WebSphere® software

IBM

Production Topologies for WebSphere Process Server and WebSphere ESB V6

Discusses messaging, CEI, business process container, and security

Includes selection criteria to pick the right topology

Features examples that include step-by-step instructions

Martin Keen
Mahesh Gandhe
Stephen Gibney
Alex Lis
Davide Veronese
Yan Zhao Zhang

Redbooks

ibm.com/redbooks

International Technical Support Organization

Production Topologies for WebSphere Process Server and WebSphere ESB V6

April 2007

SG24-7413-00

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

First Edition (April 2007)

This edition applies to WebSphere Process Server V6.0.2 and WebSphere Enterprise Service Bus V6.0.2.

Contents

Notices

Trademarks

Preface

WebSphere® Process Server and WebSphere Enterprise Service Bus (ESB) are highly available and scalable products for running enterprise-level service-oriented architecture (SOA) solutions. But does every production topology that uses these products follow the same basic pattern? This book answers that question by providing guidance on how to select and build production topologies based on the applications that you deploy to that topology. It also helps you to discern which production topology is suitable for your environment and provides step-by-step guidance on how to build that topology. The intended audience of this book is IT architects and administrators who want to create appropriate production topologies for WebSphere Process Server and WebSphere Enterprise Service Bus.

Part one of the book introduces some of the basic concepts and discusses three types of production topologies for WebSphere ESB and for WebSphere Process Server. It provides guidance on how to select the appropriate topology, discusses security considerations, and introduces the sample scenarios that we built for this book.

Part two focuses on building production topologies for WebSphere ESB. Using sample applications, this part provides step-by-step instructions for building WebSphere ESB V6.0.2 production topologies.

Part three uses a similar format but focuses on production topologies for WebSphere Process Server V6.0.2. This part includes step-by-step instructions on how to build a full support production topology that implements security.

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 (ITSO), Raleigh Center.

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

Mahesh Gandhe is an Advisory Software Engineer at IBM Burlingame Development Laboratory in California, U. S. He has seven years of experience in software development and testing. Currently, he leads the WebSphere Process

Server Stress Testing team. Previously, he worked with the WebSphere Process Server development team. Mahesh holds a Master of Computer Science degree from the University of Maryland at Baltimore County, and his thesis was conducted in collaboration with MIT Sloan School of Business.

Stephen Gibney works for IBM Software Group Services for WebSphere. He started his career working with solid state detectors at a U. K. research center in 1986, and he moved to a French research center in 1992. He made the transition to IT as a UNIX® System Administrator in 1993. He returned to the U. K. in 1998 to work in Systems Management with the Tivoli® product set. He has been working with the WebSphere product set since 2004. Stephen works as part of a team that designs, develops, and implements WebSphere solutions. Stephen holds a degree in Mathematics and Computing.

Alex Lis works for IBM Software Group Services for WebSphere. He started his career working as a freelance Web application specialist in 1996. His areas of expertise include networking, Web applications, performance and capacity, and the WebSphere product set. Alex holds a degree in Computer Science and Management from The University of Edinburgh.

Davide Veronese is a Certified IT Architect with IBM Global Business Services, Italy. He has more than 10 years of experience in J2EE™ application design, development, integration, and consulting. His areas of expertise include e-business integration, J2EE architecture, and WebSphere Process Server. Davide holds a degree in Computer Science from the University of Verona.

Yan Zhao Zhang is a Staff Software Engineer in IBM China Software Development Lab. He has four years of experience in the WebSphere integration solution field. His areas of expertise include WebSphere Process Server Network Deployment architecture and WebSphere Process Server upgrade and migration. Yan Zhao holds a master degree in Software Engineering from Northwestern Polytechincal University.

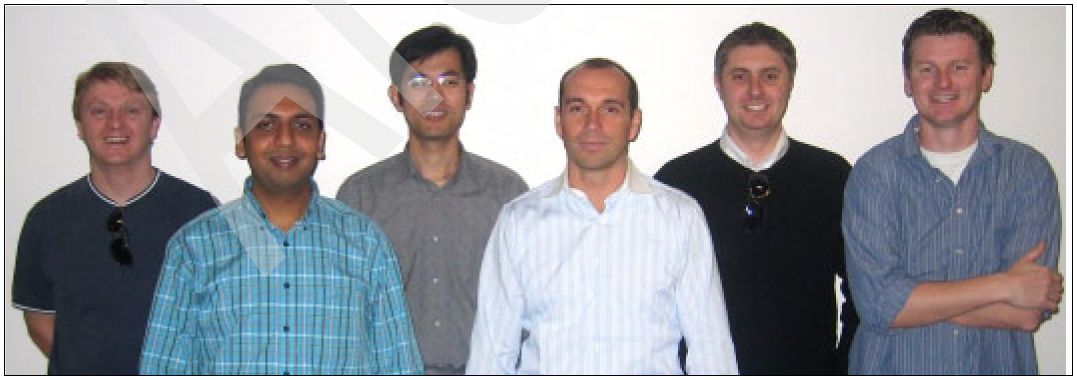


Figure 1 The team (left-to-right): Stephen, Mahesh, Yan Zhao, Davide, Alex, and Martin

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

Charlie Redlin
Rochester BringUp Lab, Rochester, U. S.

Karri Carlson-Neumann
Rochester BringUp Lab, Rochester, U. S.

Manoj Khangaonkar
IBM Senior Software Engineer, Burlingame, U. S.

Become a published author

Comments welcome

Part 1

Overview

Welcome to this book

This chapter provides guidance on how to get the most out of this book. It includes the following sections:

- An introduction to this book
- How to read this book

1.1 An introduction to this book

A warm welcome to this book from the team that wrote its content. We gathered for five intensive weeks in Raleigh, North Carolina, U. S., to create this book. We hope that you find it to be an insightful and useful read.

The motivation in writing this book was to show that there are different ways to build production topologies for WebSphere Process Server and WebSphere ESB and how through a selection process that you can pick the correct production topology for your environment.

We also wanted to show how to actually go about building these topologies. We picked two topologies for each product (WebSphere Process Server and WebSphere ESB) and provide step-by-step instructions that you can follow to build each topology.

One of the keys to selecting the correct topology is determining the type of applications that you will deploy to the topology. To help demonstrate this topic, we built four sample applications around a fictional banking organization called *ITSO Bank*. These samples are very simple, including either a basic mediation module or business process and a presentation and enterprise service layer. We deploy a sample application to each of the topologies to demonstrate how each topology works.

We hope that you are able to use the content of this book as a starting point to build your own production topologies with WebSphere Process Server and WebSphere ESB.

1.2 How to read this book

This section describes how we structured this book and provides guidance on which chapters you should read.

We divided this book into three parts. The chapters in Part 1 provide overview information: an overview of the production topologies, an overview of the basic concepts of creating network deployment topologies, an overview of security considerations, and an overview of the sample applications.

Parts 2 and 3 include step-by-step instructions for creating the production topologies. Part 2 describes topologies for WebSphere ESB V6.0.2, and Part 3 describes topologies for WebSphere Process Server V6.0.2.

To keep the step-by-step instructions in parts 2 and 3 succinct, we only enable security for one production topology (the Full Support topology for WebSphere Process Server). The security considerations that we discuss in Chapter 9, “Implementing the Full Support topology for WebSphere Process Server” on page 319 apply equally to the other topologies. None of the topologies can be considered ready for production if they are not secured.

Part 1: Overview

Part 1 includes the following chapters:

- **Chapter 1, “Welcome to this book”**

  This is the chapter that you are reading now.

- **Chapter 2, “Introducing the basic concepts and products”**

  This chapter introduces WebSphere Process Server and WebSphere ESB, and provides basic information about network deployment and clustering terminology. If you are not familiar with building topologies with WebSphere Process Server and WebSphere ESB, then this chapter is a good place to start.

- **Chapter 3, “Security considerations for WebSphere Process Server and WebSphere ESB”**

  This chapter provides an overview of the major security considerations when building WebSphere Process Server or WebSphere ESB topologies. You can apply the concepts in this chapter to all production topologies discussed within this book.

- **Chapter 4, “Production topologies for WebSphere Process Server and WebSphere ESB”**

  This chapter is the main focus of this book and is essential reading. This chapter describes three patterns of production topology for WebSphere Process Server and WebSphere ESB: the Basic topology, Loosely Coupled topology, and Full Support topology. It also provides guidance on selecting the appropriate topology for a given environment.

- **Chapter 5, “Scenarios that we use in this book”**

  This chapter describes sample applications for a fictitious company called ITSO Bank. We use the sample applications in Part 2 and Part 3 of this book.

Part 2: Production topologies for WebSphere ESB

Part 2 includes the following chapters:

- **Chapter 6, "Implementing the Basic topology for WebSphere ESB"**

  The Basic topology for WebSphere ESB is intended for simple synchronous applications. This chapter provides step-by-step instructions for building this topology and deploys one of the sample ITSO Bank applications to it.

- **Chapter 7, "Implementing the Full Support topology for WebSphere ESB"**

  The Full Support topology is suitable for the deployment of any type of WebSphere ESB application, including those that make use of CEI. This chapter provides step-by-step instructions for building this topology and deploys one of the sample ITSO Bank applications to it.

Part 3: Production topologies for WebSphere Process Server

Part 3 includes the following chapters:

- **Chapter 8, "Implementing the Loosely Coupled topology for WebSphere Process Server"**

  The Loosely Coupled topology for WebSphere Process Server allows the deployment of applications containing synchronous and asynchronous interactions. This chapter provides step-by-step instructions for building this topology and deploys one of the sample ITSO Bank applications to it.

- **Chapter 9, "Implementing the Full Support topology for WebSphere Process Server"**

  The Full Support topology for WebSphere Process Server is suitable for any type of WebSphere Process Server application, including those that make use of CEI. This chapter provides step-by-step instructions for building this topology and deploys one of the sample ITSO Bank applications to it.

  This chapter also describes how to add security to a production topology. Read this chapter for information about how to apply security to any of the production topologies.

2

Introducing the basic concepts and products

This chapter provides an introduction to the fundamental concepts and technologies that apply when deploying solutions based on the WebSphere Enterprise Service Bus and WebSphere Process Server products.

It includes the following sections:

- IBM product overview
- Service Component Architecture
- Messaging
- Business processes
- WebSphere Network Deployment components
- Clustering fundamentals

2.1 IBM product overview

This section gives an overview of WebSphere Process Server and WebSphere Enterprise Service Bus. Both are members of the product suite from IBM for integration capabilities within organizations, and each offers an implementation of principles laid down by service-oriented architecture (SOA) and the Service Component Architecture (SCA).

Figure 2-1 shows how the two products are based upon WebSphere Application Server Network Deployment with its J2EE runtime and proven quality of service strengths such as clustering, scalability, and security.

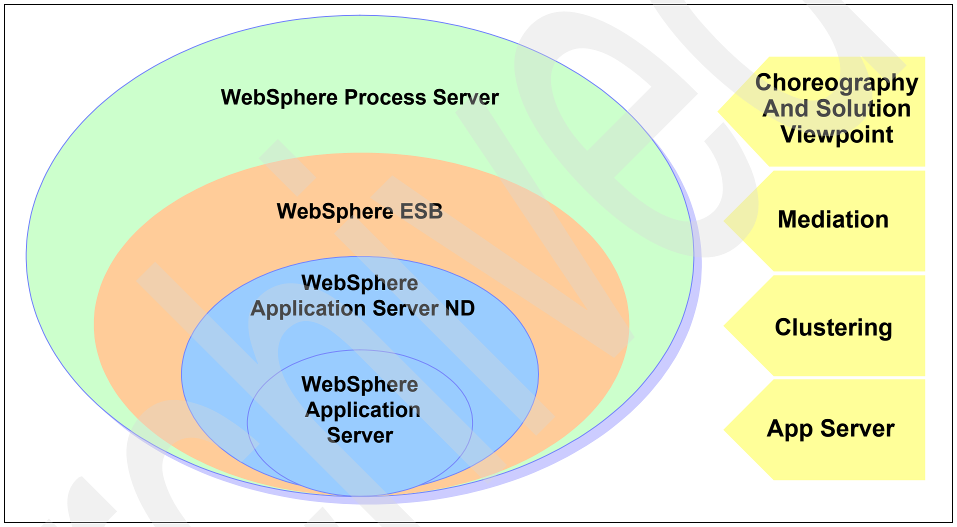

WebSphere Process Server
WebSphere ESB
WebSphere Application Server ND
WebSphere Application Server
Choreography And Solution Viewpoint
Mediation
Clustering
App Server

Figure 2-1 Relationship and services between the products

2.1.1 WebSphere Process Server

This section gives an overview of WebSphere Process Server.

Business Process Management

To stay ahead of the competition, companies are constantly striving for differentators. A common practice is to review and to refine business activities. Processes evolve over time as a company reacts and adapts to market conditions, and any method that codifies and automates processes quickly will help bring about competitive advantages.

Thanks to improvements in graphical tools and business-friendly notations, the bridge between business and IT departments is narrowing, and both parties can work closely in redefining and developing new processes and workflows. In addition, SOA promotes the building of business process models, business rules definition, business rules implementation, and technical workflow models. With these enablements, Business Process Management brings processes, people, and information together to increase efficiency for the entire enterprise.

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. It is, along with WebSphere Enterprise Service Bus, 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 the 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.

Figure 2-2 shows the major components that make up the WebSphere Process Server.

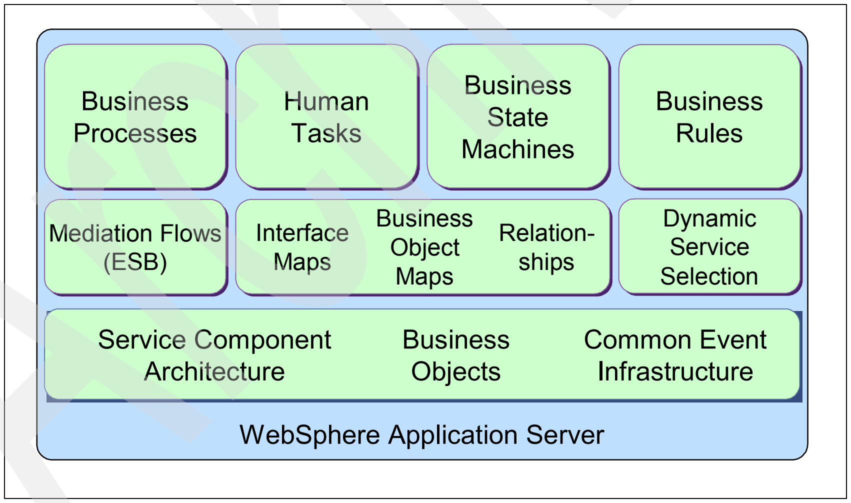


Figure 2-2 WebSphere Process Server V6 components

Business Processes component

The Business Processes component includes a WS-BPEL-compliant process engine. Users develop and deploy business processes with support for long-running and short-running processes and also make use of a robust compensation engine.

WebSphere Process Server supports dynamic business processes by:

- Visually describing processes that span people, systems, applications, tasks, rules, and the interactions among them
- Supporting long-running and short-running business processes
- Providing transaction rollback-like functionality for loosely coupled business processes that cannot be undone automatically by the application server
- Integrating fault handling for easy, in-flow exception handling
- Accepting Java™ snippets and artifacts as part of a business process

Human Tasks component

This component helps to describe and to implement automated tasks and role-based human tasks as part of automated business processes. Business processes that involve human interaction can be interruptible and persistent and can offer management of role-based task assignment, invocation, and escalation. For example, users can pause long or complex tasks and resume them later for completion. You can use existing Lightweight Directory Access Protocol (LDAP) directories (and operating system repositories and the WebSphere user registry) to access staff information and authorization.

Business State Machine components

These components provide an alternative to modeling a business process. Business can represent their business processes on states and events that sometimes are easier to model than a graph-orientated business process model. These event-oriented scenarios are sometimes hard to model in a WS-BPEL model but can be very easy to model in a state-machine diagram. The WebSphere Process Server state machine is designed to emulate Unified Modeling Language (UML) state-machine diagrams.

There is an event catalog with four main categories that details types of events that WebSphere Process Server manages:

- Common Base Event standard elements
- Business Object events
- Business Process Choreographer events
- Process Server events

Business Rules component

This component provides support for rule sets (*if-then* rules) and decision tables. Business rules are categorized into rule groups that hide implementation details from the consumer and that are accessed similar to any other component. This capability allows dynamic changes of a business process for a more responsive business environment. They include policies and conditions that are imposed by the business.

Additionally, there are a number of supporting components that deal with the differences that come with various applications and systems in a heterogeneous IT environment. These servicing components are:

- *Interface maps*: Very often interfaces of existing components match semantically but not syntactically (for example, *updateCustomer* versus *updateCustomerInDB2*). This is especially true for already existing components and services. Interface maps translate these calls so that these components can be invoked. Additionally, business object maps can be used to translate the actual business object parameters of a service invocation.
- *Business object maps:* Used to translate one type of business object into another type, these maps can be used in a variety of ways (for example, as an interface map to convert one type of parameter data into another).
- *Relationships:* A common problem with keeping business objects synchronized is that different back-end systems use different keys to represent the same objects. The WebSphere Process Server relationship service establishes relationship instances between objects in these disparate systems. These relationships are typically accessed from a business object map while one business object format is being transformed into another.
- *Selector:* Different services that all share the same interface can be selected and invoked by a selector dynamically. For example, a customer support process might use different human task implementations during different times of the day. Work is routed to different support centers (Americas, Europe, and Asia-Pacific) based on the time of day.

The key tools for modelling and assembling processes for import into the Business Process Choreographer are WebSphere Business Modeler and WebSphere Integration Developer. With these development tools, the developer can adhere to specification rules and switch off all the extensions or use them as desired.

You can find information about WebSphere Process Server online at:

http://www.ibm.com/software/integration/wps/

2.1.2 WebSphere Enterprise Service Bus

WebSphere Enterprise Service Bus compliments WebSphere Process Server by introducing enhanced integration capabilities. Services should, by definition, be reusable by a number of different consumers, so that the benefits of reduced connections are achieved. This would apply to WebSphere Process Server as a consumer or provider.

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

Figure 2-3 shows the WebSphere Enterprise Service Bus programming model and features.

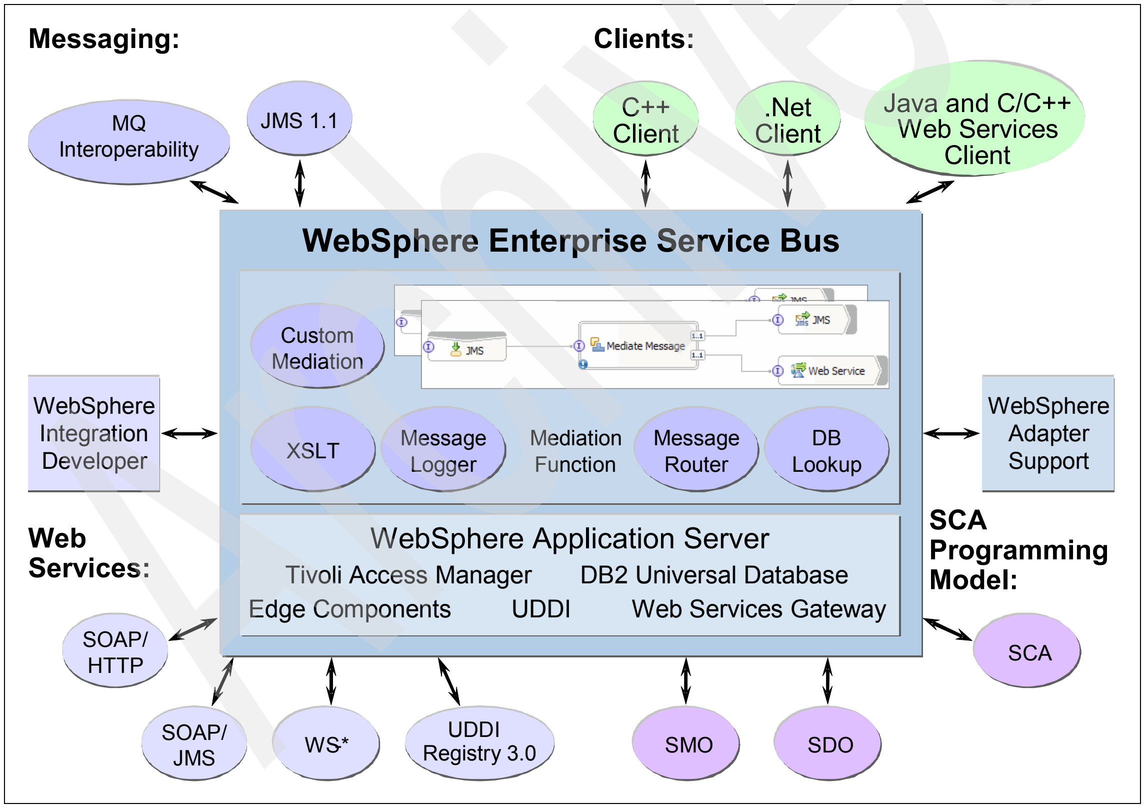


Figure 2-3 WebSphere Enterprise Service Bus components and associations

For integration to be successful, SOA states the need of having a single invocation model and a single data model. WebSphere Enterprise Service Bus uses SCA as its invocation model and that is why SCA is part of the first layer of elements shown in Figure 2-3. Also, Common Event Infrastructure (CEI), the foundation for monitoring business performance, is part of this layer. The ESB is basic for the integration strategy and there's the need to know how it behaves. The event definition is standardized using the OASIS specifications.

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 the provider.
- Handle events from different services.

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

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

2.1.3 Supporting products

This section includes brief details on the elements that are required to support the operation of both WebSphere Process Server and WebSphere Enterprise Service Bus.

WebSphere Application Server

WebSphere Application Server V6 is the IBM implementation of the Java 2 Enterprise Edition (J2EE) platform, which conforms to V1.4 of the J2EE specification. WebSphere Application Server is available in several different configurations that are designed to meet a wide range of customer requirements. Each configuration is packaged with other components that provide a unique environment.

The Network Deployment configuration offers central administration and workload management. A Network Deployment environment consists of one or more *Base* installations and a *Deployment Manager* installation. The Base application servers are added to the cell in a scalable manner and are managed by the Deployment Manager. The Network Deployment package also includes the Web Services Gateway, a UDDI Registry, and J2EE Connector Architecture services.

For more information about WebSphere Application Server refer to:

http://www.ibm.com/software/webservers/appserv/was/

For more information about the J2EE 1.4 specification refer to:

http://java.sun.com

IBM DB2 Universal Database

IBM DB2® Universal Database™ Enterprise Server Edition is a version of DB2 Universal Database that allows you to create and manage single partitioned or partitioned database environments. Partitioned database systems can manage high volumes of data and provide benefits such as high availability and increased performance.

DB2 Universal Database V8.2 delivers added features to address the ever-increasing demands and requirements on important data, which include:

- Broadened autonomic computing solutions that automate and simplify potentially time-consuming and complex database tasks.
- A significant amount of new capabilities as well as further integration of DB2 tooling into the Microsoft® .NET and WebSphere Java environments.

  These new capabilities simplify the development and deployment of DB2 applications and allow application developers to take advantage of the openness, performance, and scalability of DB2, without regard to the back-end database or the chosen application architecture.
- Integration of industry-proven high-availability disaster recovery technology, allowing line-of-business managers and the enterprise to benefit because applications face less risk of downtime.

For more information, refer to the IBM DB2 Universal Database Web site:

http://www.ibm.com/software/data/db2/udb

2.2 Service Component Architecture

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:

- 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.
- 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. An import and export also has interfaces so that the published service can be invoked.

  All components have interfaces of the WSDL type. Only Java components support Java-type interfaces. If a component, import or export, has more than one interface, all interfaces must be the same type.

  Components can be called synchronously or asynchronously, and this call is independent of whether the implementation itself is synchronous or asynchronous. The preferred interaction style can be defined as synchronous or asynchronous. The asynchronous interaction advertises to users of the interface 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.

2.2.1 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.

2.3 Messaging

A *service integration bus* or *bus* is a conceptual architectural component. It gives an administrator the ability to group a collection of resources together that provide the messaging capabilities of the bus. At run time, the bus presents these cooperating messaging resources to applications as a single entity and hides from those applications the details of how the bus is configured and where on the bus the different resources are located.

A bus is defined at the cell level. It is anticipated that, in a standard configuration, no more than one bus will be required within a cell. However, a cell can contain any number of buses.

A *bus member* is simply an application server, or cluster of application servers, that has been added as a member of a bus. Adding an application server, or cluster of application servers, as a member of a bus automatically defines a number of resources on the bus member in question. In terms of the functionality provided by a bus, the most important of the resources that are automatically defined is a messaging engine.

The *messaging engine* is the component within an application server that provides the core messaging functionality of a bus. At run time, it is the messaging engines within a bus that communicate and cooperate with each other to provide the messaging capabilities of the bus. A messaging engine is responsible for managing the resources of the bus and it also provides a connection point to which local and remote client applications can connect.

A messaging engine is associated with a bus member. When an application server is added as a member of a bus, a messaging engine is created and associated automatically with this application server.

A messaging engine is a relatively lightweight run time object. This allows a single application server to host several messaging engines (Figure 2-4). If an application server is added as a member of multiple buses, that application server is associated with multiple messaging engines—one messaging engine for each bus of which it is a member.

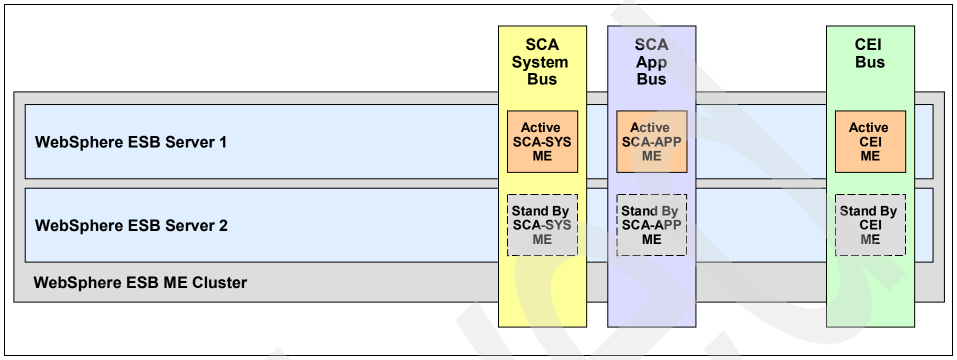


Figure 2-4   Application server with multiple messaging engines

When a cluster of application servers is added as a member of bus, a single messaging engine is created and associated with the application server cluster automatically, regardless of the number of application servers that are defined as members of the cluster. At run time, only one messaging engine is active, and this messaging engine runs within a single application server within the cluster, as shown in Figure 2-4. The application server that is chosen to host the active messaging engine will be the first cluster member to start. The remaining application servers host standby messaging engines.

The active messaging engine is able to run within any of the application servers defined as members of the cluster. If the messaging engine or the application server within which it is running should fail, one of the remaining standby messaging engines is activated, as shown in Figure 2-5. Therefore, adding an application server cluster as a member of a bus enables failover for messaging engines that are associated with that cluster.

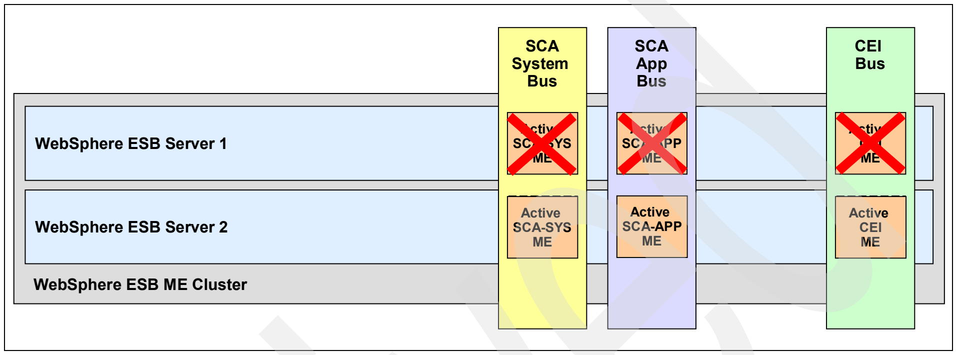


Figure 2-5 Failover of clustered messaging engines

2.4 Business processes

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

2.4.1 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.

2.4.2 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 and ultimately 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.

2.4.3 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.

2.4.4 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 with one another 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.

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 by dynamically applied. WebSphere Process Server also includes the Business Rules Manager (BRM) Web-based runtime tool for the business analyst so that business rules can be updated as business needs dictate without affecting other SCA services and deployed processes.

2.4.5 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 invocation

  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 invocation:

- 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 will then intermittently poll 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.

2.5 WebSphere Network Deployment components

WebSphere Network Deployment based products use several logical components that slot together to form highly configurable and scalable application server environments, as shown in Figure 2-6.

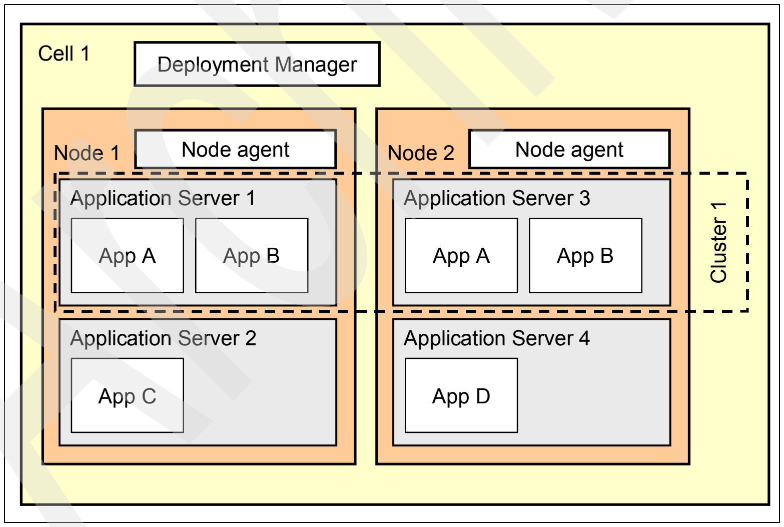


Figure 2-6 WebSphere Network Deployment components

This section describes the WebSphere Network Deployment 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 comprises a node agent, by which the node is controlled, and the application servers hosted on that node.

Node agents

The WebSphere *node agent* is a 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 *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 a:

- Stand-alone application server

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

- Singleton application server

  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

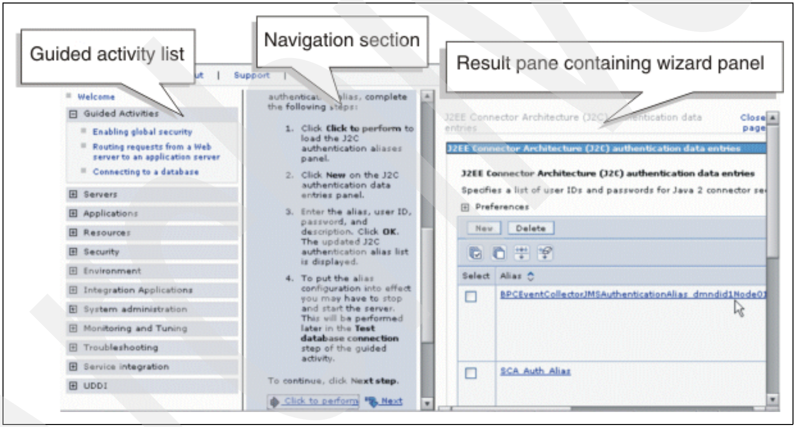
  An application server that is a cluster member resides on a node belonging to a cell, and is managed by a deployment manager residing on a separate node. The application server is part of a cluster.

Guided activities

Configuration of clusters or servers to support certain functions, such as network deployment, can be complex and require intimate architectural knowledge. IBM provides WebSphere *Guided Activities* as a way for users to perform such complex tasks.

The Guided Activity is a predefined series of steps that help the user perform the specific task. The Guided Activity describes the steps that must be completed and presents each step in the correct order.

To further simplify configuration tasks, a wizard may be presented to automatically perform a task. Figure 2-7 shows the Administrative Console displaying a guided activity.



authentica alias, complete the following steps:

1. Click **Click to perform** to load the J2C authentication aliases panel.
2. Click **New** on the J2C authentication data entries panel.
3. Enter the alias, user ID, password, and description. Click **OK**. The updated J2C authentication alias list is displayed.
4. To put the alias configuration into effect you may have to stop and start the server. This will be performed later in the **Test database connection** step of the guided activity.

To continue, click **Next step.**

J2EE Connector Architecture (J2C) authentication data entries

J2EE Connector Architecture (J2C) authentication data entries

J2EE Connector Architecture (J2C) authentication data entries

Specifies a list of user IDs and passwords for Java 2 connector se

Select	Alias
☐	BPCEventCollectorJMSAuthenticationAlias_dmndid1Node01
☐	SCA_Auth_Alias

Figure 2-7 Guided activity of Administrative Console

2.6 Clustering fundamentals

A *cluster* is a grouping of one or more fundamentally identical units that perform one task. WebSphere Network Deployment application servers are clustered to allow for higher throughput, to achieve higher levels of resiliency, or both.

2.6.1 Vertical clustering

In a *vertical cluster*, multiple application servers are placed onto the same node in order to better utilize the available resources (Figure 2-8). 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 in the event that the hardware hosting the members' node fails.

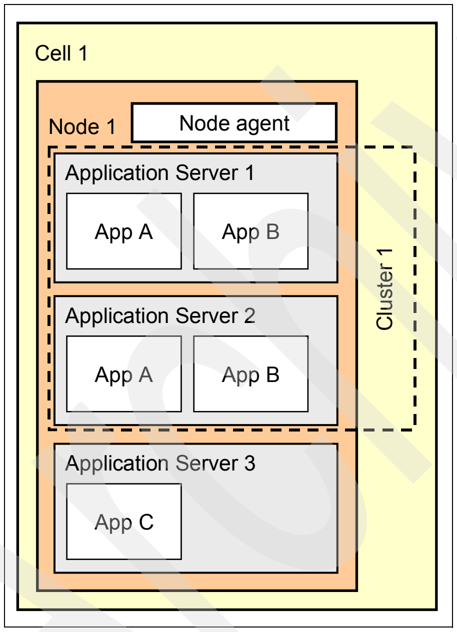


Figure 2-8 A vertically clustered WebSphere environment

2.6.2 Horizontal clustering

In a *horizontal* cluster, multiple application servers are distributed across nodes in order to utilize more physical resource (Figure 2-9). 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.

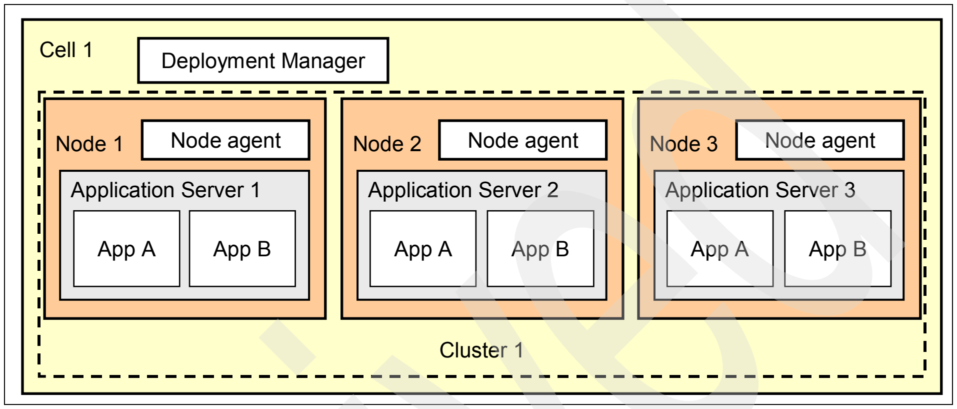


Figure 2-9 A horizontally clustered WebSphere environment

2.6.3 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 2-10).

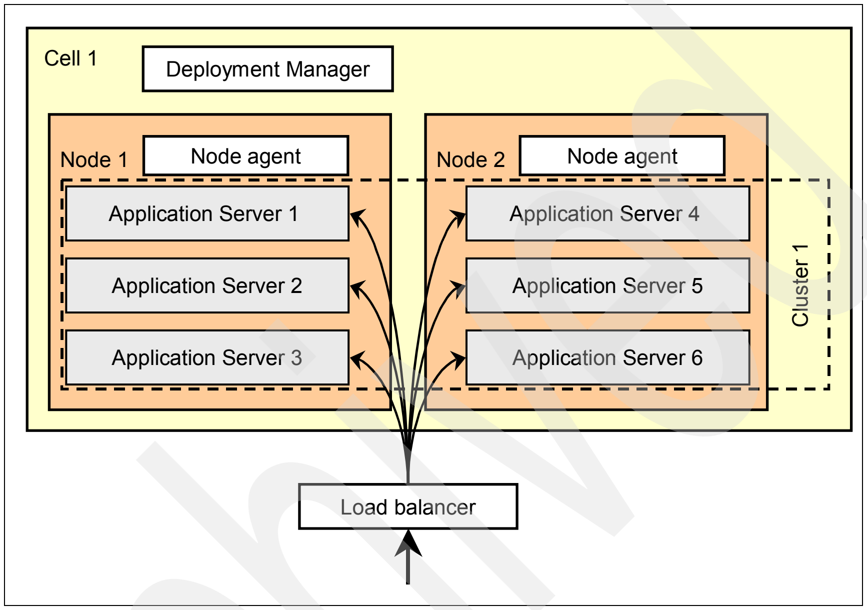


Figure 2-10 A load-balanced WebSphere environment

2.6.4 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, although 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 2-11. 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.

