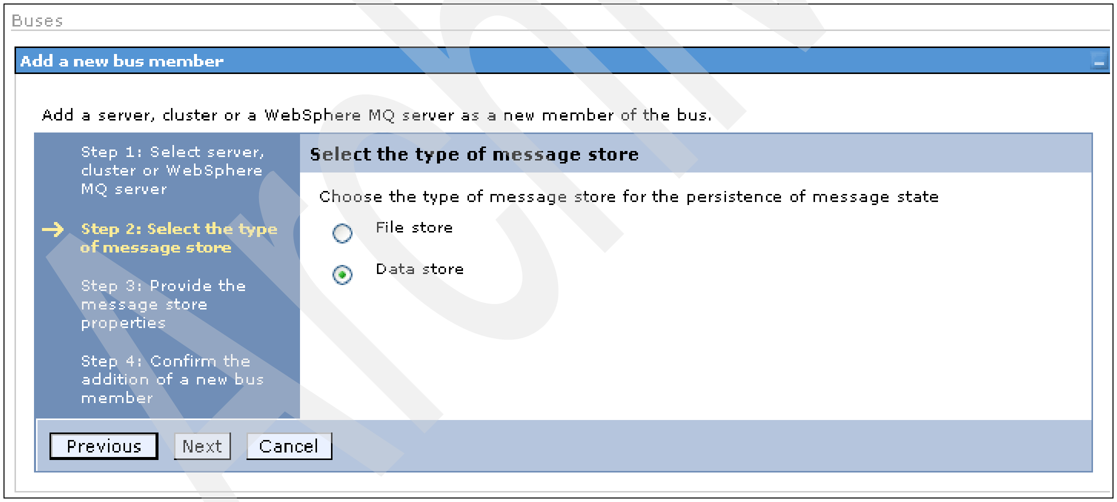


Figure 10-14 Choosing Data store for the persistence of message state

10. Specify the following message store properties as shown in Figure 10-15

   – Data source JNDI name: jdbc/fabric/me.

     You defined this data source in the previous step 10.4.3, “Creating and configuring the data sources” on page 329.

– Schema name: IBMWSSIB
– Authentication alias: CellManager01/Fabric_DB2_Authalias

Ensure the Create tables check box is selected. Click **Next**.

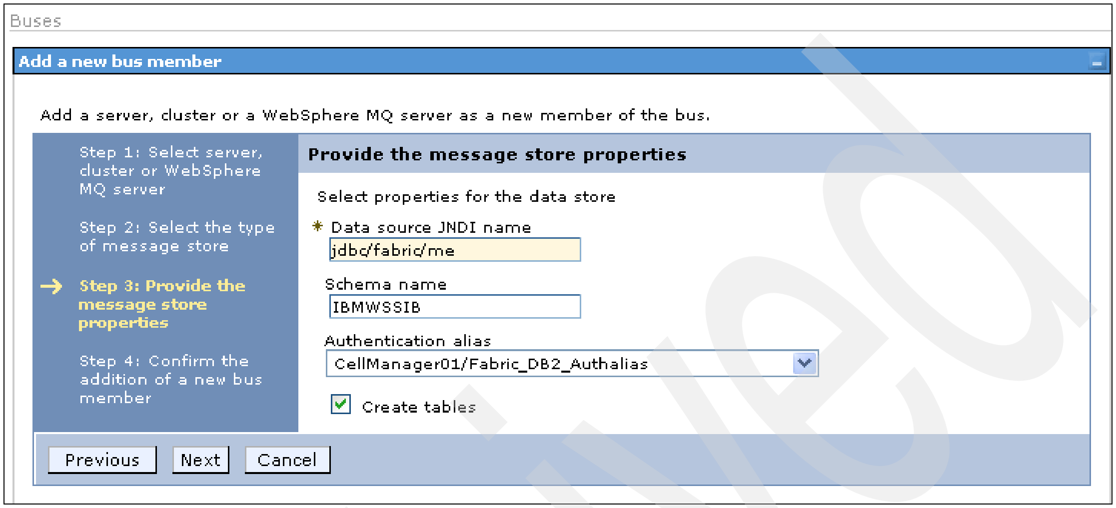


Figure 10-15 Providing message store properties

11. Click **Finish** and click **Save to the master configuration**.
12. Click **OK** after the synchronization of the nodes.

10.4.5 Creating destinations in the service integration bus

A bus destination is a virtual location within a service integration bus, to which applications attach as producers, consumers, or both, to exchange messages. This section defines how to create the following resources in the fabricbus service integration bus:

- “Creating queues”
- “Creating a topic space” on page 335

Creating queues

1. In the Integrated Solutions Console, navigate to **Service integration** → **Buses** → **fabricbus**.
2. Under Destination resources, click **Destinations**. Click **New**. In the Select destination type list, select the Queue radio button as shown in the Figure 10-16 and click **Next**.

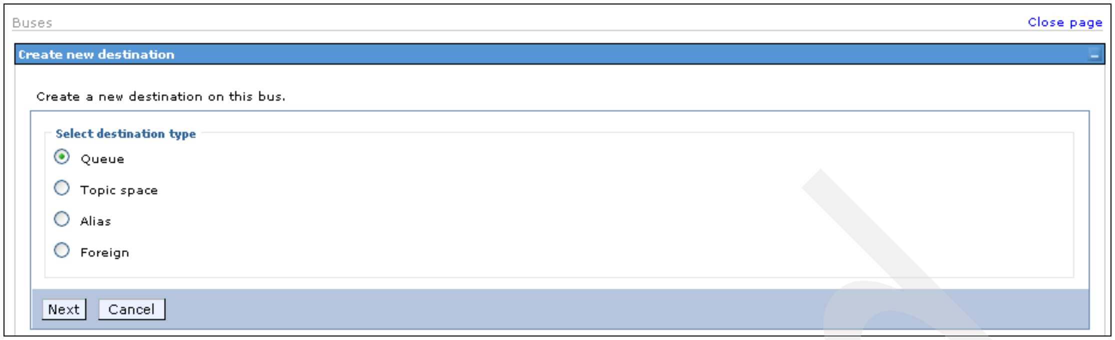


Figure 10-16 Select the Destination type

3. Enter Hub.Request.Queue in the Identifier text box as shown in the Figure 10-17, and click **Next**.

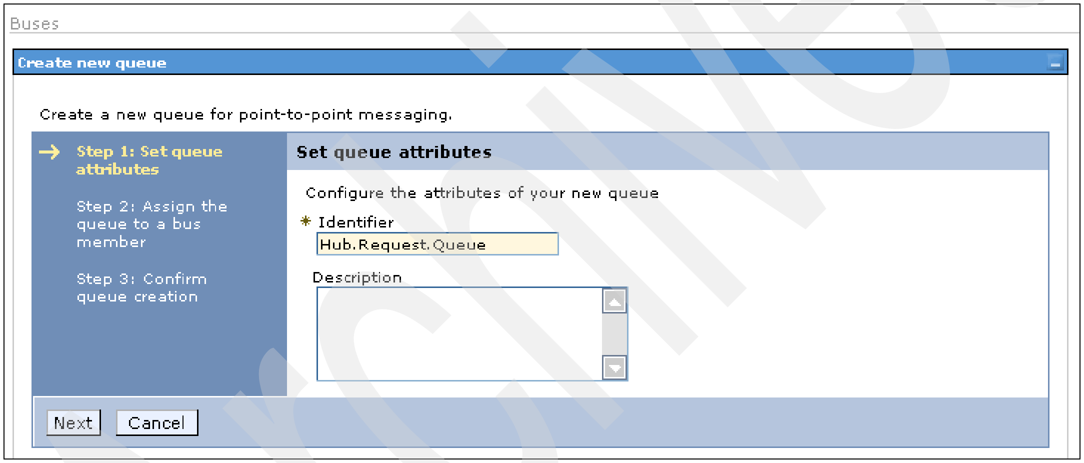


Figure 10-17 Queue creation for the fabric bus

4. Select Bus member as Cluster=RMSgold.Messaging and click **Next**.
5. Click **Finish** and click **Save to the master configuration**.
6. Click **OK** after the synchronization of the nodes.

Creating a topic space

Perform the following steps to create a topic space:

1. Navigate to **Service integration** → **Buses** → **fabricbus**.
2. Under Destination resources click **Destinations**. Click **New**.
3. Select destination type as **Topic Space** and click **Next**.
4. Specify DA.Event.Topic in the Identifier text box, and click **Next**.

5. Click **Finish** and click **Save to the master configuration**.
6. Click **OK** after the synchronization of nodes.

10.4.6 Configuring the JMS provider

A Java Messaging Service (JMS) provider enables messaging based on the Java Messaging Service (JMS). It provides J2EE connection factories to create connections for JMS destinations. This section describes the JMS resources you need to define. It contains the following sections:

- “Creating a connection factory”
- “Creating queues” on page 336
- “Creating topics” on page 337
- “Creating activation specifications” on page 337

Creating a connection factory

Perform the following steps to create a connection factory.

1. In the Integrated Solutions Console, navigate to **Resources** → **JMS** → **JMS Providers**. In the Scope column, select Cluster= RMSgold.AppTarget.
2. Click **Default Messaging Provider**.
3. Click **Connection factories** and click **New**.
4. Specify the following properties in Connection factory creation window:
   - Name: DAEventConnectionFactory
   - JNDI Name: jms/fabric/DAEventConnectionFactory
   - Bus Name: fabricbus
   - Target Type: Bus member name
5. Click **OK** and click **Save to the master configuration**.
6. Click **OK** after the synchronization of nodes.

Creating queues

Perform the following steps to create queues.

1. Navigate to **Resources** → **JMS** → **JMS Providers**. In the Scope column, select Cluster= RMSgold.AppTarget.
2. Click **Default Messaging Provider**.
3. Click **Queues** under Additional Properties and click **New**.
4. Specify the following properties in the Queues creation window:
   - Name: HUBRequestQueue
   - JNDI Name: jms/fabric/HubRequestQueue
   - Bus Name: fabricbus

– Queue Name: Hub.Request.Queue
– Delivery mode: Application

5. Click **OK** and click **Save to the master configuration**
6. Click **OK** after the synchronization of nodes

Creating topics

Perform the following steps to create topics.

1. Navigate to **Resources** → **JMS** → **JMS Providers**. In the Scope column, select Cluster= RMSgold.AppTarget.
2. Click **Default Messaging Provider**.
3. Click **Topics** under Additional Properties, and click **New**.
4. Specify the following properties in Queues creation window:
   – Name: DA Event Topic
   – JNDI Name: jms/fabric/DAEventTopic
   – Bus Name: fabricbus
   – Topic space: DA.Event.Topic
   – Delivery mode: Application
   – Read Ahead: Inherit from connection factory
5. Click **OK** and click **Save to the master configuration**.
6. Click **OK** after the synchronization of nodes.

Creating activation specifications

Perform the following steps to create activation specifications.

1. Navigate to **Resources** → **JMS** → **JMS Providers**. In the Scope column, select Cluster= RMSgold.AppTarget.
2. Click **Default Messaging Provider**.
3. Click **Activation specifications** and click **New**.
4. Specify the following properties in Activation specification creation window:
   – Name: DAPerfMon Activation
   – JNDI Name: jms/fabric/DAPerfMonActivation
   – Destination Type: Topic
   – Destination JNDI Name: jms/fabric/DAEventTopic
   – Bus Name: fabricbus
   – Acknowledge mode: Auto acknowledge
   – Subscription durability: Durable
   – Share durable subscription: In Cluster
5. Click **OK** and click **Save to the master configuration**.

6. Click **OK** after the synchronization of nodes.
7. Repeat the above steps to create a second activation specification, specifying the following information in the Activation specification creation window:
   – Name: Hub Request Activation
   – JNDI Name: jms/fabric/HubRequestActivation
   – Destination Type: Queue
   – Destination JNDI Name: jms/fabric/HubRequestQueue
   – Bus Name: fabricbus
   – Acknowledge mode: Auto acknowledge
   – Subscription durability: Durable
   – Share durable subscriptions: In Cluster
8. Repeat the above steps to create a third activation specification, specifying the following information in the Activation specification creation window:
   – Name: Hub Event Activation
   – JNDI Name: jms/fabric/HubEventActivation
   – Destination Type: Topic
   – Destination JNDI Name: jms/fabric/DAEventTopic
   – Bus Name: fabricbus
   – Acknowledge mode: Auto acknowledge
   – Subscription durability: Non Durable
   – Share durable subscriptions: In Cluster
9. Save all changes.

10.4.7 Configuring the mail provider

This section describes the steps to configure mail resources required for WebSphere Business Services Fabric.

1. Navigate to **Resources** → **Mail** → **Mail Providers**. In the Scope column, select **Cluster= RMSgold.AppTarget**.
2. Click **Built-in Mail Provider**.
3. Click **Mail Sessions** under Additional properties and click **New**.
4. Specify the following properties in Mail Session creation window:
   – Name: Fabric Mail
   – JNDI Name: mail/fabric
   – Mail transport host:
   – Mail transport protocol: smtp
   – Mail store user:
   – Mail store password:

Note: Mail transport host, Mail store user ID and Mail store password are according to your email environment setup. You can also configure mail provider later in the setup.

5. Click **OK** and click **Save to the master configuration**.
6. Click **OK** after the synchronization of nodes.

10.4.8 Configuring security

There are many options to secure the WebSphere Business Services Fabric environment including federated repositories, local operating system, and stand alone LDAP. In this book we have configured Tivoli Directory Server using the federated repositories option.

For instructions on configuring security, refer to Section 7.1.3, “Configuring LDAP” on page 181 and Section 7.1.4, “Enabling administrative security with LDAP” on page 185 to complete the security configuration.

Enabling service integration bus security

When messaging security is switched on, all users who connect to a bus must have the required authorization permissions to use the bus resources. The user accessing the bus should have a bus connector role. We have not enabled service integration bus security for fabricbus in this book.

Perform the following steps to enable service integration bus security:

1. Create a new J2C Authentication Alias for a user who is going to have the bus connector role by following the instructions in Section 10.4.2, “Creating J2C authentication for the Fabric database” on page 328.
2. Navigate to **Service Integration** → **Buses** → **fabricbus** and click **Security**. Select **Enable bus security**, select the J2C Authentication Alias for inter-engine authentication alias and the Mediations authentication alias.
3. Click **OK** and save your changes.
4. Navigate to **Service Integration** → **Buses** → **fabricbus** →**Security** and click **Users and groups** in the bus connector role. Click **New**.
5. Choose **User name** and specify the same user you have used for creating the J2C Authentication alias in the step 1
6. Click **OK** and save your changes.
7. Navigate to **Service Integration** → **Buses** → **fabricbus** →**Security** and click **Users and groups** in the bus connector role. Click **New**.

8. Navigate to **Resources** → **JMS** → **JMS Providers**. In the Scope column, select **Cluster= RMSgold.AppTarget**.
9. Click **Default Messaging Provider** and click **Connection factories**.
10. Click **DAEventConnectionFactory**.
11. Select the J2C Authentication alias created in the step1 for the Component-managed authentication alias.
12. Click **OK** and save your changes.
13. Navigate to **Service Integration** → **Buses** → **fabricbus** →**Security** and click **Users and groups** in the bus connector role. Click **New**.
14. Navigate to **Resources** → **JMS** → **JMS Providers**. In the Scope column, select **Cluster= RMSgold.AppTarget**.
15. Click **Default Messaging Provider** and click **Activation specifications**.
16. Click **DAPerfMon Activation**.
17. Select the J2C Authentication alias created in the step1 for the Authentication alias.
18. Click **OK** and save your changes.
19. Navigate to **Service Integration** → **Buses** → **fabricbus** →**Security** and click **Users and groups** in the bus connector role. Click **New**.
20. Repeat steps 15–19 for the following Activation specifications:
    - Hub Event Activation
    - Hub Request Activation

10.4.9 Configuring distributed cache

This section contains the following sections:

- “Creating the replication domain”
- “Creating object cache instances” on page 341

Creating the replication domain

The dynamic cache of WebSphere Business Services Fabric is replicated using Data Replication Service to all the cluster members. Follow the steps below to create the replication domain.

1. In the Integrated Solutions Console, navigate to **Environment** → **Replication domains**, and click **New**.
2. Enter WBSF DA Replication in the Name text box.
3. Enter Entire Domain in the Number of replicas text box (Figure 10-18).

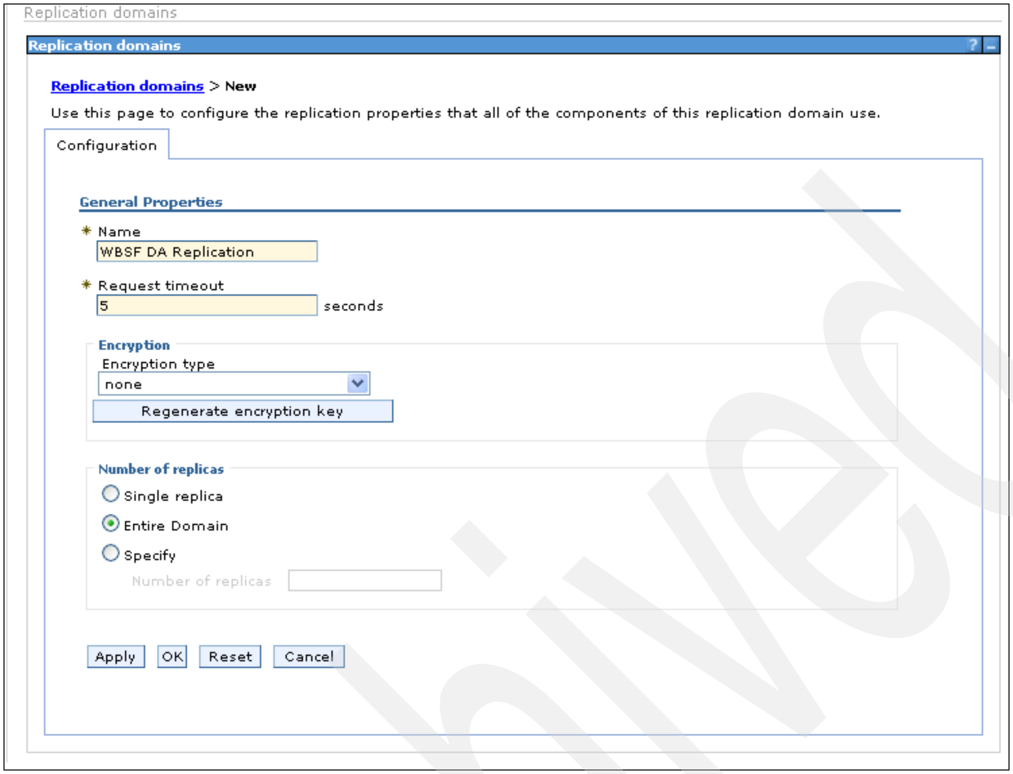


Figure 10-18 Creation of Replication Domain for Fabric

4. Save your changes.

Creating object cache instances

An object cache instance is a location, in addition to the default shared dynamic cache, where J2EE applications can store, distribute, and share data. Follow the steps below to create an object cache instance.

1. Navigate to **Resources** → **Cache Instances** → **Object cache instances**. In the Scope column, select Cluster= RMSgold.AppTarget and click **New**.
2. Specify the following properties in Object cache instances creation window:
   - Name: WBSF Context Cache
   - JNDI Name: services/cache/wbsf.contexts
   - Cache size: 20000
   - Disk Cache settings: Enable disk offload
   - Consistency settings: Enable cache replication
   - Full group Replication domain: WBSF DA Replication
   - Replication type: Both Push and Pull

– Push frequency: 0

3. Save your changes.

10.4.10 Configuring a namespace variable for CEI

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** → **RMSgold.AppTarget**. Expand **Common Event Infrastructure** and click **Common Event Infrastructure Destination**.
2. Note the JNDI Name under Event Infrastructure emitter factory JNDI name. In our scenario it is cell/clusters/RMSgold.Support/com/ibm/events/configuration/emitter/Default
3. Navigate to **Environment** → **Naming**→ **Name Space Bindings**. In the Scope column,, select Cluster= RMSgold.AppTarget and click **New**.
4. Select **String** in the Binding type and click **Next**.
5. Specify the following values:
   – Binding Identifier: wbsf-cbe-emitter-factory
   – Name in name space: wbsf-cbe-emitter-factory
   – String Value: cell/clusters/RMSgold.Support/com/ibm/events/configuration/emitter/Default
6. Click **Finish** and save your changes.

10.4.11 Installing the Fabric EAR files

WebSphere Business Services Fabric is packaged into the following four enterprise application EAR files:

- fabric-engine.ear
- fabric-catalog.ear
- fabric-webtools.ear
- fabric-webtools-help.ear.

Perform the following steps to install these EAR files, using the deployment manager node.

1. In the Integrated Solutions Console, navigate to **Applications** → **Install New Application**.

2. Choose **Remote file system** and click **Browse**.
3. Click **CellManager01** (this is the deployment manager node) as shown in Figure 10-19.

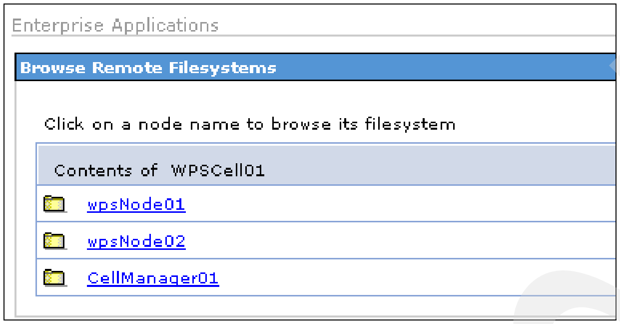


Figure 10-19 Browsing DMGR node for fabric ear files

4. Browse to the /opt/IBM/WebSphere/Fabric/FoundationPack/runtime folder.
5. Select the fabric-catalog.ear radio button as shown in Figure 10-20, and click **OK**.

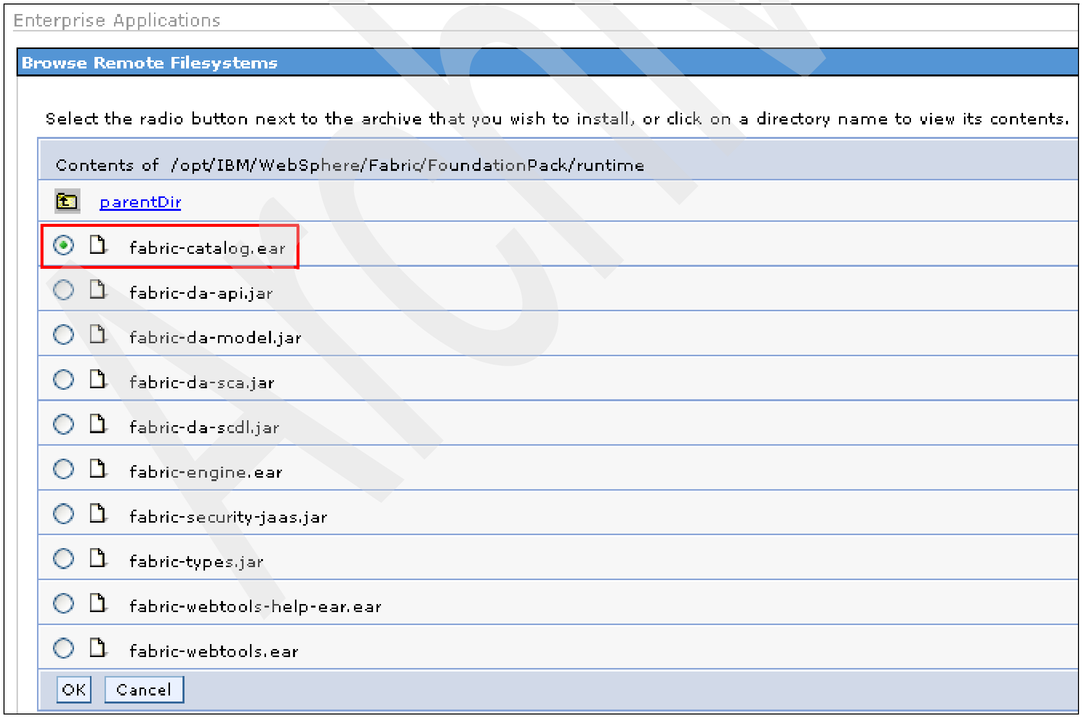


Figure 10-20 Selecting fabric-catalog.ear

6. Click **Next** in the Preparing for the application installation window.
7. Click **Next** in the Install New Application Step 1 window.
8. Select all the modules. In the Clusters and Servers text box (Figure 10-21), select WebSphere:cell=WPSCell01,cluster=RMSgold.AppTarget and click **Apply**.

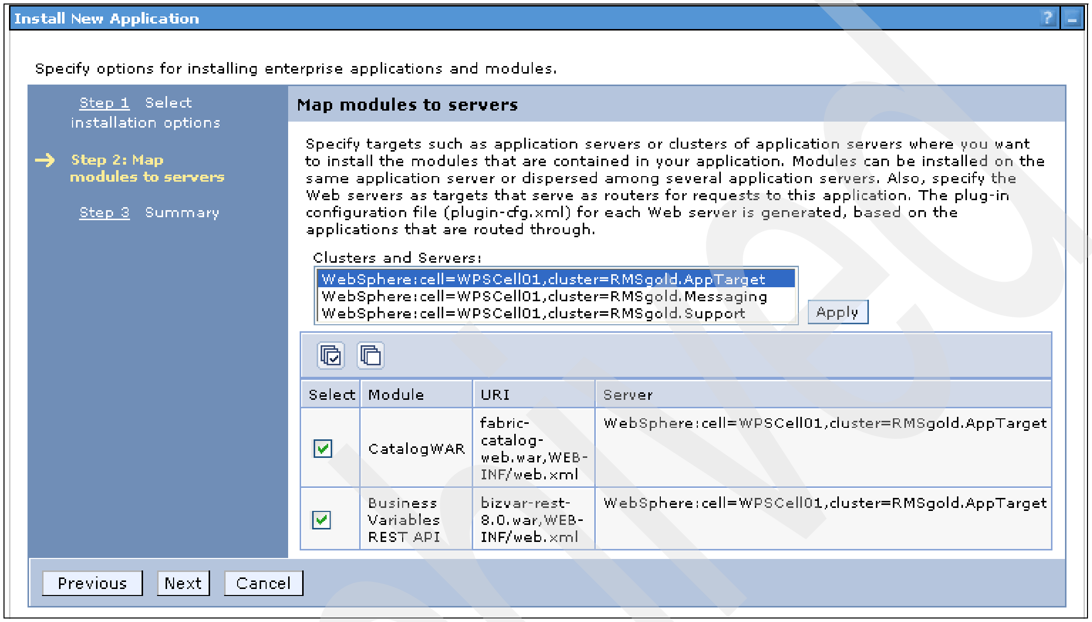


Figure 10-21 Mapping fabric-catalog.ear to RMSgold.AppTarget cluster

9. Click **Next** and click **Finish**.
10. Save your changes.
11. Repeat this procedure to install fabric-engine.ear, fabric-webtools.ear, and fabric-webtools-help.ear.

> **Tips on installing files:**
>
> - In production environment you do not need to install fabric-webtools-help.ear.
> - Make sure all your nodes and servers are running when you are saving the changes and synchronizing the nodes.

12. Navigate to **System Administration** → **Node agents**, select all the nodeagents and click Restart all the Servers on node.

10.4.12 Troubleshooting WebSphere Business Services Fabric installation

To troubleshoot problems in the installation of WebSphere Business Services Fabric, consider the following potential solutions.

- Go to DB2 and verify the creation of 8 tables for the fabricbus messaging engine in the MEDB with the schema name IBMWSSIB.
- Go to the Deployment manager console and verify the scope of the all JMS provider resources created for fabric. It should be RMSgold.AppTarget.
- Verify the JNDI Names are the same for the JMS Provider resources as specified in 10.4.6, “Configuring the JMS provider” on page 336.
- Verify the Destination JNDI Names are given properly for the activation specifications as specified in “Creating activation specifications” on page 337.
- Check the server logs for any java.io.FilePermisssion exceptions for Fabric applications related to Java security. If exceptions present grant the permissions in the policy file or disable Java 2 Security and restart all the servers.

10.4.13 Granting access to the Fabric Tools Console

Only users with the FabricAdminstrator role for the Fabric_Tool can access the Fabric console. In this book we assign the FabricAdminstrator role to the user fabric. Follow the steps below to configure the FabricAdminstrator role.

1. Navigate to **Applications** → **Enterprise Applications** → **Fabric_Tools** and click **Security role to user/group mapping**.
2. Select the **FabricAdminstrator Role** and click **Look up users**.
3. In the Search String enter fabric and click **Search**.
4. Choose user fabric under the Available list and click **>>** to add it to the Selected list as shown in Figure 10-22 on page 346. Click **OK**.

   In the Security role to user/group mapping, you can see that fabric user is mapped to FabricAdministrator.

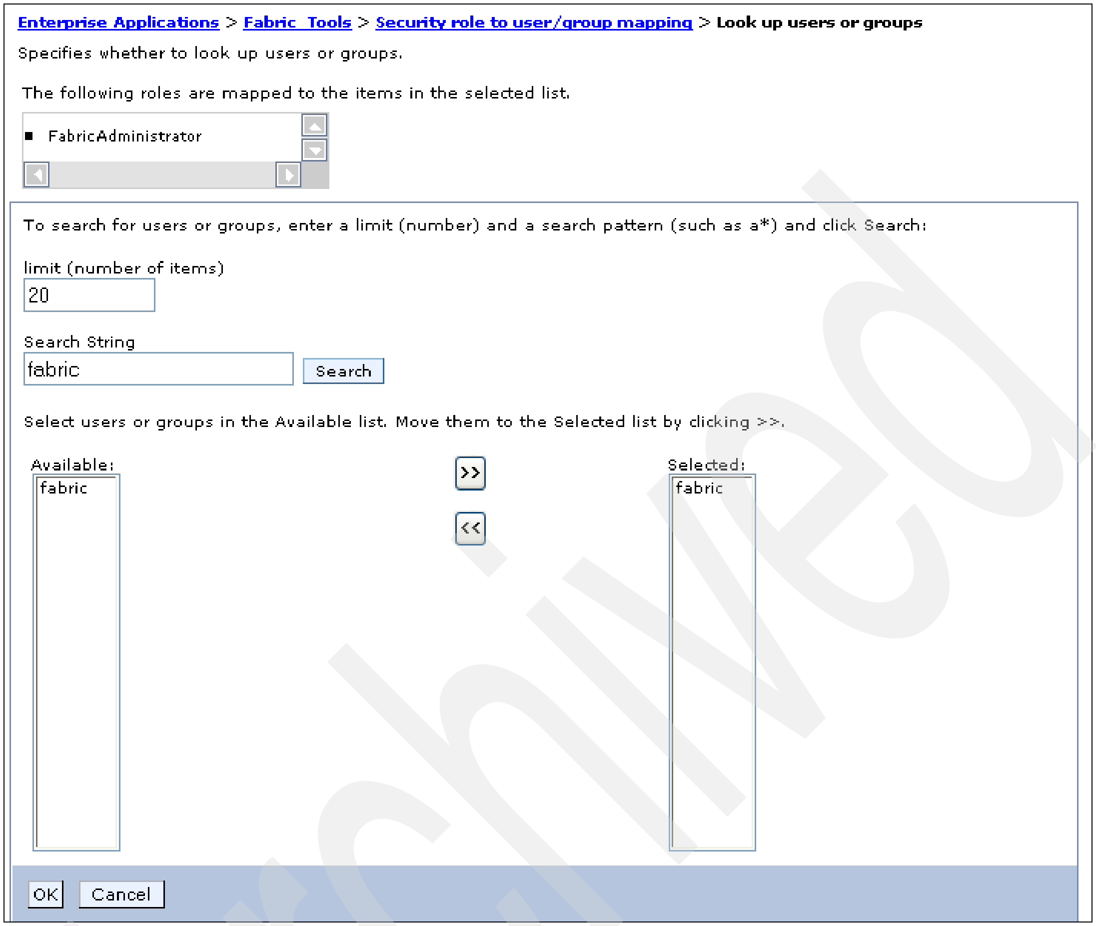


Figure 10-22 FabricAdminstrator Role Mapping

5. Click **OK** and save your changes.
6. Refresh the enterprise applications window to verify whether the Fabric_Tools is started. You can access the Fabric console after it is started.