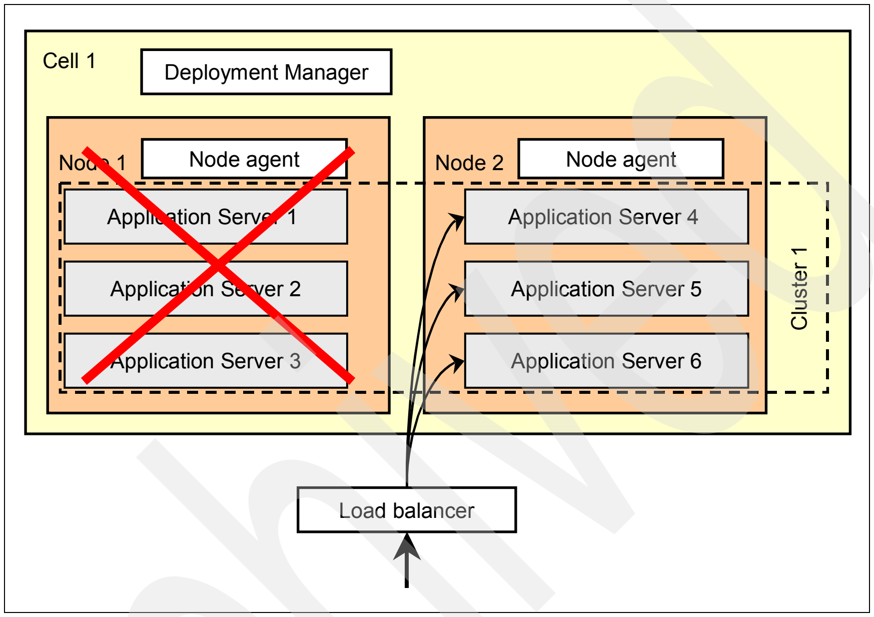


Figure 2-11 Failover in a clustered WebSphere environment

3

Security considerations for WebSphere Process Server and WebSphere ESB

This chapter addresses security considerations when building WebSphere Process Server and WebSphere Enterprise Service Bus topologies. It addresses the following topics:

- Java security
- Global security
- Service integration bus security
- Authentication
- Access control
- Integrity and privacy
- Single sign-on

For more information about security considerations with WebSphere Process Server and WebSphere ESB, refer to the article *WebSphere Process Server security overview*, which is located at:

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

3.1 Security features in WebSphere Process Server and WebSphere ESB

IBM WebSphere Process Server and WebSphere ESB provide security infrastructure and mechanisms that protect sensitive Java 2 Platform, Enterprise Edition (J2EE) resources and administrative resources. They also address various enterprise end-to-end security requirements. In addition, WebSphere Process Server and WebSphere ESB support other security providers, including:

- IBM Tivoli Access Manager (Policy Director)
- Reverse secure proxy server including WebSEAL

WebSphere Process Server and WebSphere ESB security is based on fundamental WebSphere Application Server, Java, and operating system security, as shown in Figure 3-1.

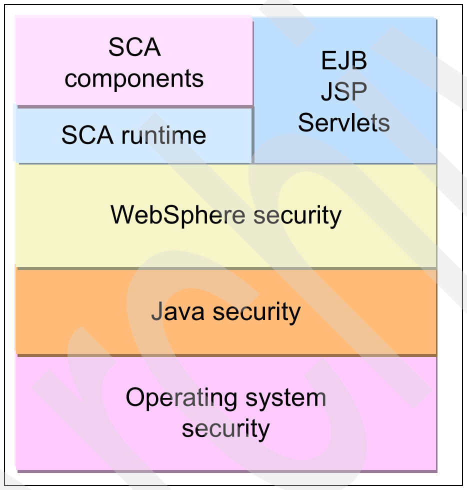


Figure 3-1 WebSphere Process Server and WebSphere ESB security stack

The following sections provide information about three levels of security in WebSphere Process Server and WebSphere ESB:

- Java security
- Global security
- Service integration bus security

3.1.1 Java 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. J2EE security guards access to Web resources such as servlets, JavaServer™ Pages™ (JSP™) files, and Enterprise JavaBeans™ (EJB) methods.

WebSphere Process Server and WebSphere ESB global security include J2EE role-based authorization, the Common Secure Interoperability Version 2 (CSIv2) authentication protocol, and Secure Sockets Layer (SSL) configuration.

Because Java 2 security is relatively new, many existing or even new applications might not be prepared for the very fine-grain access control programming model that Java 2 security is capable of enforcing. Administrators need to understand the possible consequences of enabling Java 2 security if applications are not prepared for Java 2 security. Java 2 security places some new requirements on application developers and administrators.

For more information about Java 2 security with WebSphere Application Server based products, refer to:

`http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/csec_rsecmgr2.html`

3.1.2 Global security

Global security applies to all applications that are running in an environment and determines whether security is used at all, the type of registry against which authentication takes place, and other values, many of which act as defaults.

The term *global security* represents the security configuration that is effective for the entire security domain. A security domain consists of all of the servers that are configured with the same user registry realm name. In some cases, the realm can be the machine name of a local operating system user registry. In this case, all of the application servers must reside on the same physical machine. In other cases, the realm can be the machine name of a Lightweight Directory Access Protocol (LDAP) user registry.

The basic requirement for a security domain is that the access ID that is returned by the user registry from one server within the security domain is the same access ID as that returned from the user registry on any other server within the same security domain. The access ID is the unique identification of a user and is used during authorization to determine if access is permitted to the resource.

The configuration of global security for a security domain involves configuring the following technologies:

- The common user registry
- The authentication mechanism

The global security configuration applies to every server within the security domain.

3.1.3 Service integration bus security

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

SIB supports authentication for connecting to the bus and role-based access control for accessing the destinations, sending, receiving, and browsing messages. WebSphere Process Server and WebSphere ESB uses container managed aliases to authenticate with the SIB. These aliases are set up during installation. 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). Each communication link needs to be secured with the SSL protocol.

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

3.2 Security requirements for a business integration solution

Security requirements for WebSphere Process Server and WebSphere ESB are:

- Authentication

  A mechanism that ensures the identity of users. Using this mechanism, the system can distinguish between valid users and invalid users of the system.

- Access control

  Ensures that authenticated users have necessary permissions to access only those resources or perform only those operations for which the system has granted the users permission.

- Integrity

  Ensures that data that is sent over the network is not modified in transit.

- Privacy and confidentiality

  Ensures that data that is sent over the network cannot be viewed by unauthorized parties and that it remains confidential.

- 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 only once. The authenticated context is propagated to downstream systems that can apply access control.

In this section we discuss how WebSphere Process Server and WebSphere ESB offers these required security features.

3.2.1 Authentication

When security is turned on, WebSphere Process Server and WebSphere ESB authenticate clients. WebSphere Process Server and WebSphere ESB support several authentication mechanisms:

- HTTP authentication
- Web service (WS) Security token
- Secure Socket Layer (SSL) Client authentication
- Java Authentication and Authorization Service (JAAS)

For authentication, WebSphere Process Server and WebSphere ESB support 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:

- Local operating system registries
- LDAP
- Custom registries

WebSphere Process Server and WebSphere ESB support only the Lightweight Third Party Authentication (LTPA) protocol. The Simple WebSphere Authentication Mechanism (SWAM) protocol is not supported.

Typical clients that are expected to invoke WebSphere Process Server and WebSphere ESB components include Web services clients, Web or HTTP clients, and Java clients. These clients can be authenticated in the following ways:

- Web services clients can use WS-Security/SOAP authentication.
- Web clients reference JSPs, Servlets, or plain HTML documents. You can set up these artifacts for HTTP basic authentication so that the browser prompts for a user name and password when you reference the URL.
- Java clients should use JAAS for authentication.
- All clients can be set up for SSL client authentication.

The WebSphere Process Server and WebSphere ESB runtime has some authentication aliases that are used to authenticate the runtime code for access to databases and messaging engines. The WebSphere Process Server and WebSphere ESB installer collects the user name and passwords and creates these aliases. You can modify the aliases using the Administrative Console (**Security** → **Global Security** → **JAAS Configuration** → **J2C authentication**).

Some runtime components have message driven beans (MDBs) that are configured with a *runAs* role. The installer collects the user name and password for the runAs role.

Table 3-1 and Table 3-2 list some aliases that the WebSphere Process Server and WebSphere ESB installer creates. If these aliases are not set up correctly, the server does not function correctly when security is turned on.

Table 3-1 SCA related aliases

Alias	Description	Notes
SCA_Auth_Alias	Used by runtime to authenticate with the messaging engine	User name and password entered on the SCA configuration panel of the installer

Table 3-2 Common Event Infrastructure related alias

Alias	Description	Notes
CommonEventInfrastructureJMSAuthAlias	Used by runtime to authenticate with the messaging engine	User name and password entered on the CEI configuration panel of the installer
EventAuthAlias<DBType>r	Used by runtime to authenticate with the database	User name and password entered on the CEI configuration panel of the installer

Table 3-3 and Table 3-4 list some aliases that the WebSphere Process Server installer creates.

Table 3-3 BPEL container related authentication alias

Alias	Description	Notes
BPEAuthDataAliasJMS_<node>_<server>	Used by runtime to authenticate with the messaging engine	User name and password entered on the BPEL configuration panel of the installer
BPEAuthDataAlias<DbType>_<node>_<server>	Used by runtime to authenticate with the database	Database configuration is done with BPEL provided scripts

Table 3-4 BPEL container related roles

runAsRole	Description	Notes
JMSAPIUser	Used by the BFM JMS API MDB in bpecontainer.ear	User name and password entered on the BPEL configuration panel of the installer
EscalationUser	Used by the task.ear MDB	User name and password entered on the BPEL configuration panel of the installer

3.2.2 Access control

Access control refers to ensuring that the authenticated user has permission to perform an operation. Some WebSphere Process Server and WebSphere ESB runtime components that are packaged as EAR files secure their operations using J2EE role-based security.

You can also secure components developed by users using the following SCA qualifiers:

- *securityPermission*, where you specify the role that has the permission to invoke the secured method.
- *securityIdentity*, which 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.

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 3-1, access to the one-way invoke method is restricted to users that are members of the role manager.

Example 3-1 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>
```

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 or WebSphere ESB. 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 (OS or LDAP) instead of individual users.

Access control for the BPEL container

The Business Process Execution Language (BPEL) runtime installs the EAR files with access control shown in Table 3-5.

Table 3-5 BPEL components with Access Control

Runtime EAR	Roles	Default permission	Notes
bpecontainer.ear	BPESystemAdministrator	Group name entered during install	Access to all business processes and all operations
bpecontainer.ear	BPESystemMonitor	All Authenticated users	Access to read operations
task.ear	TaskSystemAdministrator	Group name entered during install	Access to all human tasks
task.ear	TaskSystemMonitor	All authenticated users	Access to read operations
bpcexplorer.ear	WebClientUser	All authenticated users	Access to use the BPCExplorer

3.2.3 Access control for the Common Event Infrastructure

Common Event Infrastructure (CEI) installs the EAR files with access control, as shown in Table 3-6.

Table 3-6 CEI components with Access Control

Runtime EAR	Roles	Default permission
EventServer.ear	eventAdministrator	All authenticated users
EventServer.ear	eventConsumer	All authenticated users
EventServer.ear	eventUpdater	All authenticated users
EventServer.ear	eventCreator	All authenticated users
EventServer.ear	catalogAdministrator	All authenticated users
EventServer.ear	catalogReader	All authenticated users

3.2.4 Integrity and privacy

There are two widely used solutions that are supported by WebSphere Process Server and WebSphere ESB for implementing integrity and privacy:

- The Secure Sockets Layer (SSL) protocol uses a *handshake* to authenticate the end points and exchange information that is used to generate the session key that will be used by end points for encryption and decryption. SSL is a synchronous protocol and suitable for point to point communication. SSL requires that the two end points maintain a connection with each other for the duration of the SSL session.
- WS-Security is a standard that defines SOAP extensions for securing SOAP messages. WS-Security adds support for authentication, integrity, and privacy for a single SOAP message. Unlike SSL, there is no handshake to establish a session key, which makes WS-Security suitable for securing messages in an asynchronous environment, such as SOAP over Java Message Service (JMS).

In a business integration environment, several systems are often involved:

- Multiple enterprise applications
- Adapters
- Multiple servers running WebSphere Application Server, WebSphere Process Server, and WebSphere ESB
- Partner systems outside the intranet

The communication could be point to point or publish/subscribe. Often, the communication is asynchronous. With service-oriented architecture (SOA), most systems are likely to be communicating using Web service and SOAP protocols. This communication makes WS-Security the preferred choice for securing data sent over the network. When SOAP is not involved, you might still need to use SSL.

3.2.5 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 Process Server and WebSphere ESB is to integrate Web applications with back-end 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 Process Server and WebSphere ESB. *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 3-2 illustrates trust association between WebSEAL and WebSphere Process Server is established using SSL.

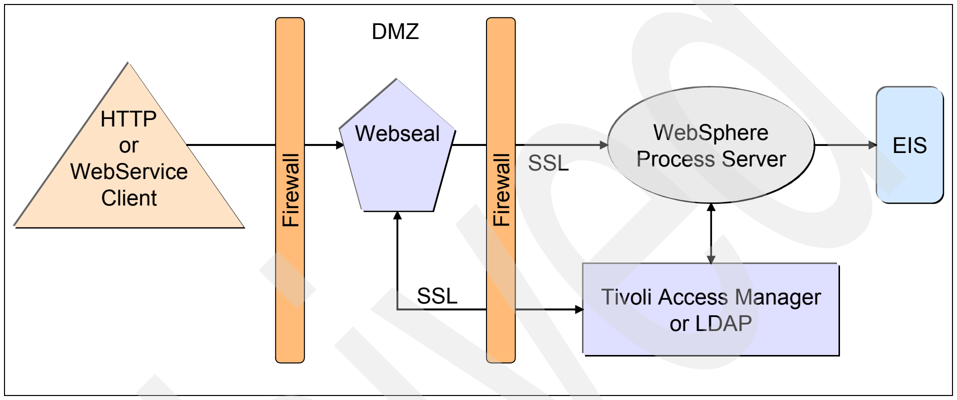


Figure 3-2 Single sign-on

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 Process Server and WebSphere ESB 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 Process Server and WebSphere ESB provide an SPI for developers to create their own custom mapping modules.

4

Production topologies for WebSphere Process Server and WebSphere ESB

When designing a production topology for WebSphere Process Server or WebSphere Enterprise Service Bus (ESB), you can choose to build a topology that supports everything these products have to offer, or you can build a smaller topology that supports just the product components and qualities of service you need.

This chapter introduces three production topologies for the WebSphere Process Server and WebSphere ESB products, each with varying levels of support for product components and qualities of service. You can use this chapter to learn about these topologies and to determine which production topology is right for you.

This chapter includes the following sections:

- ► Overview of three production topologies
- ► Basic topology
- ► Loosely Coupled topology
- ► Full Support topology
- ► Selecting the appropriate production topology

4.1 Overview of three production topologies

This chapter defines three recommended production topologies for WebSphere Process Server and WebSphere ESB configurations:

- Basic topology
- Loosely Coupled topology
- Full Support topology

These topologies differ in terms of the WebSphere Process Server or WebSphere ESB components and the qualities of service (QoS) that they support. Therefore, based on the components that are used for a business solution and QoS requirements of a business solution, only one topology might be the most suitable for that solution. Thus, you should choose the most suitable topology for a given business solution after you understand thoroughly the business solution components and QoS requirements of the solution.

This section details the components that are supported and QoS that is provided by the three production topologies.

4.1.1 Component support offered by the production topologies

Component support is a broad term used to describe the supported components, bindings, and the invocation patterns for those components for WebSphere Process Server and WebSphere ESB.

Table 4-1 summarizes the component support for each production topology.

Table 4-1 Summary of components supported by three production topologies

	Basic topology	**Loosely Coupled topology**	**Full Support topology**
WebSphere ESB / WebSphere Process Server components supported	All components except ► Long-running business processes ► Business state machines ► Human Task Manager ► Business rule manager	All components except ► Business rule manager	All components
Bindings supported (for SCA imports and exports)	► Web service ► SCA	► Web service ► SCA ► Stateless session bean binding	All bindings
Invocation patterns supported	Synchronous invocations only	All invocation patterns except asynchronous, 2-way with deferred response	All invocation patterns
CEI support	No	No	Yes
Security	Yes	Yes	Yes

4.1.2 Quality of service support offered by the production topologies

All the production topologies that we describe in this book offer the *failover* and *load balancing* capabilities that are offered by WebSphere Network Deployment. In addition, these topologies also offer the QoS that is summarized in Table 4-2.

Table 4-2 Summary of QoS support for three production topologies

Name of QoS	**Definition**	***Basic topology***	***Loosely Coupled topology***	***Full Support topology***
Message sequencing	Business solutions that require preservation of the order of service invocations are said to require QoS message sequencing.	Yes	No	Yes
Cluster size tolerance (Refer to 4.5.1, “Decision factors affecting the topology selection” on page 59 for more information.)	Business solutions that do not require the client business processes to be aware of changes to the cluster size changes are said to be cluster size tolerant.	No	No	Yes
Cluster location tolerance (Refer to 4.5.1, “Decision factors affecting the topology selection” on page 59 for more information.)	Business solutions that do not require the client business processes to be aware of changes to the deployment targets on which business applications are installed are said to be cluster location tolerant.	No	No	Yes

Name of QoS	Definition	*Basic topology*	*Loosely Coupled topology*	*Full Support topology*
Constrained hardware usage	In production environment, sometimes customer can buy hardware as desired for topology or sometimes they have to design a topology to take into considerations hardware constraints. Business solutions that are constrained by hardware, either in terms of the number of machines or the number of concurrent processes per machine, are said to require QoS constrained hardware usage.	Yes	Yes	No

The final choice of a production topology for a given business solution depends on the combination of component support and QoS characteristics that are needed by the business solution.

In the following sections, we discuss the architectural details of the three types of topology and discuss the process for selecting the appropriate topology for a given a business solution.

4.2 Basic topology

This section describes the architectural details and a sample Basic topology for the WebSphere Process Server and WebSphere ESB products.

4.2.1 Architectural details for a Basic topology for WebSphere ESB

Figure 4-1 shows architectural details of the Basic topology for WebSphere ESB.

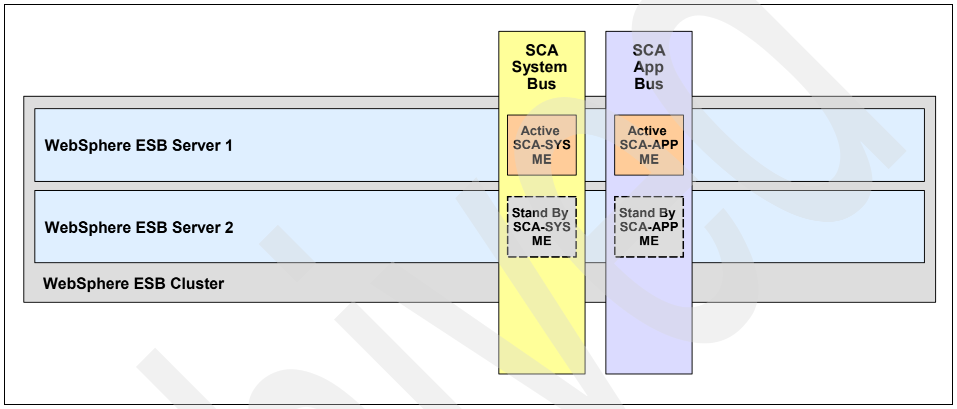


Figure 4-1 Basic topology for WebSphere ESB

The key internal architectural details of this topology are:

- The topology is comprised of only one cluster in a cell.
- The cluster is the only member of the SCA-System and SCA-Application buses.
- Business applications and messaging engines reside together inside same application server in the cluster. Therefore, there is no clear distinction between the messaging layer and the application layer.
- There is only one active messaging engine on the SCA-System and SCA-Application buses:
  - By default, this active messaging engine starts on one of the application servers inside the cluster.
  - The active messaging engine has a stand-by messaging engine on each of the remaining application servers within the cluster. When the active messaging engine stops, the high availability (HA) manager activates one of the remaining stand-by messaging engines.

In this topology, because there are only synchronous invocations, the messaging engines are not used for communication. Therefore, the business solution designer should not be concerned about internal architectural details about messaging engines and buses.

Sample Basic topology for WebSphere ESB

Figure 4-2 shows a sample Basic WebSphere ESB topology. This sample Basic topology for WebSphere ESB is comprised of a single cluster into which the WebSphere mediation module is deployed.

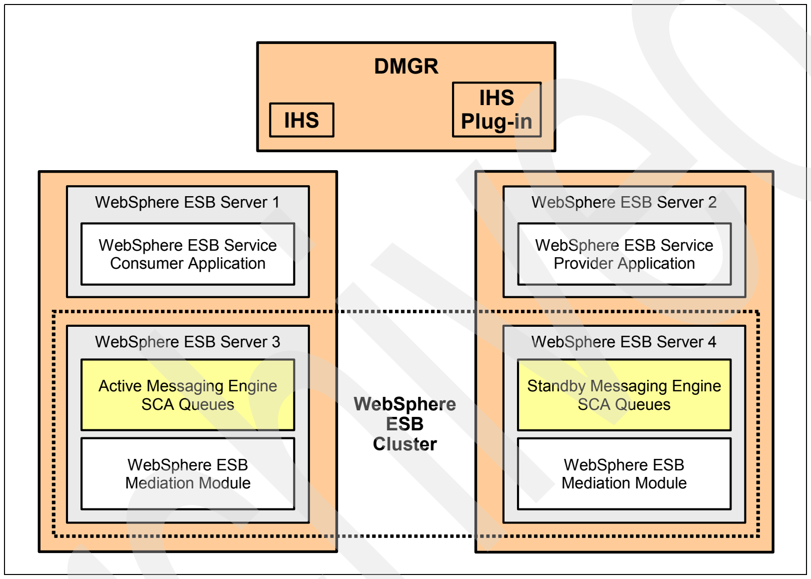

DMGR
IHS
IHS Plug-in
WebSphere ESB Server 1
WebSphere ESB Service Consumer Application
WebSphere ESB Server 2
WebSphere ESB Service Provider Application
WebSphere ESB Server 3
Active Messaging Engine SCA Queues
WebSphere ESB Mediation Module
WebSphere ESB Cluster
WebSphere ESB Server 4
Standby Messaging Engine SCA Queues
WebSphere ESB Mediation Module

Figure 4-2 Sample Basic topology for WebSphere ESB

The WebSphere ESB service provider and WebSphere ESB service consumer applications can be deployed either into the same cell or into separate cells. In the sample topology, one singleton WebSphere ESB server is used to host each of these applications. In the sample topology, the consumer application, WebSphere mediation module, and service provider applications are all Web services and communicate with each other through SOAP/HTTP. *WebSphere ESB Server 1* and *WebSphere ESB Server 2*, as shown in Figure 4-2, are not added to the SCA-System and SCA-Application buses.

The service consumer application invokes the application within the WebSphere ESB mediation module synchronously. Similarly, the mediation module invokes the service provider application synchronously.

Note: The Basic topology for WebSphere ESB is implemented in Chapter 6, “Implementing the Basic topology for WebSphere ESB” on page 93.

4.2.2 Architectural details for a Basic topology for WebSphere Process Server

Figure 4-3 shows architectural details of the Basic topology for WebSphere Process Server.

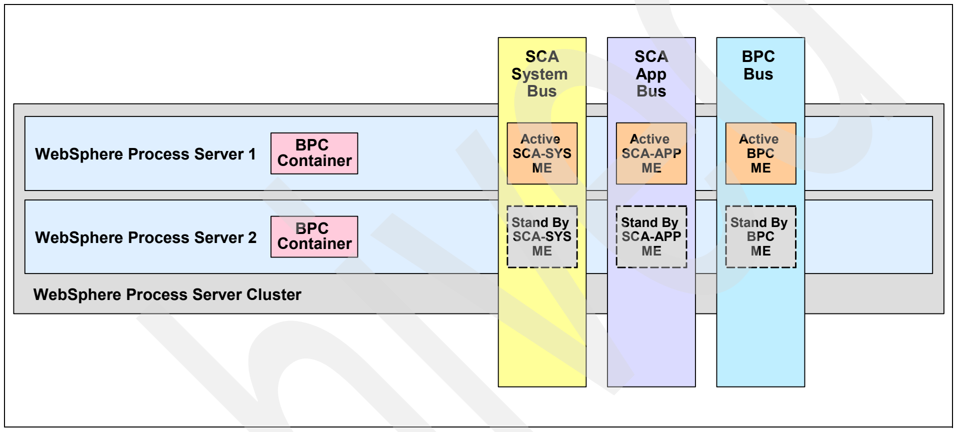


Figure 4-3 Basic topology for WebSphere Process Server

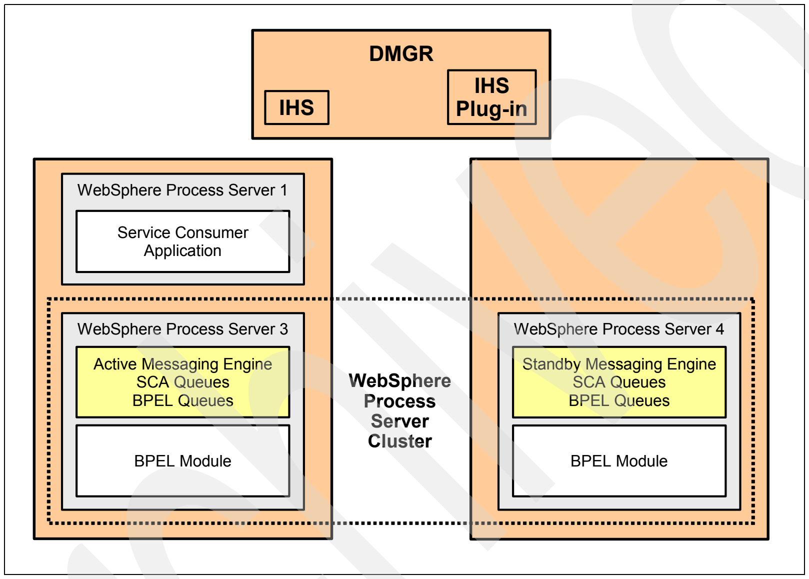
The key internal architectural details of this topology are:

- The topology is comprised of only one cluster in a cell.
- The cluster is the only member of the SCA-System, SCA-Application and BPC buses.
- Business applications and messaging engines reside together inside the same application server within the cluster. Therefore, there is no clear distinction between the messaging layer and the application layer.
- There is only one active messaging engine on the SCA-System, SCA-Application and BPC buses:
  - By default, this active messaging engine starts on one of the Application Servers inside the cluster.
  - The active messaging engine has a stand-by messaging engine on each of the remaining application servers within the cluster. When the active messaging engine stops, the HA manager activates one of the remaining stand-by messaging engines.

In this topology as there are only synchronous invocations, therefore the messaging engines are not used for communication. Therefore the business solution designer should not be concerned about internal architectural details about messaging engines and buses.

Sample Basic topology for WebSphere Process Server

Figure 4-4 shows a sample Basic topology for WebSphere Process Server. This sample Basic topology is comprised of a single cluster into which the business process applications are deployed.



DMGR
IHS
IHS Plug-in
WebSphere Process Server 1
Service Consumer Application
WebSphere Process Server 3
Active Messaging Engine
SCA Queues
BPEL Queues
BPEL Module
WebSphere Process Server Cluster
WebSphere Process Server 4
Standby Messaging Engine
SCA Queues
BPEL Queues
BPEL Module

Figure 4-4 Sample Basic topology for WebSphere Process Server

The BPEL container is configured on this WebSphere Process Server cluster, and the cluster is a member of the SCA-System, SCA-Application and BPC buses.

Client applications that invoke business applications within the cluster can be either deployed into the same cell or into separate cells. In the sample topology, one singleton WebSphere Process server hosts these client applications. These applications invoke the business applications within the cluster synchronously using SOAP/HTTP. *WebSphere Process Server 1*, as shown in Figure 4-4, is not added to the SCA-System, SCA-Application or BPC buses.

4.3 Loosely Coupled topology

This section describes the architectural details and a Loosely Coupled sample topology for the WebSphere Process Server and WebSphere ESB products.

4.3.1 Architectural details for a Loosely Coupled topology for WebSphere ESB

Figure 4-5 gives architectural details of the Loosely Coupled topology for WebSphere ESB.

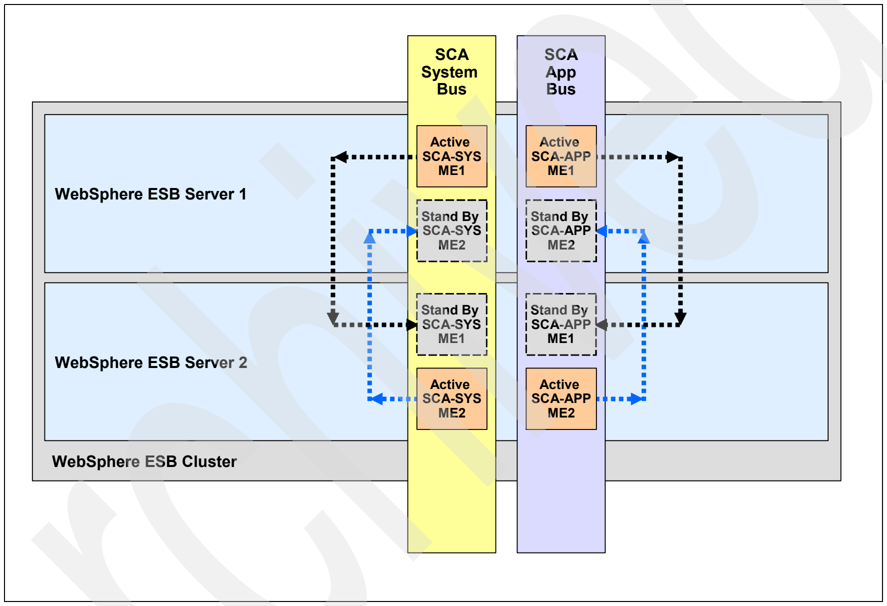


Figure 4-5 Loosely Coupled topology for WebSphere ESB

The key internal architectural details of this topology are:

- The topology makes use of destination partitioning features that are supported by the service integration bus (SI Bus).
- The topology can include one or more application clusters.
- Every cluster is a member of the SCA-System and SCA-Application buses.
- Business applications and messaging engines reside together inside the same application server within the cluster. (No distinction between messaging layer and application layer for a solution.)

- There are multiple active messaging engines for every bus within the cluster:
  - By default, an active messaging engine starts on every application server within the cluster for each bus (SCA-System and SCA-Application).
  - Each active messaging engine has a stand-by messaging engine on the other application servers within the cluster. When an active messaging engine stops, the HA manager activates one of the remaining stand-by messaging engines.

Sample topology for a Loosely Coupled topology for WebSphere ESB

The sample Loosely Coupled topology for WebSphere ESB is comprised of one or more clusters. In Figure 4-6, there are two clusters.

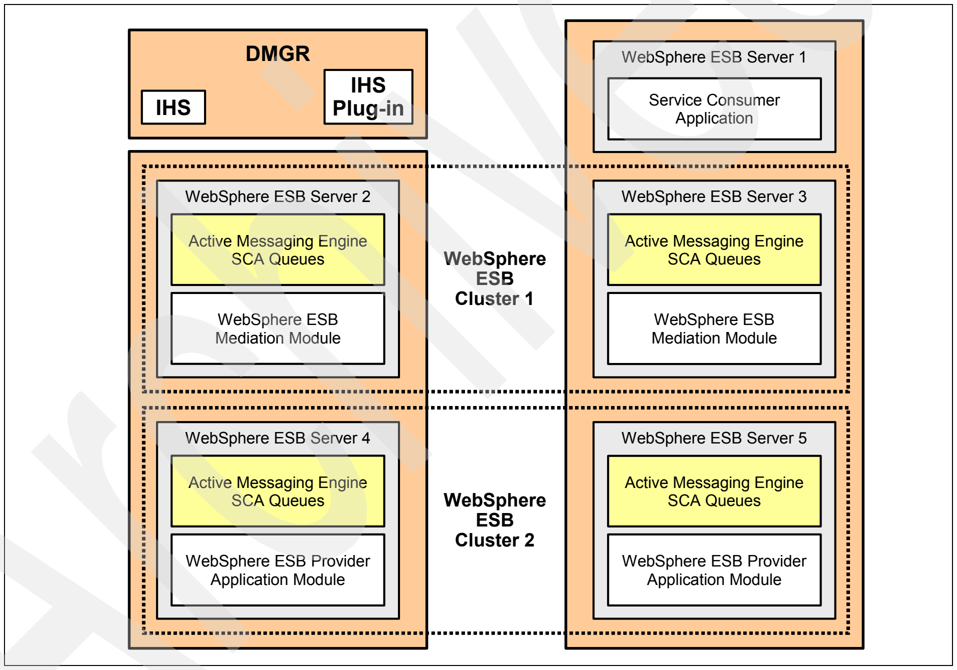

DMGR
IHS
IHS Plug-in
WebSphere ESB Server 1
Service Consumer Application
WebSphere ESB Server 2
Active Messaging Engine SCA Queues
WebSphere ESB Mediation Module
WebSphere ESB Server 3
Active Messaging Engine SCA Queues
WebSphere ESB Mediation Module
WebSphere ESB Cluster 1
WebSphere ESB Server 4
Active Messaging Engine SCA Queues
WebSphere ESB Provider Application Module
WebSphere ESB Server 5
Active Messaging Engine SCA Queues
WebSphere ESB Provider Application Module
WebSphere ESB Cluster 2

Figure 4-6 Sample Loosely Coupled topology for WebSphere ESB

The service provider and service consumer applications can be deployed into the same cell or into separate cells. In Figure 4-6, the service provider application is deployed into the WebSphere ESB cluster, and the services consumer application is deployed onto a singleton server. The service consumer application and WebSphere ESB mediation module are Web services. These services communicate with each other through SOAP/HTTP and invocation occurs synchronously.

The WebSphere ESB mediation module and WebSphere ESB provider application communicate with each other using SCA bindings. The communication method is asynchronous, 2-way with callback.

The *WebSphere ESB Cluster 1* and *WebSphere ESB Cluster 2* are added to the SCA-System and SCA-Application. The *WebSphere ESB Server 1* is not a member of any of the buses.

4.3.2 Architectural details for a Loosely Coupled topology for WebSphere Process Server

Figure 4-7 gives architectural details of the Loosely Coupled topology for WebSphere Process Server.

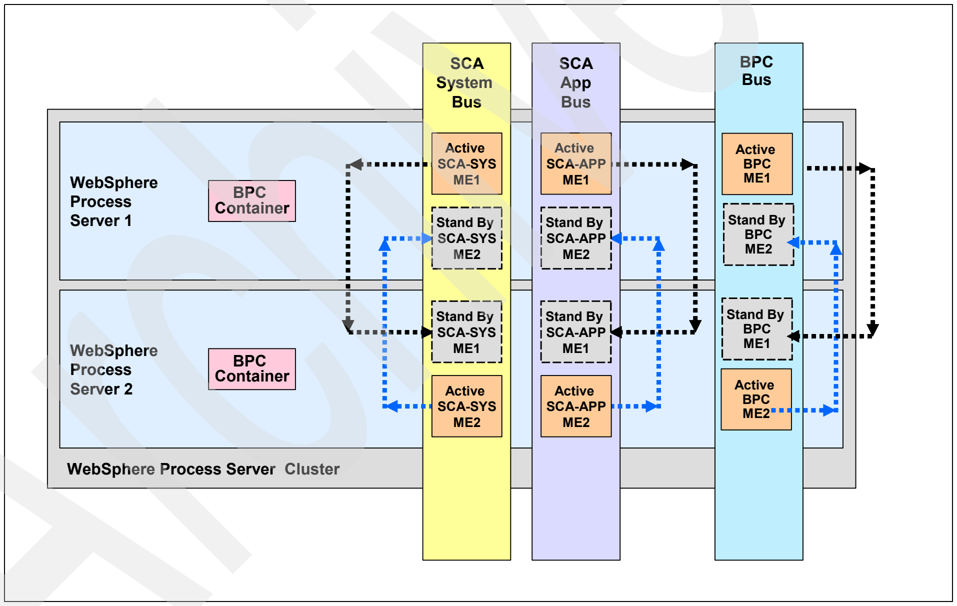


Figure 4-7 Loosely Coupled topology for WebSphere Process Server

The key internal architectural details of this topology are:

- The topology makes use of destination partitioning features that are supported by the SI Bus for the SCA-System and SCA-Application and the BPC buses.
- The topology can include one or more application clusters.
- Every cluster is a member of the SCA-System, SCA-Application, and BPC buses.
- Business applications and messaging engines reside together inside each application server within the cluster. Therefore, there is no clear distinction between the messaging layer and the application layer.
- For the SCA-System, SCA-Application, and BPC buses within each cluster, there is an active messaging engine on each cluster member application server. Destination partitioning features are used for the SCA-System and SCA-Application and BPC bus:
  - By default, an active messaging engine starts on *every* application server within the cluster for both the SCA-System and SCA-Application buses.
  - Each active messaging engine has a stand-by messaging engine on the other application servers within the cluster. When the active messaging engine stops, the HA manager activates one of the remaining stand-by messaging engines.

Sample topology for a Loosely Coupled topology for WebSphere Process Server

The Loosely Coupled topology for WebSphere Process Server is comprised of one or more clusters. In Figure 4-8, there are two clusters. Client processes that consume services inside the WebSphere Process Server cluster can be deployed into a single cell or into separate cells.

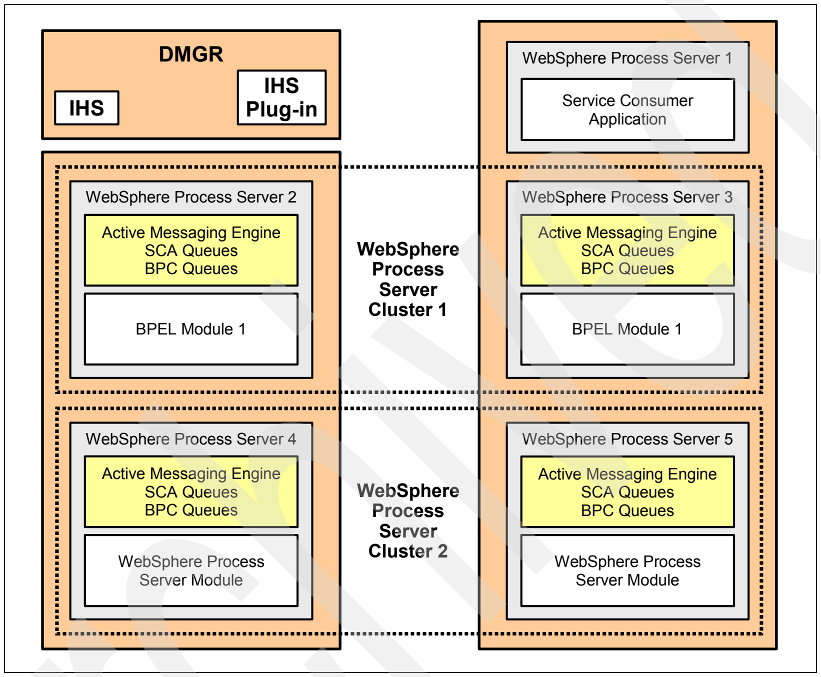

DMGR
IHS
IHS Plug-in
WebSphere Process Server 1
Service Consumer Application
WebSphere Process Server 2
Active Messaging Engine
SCA Queues
BPC Queues
BPEL Module 1
WebSphere Process Server Cluster 1
WebSphere Process Server 3
Active Messaging Engine
SCA Queues
BPC Queues
BPEL Module 1
WebSphere Process Server 4
Active Messaging Engine
SCA Queues
BPC Queues
WebSphere Process Server Module
WebSphere Process Server Cluster 2
WebSphere Process Server 5
Active Messaging Engine
SCA Queues
BPC Queues
WebSphere Process Server Module

Figure 4-8 Sample Loosely Coupled topology for WebSphere Process Server

Client applications are Web services. They invoke BPEL processes using SOAP/HTTP. The interaction style is synchronous.

The BPEL process within *Cluster 1* invokes other applications inside the WebSphere Process server *Cluster 2* using SCA bindings. The invocation method used is asynchronous, 2-way with callback.

The *WebSphere Process Server Cluster 1* and *WebSphere Process Server Cluster 2* are both added to the SCA-System, SCA-Application, and BPC buses. The *WebSphere Process Server 1* is not member of any of the buses.

Note: The Loosely Coupled topology for WebSphere Process Server is implemented in Chapter 8, “Implementing the Loosely Coupled topology for WebSphere Process Server” on page 231.

4.4 Full Support topology

This section describes the architectural details and a sample scenario for the Full Support topology for WebSphere Process Server and WebSphere ESB.

4.4.1 Architectural details for Full Support topology for WebSphere ESB

Figure 4-9 shows the architectural details of the Full Support topology for WebSphere ESB.

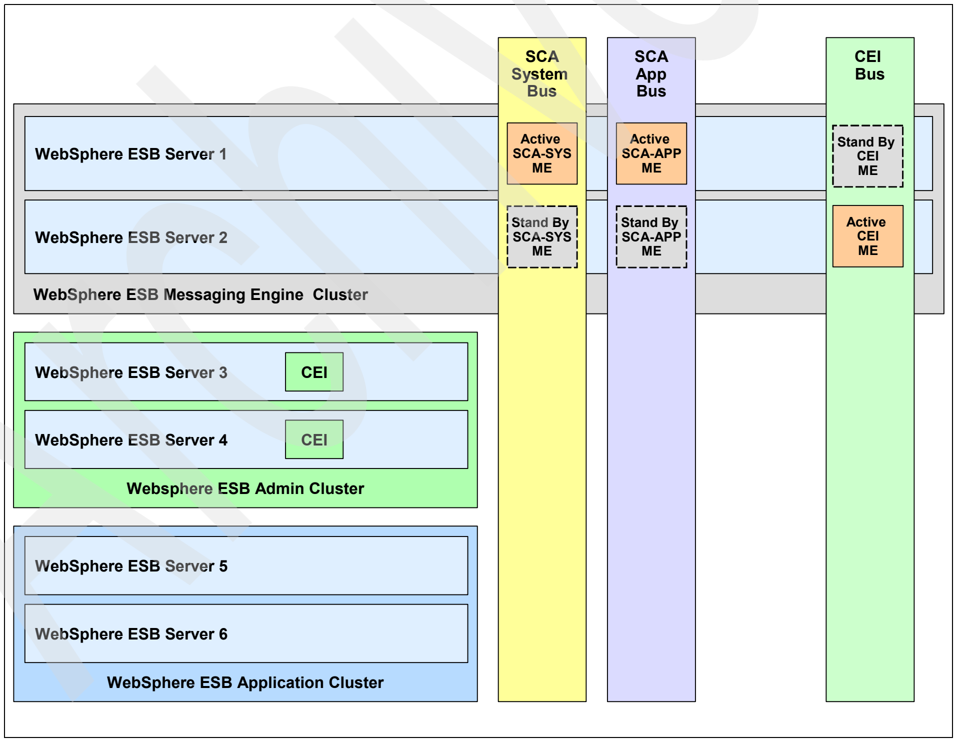


Figure 4-9 Full Support topology for WebSphere ESB

The key internal architectural details of the topology are:

- The topology is comprised of one messaging engine cluster, one support cluster, and one or more application clusters.
- The messaging engine cluster is the only member of the SCA-System, SCA-Application, and CEI buses.
- Business applications and messaging engines reside in separate application servers within the cluster. Therefore, there is a clear distinction between the messaging layer and the application layer.
- There is only one active messaging engine on the SCA-System, SCA-Application, and CEI buses. This messaging engine is located within the messaging engine cluster:
  - By default, an active messaging engine starts on one of the application servers inside the messaging engine cluster.
  - The active messaging engine has stand-by messaging engines on each of the other application servers within the messaging engine cluster. When an active messaging engine stops, the HA manager activates one of the remaining stand-by messaging engines.

Sample topology for Full Support topology for WebSphere ESB

Figure 4-10 shows a sample Full Support topology for WebSphere ESB. The topology is comprised of one messaging engine cluster, one support cluster and one or more WebSphere ESB application clusters.

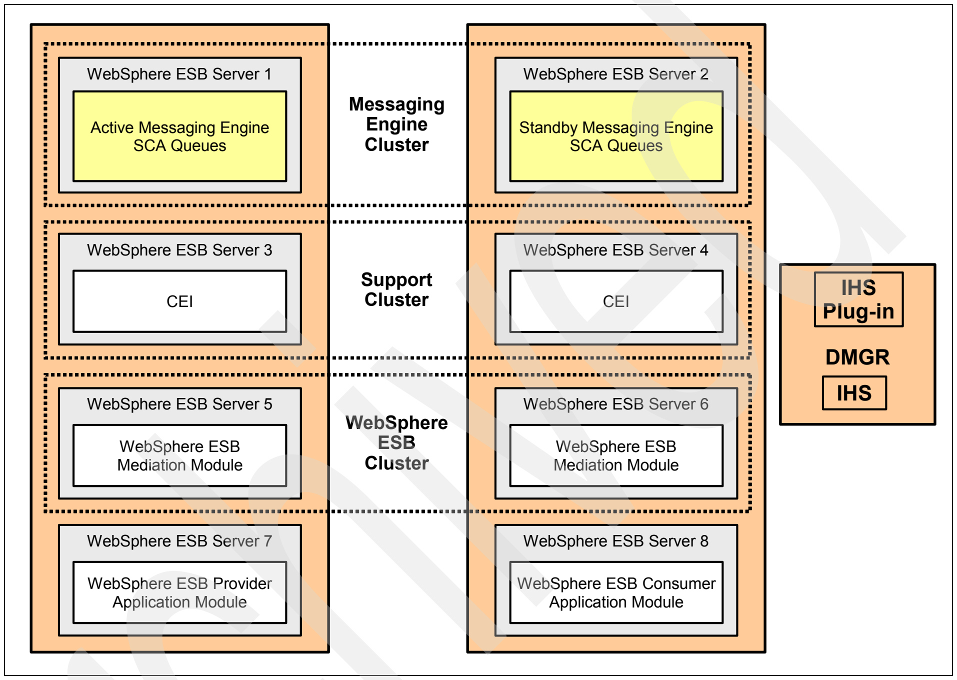


Figure 4-10 Sample Full Support topology for WebSphere ESB

The messaging engine cluster includes one or more WebSphere ESB servers. A single server hosts the active messaging engines for each of the SCA-Application, SCA-System, and CEI buses.

The support cluster includes the CEI-related applications and supports the asynchronous method for querying events.

The WebSphere ESB cluster includes the mediation modules. The service provider applications and the service consumer applications are installed on two separate singleton servers.

In the sample topology, the service consumer application uses MQ-JMS bindings. The consumer application invokes a WebSphere ESB mediation module asynchronously with 2-way, deferred response call.

The WebSphere ESB mediation module invokes the provider application using the SCA binding with a synchronous call.

The messaging engine cluster is the only member of the SCA-System, SCA-Application and CEI buses in this topology. All other servers and clusters refer to this remote messaging engine cluster for their messaging requirements.

Note: The Full Support topology for WebSphere ESB is implemented in Chapter 7, "Implementing the Full Support topology for WebSphere ESB" on page 151.

4.4.2 Architectural details for Full Support topology for WebSphere Process Server

Figure 4-11 shows the architectural details of the Full Support topology for WebSphere Process Server.

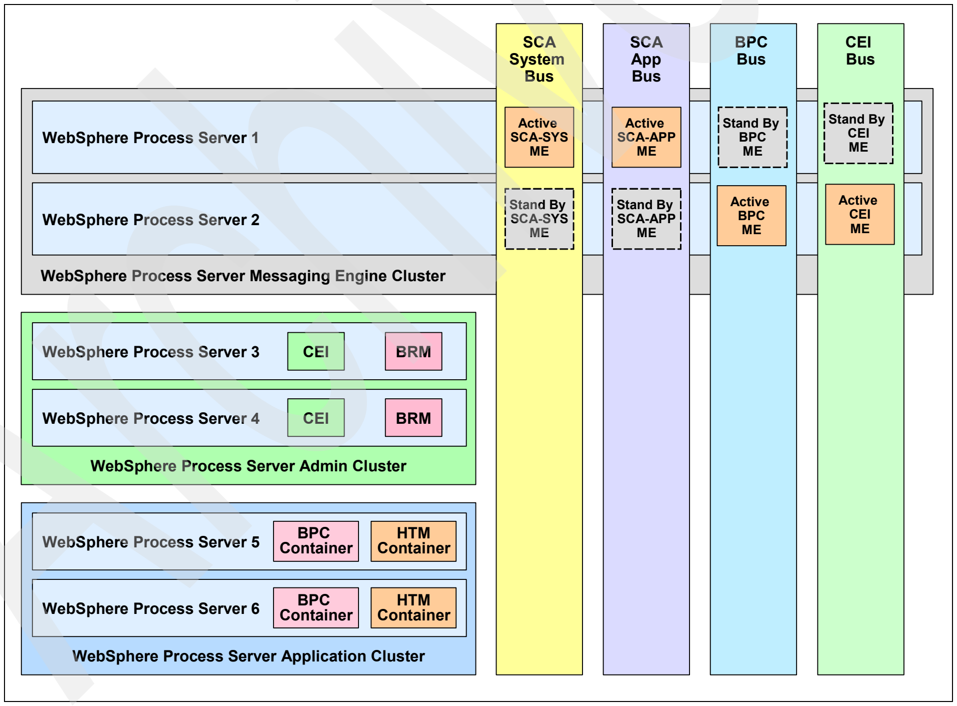

SCA System Bus
SCA App Bus
BPC Bus
CEI Bus
WebSphere Process Server 1
Active SCA-SYS ME
Active SCA-APP ME
Stand By BPC ME
Stand By CEI ME
WebSphere Process Server 2
Stand By SCA-SYS ME
Stand By SCA-APP ME
Active BPC ME
Active CEI ME
WebSphere Process Server Messaging Engine Cluster
WebSphere Process Server 3
CEI
BRM
WebSphere Process Server 4
CEI
BRM
WebSphere Process Server Admin Cluster
WebSphere Process Server 5
BPC Container
HTM Container
WebSphere Process Server 6
BPC Container
HTM Container
WebSphere Process Server Application Cluster

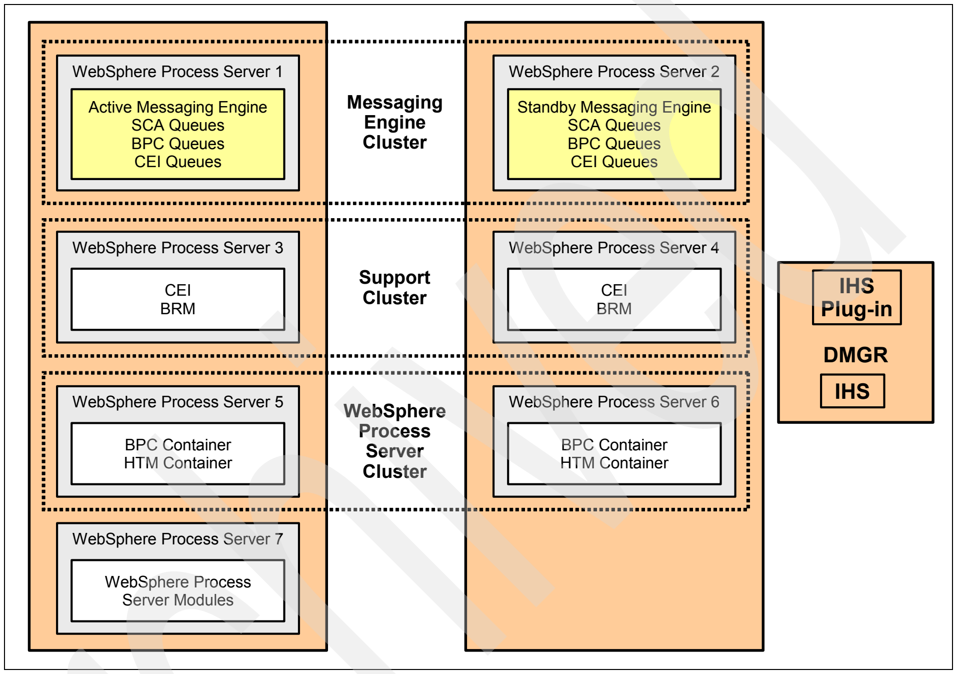
Figure 4-11 Full Support topology for WebSphere Process Server

The key internal architectural details of the topology are:

- The topology is comprised of one messaging engine cluster, one support cluster, and one or more application clusters.
- The messaging engine cluster is the only member of the SCA-System, SCA-Application, BPC, and CEI buses.
- The business applications and the messaging engines reside in different application servers within the cluster. Therefore, there is a clear distinction between the messaging and the application layer.
- There is only one active messaging engine on the SCA-System, SCA-Application, BPC and CEI buses. This messaging engine is located within the messaging engine cluster:
  - By default, an active messaging engine starts on one of the application servers inside the messaging engine cluster.
  - The active messaging engine has a stand-by messaging engine on each of the other application servers within the messaging engine cluster. When the active messaging engine stops, the HA manager activates one of the remaining stand-by messaging engines.

Sample topology for Full Support topology for WebSphere Process Server

Figure 4-12 shows a sample Full Support topology for WebSphere Process Server. The topology is comprised of one messaging engine cluster, one support cluster and one or more WebSphere Process Server application clusters.



WebSphere Process Server 1
Active Messaging Engine
SCA Queues
BPC Queues
CEI Queues
Messaging Engine Cluster
WebSphere Process Server 2
Standby Messaging Engine
SCA Queues
BPC Queues
CEI Queues
WebSphere Process Server 3
CEI
BRM
Support Cluster
WebSphere Process Server 4
CEI
BRM
IHS Plug-in
DMGR
IHS
WebSphere Process Server 5
BPC Container
HTM Container
WebSphere Process Server Cluster
WebSphere Process Server 6
BPC Container
HTM Container
WebSphere Process Server 7
WebSphere Process Server Modules

Figure 4-12 Sample Full Support Topology for WebSphere Process Server

The messaging engine cluster includes one or more WebSphere Process Server members. A single member server hosts the active messaging engines for each of the SCA-Application, SCA-System, BPC, and CEI buses.

The support cluster includes the CEI-related and business rules manager-related applications. The cluster supports asynchronous calls for querying events.

In this example, there is one WebSphere Process Server application cluster. The BPC and HTM containers are configured within this cluster, and business processes are deployed within this cluster. The solution includes long-running BPEL flows and interactions with human tasks.

In addition, the topology includes one singleton WebSphere Process Server containing the client applications for invoking business processes. The singleton server and WebSphere Process Server cluster refer to the remote messaging engine cluster for their messaging needs.

The client application that is running on the singleton server invokes business processes using SCA bindings. The invocation is made asynchronously with a call back mechanism.

Note: The Full Support topology for WebSphere Process Server with security is implemented in Chapter 9, “Implementing the Full Support topology for WebSphere Process Server” on page 319.

4.5 Selecting the appropriate production topology

This section discusses the selection criteria when choosing the appropriate production topology for a given business solution.

4.5.1 Decision factors affecting the topology selection

Factors that affect the topology selection can be broadly classified into the following categories:

- **Component support type of factors**

  Table 4-1 summarizes the components, bindings, and invocation patterns that are supported by each topology.

- **QoS type of factors**

  Table 4-2 summarizes the QoS levels that each topology offers.

Most of the production environments are not static. When demand for a service grows, you might need to purchase additional hardware or replace the hardware that has failed. Therefore, some business solutions need QoS that requires service clients to be unaware of such environment alterations. We discuss these requirements in the following sections.

Cluster size tolerance

Sometimes, it might be necessary to modify the size of a cluster in a production environment. For a topology to be tolerant of changes to the cluster size, client applications should require no notice or modification that the cluster has changed. A production environment that requires client processes to re-establish their connections to the cluster is said to be *cluster size intolerant*.

Figure 4-13 illustrates cluster *size* tolerance and intolerance.

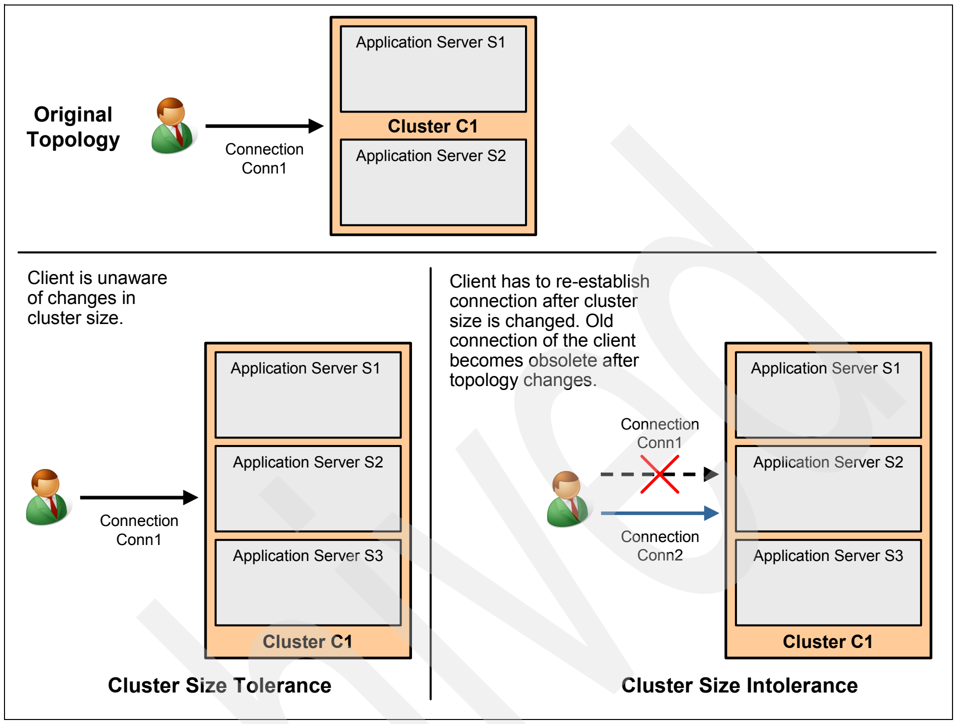


Figure 4-13 Cluster size tolerance

Cluster location tolerance

Sometimes, it might be necessary to modify the location of cluster members in a production environment. For a topology to be tolerant of changes to the location of cluster members, client applications should require no notice or modification that the cluster has changed. A production environment that requires client processes to re-establish their connections to the cluster is said to be *cluster location intolerant*.

Figure 4-14 illustrates cluster *location* tolerance and intolerance.

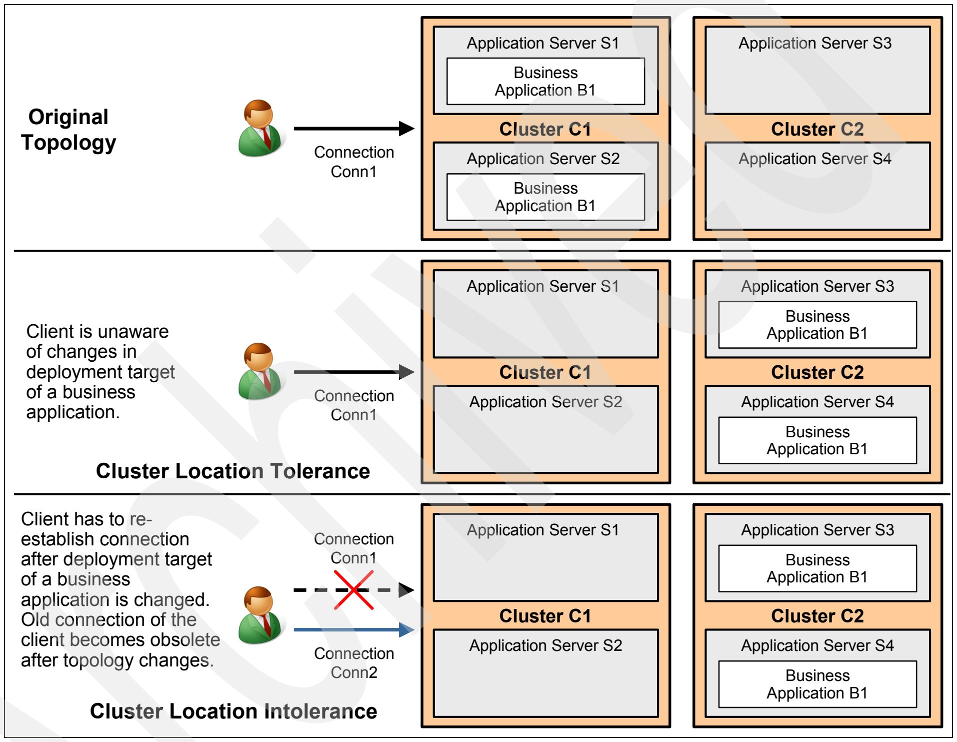
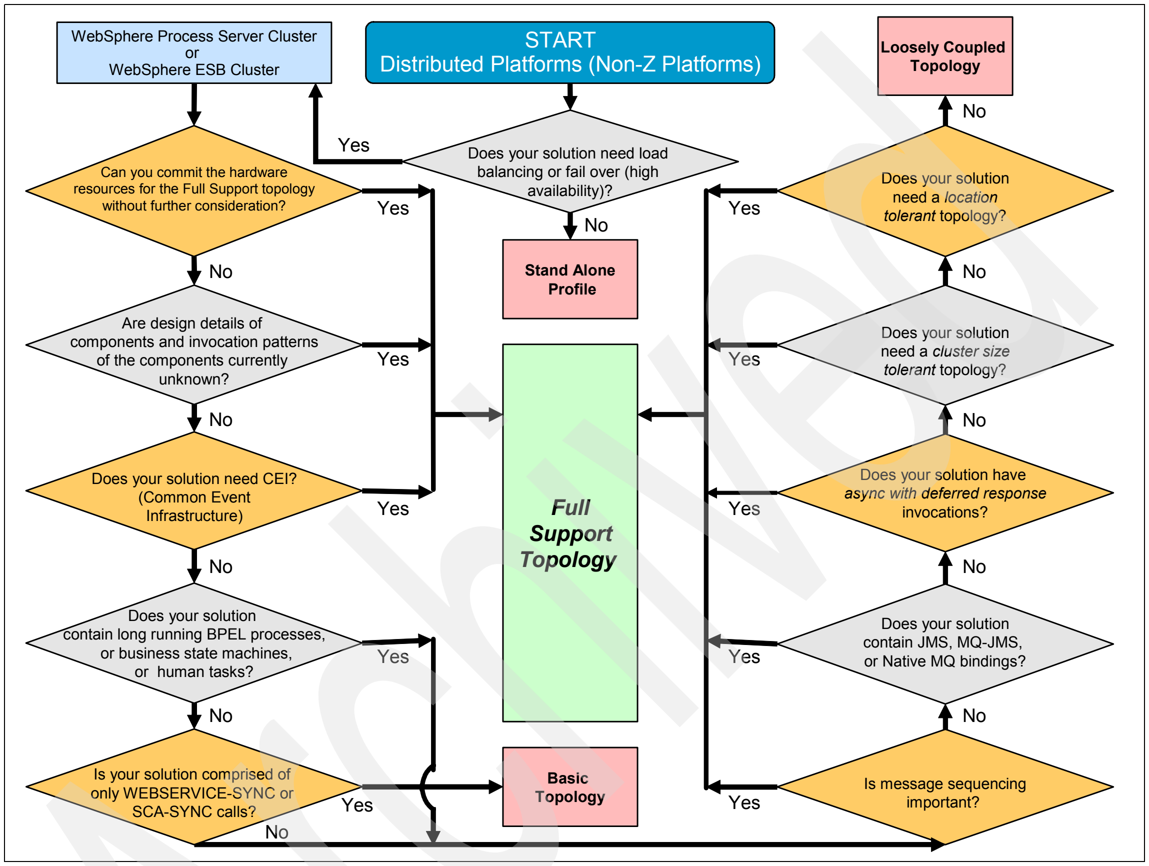


Figure 4-14 Cluster location tolerance

4.5.2 Topology selection flow chart

Figure 4-15 provides a flow chart for selecting the most appropriate production topology for a given business solution.



WebSphere Process Server Cluster or WebSphere ESB Cluster
START Distributed Platforms (Non-Z Platforms)
Loosely Coupled Topology
Does your solution need load balancing or fail over (high availability)?
Yes
No
Stand Alone Profile
Can you commit the hardware resources for the Full Support topology without further consideration?
Yes
No
Are design details of components and invocation patterns of the components currently unknown?
Yes
No
Does your solution need CEI? (Common Event Infrastructure)
Yes
No
Does your solution contain long running BPEL processes, or business state machines, or human tasks?
Yes
No
Is your solution comprised of only WEBSERVICE-SYNC or SCA-SYNC calls?
Yes
No
Full Support Topology
Basic Topology
Does your solution need a location tolerant topology?
Yes
No
Does your solution need a cluster size tolerant topology?
Yes
No
Does your solution have async with deferred response invocations?
Yes
No
Does your solution contain JMS, MQ-JMS, or Native MQ bindings?
Yes
No
Is message sequencing important?
Yes

Figure 4-15 Flow chart for selecting the correct production topology for a business solution

Note: This flow chart applies only to distributed platforms. We do not address production topologies for System z™ in this book.

4.6 Summary

To summarize our information about the production topologies or WebSphere Process Server or WebSphere Enterprise Service Bus:

- A non-clustered *stand-alone configuration* is sufficient for solutions in which failover and load-balancing are not required.
- *Basic topology* is a simple clustered solution, but in some circumstances this topology might be sufficient for your needs.
- *Loosely Coupled topology* should be chosen carefully by taking into consideration various conditions that are shown in Figure 4-15 on page 62.
- *Full Support topology* is the most capable and commonly used topology.

You can use the flow chart in Figure 4-15 on page 62 to select an appropriate topology for mediation scenarios that are running in WebSphere ESB and business process scenarios that are running in WebSphere Process Server.

5

Scenarios that we use in this book

In this chapter, we introduce the two business scenarios that we use in this book:

- The first business scenario describes a simple ITSO Bank application that implements mediation functionalities between two different bank departments.
- The second scenario describes a simple business process application where the account closure process uses several existing ITSO enterprise services.

We present each scenario in two different versions: a simple setup and a more complex setup.

We deploy these solutions to the topologies that we describe in this book. The target runtime for the mediation scenario is WebSphere Enterprise Service Bus, and the target runtime for the business process solution is WebSphere Process Server.

5.1 Mediation scenario context

This section introduces the ITSO Bank mediation scenarios. We describe these scenarios from the business and the IT perspectives. We simplified this business scenario for the purposes of this book.

The ITSO Bank is an imaginary U. S. company in our business scenario. As the ITSO Bank grows, the business requirements expand. The requirement to support potential growth defines which topology is most appropriate.

The ITSO Bank has grown from a local to a large business, and the customer requirements have become more stringent. The growth has come from the acquisition of an English bank, which means that two separate and overlapping stock brokers must be maintained—one for the New York and one for London.

Each stock broker has a different interface to accept orders from the Bank. ITSO Bank customers can place stock orders through a traditional trading Web application. The order management process can use either stock broker to fulfill customer requests.

The mediation issues that we describe in this section are not unique to the finance industry or to company mergers. For any company, continual business change and organizational consolidation can result in a myriad of hardware, software, and application components.

The interfaces between these components are mediated by business logic components that must be flexible and dynamic. The finance industry integration scenario that we present is an example that can be applied to other industries.

Figure 5-1 gives a high-level overview of the mediation scenario.

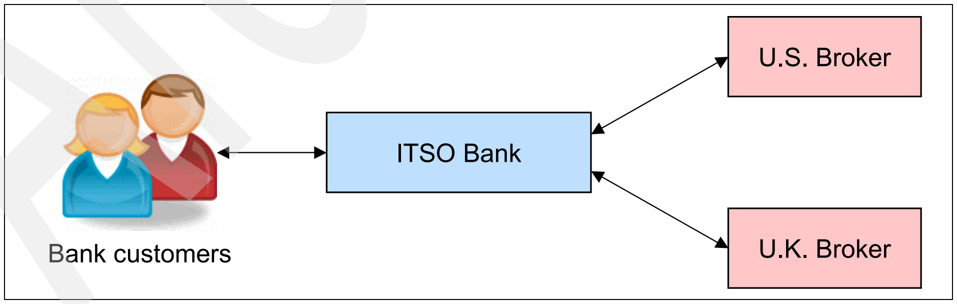


Figure 5-1 High-level overview of mediation scenario

5.1.1 Business goals

The following statements are true for stock orders:

- Each stock broker is more familiar with the local stock market.
- Each broker can manage the local market more rapidly.
- The purchasing costs applied by the stock broker are lower locally.

By integrating their stock ordering processes ITSO Bank plans to:

- Reduce costs by reducing the staff workload that is associated with placing stock orders with the target stock broker (New York for U. S stocks and London for U. K. stocks)
- Increase customer satisfaction by decreasing the costs for the stock order.

The ITSO Bank moves to an on demand environment, where only one stock order service is provided and requests are redirected to the most appropriate broker according to the stock location.

Any stock requests outside the U. S. or U. K. are managed by the U. S. stock broker. In the complex scenario, this is considered an exception, and it is handled by an *Event Emitter*. The Event Emitter mediation primitive sends out business events, during a mediation flow. The events are generated in the form of Common Base Events (CBE) and are sent to a Common Event Infrastructure (CEI) server via the SCA infrastructure which provides basic event management services, such as event generation, transmission, persistence, and consumption.

Note: The Event Emitter is a new mediation primitive available with WebSphere Integration Developer 6.0.2. In Appendix D, “New product features used in the sample applications” on page 455 we describe how this primitive has been used in our sample application.

Two different levels of complexity are provided for the mediation business scenario: a simple and complex.

- In the initial phase, the ITSO Bank requires a standard quality of service. The solution provided is tested by bank employees in a production environment. In this first phase, the event emitting and logging capabilities are not business requirements.