10.5 Verifying the Fabric installation and configuration

This section describes the steps to verify the installation and configuration of WebSphere Business Services Fabric

1. Login to deployment manager console and verify the Fabric applications are started as shown in Figure 10-23.

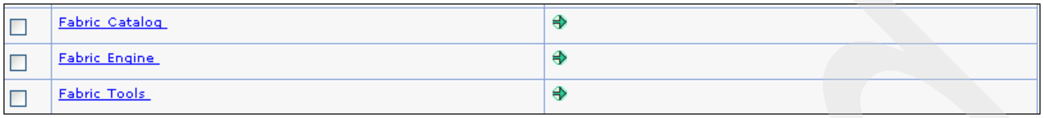


Figure 10-23 Fabric core applications status

> **Note:** We skipped installing the Fabric_Tools_Help EAR. If you have installed the help EAR you should be able to see all four fabric applications running.

2. In the deployment manager console navigate to **Service integration** → **Buses** → **fabricbus** and click **Messaging engines**. The fabricbus messaging engine should be up and running as shown in Figure 10-24.

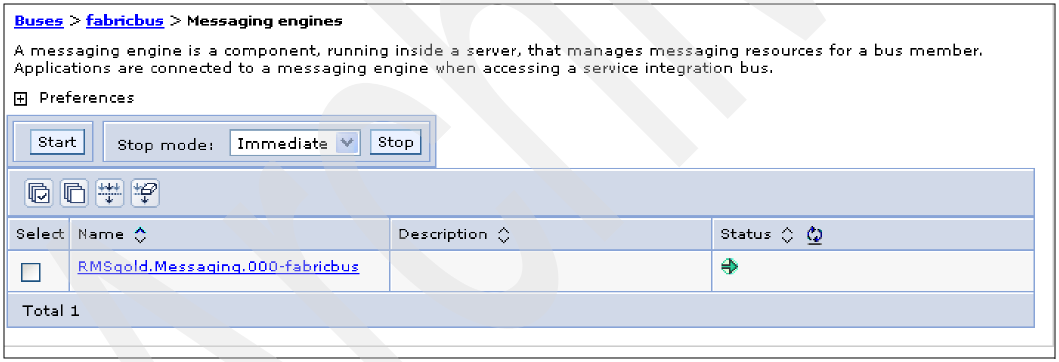


Figure 10-24 fabricbus Messaging Engine Status

3. Type the following url in the browser http://<XXX>:<ZZZ>/fabric, where XXX is the IP or host name of any one of the members of RMSGold.AppTarget Cluster and ZZZ is the defaul-host port of the cluster member, for example http://itsnode1:9080/fabric. You should be able to login using the user fabric and see the window shown in Figure 10-25 on page 348.

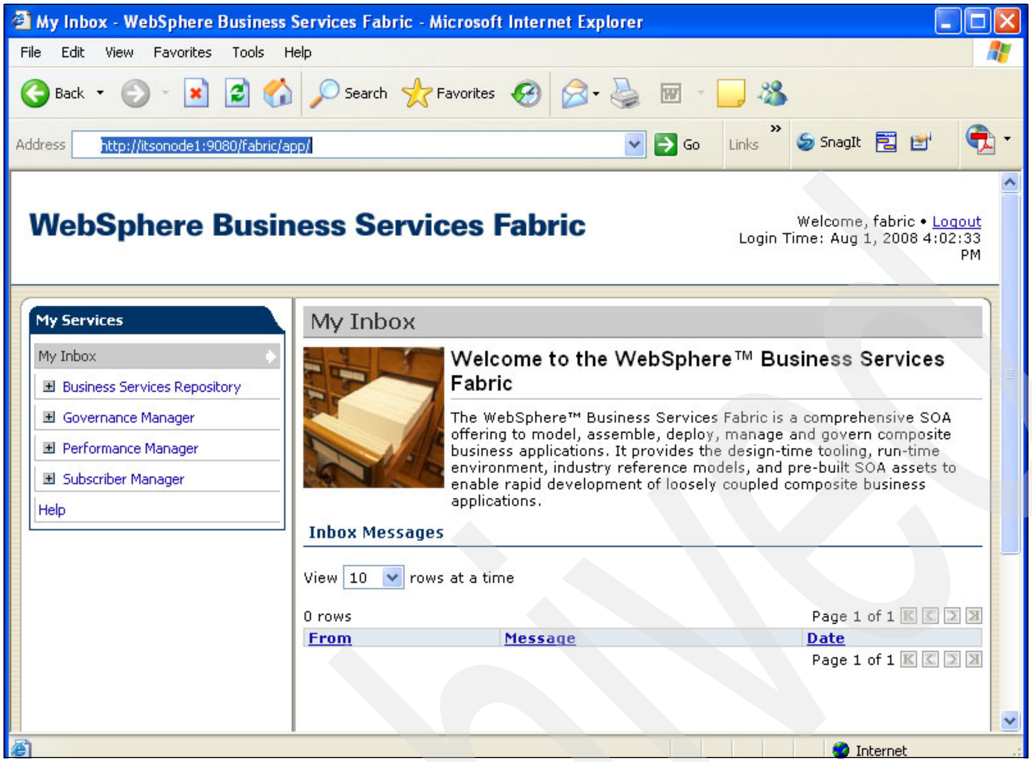


Figure 10-25 WebSphere Business Services Fabric Welcome window

10.6 Installing and testing the sample application

In this section you will test the WebSphere Business Services Fabric environment you have created using the sample enterprise application described in Chapter 4, “Business scenario used in this book” on page 77.

The enterprise application installed in this section is supplied with the additional materials provided with this book. For instructions on how to obtain this additional material, refer to Appendix A, “Additional material” on page 449.

10.6.1 Importing the Fabric Content Pack Archive files

Perform the following steps to import the Fabric Content Pack Archive (FCA) files.

1. Login to the WebSphere Business Services Fabric console by using any one of the RMSGold.AppTarget cluster member URLs with fabric as the user ID.
2. Navigate to **Governance Manager** → **Import/Export**.
3. Click **Browse** and locate the following folder in the additional materials supplied with this book: Scenarios\Fabric\FCA. Choose OrganizationsUsersandRoles20080804-owl.zip and click **Import file**.
4. Repeat these steps to install the following FCA files in this order:
   - FabricGovernance20080804-owl.zip
   - ITSOBankOntPrj20080804-owl
   - ITSOBankCBAPrj20080804-owl.zip

10.6.2 Configuring Enrollments

Perform the following steps to configure enrollments.

1. In the Fabric console, navigate to **Subscriber Manager** → **Manage Subscriber**. Click **ITSOBankOrg** → **Users** → **Grant User Roles**.
2. Select the user fabric from the Available users selection box and press the → button to move it to the Selected users selection box.
3. Select the Administrator role from Role Selection selection box and press the → button to move it to the Selected users selection box.
4. Click **Grant Users Roles**.
5. Navigate to **Subscriber Manager** → **Manage Subscriber**. Click **ITSOBankOrg** → **Enrollments**.
6. Select **ITSOBankLoanAPP** under the Enrollment Selection and click **Save Enrollments**.
7. Navigate to **Subscriber Manager** → **Manage Subscriber**. Click **ITSOBankOrg** → **Users** and click the user ID **fabric**.
8. Click the **Subscriptions** tab.
9. Select **ITSOBankLoanAPP** under Subscription Selection and click **Save Subscriptions**.

10.6.3 Installing EAR Files

Perform the following steps to install EAR files.

1. Login to deployment manager console using user wps.
2. Navigate to **Applications** → **Install New Application**.
3. Choose **Local file system** and click **Browse**.
4. In the additional material, navigate to the folder Scenarios\Fabric\EAR, choose ITSO_implApp.ear and click **Next**.
5. Click **Next** in the Install New Application Step 1 window.
6. Select all the modules. Under Clusters and Servers, select the AppTarget cluster and Itsowebserver and click **Apply**.
7. Select default_host in the Virtual host selection box, as shown in the Figure 10-26.

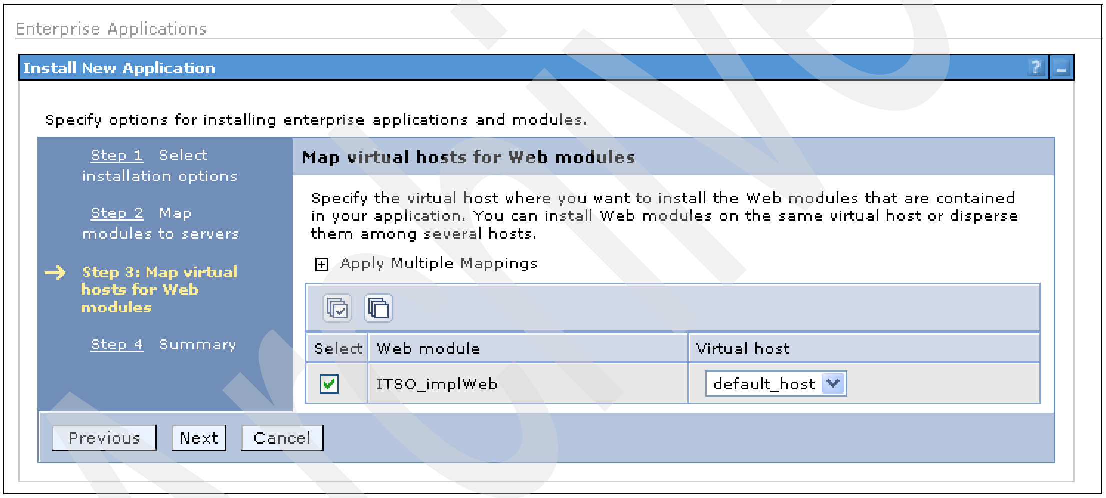


Figure 10-26 Mapping Web module to default_host

8. Click **Next** and click **Finish**.
9. Save your changes.
10. Repeat these steps to install WebSphereEnvUtilApp.ear and ITSOApp.ear from the folder code\Scenarios\Fabric\EAR directory of the additional materials.
11. Navigate to **Applications** → **Enterprise Applications**.
12. Choose **ITSOAPP**, **ITSO_implApp**, and **WebSphereEnvUtilApp** and click **Start**.

10.6.4 Mapping modules to the Web server

As we have a Web server in our topology, we need to map some of the applications to take advantage of the load balancing provided by Web server.

1. In the Integrated Solutions Console, navigate to **Applications** → **Enterprise Applications** → **BPCExplorer_RMSgold.Support** → **Manage Modules**.
2. Select all the modules.
3. Choose your Web server (in our environment, this is itsowebserver) and the cluster where the application is mapped under Clusters and Servers.
4. Click **Apply** and click **OK**.
5. Save your changes.
6. Repeat these steps for the following applications:
   - Fabric_Catalog
   - Fabric_Engine
   - Fabric_Tools
7. Navigate to **Servers** → **Web servers** → **itsowebserver**→ **Generate Plug-in**.
8. Select **itsowebserver** → **Propagate Plug-in**.

10.6.5 Changing SCA Import URLs

In the sample application, all the Web service SCA Imports are bound to URLs that point to the itsodmgr. There is no port explicitly defined in the URLs. Therefore, the default HTTP port of 80 is used. If your environment needs different hosts and ports from the ones hard-coded in the sample application, perform the following steps.

1. In the Integrated Solutions Console, navigate to **Applications** → **SCA Modules** and click **ITSO**.
2. Expand **Imports** → **CreditCheckImport**.
3. Expand **Binding** and click **CreditCheckExport_CreditCheckHttpService**.
4. Change the host name and port number in the Endpoint text box and click **OK**.
5. Repeat these steps for RatingImport, VinLookImport, and VerifyCustomerImport of the SCA module ITSO.
6. Repeat these steps for SubscriberManagerImport present in the module WebSphereEnvUtil.
7. Save your changes.

10.6.6 Changing endpoints URLs in Fabric Composition Studio

In the sample application, all the endpoints defined in WebSphere Business Services Fabric point to itsodmgr, using the default HTTP port of 80. If you need different hosts and ports from the ones hard-coded in the sample application, perform the following steps.

1. Login to the Fabric console with user fabric.
2. Navigate to **Governance Manager** → **Manage Teams** and click the team name **ITSOBankOrg**.
3. Move the fabric user from the Available users column to the Selected users column by pressing the → button.
4. Click **Save**.
5. Open WebSphere Integration Developer in a new workspace.
6. Click **File** → **New** → **Project**.
7. Choose Fabric Project under Business Services Fabric and click **Next**.
8. Specify ITSOCBA in the project name and click **Next**.
9. Click **Configure**, specify the following values. Click **OK**.
   – Host name: One of the host names of the RMSGold.AppTarget cluster
   – Port: Specify the default-host port of the server you specify
   – User: fabric
   – Password: passw0rd
10. Click **Next**, choose ITSOBankCBAPrj and click **Finish**.
11. Click **yes** on Associated Perspective dialog window.
12. Expand **Endpoint** under ITSOCBA in the Business Service Explorer.
13. Click **HighRiskLoanProviderEndPoint** and click Protocol tab.
14. Change the host name and port in the URL text box and click **File** → **Save**.
15. Repeat steps 12 and 13 for the endpoints LowRiskLoanProviderEndpoint, MediumRiskLoanProviderEndpoint, and PremiumLoanProviderEndpoint.
16. Click the Active changes in the Repository Changes Explorer and click **Submit Changelist**.
17. Choose **ITSOCBA** and click **Next**.
18. Move all the changes from the Available changes column to the Selected changes column and click **Finish**.
19. Login to fabric console by using user fabric and click **My Inbox**.
20. Click the latest Change List Submitted.
21. Click **Approve** and click **Publish**.

10.6.7 Testing the sample application

Perform the following steps to test the sample application.

1. Login to Business Process Choreographer console with user fabric. In our environment we used http://itsodmgr/bpc as the URL.
2. Click **My Process Templates**, select TestLoanProcess and click **Start Instance**.
3. Enter the following values:
   – CustomerIdentificationNumber: 100
   – CustomerAddress: <Any string value>
   – VIN: 12345678901234567
   – LoanAmountRequested: 1000
   – BankID: ITSO
4. Click **Submit**. The results should look like Figure 10-27.

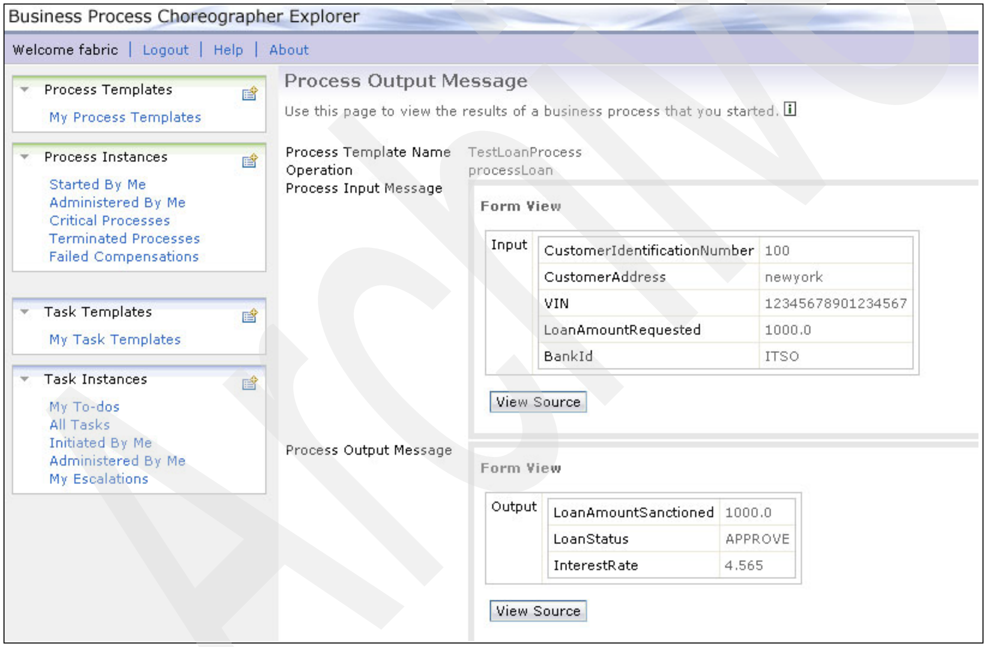


Figure 10-27 Sample application test result

> **Note:** In the sample application, the TestLoanProcess acts as a proxy to the New Loan Process. The TestLoanProcess calls the New Loan Process through an context injector which injects the required context for WebSphere Business Services Fabric.

10.7 Enabling WebSphere Business Services Fabric events

WebSphere Business Services Fabric generates several Common Base Events (CBEs) that can be monitored by WebSphere Business Monitor. By monitoring the Fabric-generated CBEs, custom Key Performance Indicators (KPIs) can be created. KPIs provide useful data on the business decisions that Fabric makes.

Table 10-3 describes the Fabric events that are related to SCA component information, selection policies, context and the success or failure status.

Table 10-3 Events list

Event	Description
Context Extraction Event	This event is fired whenever the Dynamic Assembler processes a context extractor. It captures the current and parent contexts.
Dynamic Selection Event	This event is fired on every successful service invocation. The event captures details about the dynamic selection of an endpoint, such as endpoint ID and address.
Endpoint Not Available Event	This event is fired when the selected endpoint is not available at the time the request is made. For example, the selected endpoint is not available at the specified hours of operation. This event captures information about the resulting error.
No Endpoint For Policy Event	This event is fired when the Dynamic Assembler does not find endpoints that match the criteria available in the policies. This event captures information about the resulting error.
Technical Error Event	This event is fired when a plug-in, such as a Context Extraction or Policy Guard, fails. This event captures information about the resulting error

10.7.1 Enabling events in the sample application

WebSphere Business Services Fabric events can be enabled or disabled through a Fabric policy, wherein each event has an associated policy assertion type. When the assertions are enabled, the Dynamic Assembler fires them for the appropriate event conditions. Dynamic Selection Event has been already added to the LoanProviderPolicy in the sample application.

Adding events to policies

The following steps demonstrate how to add an event to a policy:

1. Open a WebSphere Integration Developer workspace.
2. Import the ITSOFabricPI.zip project interchange file into your workspace. This file is supplied in the \Scenarios\Fabric\PI directory in the additional materials supplied with this book.
3. Click **Window** → **Open Perspective** → **Other**. Choose Business Service in the Open Perspective dialog box and click **OK**.
4. Expand **Policy** and double click **LoanProvidePolicy**.
5. Click the **Contract** tab and click **Add**.
6. Expand **Manageability Assertion** → **CBE Enablement Assertion**.
7. Click **Dynamic Selection Event Enablement Assertion** and click **OK** as shown in Figure 10-28.

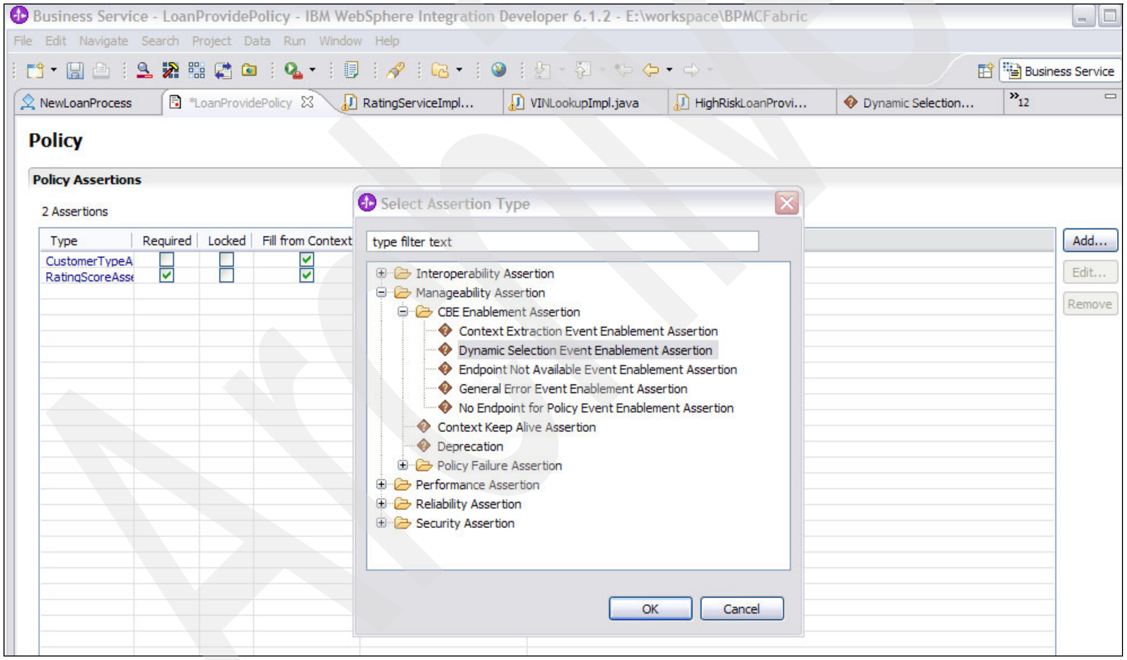


Figure 10-28 Selecting Dynamic Selection Event Enablement Assertion

8. Clear the Required check box and select the enable Dynamic Selection event check box, as shown in Figure 10-29.

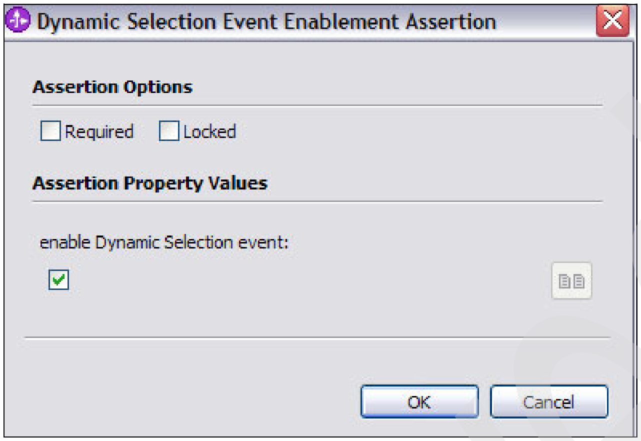


Figure 10-29 Enabling dynamic Selection event

9. Click the active changes in the Repository Changes Explorer and click **Submit Changelist**.
10. Choose **ITSOCBA** and click **Next**.
11. Move all the changes from the Available Changes column to the Selected Changes column and click **Finish**.
12. Login to the Fabric console using user ID fabric and click **My Inbox**.
13. Click the latest Change List Submitted.
14. Click **Approve** and click **Publish**.
15. Test the sample application by specifying the following values. Follow the steps in “Testing the sample application” on page 353 to test the application.
    – CustomerIdentificationNumber: 200
    – CustomerAddress: <Any string value>
    – VIN: 123
    – LoanAmountRequested: 1000
    – BankID: ITSO
16. The results should be as shown in Figure 10-30 on page 357.

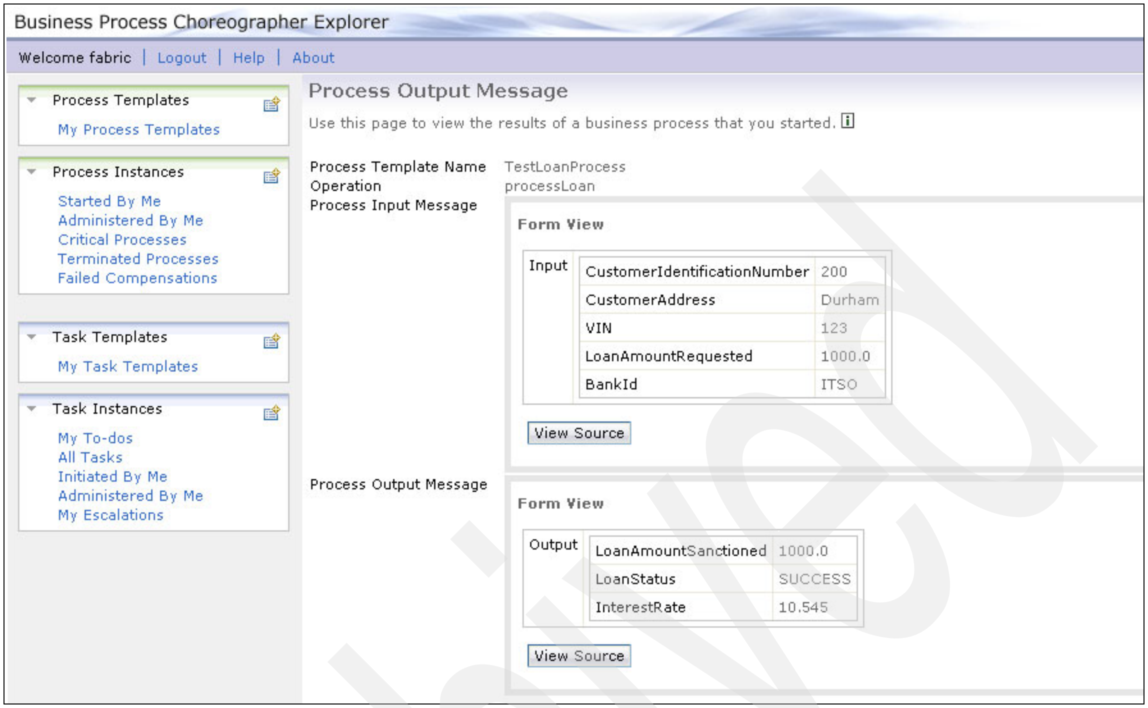


Figure 10-30 Sample application test result

Viewing events

We have tested the sample application twice. There is a Dynamic Selection Event generated for each test. Login to CBE Browser to view the events.

1. Login to CBE Browser by using the user wps. In our environment the URL is http://itsodmgr:9060/ibm/console/cbebrowser.
2. Click **All Events**. There should be two events as shown in Figure 10-31.

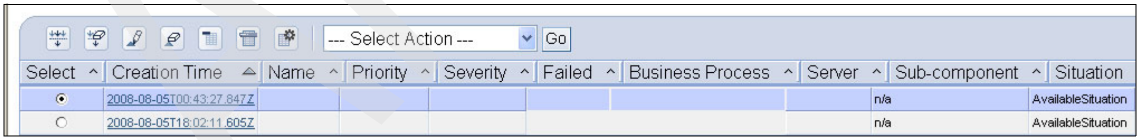
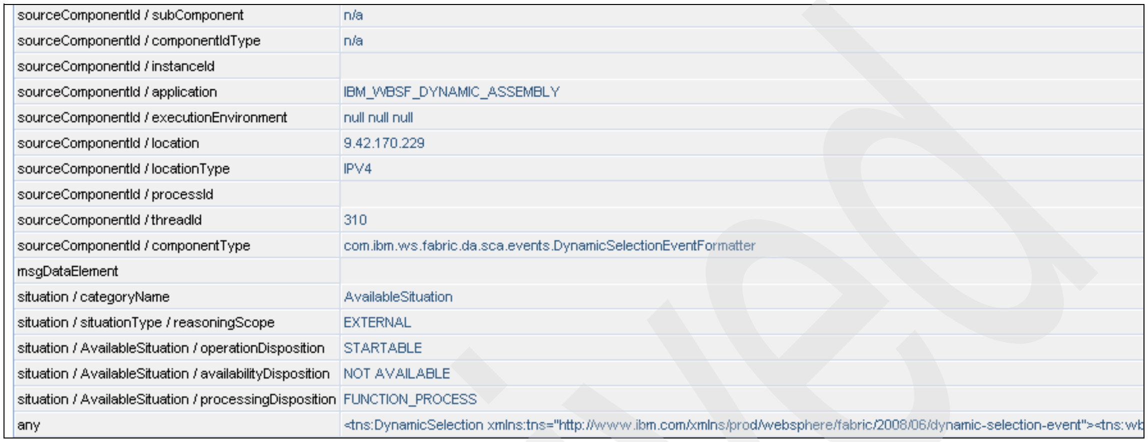


Figure 10-31 CBE Event List

3. Click the first event link. The event data will be listed.

4. Look for the <tns:address> XML tag in the any text box. The <tns:address> tag will contain the endpoint URL http://itsodmgr/ITSO_implWeb/sca/PremiumLoanProviderExport, which implies that PremiumLoanProvider endpoint is selected when we tested the application the first time (Figure 10-32).



sourceComponentId / subComponent	n/a
sourceComponentId / componentIdType	n/a
sourceComponentId / instanceId	
sourceComponentId / application	IBM_WBSF_DYNAMIC_ASSEMBLY
sourceComponentId / executionEnvironment	null null null
sourceComponentId / location	9.42.170.229
sourceComponentId / locationType	IPV4
sourceComponentId / processId	
sourceComponentId / threadId	310
sourceComponentId / componentType	com.ibm.ws.fabric.da.sca.events.DynamicSelectionEventFormatter
msgDataElement	
situation / categoryName	AvailableSituation
situation / situationType / reasoningScope	EXTERNAL
situation / AvailableSituation / operationDisposition	STARTABLE
situation / AvailableSituation / availabilityDisposition	NOT AVAILABLE
situation / AvailableSituation / processingDisposition	FUNCTION_PROCESS
any	<tns:DynamicSelection xmlns:tns="http://www.ibm.com/xmlns/prod/websphere/fabric/2008/06/dynamic-selection-event"><tns:wb

Figure 10-32 Event Details

5. Click the second event link Figure 10-31 on page 357 and look for the <tns:address> XML tag in the any text box. The <tns:address> tag contains the endpoint URL http://itsodmgr/ITSO_implWeb/sca/HighRiskLoanProviderExport which implies that HighRiskLoanProvider endpoint is selected when we tested the application the second time.

11

Incorporating WebSphere Business Monitor into a production topology

In this chapter we introduce WebSphere Business Monitor product as one of the service oriented architecture (SOA) products stack key. Incorporating WebSphere Business Monitor into WebSphere Process Server production topology assists business users to monitor runtime business processes. Therefore, they can make decisions or actions based on the monitored runtime data.

WebSphere Business Monitor can be installed in multiple topologies:

- Components on a single server.
- Components across multiple systems
- Components into a clustered topology

  The clustered topology will achieve a highly available environment with fail over support.

This chapter guides you through the necessary steps for installing WebSphere Business Monitor in a clustered topology using the Remote Messaging and Remote Support topology pattern.

11.1 WebSphere Business Monitor overview

WebSphere Business Monitor is a business-activity monitoring application that measures business performance, monitors real-time and completed processes, and reports on business operations. It measures business performance, monitors business processes, detects business situations, issues related alerts, and graphically presents business information. This helps the user identify business problems, correct exceptions, and change processes accordingly.

11.1.1 Install 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. This topology requires the following components:

- WebSphere Application Server V6.1.0.17 or 19
- WebSphere Process Server V6.1.2
- DB2 UDB ESE V8.2.6 fix pack 13 or V9.1
- IBM Tivoli Directory Server V6.0

> **Note:** Detailed step-by-step instructions for building the Remote Messaging and Remote Support topology pattern in WebSphere Process Server are provided in Chapter 5, “Configuring a Remote Messaging and Remote Support topology” on page 89.

The following software should be prepared before starting installation:

- WebSphere Business Monitor V6.1.2
- Alphablox V9.5

> **Note:** For Linux and AIX® systems. it is suggested to perform installations using a user with administrative privileges.

For more information regarding the supported WebSphere Business Monitor products, refer to the following Web page:

http://www-306.ibm.com/software/integration/wbimonitor/requirements/

11.1.2 Installation overview

This chapter describes how to create the topology shown in Figure 11-1. The installation steps in this chapter were performed in SUSE Linux Enterprise Server 10 SP1.

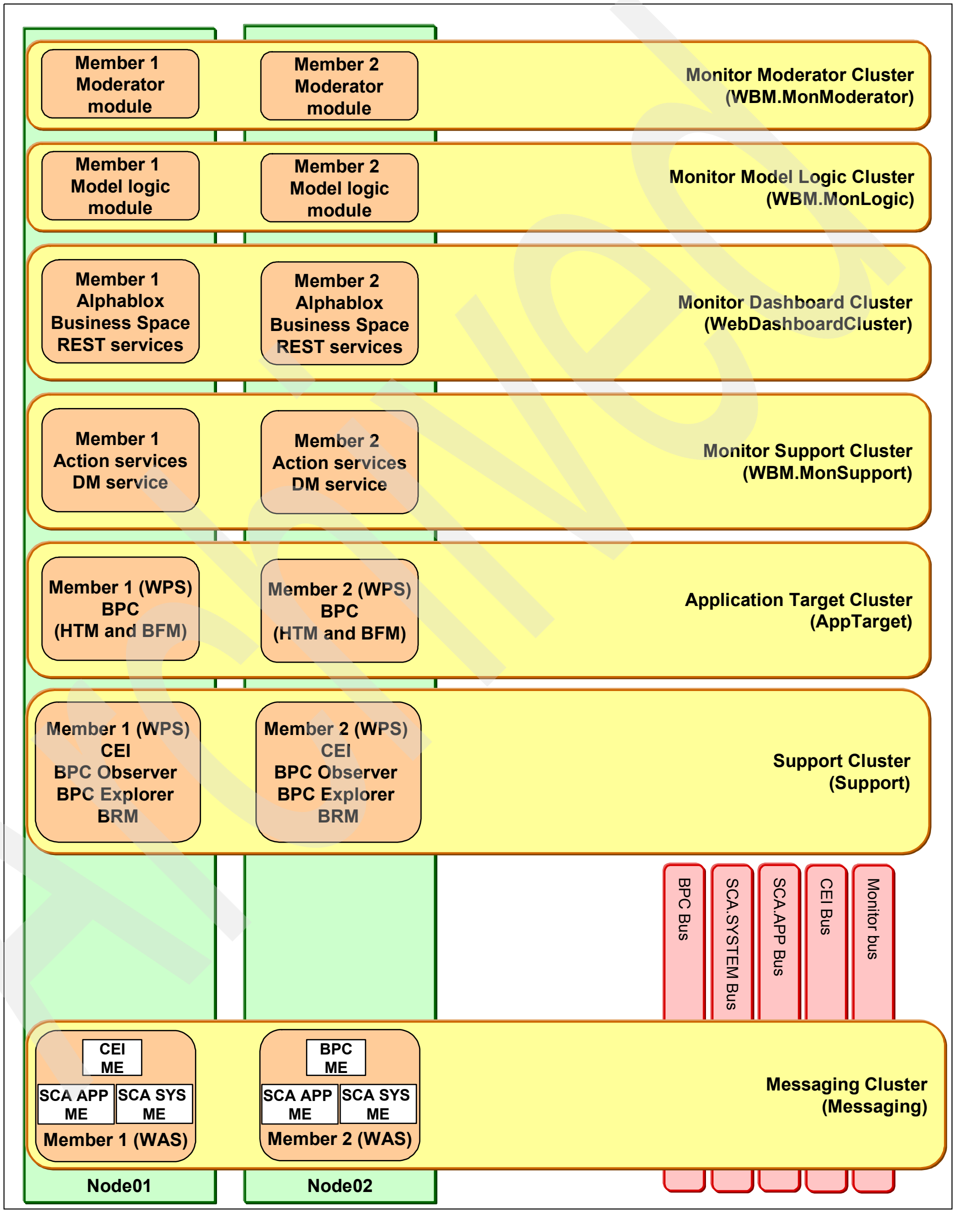


Figure 11-1 WebSphere Business Monitor topology

The topology shown in Figure 11-1 on page 361 contains the following clusters:

- 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.

- Monitor Support cluster

  The monitor support cluster was created for WebSphere Business Monitor. It contains the Monitor action services, and the data movement service.

- Monitor Dashboard cluster

  The monitor dashboard cluster was created for WebSphere Business Monitor. It contains the REST service, Alphablox, and Business Space.

- Monitor Model Logic cluster

  The monitor model logic cluster was created for WebSphere Business Monitor. It is to where the Monitor model logic module is deployed.

- Monitor Moderator cluster

  The monitor moderator cluster was created for WebSphere Business Monitor. The moderator is concerned with multiple event sources and out of sequence events.

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 (a service integration bus link) between a WebSphere Process Server cell and WebSphere Business Monitor cell is not required.

11.2 Creating the WebSphere Business Monitor profiles, database, and deployment manager

This section contains the following sections:

- "Installing the WebSphere Business Monitor binaries"
- "Creating the WebSphere Business Monitor database" on page 371
- "Augmenting the WebSphere Business Monitor profile with the WebSphere Process Server deployment manager profile" on page 372

11.2.1 Installing the WebSphere Business Monitor binaries

The first step to create a clustered WebSphere Business Monitor production topology is to install cluster nodes binaries. One of these nodes will be installed on the same machine as the WebSphere Process Server deployment manager. This node will be augmented to the existing WebSphere Process Server deployment manager profile.

Perform the following steps to install each node in the cluster:

1. Extract the compressed file of the WebSphere Business Monitor installation.
2. Prepare the Linux system for installation:
   a. Prepare the required disk space for source and installation.
   b. Adjust files permissions appropriately.
   c. Ensure network connectivity with other cluster members or machines.
3. Stop the WebSphere Process Server deployment manager before starting the installation.
4. From the CD installation image of WebSphere Business Monitor, run the Launchpad.sh script.
5. In the WebSphere Business Monitor installation page (Figure 11-2 on page 364), click the **WebSphere Business Monitor Installation** link.

Figure 11-2 WebSphere Business Monitor installation page

6. Click the **Launch the installation wizard of WebSphere Business Monitor** link.
7. In the Installation wizard welcome window (Figure 11-3), click **Next**.

Figure 11-3 Welcome window of the installation wizard

8. In the Software License Agreement window, accept the license agreement, then click **Next**.
9. In the System prerequisites check window (Figure 11-4) the launchpad checks the system against the installation requirements. Click **Next**.

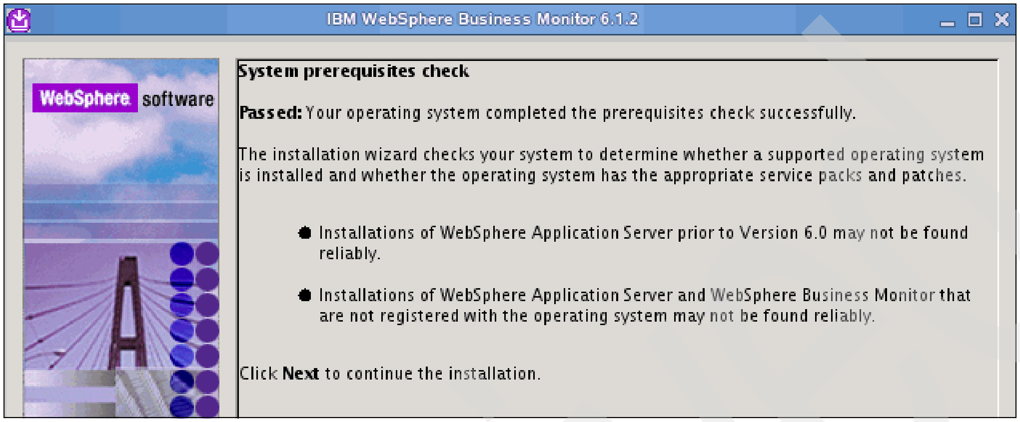


Figure 11-4 System Prerequisites check window

10. In the Installation type selection window (Figure 11-5) select **Advanced installation** then click **Next**.

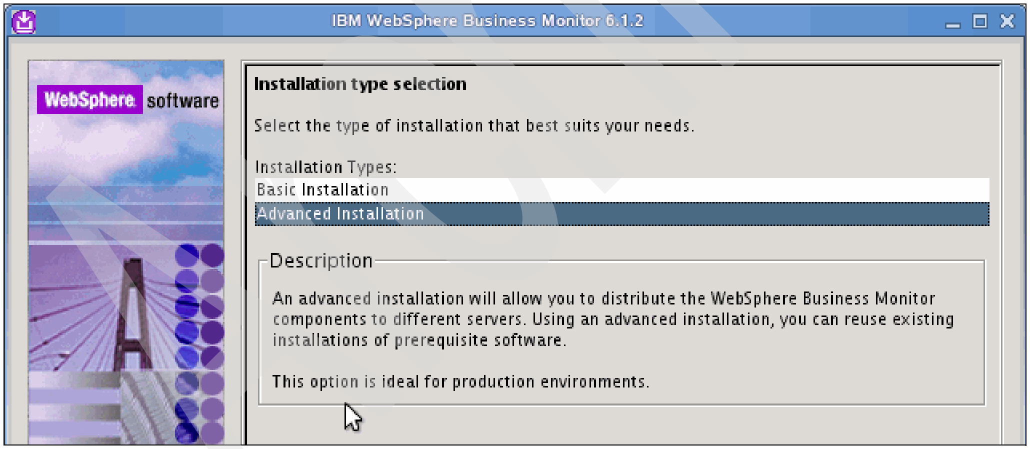


Figure 11-5 Installation type selection window

11. In the Component selection window (Figure 11-6) select the check boxes for the following components, then click **Next**:

- Business Monitor Server including Business Space
- Monitor Database
- Information center (optional)

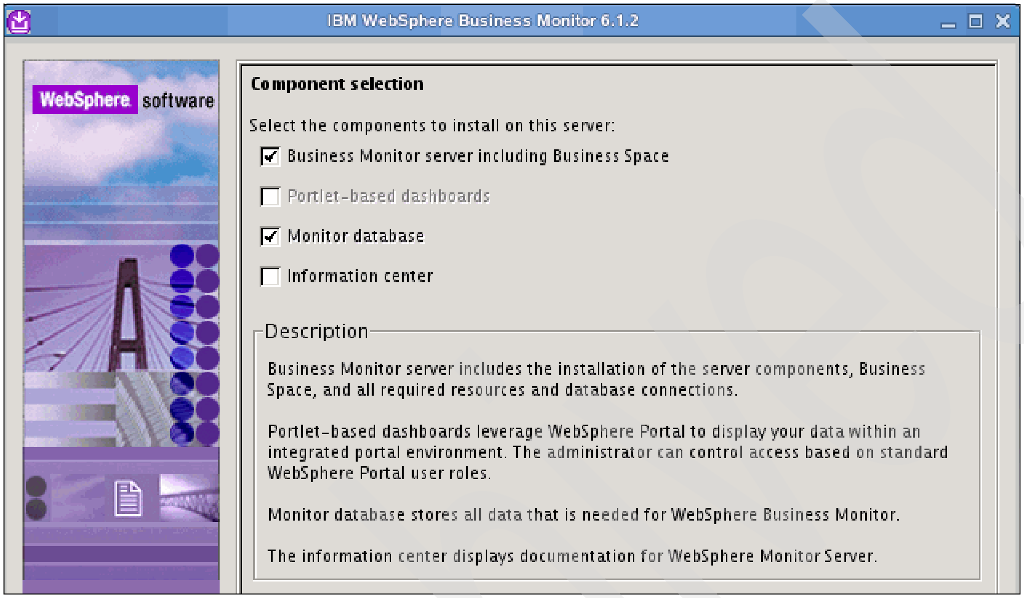


Figure 11-6 Component selection window

12. In the Detected WebSphere Application Server window (Figure 11-7) select the Use the existing installation of WebSphere Application Server Network Deployment radio button. Click **Next**.

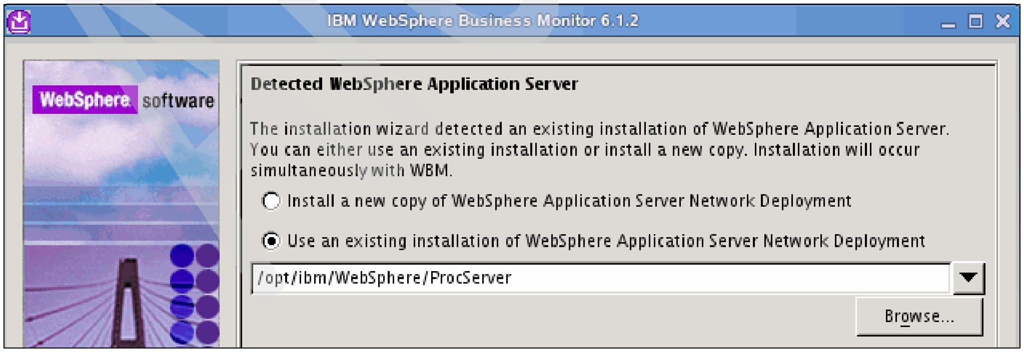


Figure 11-7 Detected WebSphere Application Server window

13. In the Feature selection window (Figure 11-8) select the Alphablox features for Business Space radio button. Click **Next**.

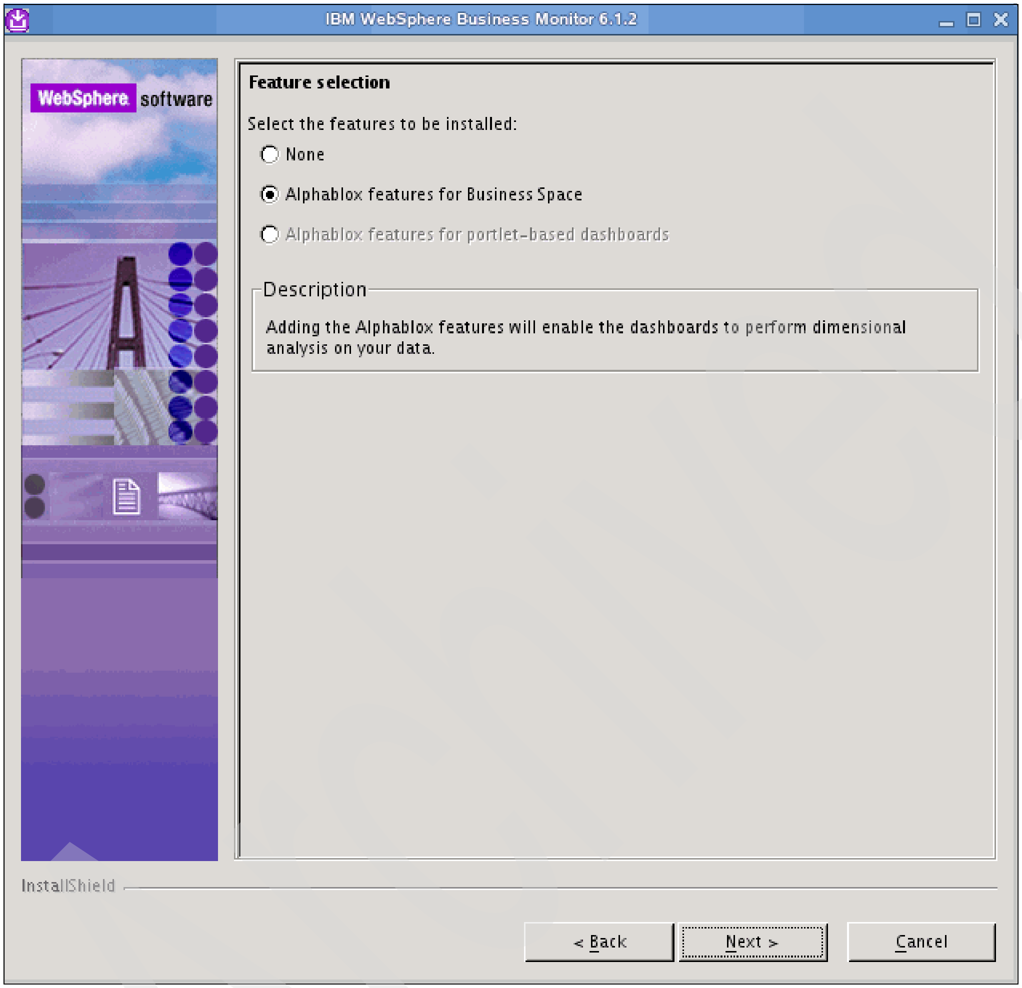


Figure 11-8 Feature Selection window

14. In WebSphere Business Monitor profile environments window (Figure 11-9) select **None**, then click **Next**.

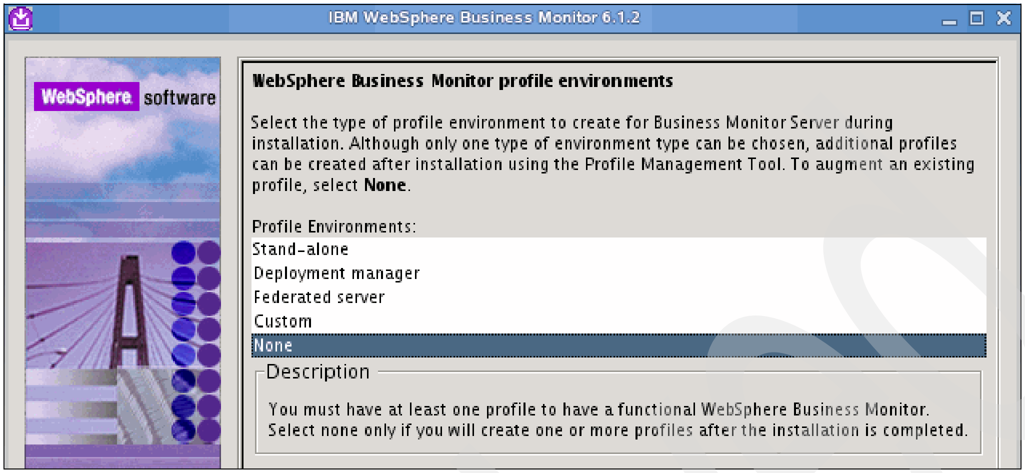


Figure 11-9 WebSphere Business Monitor profile environment window

15. A warning message appears (Figure 11-10). Click **Yes** to confirm the selected option. We will create profiles later.

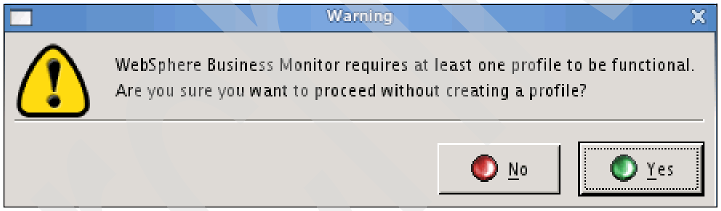


Figure 11-10 Profile warning message

16. The Installation summary window (Figure 11-11) summarizes the components to be installed. Click **Next** to start the installation.

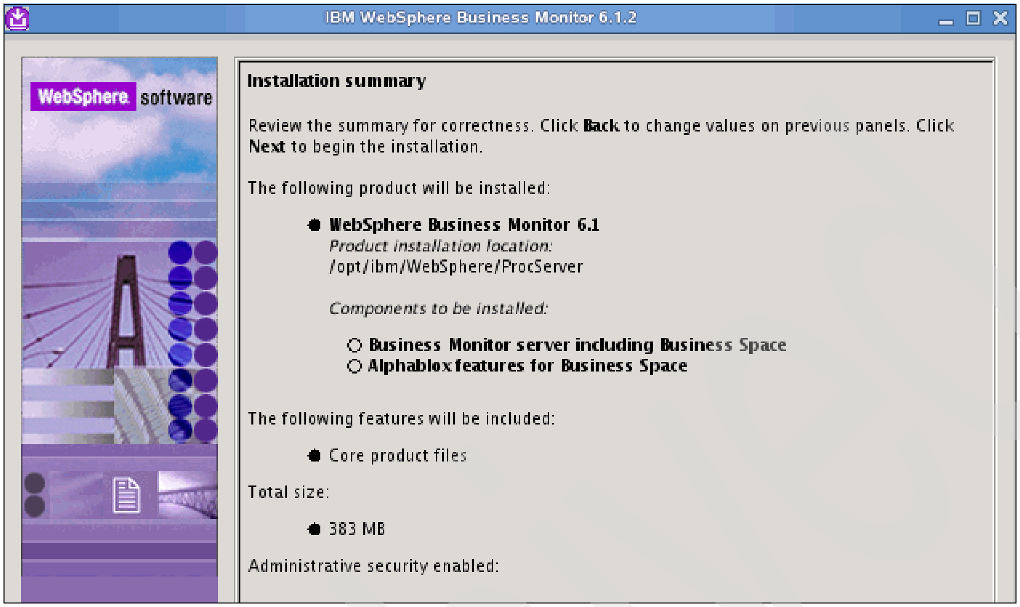


Figure 11-11 Installation Summary window

17. When the installation process completes, an Installation results window appears (Figure 11-12). Click **Finish** if the installation is successful. If the installation is not successful, follow the instructions in the installation result window to diagnose the installation problem.

Figure 11-12 Installation Success window

18. In the First steps window (Figure 11-13) click the **Installation verification** link to verify that the WebSphere Business Monitor components are installed.

Figure 11-13 WebSphere Business Monitor first steps

A window is displayed showing the verification steps (Figure 11-14). Make sure that all steps are passed.

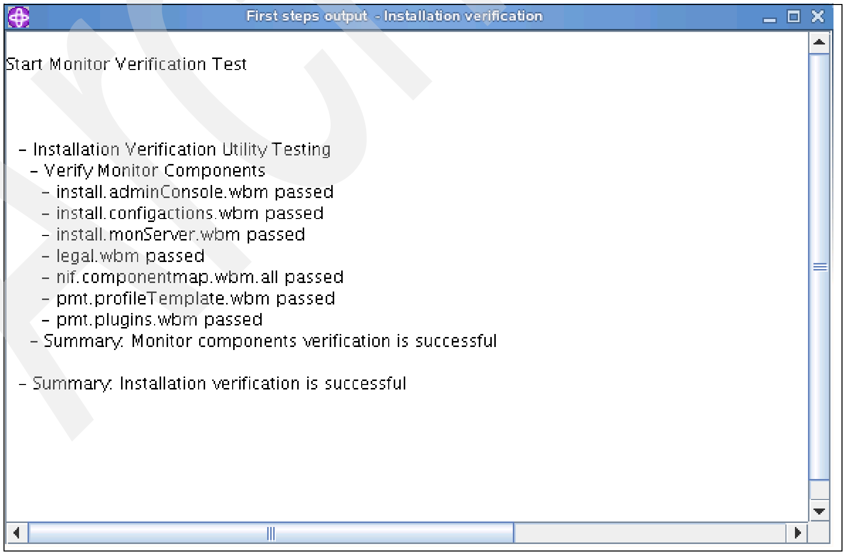


Figure 11-14 Installation verification window

11.2.2 Creating the WebSphere Business Monitor database

WebSphere Business Monitor uses a single database for persistence. The default name is MONITOR. We create this database on the same DB2 server used by the existing Remote Messaging and Remote Support topology. Perform the following steps to create the WebSphere Business Monitor database:

1. Perform the following steps to prepare the database creation script file for execution.

   a. Locate the script createDatabaseDb2.dll, found at: monitor_CD_root/WBM/scripts/database.

   > **Note:** monitor_CD_root represents the directory where you extracted the WebSphere Business Monitor CD or downloadable image.

   b. Edit the following variables in the createDatabaseDb2.dll script:

   - $DBNAME$: This variable represents the name of the Monitor database (for example, MONITOR).
   - $SCHEMA$: This variable represents the name of the Monitor schema (for example, MONITOR).
   - $TSDIR$: This variable represents the table space directory. If $TSDIR$ is omitted from the data file specification of a table space, the data file will be created in the Database Manager directory.
   - $TERRITORY$: This variable represents the locale of the data in the database (for example, en_US).

   c. Save and close the file.

2. Open the DB2 command line interface and run the createDatabaseDb2.dll script using the following command:

```
db2 -tf createDatabaseDb2.ddl
```

3. Bind the command line interface to the Monitor database using the following commands:

```
db2 connect to MONITOR
db2 bind DB2_installation_directory/bnd/@db2cli.lst blocking all
grant public
db2 connect reset
```

> **Note:** DB2_installation_directory represents the directory where DB2 is installed. For example: /home/db2inst1/sqllib/.

4. The result of bind command should be as shown in Figure 11-15.

```
LINE     MESSAGES FOR db2cli.lst
------   --------------------------------------------------------------------
         SQL0061W  The binder is in progress.
         SQL0091N  Binding was ended with "0" errors and "0" warnings.
minst1@db2v91:~> _
```

Figure 11-15 Result of Binding command

11.2.3 Augmenting the WebSphere Business Monitor profile with the WebSphere Process Server deployment manager profile

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 profiles, 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. In the Profile Management Tool window click **Augment** (Figure 11-16).

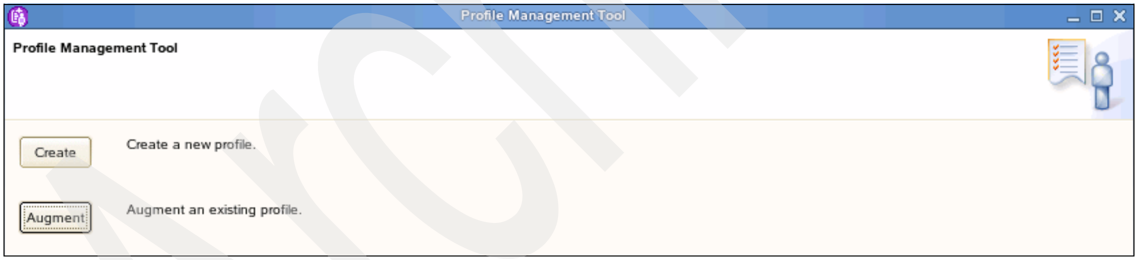


Figure 11-16 Profile Augmentation tool

3. In the Welcome to Profile Management Tool window (Figure 11-17) click **Next**.

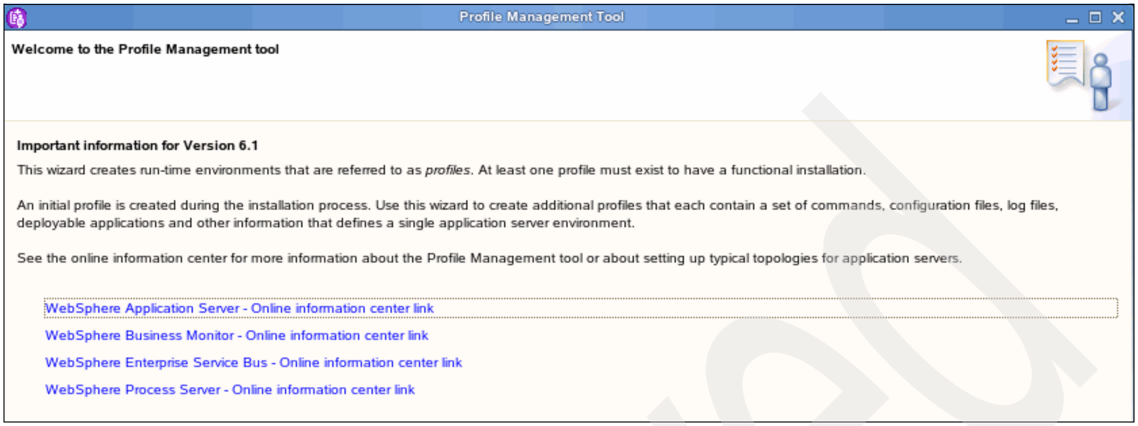


Figure 11-17 Profile Management Tool Welcome page

4. In the Profile Selection window (Figure 11-18) select the name of the existing WebSphere Process Server deployment manager profile (in our example Dmgr01) profile name and click **Next.**

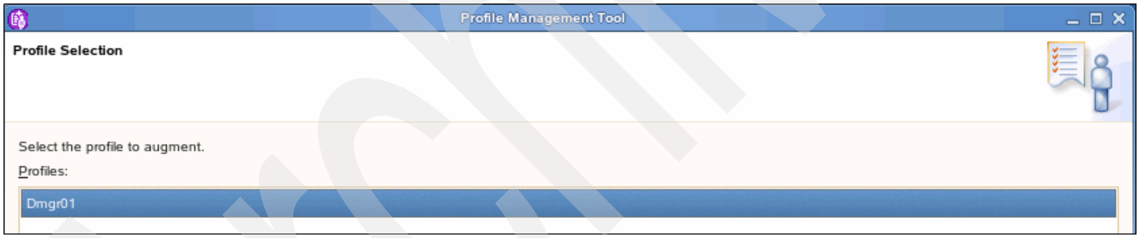


Figure 11-18 Profile Selection window

5. In the Augment Selection window (Figure 11-19) select **WebSphere Business Monitor deployment manager** then click **Next**.

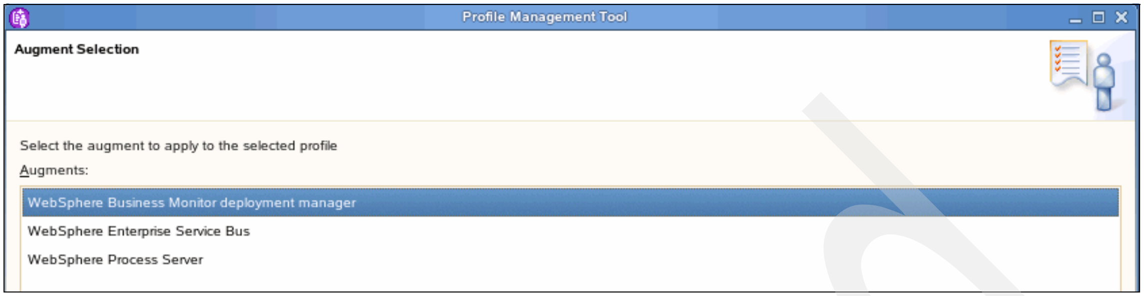


Figure 11-19 Augmentation Selection window

6. In the Profile Augmentation Options window (Figure 11-20) select the Advanced profile augmentation radio button. This will enable us to manually configure the monitor database and credentials. Click **Next**.

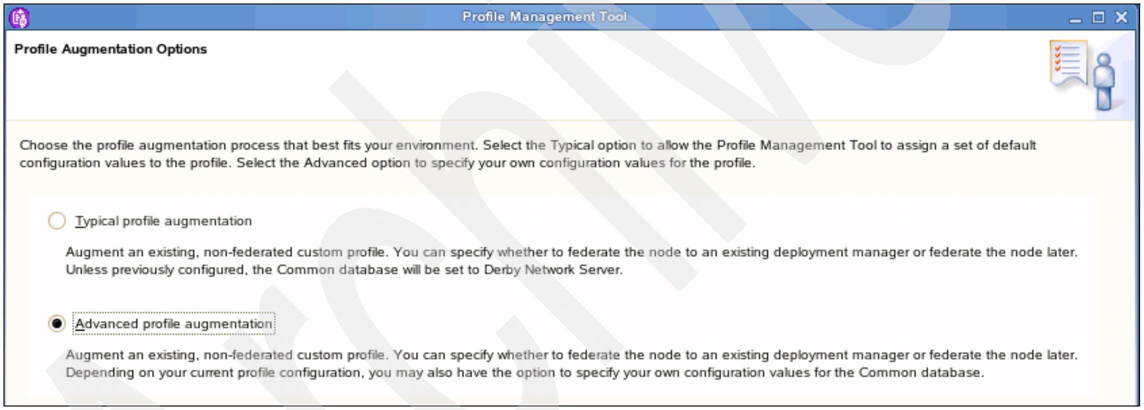


Figure 11-20 Profile Augmentation Options window

7. In the Database Configuration window (Figure 11-21 on page 375) perform the following steps:
   a. Select **DB2 Universal Database** in the Database product drop-down menu.
   b. Select the Use an existing database radio button in the Database creation options sections.
   c. Click **Next**.

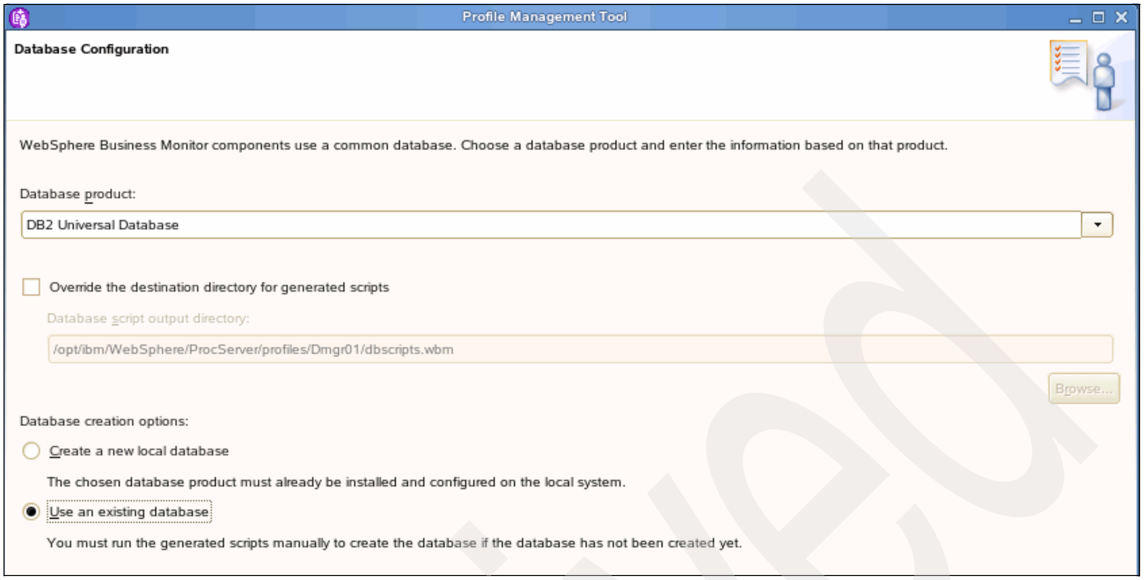


Figure 11-21 Database Configuration window

8. In the Database Configuration (Part 2) window (Figure 11-22 on page 376) perform the following steps:
   a. Enter a DB2 authenticated user name in the User name text box.
   b. Enter the DB2 user password in Password and Confirm password text boxes.
   c. Populate the Location (directory) of JDBC drive classpath files text box. The default is: /opt/ibm/WebSphere/ProcServer/UniversalDriver.wbm/lib
   d. Select the required JDBC driver type. As we are using a remote database, set the JDBC driver type to 4. If DB2 is installed locally on the WebSphere Business Monitor deployment manager machine, and you are planning to catalog the remote MONITOR database or create it on this machine, you should select JDBC drive type 2.
   e. Enter the host name or IP of the database server machine in the Database server host name or IP address text box. It is recommended to adjust the hosts file on the two machines: the DB2 server machine and the WebSphere Business Monitor machine. The hosts file path is:
      - Windows: winodws_installation_folder\system32\drivers\etc
      - Linux: /etc
   f. Enter the database port number in the Database TCP/IP service port or listener port text box. The default is 50000.

g. Click **Next**.