  The bank gains experience with the mediation technologies in a production environment with limited investment in hardware and systems management skills.
- The ITSO Bank decides to move to the complex scenario after testing the solution and proving that it meets the business goals. For the complex

mediation scenario the solution also has to meet other requirements, such as availability, performance, scalability, and event logging.

5.1.2 Simple mediation scenario

Note: We use the simple mediation application as the sample scenario that is deployed to the Basic topology for WebSphere ESB. We describe this scenario in Chapter 6, “Implementing the Basic topology for WebSphere ESB” on page 93.

The *best broker decision* is driven by specific business logic. The intention of the simple mediation scenario is to isolate this logic inside a mediation module. The module is composed of:

- One mediation flow that implements the logic to choose the best stock broker
- One interface to make the mediation functionalities available to web trading application
- Two interfaces for importing the remote broker services

Figure 5-2 shows how we implement the simple scenario:

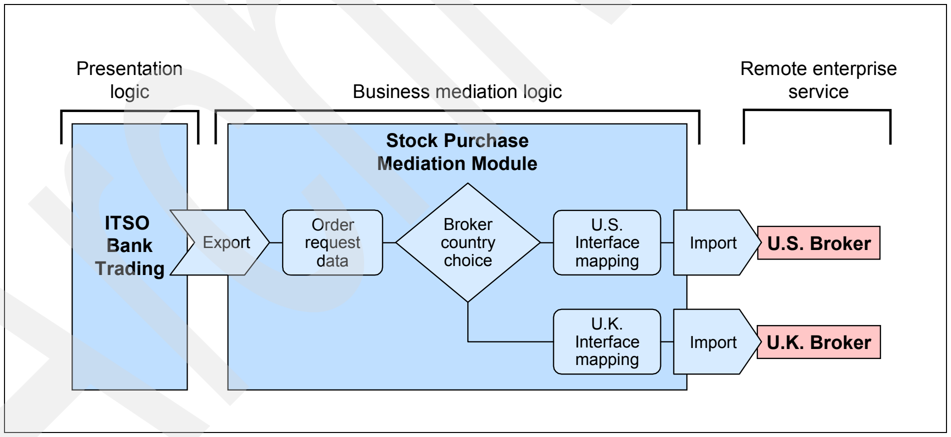


Figure 5-2 Simple mediation scenario used for WebSphere ESB topology patterns

The ITSO Bank mediation solution is composed of three modules:

- Presentation module, including the presentation logic

  This is a simple representation of a user interface for the ITSO trading application. This interface is used to invoke the mediation module by submitting a stock order request. The target runtime environment for this application is WebSphere Application Server.

- Mediation module, including the mediation logic

  The mediation module contains: a simple SCA component to implement the mediation logic, the interfaces for importing remote broker services and the interface to make the mediation logic available to the Web trading application. The target runtime environment for this module is WebSphere ESB.

- Service module, including the remote broker services

  This is a simple EAR application containing the two broker services. These services are invoked by the mediation module using a Web services binding. The target runtime environment for this application is WebSphere Application Server.

Mediation modules are created using WebSphere Integration Developer and deployed to WebSphere ESB through an EAR file.

For this mediation scenario, the ITSO Bank mediation module implements the following functionality:

- Determination of the most suitable stock broker to fulfill the order
- Interface mapping logic between the bank and the stock brokers

In this scenario, the interfaces between the trading web application, the mediation module and the remote services are invoked through Web service (SOAP/HTTP).

Example use of the sample application

This scenario is shipped in three different EAR files:

- ITSOESBPresApp: This application includes the simple user interface to the mediation module (presentation layer).
- ITSOESBBusApp: This application includes the mediation logic (business layer).
- ITSOESBSrvApp: This application includes the two services which represent the U. S. and U. K. brokers (service layer).

These files are available in the additional material that accompanies this book. For more information about how to obtain this additional material, see Appendix A, “Additional material” on page 427.

You can import all WSDL interface descriptions and the project files into WebSphere Integration Developer using Project Interchange file capabilities. Then, you can browse the Web service using the Web service editor.

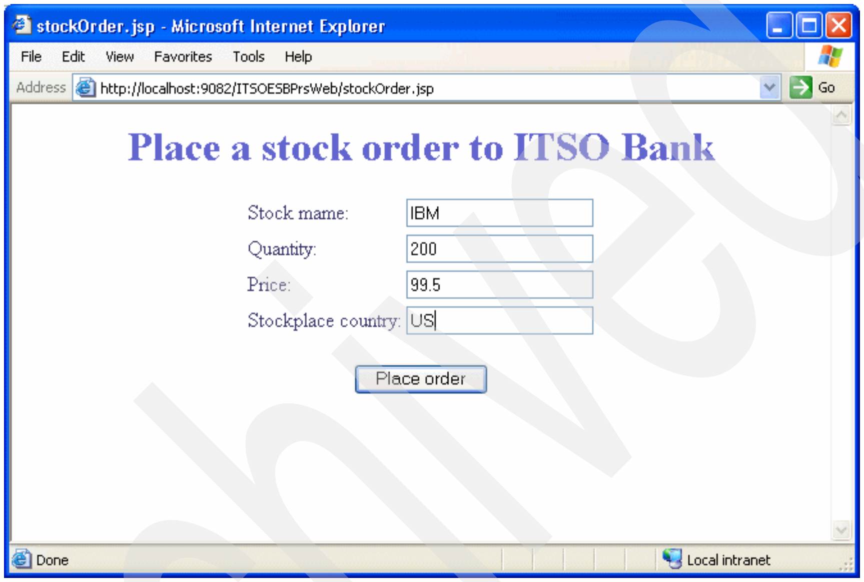
After installing the applications, you can test the mediation components by accessing the Web interface by following these steps:

1. Open a Web browser and enter the following URL:

   `http://<presentation host>:<port>/ITSOESBPrsWeb/stockOrder.jsp`

   For example: `http://ebt0.itso:9080/ITSOESBPrsWeb/stockOrder.jsp`

   An input form for stock order displays (Figure 5-3).



http://localhost:9082/ITSOESBPrsWeb/stockOrder.jsp

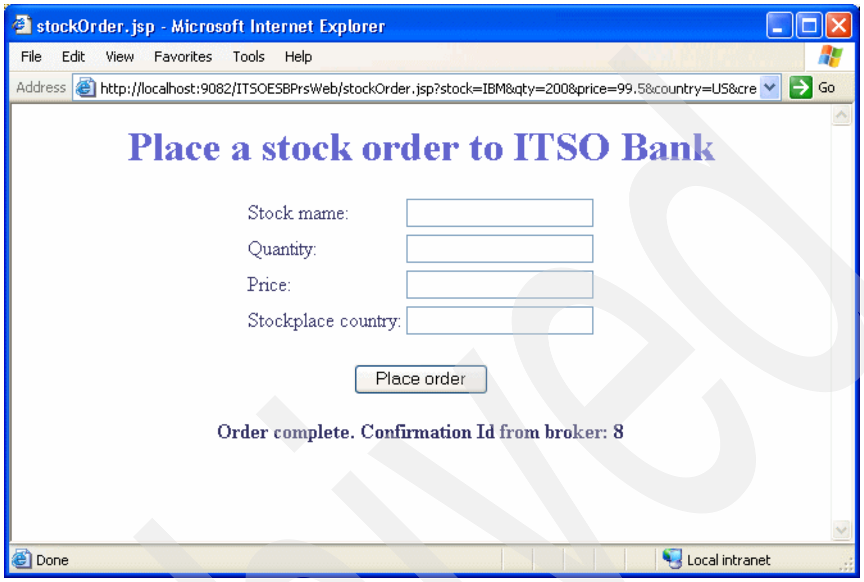
Place a stock order to ITSO Bank

Stock mame:	IBM
Quantity:	200
Price:	99.5
Stockplace country:	US

Figure 5-3 stockOrder.jsp request

2. Complete all the fields and click **Place Order**.

   A message displays that includes the broker name and confirmation ID (Figure 5-4).


stockOrder.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:9082/ITSOESBPrsWeb/stockOrder.jsp?stock=IBM&qty=200&price=99.5&country=US&cre
Go

Place a stock order to ITSO Bank

Stock mame:	
Quantity:	
Price:	
Stockplace country:	

Order complete. Confirmation Id from broker: 8

Figure 5-4 stockOrder.jsp response

3. From the component perspective, the message is composed inside the mediation module. The confirmation ID is returned by the remote broker service. This value is appended to the string `Confirmation Id from US broker:` when the U. S. broker is invoked. A similar message is displayed when the U. K. broker is invoked, where `UK` appears in the string instead of `US`.

5.1.3 Complex mediation scenario

Note: We use the complex mediation application as the sample scenario that is deployed to the Full Support topology for WebSphere ESB. We describe the scenario in Chapter 7, "Implementing the Full Support topology for WebSphere ESB" on page 151.

In this section, we introduce some new requirements from the ITSO Bank mediation solution. For the complex scenario, we add the event emitting functionality to the application capabilities that are already available in the simple mediation scenario. Also, the solution requirements are more complex.

In the simple mediation scenario, the potential users of the ITSO Bank solution are the bank employees. After this first testing phase, ITSO Bank verifies that the solution has reached the business goals. Then, the business decides to make the solution available to external Web customers.

In this scenario, the requirements around the mediation are more rigorous, and new service levels are required concerning:

- Availability
- Scalability
- Performance (response time, throughput, and capacity)
- Event emitting and logging when the marketplace is neither the U. S. nor the U. K.

Figure 5-5 shows how the application logic is modified by using the Event Emitter mediation primitive.

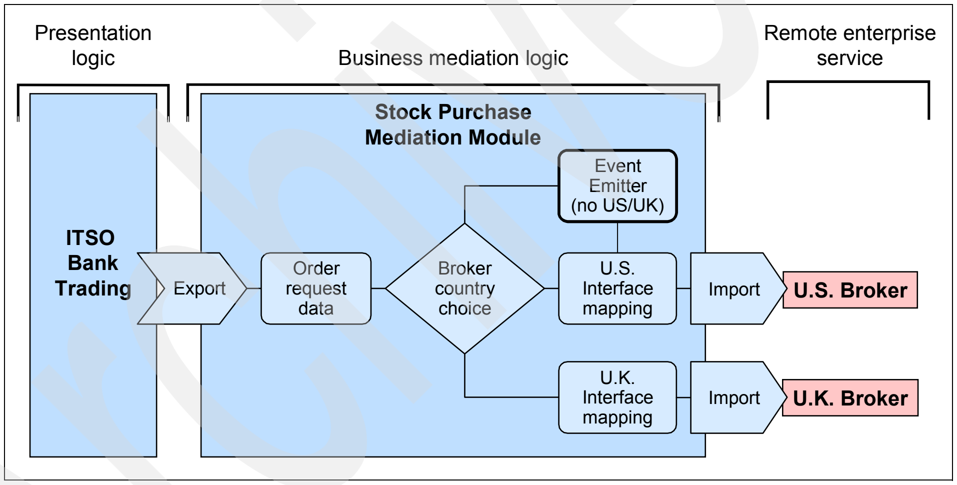


Figure 5-5 Complex mediation scenario used for WebSphere ESB topologies patterns

Example use of the sample application

This book describes two applications for the simple and complex mediation scenarios. New non-functional business requirements and the need to use the CEI infrastructure are the key differences between each application. Business factors define the chosen topology.

From the user perspective, there is no difference between the simple and the complex scenario. The application presentation is unchanged.

In the complex scenario the Common Event Infrastructure is used for event emitting and logging. The CBE Event Browser can be used to view these logged events.

5.2 Business process scenario context

This section describes the high-level sample business process scenario, how the process works, and depicts the flow through the sample application implementation.

The business scenario is focused on the ITSO Bank, the same imaginary company that we use for the mediation scenario. In this section we introduce the role of each of the services involved in the business process. We discuss the business criteria that shape the scenario, as well as the IT requirements that drive the implementation.

The integration issues that we describe in this section are not unique to the finance industry or to company mergers. For any company, continual business change and organizational consolidation can result in a myriad of hardware, software, and application components, which must remain flexible and dynamic. The finance industry integration scenario that we present is an example that you can apply to other industries.

The ITSO Bank is focusing attention on the account closure process. Use of this process by the ITSO Bank employees is supported by many bank applications and services. In addition, the ITSO Bank must meet the requirements concerning the integration of these applications and services. The applications and services that make up the account closure process have satisfied IT functional requirements, but they are disparate. The aim is to focus on the account closure process in terms of integration, coordination, and choreography of the different services.

To meet these requirements, the ITSO Bank builds a new solution based on an automatic business process to manage the account closure request. Essentially, this solution should present a single account closure process by use of a set of existing enterprise services.

This ITSO Bank simple scenario is a simplified application that will be used by the bank employees. The scenario is composed of:

- A number of enterprise services and applications. The new process is composed of the applications and services from existing enterprise information systems.
- A simple business process. The process has been formalized, and it is available through the new ITSO business solution.
- A Web-based GUI. This is a basic application that demonstrates how a business process can be driven by a typical Web application.

Some of the enterprise services interfaces are based on SOAP/HTTP protocol. Each Web services engine provider involved in the ITSO Bank sample business scenario must provide an implementation for a Web services interface.

An implementation of the ITSO Bank sample business scenario is provided. It uses Web services implementations that are written in J2EE running in WebSphere Application Server.

To demonstrate how knowledge of the application drives the topology choice, we provide two different implementations of the account closure scenario:

- The release for the simple ITSO Bank application has no Human Tasks. It is completely automated.
- The final release for the complex ITSO Bank application requires human interaction.

5.2.1 Business goals

The scope of ITSO Bank application is the integration of existing bank services into a business process. The intention is to:

- Reduce costs by reducing the current staff workload associated with management of account closures. With automatic account closure, the employee is no longer required to submit the same information multiple times for different applications.
- Increase customer satisfaction by decreasing account closure response times.
- Decouple employee knowledge and the account closure process, thus eliminating the need to retrain employees if the account closure process changes.

Figure 5-6 illustrates the ITSO Bank account closure process before the business solution is implemented. The logic concerning the choice of the bank service is held by the bank employee.

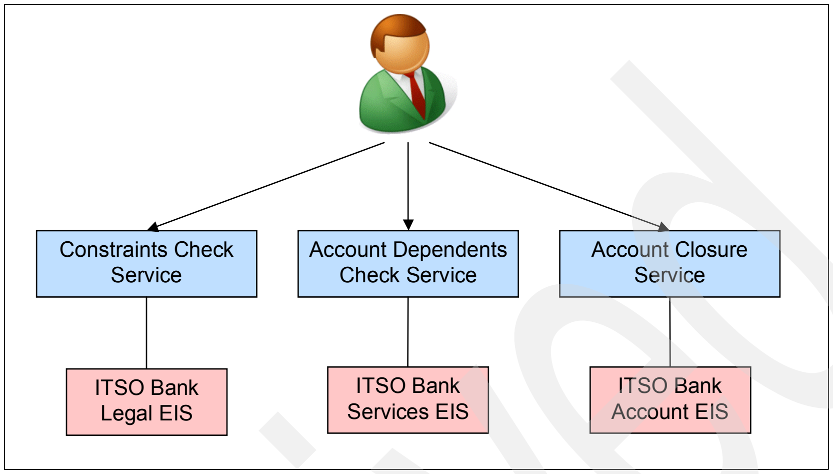


Figure 5-6 Account closure process prior to the ITSO Bank business solution

5.2.2 Simple business process scenario

Note: We describe the simple business process application that we use as the sample scenario that is deployed to the Loosely Coupled topology for WebSphere Process Server in Chapter 8, “Implementing the Loosely Coupled topology for WebSphere Process Server” on page 231.

The business scenario that we discuss in this section is composed of three sequential phases: request, analysis, and closure. Each of these phases is implemented by means of an existing enterprise service:

1. Request phase

   This is the first phase of ITSO Bank account closure process. In this phase the *Constraints check* service is invoked. This service already exists in the bank enterprise information system. The service checks for the presence of constraints (for example legal constraints) that inhibit account closure. If a blocking constraint is raised by this service, an error message is returned to the business user

2. Analysis phase

   This phase checks whether there are account dependent services linked to the account (for example, credit card, debit card, taxes payment, and so forth). If no dependencies are present the process continues. Otherwise, a message is returned to the user that includes the dependency information and the actions that have to be managed before account closure can be completed.

Note: To simplify the process proposed in our scenario, the business exceptions, such as constraints and account dependencies, are not managed automatically. In a real business process, exceptions are better managed using WebSphere Integration Developer.

3. Closure phase

   The final process phase is closure. The *Closure service* is invoked to submit the account closure request and the process completes with a confirmation message to the user.

In this solution, we have implemented the three enterprise services as sample Java objects to be deployed in WebSphere Application Server with simple logic. The business process invokes the enterprise services through Web service (SOAP/HTTP) bindings.

Figure 5-7 illustrates the simple business account closure process.

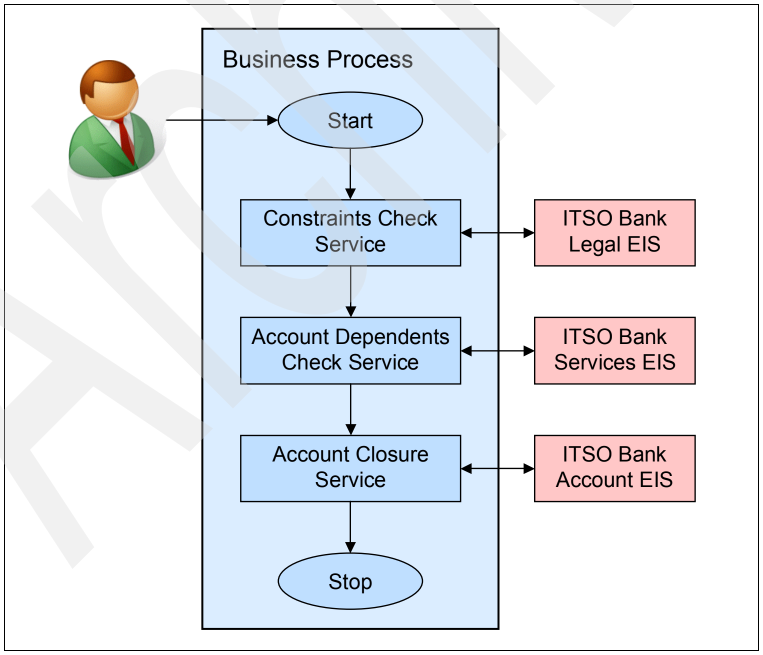


Figure 5-7 Account closure process, simple business scenario

In the simple business scenario, the process is completely automated with the following binding invocations:

1. The bank employee submits the account closure order to the process. The interaction between the presentation and the business module is implemented using an SCA binding export.
2. The *Constraints check* service is invoked. The connection between the process and the service is a Web service binding invoked synchronously.
3. The next service is invoked in the same way, with a sysnchronous Web service binding.
4. The process invokes the *Closure* service with an asynchronous one-way Web service binding.
5. The process terminates successfully and confirmation is returned.

Example use of the sample application

This scenario is shipped in three different EAR files:

- ITSOWPSSmplPresApp, which includes the simple user interface for the process (presentation layer)
- ITSOWPSSmplBusApp, which includes the business module that implements the business process defined by the bank (business layer)
- ITSOWPSSmplSrvApp, which includes the three services that are used by the process (service layer)

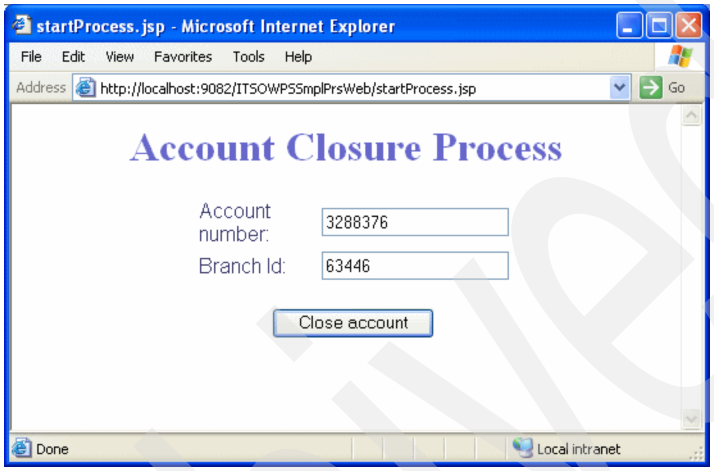
These files are available in the additional material that accompanies this book. For more information about how to obtain this additional material, see Appendix A, “Additional material” on page 427.

After installing the applications, you can test the business components by accessing the Web interface. Follow these steps:

1. Open a Web browser and enter the following URL:

   `http://<presentation-host>:<port>/ITSOWPSSmplPrsWeb/startProcess.jsp`

   An input form for account data displays (Figure 5-8).


startProcess.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address
Go
Close account
Done
Local intranet

http://localhost:9082/ITSOWPSSmplPrsWeb/startProcess.jsp

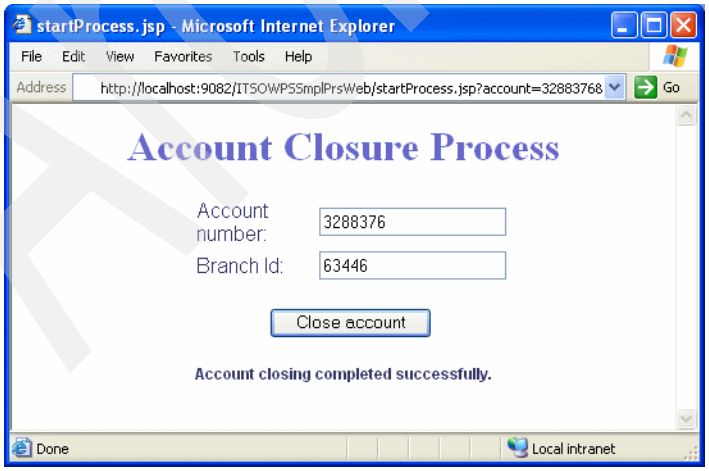
Account Closure Process

Account number: 3288376

Branch Id: 63446

Figure 5-8 Account closure request

2. Complete all the fields and click **Close account**.
3. If no constraints or dependencies are found, confirmation of successful account closure is returned (Figure 5-9).


startProcess.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address
Go
Close account
Done
Local intranet

http://localhost:9082/ITSOWPSSmplPrsWeb/startProcess.jsp?account=32883768

Account Closure Process

Account number: 3288376

Branch Id: 63446

Account closing completed successfully.

Figure 5-9 Account closure response

4. If constraints or dependencies are found, a warning message displays, and the account closure process is not invoked (for example, `Accounts “111111” or “123456” do not complete`). Verify in the server console that the process stops before the final closure service is invoked.

5.2.3 Complex business process scenario

Note: We describe the complex business process application that we use as the sample scenario that is deployed to the Full Support topology for WebSphere Process Server in Chapter 9, “Implementing the Full Support topology for WebSphere Process Server” on page 319.

The next business scenario assumes the first release of ITSO Bank business solution satisfied the business requirements.

The bank has decided to change the process that drives account closure. A bank employee must now check the account balance before the process completes with the invocation of the closure service. Therefore, the process must invoke the enterprise service providing the account balance. After the balance has been calculated, a human task is created inside the process. The simple human task provides the bank employee with the account balance information. Based on this value (for example, the balance exceeds U. S. $10 000), the employee decides whether to proceed with account closure or to take another action. In the second case, the process stops without invocation of the closure service, and the bank employee takes some account retention action.

Note: WebSphere Integration Developer and WebSphere Process Server provides specific capabilities to automate this decision. By means of Business Rules the threshold (in our example U. S. $10 000) can be configured at runtime, and a new version of the process is not required if the threshold changes. We use a human task instead of a business rule.

Figure 5-10 illustrates the complex business account closure process.

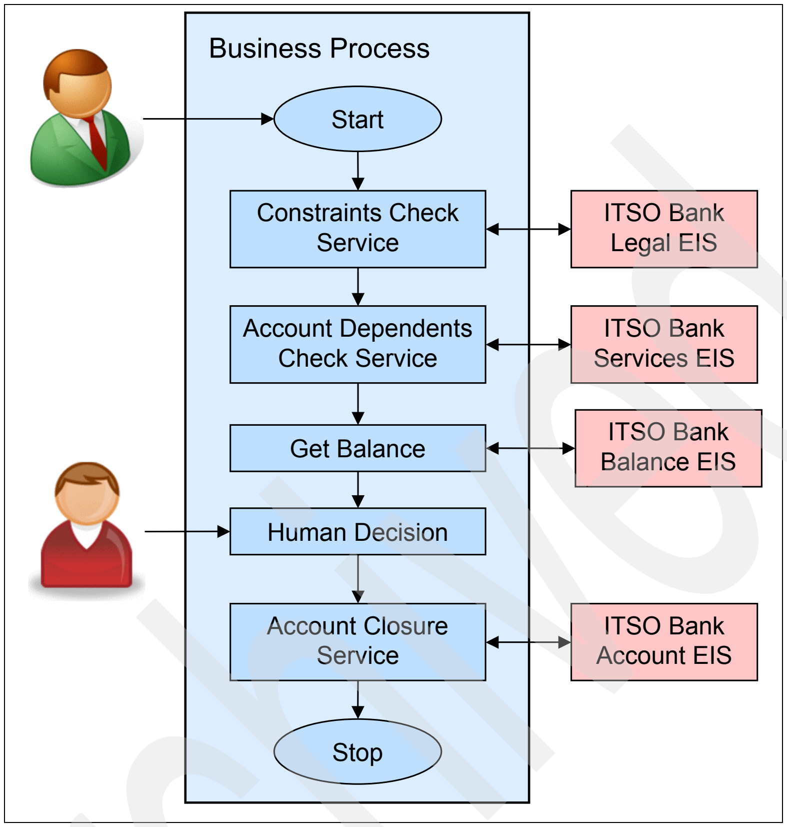


Figure 5-10 Account closure process, complex business process scenario

Example use of the sample application

This scenario is shipped in three EAR files:

- ITSOWPSCplxPresApp, which includes the user interface for the process (presentation layer).
- ITSOWPSCplxBusApp, which includes the business module that implements the business process defined by the bank (business layer).
- ITSOWPSCplxSrvApp, which includes the service that provides the information about the account balance (service layer). For the complex scenario, we use the same three services that are used in the simple scenario the process (service layer).

These files are available in the additional material that accompanies this book. For more information about how to obtain this additional material, see Appendix A, “Additional material” on page 427.

You can test the business components by using the simple application Web interface. (We discuss how the application works for an account closure request in “Example use of the sample application” on page 77.)

A new management task for the bank employee is to check for approval requests and to take ownership of them.

Note: The scope of the simple application is to provide an example of long-running process, so the human task is managed by the user manually. WebSphere Process Server supports multi-level escalation for human tasks including e-mail notification.

In this section we explain how to use the application for the account closure approval. The bank employee accesses the simple user interface to check for the existence of any account closure requests that are ready to be completed.

When a user clicks **Available tasks**, the Web application queries WebSphere Process Server for the availability of any human tasks. If an account closure request is stopped at the human decision step, the task is retrieved, and the account balance information is displayed. The bank employee makes a decision whether to approve the account closure. If approved the closure service is invoked. Otherwise, the process stops and the account stays open.

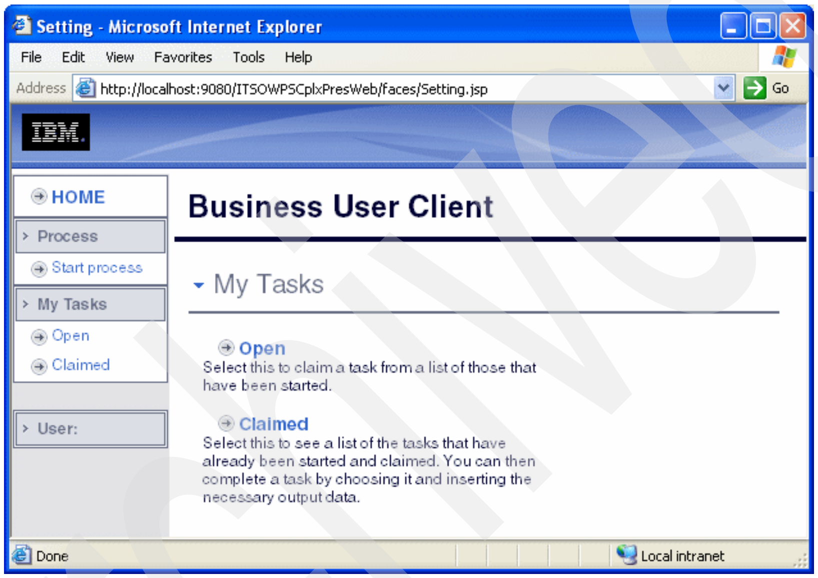
After installing the applications, you can test the business components by accessing the Web interface. Follow these steps:

1. Open a Web browser and enter the following URL:

   `http://<presentation host>:<port>/ITSOWPSCplxPrsWeb/faces/Index.jsp`

   An input form for account data displays.

   The browser displays a welcome page (Figure 5-11). To start the account closure process for the complex scenario, click **Start process** in the left menu.

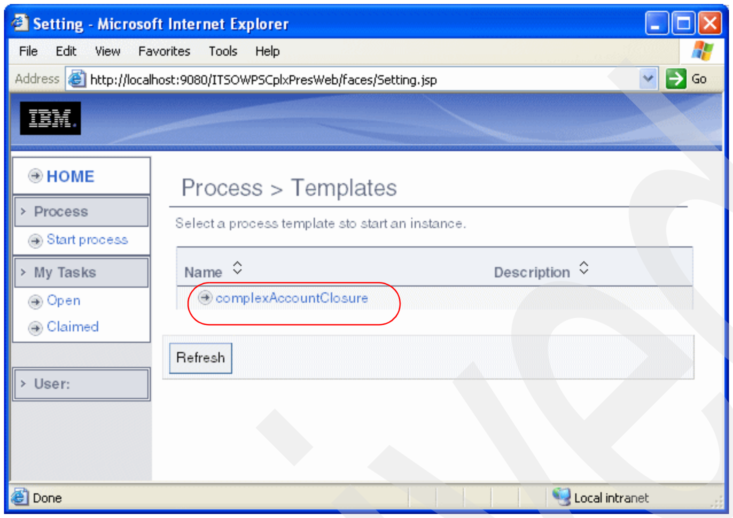

Setting - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:9080/ITSOWPSCplxPresWeb/faces/Setting.jsp
Go
IBM
HOME
Process
Start process
My Tasks
Open
Claimed
User:
Business User Client
My Tasks
Open
Claimed
Done
Local intranet

Select this to claim a task from a list of those that have been started.

Select this to see a list of the tasks that have already been started and claimed. You can then complete a task by choosing it and inserting the necessary output data.

Figure 5-11 Welcome page

2. The new page shows a list of available templates. For our scenario, the template's name is *complexAccountClosure*. To start a new process instance, click **complexAccountClosure**, as shown in Figure 5-12.


Setting - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:9080/ITSOWPSCplxPresWeb/faces/Setting.jsp
Go
HOME
Process
Start process
My Tasks
Open
Claimed
User:
Done
Local intranet

Process > Templates

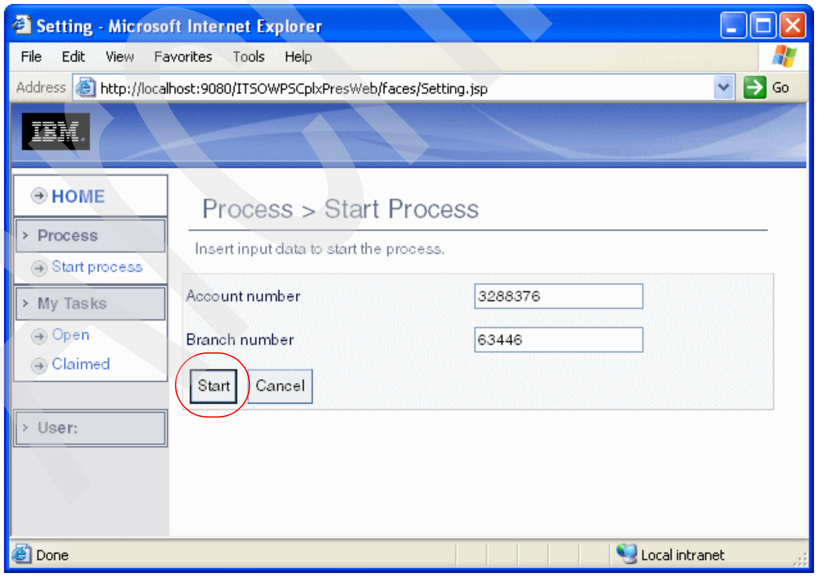
Select a process template sto start an instance.

Name	Description
complexAccountClosure	

Refresh

Figure 5-12 List of available process templates

3. In the next page, enter the account data and click **Start** (Figure 5-13).


Setting - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:9080/ITSOWPSCplxPresWeb/faces/Setting.jsp
Go
HOME
Process
Start process
My Tasks
Open
Claimed
User:
Done
Local intranet

Process > Start Process

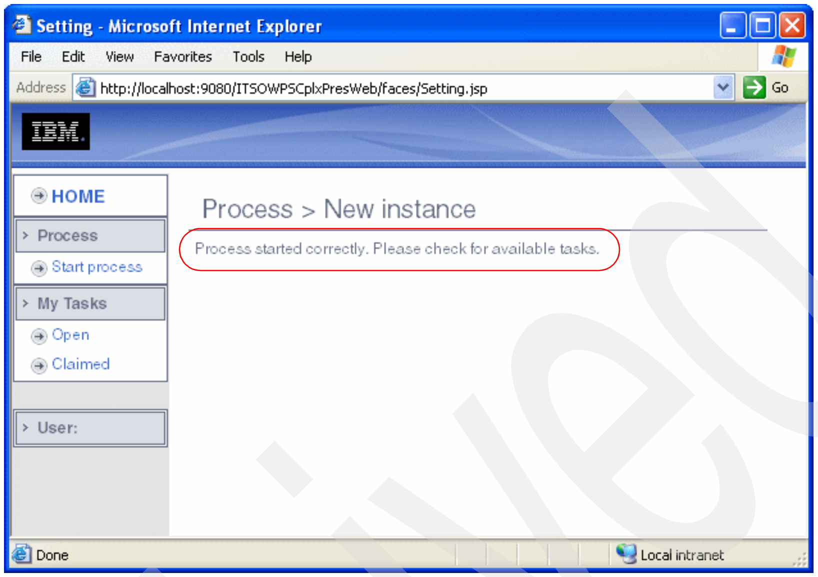
Insert input data to start the process.

Account number	3288376
Branch number	63446

Start Cancel

Figure 5-13 Insert account data

4. At this point the process starts and a confirmation message displays (Figure 5-14).

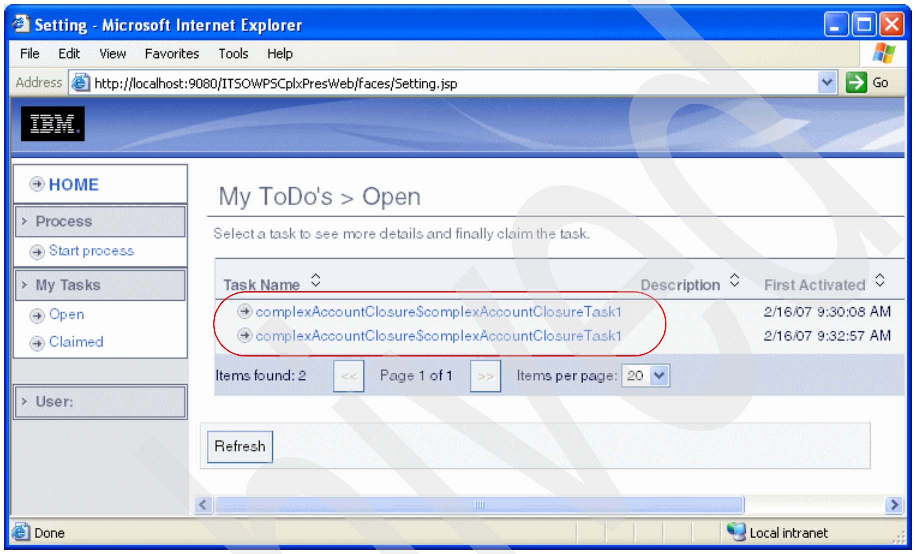

Setting - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:9080/ITSOWPSCplxPresWeb/faces/Setting.jsp
Go
IBM
HOME
Process
Start process
My Tasks
Open
Claimed
User:
Process > New instance

Process started correctly. Please check for available tasks.

Figure 5-14 New process instance started successfully

5. Repeat steps 2 and 3, and start a new account closure process.

6. Two process instances are started and the bank employee *has no knowledge of the process logic*. Subsequently, the employee must check for tasks that require approval. In our example, two tasks are present in the list, as shown in Figure 5-15. The bank employee must complete the tasks by approving or denying the account closure. To check the availability of tasks, click **Open** in the **My Tasks** section of menu.


Setting - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:9080/ITSOWPSCplxPresWeb/faces/Setting.jsp
Go
IBM
HOME
Process
Start process
My Tasks
Open
Claimed
User:
Refresh
Done
Local intranet

My ToDo's > Open

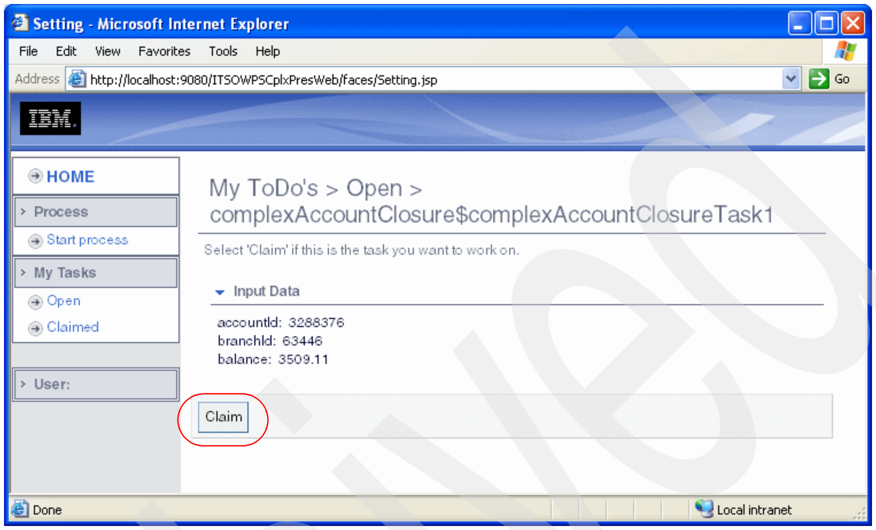
Select a task to see more details and finally claim the task.

Task Name	Description	First Activated
complexAccountClosure$complexAccountClosureTask1		2/16/07 9:30:08 AM
complexAccountClosure$complexAccountClosureTask1		2/16/07 9:32:57 AM

Items found: 2 << Page 1 of 1 >> Items per page: 20

Figure 5-15 Available tasks list

7. Click **Claimed** in the menu to verify that no task is in the *Claimed* state. To claim status, select one of the available tasks, and click the task name. A new page displays with the task information. On this page, click **Claim** to claim the task, as shown in Figure 5-16.


Setting - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:9080/ITSOWPSCplxPresWeb/faces/Setting.jsp
Go
IBM.
HOME
Process
Start process
My Tasks
Open
Claimed
User:
Claim
Done
Local intranet

My ToDo's > Open > complexAccountClosure$complexAccountClosureTask1

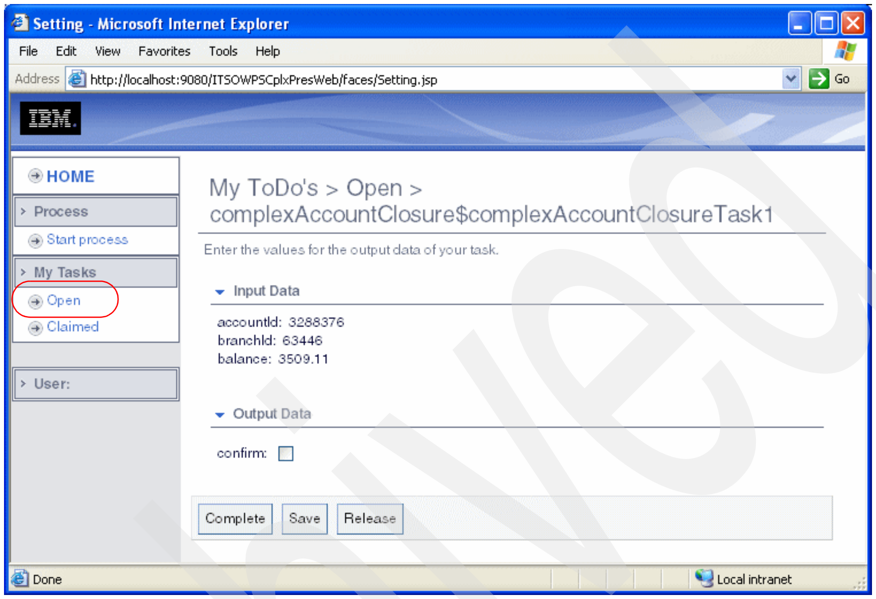
Select 'Claim' if this is the task you want to work on.

Input Data

accountId: 3288376
branchId: 63446
balance: 3509.11

Figure 5-16 Task details

8. The next page provides details on the account closure request and the calculated balance. In this page, do *not* click **Complete**. Before completing the task, verify that only one task is in the *Open* state. Click **Open** in the menu (Figure 5-17).



My ToDo's > Open > complexAccountClosure$complexAccountClosureTask1

Enter the values for the output data of your task.

Input Data

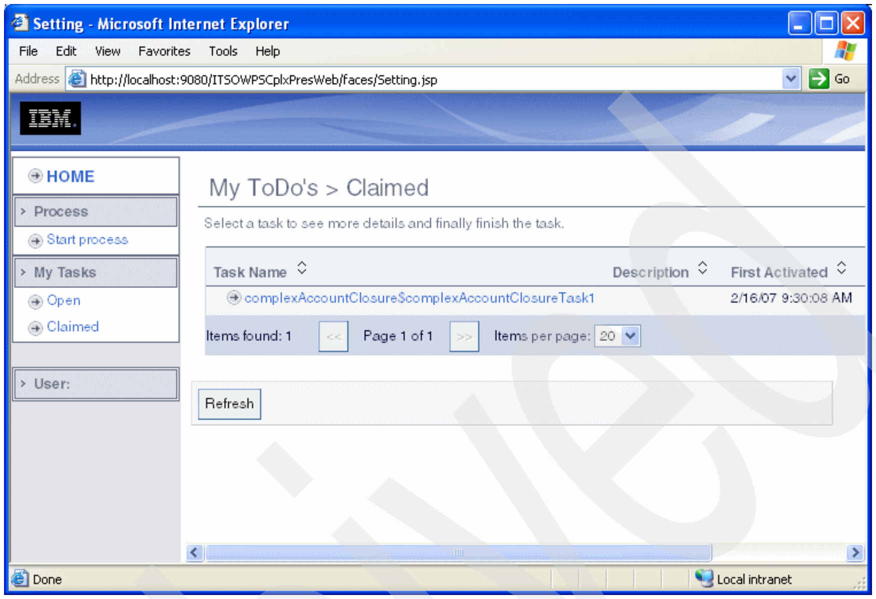
accountId: 3288376
branchId: 63446
balance: 3509.11

Output Data

confirm: ☐

Figure 5-17 Account closure confirmation page

9. To complete the task, click **Claimed** in the menu. The task that you claimed should display (Figure 5-18).


Setting - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:9080/ITSOWPSCplxPresWeb/faces/Setting.jsp
Go
IBM.
HOME
Process
Start process
My Tasks
Open
Claimed
User:
Refresh
Done
Local intranet

My ToDo's > Claimed

Select a task to see more details and finally finish the task.

Task Name	Description	First Activated
complexAccountClosure$complexAccountClosureTask1		2/16/07 9:30:08 AM

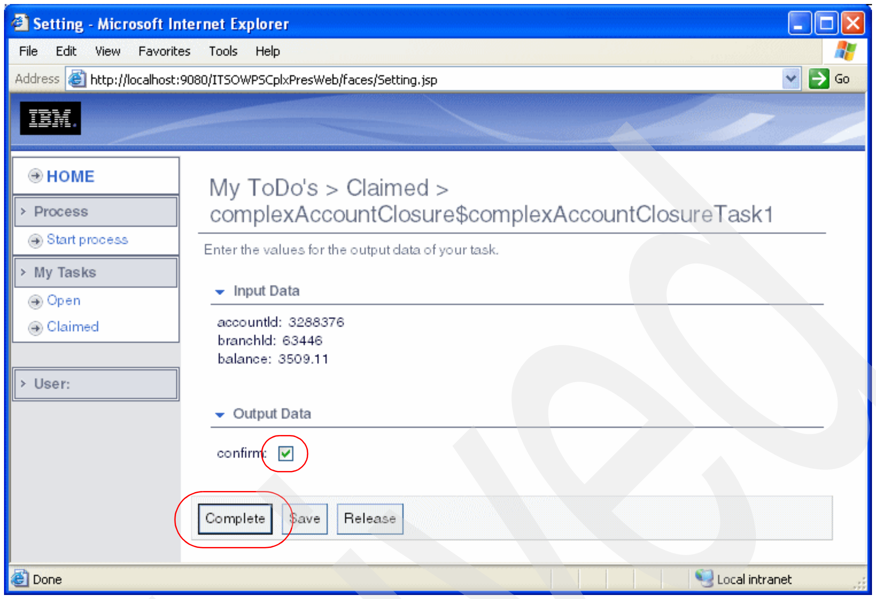
Items found: 1 << Page 1 of 1 >> Items per page: 20

Figure 5-18 One task to claim

10. Click the task name to complete the task. A page with the account closure request details and account balance displays.

    At this point, the process is waiting for a human decision. If the account balance is more than U. S. $10 000, the bank employee must click **Suspend** and some retention actions are required. In this case, the process stops without invoking the closure service, and the account stays open. To complete our example, we decide to confirm the account closure.

11.Select **confirm** and click **Complete** as shown in Figure 5-19.


Setting - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:9080/ITSOWPSCplxPresWeb/faces/Setting.jsp
Go
HOME
Process
Start process
My Tasks
Open
Claimed
User:
Complete
Save
Release
Done
Local intranet

My ToDo's > Claimed > complexAccountClosure$complexAccountClosureTask1

Enter the values for the output data of your task.

Input Data

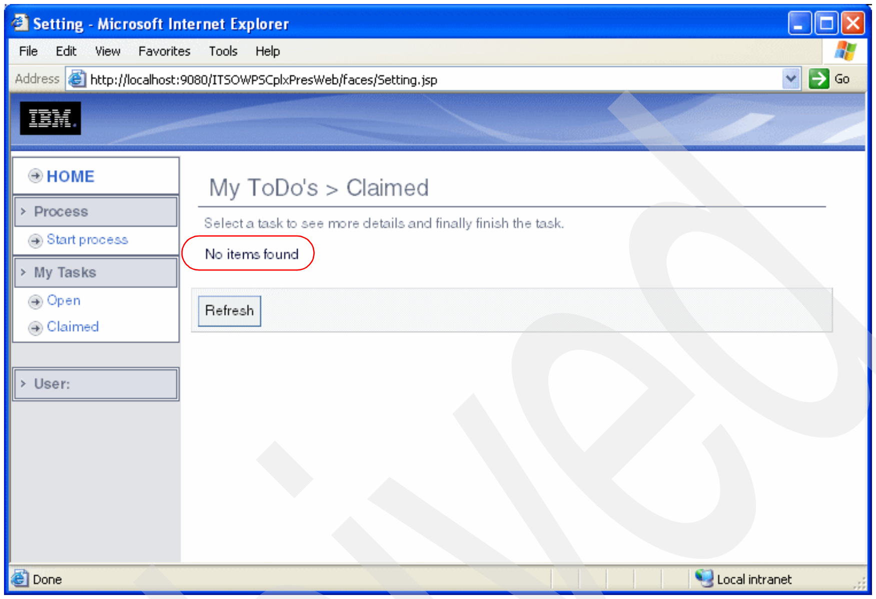
accountId: 3288376
branchId: 63446
balance: 3509.11

Output Data

confirm: ✓

Figure 5-19 Complete the task

12. The claimed task page is refreshed, and the task is no longer available, as shown in Figure 5-20.


Setting - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:9080/ITSOWPSCplxPresWeb/faces/Setting.jsp
Go
IBM.
HOME
Process
Start process
My Tasks
Open
Claimed
User:
My ToDo's > Claimed
Refresh
Done
Local intranet

Select a task to see more details and finally finish the task.

No items found

Figure 5-20 Task complete, and no claimed tasks are in the list

At this point, the bank employee has finished all activity with the closure of this account. After confirmation, the process completes after invoking the *Closure* service.

Note: Provision of a console for process tracing is out of the scope of this scenario. You can use the Business Process Choreographer (BPC) Explorer application to verify the Process State and the Process Output Message.

Part 2

Production topologies for WebSphere ESB

6

Implementing the Basic topology for WebSphere ESB

This chapter details the steps to implement the *Basic topology* for WebSphere ESB and to deploy a sample application to this topology. We discuss the following topics:

- Selecting and implementing the Basic topology
- Generic steps for creating WebSphere ESB clusters
- Specific configuration steps for the Basic topology
- Installing and testing the scenario

For simplicity, this chapter does not implement any security. To be considered a production topology, the addition of security is essential. Chapter 9, "Implementing the Full Support topology for WebSphere Process Server" on page 319 shows how to add security to a production topology.

6.1 Selecting and implementing the Basic topology

This section includes the following sections:

- Scenario requirements

  Summarizes the fictitious business scenario that we deploy to the Basic topology that we build in this chapter.
- Selecting the appropriate topology

  Demonstrates why we select the Basic topology for WebSphere ESB to host this business scenario.
- Overview of how to implement the Basic topology

  Describes the steps that are necessary to build the Basic topology for WebSphere ESB.

Note: For an overview of the Basic topology, refer to 4.2, "Basic topology" on page 44.

6.1.1 Scenario requirements

You must build a production topology to host a WebSphere ESB mediation flow that is built for the factious organization ITSO Bank. The mediation flow (shown in Figure 6-1) performs the following functions:

1. Bank customers place stock orders through a traditional trading Web application.
2. The Web application sends a synchronous call to the Stock Purchase mediation flow running in WebSphere ESB.
3. The mediation flow determines by which of two stock brokers the stock order should be fulfilled—either a U. S. broker or a U. K. .broker.
4. A Web service request is sent to the appropriate broker, and a response is returned to the trading Web application (through the mediation response flow).

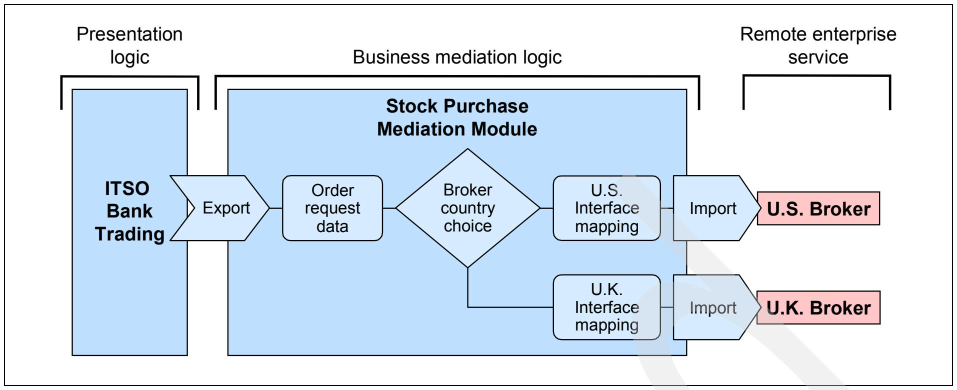


Figure 6-1 Simple mediation scenario for WebSphere ESB

ITSO Bank has the following business and IT requirements for this scenario:

- Failover capability
- Limited hardware availability
- The application internals are well understood
- The flow from consumer to provider is synchronous
- Event management services are not required

Note: For more detailed information about this scenario, refer to 5.1.2, "Simple mediation scenario" on page 68.

6.1.2 Selecting the appropriate topology

Based on the ITSO Bank requirements for the mediation flow scenario, we can select an appropriate production topology to host the Stock Purchase mediation module. To help us select the appropriate topology, we use the topology selection flowchart, as described in 4.5.2, "Topology selection flow chart" on page 62. Figure 6-2 shows that by using the topology selection flowchart, we can select the Basic topology as the topology of choice for the ITSO Bank simple mediation flow scenario.

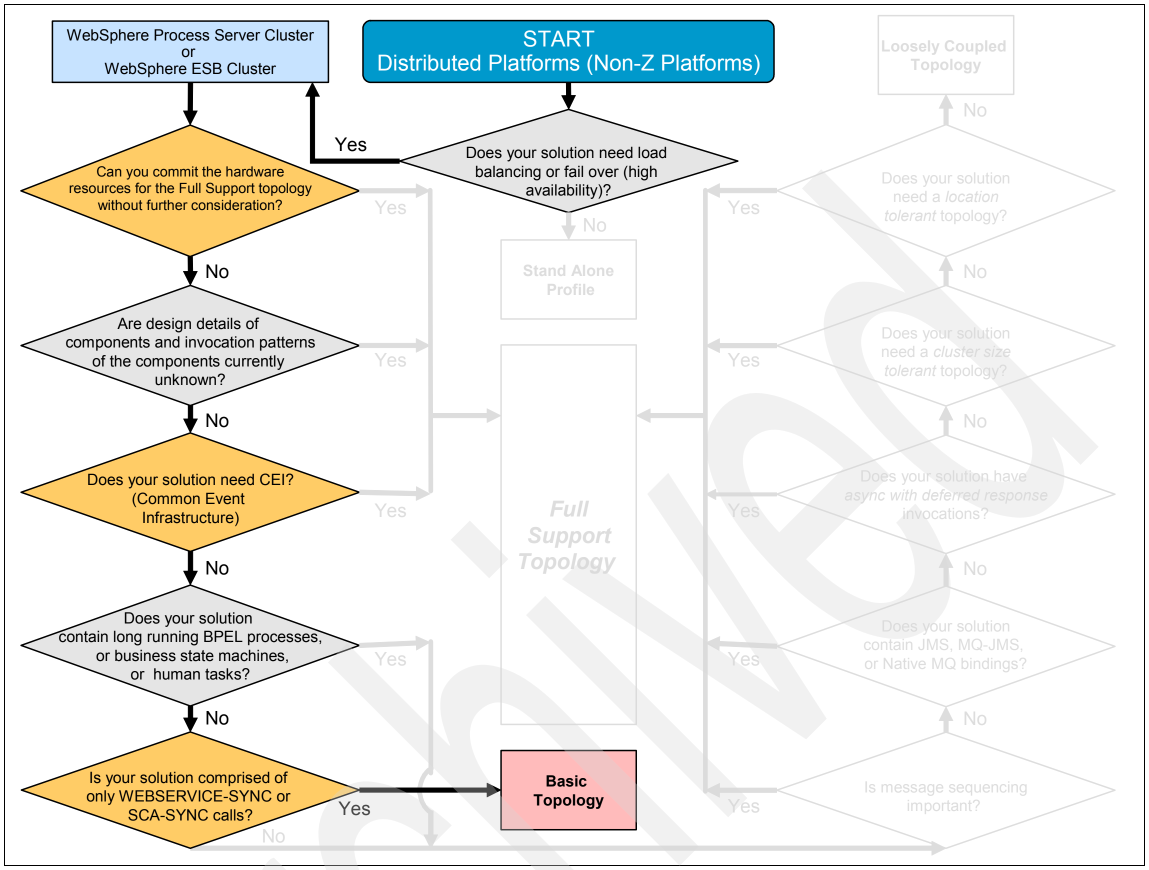

START
Distributed Platforms (Non-Z Platforms)
Does your solution need load balancing or fail over (high availability)?
Yes
No
Stand Alone Profile
WebSphere Process Server Cluster or WebSphere ESB Cluster
Can you commit the hardware resources for the Full Support topology without further consideration?
Yes
No
Are design details of components and invocation patterns of the components currently unknown?
Yes
No
Does your solution need CEI? (Common Event Infrastructure)
Yes
No
Does your solution contain long running BPEL processes, or business state machines, or human tasks?
Yes
No
Is your solution comprised of only WEBSERVICE-SYNC or SCA-SYNC calls?
Yes
No
Basic Topology
Full Support Topology
Loosely Coupled Topology
No
Does your solution need a location tolerant topology?
Yes
No
Does your solution need a cluster size tolerant topology?
Yes
No
Does your solution have async with deferred response invocations?
Yes
No
Does your solution contain JMS, MQ-JMS, or Native MQ bindings?
Yes
No
Is message sequencing important?
Yes

Figure 6-2 Topology selection for the simple mediation flow scenario

The main decision points for ITSO Bank are:

- A highly available solution with load balancing is required, so a clustered topology is needed.
- ITSO Bank does not want to commit the hardware resources that are required to implement the Full Support topology unless the applications that they are deploying to the topology require those resources.
- Their application solution uses only synchronous Web services calls. Therefore, the Basic topology meets their needs and is the topology selected.

6.1.3 Overview of how to implement the Basic topology

Figure 6-3 summarizes the steps to build the Basic topology.

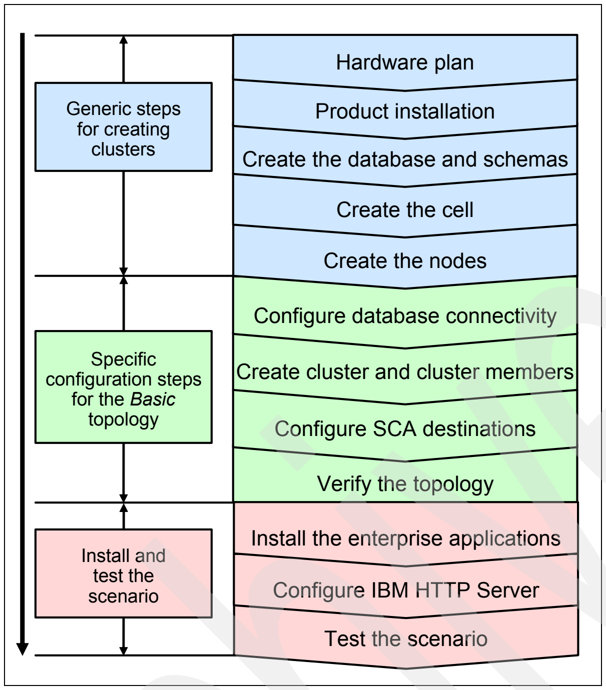

Generic steps for creating clusters
Hardware plan
Product installation
Create the database and schemas
Create the cell
Create the nodes
Specific configuration steps for the *Basic* topology
Configure database connectivity
Create cluster and cluster members
Configure SCA destinations
Verify the topology
Install and test the scenario
Install the enterprise applications
Configure IBM HTTP Server
Test the scenario

Figure 6-3 Building the Basic topology for WebSphere ESB

Figure 6-4 shows the simple mediation application on this topology.

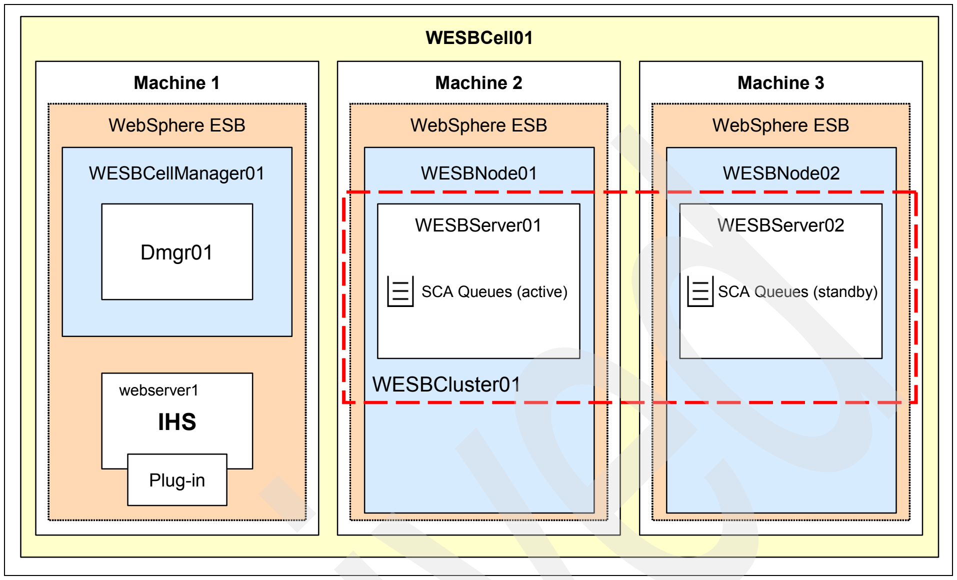


Figure 6-4 Basic topology implementation

6.2 Generic steps for creating WebSphere ESB clusters

This section describes the hardware plan for the Basic topology and describes how to prepare a WebSphere ESB cluster.

6.2.1 Hardware plan for the topology

This section describes the hardware that is required to support the Basic topology. In this topology (illustrated in Figure 6-5 on page 100), three machines support the cell hosting for the mediation module and one machine hosts the client and Web services applications for testing purposes. A final shared machine provides the database functionality. The hardware plan includes:

- The first machine (`ebt0`) hosts the deployment manager and the IBM HTTP Server and associated plugin.
- The second and third machines (`ebt1` and `ebt2`) host the mediation cluster in a horizontally clustered configuration, as well as the associated nodes.
- The fourth machine (`ebt3`) hosts the client application to access the mediation module and the Web services applications in two stand-alone WebSphere Application Servers.

- A fifth machine (`db2`), not shown in the diagram hosts the database that is required to support the messaging engines.

> **Note:** Although this is the configuration we used, you can choose any combination of machines to create this topology.

Table 6-1 describes how we use these machines. You can choose to use a different combination of machines and operating systems. In total, the topology in this chapter consumes about 2.5 GB of memory.

Table 6-1 Systems that we use for this topology

System name	Host name	Operating system
ebt0	ebt0.itso	Windows® 2003 Server
ebt1	ebt1.itso	Windows 2003 Server
ebt2	ebt2.itso	Windows 2003 Server
ebt3	ebt3.itso	Windows 2003 Server
db2	db2.itso	Red Hat Advanced Server 4 U4

> **Note:** You can define a hosts file that maps the host names that are defined in Table 6-1 to the IP address of each of your machines to help when following the steps in this chapter.
>
> You can find the hosts file on Windows platforms at:
>
> `<WINDOWS_HOME>\system32\drivers\etc`

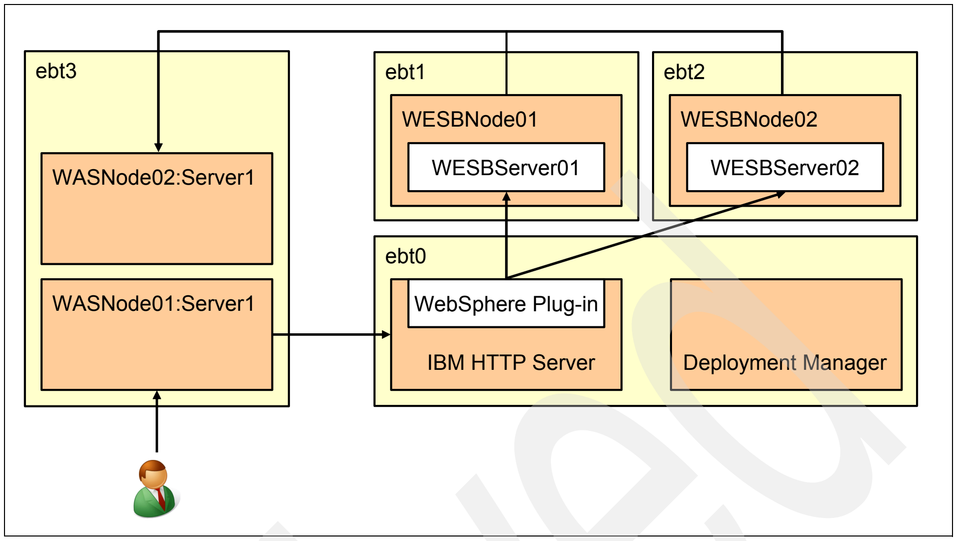


Figure 6-5   Hardware plan for the Basic topology

The end-to-end flow for the topology, as shown in Figure 6-5, is:

1. A user makes a request from a browser to the front-end client application, running as a JSP on `WASNode01:server1` on host `ebt3`.
2. The client application is configured to forward the request to IBM HTTP Server running on `ebt0`.
3. The WebSphere Plugin distributes the request to a cluster member hosting the mediation module. The cluster member is `WESBServer01` or `WESBServer02` hosted on `ebt1` or `ebt2` respectively.
4. The mediation application running on this cluster member performs the appropriately mediated call to one of the Web services that is hosted on `WASNode02:server1` on host `ebt3`.
5. The return path through the infrastructure is the reverse of this flow.

6.2.2 Product installation

It is assumed that the following products have been installed:

- WebSphere ESB V6.0.2
- IBM HTTP Server V6
- DB2 Universal Database Enterprise Server V8.2

Note: For information about how to install these products, refer to Appendix B, “Product installation” on page 429.

6.2.3 Creating the messaging database

This task describes how to create the messaging database. For this chapter, we assume the Relational Database Management System (RDBMS) is DB2, the instance is called `ebtinst1`, and the administrative user name is `ebtinst1`.

Log in as the database administrative user and issue the following commands from a DB2 command window to create the required database:

```
db2start
db2 create database MEDB
```

If the **`create`** command is successful, you see the following message:

```
DB20000I The CREATE DATABASE command completed successfully
```

6.2.4 Copying the database driver files

The Java driver files for DB2 are located on the database system. In our DB2 example, they are in the directory /opt/IBM/db2/V8.1/java.

On each of the nodes and on the deployment manager, copy the database driver files to a standard location. In this example the standard location for the driver files is c:\db2client.

6.2.5 Creating the messaging database schemas

The messaging database includes two schemas that support the messaging engines in this topology:

- SCA Application
- SCA System

You can create the DB2 commands for generating the schemas on any system with the WebSphere ESB product installed. On any WebSphere ESB node, create the data definition language (DDL) files for the Service Integration Bus (SIB) SCA schemas as detailed in Table 6-2.

Table 6-2 WebSphere ESB SCA schemas

Schema name	Database	Purpose
SCASYS	MEDB	System bus messaging store
SCAAPP	MEDB	Application bus messaging store

In this example:

- The database user name is `ebtinst1`.
- The script to generate the schemas is located in the %*WAS_HOME*%\bin directory (for example, C:\IBM\WebSphere\ESB\bin).
- The database target for the DDL generated is running on a Linux® system.

To create the messaging database schemas, follow these steps:

1. Open a Windows command prompt and issue the following command:

   ```
   cd C:\IBM\WebSphere\ESB\bin
   ```

   > **Note:** Each of the commands in this series of steps is a single line although in the format of this book they might wrap. Also, the commands assume that the database is running on a UNIX platform. If you are using Windows, you need to replace -platform unix with `-platform windows`.

2. Issue the following command for the SCASYS DDL generation - SCA System Schema:

   ```
   sibDDLGenerator.bat -system db2 -version 8.1 -platform unix -schema
   SCASYS -statementend ; -user ebtinst1 >
   c:\createSIBSchema_SCASYS.ddl
   ```

3. Issue the following command for the SCAAPP DDL generation - SCA Application Schema:

   ```
   sibDDLGenerator.bat -system db2 -version 8.1 -platform unix -schema
   SCAAPP -statementend ; -user ebtinst1 >
   c:\createSIBSchema_SCAAPP.ddl
   ```

4. Transfer the DDL files to the DB2 database server and execute them.

   a. Transfer c:\createSIBSchema_SCASYS.ddl and c:\createSIBSchema_SCAAPP.ddl to the DB2 database server. In this example, the DDL files were transferred to /tmp on the database server.

      > **Attention:** If you transfer these files from Windows to UNIX, there might be return characters (`\r`) at the end of each line. To run the DDL scripts, you must remove these characters. A UNIX utility such as `dos2unix` will remove these characters.

b. Log in to the DB2 server as the appropriate user (`ebtinst1`) and run the following commands to create the tables:

```
db2 connect to MEDB
db2 -tf /tmp/createSIBSchema_SCAAPP.ddl
db2 -tf /tmp/createSIBSchema_SCASYS.ddl
db2 connect reset
```

Note: These DB2 commands report various informational index and primary key messages, such as:

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

These are not errors. You can ignore these messages.

6.2.6 Creating the WebSphere ESB cell

This task creates the WebSphere ESB cell, which is achieved by creating a deployment manager profile. Follow these steps:

1. Login to the `ebt0.itso` machine, which becomes the deployment manager.
2. Open the profile creation wizard by selecting **Start** → **All Programs** → **IBM WebSphere** → **Enterprise Service Bus 6.0** → **Profile creation wizard**.

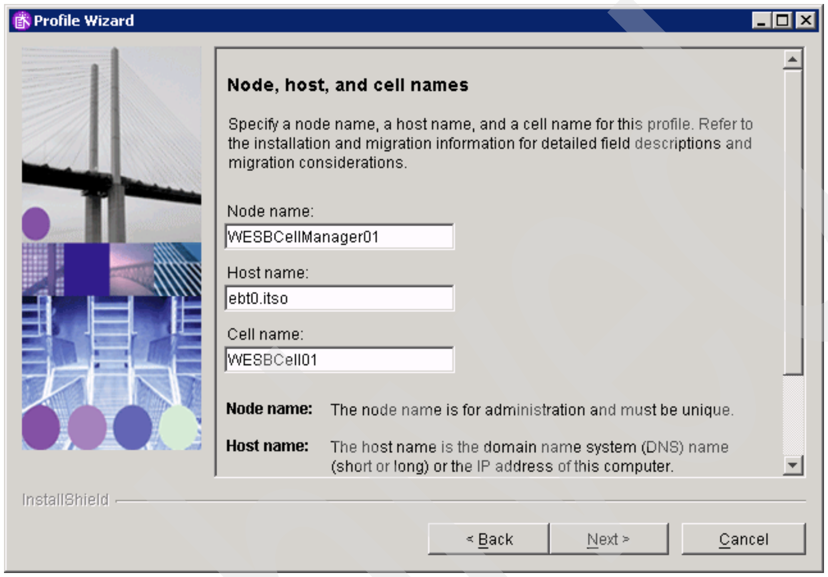
Note: For information about how to launch the Profile creation wizard on other platforms, see:

`http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tpro_instances.html`

3. In the Profile creation wizard, click **Next**.
4. Select **Deployment manager profile** and click **Next**.
5. Leave the default profile name as *Dmgr01* and click **Next**.
6. Leave the default profile directory and click **Next**.

7. Specify the node name, host name, and cell name according to the follow specifications and click **Next**, as shown in Figure 6-6.

   – Node name: `WESBCellManager01`
   – Host name: `ebt0.itso`
   – Cell name: `WESBCell01`



Node, host, and cell names

Specify a node name, a host name, and a cell name for this profile. Refer to the installation and migration information for detailed field descriptions and migration considerations.

Node name: WESBCellManager01

Host name: ebt0.itso

Cell name: WESBCell01

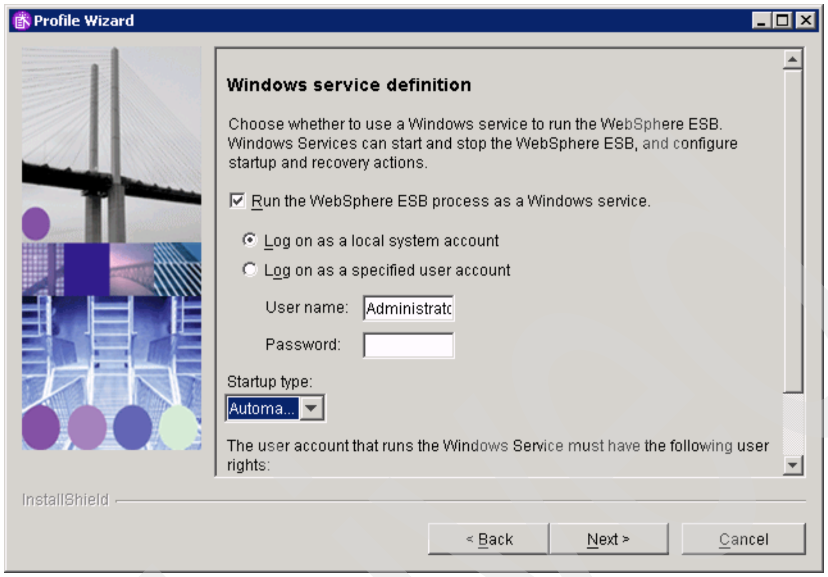
Node name: The node name is for administration and must be unique.

Host name: The host name is the domain name system (DNS) name (short or long) or the IP address of this computer.

Figure 6-6 Node, host, and cell name configuration

8. Leave the port numbers at their default settings and click **Next**.

9. In the Windows service definition panel, select **Automatic** for the Startup type and click **Next,** as shown in Figure 6-7.



Windows service definition

Choose whether to use a Windows service to run the WebSphere ESB. Windows Services can start and stop the WebSphere ESB, and configure startup and recovery actions.

[✓] Run the WebSphere ESB process as a Windows service.

(●) Log on as a local system account

( ) Log on as a specified user account

User name: Administratc

Password:

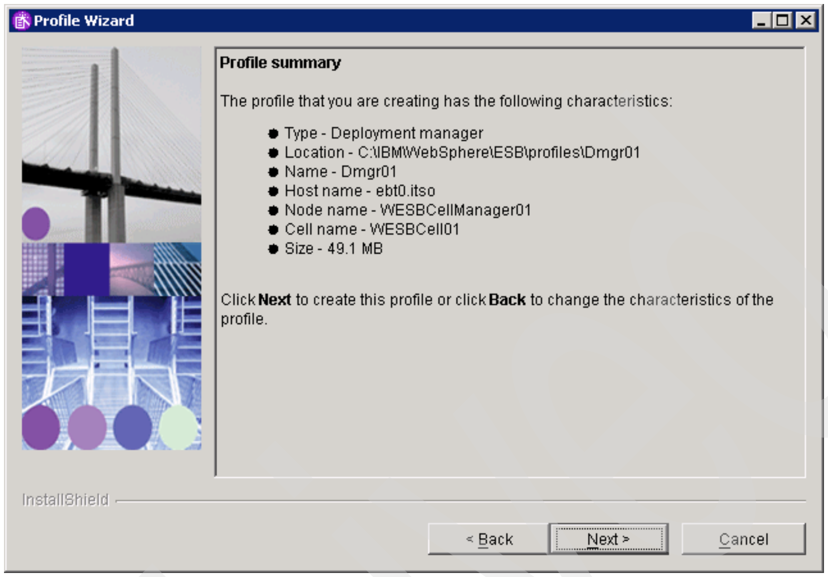
Startup type: Automa...

The user account that runs the Windows Service must have the following user rights:

Figure 6-7 Windows service settings

10. On the Service Component Architecture configuration panel, click **Next**.

11. Confirm that the details are correct on the Profile summary panel and click **Next,** as shown in Figure 6-8.



Profile summary

The profile that you are creating has the following characteristics:

- Type - Deployment manager
- Location - C:\IBM\WebSphere\ESB\profiles\Dmgr01
- Name - Dmgr01
- Host name - ebt0.itso
- Node name - WESBCellManager01
- Cell name - WESBCell01
- Size - 49.1 MB

Click **Next** to create this profile or click **Back** to change the characteristics of the profile.

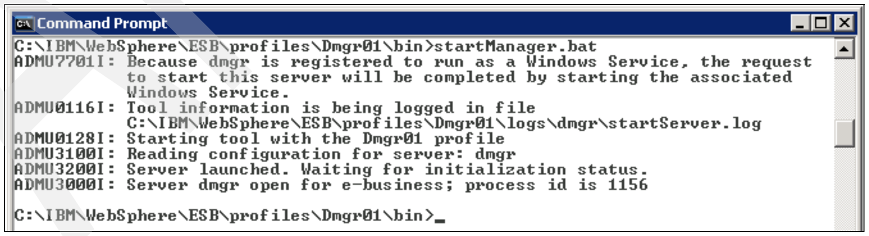
Figure 6-8 Deployment manager creation summary

12. A panel displays the progress. When complete, clear the **Launch the First Steps console** check box and click **Finish**. The deployment manager now exists.

13. Open a command prompt on ebt0, change to the %*WAS_HOME*%\profiles\Dmgr01\bin directory, and issue this command:

```
startManager.bat
```

Figure 6-9 shows the output from this command.



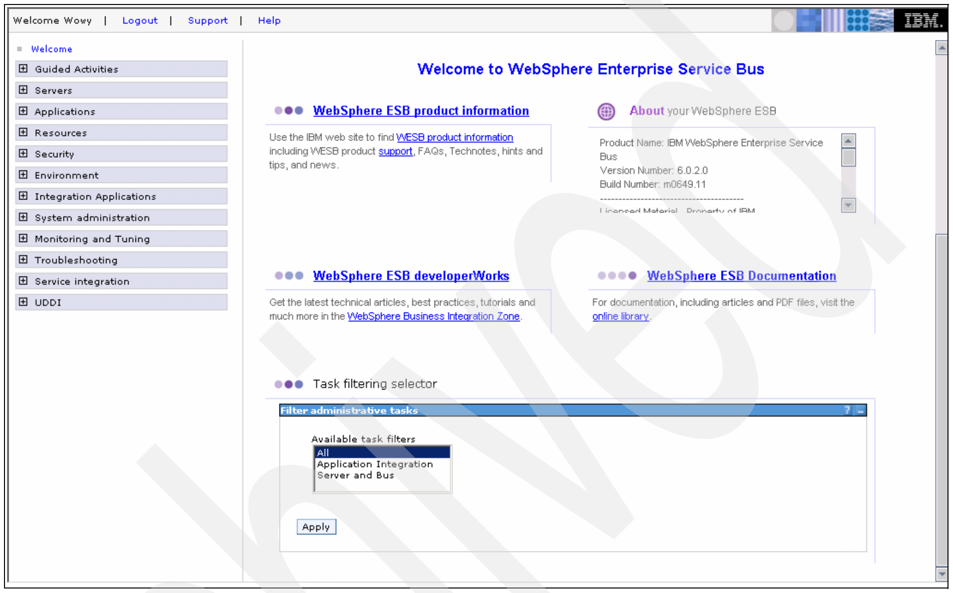
```
C:\IBM\WebSphere\ESB\profiles\Dmgr01\bin>startManager.bat
ADMU7701I: Because dmgr is registered to run as a Windows Service, the request
           to start this server will be completed by starting the associated
           Windows Service.
ADMU0116I: Tool information is being logged in file
           C:\IBM\WebSphere\ESB\profiles\Dmgr01\logs\dmgr\startServer.log
ADMU0128I: Starting tool with the Dmgr01 profile
ADMU3100I: Reading configuration for server: dmgr
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server dmgr open for e-business; process id is 1156