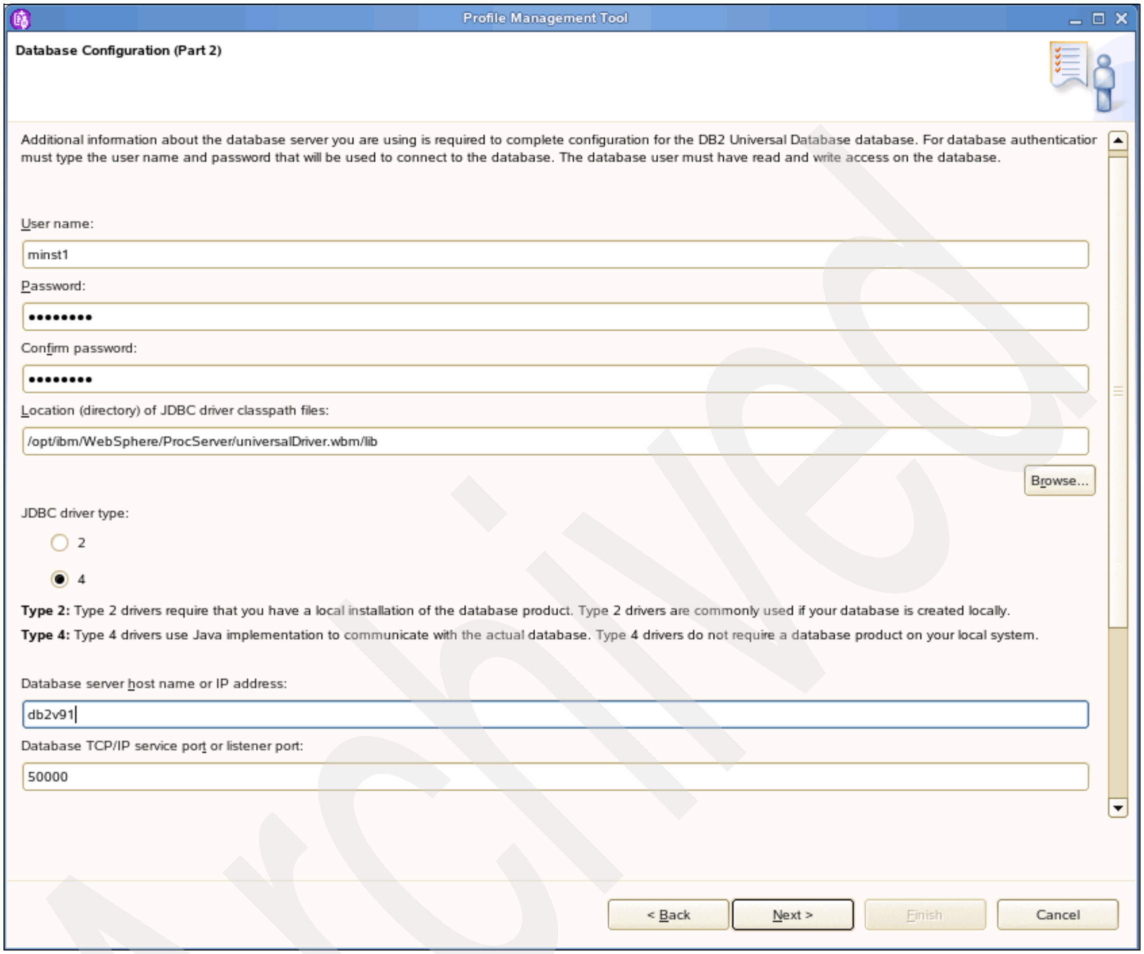


Figure 11-22 Database Configuration (Part 2) window

9. The Profile Augmentation Summary window (Figure 11-23) contains a detailed description of the augmentation artifacts. To start augmenting the deployment manager profile, click **Augment**.

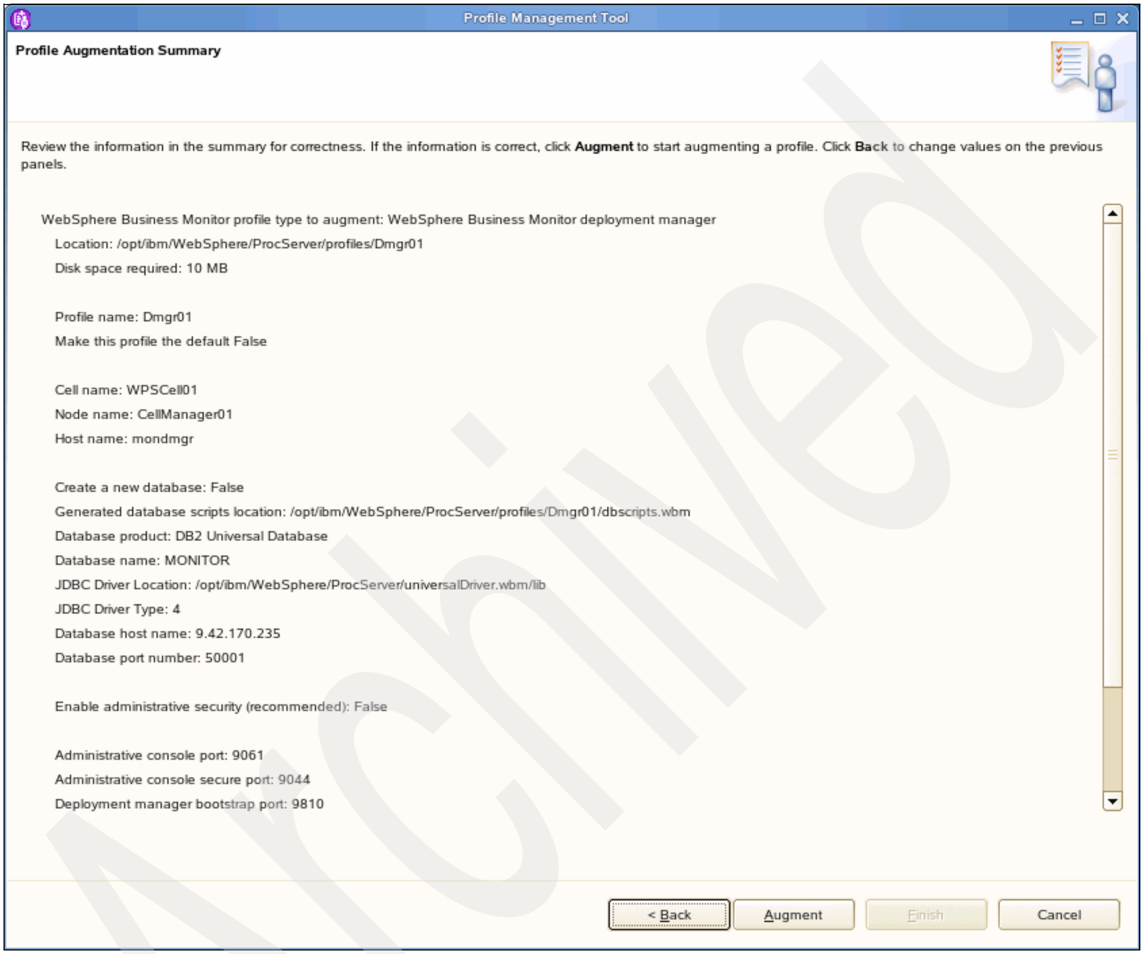


Figure 11-23 Profile Augmentation Summary window

10. In the Profile Augmentation Complete window (Figure 11-24) click **Finish**. If the augmentation failed, check the logs for encountered problems and fix them.

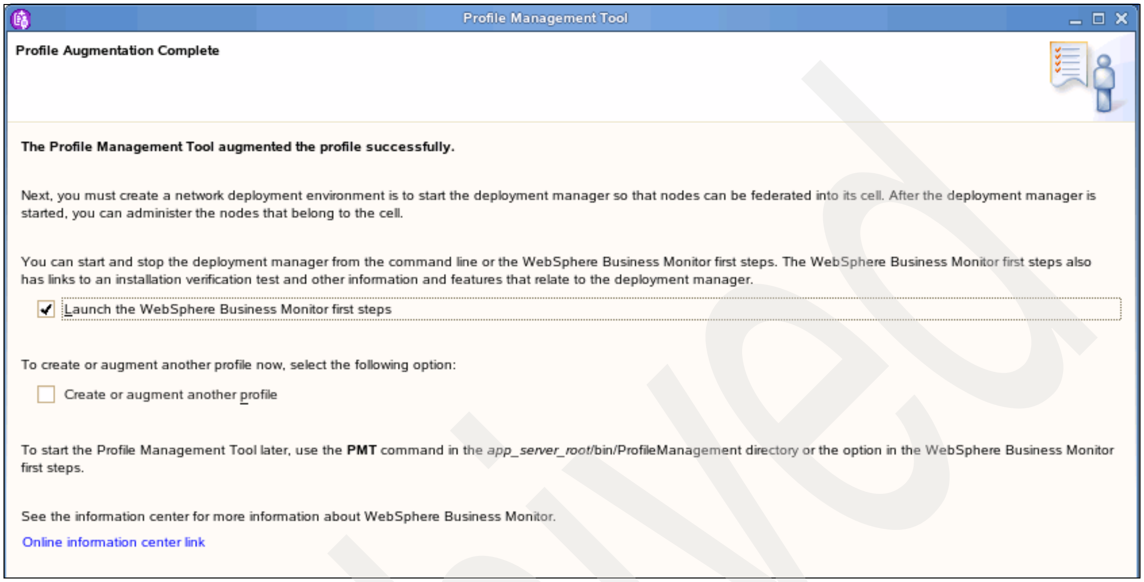


Figure 11-24 Profile Augmentation Complete window.

11.3 Creating and federating clusters members

After installing cluster members binaries in 11.2.1, "Installing the WebSphere Business Monitor binaries" on page 363 you should create a profile for each cluster member node and federate it to the WebSphere Business Monitor deployment manager. The following section describe the required steps for creating node profiles and federating them into the deployment manager.

Notes on creating and federating clusters members:

- It is suggested to create separate profiles for WebSphere Business Monitor cluster members instead of augmenting them with the WebSphere Process Server profiles.
- 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 will fail.

To create and federate cluster member nodes, perform the following steps:

1. Run the profile management tool:

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

2. In the Profile Management Tool window (Figure 11-25) click **Create**.

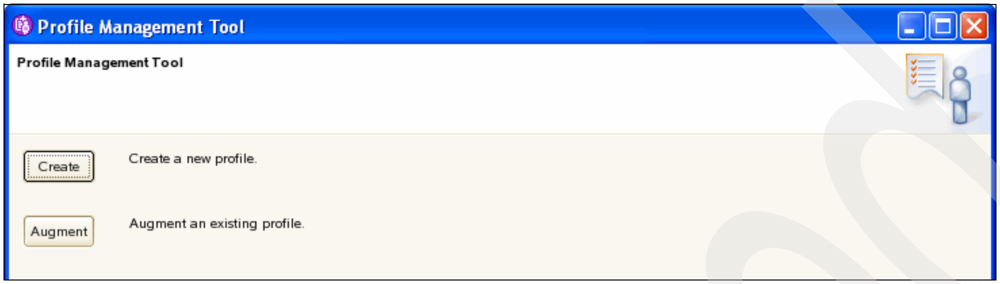


Figure 11-25 Profile Management tool

3. In the Welcome to the Profile Management tool window click **Next**.
4. In the Environment Selection window (Figure 11-26) select **WebSphere Business Monitor** and click **Next**.

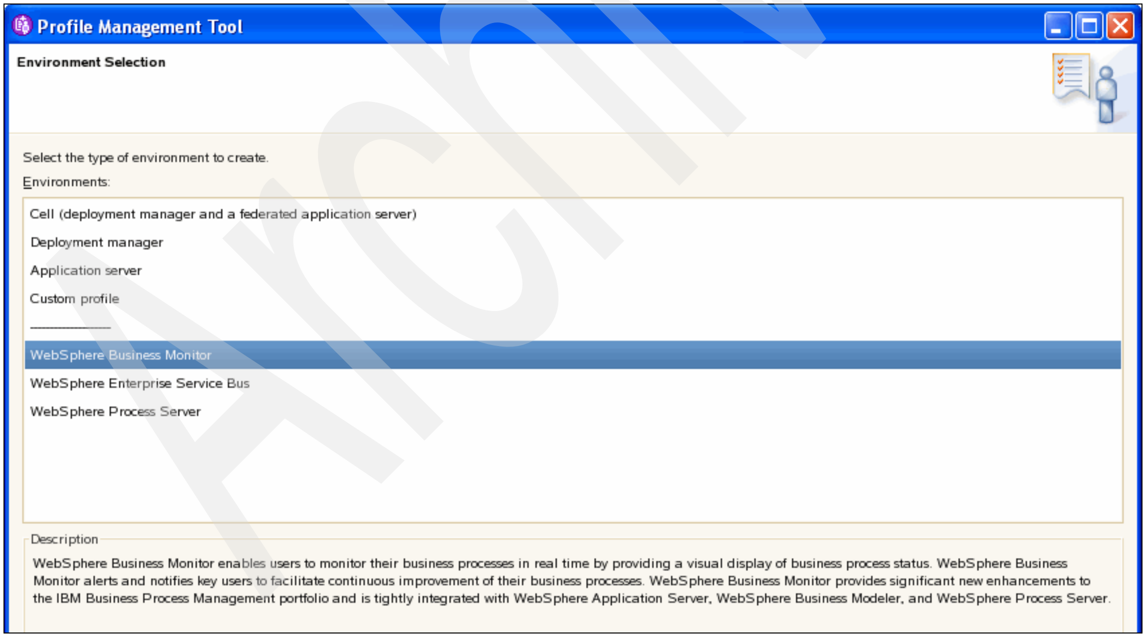


Figure 11-26 Environment Select window

5. In the Profile Type Selection window (Figure 11-27) select **WebSphere Business Monitor custom profile**. This will enable you to create a custom profile without an administration console and federate it to the required deployment manager. Click **Next**.

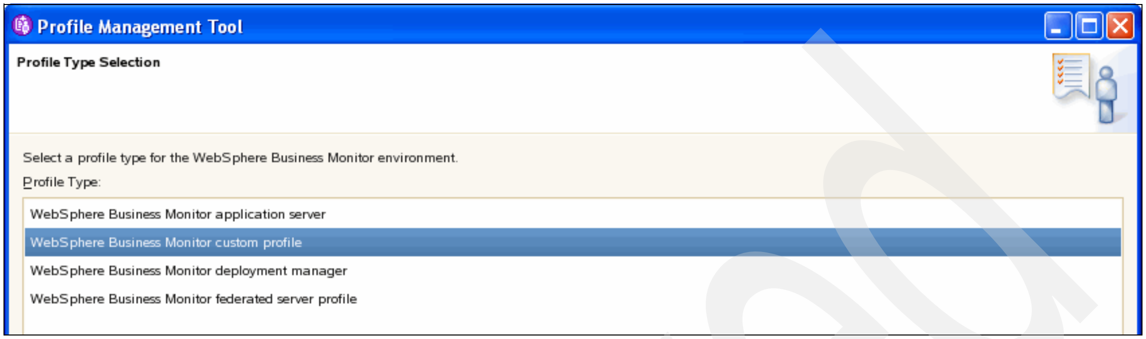


Figure 11-27 Profile Type Selection window

6. In the Profile Creation Options window (Figure 11-28) select the Advanced profile creation radio button to configure the database, deployment manager host name, and security. Click **Next**.

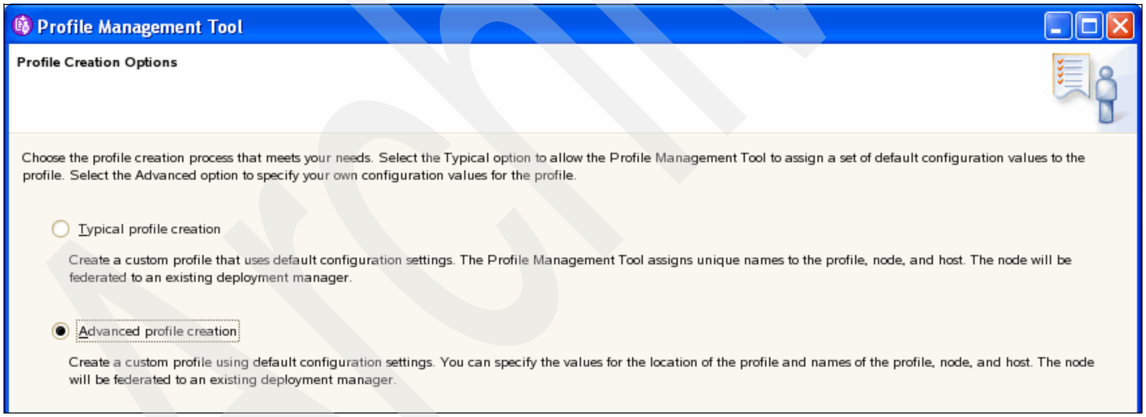


Figure 11-28 Profile Creation Options window

7. In the Profile Name and Location window (Figure 11-29) perform the following steps:
   a. Enter the required profile name in Profile name text box. We entered MonServer01.
   b. Enter the required folder name and path for the created profile in the Profile directory text box.
   c. Click **Next**.

Figure 11-29 Profile Name and Location

8. In the Node and Host Names window (Figure 11-30) perform the following steps:
   a. Enter the required node name in Node name text box. This name should be unique among cluster members.
   b. Enter the profile host name the Host name text box.
   c. Click **Next**.

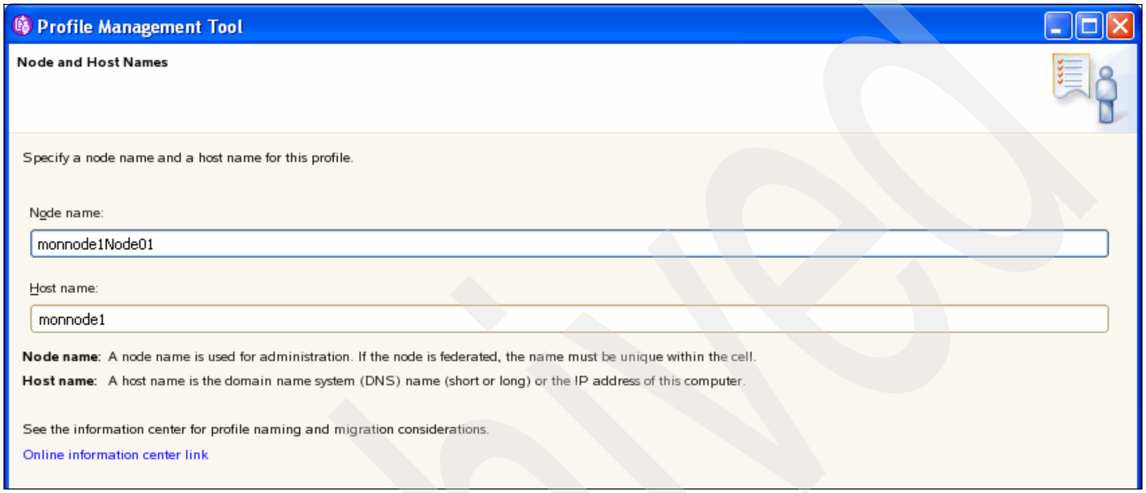


Figure 11-30 Node and Host Names window

9. In the Federation window (Figure 11-31) perform the following steps:
   a. Enter the host name or IP address of the deployment manager in the Deployment manager host name or IP address text box.
   b. Enter the SOAP port number of the deployment manager in the Deployment manager SOAP port number text box. The default is 8879.
   c. If security is enabled on the deployment manager, populate the User name and Password text boxes.
   d. Click **Next**.

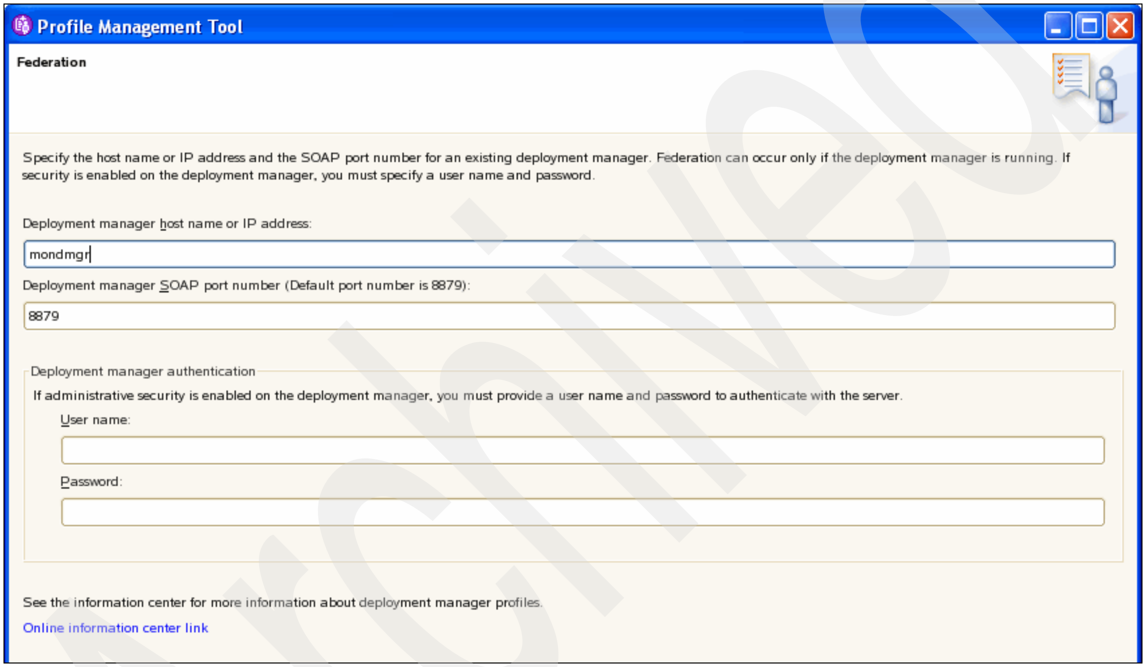


Figure 11-31 Federation window

10. In the Port Values Assignment window (Figure 11-32), keep the default values unless other port numbers are required. Click **Next**.

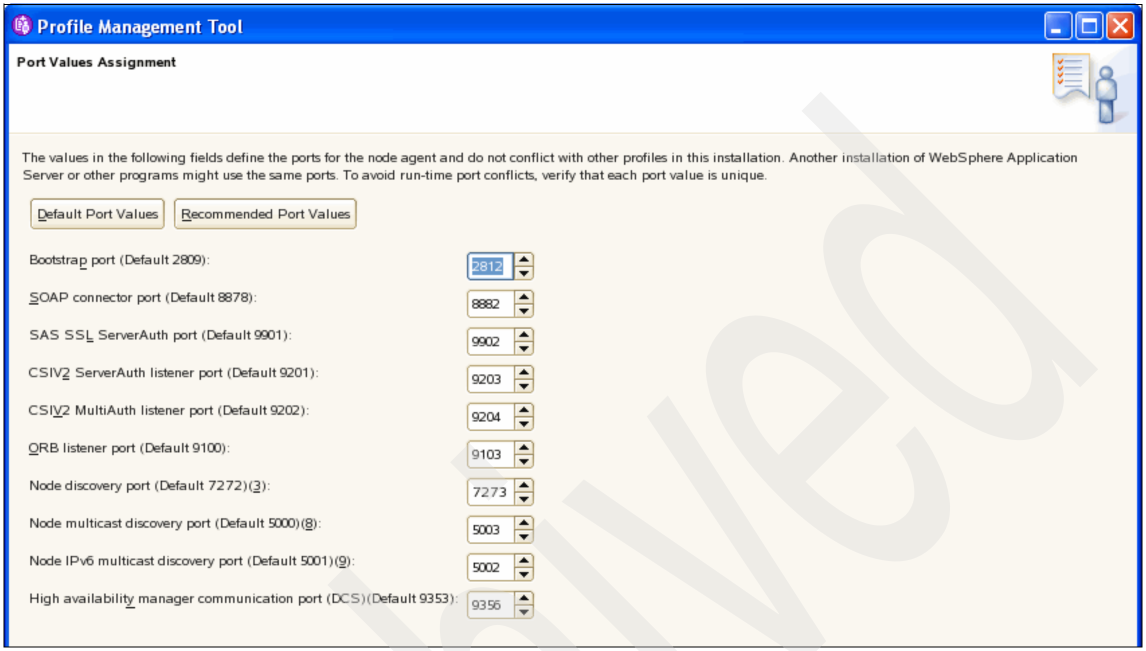


Figure 11-32 Port Values Assignment window

11. In the Database Configuration window (Figure 11-33) perform the following steps:
   a. Select **DB2 Universal Database** in the Choose the database product used on the Deployment Manager drop-down menu.
   b. Enter the location of the JDBC driver in the Location (directory) of JDBC driver classpath files text box.
   c. Click **Next**.

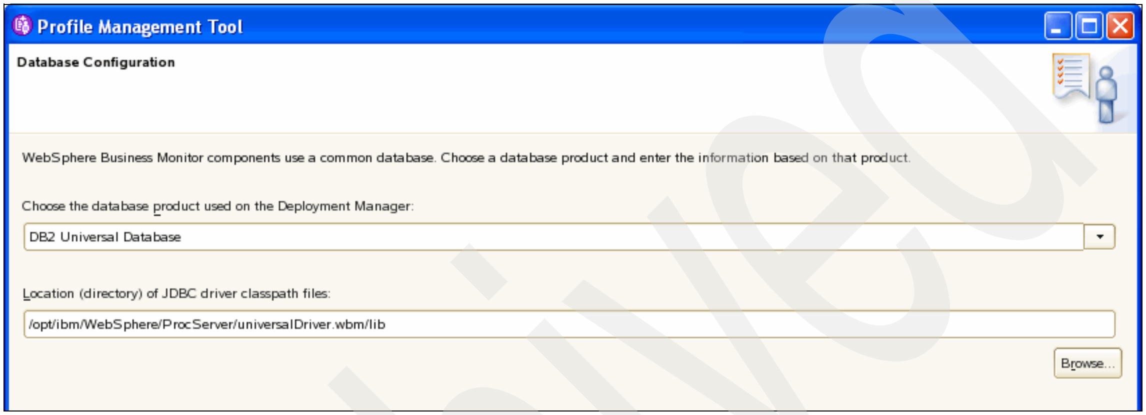


Figure 11-33 Database Configuration

12. The Profile Creation Summary window (Figure 11-34) contains detailed information about the profile to be created and federated. Click **Create**.

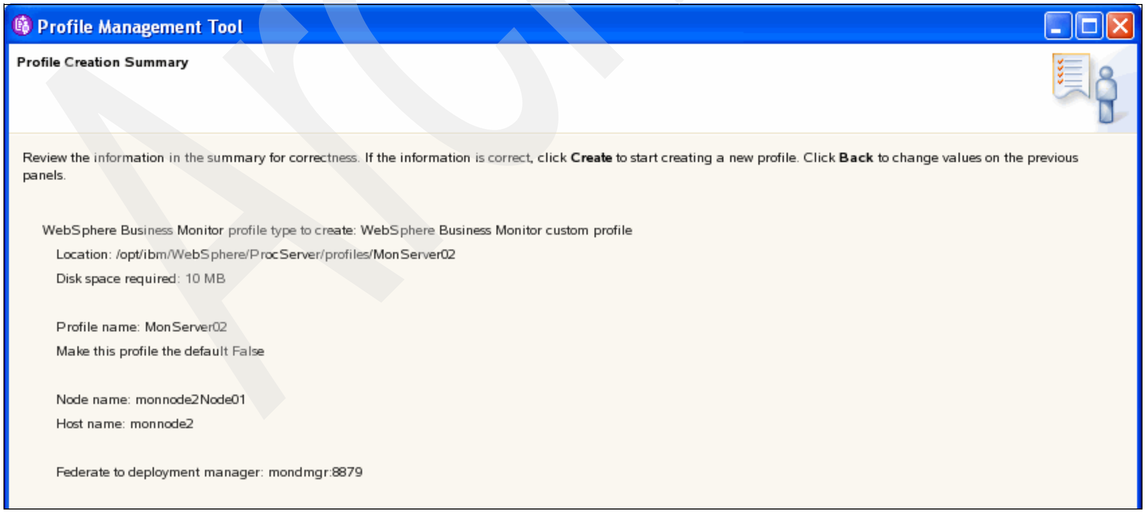


Figure 11-34 Profile Creation Summary

Note: If the profile creation failed, check the log file indicated in the Profile Management Tool window.

Repeat this process to create and federate additional cluster node profiles. We created a second profile called MonServer02.

11.4 Creating and configuring WebSphere Business Monitor clusters

In this section we describe how to create and configure the WebSphere Business Monitor clusters that we will add to the Remote Messaging and Remote Support topology.

The following steps are required to perform the necessary configurations for finalizing the WebSphere Business Monitor cluster installation:

1. Create WebSphere Business Monitor clusters
2. Enable CEI for a WebSphere Business Monitor cluster
3. Create the WebSphere Business Monitor bus
4. Create an Emitter Factory.
5. Install WebSphere Business Monitor applications.

In this section we will create four clusters:

- Monitor Model Logic cluster
- Monitor Support cluster
- Monitor Dashboard cluster
- Monitor Moderator cluster

Although we adopt this grouping of clusters in this chapter, other cluster topologies can also be created. Alternatively, all WebSphere Business Monitor functionality could be placed into a single cluster.

Table 11-1 on page 387 displays the possible cluster types for WebSphere Business Monitor that can be created, and the applications that should be deployed on each cluster. Additionally, Table 11-1 on page 387 displays whether each cluster type could be configured for load balancing (LB) and high availability (HA) or HA only.

Table 11-1 Cluster types and required applications

Cluster type	**Required applications**	**CEI enabled**	**LB and HA / HA**
Monitor model logic cluster	<Monitor_model> logic module	True	Both LB and HA
Monitor model event moderator cluster	<Monitor_model> moderator module	True	HA
Monitor Dashboard cluster	► Business Space application ► Alphablox application ► [optional]MobileDashboard.ear ► WBMDashboardRESTProxy.ear ► MonitorRestServices.ear	False	Both LB and HA
Monitor support cluster	► Monactionmgr.ear ► DmsService.ear	False	Both LB and HA
Monitor Messaging engine cluster		True	Both LB and HA

Some tips for when creating clusters:

- Save the changes to master configuration after each configuration step.
- Synchronize the configuration changes among servers, clusters, and nodes.
- Restart the deployment manager and all nodes and servers after each configuration change in clusters and applications for the changes to take effect.

11.4.1 Creating the WebSphere Business Monitor clusters

In this section you will create four clusters, as shown in Table 11-2.

Table 11-2 Cluster information

Type of cluster	**Cluster name**
Monitor Support cluster	WBM.MonSupport
Monitor Dashboard (Business Space) cluster	WebDashboardCluster
Monitor Moderator cluster	WBM.MonModerator
Monitor Model Logic cluster	WBM.MonLogic

Perform the following steps to create each WebSphere Business Monitor cluster. We start by defining the Monitor Support cluster.

1. Open the WebSphere Business Monitor deployment manager Integrated Solutions Console. Select **Servers** → **Clusters** (Figure 11-35). Click **New**.