C:\IBM\WebSphere\ESB\profiles\Dmgr01\bin>_
```

Figure 6-9 Starting the deployment manager

14. It should now be possible to connect to the Administrative Console that is running on the deployment manager by opening a browser to:

    http://ebt0.itso:9060/ibm/console

15. Log in to the Administrative Console (as any user). Figure 6-10 shows the Web page that opens.



Welcome to WebSphere Enterprise Service Bus

WebSphere ESB product information

Use the IBM web site to find WESB product information including WESB product support, FAQs, Technotes, hints and tips, and news.

About your WebSphere ESB

Product Name: IBM WebSphere Enterprise Service Bus
Version Number: 6.0.2.0
Build Number: m0649.11

Licensed Material - Property of IBM

WebSphere ESB developerWorks

Get the latest technical articles, best practices, tutorials and much more in the WebSphere Business Integration Zone.

WebSphere ESB Documentation

For documentation, including articles and PDF files, visit the online library.

Figure 6-10 First login to the Administrative Console

6.2.7 Creating the nodes

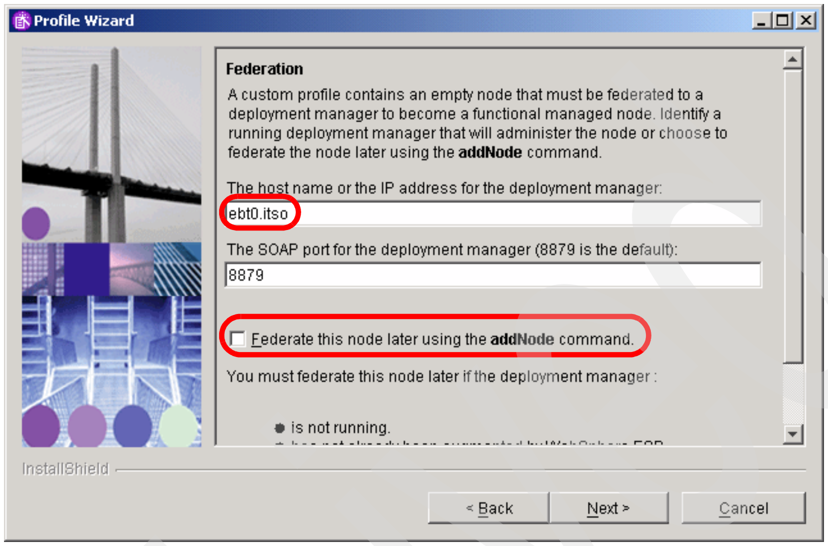
The next step in building this topology is to create the profiles and associated nodes for the topology. The deployment manager must be running on `ebt0.itso` to enable node federation. The nodes are created on `ebt1.itso` and `ebt2.itso`.

Note: Ensure that the time difference between the deployment manager machine and the custom profile machines is within *five* minutes.

To create the nodes, follow these steps:

1. On `ebt1`, open the profile creation wizard by selecting **Start** → **All Programs** → **IBM WebSphere** → **Enterprise Service Bus 6.0** → **Profile creation wizard**.
2. In the Profile creation wizard, click **Next**.
3. Select **Custom profile** radio and click **Next**.

4. Enter the host name of the deployment manager, in this example `ebt0.itso`, leave the SOAP port as the default (8879), ensure that **Federate this node later** is *not* selected, and click **Next**, as shown in Figure 6-11.



Federation

A custom profile contains an empty node that must be federated to a deployment manager to become a functional managed node. Identify a running deployment manager that will administer the node or choose to federate the node later using the **addNode** command.

The host name or the IP address for the deployment manager:

ebt0.itso

The SOAP port for the deployment manager (8879 is the default):

8879

☐ Federate this node later using the **addNode** command.

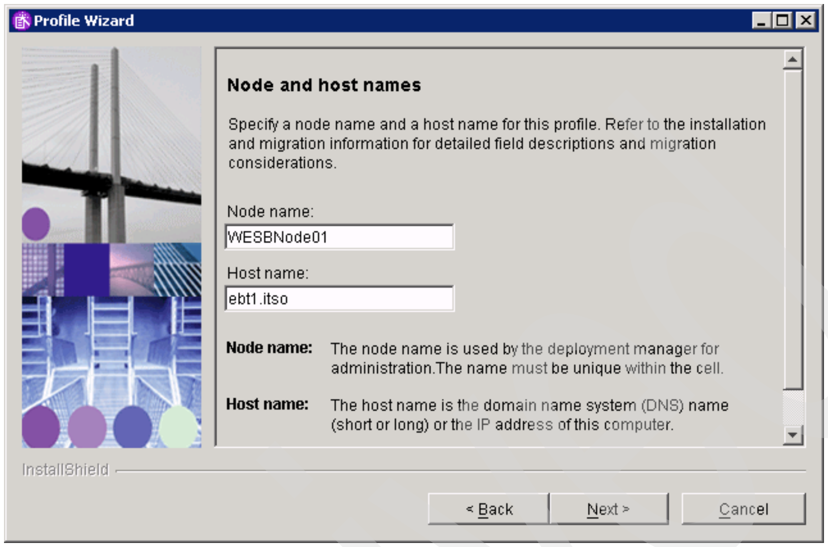
You must federate this node later if the deployment manager :

- is not running.

Figure 6-11 Deployment manager location and federation

5. Set the Profile name appropriately (`WESBNode01`) and click **Next**.
6. Accept the default profile directory and click **Next**.

7. Specify the node name (`WESBNode01`) and host name (`ebt1.itso`) and click **Next**, as shown in Figure 6-12.



Node and host names

Specify a node name and a host name for this profile. Refer to the installation and migration information for detailed field descriptions and migration considerations.

Node name:
WESBNode01

Host name:
ebt1.itso

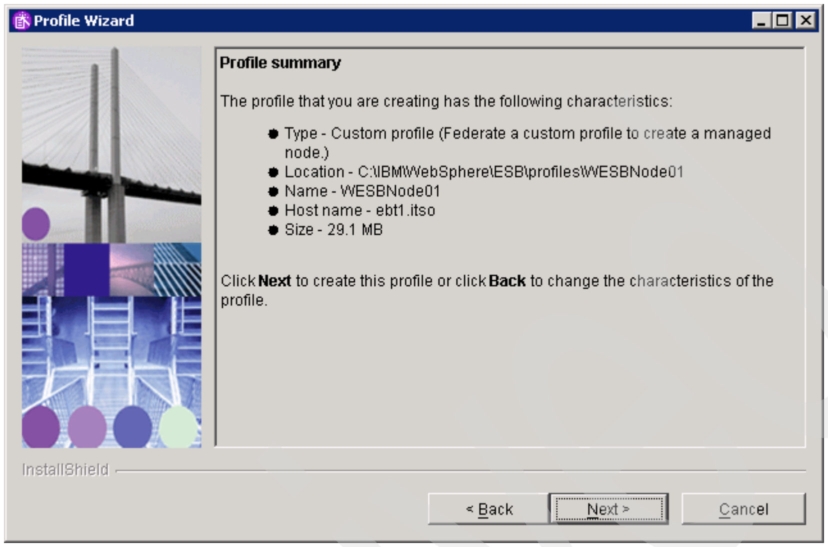
Node name: The node name is used by the deployment manager for administration.The name must be unique within the cell.

Host name: The host name is the domain name system (DNS) name (short or long) or the IP address of this computer.

Figure 6-12 Node and host name

8. Leave the default port numbers and click **Next**.

9. Confirm that the details are correct on the Profile summary panel and click **Next**, as shown in Figure 6-13.



Profile summary

The profile that you are creating has the following characteristics:

- Type - Custom profile (Federate a custom profile to create a managed node.)
- Location - C:\IBM\WebSphere\ESB\profiles\WESBNode01
- Name - WESBNode01
- Host name - ebt1.itso
- Size - 29.1 MB

Click **Next** to create this profile or click **Back** to change the characteristics of the profile.

Figure 6-13 Node creation summary

10. A panel displays the progress. When complete, clear the "Launch the First Steps console" check box and click **Finish**.

The node now exists and is federated into the cell. The federation process starts the node automatically.

Repeat these steps for the second node, using the values shown in Table 6-3.

Table 6-3 Second node details

Setting	Value
Profile name	WESBNode02
Node name	WESBNode02
Host name	ebt2.itso

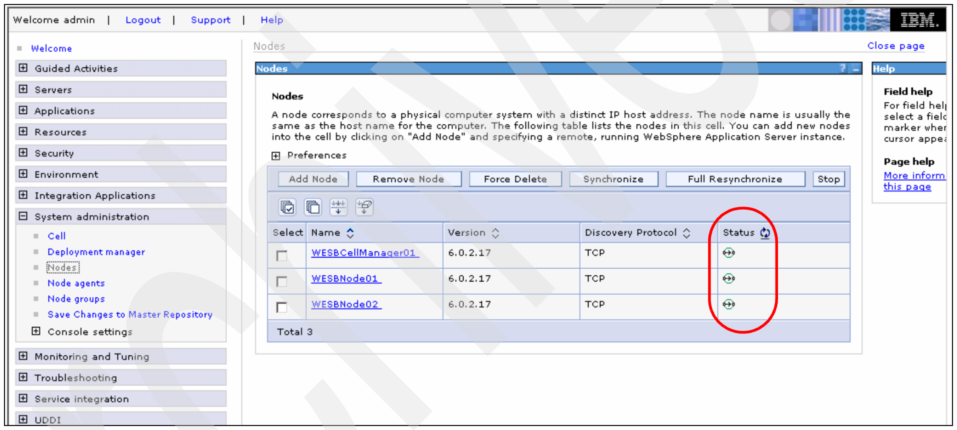
After you have created both nodes, verify that they are running in the Administrative Console by following these steps:

1. Open a browser to `http://ebt0:9060/ibm/console`.

   > **Note:** You need to logout from any existing console sessions.

2. Navigate to **System administration** → **Nodes** and verify that all nodes are running as indicated by the green symbol next to each node, as shown in Figure 6-14.

   > **Note:** Upon initial log in to the Administrative Console, you might need to change Task filter selection to *All* and click **Apply** to view all available functionality.



Nodes

A node corresponds to a physical computer system with a distinct IP host address. The node name is usually the same as the host name for the computer. The following table lists the nodes in this cell. You can add new nodes into the cell by clicking on "Add Node" and specifying a remote, running WebSphere Application Server instance.

Select	Name	Version	Discovery Protocol	Status
☐	WESBCellManager01	6.0.2.17	TCP	⊕
☐	WESBNode01	6.0.2.17	TCP	⊕
☐	WESBNode02	6.0.2.17	TCP	⊕
Total 3				

Figure 6-14 Successful node installation and startup

You have now constructed the topology shown in Figure 6-15.

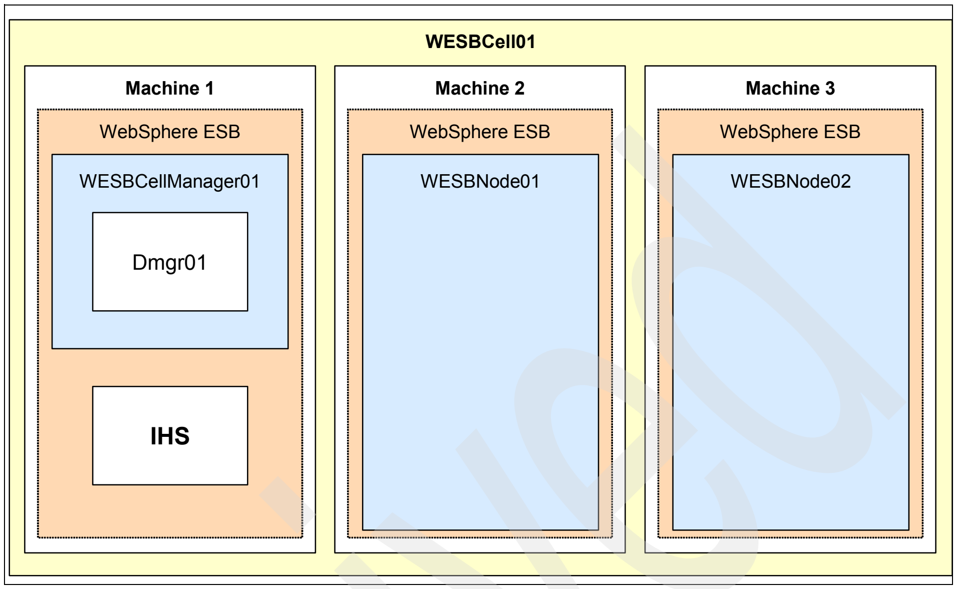


Figure 6-15 Cell and nodes defined

6.3 Specific configuration steps for the Basic topology

This section describes the following:

- Configuring database connectivity
- Configuring the cell
- Verifying the topology

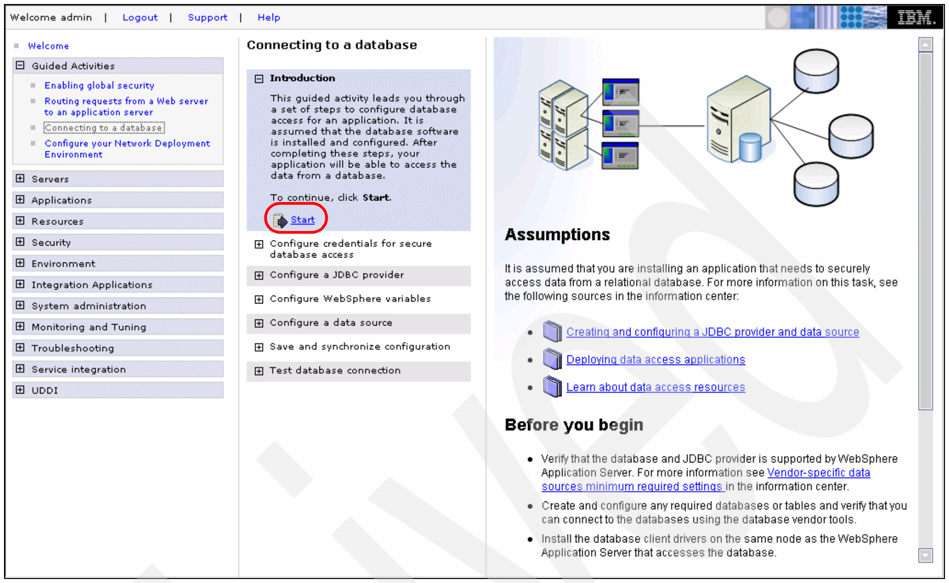
6.3.1 Configuring database connectivity

This section describes how to connect successfully to the database that is required by WebSphere ESB in the Basic topology (the MEDB - Messaging database):

1. On `ebt0`, open the cell deployment manager Administrative Console and login. We used the following URL:

   ```
   http://ebt0.itso:9060/ibm/console
   ```

2. Navigate to **Guided Activities** → **Connecting to a database** and click Start, as shown in Figure 6-16.



Welcome admin | Logout | Support | Help

Connecting to a database

Introduction

This guided activity leads you through a set of steps to configure database access for an application. It is assumed that the database software is installed and configured. After completing these steps, your application will be able to access the data from a database.

To continue, click **Start**.

Start

Configure credentials for secure database access
Configure a JDBC provider
Configure WebSphere variables
Configure a data source
Save and synchronize configuration
Test database connection

Assumptions

It is assumed that you are installing an application that needs to securely access data from a relational database. For more information on this task, see the following sources in the information center:

- Creating and configuring a JDBC provider and data source
- Deploying data access applications
- Learn about data access resources

Before you begin

- Verify that the database and JDBC provider is supported by WebSphere Application Server. For more information see Vendor-specific data sources minimum required settings in the information center.
- Create and configure any required databases or tables and verify that you can connect to the databases using the database vendor tools.
- Install the database client drivers on the same node as the WebSphere Application Server that accesses the database.

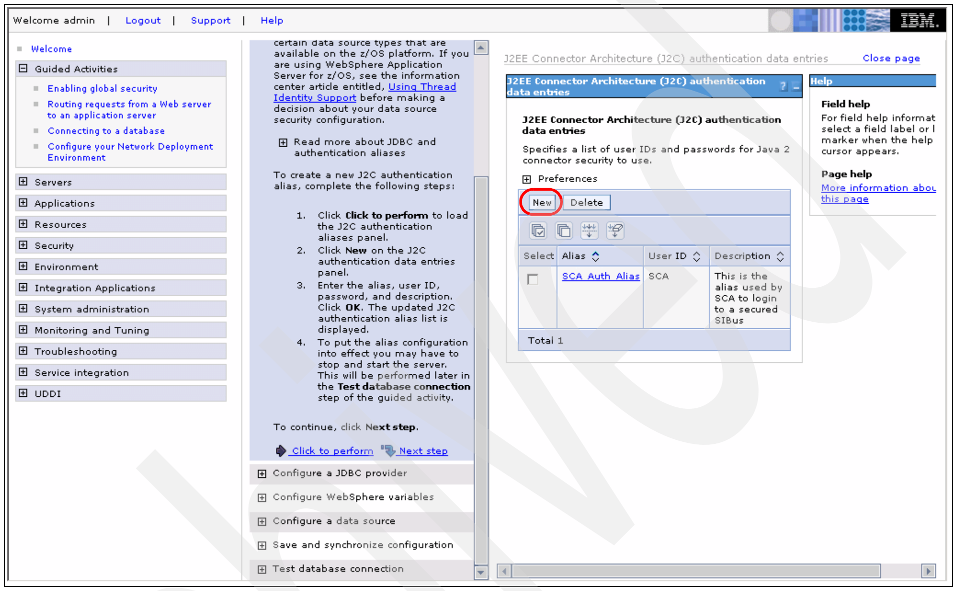
Figure 6-16 Connecting to a database guided activity

3. Select **Click to perform** in the new panel.

Authenticating the alias

You need to set the user and password values for database access. Follow these steps:

1. Click **New** to create a new authentication alias to connect to the database, as shown in Figure 6-17.



Welcome admin | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

certain data source types that are available on the z/OS platform. If you are using WebSphere Application Server for z/OS, see the information center article entitled, Using Thread Identity Support before making a decision about your data source security configuration.

Read more about JDBC and authentication aliases

To create a new J2C authentication alias, complete the following steps:

1. Click **Click to perform** to load the J2C authentication aliases panel.
2. Click **New** on the J2C authentication data entries panel.
3. Enter the alias, user ID, password, and description. Click **OK**. The updated J2C authentication alias list is displayed.
4. To put the alias configuration into effect you may have to stop and start the server. This will be performed later in the **Test database connection** step of the guided activity.

To continue, click **Next step**.

Click to perform   Next step

- Configure a JDBC provider
- Configure WebSphere variables
- Configure a data source
- Save and synchronize configuration
- Test database connection

J2EE Connector Architecture (J2C) authentication data entries   Close page

J2EE Connector Architecture (J2C) authentication data entries

J2EE Connector Architecture (J2C) authentication data entries

Specifies a list of user IDs and passwords for Java 2 connector security to use.

Preferences

New   Delete

Select	Alias	User ID	Description
☐	SCA_Auth_Alias	SCA	This is the alias used by SCA to login to a secured SIBus
Total 1			

Help

Field help

For field help informat select a field label or l marker when the help cursor appears.

Page help

More information abou this page

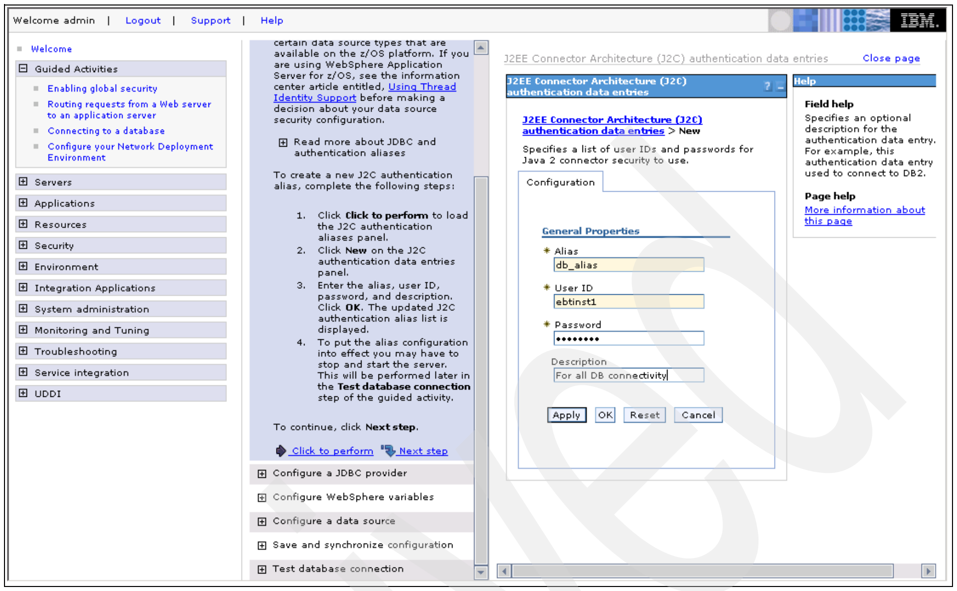
Figure 6-17 Creating a new authentication alias

2. Complete the appropriate information for the database, as shown in Table 6-4, and click **OK**.

Table 6-4 Authentication alias parameters

Parameter	Value
Alias	db_alias
User ID	ebtinst1
Password	itso4you
Description	For all DB connectivity

Figure 6-18 shows an example.



certain data source types that are available on the z/OS platform. If you are using WebSphere Application Server for z/OS, see the information center article entitled, Using Thread Identity Support before making a decision about your data source security configuration.

⊞ Read more about JDBC and authentication aliases

To create a new J2C authentication alias, complete the following steps:

1. Click **Click to perform** to load the J2C authentication aliases panel.
2. Click **New** on the J2C authentication data entries panel.
3. Enter the alias, user ID, password, and description. Click **OK**. The updated J2C authentication alias list is displayed.
4. To put the alias configuration into effect you may have to stop and start the server. This will be performed later in the **Test database connection** step of the guided activity.

To continue, click **Next step**.

Click to perform
Next step
⊞ Configure a JDBC provider
⊞ Configure WebSphere variables
⊞ Configure a data source
⊞ Save and synchronize configuration
⊞ Test database connection
J2EE Connector Architecture (J2C) authentication data entries
Close page
J2EE Connector Architecture (J2C) authentication data entries

J2EE Connector Architecture (J2C) authentication data entries > **New**

Specifies a list of user IDs and passwords for Java 2 connector security to use.

Configuration

General Properties

* Alias	db_alias
* User ID	ebtinst1
* Password	••••••••
Description	For all DB connectivity

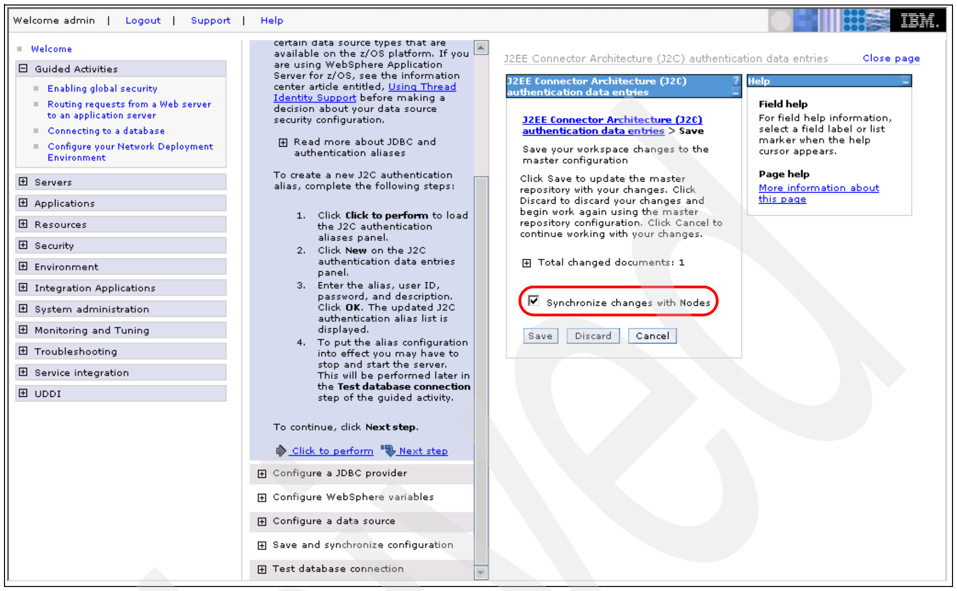
Field help
Specifies an optional description for the authentication data entry. For example, this authentication data entry used to connect to DB2.

Page help
More information about this page

Figure 6-18 Authentication data entries

3. After the values are configured, click **Save** towards the top of the Web page.

4. Select **Synchronize changes with Nodes** and click **Save**, as shown in Figure 6-19.



Welcome admin | Logout | Support | Help

Welcome
Guided Activities
Enabling global security
Routing requests from a Web server to an application server
Connecting to a database
Configure your Network Deployment Environment
Servers
Applications
Resources
Security
Environment
Integration Applications
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI

certain data source types that are available on the z/OS platform. If you are using WebSphere Application Server for z/OS, see the information center article entitled, Using Thread Identity Support before making a decision about your data source security configuration.

Read more about JDBC and authentication aliases

To create a new J2C authentication alias, complete the following steps:

1. Click **Click to perform** to load the J2C authentication aliases panel.
2. Click **New** on the J2C authentication data entries panel.
3. Enter the alias, user ID, password, and description. Click **OK**. The updated J2C authentication alias list is displayed.
4. To put the alias configuration into effect you may have to stop and start the server. This will be performed later in the **Test database connection** step of the guided activity.

To continue, click **Next step**.

Click to perform Next step

Configure a JDBC provider
Configure WebSphere variables
Configure a data source
Save and synchronize configuration
Test database connection

J2EE Connector Architecture (J2C) authentication data entries Close page

J2EE Connector Architecture (J2C) authentication data entries

J2EE Connector Architecture (J2C) authentication data entries > **Save**

Save your workspace changes to the master configuration

Click Save to update the master repository with your changes. Click Discard to discard your changes and begin work again using the master repository configuration. Click Cancel to continue working with your changes.

Total changed documents: 1

[✓] Synchronize changes with Nodes

Save Discard Cancel

Help

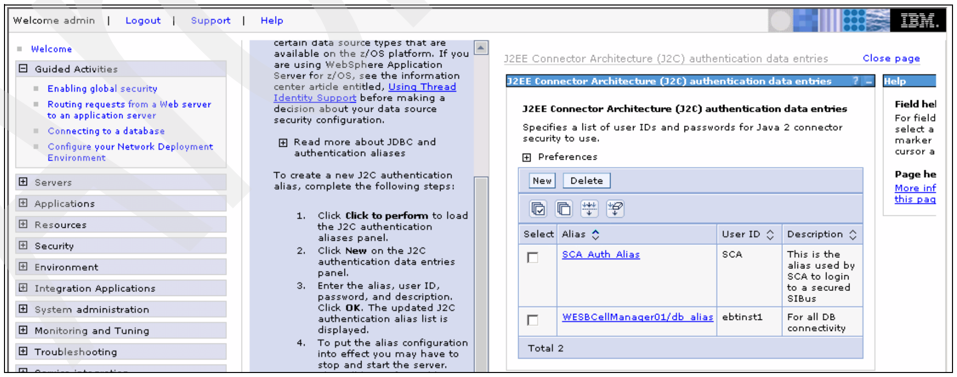
Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

Figure 6-19 Synchronizing the changes

5. When completed, verify that there were no errors with the synchronization and click **OK**.

A new authentication alias called `WESBCellManager01/db_alias` is created, as shown in Figure 6-20.



Welcome admin | Logout | Support | Help

Welcome
Guided Activities
Enabling global security
Routing requests from a Web server to an application server
Connecting to a database
Configure your Network Deployment Environment
Servers
Applications
Resources
Security
Environment
Integration Applications
System administration
Monitoring and Tuning
Troubleshooting

certain data source types that are available on the z/OS platform. If you are using WebSphere Application Server for z/OS, see the information center article entitled, Using Thread Identity Support before making a decision about your data source security configuration.

Read more about JDBC and authentication aliases

To create a new J2C authentication alias, complete the following steps:

1. Click **Click to perform** to load the J2C authentication aliases panel.
2. Click **New** on the J2C authentication data entries panel.
3. Enter the alias, user ID, password, and description. Click **OK**. The updated J2C authentication alias list is displayed.
4. To put the alias configuration into effect you may have to stop and start the server.

J2EE Connector Architecture (J2C) authentication data entries Close page

J2EE Connector Architecture (J2C) authentication data entries

J2EE Connector Architecture (J2C) authentication data entries

Specifies a list of user IDs and passwords for Java 2 connector security to use.

Preferences

New Delete

Select	Alias	User ID	Description
☐	SCA_Auth_Alias	SCA	This is the alias used by SCA to login to a secured SIBus
☐	WESBCellManager01/db_alias	ebtinst1	For all DB connectivity
Total 2			

Help

Field hel
For field select a marker cursor a

Page he
More inf this pag

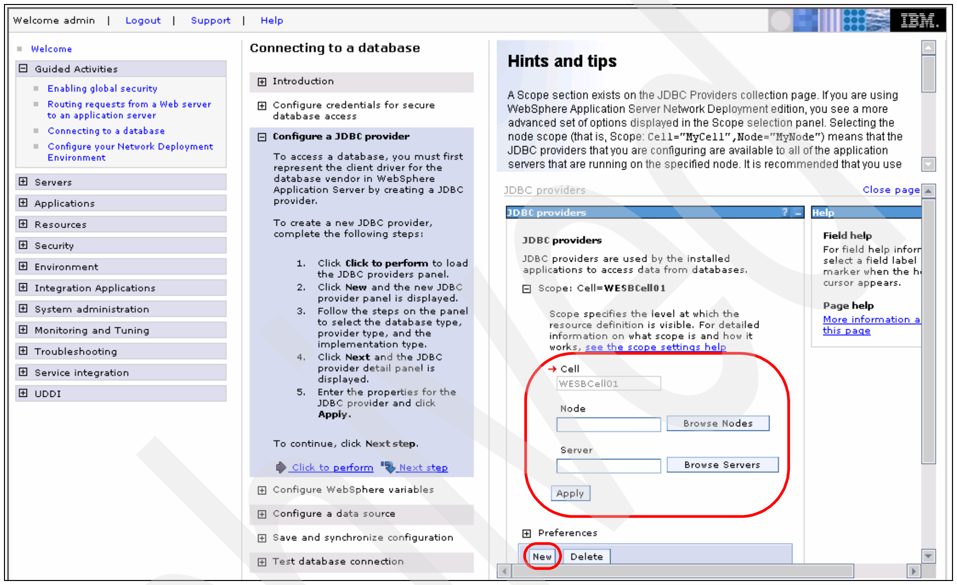
Figure 6-20 New authentication alias created

6. Select **Next step** to continue to the Configure a JDBC™ provider panel.

Creating a JDBC provider

The next step creates a JDBC provider for use by the data sources later:

1. In this panel, select **Click to perform**.
2. Set the scope to cell level only by clearing the Node and Server entry boxes and clicking Apply, as shown in Figure 6-21.



Configure a JDBC provider

To access a database, you must first represent the client driver for the database vendor in WebSphere Application Server by creating a JDBC provider.

To create a new JDBC provider, complete the following steps:

1. Click **Click to perform** to load the JDBC providers panel.
2. Click **New** and the new JDBC provider panel is displayed.
3. Follow the steps on the panel to select the database type, provider type, and the implementation type.
4. Click **Next** and the JDBC provider detail panel is displayed.
5. Enter the properties for the JDBC provider and click **Apply**.

To continue, click **Next step**.

Hints and tips

A Scope section exists on the JDBC Providers collection page. If you are using WebSphere Application Server Network Deployment edition, you see a more advanced set of options displayed in the Scope selection panel. Selecting the node scope (that is, Scope: `Cell="MyCell",Node="MyNode"`) means that the JDBC providers that you are configuring are available to all of the application servers that are running on the specified node. It is recommended that you use

JDBC providers

JDBC providers are used by the installed applications to access data from databases.

Scope: Cell=**WESBCell01**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help

Cell: WESBCell01

Node:

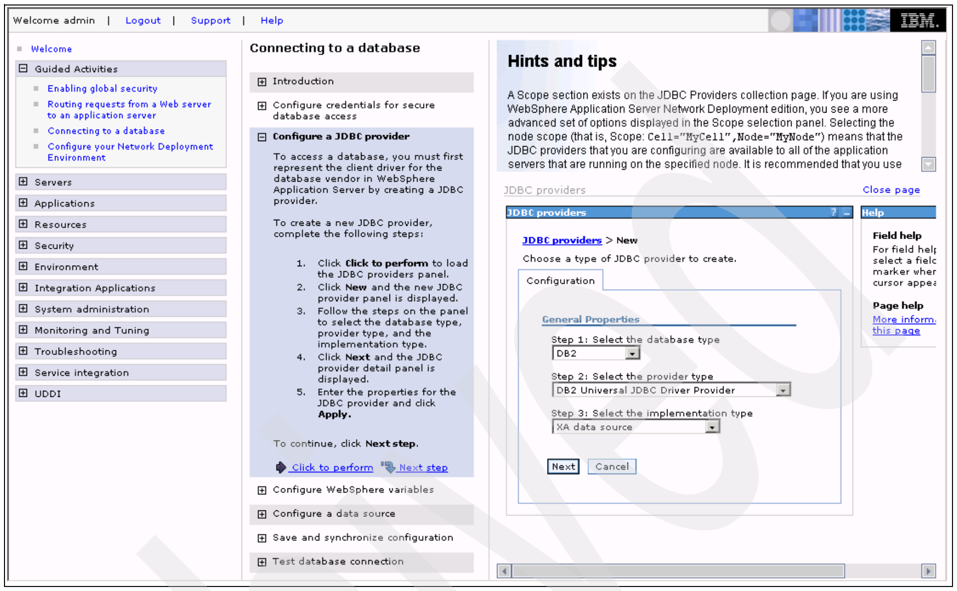
Server:

For field help inforr select a field label marker when the h cursor appears.

Figure 6-21   Setting the cell scope

3. In the JDBC providers panel below the scope, click **New**, also highlighted in Figure 6-21.

4. From the Configuration panel, select **DB2** as the database type, set the provider type to **DB2 Universal JDBC Driver Provider**, and the implementation type to **XA data source**. Click **Next**, as shown in Figure 6-22.



Welcome admin | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Connecting to a database

- Introduction
- Configure credentials for secure database access
- **Configure a JDBC provider**

  To access a database, you must first represent the client driver for the database vendor in WebSphere Application Server by creating a JDBC provider.

  To create a new JDBC provider, complete the following steps:

  1. Click **Click to perform** to load the JDBC providers panel.
  2. Click **New** and the new JDBC provider panel is displayed.
  3. Follow the steps on the panel to select the database type, provider type, and the implementation type.
  4. Click **Next** and the JDBC provider detail panel is displayed.
  5. Enter the properties for the JDBC provider and click **Apply.**

  To continue, click **Next step.**

  Click to perform Next step
- Configure WebSphere variables
- Configure a data source
- Save and synchronize configuration
- Test database connection

Hints and tips

A Scope section exists on the JDBC Providers collection page. If you are using WebSphere Application Server Network Deployment edition, you see a more advanced set of options displayed in the Scope selection panel. Selecting the node scope (that is, Scope: `Cell="MyCell",Node="MyNode"`) means that the JDBC providers that you are configuring are available to all of the application servers that are running on the specified node. It is recommended that you use

JDBC providers Close page

JDBC providers

JDBC providers > **New**

Choose a type of JDBC provider to create.

Configuration

General Properties

Step 1: Select the database type
DB2

Step 2: Select the provider type
DB2 Universal JDBC Driver Provider

Step 3: Select the implementation type
XA data source

Next Cancel

Help

Field help
For field hel
select a fielc
marker wher
cursor appe

Page help
More inform.
this page

Figure 6-22 JDBC provider configuration

5. In the next panel, which names the provider and locates the driver files, accept all the defaults and click **OK**.
6. Save and synchronize these changes and click **OK** when complete.
7. Select **Next step** to open the WebSphere variables panel and then in this panel select **Click to perform**.

Setting the WebSphere variables

The JDBC provider uses WebSphere environment variables to identify the driver file locations. For these variables to resolve, you need to set their values appropriately. The scope for these variables should be **Cell** or **Node**, as shown in Table 6-5 and Table 6-6.

Note: When setting the scope, the selection is not applied until you click **Apply**.

To set the WebSphere variables:

1. Clear the node and server entries, and click **Apply** to set the active scope to the cell.
2. Create or modify each environment variable, and set the value as defined in Table 6-5.

Table 6-5 Cell scope environment variables

Name	**Value**	**Scope**
UNIVERSAL_JDBC_DRIVER_PATH	c:\db2client	Cell
DB2UNIVERSAL_JDBC_DRIVER_PATH	c:\db2client	Cell
DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH	c:\db2client	Cell

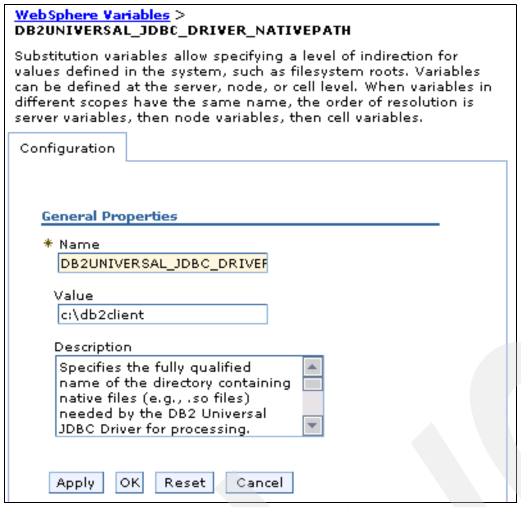
3. When you have defined all three variables at the cell level, click **Browse Nodes**. Select **WESBNode01** and click **OK**. Click **Apply** to set the active scope to the node.
4. Define the node scope environment variable DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH, and set the value as shown in Table 6-6. Also delete the other two variables shown in Table 6-6.

Attention: At the node scope, you *must* delete the environment variables UNIVERSAL_JDBC_DRIVER_PATH and DB2UNIVERSAL_JDBC_DRIVER_PATH.

Table 6-6 Node scope environment variables

Name	**Value**	**Scope**
DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH	c:\db2client	WESBNode01 WESBNode02
UNIVERSAL_JDBC_DRIVER_PATH	*Delete this variable*	WESBNode01 WESBNode02
DB2UNIVERSAL_JDBC_DRIVER_PATH	*Delete this variable*	WESBNode01 WESBNode02

Figure 6-23 provides an example of the environment variable configuration.


WebSphere Variables > DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH

Substitution variables allow specifying a level of indirection for values defined in the system, such as filesystem roots. Variables can be defined at the server, node, or cell level. When variables in different scopes have the same name, the order of resolution is server variables, then node variables, then cell variables.

Configuration

General Properties

* Name: DB2UNIVERSAL_JDBC_DRIVEF

Value: c:\db2client

Description: Specifies the fully qualified name of the directory containing native files (e.g., .so files) needed by the DB2 Universal JDBC Driver for processing.

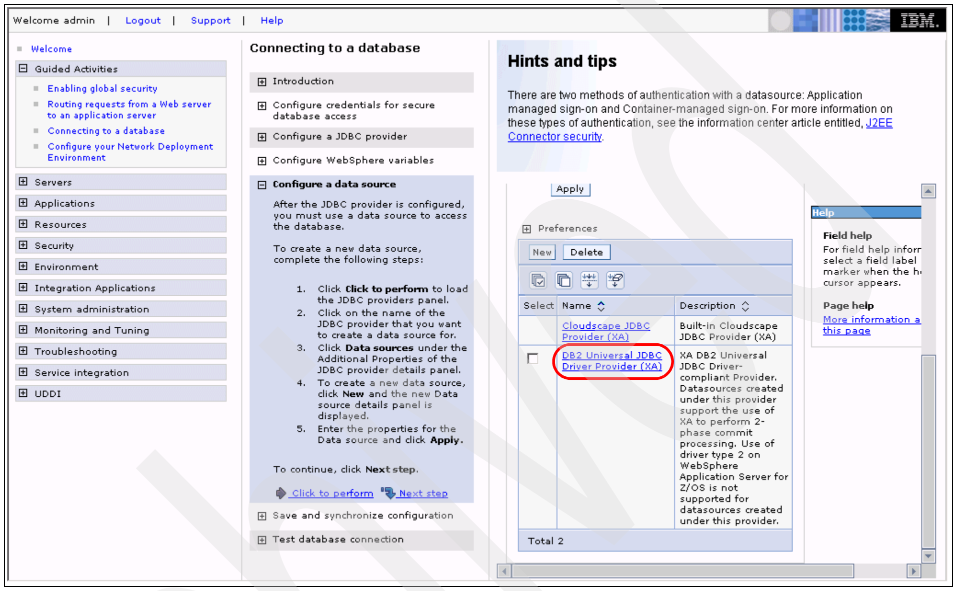
Figure 6-23 Setting an environment variable value

5. Repeat this process for `WESBNode02` to remove the two environment variables, and set the third to the value shown in Table 6-6 on page 119.
6. Save and synchronize these changes.
7. Select **Next step** and then **Click to perform**.

Configuring the data sources

The next step is to configure the data sources using the JDBC provider that you just created:

1. In the new panel, click the JDBC provider **DB2 Universal JDBC Driver Provider (XA)**, as shown in Figure 6-24.



Welcome admin | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Connecting to a database

- Introduction
- Configure credentials for secure database access
- Configure a JDBC provider
- Configure WebSphere variables
- **Configure a data source**

  After the JDBC provider is configured, you must use a data source to access the database.

  To create a new data source, complete the following steps:

  1. Click **Click to perform** to load the JDBC providers panel.
  2. Click on the name of the JDBC provider that you want to create a data source for.
  3. Click **Data sources** under the Additional Properties of the JDBC provider details panel.
  4. To create a new data source, click **New** and the new Data source details panel is displayed.
  5. Enter the properties for the Data source and click **Apply**.

  To continue, click **Next step**.

  Click to perform   Next step
- Save and synchronize configuration
- Test database connection

Hints and tips

There are two methods of authentication with a datasource: Application managed sign-on and Container-managed sign-on. For more information on these types of authentication, see the information center article entitled, J2EE Connector security.

Apply

Preferences

New   Delete

Select	Name	Description
	Cloudscape JDBC Provider (XA)	Built-in Cloudscape JDBC Provider (XA)
☐	DB2 Universal JDBC Driver Provider (XA)	XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on WebSphere Application Server for Z/OS is not supported for datasources created under this provider.
Total 2		

Help

Field help

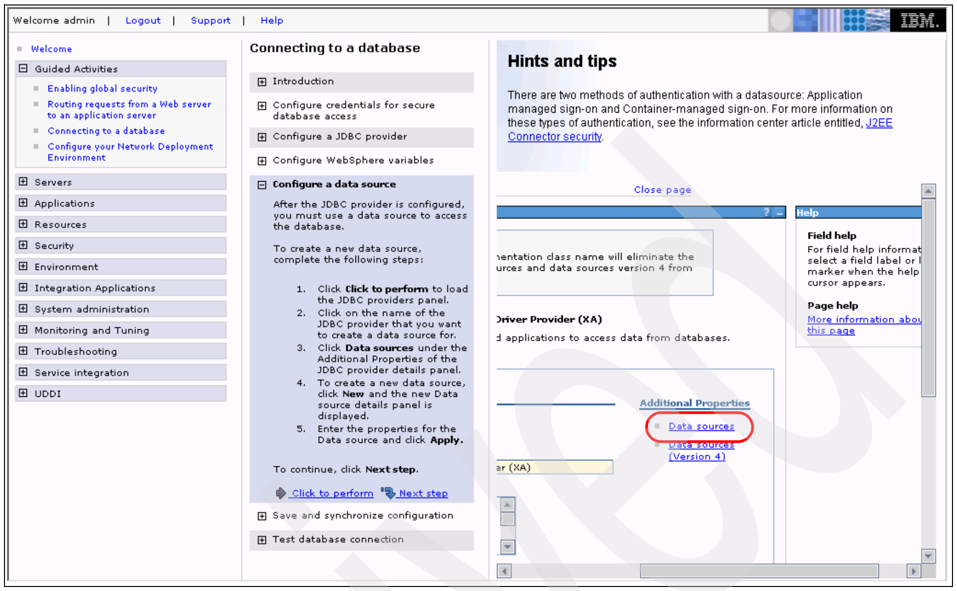
For field help inforr select a field label marker when the h cursor appears.

Page help

More information a this page

Figure 6-24 JDBC provider selection for data source creation

2. Scroll to the right and click **Data sources**, as shown in Figure 6-25.



Connecting to a database

Introduction

Configure credentials for secure database access

Configure a JDBC provider

Configure WebSphere variables

Configure a data source

After the JDBC provider is configured, you must use a data source to access the database.

To create a new data source, complete the following steps:

1. Click **Click to perform** to load the JDBC providers panel.
2. Click on the name of the JDBC provider that you want to create a data source for.
3. Click **Data sources** under the Additional Properties of the JDBC provider details panel.
4. To create a new data source, click **New** and the new Data source details panel is displayed.
5. Enter the properties for the Data source and click **Apply.**

To continue, click **Next step.**

Click to perform Next step

Save and synchronize configuration

Test database connection

Hints and tips

There are two methods of authentication with a datasource: Application managed sign-on and Container-managed sign-on. For more information on these types of authentication, see the information center article entitled, J2EE Connector security.

ɲentation class name will eliminate the
urces and data sources version 4 from

Driver Provider (XA)

d applications to access data from databases.

For field help informat
select a field label or l
marker when the help
cursor appears.

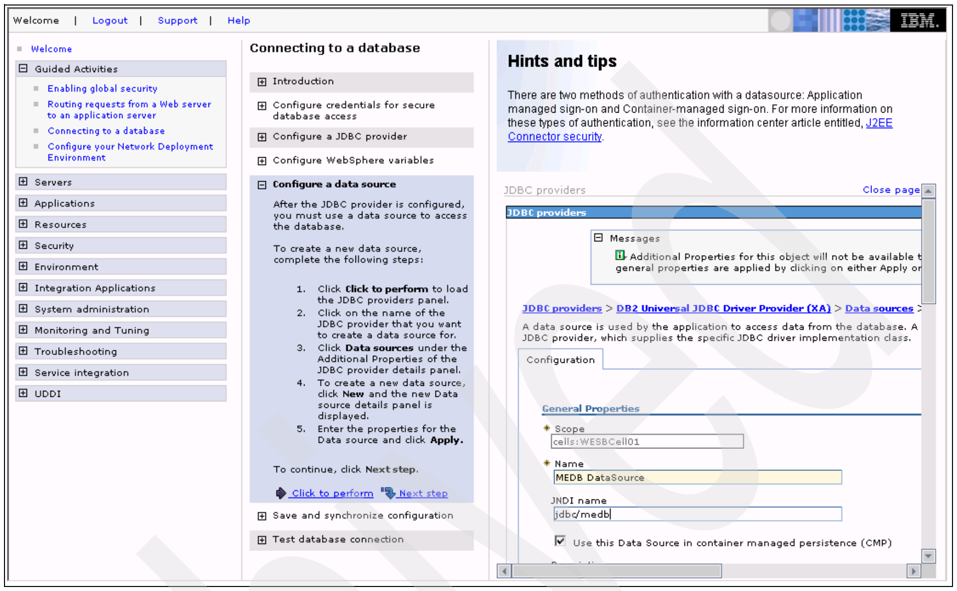
Figure 6-25 Selecting the additional data source property

3. You next need to create a data source for the messaging database, MEDB. Click **New** to create a data source and set the parameters, as shown in Table 6-7.

Table 6-7 Data source configuration

Parameter	Messaging Parameters
Name	MEDB DataSource
JNDI name	jdbc/medb
Description	Messaging engine data source
Component-managed authentication alias	WESBCellManager01/db_alias
Authentication alias for XA recovery	Use component-managed authentication alias
Database name	MEDB
Driver type	4
Server name	db2.itso
Port	50000

Figure 6-26 shows the MEDB data source configuration page. You need to enter the values that are identified in Table 6-7. To enter these values, you need to scroll down in this panel.


Welcome | Logout | Support | Help

Welcome

Guided Activities
- Enabling global security
- Routing requests from a Web server to an application server
- Connecting to a database
- Configure your Network Deployment Environment

Servers
Applications
Resources
Security
Environment
Integration Applications
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI

Connecting to a database

Introduction
Configure credentials for secure database access
Configure a JDBC provider
Configure WebSphere variables
Configure a data source

After the JDBC provider is configured, you must use a data source to access the database.

To create a new data source, complete the following steps:

1. Click **Click to perform** to load the JDBC providers panel.
2. Click on the name of the JDBC provider that you want to create a data source for.
3. Click **Data sources** under the Additional Properties of the JDBC provider details panel.
4. To create a new data source, click **New** and the new Data source details panel is displayed.
5. Enter the properties for the Data source and click **Apply.**

To continue, click Next step.

Click to perform Next step

Save and synchronize configuration
Test database connection

Hints and tips

There are two methods of authentication with a datasource: Application managed sign-on and Container-managed sign-on. For more information on these types of authentication, see the information center article entitled, J2EE Connector security.

JDBC providers Close page

JDBC providers

Messages
Additional Properties for this object will not be available t
general properties are applied by clicking on either Apply or

JDBC providers > DB2 Universal JDBC Driver Provider (XA) > Data sources >

A data source is used by the application to access data from the database. A JDBC provider, which supplies the specific JDBC driver implementation class.

Configuration

General Properties

* Scope
cells:WESBCell01

* Name
MEDB DataSource

JNDI name
jdbc/medb

✓ Use this Data Source in container managed persistence (CMP)

Figure 6-26 Data source configuration

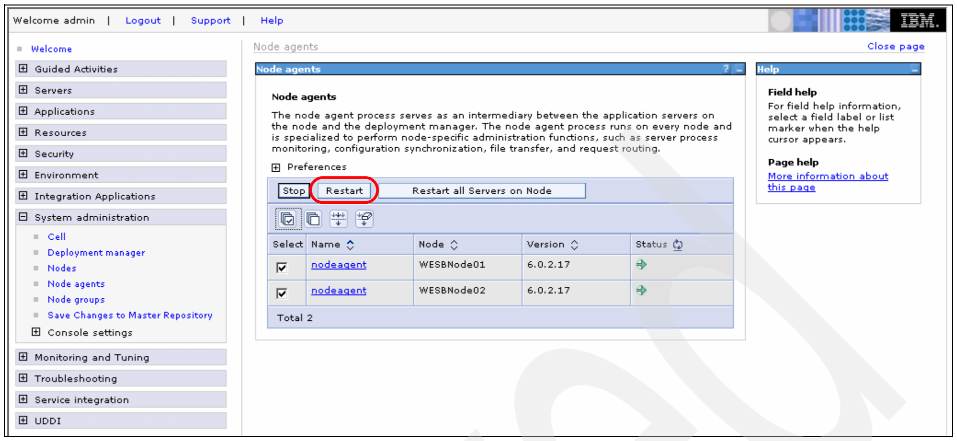
4. Click **OK** to create the new data source. When created save, and synchronize the changes and click **OK** to complete.
5. Click **Next step** and then **Next step** again in the Save and synchronize configuration panel (because you have already saved and synchronized).

Restarting the environment

You need to restart all nodes and the deployment manager to ensure that the changes that you have made are distributed cell wide. Follow these steps:

1. Open the Administrative Console for the deployment manager hosted on `ebt0`.
2. Navigate to **System administration** → **Node agents** and select both nodes.

3. Click **Restart** to cause the nodes to stop and start, as shown in Figure 6-27.



Node agents

The node agent process serves as an intermediary between the application servers on the node and the deployment manager. The node agent process runs on every node and is specialized to perform node-specific administration functions, such as server process monitoring, configuration synchronization, file transfer, and request routing.

Select	Name	Node	Version	Status
✓	nodeagent	WESBNode01	6.0.2.17	➔
✓	nodeagent	WESBNode02	6.0.2.17	➔

Total 2

Help

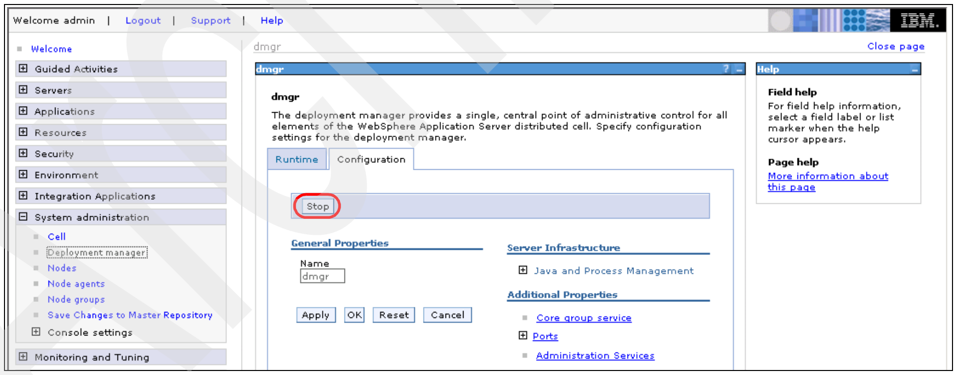
Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

Figure 6-27 Node restart

Note: You need to refresh the Node agents view to see the change in status.

4. Navigate to **System administration** → **Deployment manager** and click **Stop**, as shown in Figure 6-28.



dmgr

The deployment manager provides a single, central point of administrative control for all elements of the WebSphere Application Server distributed cell. Specify configuration settings for the deployment manager.

Name: dmgr

Help

Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

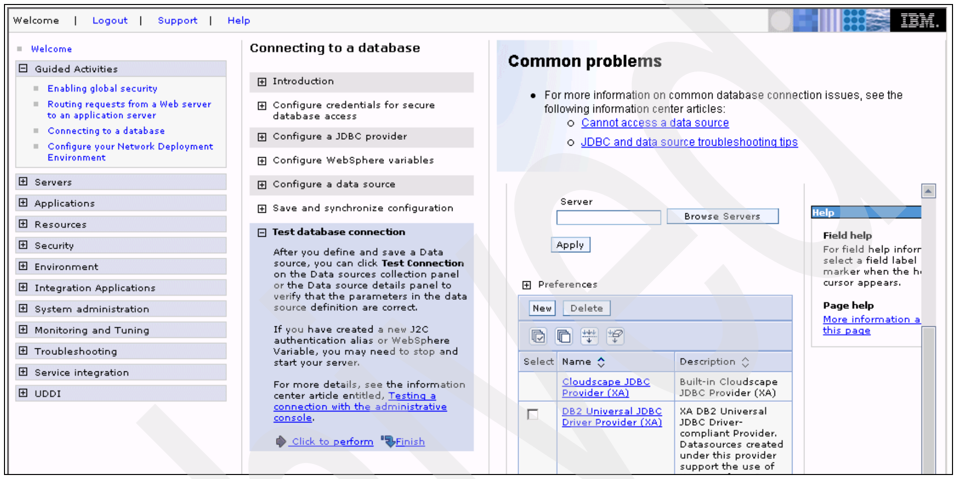
Figure 6-28 Stopping the deployment manager

5. Click **OK** in the verification panel and then start the deployment manager, as described in step 13 of 6.2.6, “Creating the WebSphere ESB cell” on page 103.

Testing database connectivity

After the environment has restarted, ensure that the MEDB database is running and test the newly created data sources:

1. Open the `ebt0` Administrative Console and log in.
2. Navigate to **Guided Activities** → **Connecting to a database** → **Test database connection**, as shown in Figure 6-29.



After you define and save a Data source, you can click **Test Connection** on the Data sources collection panel or the Data source details panel to verify that the parameters in the data source definition are correct.

If you have created a new J2C authentication alias or WebSphere Variable, you may need to stop and start your server.

For more details, see the information center article entitled, Testing a connection with the administrative console.

Common problems

- For more information on common database connection issues, see the following information center articles:
  - Cannot access a data source
  - JDBC and data source troubleshooting tips

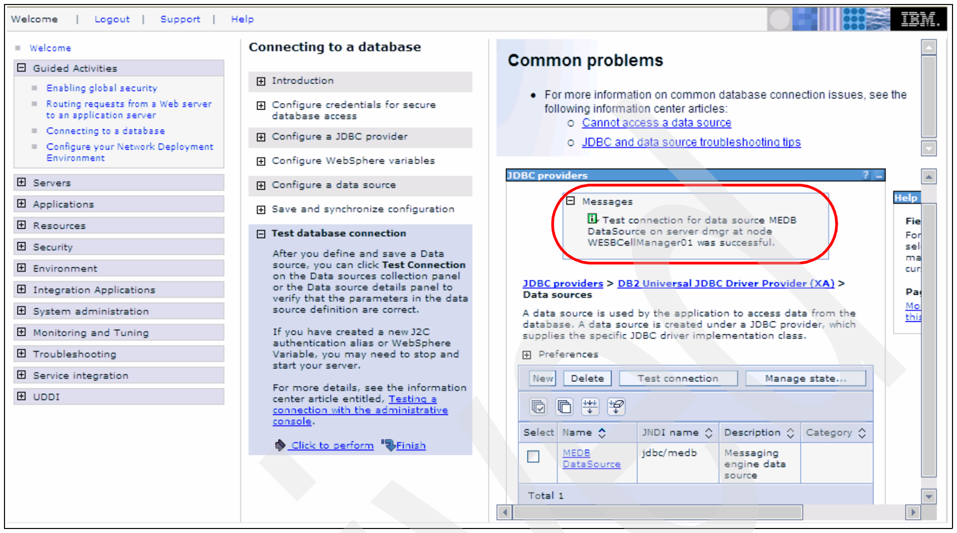
Select	Name	Description
	Cloudscape JDBC Provider (XA)	Built-in Cloudscape JDBC Provider (XA)
☐	DB2 Universal JDBC Driver Provider (XA)	XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of

For field help infor
select a field label
marker when the h
cursor appears.

Figure 6-29 Testing the new data sources

3. Select **Click to perform**.
4. Verify that the scope is set to **Cell** and click the JDBC provider. In this example, the provider is called **DB2 Universal JDBC Driver Provider (XA)**.
5. In the next panel, scroll right and under **Additional Properties**, click **Data sources**.

6. From the data sources panel, select **MEDB DataSource** and click **Test connection**, as shown in Figure 6-30.



Connecting to a database

Test database connection

After you define and save a Data source, you can click **Test Connection** on the Data sources collection panel or the Data source details panel to verify that the parameters in the data source definition are correct.

If you have created a new J2C authentication alias or WebSphere Variable, you may need to stop and start your server.

For more details, see the information center article entitled, Testing a connection with the administrative console.

Common problems

- For more information on common database connection issues, see the following information center articles:
  - Cannot access a data source
  - JDBC and data source troubleshooting tips

Messages

Test connection for data source MEDB DataSource on server dmgr at node WESBCellManager01 was successful.

JDBC providers > DB2 Universal JDBC Driver Provider (XA) >
Data sources

A data source is used by the application to access data from the database. A data source is created under a JDBC provider, which supplies the specific JDBC driver implementation class.

Select	Name	JNDI name	Description	Category
☐	MEDB DataSource	jdbc/medb	Messaging engine data source	
Total 1				

Figure 6-30 Successful database connections

7. The Messages dialog box (at the top in Figure 6-30) should indicate that the data source successfully connected to the database.

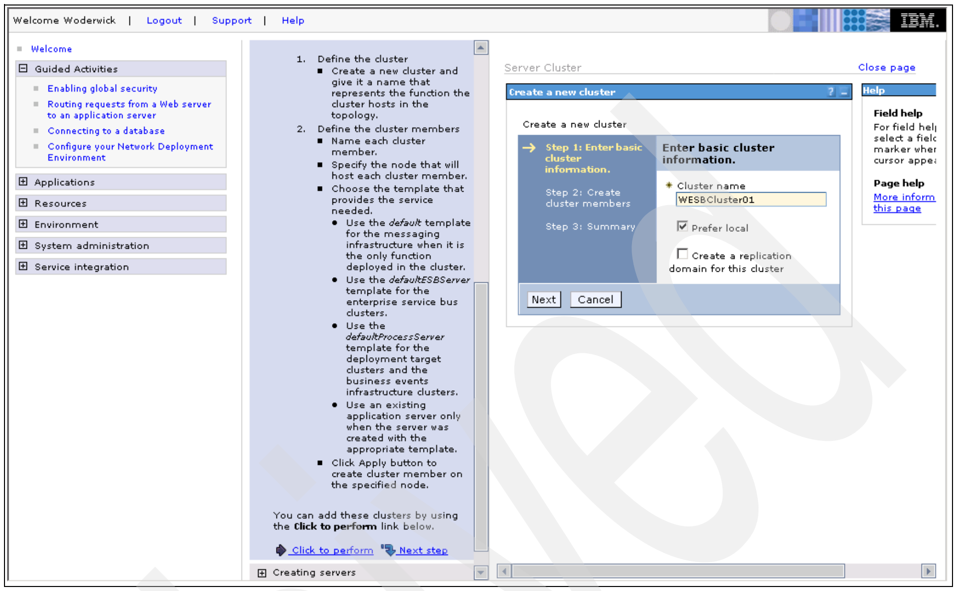
8. Click **Finish**.

6.3.2 Configuring the cell

To configure the network deployment infrastructure with the options that are required for the Basic topology, follow these steps:

1. Open the Administrative Console on `ebt0` and login.
2. Navigate to **Guided Activities** → **Configure your Network Deployment Environment** and click Start.
3. Scroll down and select **Click to perform** in the expanded panel to see the list of nodes.
4. The nodes are already federated, so select **Next step**.
5. The database configuration is already complete, so select **Next step**.
6. In the Creating clusters and cluster members panel, select **Click to perform**.

7. Click **New** and enter a Cluster name of `WESBCluster01`, as shown in Figure 6-31.



1. Define the cluster
   - Create a new cluster and give it a name that represents the function the cluster hosts in the topology.
2. Define the cluster members
   - Name each cluster member.
   - Specify the node that will host each cluster member.
   - Choose the template that provides the service needed.
     - Use the *default* template for the messaging infrastructure when it is the only function deployed in the cluster.
     - Use the *defaultESBServer* template for the enterprise service bus clusters.
     - Use the *defaultProcessServer* template for the deployment target clusters and the business events infrastructure clusters.
     - Use an existing application server only when the server was created with the appropriate template.
   - Click Apply button to create cluster member on the specified node.

You can add these clusters by using the **Click to perform** link below.

Create a new cluster

Step 1: Enter basic cluster information.

Step 2: Create cluster members

Step 3: Summary

Enter basic cluster information.

* Cluster name: WESBCluster01

[✓] Prefer local

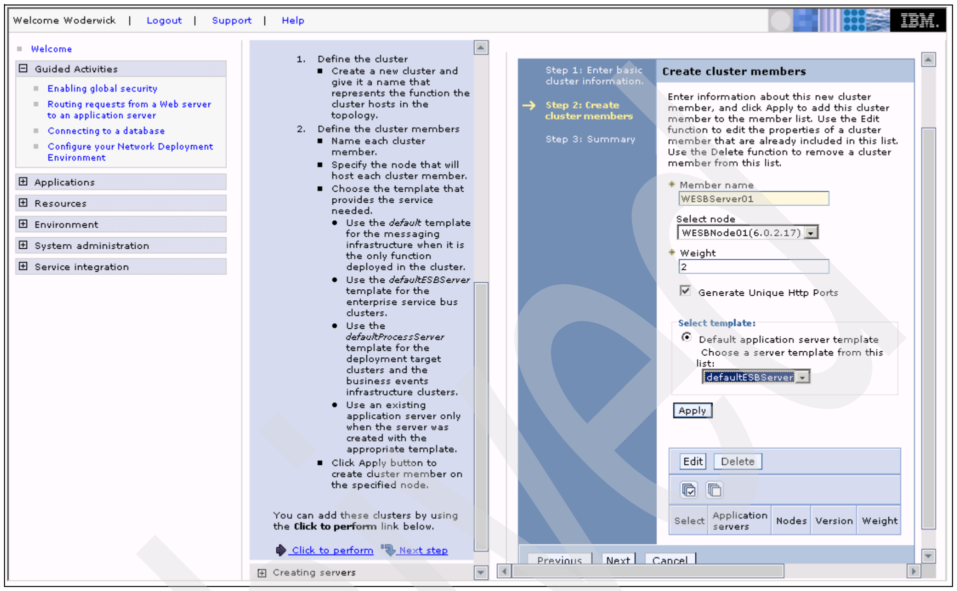
[ ] Create a replication domain for this cluster

Field help
For field help, select a field marker when cursor appears

Page help
More information about this page

Figure 6-31 Cluster creation

8. Click **Next** and then add a first member of the cluster named `WESBServer01` to the node `WESBNode01`. Set the template to `defaultESBServer`, as shown in Figure 6-32.


Welcome Woderwick | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Applications
- Resources
- Environment
- System administration
- Service integration

1. Define the cluster
   - Create a new cluster and give it a name that represents the function the cluster hosts in the topology.
2. Define the cluster members
   - Name each cluster member.
   - Specify the node that will host each cluster member.
   - Choose the template that provides the service needed.
     - Use the *default* template for the messaging infrastructure when it is the only function deployed in the cluster.
     - Use the *defaultESBServer* template for the enterprise service bus clusters.
     - Use the *defaultProcessServer* template for the deployment target clusters and the business events infrastructure clusters.
     - Use an existing application server only when the server was created with the appropriate template.
   - Click Apply button to create cluster member on the specified node.

You can add these clusters by using the **Click to perform** link below.

Click to perform   Next step

Creating servers

Step 1: Enter basic cluster information.
→ Step 2: Create cluster members
Step 3: Summary

Create cluster members

Enter information about this new cluster member, and click Apply to add this cluster member to the member list. Use the Edit function to edit the properties of a cluster member that are already included in this list. Use the Delete function to remove a cluster member from this list.

* Member name: WESBServer01

Select node: WESBNode01(6.0.2.17)

* Weight: 2

✓ Generate Unique Http Ports

Select template:
- Default application server template
- Choose a server template from this list: defaultESBServer

Apply

Edit   Delete

Select	Application servers	Nodes	Version	Weight

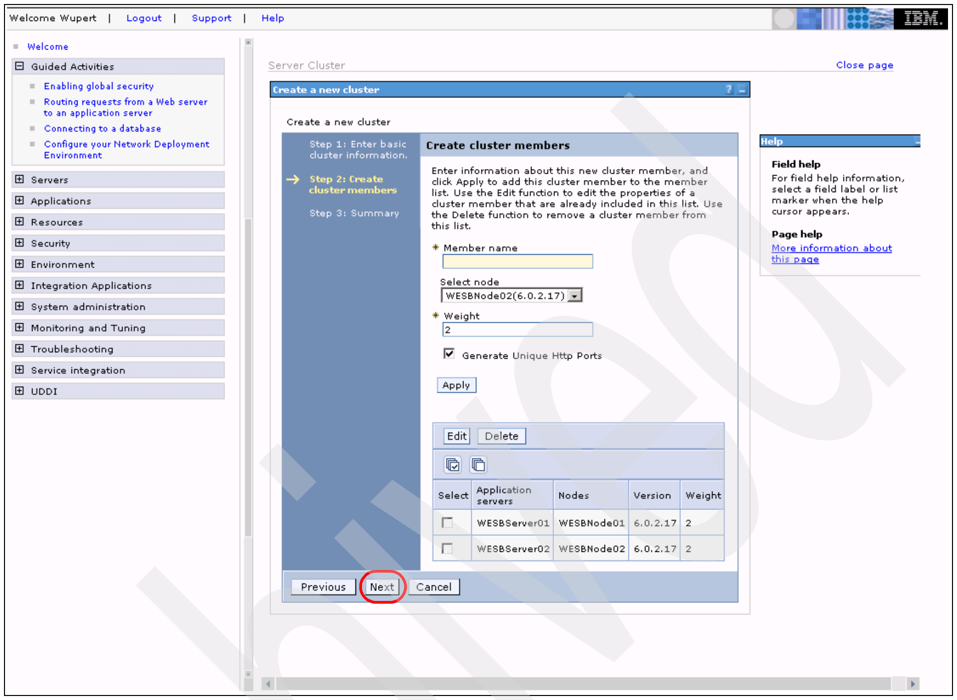
Previous   Next   Cancel

Figure 6-32 Adding the first cluster member

9. Click **Apply** to add this server to the new cluster.

10. Repeat this process for `WESBServer02` on `WESBNode02`. However, the template option will not be available now because all cluster members are of the type that is defined by the first member.

11. Click **Apply** to add the second cluster member and then click **Next**, as shown in Figure 6-33.



Create a new cluster

Create cluster members

Enter information about this new cluster member, and click Apply to add this cluster member to the member list. Use the Edit function to edit the properties of a cluster member that are already included in this list. Use the Delete function to remove a cluster member from this list.

* Member name:

Select node: WESBNode02(6.0.2.17)

* Weight: 2

[✓] Generate Unique Http Ports

Select	Application servers	Nodes	Version	Weight
☐	WESBServer01	WESBNode01	6.0.2.17	2
☐	WESBServer02	WESBNode02	6.0.2.17	2

Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

Figure 6-33 Cluster members added

12. At the summary page, click **Finish**.

13. Save and synchronize the changes.

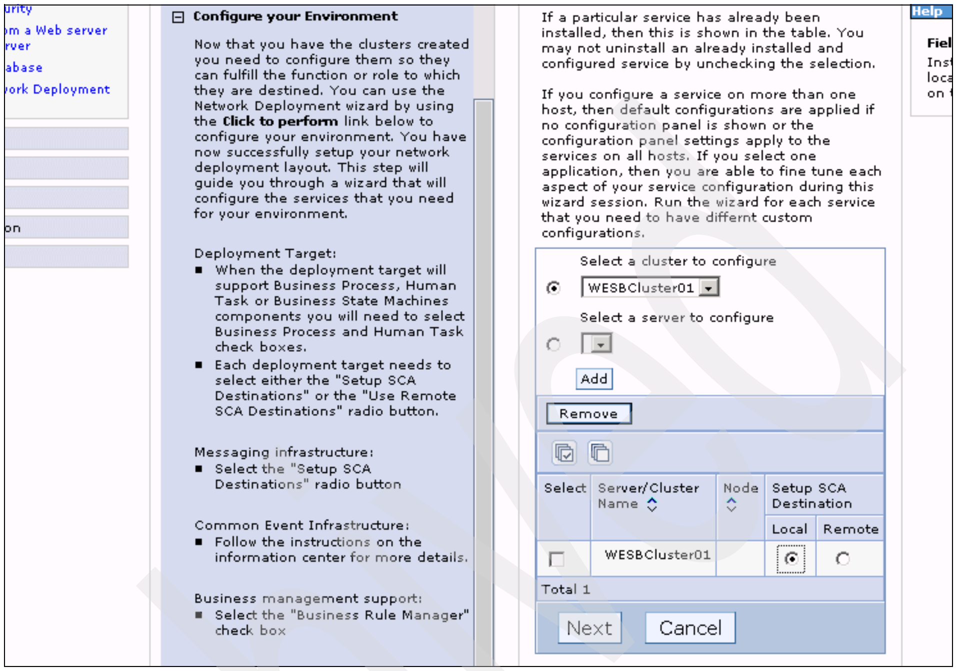
14. Click **Next step** to continue, and then click **Next step** at the **Creating servers** option because you require no more servers.

Configuring the environment

To configure the environment, follow these steps:

1. In the Configure your Environment panel, select **Click to perform**.
2. Click the radio button to configure a cluster.
3. Select WESBCluster01 from the drop-down menu and then click **Add**.

4. Select **Local** for the **SCA Destination** for this cluster, and click **Next,** as shown in Figure 6-34.


urity
om a Web server
rver
abase
ork Deployment
on

Configure your Environment

Now that you have the clusters created you need to configure them so they can fulfill the function or role to which they are destined. You can use the Network Deployment wizard by using the **Click to perform** link below to configure your environment. You have now successfully setup your network deployment layout. This step will guide you through a wizard that will configure the services that you need for your environment.

Deployment Target:
- When the deployment target will support Business Process, Human Task or Business State Machines components you will need to select Business Process and Human Task check boxes.
- Each deployment target needs to select either the "Setup SCA Destinations" or the "Use Remote SCA Destinations" radio button.

Messaging infrastructure:
- Select the "Setup SCA Destinations" radio button

Common Event Infrastructure:
- Follow the instructions on the information center for more details.

Business management support:
- Select the "Business Rule Manager" check box

If a particular service has already been installed, then this is shown in the table. You may not uninstall an already installed and configured service by unchecking the selection.

If you configure a service on more than one host, then default configurations are applied if no configuration panel is shown or the configuration panel settings apply to the services on all hosts. If you select one application, then you are able to fine tune each aspect of your service configuration during this wizard session. Run the wizard for each service that you need to have different custom configurations.

(●) Select a cluster to configure: WESBCluster01

( ) Select a server to configure:

Add

Remove

Select	Server/Cluster Name	Node	Setup SCA Destination	
			Local	Remote
☐	WESBCluster01		●	○

Total 1

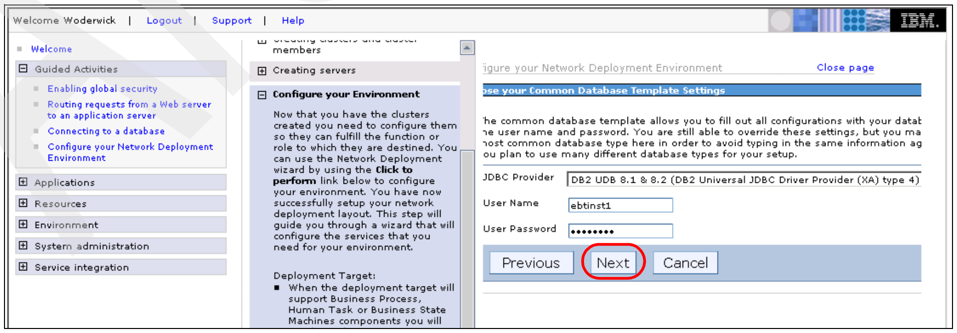
Next | Cancel

Help
Fiel
Inst
loca
on t

Figure 6-34 Configuring SCA locally

5. From the JDBC provider drop-down list, select the appropriate provider. In this example the provider is **DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC Driver Provider (XA) type 4)**.

6. Set the user name to `ebtinst1` and the user password to `itso4you`, and click **Next**, as shown in Figure 6-35.


Welcome Woderwick | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Applications
- Resources
- Environment
- System administration
- Service integration

Creating clusters and cluster members

Creating servers

Configure your Environment

Now that you have the clusters created you need to configure them so they can fulfill the function or role to which they are destined. You can use the Network Deployment wizard by using the **Click to perform** link below to configure your environment. You have now successfully setup your network deployment layout. This step will guide you through a wizard that will configure the services that you need for your environment.

Deployment Target:
- When the deployment target will support Business Process, Human Task or Business State Machines components you will

igure your Network Deployment Environment — Close page

ose your Common Database Template Settings

he common database template allows you to fill out all configurations with your datat
ne user name and password. You are still able to override these settings, but you ma
nost common database type here in order to avoid typing in the same information ag
ou plan to use many different database types for your setup.

JDBC Provider: DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC Driver Provider (XA) type 4)
User Name: ebtinst1
User Password: ••••••••

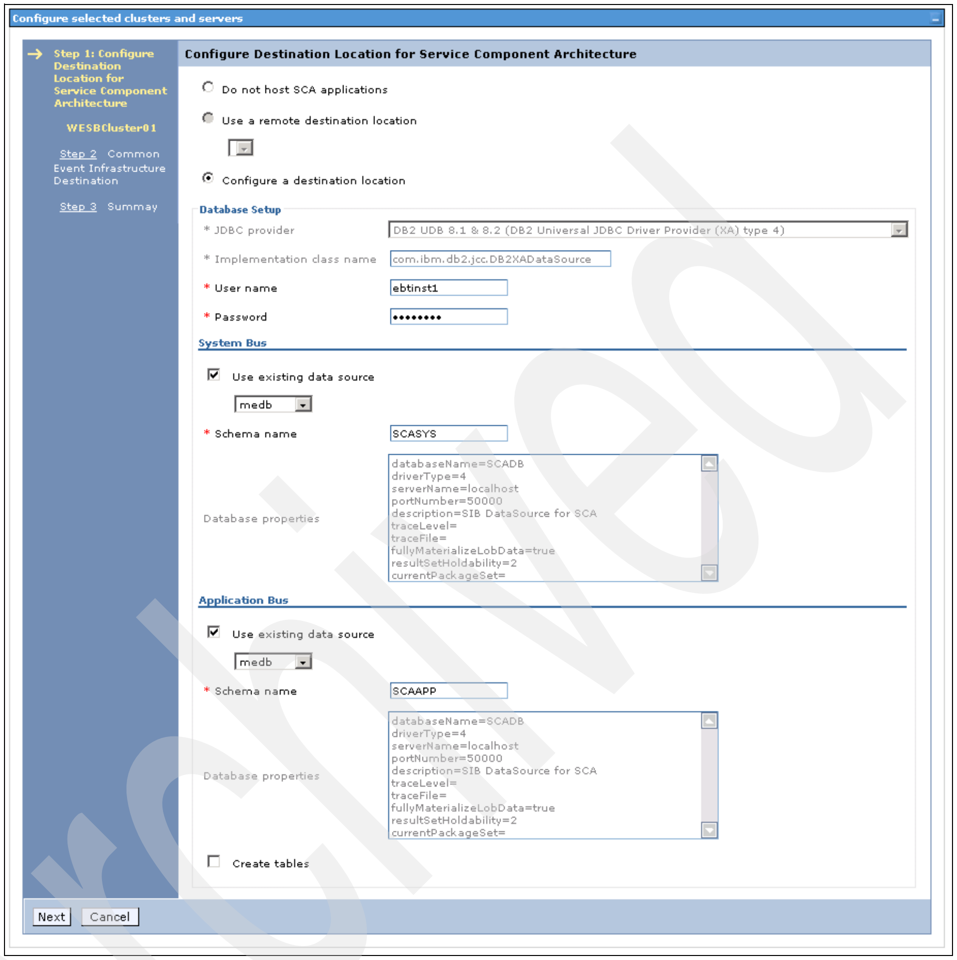
Previous | Next | Cancel

Figure 6-35 JDBC provider selection

7. Under System Bus, **select Use existing data source**. In the drop-down menu below, select the data source **medb**.
8. Set the schema name to `SCASYS`.
9. Under Application Bus, select **Use existing data source**. In the drop-down menu below, select the data source **medb**.
10. Set the schema name to `SCAAPP`.
11. Clear **Create tables** because you have already created the tables manually with the `sibDDLgenerator.bat` command.

   Note: If the database is DB2, this option creates the SCA database schemas the first time that you start the SCA messaging engines.

Figure 6-36 shows these settings.



Configure selected clusters and servers

→ Step 1: Configure Destination Location for Service Component Architecture

WESBCluster01

Step 2 Common Event Infrastructure Destination

Step 3 Summary

Configure Destination Location for Service Component Architecture

○ Do not host SCA applications

○ Use a remote destination location

● Configure a destination location

Database Setup

* JDBC provider	DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC Driver Provider (XA) type 4)
* Implementation class name	com.ibm.db2.jcc.DB2XADataSource
* User name	ebtinst1
* Password	••••••••

System Bus

[✓] Use existing data source

medb

* Schema name	SCASYS

Database properties