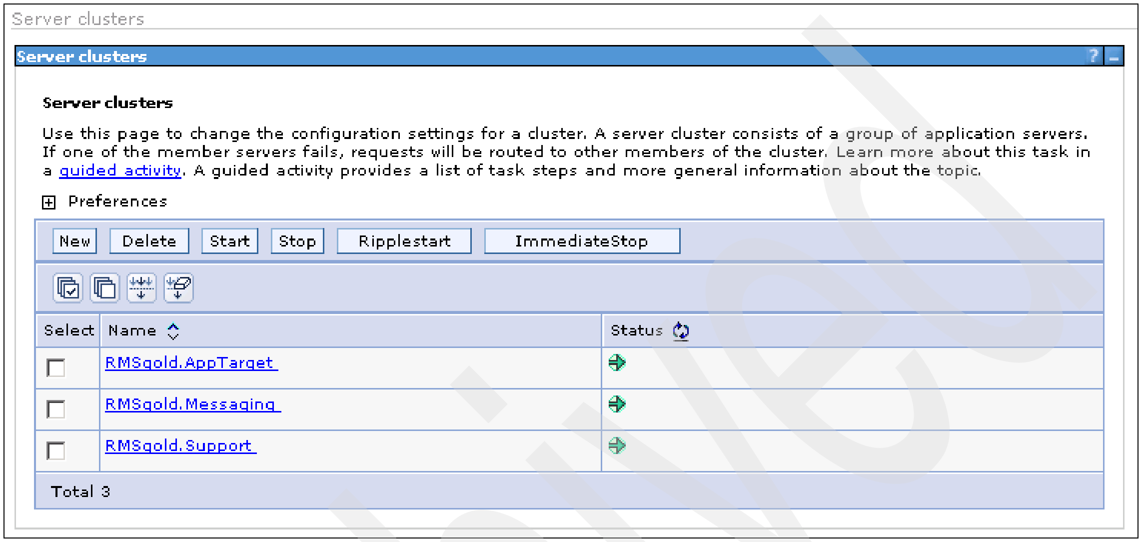


Figure 11-35 Clusters page

2. In step 1 in the cluster creation wizard (Figure 11-36), enter the required cluster name in the Cluster name text box. For the Monitor Support cluster we entered a cluster name of WBM.MonSupport. Click **Next**.

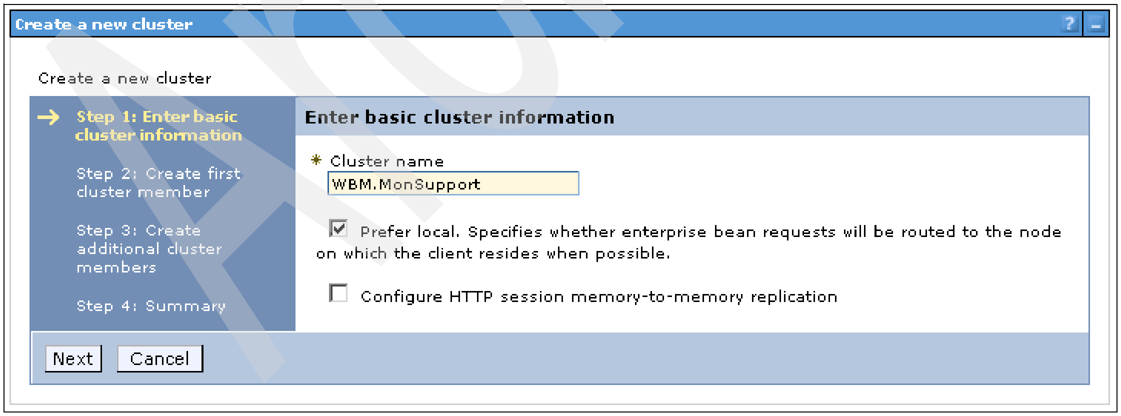


Figure 11-36 Step 1: Enter basic cluster information

3. In step 2 (Figure 11-37), perform the following steps for the first member of the cluster:

   a. Enter the required member name in Member name text box. We used monnode1.

   b. Select the corresponding application server from the Select node list.

   c. Select the Create the member using an application server template radio button, and select default_defaultWBM from the list.

      All other text boxes should keep their default values.

   d. Click **Next**.

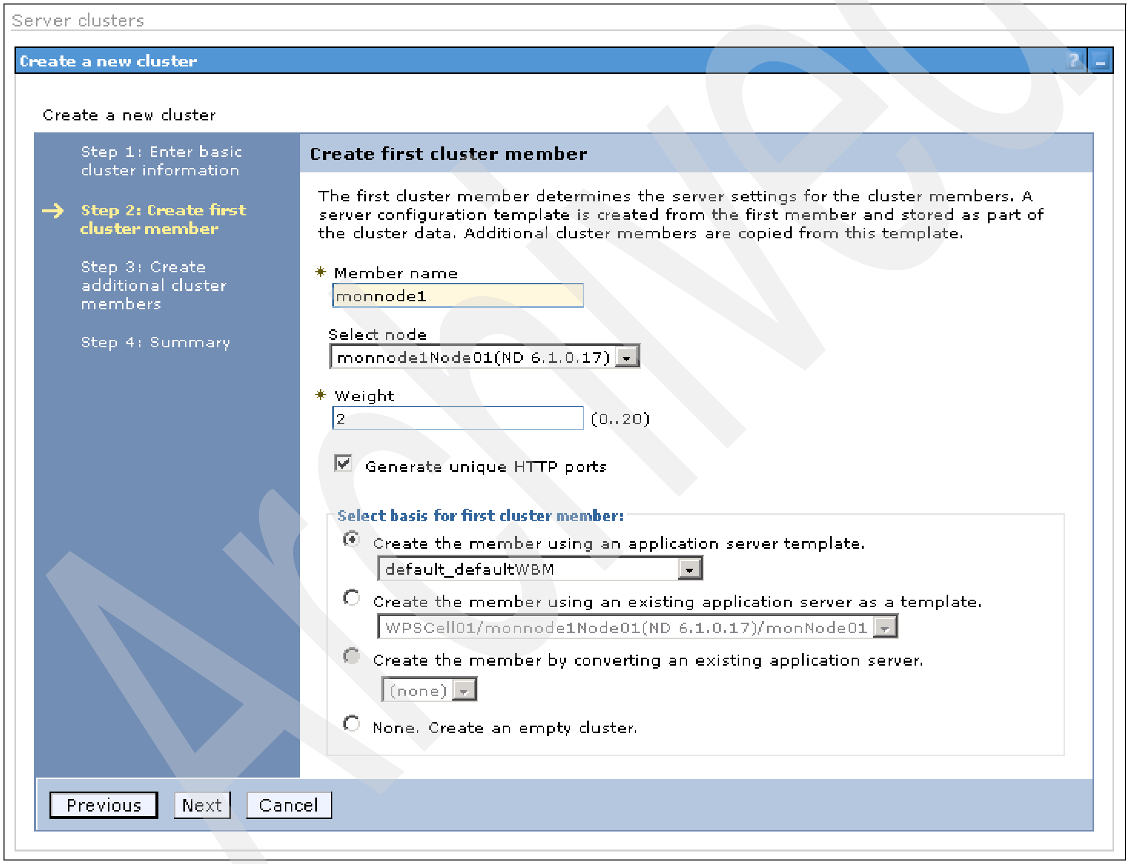


Figure 11-37 Step 2: Create first cluster member

4. In step 3 (Figure 11-38), we add a second cluster member. Perform the following steps:
   a. Type the required member name in Member name text box. We used monnode2.
   b. Select the corresponding application server from the Select node list.
   c. Click **Add member**. The new member should be added to the table.

   > **Note:** Repeat these steps to add as many cluster members as necessary to the cluster.

   d. Click **Next**.

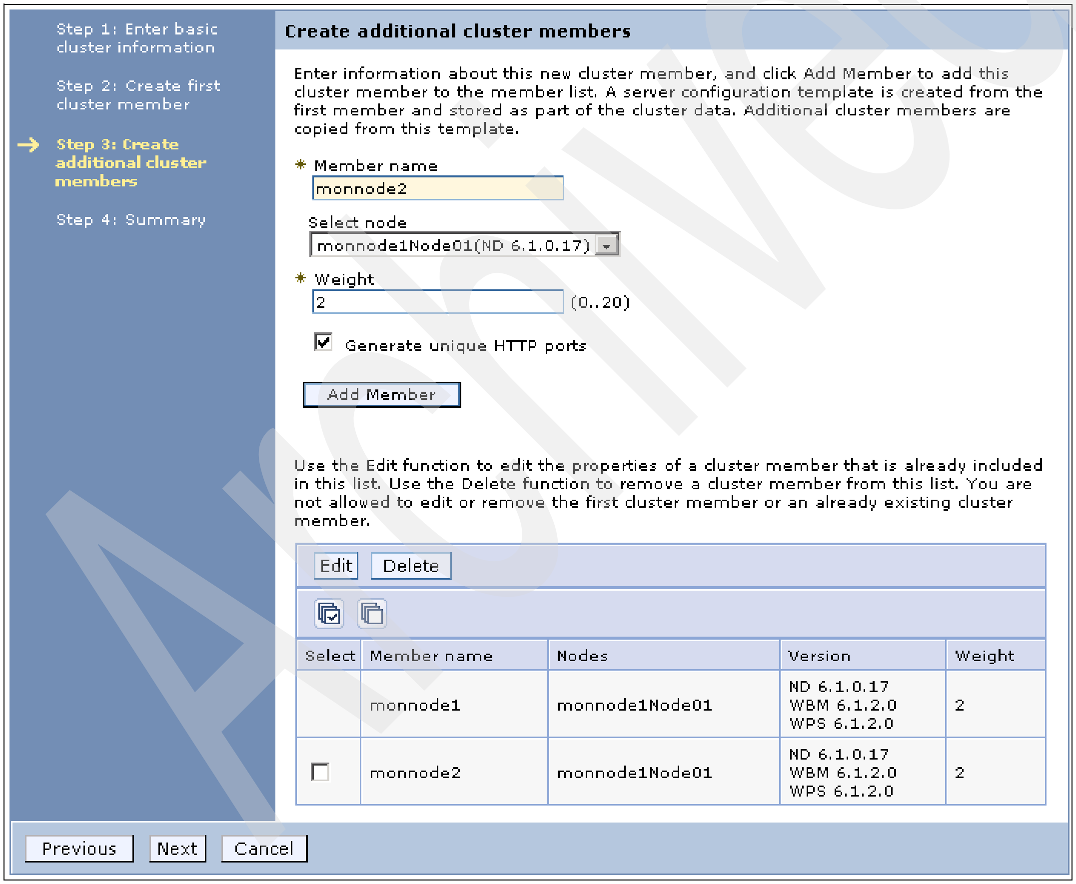


Figure 11-38 Step 3: Create additional cluster members

5. In step 4 (Figure 11-39), a summary of the cluster creation is displayed with all cluster member information. Click **Next**.

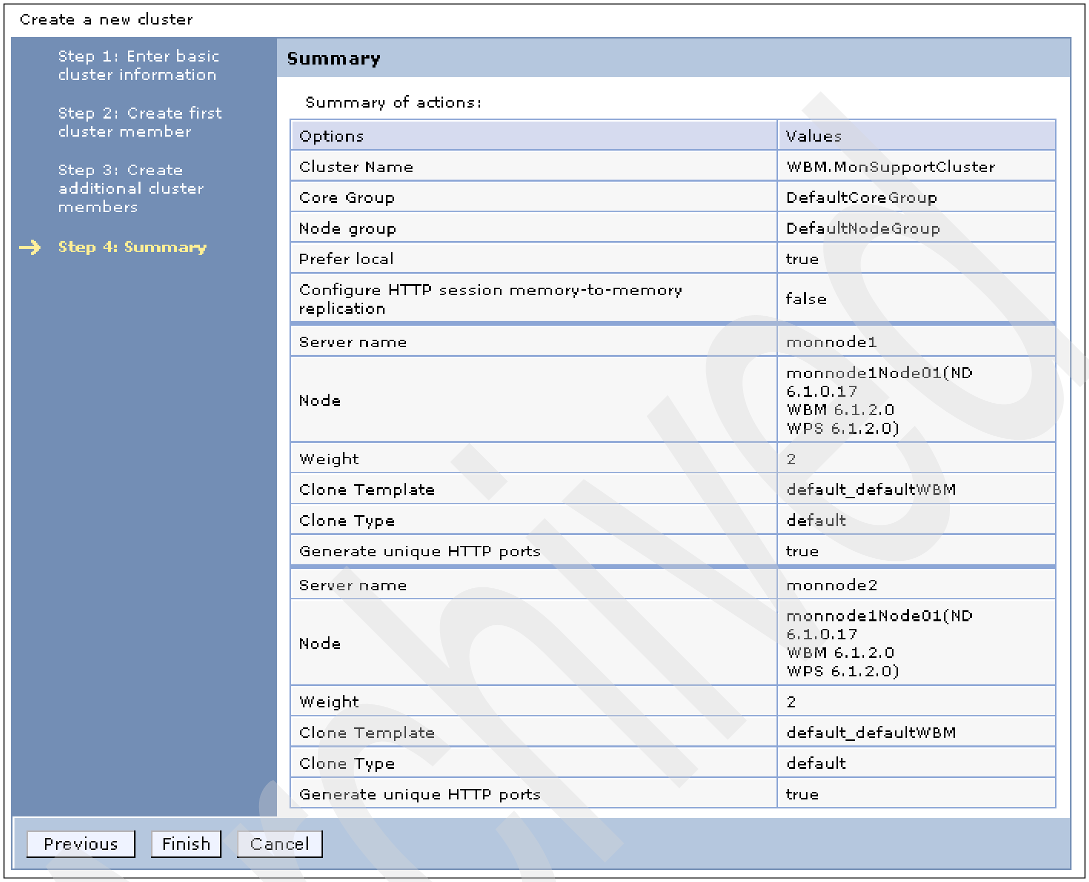


Figure 11-39 Step 4: Summary of cluster creation

Repeat these steps to create the Monitor Dashboard cluster, Monitor Moderator cluster and Monitor Model Logic cluster using the cluster names specified in Table 11-2 on page 387.

11.4.2 Enable CEI for the Monitor Moderator and Monitor Model Logic clusters

Perform the following steps for each member of Monitor Model Logic cluster to enable CEI:

1. Open the WebSphere Business Monitor deployment manager Integrated Solutions Console.
2. In the Integrated Solutions Console click **Servers** → **Application servers**. In the corresponding application server, click **Container Settings** → **Container Services** → **Common Event Infrastructure Service** (Figure 11-40).

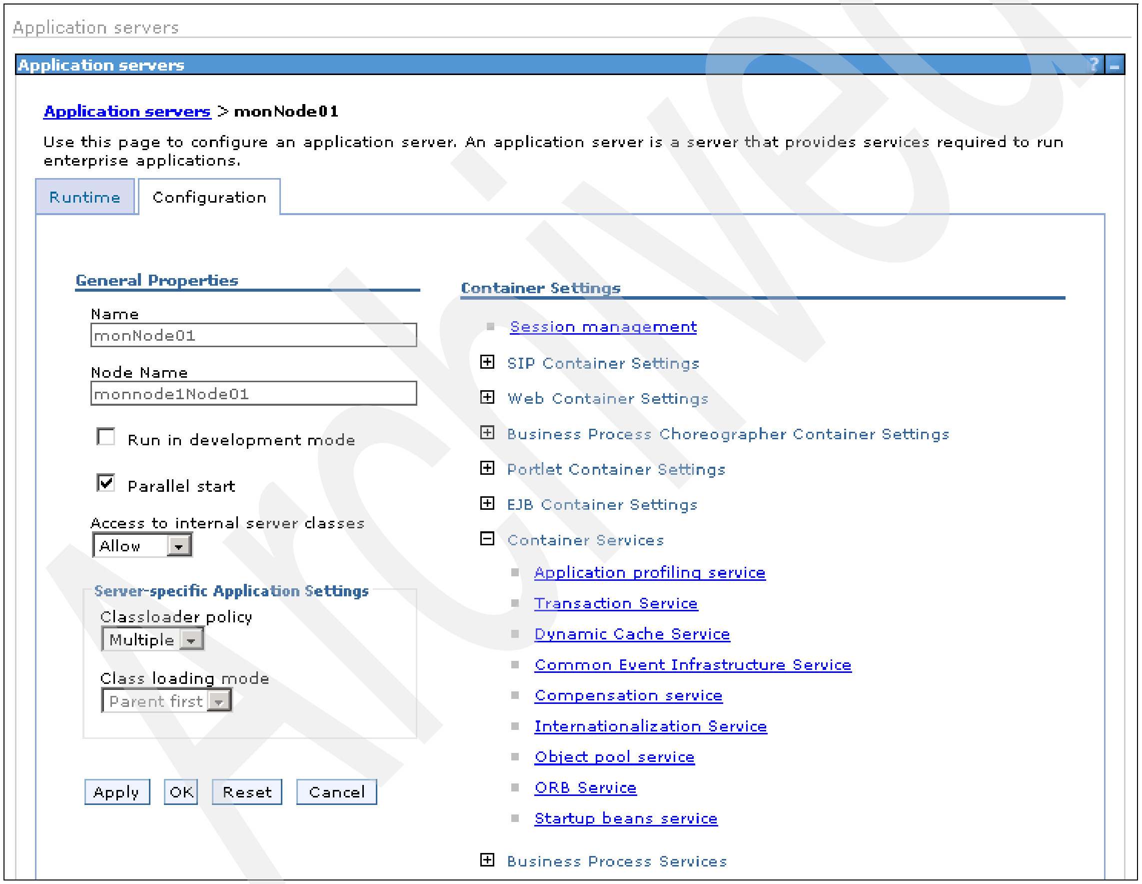


Figure 11-40 Common Event Infrastructure Service

3. Select the Enable service at server startup check box (Figure 11-41).

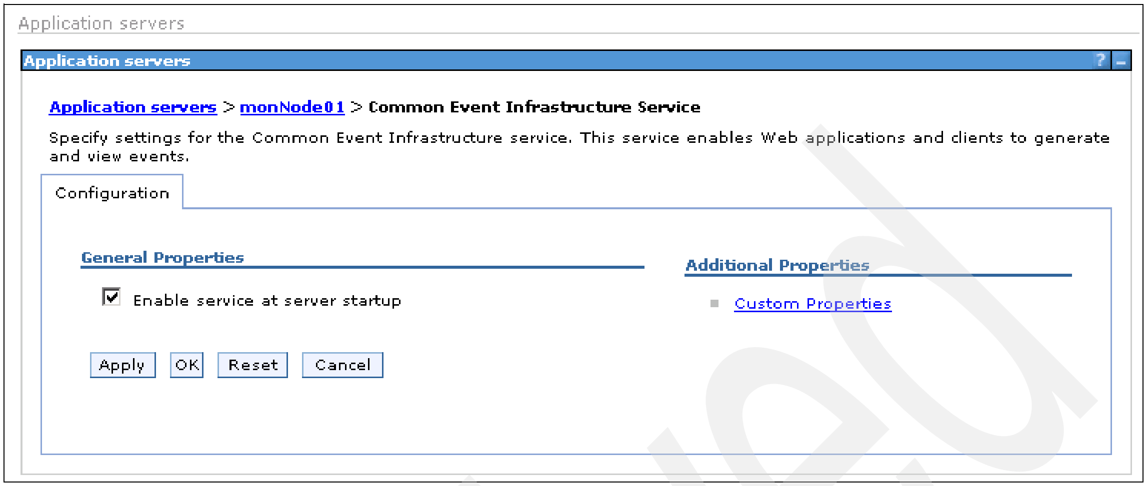


Figure 11-41 Enable event service

Repeat these steps for each cluster member in the Monitor Moderator cluster.

11.4.3 Creating the WebSphere Business Monitor bus

After creating the WebSphere Business Monitor clusters, it is a mandatory to create a Monitor service integration bus and its related artifacts, so that WebSphere Business Monitor can receive and send events.

Perform the following steps to create a Monitor bus:

1. In the WebSphere Business Monitor deployment manager, locate the script to create monitor bus, which is found at the following location:

   ```
   <DmgrInstallationDir>/scripts.wbm/sib/monitorSIBConfig.py
   ```

   DmgeInstallationDir is the installation folder of the WebSphere Business Monitor deployment manager.

2. Open a command line window and change to the following directory

   ```
   Monitor_deployment_manager profile/bin
   ```

3. Run the following command to execute the script as depicted in Figure 11-42.

   ```
   /wsadmin.sh -f ../scripts.wbm/sib/monitorSIBConfig.py
   ```

```
login as: root
Using keyboard-interactive authentication.
Password:
Last login: Thu Jul 31 16:05:24 2008 from 9.42.171.207
mondmgr:~ # cd /opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/
mondmgr:/opt/ibm/WebSphere/ProcServer/profiles/Dmgr01 # cd bin
mondmgr:/opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/bin # ./wsadmin.sh -f ../.
./../scripts.wbm/sib/monitorSIBConfig.py
```

Figure 11-42 Running monitorSIBConfig.py

4. Provide the script with the following information (Figure 11-43):
   - Cluster Name: The name of the Messaging Engine cluster. In our example this is RMSgold.Messaging.
   - The datastore: The type of datastore used by the bus (datastore or filestore) and the configuration parameters of the data source (user name, password, and JNDI name)

```
mondmgr:/opt/ibm/WebSphere/ProcServer/profiles/Dmgr01/bin # ./wsadmin.sh -f ../../../scripts.wb
WASX7209I: Connected to process "dmgr" on node CellManager01 using SOAP connector;  The type of

This script is being run in interactive mode.
Supply answers to the following questions.

Do you wish to add a server or cluster to the bus?
Enter 'server' or 'cluster' [server] : cluster

Clusters in this cell are:
RMSgold.AppTarget
RMSgold.Messaging
RMSgold.Support
WBM.MonLogic
WBM.MonSupport

Enter the name of the cluster you wish to add to the bus.
Hit enter to accept the default [RMSgold.AppTarget] : RMSgold.Messaging

Do you wish to use 'datastore' or 'filestore' for the messaging engine?
Hit enter to accept the default [datastore] : datastore

Enter the JNDI name of the datasource that will be used to access the database.
This datasource must already exist and be correctly configured to connect to the database.
jdbc/wbm/MonitorMEDatabase

Authentication Aliases in this Cell are :
  SCA_Auth_Alias
  WPSDB_Auth_Alias
  CommonEventInfrastructureJMSAuthAlias
  WPSCell01/RMSgold.Support/EventAuthDataAliasDB2
  CEIME_RMSgold.Messaging_Auth_Alias
  SCASYSME00_Auth_Alias
  SCAAPPME00_Auth_Alias
  BPCDB_RMSgold.AppTarget_Auth_Alias
  BPC_Auth_Alias
  BPCME_00_Auth_Alias
  BPCEDB_RMSgold.Support_Auth_Alias
  MonitorBusAuth
```

Figure 11-43 monitorSIBConfig.py interactive script

> **Note:** The bus related datastore information can be collected from the WebSphere Business Monitor Integrated Solutions Console at **Resources** → **JDBC drivers** → **Data sources**.

11.4.4 Creating the WebSphere Business Monitor event emitter factory

In this section we will create an event emitter factory for WebSphere Business Monitor. We will create this in the WebSphere Process Server Support cluster, as this cluster contains the CEI infrastructure.

Perform the following steps:

1. Open the WebSphere Business Monitor deployment manager Integrated Solutions Console.
2. In the Integrated Solutions Console click **Service integration** → **Common Event Infrastructure** → **Event emitter factories**.
3. Set the Scope to Cell as shown in Figure 11-44.

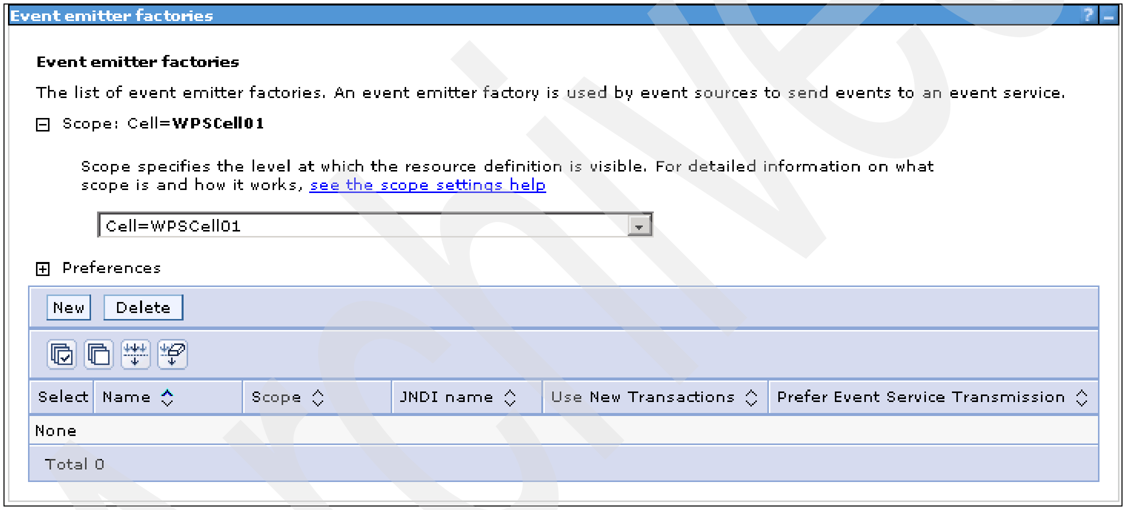


Figure 11-44 Event emitter factories page

4. Click **New**.
5. Perform the following steps in the Configuration tab (Figure 11-45 on page 397):
   a. Enter MonitorEventService as the event emitter factory name in the Name text box.
   b. Enter com/ibm/monitor/MonitorEmitterFactory as the JNDI name in the JNDI name text box.
   c. Click **OK**.

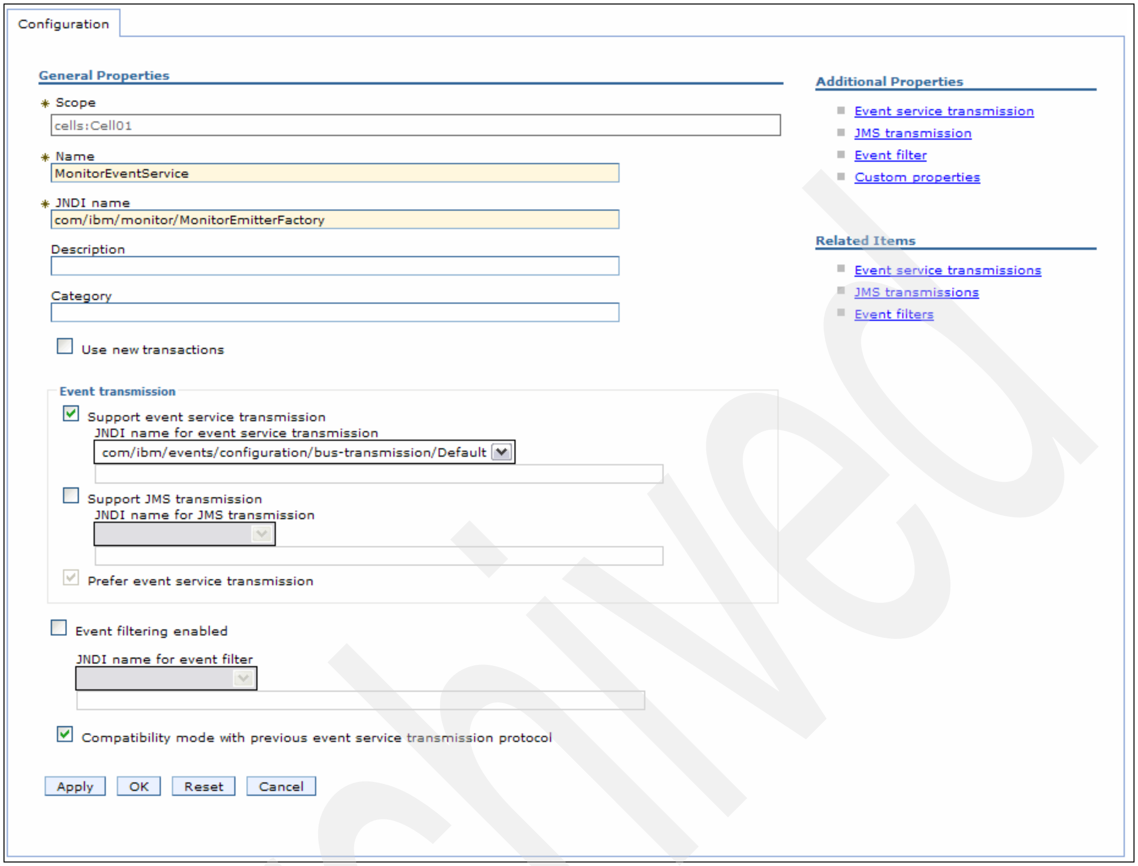


Figure 11-45 New event emitter factory page

6. Open the Event Emitter Factories page. Click the recently created event emitter factory.
7. In the Additional properties, click the **Event service transmission** link.

8. In General properties tab, perform the following steps (Figure 11-46):
   a. Enter a name for event service transmission in the Name text box.
   b. Type com/ibm/events/configuration/bus-transmission/Default as the JNDI name in the JNDI name text box.
   c. In Event service JNDI name section, select the Select an event service within this cell radio button.
   d. Click **OK**.

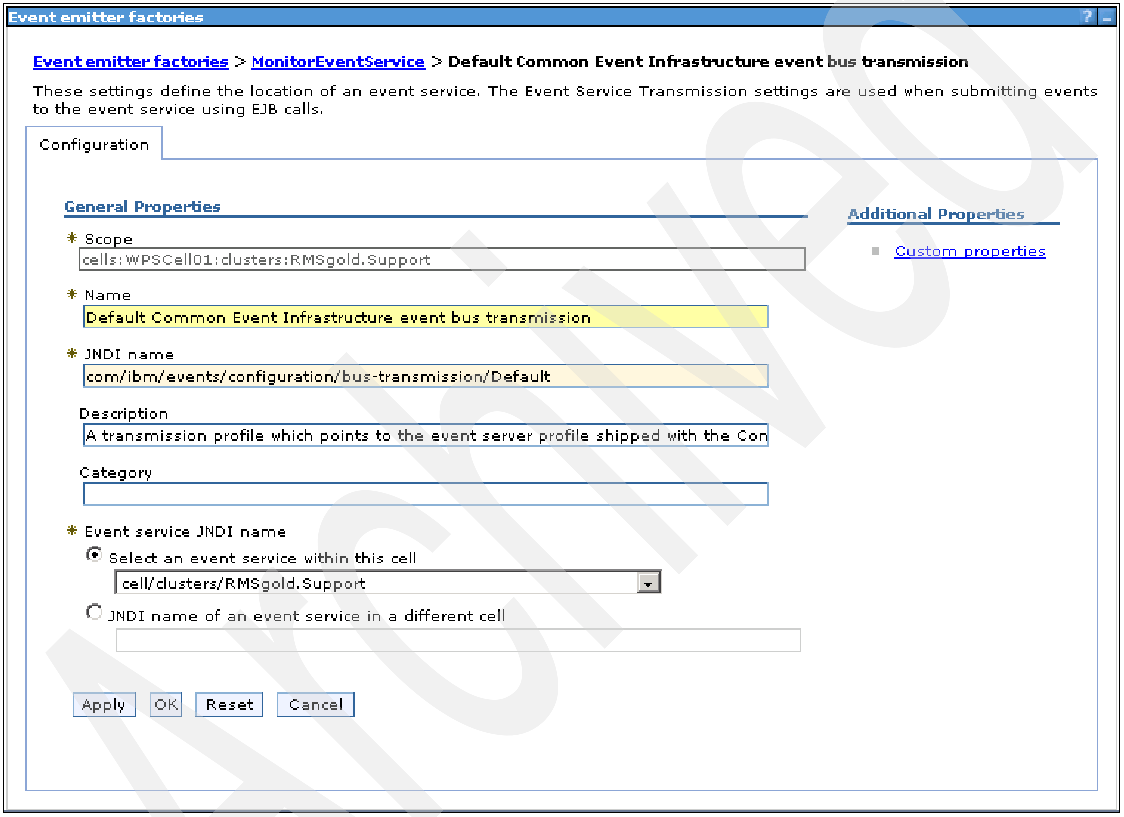


Figure 11-46 Event service transmission page

9. In the MonitorEventService page, check the Support event service transmission radio button.
10. Select the created event service transmission from the list.
11. Click **OK**.

11.4.5 Installing WebSphere Business Monitor applications

After installing and configuring the WebSphere Business Monitor clusters, you need to install the WebSphere Business Monitor applications into the clusters. For more information refer to Table 11-1 on page 387.

Installing applications into the Monitor Support cluster

There are two applications that should be installed on the Monitor Support cluster, as shown in Table 11-3. The location of the WebSphere Business Monitor applications are at <WPS_Installation_Folder>\installableApps.wbm

Table 11-3 Applications installed into the Monitor Support cluster

Application name	File name
Action manager service application	monactionmgr.ear
Database management service application	DmsService.ear

Perform the following instructions to install each application:

1. Open WebSphere Business Monitor Integrated Solutions Console.
2. Click **Applications** → **Enterprise Applications** and click **Install**.
3. Click **Browse** (Figure 11-47).

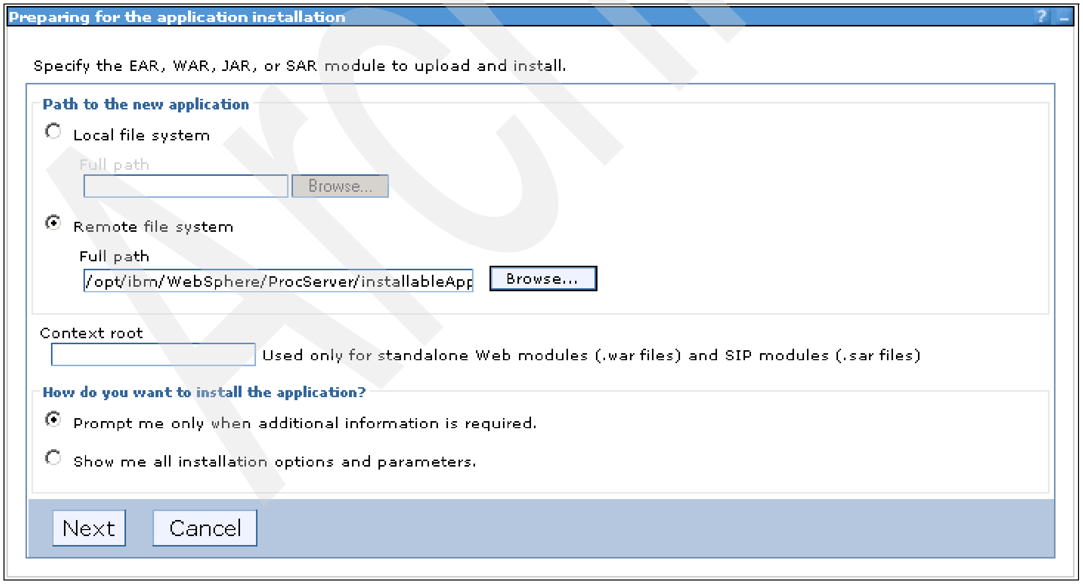


Figure 11-47 Install new application

4. Select the application EAR file (Figure 11-48). Install the action manager service application by selecting monactionmgr.ear. Click **OK** then **Next**.

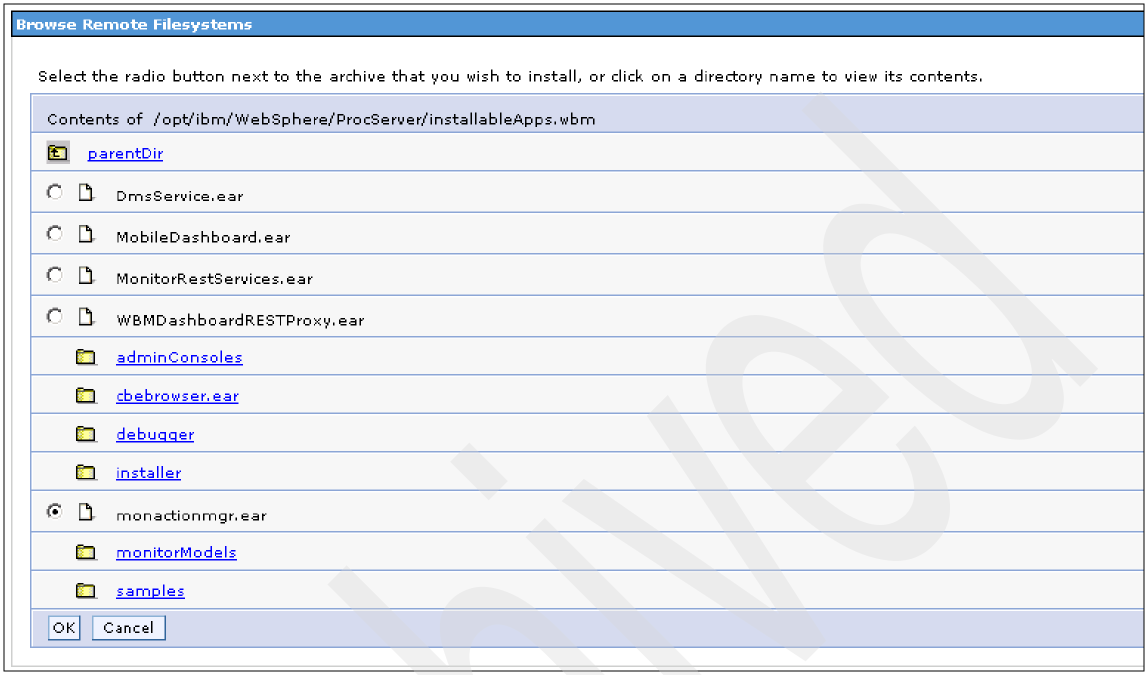


Figure 11-48 install application wizard: select application

5. In step 1 of the wizard (Figure 11-49) there are no changes required. Click **Next**.

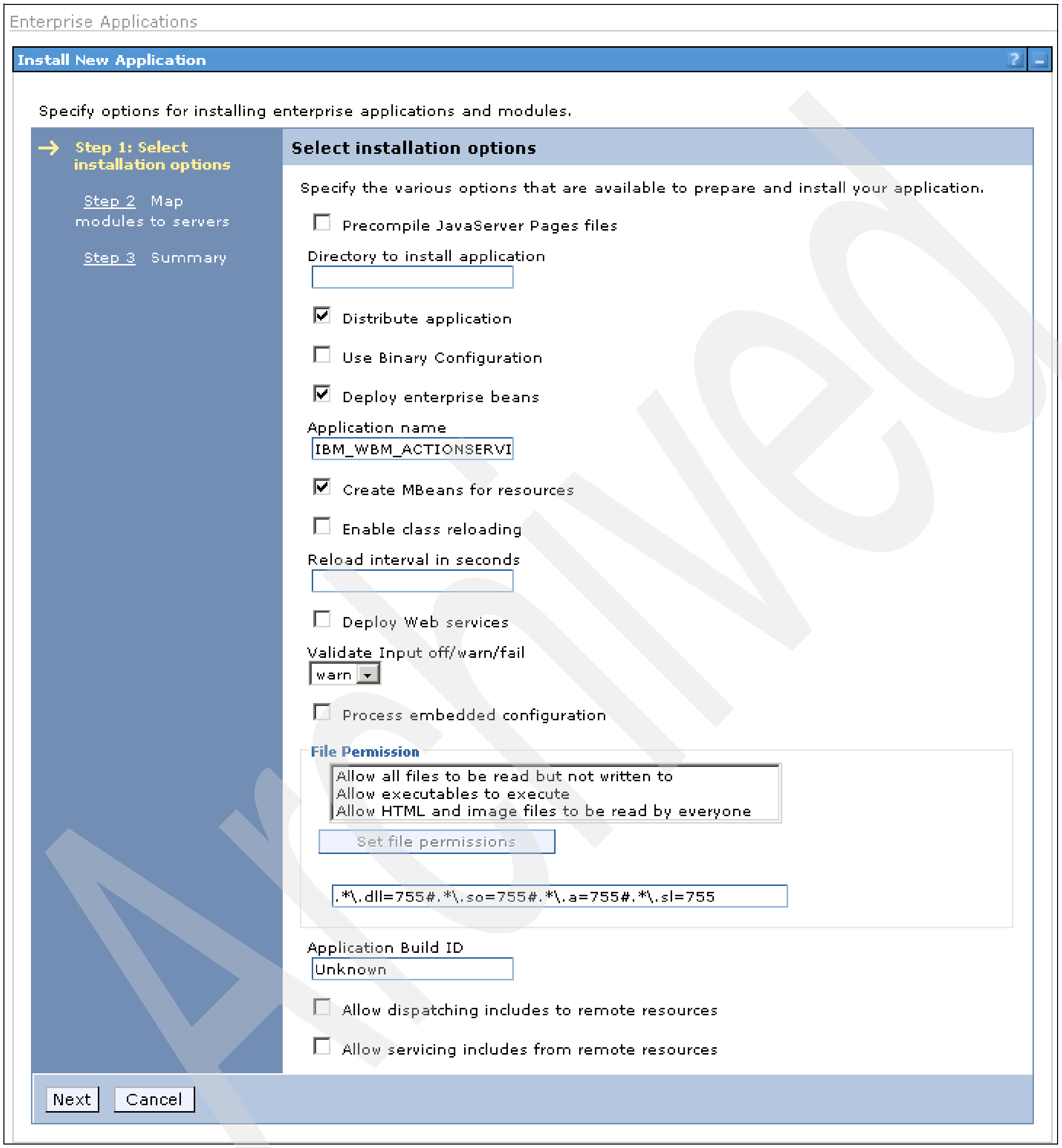


Figure 11-49 Install application wizard: Step 1

6. In step 2, perform the following steps (Figure 11-50):
   a. Select the Monitor Support cluster WBM.MonSupport from the Clusters and Servers list.
   b. Check the applications in the table.
   c. Click **Next**.

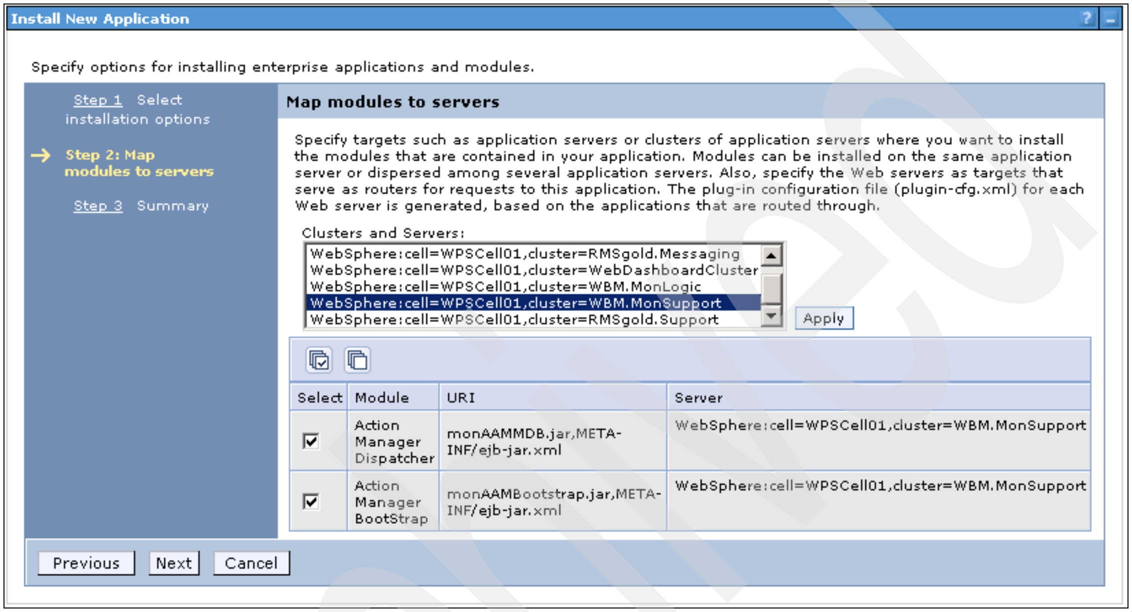


Figure 11-50 Install application wizard: Step 2

7. In step 3, a summary of the selected application is displayed (Figure 11-51). Click **Finish**.

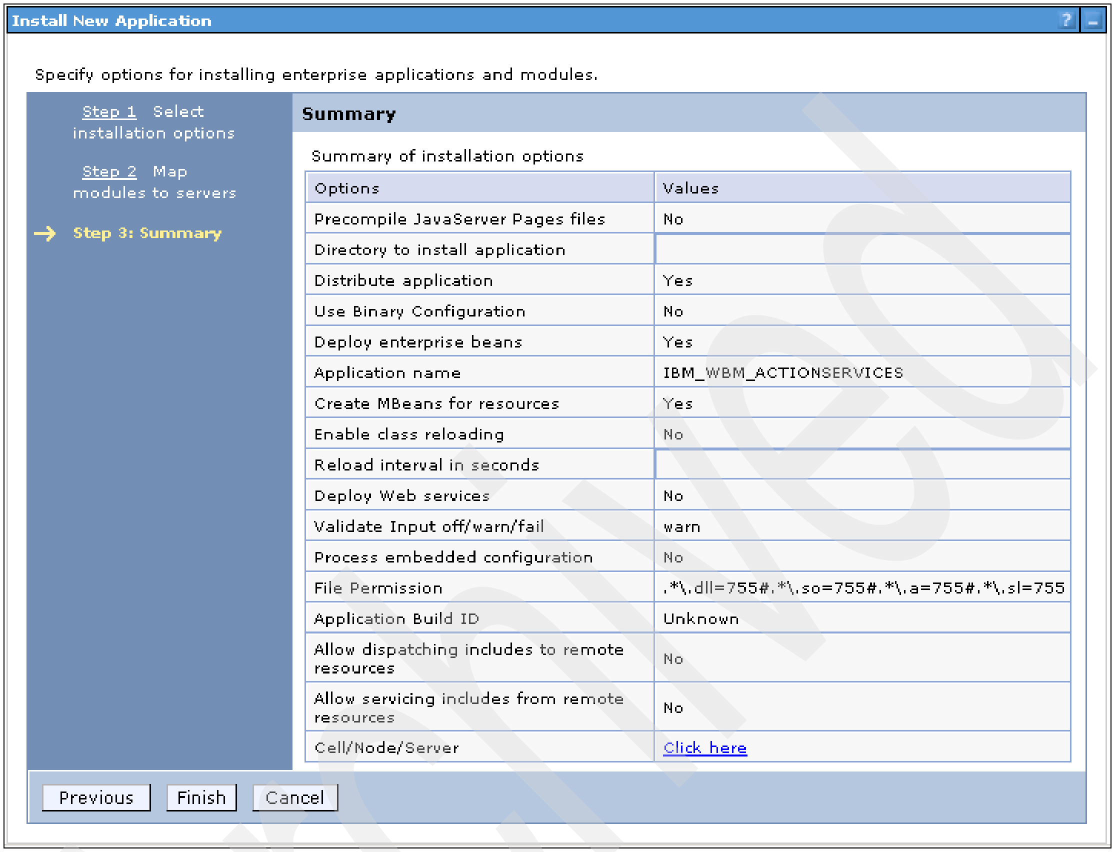


Figure 11-51 Install application wizard: step 3

> **Note:** The changes will not take effect until you restart the deployment manager and all servers.

Repeat the above instructions to install the database management service application for the Monitor support cluster as listed in Table 11-3 on page 399. After installing the applications, they should appear in the enterprise applications in Integrated Solutions Console (Figure 11-52).

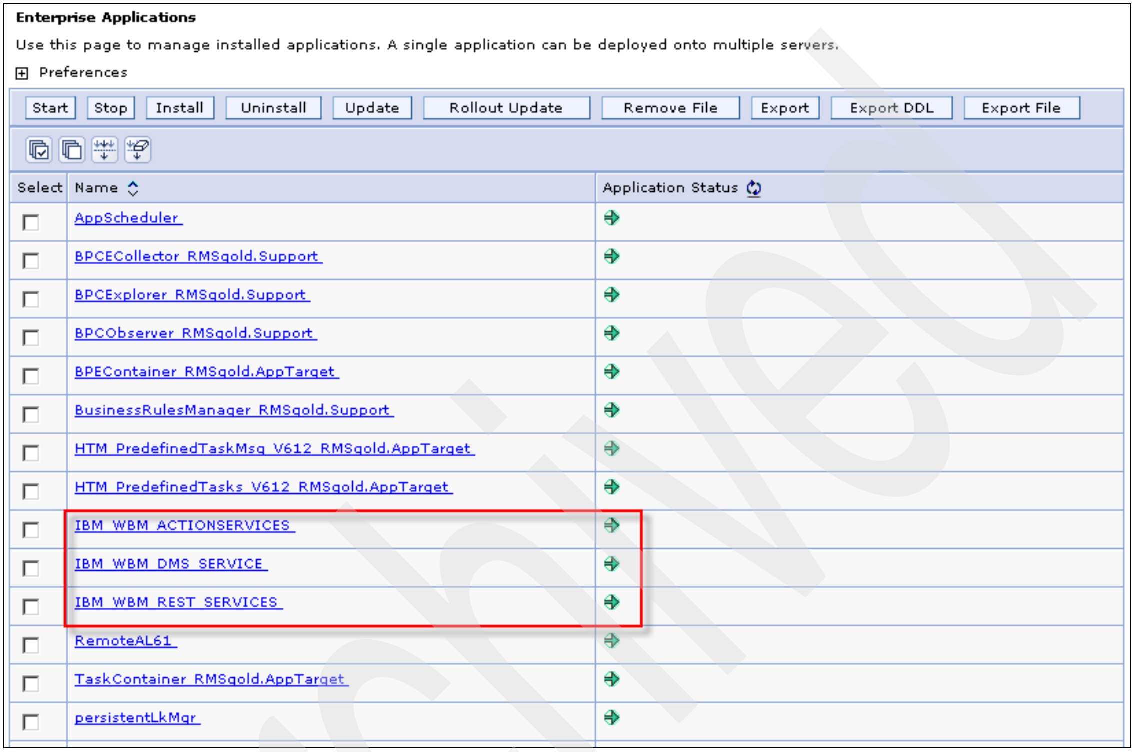


Figure 11-52 MonitorSupport applications

Creating the Monitor action services group profile

After installing the action manager services application, you must create an event group profile to enable WebSphere Business Monitor to send events against the defined business situations.

Complete the following steps to create the event group profile:

1. Open WebSphere Business Monitor Integrated Solutions Console.
2. In the navigation window, click **Service integration** → **Common Event Infrastructure** → **Event service**.
3. Under Additional Properties, click **Event services**.

4. Click the **Default Common Event Infrastructure event server** link (Figure 11-53).

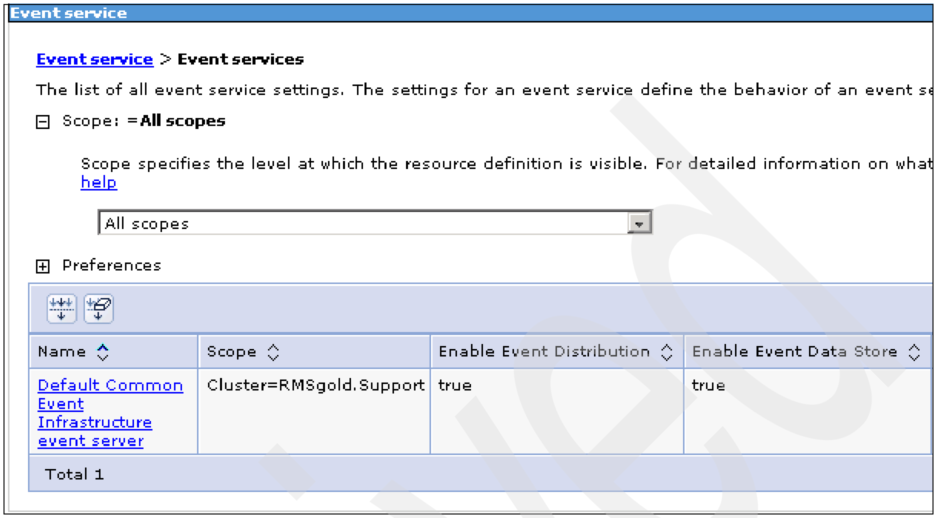


Figure 11-53 Event Services

5. Under Additional Properties, click **Event groups** (Figure 11-54).

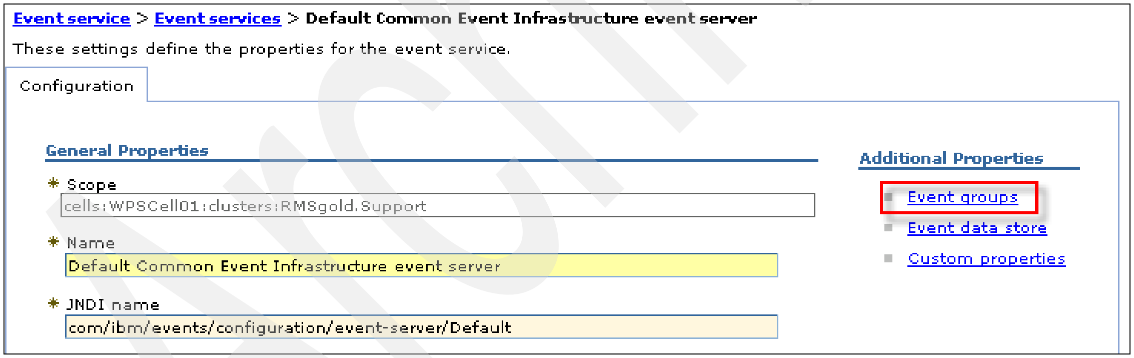


Figure 11-54 Event group link

6. In the Default CEI event server page (Figure 11-56), click **New**.

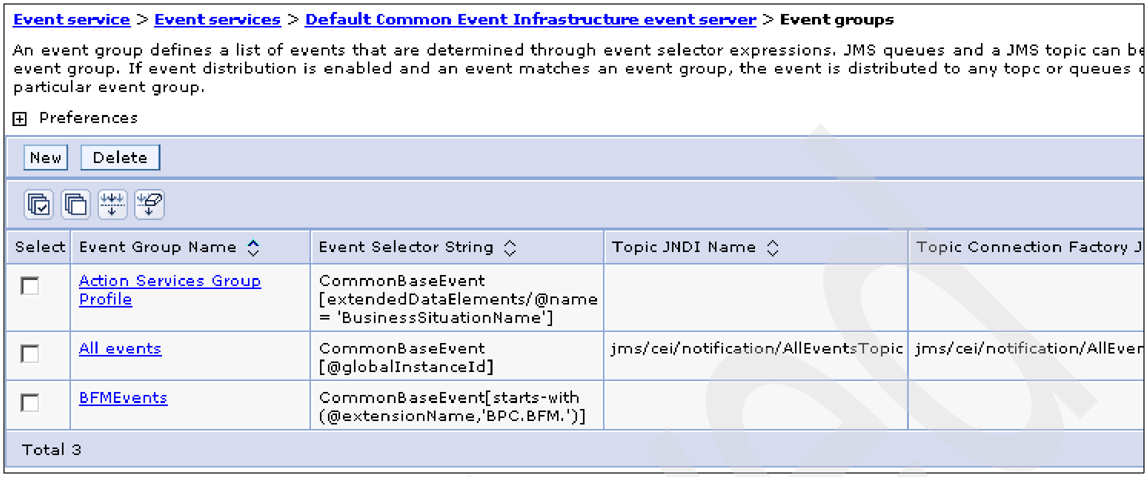


Figure 11-55 Event groups

7. In the configuration page (Figure 11-56), complete the following steps:

   a. Enter Action Services Group Profile for the Event group name text box.

   b. Enter CommonBaseEvent[extendedDataElements/@name = 'BusinessSituationName'] for the Event selector string text box. Click **Apply**.

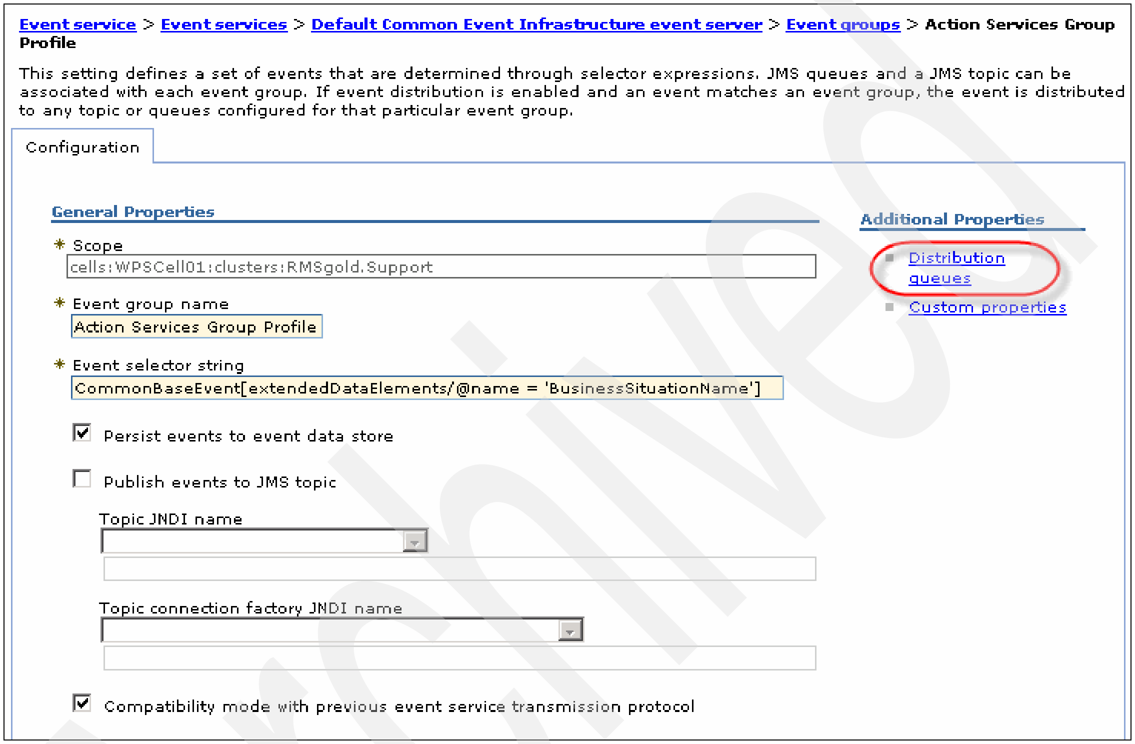


Figure 11-56 Event group configuration

8. Under Additional Properties, click the **Distribution queues** link.

9. Click **New**.

10. In the Configuration tab (Figure 11-57) complete the following:

a. Select jms/ActionManager/Queue from the Queue JNDI name drop-down list.

b. Select jms/ActionManager/QueueConnFactory from the Queue connection factory JNDI name list.

c. Click **Apply**.

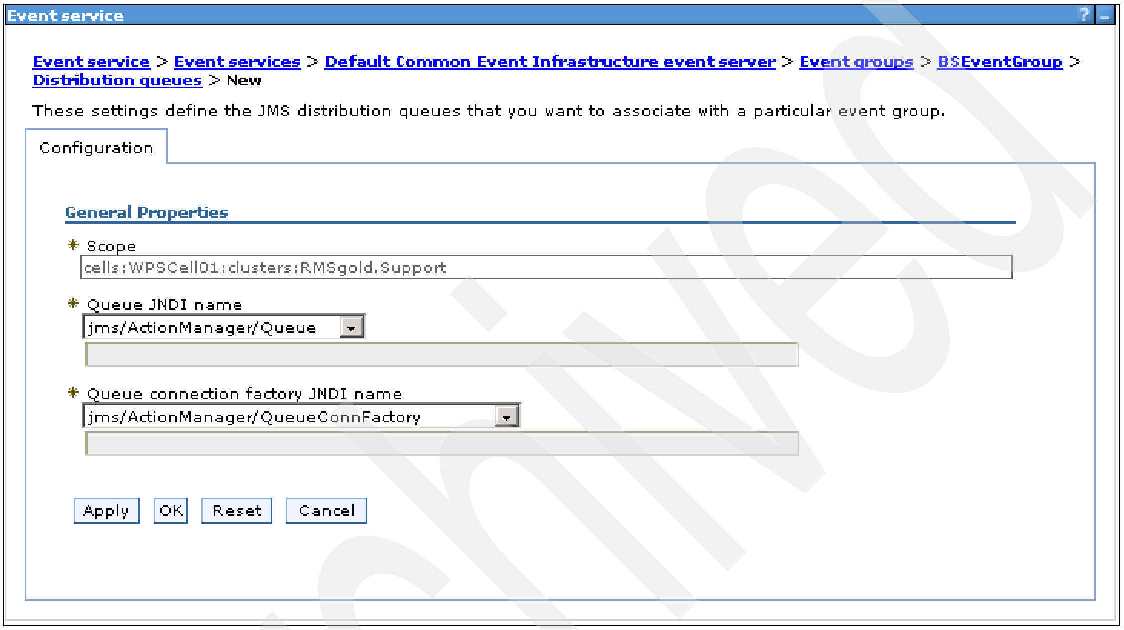


Figure 11-57

11. Click **Save** to save changes to master configuration.

Installing applications into the Monitor Dashboard cluster

There are three ways to view WebSphere Business Monitor dashboards:

- Using the Business Space application, where dashboards are widgets.
- Using WebSphere Portal Server and portlet dashboards.
- Using handheld devices.

  In this situation, you can view dashboards on smart mobile phone.

Four applications could be installed in the Monitor Dashboard cluster:

- Mobile dashboard application

  You should deploy this application if you are going to view dashboards on mobile devices.

- Dashboard PROXY services application

  This application should be installed if you are going to use portlet dashboards.

- Alphablox application

  This is used by Business Space.

- REST services application

Table 11-4 shows the file names of the applications that can be installed as EAR files into WebSphere Business Monitor.

Table 11-4 Applications that can be installed into the Monitor Dashboard cluster

Application name	File name
Dashboard PROXY services application	WBMDashboardRESTProxy.ear
Mobile dashboard application	MobileDashboard.ear
REST services application	MonitorRestServices.ear

In this installation scenario, we are using the Business Space application to view dashboards widgets. The steps for installing Business Space and the Alphablox application are described in Section 11.5, "Installing and configuring Dashboards and Business Space" on page 410.

If you need the ability to use portlet or mobile dashboards, install the appropriate EAR file listed in Table 11-4 into the Monitor Dashboard cluster.

11.5 Installing and configuring Dashboards and Business Space

WebSphere Business Monitor supports Business Space and you can view dashboards and views as widgets. In the following sections we provide details to install and configure a business dashboard widget in the Business Space application.

This topic contains the following sections:

- “Installing and configuring IBM Business Space for WebSphere”
- “Installing and configuring IBM Alphablox” on page 414
- “Configure Business Space for dashboard widgets” on page 440

11.5.1 Installing and configuring IBM Business Space for WebSphere

While installing WebSphere Application Server, the Business Space feature is enabled but is not installed or configured. To be able to use Business Space, we must install it first.

Installing Business Space

Complete the following steps to install the Business Space application

1. Open WebSphere Business Monitor Integrated Solutions Console.
2. Select **Servers** → **Clusters**.
3. Click **WebDashboardCluster** to install the Business Space application on this cluster.
4. Select the **Configuration** tab.

5. Under Business Integration section (Figure 11-58) click the **Business Space Configuration** link.

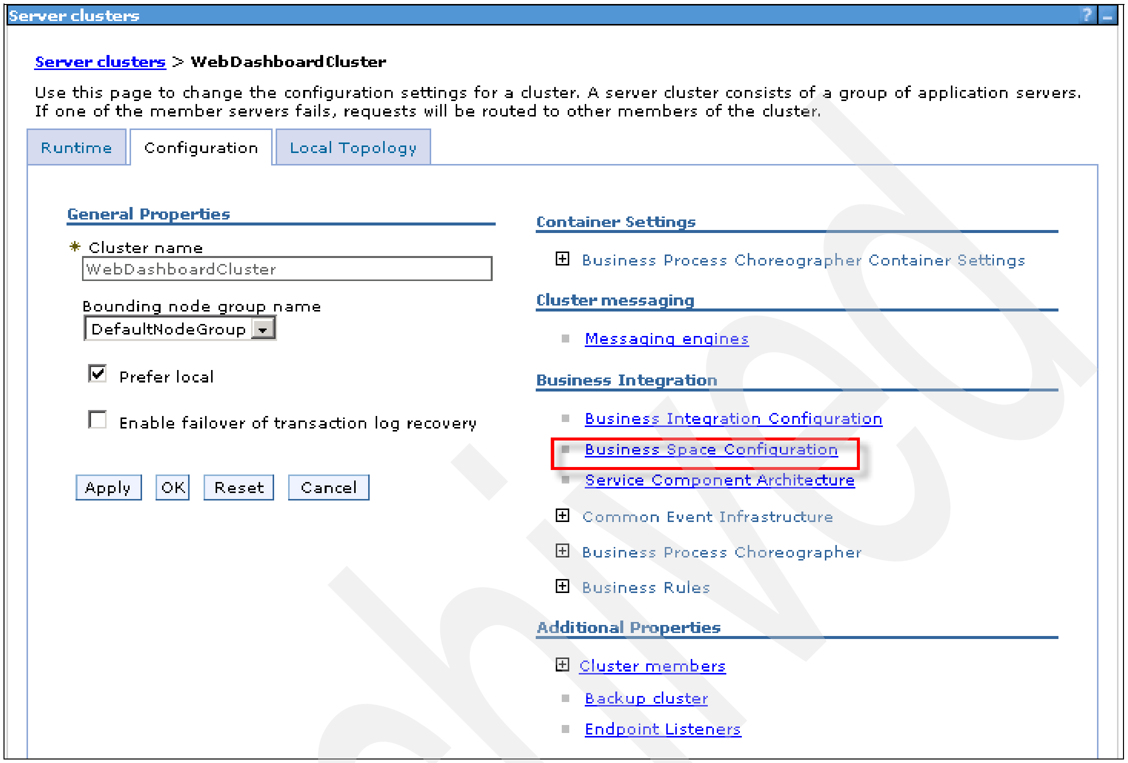


Figure 11-58 Business Space configuration link

6. In the Configuration tab (Figure 11-59), complete the following steps:
   a. Select the Install Business Space service check box.
   b. Type the required database schema name for Business Space database in the Database schema name text box.
   c. Select the target datasource to be used in configuring Business Space from the Create Business Space datasource using drop-down menu.
   d. Click **OK**.