```
databaseName=SCADB
driverType=4
serverName=localhost
portNumber=50000
description=SIB DataSource for SCA
traceLevel=
traceFile=
fullyMaterializeLobData=true
resultSetHoldability=2
currentPackageSet=
```

Application Bus

[✓] Use existing data source

medb

* Schema name	SCAAPP

Database properties

```
databaseName=SCADB
driverType=4
serverName=localhost
portNumber=50000
description=SIB DataSource for SCA
traceLevel=
traceFile=
fullyMaterializeLobData=true
resultSetHoldability=2
currentPackageSet=
```

[ ] Create tables

Next Cancel

Figure 6-36 SCA Values

12. Click **Next**. Clear **Enable service at server startup** because this topology is not CEI enabled.
13. Click **Next** and **Finish** on the summary page.
14. Save the changes to the master configuration and then save and synchronize these changes. Click **OK** when this has completed.
15. Click **Finish**.

You have now constructed the topology that is shown in Figure 6-37.

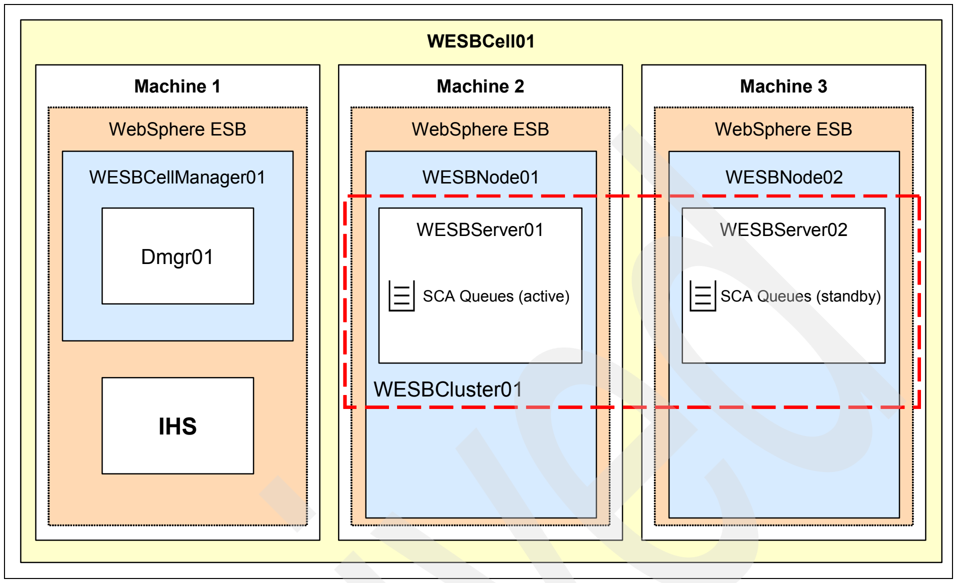


Figure 6-37 Cluster with SCA queues

6.3.3 Verifying the topology

In this section, you verify the database schema and start the cluster.

Checking the database schema

In this example, DB2 is the RDBMS. To check the database schema, follow these steps:

1. Log in to the database server as the instance user. In this example this is `ebtinst1`.
2. Run the following commands:

```
db2 connect to MEDB
db2 select tabname from syscat.tables where tabschema=\'SCAAPP\'
db2 select tabname from syscat.tables where tabschema=\'SCASYS\'
```

Note: On Windows platforms do not use the backslashes (\) with the **`db2 select`** command.

Each of the schemas has eight tables. Figure 6-38 shows example output for the SCAAPP schema.

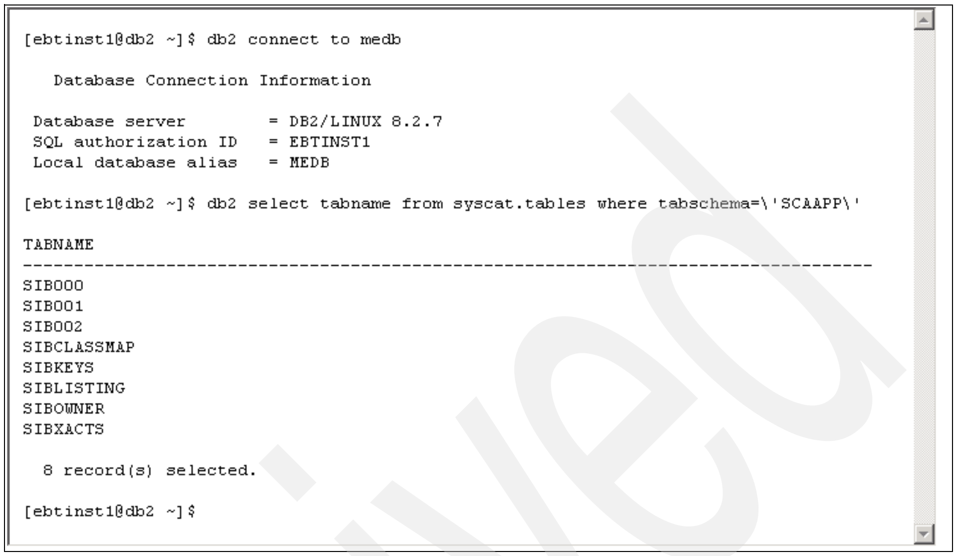

```
[ebtinst1@db2 ~]$ db2 connect to medb

   Database Connection Information

 Database server        = DB2/LINUX 8.2.7
 SQL authorization ID   = EBTINST1
 Local database alias   = MEDB

[ebtinst1@db2 ~]$ db2 select tabname from syscat.tables where tabschema=\'SCAAPP\'

TABNAME
-----------------------------------------------------------------------------------
SIB000
SIB001
SIB002
SIBCLASSMAP
SIBKEYS
SIBLISTING
SIBOWNER
SIBXACTS

  8 record(s) selected.