> **Note:** You can create a new data source before installing Business Space or you can use the Monitor data source. Use jdbc/bpm/BusinessSpace as the JNDI name if you create a new data source.

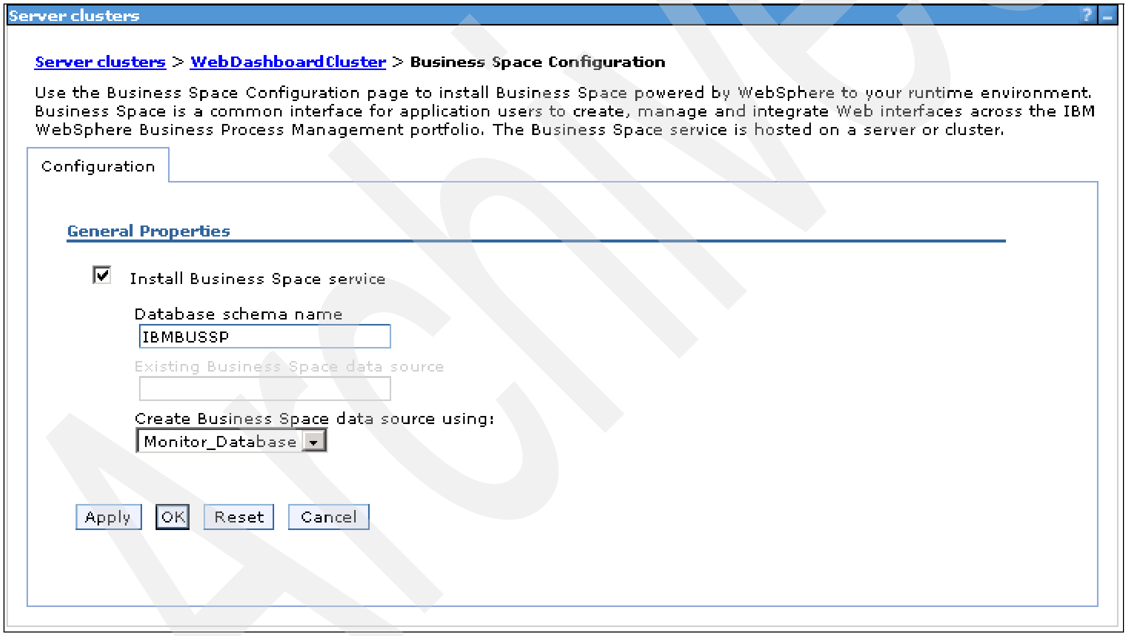


Figure 11-59 Business Space Installation

Creating Business Space database tables

You can use the MONITOR database to create the Business Space application tables, or you can create a new database. Complete the following to create Business Space database tables in the MONITOR database.

1. Open a DB2 command line window.
2. Switch the directory to the following file path:

   ```
   Monitor_installation_folder/BusinessSpace/wbm/dbscripts/DB2/createTa
   ble_BusinessSpace.sql
   ```

3. In the createTable_BusinessSpace.sql script, replace the @TSDIR@ (Table space directory) and @SCHEMA@ (Schema name) variables with the following values:
   – Change @TSDIR@ to IBMBPMBS
   – Change @SCHEMA@ to IBMBUSSP
4. Run the following commands (Figure 11-60):

   ```
   db2 connect to MONITOR
   db2 -tf createTable_BusinessSpace.sql
   ```

```
C:\Sources\logs>db2 connect to MONITOR

   Database Connection Information

 Database server        = DB2/NT 9.1.3
 SQL authorization ID   = SHAMSELD...
 Local database alias   = MONITOR


C:\Sources\logs>db2 -tf createTable_BusinessSpace.sql
DB20000I  The SQL command completed successfully.

DB20000I  The SQL command completed successfully.

DB20000I  The SQL command completed successfully.

DB20000I  The SQL command completed successfully.

DB20000I  The SQL command completed successfully.

DB20000I  The SQL command completed successfully.

DB20000I  The SQL command completed successfully.
```

Figure 11-60 Creating Business Space tables

5. Bind the command line interface to the Business Space database using the following commands:

```
db2 connect to database_name
db2 bind DB2_installation_directory\bnd\@db2cli.lst blocking all
grant public
db2 connect reset
```

   In the commands above, database_name represents the name of the Business Space database (for example: MONITOR), and DB2_installation_directory represents the directory where DB2 is installed.

6. Restart the servers.

> **Note:** If you use a new database for installing business space, you should perform the following steps:
>
> 1. Create the business space database.
> 2. Create a new JDBC provider for Business Space with the following information about the cluster level:
>   – Database type: DB2
>   – Provider Type: DB2 Universal JDBC Driver Provider
>   – Implementation type: XA data source
>   – Name: DB2 Universal JDBC Driver Provider (XA)
> 3. Create a new data source using the following information:
>   – Data source name: Business Space Datasource
>   – JNDI Name: jdbc/bpm/BusinessSpace
> 4. Configure Business Space for the newly created database.

11.5.2 Installing and configuring IBM Alphablox

To be able to view WebSphere Business Monitor dashboards in Business Space, it is mandatory to install IBM Alphablox on each member of the Monitor Dashboard cluster.

The installation can be done using command line or GUI. In this scenario, IBM Alphablox is installed using the command line.

> **Note:** It is recommended to install DB2 Alphablox on each cluster member sequentially, so that the installation options for the Alphablox cluster will appear in the wizard during installation.

Complete the following instructions to install DB2 Alphablox application for each cluster node:

1. Locate the IBM Alphablox software directory. Run the install.sh script to start the installation by entering the following command:

   ```
   ./install.bin
   ```

2. In the Choose Locale section (Figure 11-61), enter the required language number used to perform the installation and press enter.

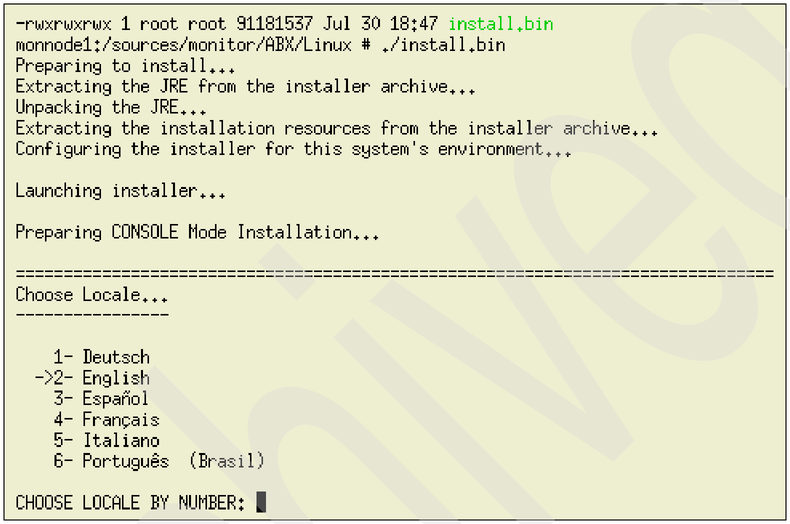


Figure 11-61 IBM Alphablox: welcome window

3. In the License Agreement section, enter 1 to accept the terms in the license agreement and press enter.

4. In the IBM Alphablox 9.5 installation section, press enter to continue if this is a new installation.

   > **Note:** If you are going to upgrade an existing IBM Alphablox installation, you should stop the IBM Alphablox application before proceeding in the upgrade operation.

5. In the Installation Location section (Figure 11-62), enter the required path in which the IBM Alphablox application will be installed.

```
===============================================================================
Installation Location
---------------------

Enter the Instance Name for IBM Alphablox (the default is AlphabloxAnalytics)
and choose the directory to which IBM Alphablox will be installed. If you
choose a directory in which an existing version of IBM Alphablox is installed,
the installation process will guide you through an upgrade to IBM Alphablox.

Destination Directory [/opt/Alphablox]::
```

Figure 11-62 IBM Alphablox: installation location

6. In the Server Instance Name section, enter the required instance name for the installation and press enter.
7. In the Select Installation Set section (Figure 11-63), enter the required installation type number. It is suggested to select a typical installation type.

```
===============================================================================
Select Installation Set
-----------------------

Please choose the Install Set to be installed by this installer.

    1- Compact
  ->2- Typical

    3- Customize...

ENTER THE NUMBER FOR THE INSTALL SET, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
   : 2
```

Figure 11-63 IBM Alphablox: Installation type

8. In the Choose Application Server section (Figure 11-64), select WebSphere as the required application server and press enter.

```
===============================================================================
Choose Application Server
-------------------------

Choose an Application Server to use with IBM Alphablox.

    1- Tomcat
  ->2- WebSphere

PLEASE SELECT ONE OF THE ITEMS, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
   : 2
```

Figure 11-64 IBM Alphablox: Application Server

9. A warning asking to shut down WebSphere Application Server appears. Ensure that WebSphere Application servers are stopped during installation and press enter.
10. In the WebSphere Root Directory section (Figure 11-65), enter the path of the WebSphere Business Monitor directory and press enter.

```
Enter the WebSphere Application Server Directory.
NOTE: For clustered servers, required applications must be manually deployed.
See post-installation steps in the Installation Guide.

WebSphere Root Directory []:: /opt/ibm/WebSphere/ProcServer
```

Figure 11-65 IBM Alphablox: WebSphere Application Server installation path

11. If you are going to perform the installation over cluster environment, enter Y to install to a cluster server (Figure 11-66) and press enter.

```
WebSphere Root Directory []:: /opt/ibm/WebSphere/ProcServer
Installing to a clustered server? (Y/N): Y
```

Figure 11-66 IBM Alphablox: Cluster environment

12. Select the required node in which IBM Alphablox will be installed by entering the corresponding number (Figure 11-67) and press enter.

```
Please select where to install the applications.
Node:
  ->1- CellManager01
    2- wpsNode01
    3- monnode2Node01
    4- monnode1Node01
    5- wpsNode02

PLEASE SELECT ONE OF THE ITEMS, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
   : 1
```

Figure 11-67 IBM Alphablox: selecting node

13. Select the profile in which IBM Alphablox will be installed (Figure 11-68).

> **Note:** This step appears if more than one server exists on the same node.

```
Please select where to install the applications.
Profile:
  ->1- Custom01
    2- MonServer01

PLEASE SELECT ONE OF THE ITEMS, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
   : 2
```

Figure 11-68 IBM Alphablox: profiles selection

14. In the port configuration section (Figure 11-69), keep the default values and press enter.

```
Verify the following details used to make administrative connections to the
WebSphere server.
HTTP Port: [wsadmin>]:: 9080
SOAP Connector Port: [8879]::
WebSphere Administrator Name: []:: wps
WebSphere Administrator Password::
```

Figure 11-69 IBM Alphablox: port configuration

15. In the Configure IBM Alphablox section (Figure 11-70), keep default values and press enter.

```
===============================================================================
Configure IBM Alphablox
-----------------------

Enter values for the following configuration settings or accept the defaults.

Telnet Console Port [20023]::
Server Log File Name [Server.log]::
Console Message Level
    1- DEBUG
    2- VERBOSE
  ->3- INFO
    4- SYSTEM
    5- WARNING
    6- ERROR
    7- FATAL

PLEASE SELECT ONE OF THE ITEMS, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
    :
SMTP Server []::
```

Figure 11-70 IBM Alphablox: Configure IBM Alphablox

16. In the Choose Java Location section (Figure 11-71), keep the default value of the Java path in the WebSphere Process Server installation folder. Then press enter.

```
===============================================================================
Choose Java Location
--------------------

Enter the directory where a JRE or JDK of at least version 1.4 is installed.

Java Directory [/opt/ibm/WebSphere/ProcServer/java]::



===============================================================================
Enable Additional Drivers
-------------------------

You may select a directory that contains additional drivers. The Alphablox lib
directory must not be used.

Do you want to enable additional drivers for IBM Alphablox? (Y/N)
   : Y
Location of Additional Drivers []:: /opt/ibm/WebSphere/ProcServer/universalDriver.wbm/lib

Driver Information:Enabled: DB2 Type 4, Derby

Is the driver information above correct? (Y/N): Y
```

Figure 11-71 IBM Alphablox: Choose Java Location

17. In the Enable Additional Drivers section (Figure 11-71), perform the following steps:

a. Add a new additional driver by entering Y to enable additional drivers for IBM Alphablox.

b. Press enter to continue.

c. Type the following path as the path of the additional driver

`/Monitor_Installation_folder/universalDriver.wbm/lib`

d. Press enter to continue.

e. Type Y to confirm that this driver is for DB2 type 4.

f. Press enter to continue.

18. In the Configure Repository section (Figure 11-72), perform the following steps:

   a. Select DB2 as the database type. Press enter to continue.

   b. Provide the following repository information:

      i. Enter the database server host name or IP in the Server text box.
      ii. Enter the database server port number in the Port text box.
      iii. Enter the database alias in the Alias text box.
      iv. Enter the database server user name in the Name text box.
      v. Enter the database server password in the Password text box.

   c. The wizard will test the connection using the provided information. Press enter to start connection testing.

```
===============================================================================
Configure Repository
--------------------

Select the repository type to use with this installation of IBM Alphablox. If
you are running IBM Alphablox in a clustered environment, you must select the
'Database Repository'.

Database
  ->1- DB2
    2- Derby

PLEASE SELECT ONE OF THE ITEMS, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
   : 1
Server []:: 9.42.171.207
Port []:: 50000
Alias []:: MONITOR
User []:: db2admin
Password:
The installer will now run a database connection test using the information
that you have provided
PRESS <ENTER> TO CONTINUE:
```

Figure 11-72 IBM Alphablox: repository configuration

19. The test connection result will be displayed. Press enter to continue the installation.

> **Note:** If there is problem in testing the connection, the wizard will repeat the repository configuration section, asking you to provide the correct configuration.

20. In the Configure Cluster section (Figure 11-73), provide information relating to the IBM Alphablox cluster. To enable clustering, type `1` and press enter to continue. Provide the following configuration information for the cluster:

   – Cluster port number
   – Cluster subnet mask

```
===============================================================================
Configure Clustering
--------------------

Enter the configuration information for the clustering settings in your
configuration.

    1- Yes
  ->2- No

Enable Alphablox Clustering: 1
Cluster Port Number (DEFAULT: 7855):
Cluster Subnet Mask (DEFAULT: 255.255.255.0):
```

Figure 11-73 IBM Alphablox: configure cluster

21. In the Configure Conversion Utility section there are a number of choices. If this is the first node in which the IBM Alphablox application will be installed, keep the default values and press enter to continue. Table 11-5 provides the required configuration options.

Table 11-5 Configure conversion properties.

Property	At first node	At other nodes
Conversion Operation	Copy	Copy
Move Server Properties	All	All
Existing Repository tables	N/A	Update
User Defined DDL schema file	No	No

22. Press enter in the Pre-Installation Summary window (Figure 11-74 on page 422) to start installation.

```
================================================================================
Pre-Installation Summary
------------------------

Please Review the Following Before Continuing:

Install Set
    Typical

Product Components:
    Core,
    Tools,
    FastForward,
    Examples,
    Relational Reporting,
    Applications,
    IBM Alphablox,
    Query Builder

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:  6.1.0.17
    WebSphere Start File:  setupCmdLine.sh
    WebSphere Cluster Install:  true
    WebSphere Profile:  Dmgr01
    WebSphere Cell:  WPSCell01
    WebSphere Node:  CellManager01
    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/lib
DB2 Driver Type:  4
Drivers:  Enabled: DB2 Type 4, Derby
Repository Type:  Database
    Database Type:  DB2
    Database Server:  9.42.171.207
    Database Port:  50000
    Database Alias:  MONITOR
    Database User:  db2admin

CLUSTERING: Enabled:
    Port::  7855
    Subnet Mask::  255.255.255.0

Repository Conversion Utility:
    Operation:  Copy
    Existing Tables:

Disk Space Information (for Installation Target):
    Required:  162,950,729
    Available:  924,033,024




PRESS <ENTER> TO CONTINUE:
```

Figure 11-74 IBM Alphablox: pre-installation summary

23. The IBM Alphablox 9.5 Installation Complete section displays the status of the installation. If there is a problem, check the installation logs mentioned in the installation status. Press enter to exit installation.

24. Deploy the IBM Alphablox libraries using the DeployWebSphereLibraries.sh utility. Change the current directory to the following path:

```
<AlphabloxInstallationDir>/tools/was_shared_lib
```

25. Run the following command to deploy the libraries:

```
DeployWebSphereLibraries.sh -conntype SOAP -username DM_user
-password DM_password.
```

The utility launches an interactive script.

26. Type 1 to install libraries (Figure 11-75).

```
monnode1:/opt/Alphablox/tools/was_shared_lib #
monnode1:/opt/Alphablox/tools/was_shared_lib # ./DeployWebSphereLibraries -connt
ype SOAP -username wps  -password passw0rd
Please select from one of the following options

1) Install libraries
2) Uninstall libraries
3) Search for installed libraries
4) Toggle trace
---------------------
5) Exit

Select (1-5):1
```

Figure 11-75   Install libraries

27. Select the required level in which you want to install (Figure 11-76). We selected cluster.

```
Select the level you wish to deploy the libraries.

1) Cluster
2) Cell
3) Node
4) Server
---------------------
5) Back

Select (1-5):1
```

Figure 11-76   Select installation level

28. If you selected the Cluster option in the previous step, the interactive script displays the available clusters to select the required one. Select the Monitor Dashboard cluster. Press enter to continue (Figure 11-77).

```
Select (1-5):1
Please Wait...
Please select the Cluster you wish to target.
1*) RMSgold.Support(cells/WPSCell01/clusters/RMSgold.Support|cluster.xml#ServerC
luster_1217012828347)
2) WebDashboardCluster(cells/WPSCell01/clusters/WebDashboardCluster|cluster.xml#
ServerCluster_1217967271939)
3) WBM.MonLogic(cells/WPSCell01/clusters/WBM.MonLogic|cluster.xml#ServerCluster_
1217540234330)
4) WBM.MonSupport(cells/WPSCell01/clusters/WBM.MonSupport|cluster.xml#ServerClus
ter_1217540609873)
5) RMSgold.Messaging(cells/WPSCell01/clusters/RMSgold.Messaging|cluster.xml#Serv
erCluster_1217012830555)
6) RMSgold.AppTarget(cells/WPSCell01/clusters/RMSgold.AppTarget|cluster.xml#Serv
erCluster_1217012821425)
---------------------
7) Back

Select (1-7):2
```

Figure 11-77 Select cluster

29. A list of available members on this cluster is displayed. Press enter to start deployment (Figure 11-78).

```
Select (1-7):2
The following servers have been found:
monDash01(cells/WPSCell01/nodes/monnode1Node01/servers/monDash01|server.xml#Serv
er_1217967278794)
monDash02(cells/WPSCell01/nodes/monnode2Node01/servers/monDash02|server.xml#Serv
er_1217967280466)
Do you wish to continue?[Y/N]:
```

Figure 11-78 Start installation

30. Make sure the deployment operation is successful (Figure 11-79).After you see a message indicating that the libraries were successfully installed, you can select 5) Back and 5) Exit to stop the utility.

```
Processing object monDash01(cells/WPSCell01/nodes/monnode1Node01/servers/monDash
01|server.xml#Server_1217967278794)...
Libraries successfully installed!
Processing object monDash02(cells/WPSCell01/nodes/monnode2Node01/servers/monDash
02|server.xml#Server_1217967280466)...
Libraries successfully installed!
```

Figure 11-79 Installation complete summary

31. Start WebSphere Business Monitor deployment manager and clusters to finalize the installation of IBM Alphablox. To finalize the IBM Alphablox installation, two applications should be installed on the Monitor Dashboard cluster:

   – AlphabloxPlatform.ear
   – ApplicationStudio.ear

32. Open the WebSphere Business Monitor Integrated Solutions Console. Go to **Applications** → **Enterprise Applications** and click **Install**.

33. Use the remote file system path setting to browse through the network to locate the AlphabloxPlatform.ear file (Figure 11-80). Browse to <AlphabloxInstallationDir>/installableApps/AlphabloxPlatform.ear

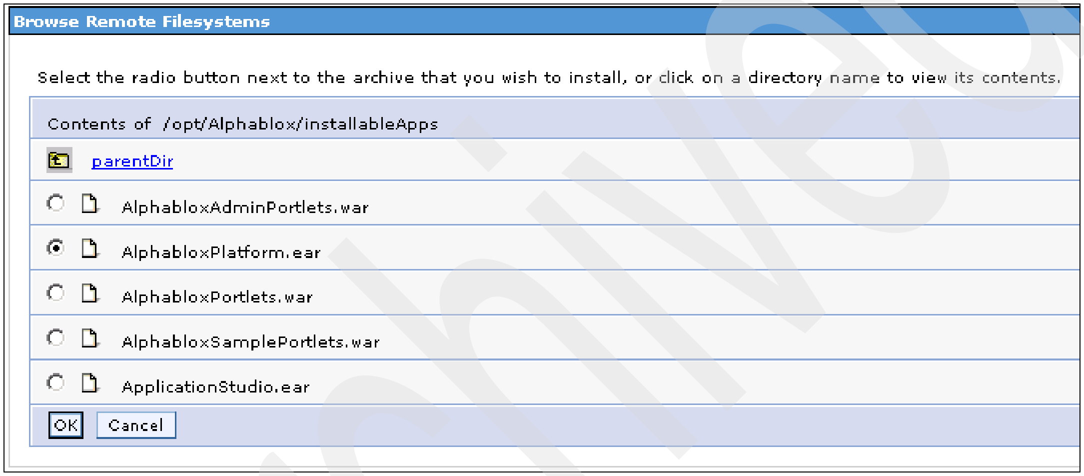


Figure 11-80 Select ear file window

34. Select the Show me all installation options and parameters radio button (Figure 11-81). Then click **Next**.

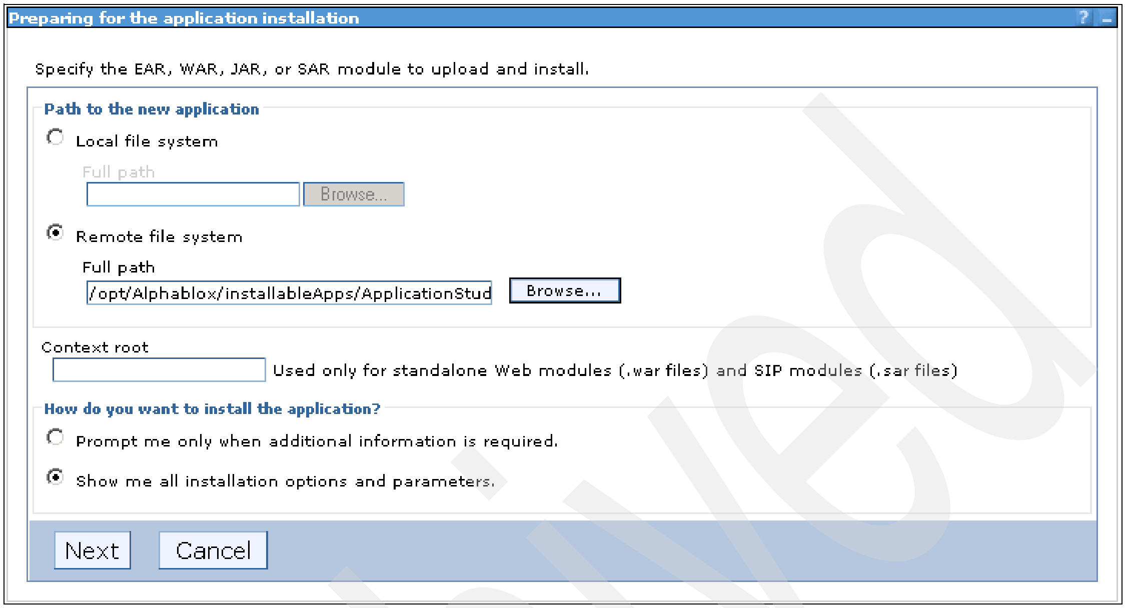


Figure 11-81 Prepare for installation

35. In the Default Bindings Options window (Figure 11-82), leave the default settings, unless you require different bindings. Click **Next**.

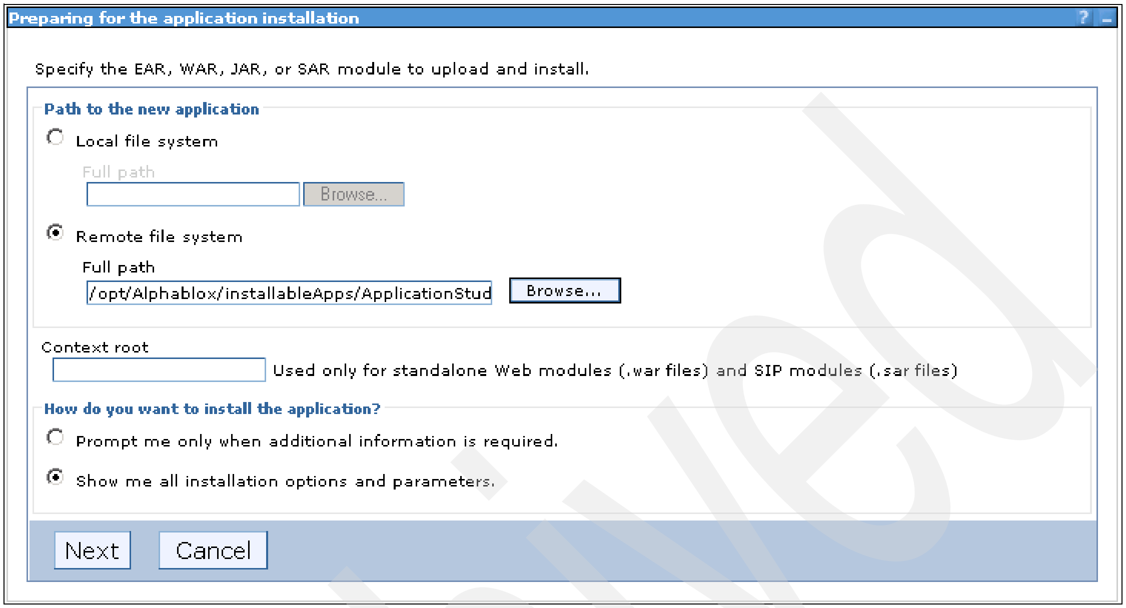


Figure 11-82 Binding options

36. In the Step 1: Select installation options page, keep the default values and press **Next** (Figure 11-83).

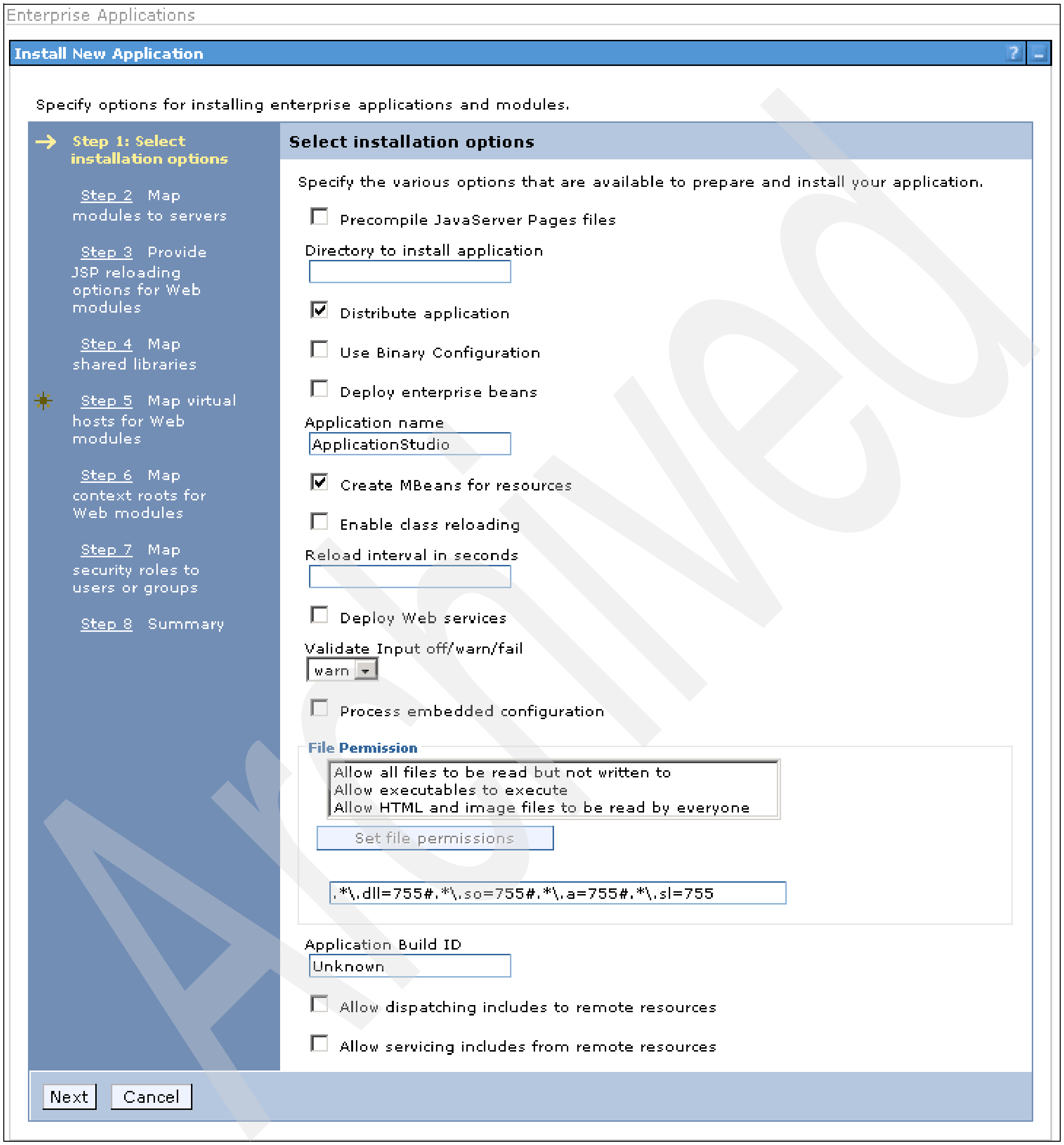


Figure 11-83 IBM Alphablox installation: Step 1

37. In the Step 2: Map modules to servers window (Figure 11-84), map the IBM Alphablox modules to the Monitor Dashboard cluster by selecting the modules and the appropriate cluster. Click **Apply**, then click **Next**.

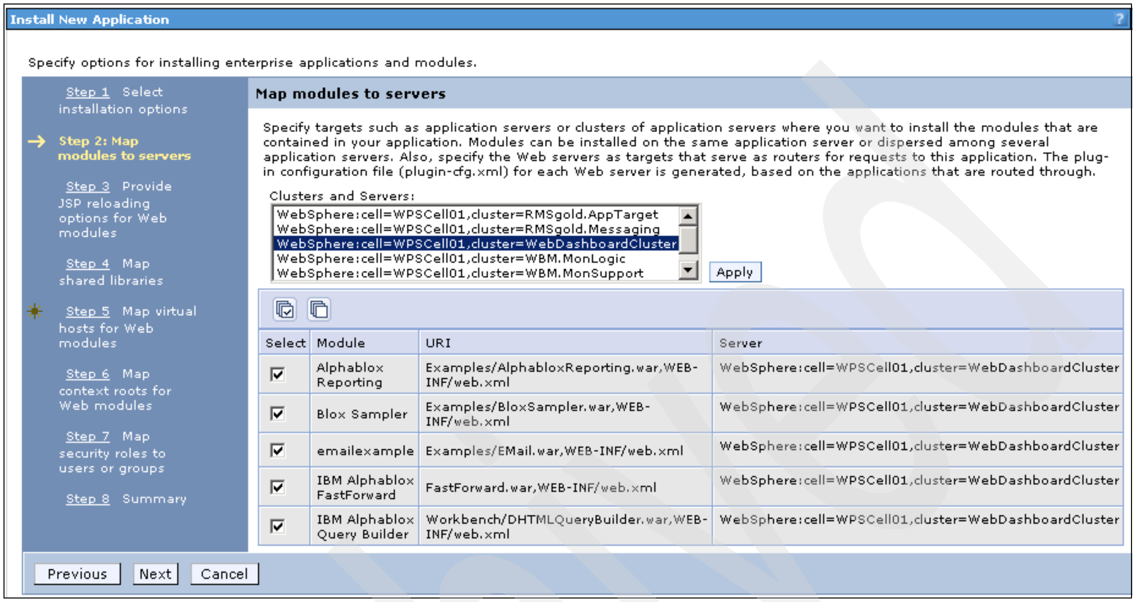


Figure 11-84 IBM Alphablox installation: Step 2

38.In the Step 3: Provide JSP reloading options for web modules window (Figure 11-85) keep the default values and press **Next**.

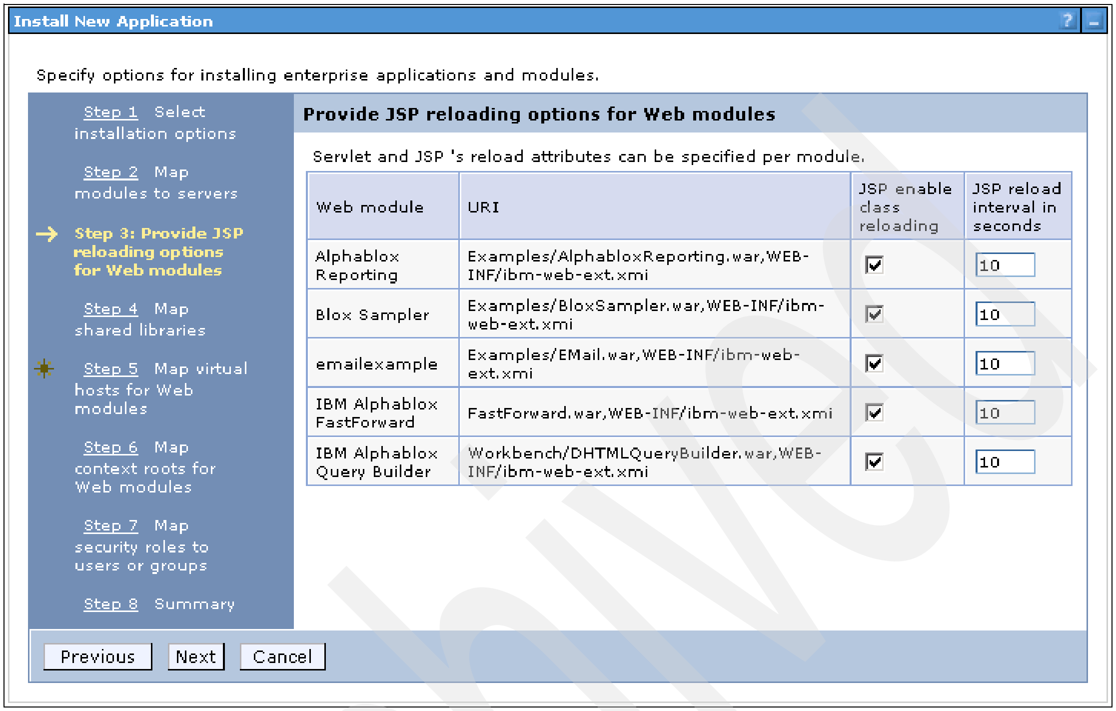


Figure 11-85 IBM Alphablox installation: Step 3

39. In the Step 4: Map shared library window (Figure 11-86), keep the default values and press **Next**.

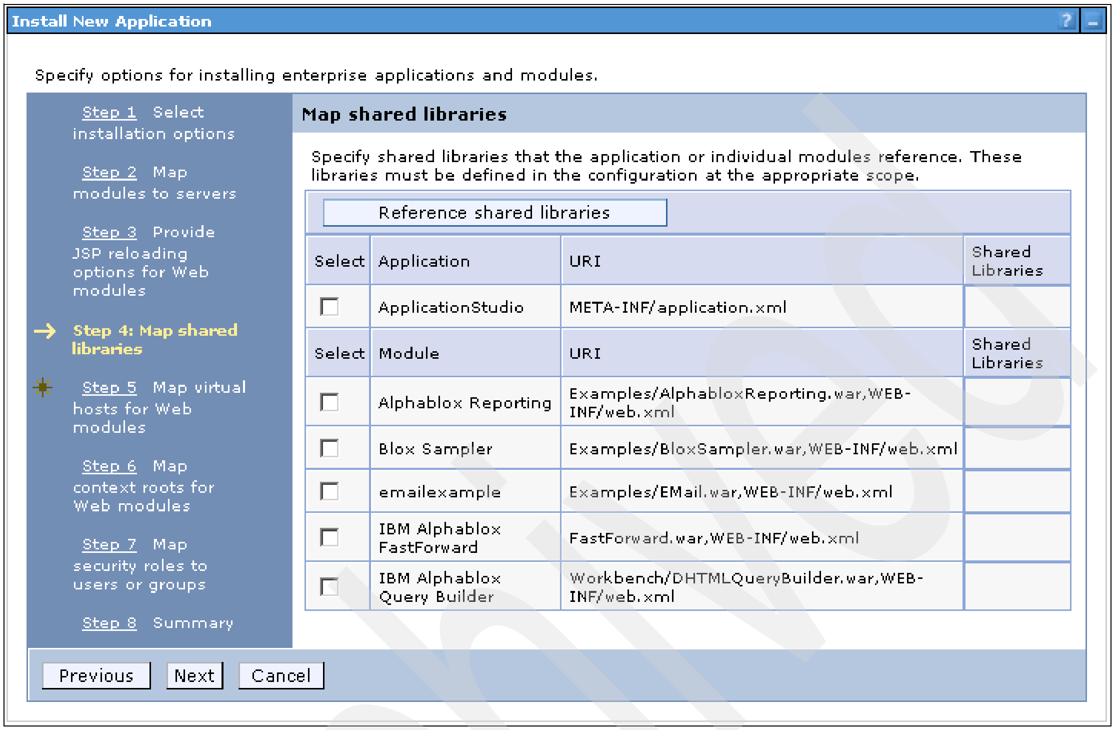


Figure 11-86 IBM Alphablox installation: Step 4

40. In the Step 5: Map virtual hosts for web modules window (Figure 11-87) keep the default values and press **Next**.

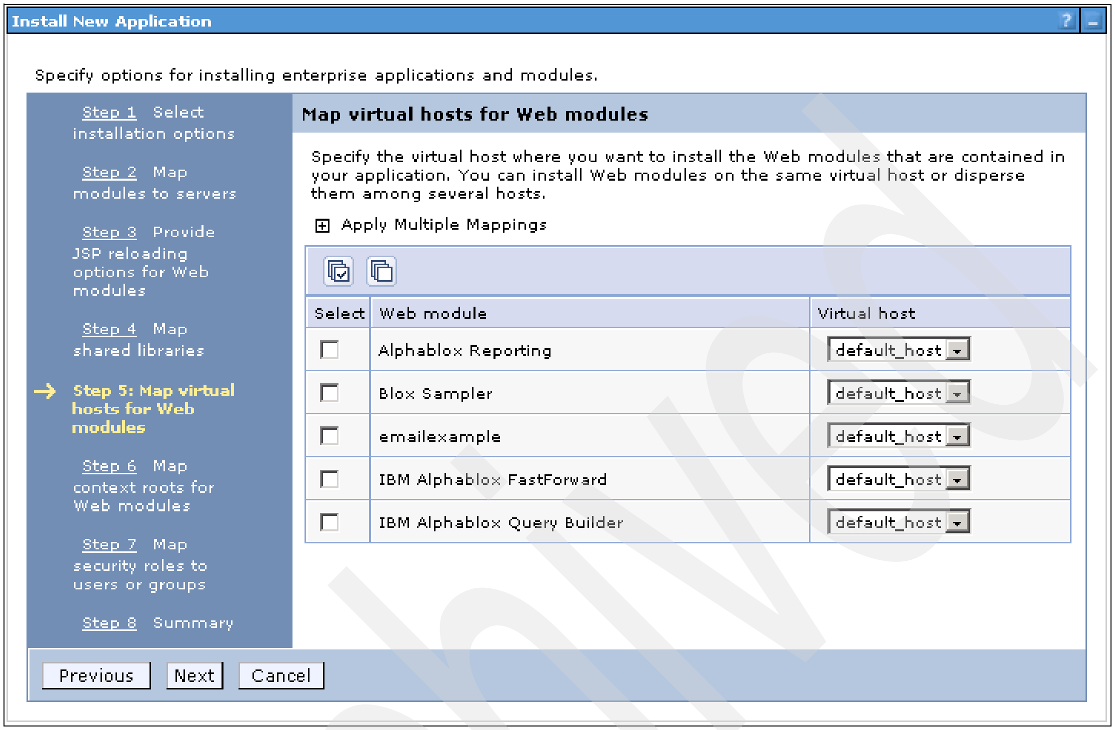


Figure 11-87 IBM Alphablox installation: Step 5

41. In the Step 6: Map context roots for web modules window (Figure 11-88). Keep the default values and press **Next**.

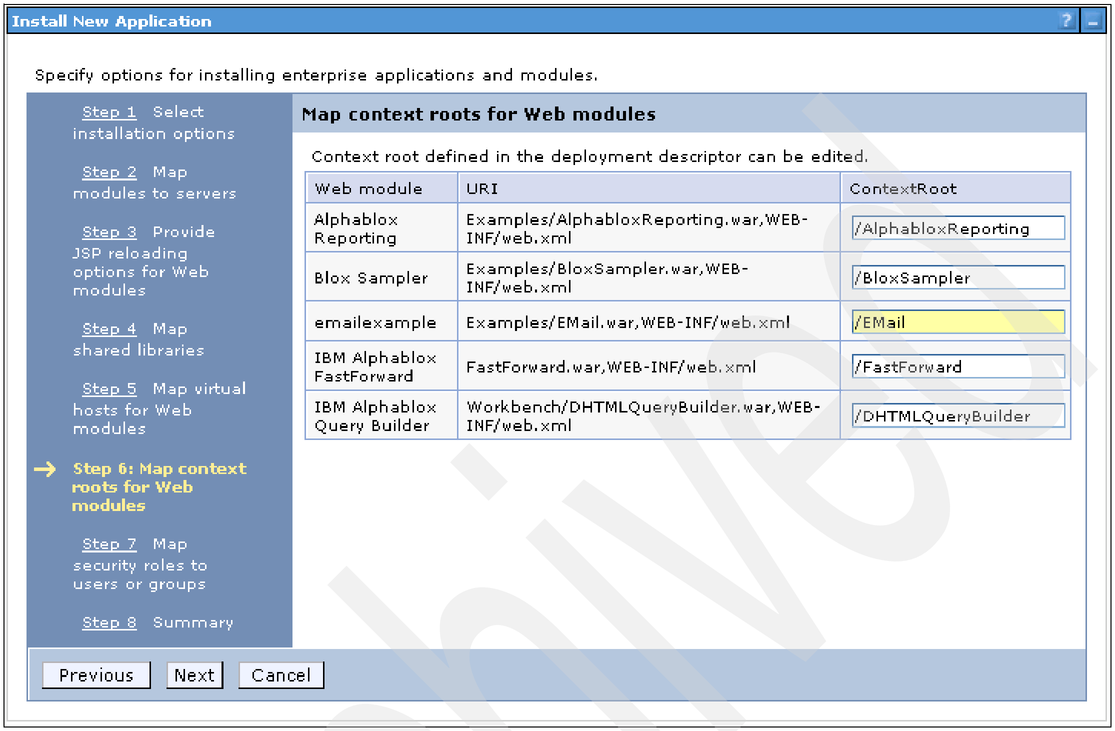


Figure 11-88 IBM Alphablox installation: Step 6

42. In the Step 7: Map security roles to users/groups window (Figure 11-89), add at least one user for each of the three user roles, AlphabloxAdministrator, AlphabloxDeveloper and AlphabloxUsereach:

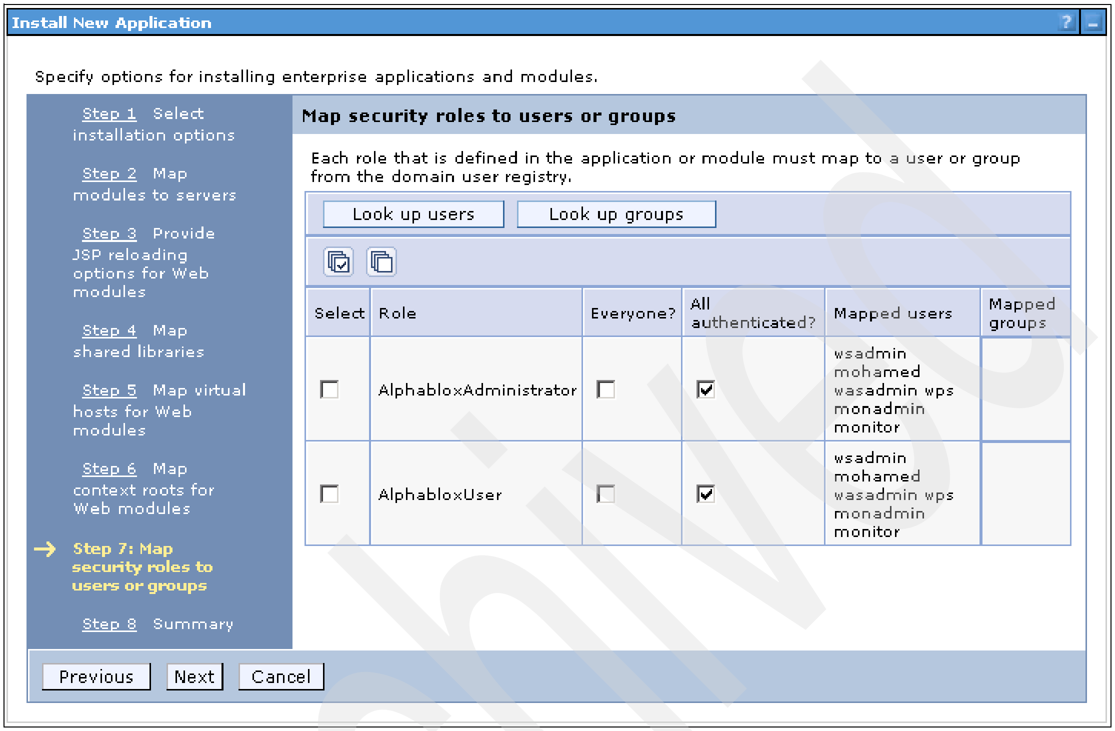


Figure 11-89 IBM Alphablox installation: Step 7

a. For each role, select the check box in front of that role, then click **Lookup Users** or **Lookup Groups** to add administrative users. You must select at least one user. After you have added the users or groups, they should be listed in the Mapped Users or Mapped Groups value for this role (Figure 11-90 on page 435).

b. Select the check box under the All Authenticated column for each role. This allows all authenticated users to access applications (Figure 11-90 on page 435).

c. When finished, press **Next**.

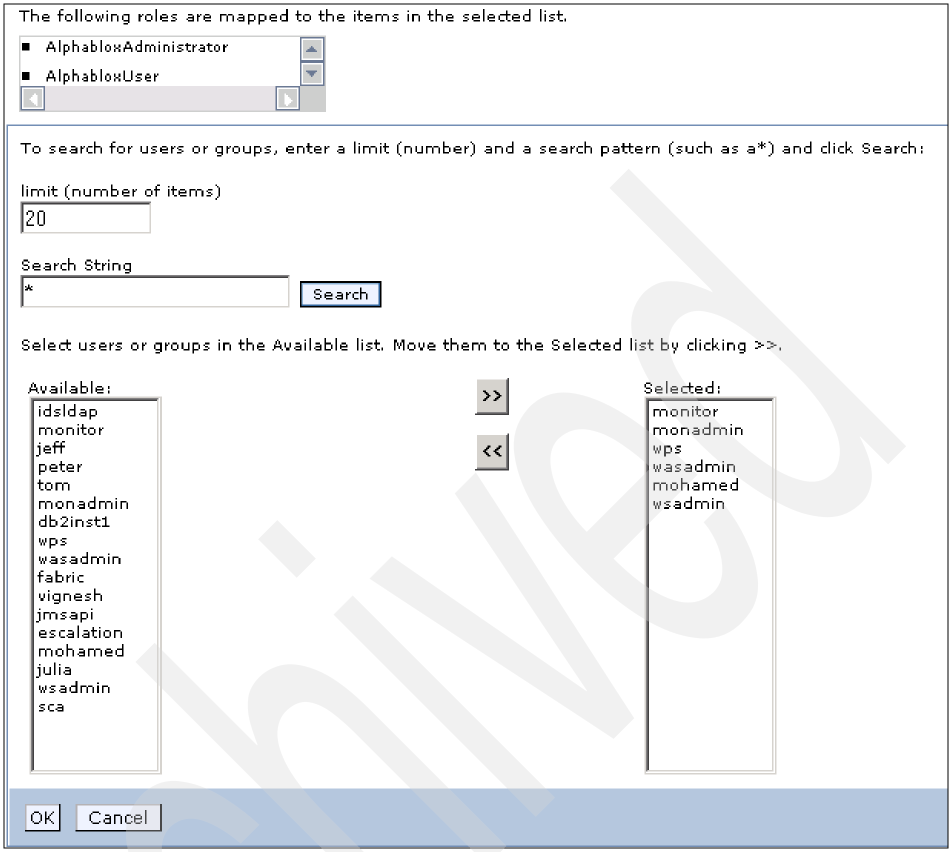


Figure 11-90 User mapping

43.In the Step 8: Summary window (Figure 11-91), scroll to the bottom of this window, then press the **Finish**.

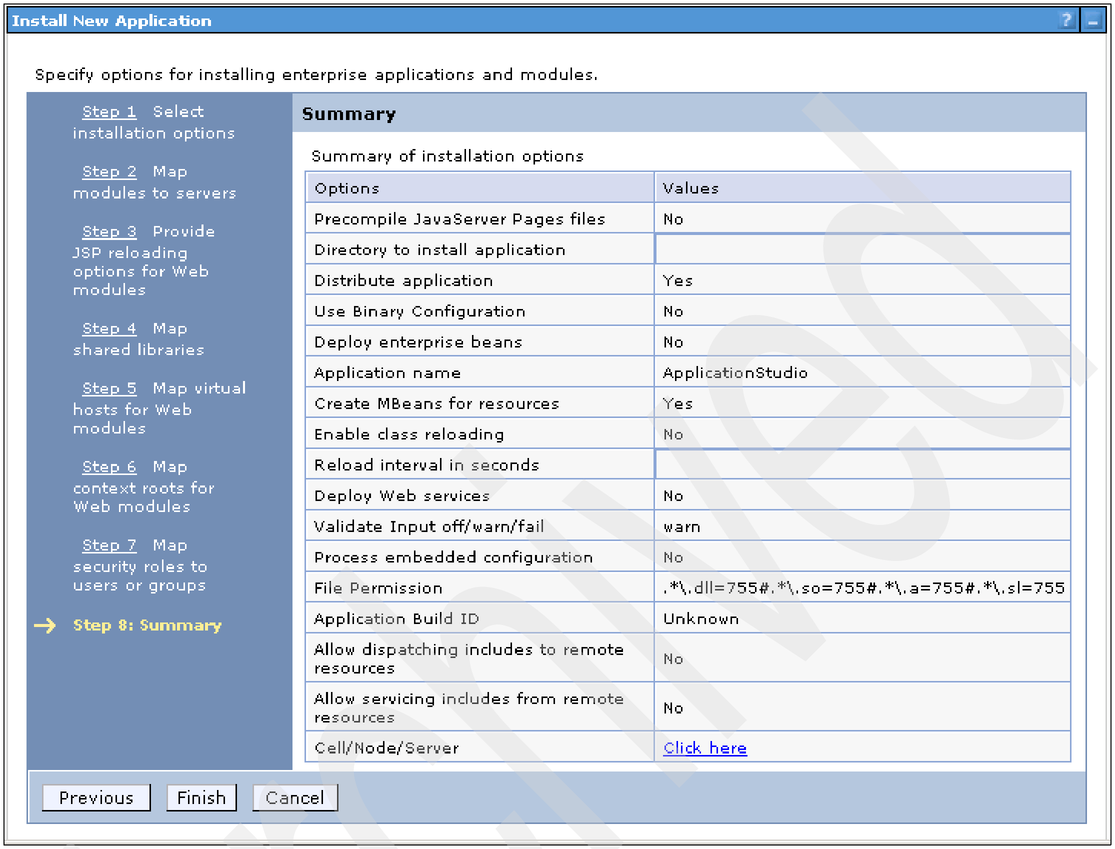


Figure 11-91 IBM Alphablox installation: Step 8

44.The application is then deployed and Application AlphabloxPlatform installed successfully will be displayed. Click **Save** (Figure 11-92).

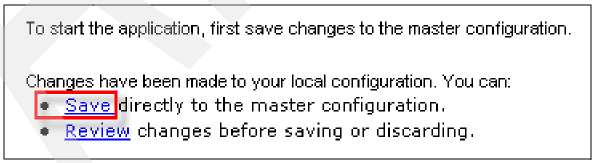


Figure 11-92 Save to master configuration

45.Repeat this process to install the ApplicationStudio.ear application.

46. After installing both EAR files, ensure that the AlphabloxPlatform and ApplicationStudio applications are started correctly by clicking **Applications** → **Enterprise applications**.

47. Click the AlphabloxPlatform application name, then click the Configuration tab.

48. Under the Detail Properties section, click the **Starting behavior** link (Figure 11-93).

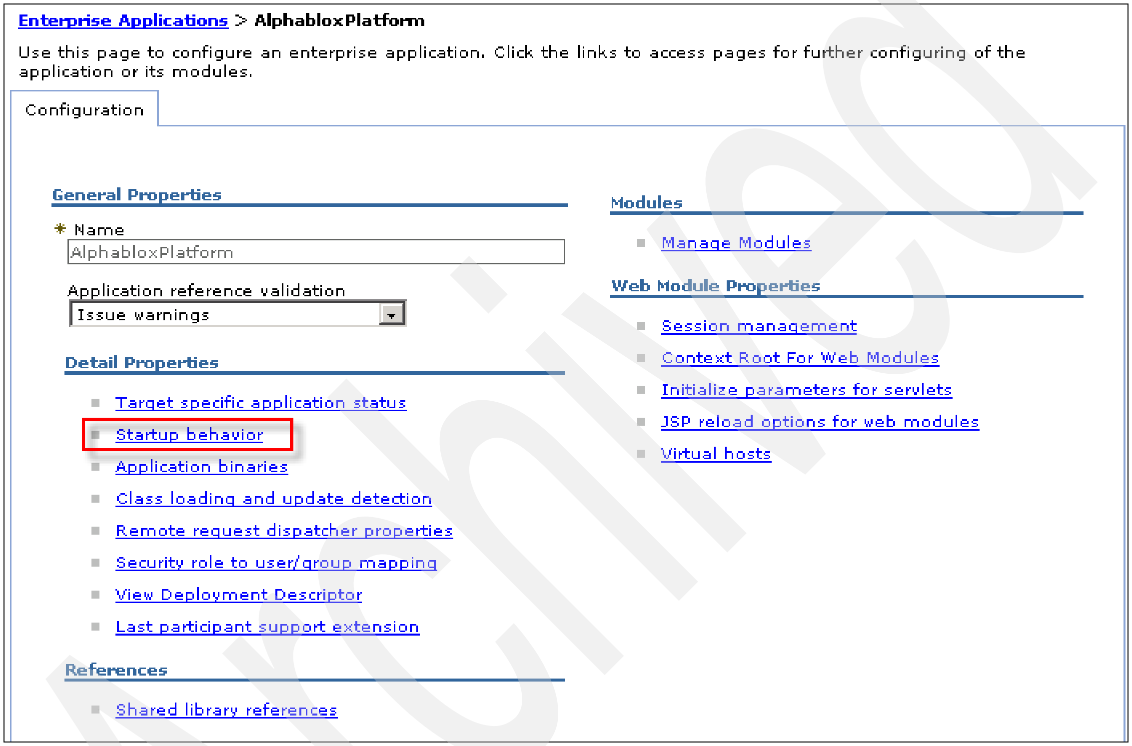


Figure 11-93 AlphabloxPlatform starting behavior

49. In the Configuration tab (Figure 11-94) change the value of the Starting Order text box to 1000. This will help ensure that the AlphabloxPlatform application starts after all other applications have loaded.

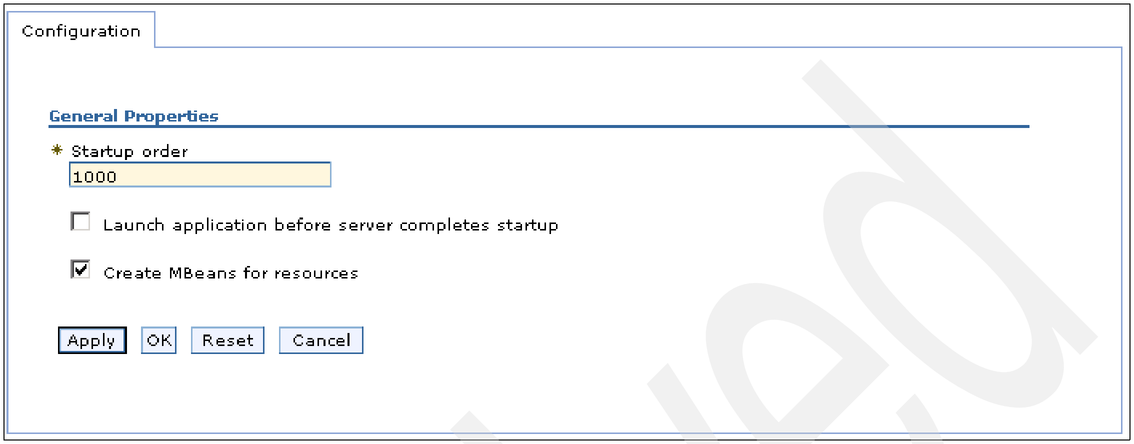


Figure 11-94 Starting Order value

50. Scroll to the bottom of the window and click **OK**. Save changes to the master configuration.

51. Restart the WebSphere Business Monitor clusters, and verify in the logs that the dashboards server is started successfully by locating the following message: IBM Alphablox (AlphabloxAnalytics) started.

Post-installation configurations

For WebSphere vertical clusters, you need to 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, or physical machine. A horizontal cluster has cluster members on multiple nodes across many machines in a cell. You can configure either type of cluster, or have a combination of vertical and horizontal clusters.

Log in to the WebSphere Integrated Solutions Console. Under **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 was as follows:

   ```
   http://ABX_Cluster_member:9081/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.
   d. Enter the database user name in the Default Username text box.
   e. Enter the database password in the Default Password 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.

> **Note:** You must restart the server for the created data sources to be reflected on other cluster members.

11.5.3 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 BusinessSpace/registry directory as indicated below.

The Business Space registration files are located in the following path: <WebSphere_Process_Server_installation>\BusinessSpace\wbm\registryData in all nodes of Business Space cluster.

Complete the following steps 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 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)

   An example of an edited monitorWidgets.xml file is shown in Example 11-1 on page 441.

Example 11-1 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 as indicated below in bold. Example 11-2 shows monitorEndpoints.xml and Example 11-3 on page 443 shows monitorABXEndpoints.xml.

Example 11-2 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: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 -->
```

Example 11-3 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.

11.6 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 and WebSphere Portal.

Configure access to the monitor model resources using Monitor Data Security in the administrative 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

11.7 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 no problems are expected to occur during runtime.

11.7.1 Maintain the WebSphere Business Monitor Server

To maintain a functioning and well performing system, perform the following tasks:

- Verify the Monitor server is running.
- Verify the Monitor messaging engine is running.
- Verify the Monitor JDBC connections are working correctly.
- Verify the Service Integration Bus Link (SIB link) is started if you are using remote Common Event Infrastructure.
- Verify that messages in monitor models queues are not accumulating.
- Verify that the sum of all monitor models queue depths will not exceed the messaging engine maximum number of messages threshold.
- Check the event consumption rate of WebSphere Business Monitor Server.
- Verify WebSphere Business Monitor is not running in error mode (slow or blocked event consumption)

11.7.2 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:

- Backup 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 table space size regularly.
- Check database sizes regularly to avoid out of disk space. This includes the Monitor database and messaging engines database.
- Use database tools to refine and tune indexes and tables.

11.7.3 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.
- Do not use the default Derby as a database. For high performance, use a database management system such as DB2 or Oracle®.
- 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 message consumption patch size according to workload (flow of events).

Part 4

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 Web server. Point your Web browser at:

`ftp://www.redbooks.ibm.com/redbooks/SG247665`

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the IBM Redbooks form number, SG247665.

How to use the Web material

Create a subdirectory (folder) on your workstation, and extract the contents of the Web material zip file into this folder.

Abbreviations and acronyms

BPC	Business Process Choreographer
BPEDB	Business Process Choreographer database
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Modeling Notation
BRM	Business Rules Manager
CBA	Composite Business Application
CBEs	Common Base Events
CEI	Common Event Infrastructure
CMP	container Managed Persistence
CRM	Customer Relationship Management
DA	Dynamic Assembler
EAR	Enterprise Application Archive
EIS	Enterprise Information System
EJB	Enterprise JavaBeans
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
HA	High Availability
HAGEO	High Availability Geographic Cluster
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines Corporation
ITSO	International Technical Support Organization
J2EE	Java 2 Enterprise Edition
JDBC	Java Database Connectivity
JMS	Java Message Service
JMX	Java Management Extensions
JTA	Java Transaction API
JVM	Java Virtual Machine
KPI	Key Performance Indicator
LB	Load Balancing
LTPA	Lightweight Third Party Authentication
MDB	Message-Driven Beans
ORB	Object Request Broker
PMI	Performance Monitoring Infrastructure
POJO	Plain Old Java Object
QoS	Quality of Service
SCA	Service Component Architecture
SCDL	Service Component Definition Language
SDO	Service Data Object
SIP	Session Initiation Protocol
SLA	Service Level Agreement
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
TEPS	Tivoli Enterprise Portal Server
TLS	Transport Layer Security

TPV	Tivoli Performance Viewer
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

For information about ordering these publications, see "How to get Redbooks" on page 454. Note that some of the documents referenced here may be available in softcopy only.

- *Production Topologies for WebSphere Process Server and WebSphere ESB V6*, SG24-7413
- *IBM WebSphere Application Server V6.1 Security Handbook*, SG24-6316
- *WebSphere Application Server Network Deployment V6: High Availability Solutions*, SG24-6688
- *IBM Tivoli Composite Application Manager Family Installation, Configuration, and Basic Usage*, SG24-7151
- *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240
- *Solution Deployment Guide for IBM Tivoli Composite Application Manager for WebSphere*, SG24-7293
- *Getting Started with IBM WebSphere Business Services Fabric V6.1*, SG24-7614
- *Large-Scale Implementation of IBM Tivoli Composite Application Manager for WebSphere and Response Time Tracking*, REDP-4162
- *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

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

WebSphere Business Process Management V6.1.2 Production Topologies

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

WebSphere Business Process Management V6.1.2 Production Topologies

Securing, administering, and extending WebSphere Process Server topologies

Incorporating WebSphere Business Services Fabric

Integrating WebSphere Business Monitor

The IBM WebSphere Dynamic Process Edition is a comprehensive set of role-based, SOA-enabled product capabilities providing customers 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: IBM WebSphere Process Server, WebSphere Business Services Fabric, and WebSphere Business Monitor.

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 book also contains a chapter on extending existing production topologies to add components such as additional clusters.

This Redbooks publication also provides practical examples demonstrating how to incorporate WebSphere Business Services Fabric and WebSphere Business Monitor into existing topologies. The book contains extensive examples of working with all of these products in distributed environments. A separate publication covering z/OS is forthcoming.

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

SG24-7665-00 073843180X