[ebtinst1@db2 ~]$
```

Figure 6-38 SCAAPP schema

Starting the cluster

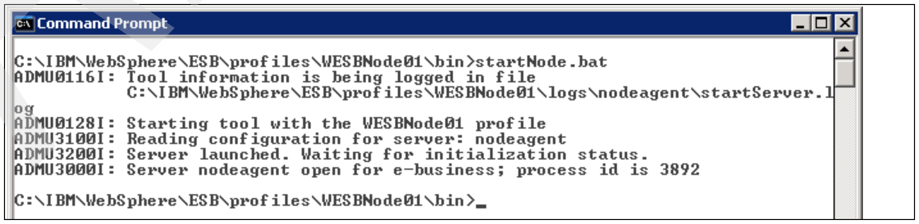
To start the cluster, follow these steps:

1. Ensure that all nodes are running. If a node is not running, start it with the following commands:
   a. Open a command prompt on the node's host machine.
   b. Change directory to the node's bin directory:

      `cd\IBM\WebSphere\ESB\profiles\WESBNode01\bin`

   c. Start the node using this command (Figure 6-39):

      `startNode.bat`



```
C:\IBM\WebSphere\ESB\profiles\WESBNode01\bin>startNode.bat
ADMU0116I: Tool information is being logged in file
           C:\IBM\WebSphere\ESB\profiles\WESBNode01\logs\nodeagent\startServer.l
og
ADMU0128I: Starting tool with the WESBNode01 profile
ADMU3100I: Reading configuration for server: nodeagent
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server nodeagent open for e-business; process id is 3892

C:\IBM\WebSphere\ESB\profiles\WESBNode01\bin>_
```

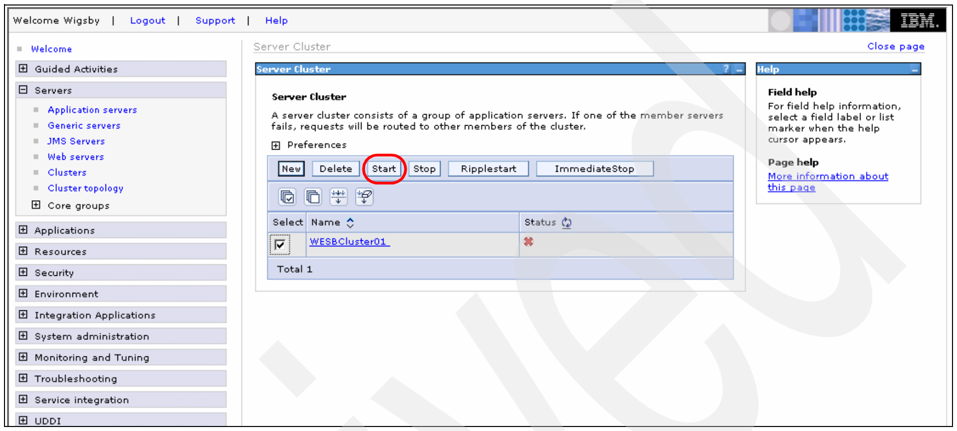
Figure 6-39 Starting a node at the command line

2. Start the cluster:

   a. Open the Administrative Console and login.

   b. Navigate to **Servers** → **Clusters**.

   c. Select **WESBCluster01** and click **Start**, as shown in Figure 6-40.



Server Cluster

A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster.

Preferences

Select	Name	Status
✓	WESBCluster01	✖
Total 1		

Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

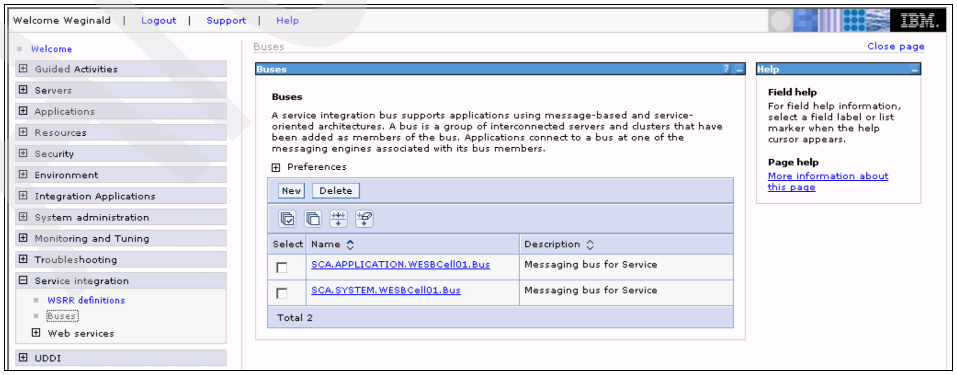
Figure 6-40 Starting the WESBCluster01 cluster

   d. Verify that the cluster starts successfully. The red cross should change to a solid green arrow. You need to refresh the server clusters view to see the change in status.

3. Verify that the messaging engines have started:

   a. Navigate to **Service integration** → **Buses**.

   There should be two buses listed—one for SCA Application and one for SCA System—as shown in Figure 6-41.



Buses

A service integration bus supports applications using message-based and service-oriented architectures. A bus is a group of interconnected servers and clusters that have been added as members of the bus. Applications connect to a bus at one of the messaging engines associated with its bus members.

Preferences

Select	Name	Description
☐	SCA.APPLICATION.WESBCell01.Bus	Messaging bus for Service
☐	SCA.SYSTEM.WESBCell01.Bus	Messaging bus for Service
Total 2		

Field help
For field help information, select a field label or list marker when the help cursor appears.

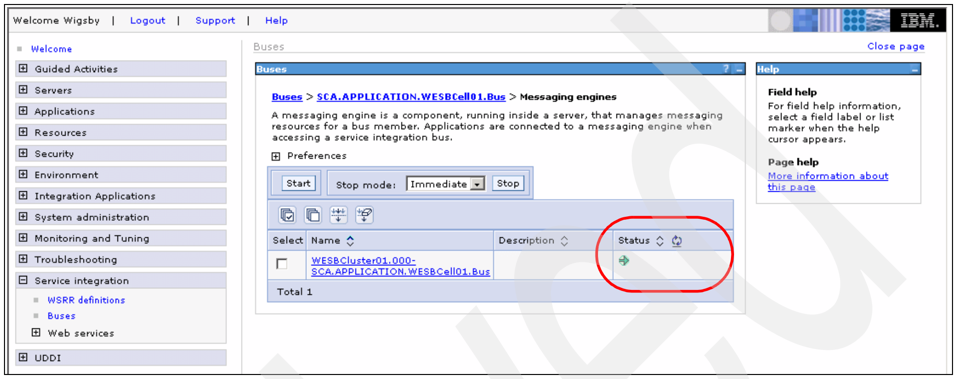
Page help
More information about this page

Figure 6-41 Service integration buses

b. Click **SCA.APPLICATION.WESBCell01.Bus**.

c. Under the Topology section, click **Messaging engines**.

d. The status of the `WESBCluster01.000-SCA.APPLICATION.WESBCell01.Bus` messaging engine should be started, as shown in Figure 6-42.



Buses > SCA.APPLICATION.WESBCell01.Bus > Messaging engines

A messaging engine is a component, running inside a server, that manages messaging resources for a bus member. Applications are connected to a messaging engine when accessing a service integration bus.

Select	Name	Description	Status
☐	WESBCluster01.000-SCA.APPLICATION.WESBCell01.Bus		➔

Total 1

Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

Figure 6-42 Messaging engine in started state

e. Repeat this process for the `SCA.SYSTEM.WESBCell01.Bus`.

6.4 Installing and testing the scenario

This section describes how to install and test the topology that you have built by deploying presentation logic, business module, and enterprise service layers.

6.4.1 Installing the enterprise applications

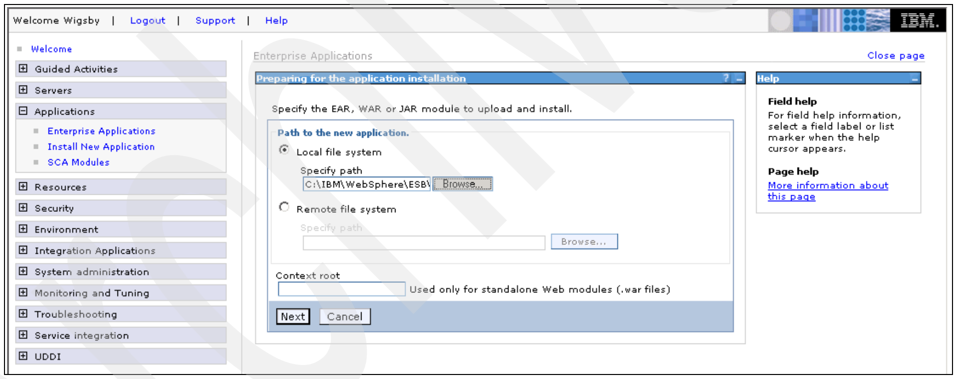
To test the business mediation logic for this topology, you need to create a presentation logic layer and a remote enterprise service layer. The presentation logic layer provides the client for the business module, and the remote enterprise service layer provides the services that are mediated.

In this example the presentation logic and remote enterprise service layers are provided by two stand-alone WebSphere Application Server instances. The presentation logic enterprise application, ITSOESBSmplPrsApp.ear, is installed on `ebt3` to `WASNode01` on `server1`. The enterprise service enterprise application, ITSOESBSmplSrvApp.ear, is installed on `ebt3` to `WASNode02` on `server1`.

Deploying the mediation application

The application supporting the business mediation logic is provided by a single EAR file. You must install this application to the cluster `WESBCluster01`. Follow these steps:

1. Open the `ebt0` Administrative Console and log in.
2. Navigate to **Applications** → **Install New Application**.
3. You install the enterprise application ITSOESBSmplBusApp.ear, which includes the mediation logic. This file is included in the additional material that is supplied with this book. To learn how to obtain it, see Appendix A, “Additional material” on page 427. Copy ITSOESBSmplBusApp.ear from the additional material to c:\IBM\WebSphere\ESB\installableApps.
4. In the Administrative Console, select **Local file system** and click **Browse**.
5. Navigate to the c:\IBM\WebSphere\ESB\installableApps folder.
6. Select the ESB mediation module to be installed, in this example ITSOESBSmplBusApp.ear, and click **Open**.
7. From the Administrative Console, click **Next**, as shown in Figure 6-43.



Specify the EAR, WAR or JAR module to upload and install.

Path to the new application.

● Local file system

Specify path: C:\IBM\WebSphere\ESB\

○ Remote file system

Specify path

Context root: Used only for standalone Web modules (.war files)

Field help

For field help information, select a field label or list marker when the help cursor appears.

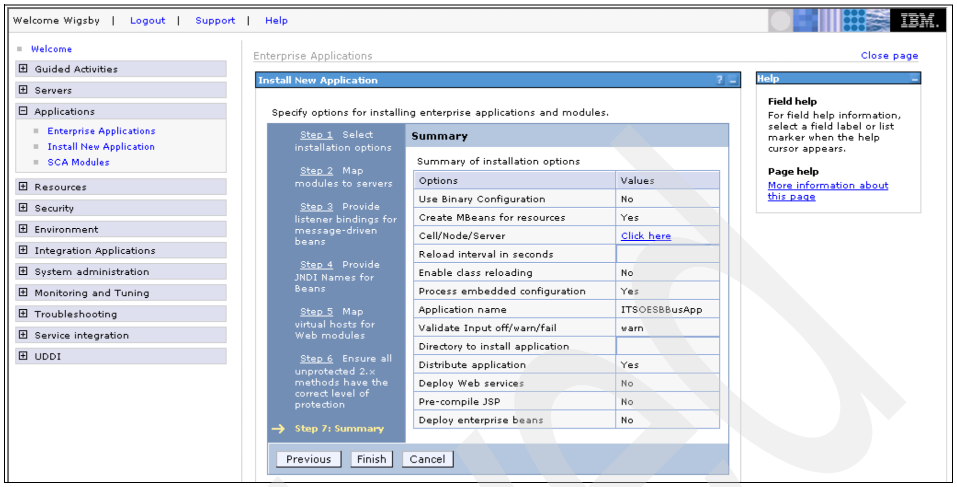
Page help

More information about this page

Figure 6-43 Installing the business module

8. In the next panel, leave the settings at their default, and click **Next**. Leave all the application deployment settings as the default.
9. Click the link to the summary panel, in this example **Step 7: Summary**.

10. Click **Finish**, as shown in Figure 6-44.


Welcome Wigsby | Logout | Support | Help
IBM.
Welcome
Guided Activities
Servers
Applications
Enterprise Applications
Install New Application
SCA Modules
Resources
Security
Environment
Integration Applications
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Enterprise Applications
Close page
Install New Application
Help
Field help
For field help information, select a field label or list marker when the help cursor appears.
Page help
More information about this page
Step 1 Select installation options
Step 2 Map modules to servers
Step 3 Provide listener bindings for message-driven beans
Step 4 Provide JNDI Names for Beans
Step 5 Map virtual hosts for Web modules
Step 6 Ensure all unprotected 2.x methods have the correct level of protection
Step 7: Summary
Previous
Finish
Cancel

Specify options for installing enterprise applications and modules.

Summary

Summary of installation options

Options	Values
Use Binary Configuration	No
Create MBeans for resources	Yes
Cell/Node/Server	Click here
Reload interval in seconds	
Enable class reloading	No
Process embedded configuration	Yes
Application name	ITSOESBBusApp
Validate Input off/warn/fail	warn
Directory to install application	
Distribute application	Yes
Deploy Web services	No
Pre-compile JSP	No
Deploy enterprise beans	No

Figure 6-44 Business module installation summary

11. After the application is shown as having been installed successfully, click **Save to Master Configuration**.

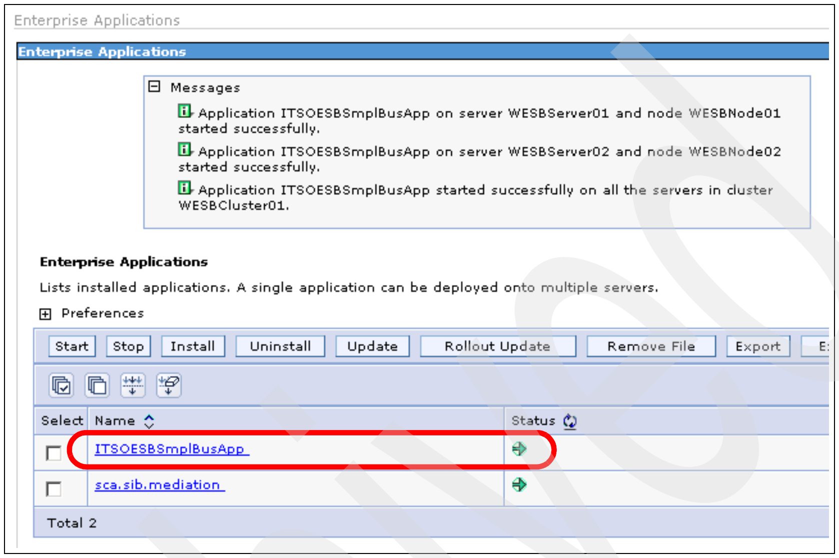
12. Select **Synchronize changes with Nodes**, and click **Save**.

13. When completed successfully, click **OK**.

14. Start the application (Figure 6-45):

   a. Navigate to **Applications** → **Enterprise Applications**.

   b. Select **ITSOESBSmplBusApp** and click **Start**.


Enterprise Applications

Enterprise Applications

Messages

Application ITSOESBSmplBusApp on server WESBServer01 and node WESBNode01 started successfully.

Application ITSOESBSmplBusApp on server WESBServer02 and node WESBNode02 started successfully.

Application ITSOESBSmplBusApp started successfully on all the servers in cluster WESBCluster01.

Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

Preferences

Select	Name	Status
☐	ITSOESBSmplBusApp	➔
☐	sca.sib.mediation	➔
Total 2		

Figure 6-45 Installed enterprise application ITSOESBSmplBusApp

Preparing the presentation logic and enterprise service layers

You need to define a WebSphere Application Server server to host the presentation logic layer and then install the presentation logic application. Follow these steps:

1. Create a new WebSphere Application Server profile using the profile creation wizard on `ebt3`. On Windows platforms, launch the Profile Wizard by clicking **Start** → **All Programs** → **IBM WebSphere** → **Application Server Network Deployment V6** → **Profile creation wizard**.

2. Define a profile with the following properties:

   – Profile type: `Application Server profile`
   – Profile name: `WASNode01`
   – Node name: `WASNode01`
   – Host name: `ebt3.itso`

3. When the profile is created, navigate to the bin directory of the profile (which on our system was C:\IBM\WebSphere\ESB\profiles\WASNode01\bin) and start the server using the command:

```
startServer server1
```

4. When the server is started, launch the Administrative Console:

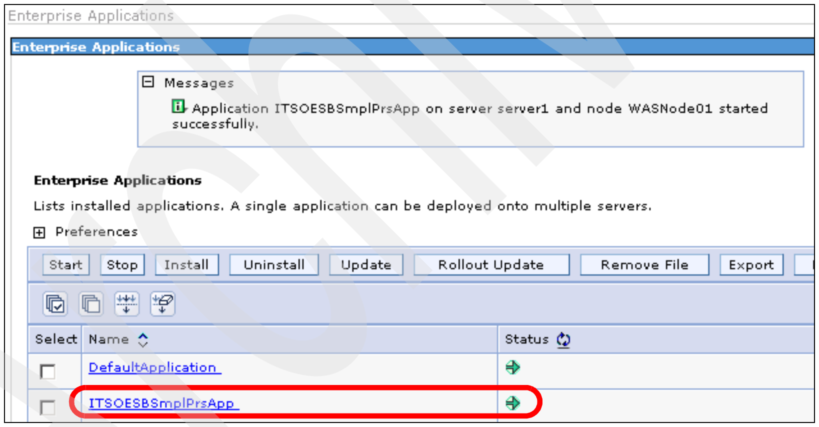
```
http://ebt3.itso:9060/ibm/console
```

5. Install the presentation logic enterprise application ITSOESBSmplPrsApp.ear. This file is located in the additional material that is supplied with this book. To learn how to obtain it, see Appendix A, "Additional material" on page 427. To install the application, follow these steps:

   a. Copy ITSOESBSmplPrsApp.ear from the additional material that is supplied with this book to c:\IBM\WebSphere\ESB\installableApps.

   b. In the Administrative Console, click **Applications** → **Install New Application** and install ITSOESBSmplPrsApp.ear. When the application is installed, save the configuration.

   c. Click **Applications** → **Enterprise Applications and start ITSOESBSmplPrsApp** (Figure 6-46).



Application ITSOESBSmplPrsApp on server server1 and node WASNode01 started successfully.

Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

Select	Name	Status
☐	DefaultApplication	➔
☐	ITSOESBSmplPrsApp	➔

Figure 6-46 Installed enterprise application ITSOESBSmplPrsApp

You need to now define a WebSphere Application Server server to host the enterprise service layer and then install the enterprise service application. Follow these steps:

1. Create a new WebSphere Application Server profile using the profile creation wizard on `ebt3`. On Windows platforms, launch the Profile Wizard by clicking **Start** → **All Programs** → **IBM WebSphere** → **Application Server Network Deployment V6** → **Profile creation wizard**.

2. Define a profile with the following properties:
   - Profile type: `Application Server profile`
   - Profile name: `WASNode02`
   - Node name: `WASNode02`
   - Host name: `ebt3.itso`
3. When the profile is created, navigate to the bin directory of the profile (which on our system was C:\IBM\WebSphere\ESB\profiles\WASNode02\bin) and start the server using the command:

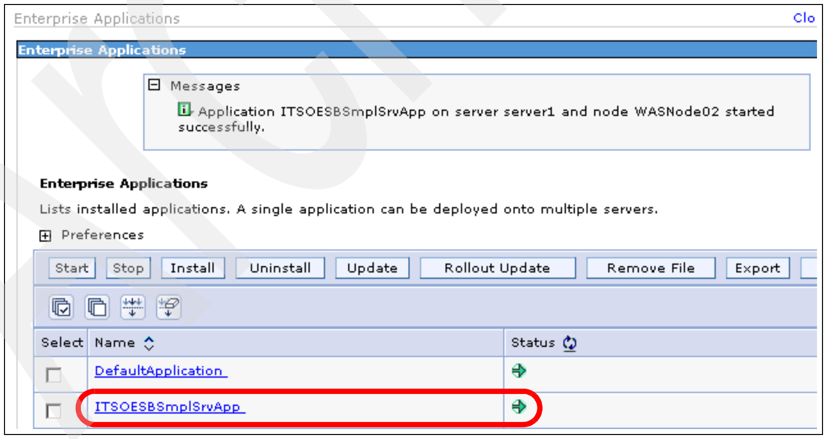
   ```
   startServer server1
   ```
4. When the server is started, launch the Administrative Console:

   ```
   http://ebt3.itso:9061/ibm/console
   ```
5. Install the presentation logic enterprise application ITSOESBSmplSrvApp.ear. This file is located in the additional material that is supplied with this book. To learn how to obtain it, see Appendix A, "Additional material" on page 427. Then, follow these steps:

   a. Copy ITSOESBSmplSrvApp.ear from the additional material that is supplied with this book to c:\IBM\WebSphere\ESB\installableApps.

   b. In the Administrative Console, click **Applications** → **Install New Application** and install ITSOESBSmplSrvApp.ear. When the application is installed, save the configuration.

   c. Click **Applications** → **Enterprise Applications and start ITSOESBSmplSrvApp** (Figure 6-47).



Application ITSOESBSmplSrvApp on server server1 and node WASNode02 started successfully.

Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

Select	Name	Status
☐	DefaultApplication	➔
☐	ITSOESBSmplSrvApp	➔

Figure 6-47 Installed enterprise application ITSOESBSmplSrvApp

6.4.2 Configuring IBM HTTP Server

You need to install and configure a router in front of the mediation cluster so that all members of the cluster can be accessed by the presentation logic layer.

Client requests from the browser are made to the presentation layer that runs in a stand-alone WebSphere Application Server. The JSP running on this stand-alone application server cannot make requests to multiple destinations hosting the Web service. The IBM HTTP Server and WebSphere Plugin work together to provide a single point of access for the JSP, routing requests from the presentation logic layer to the mediation cluster members, as illustrated in Figure 6-48. Mediation then takes place, and a Web services call is made to the appropriate target. The response is directed back through the original cluster member.

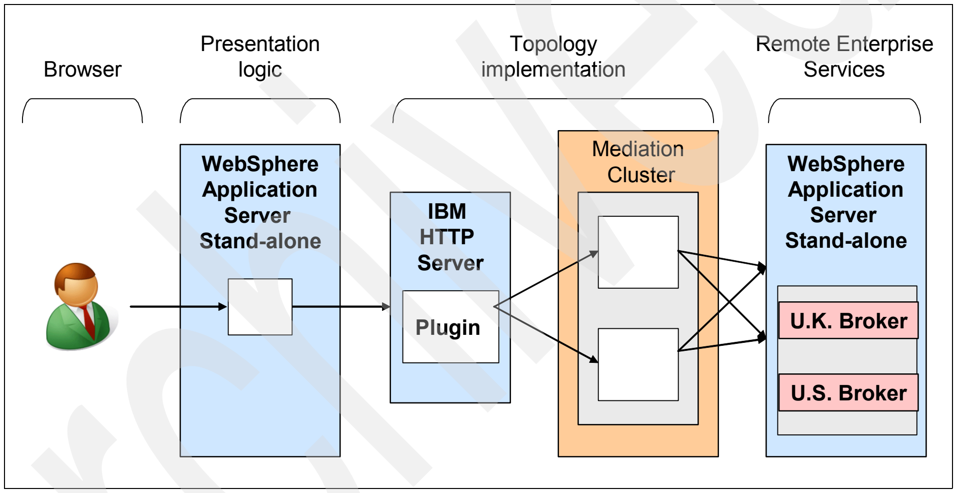


Figure 6-48 Load-balanced and failover-capable mediation cluster

The following products must already be installed:

- IBM HTTP Server in c:\IBMIHS
- The WebSphere plugin for IBM HTTP Server in c:\IBM\WebSphere\Plugins

For instructions on how to install these products, refer to Appendix B, “Product installation” on page 429.

Adding the Web server to the cell

To add the Web server to the cell, follow these steps:

1. Open a command prompt on the machine where the Web server plugin is installed.

2. Change to the c:\IBM\WebSphere\Plugins\bin directory.

   Note: In this example, the Web server is named `webserver1`. Therefore, the configuration script is named configurewebserver1.bat.

3. Copy the configurewebserver1.bat file to %*WAS_HOME*%\bin on any node in the cell.
4. Change to the %*WAS_HOME*%\bin directory and run configurewebserver1.bat.

This script adds the Web server into the cell and deploys the applications to the Web server.

Configuring the plug-in

To configure the plug-in, follow these steps:

1. Open the Administrative Console on `ebt0` and login.
2. Navigate to **Servers** → **Web servers**.
3. Select **webserver1** and click **Generate Plug-in**.
4. When complete, the message text identifies the location of the generated plug-in file. In this example, it is:

   c:\IBM\WebSphere\ESB\profiles\Dmgr01\config\cells\WESBCell01\nodes\ebt0.itso\servers\webserver1\plugin-cfg.xml

5. Move the plugin-cfg.xml configuration file.

   The location of plugin-cfg.xml that the WebSphere Plug-in uses is specified by the `WebSpherePluginConfig` variable within the httpd.conf configuration file of IBM HTTP Server. Therefore, you must move the generated plugin-cfg.xml to the path that is defined by this variable. In this example, the command is:

   ```
   move
   c:\IBM\WebSphere\ESB\profiles\Dmgr01\config\cells\WESBCell01\nodes\e
   bt0.itso\servers\webserver1\plugin-cfg.xml
   c:\IBM\WebSphere\Plugins\config\webserver1\plugin-cfg.xml
   ```

6. Restart IBM HTTP Server for the changes to take effect.

   ```
   cd c:\IBMIHS\bin
   Apache -k restart
   ```

Configuring the application deployment

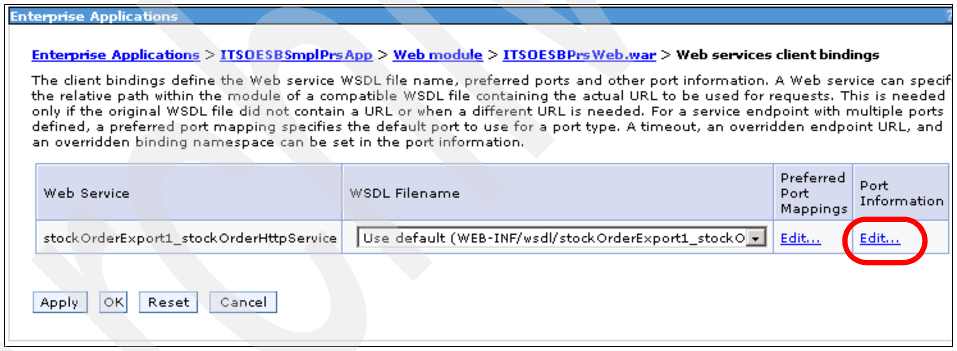
There are two additional steps to configure the application deployment:

1. Configuring the presentation application for the mediation export
2. Configuring the mediation application for the remote enterprise services

Presentation logic connectivity to mediation layer

To configure the presentation logic connectivity to the mediation layer:

1. Open the presentation logic layer's WebSphere Application Server Administrative Console and login (for example `http://ebt3:9060/ibm/console`).
2. Navigate to **Applications** → **Enterprise Applications**.
3. Click the presentation application, **ITSOESBSmplPrsApp**.
4. Under Related Items, click **Web modules**.
5. Click the Web Archive (WAR) file, **ITSOESBPrsWeb.war**.
6. Under Additional Properties, click **Web services client bindings**.
7. Under Port Information, click **Edit** (Figure 6-49).



Enterprise Applications > ITSOESBSmplPrsApp > Web module > ITSOESBPrsWeb.war > **Web services client bindings**

The client bindings define the Web service WSDL file name, preferred ports and other port information. A Web service can specif the relative path within the module of a compatible WSDL file containing the actual URL to be used for requests. This is needed only if the original WSDL file did not contain a URL or when a different URL is needed. For a service endpoint with multiple ports defined, a preferred port mapping specifies the default port to use for a port type. A timeout, an overridden endpoint URL, and an overridden binding namespace can be set in the port information.

Web Service	WSDL Filename	Preferred Port Mappings	Port Information
stockOrderExport1_stockOrderHttpService	Use default (WEB-INF/wsdl/stockOrderExport1_stockO	Edit...	Edit...

Figure 6-49 Editing port information for the presentation logic layer

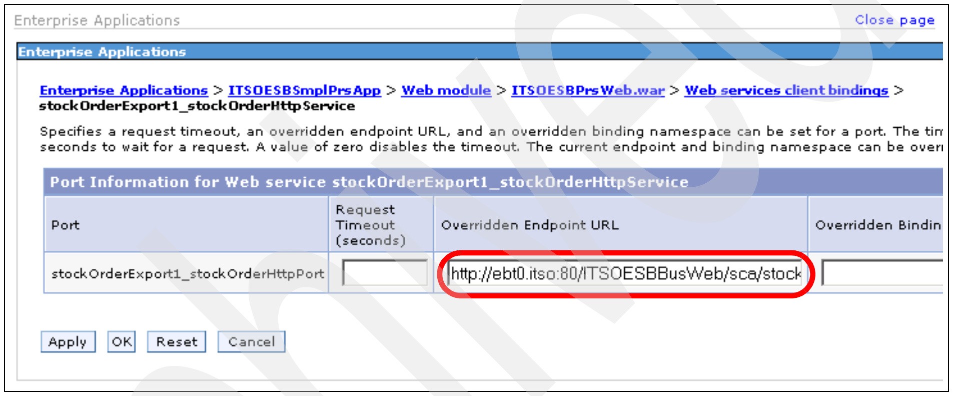
8. Change the Overridden Endpoint URL to that of the IBM HTTP Server, in this example:

   `http://ebt0.itso:80/ITSOESBBusWeb/sca/stockOrderExport`

Note: This Endpoint URL is constructed from:

- The host name of the IBM HTTP Server and the port on which it listens. In this example `ebt0.itso:80`.
- The context-root of the Web module that is defined in the deployment descriptor of the mediation application, `ITSOESBBusWeb`.
- The Web service URL-pattern that is defined in the deployment descriptor of the WAR file within the mediation application. In this example `sca/stockOrderExport`.

Figure 6-50 shows this step.



Enterprise Applications

Close page

Enterprise Applications

Enterprise Applications > ITSOESBSmplPrsApp > Web module > ITSOESBPrsWeb.war > Web services client bindings >
stockOrderExport1_stockOrderHttpService

Specifies a request timeout, an overridden endpoint URL, and an overridden binding namespace can be set for a port. The tir
seconds to wait for a request. A value of zero disables the timeout. The current endpoint and binding namespace can be overi

Port Information for Web service stockOrderExport1_stockOrderHttpService

Port	Request Timeout (seconds)	Overridden Endpoint URL	Overridden Bindin
stockOrderExport1_stockOrderHttpPort		http://ebt0.itso:80/ITSOESBBusWeb/sca/stock	

Apply OK Reset Cancel

Figure 6-50 Configuring the Endpoint URL

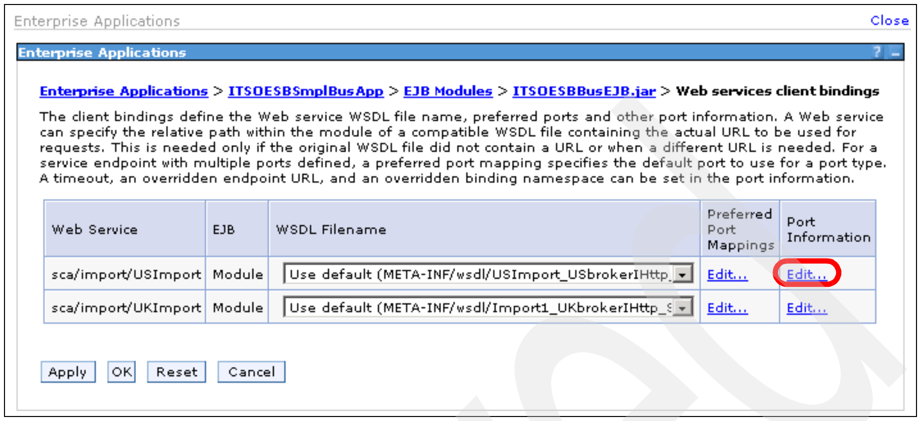
9. Click **OK**. Save the changes.

Mediation layer connectivity to remote enterprise services

To configure the mediation layer connectivity to the remote enterprise services:

1. Open the `WESBCell` deployment manager's Administrative Console and login.
2. Navigate to **Applications** → **Enterprise Applications**.
3. Click **ITSOESBSmplBusApp**.
4. Under Related Items, click **EJB Modules**.
5. Click the JAR file that is presented, **ITSOESBBusEJB.jar**.
6. Under Additional Properties, click **Web services client bindings**.

7. For the sca/import/USImport service, click **Edit** under **Port Information,** as shown in Figure 6-51.



Enterprise Applications | Close

Enterprise Applications

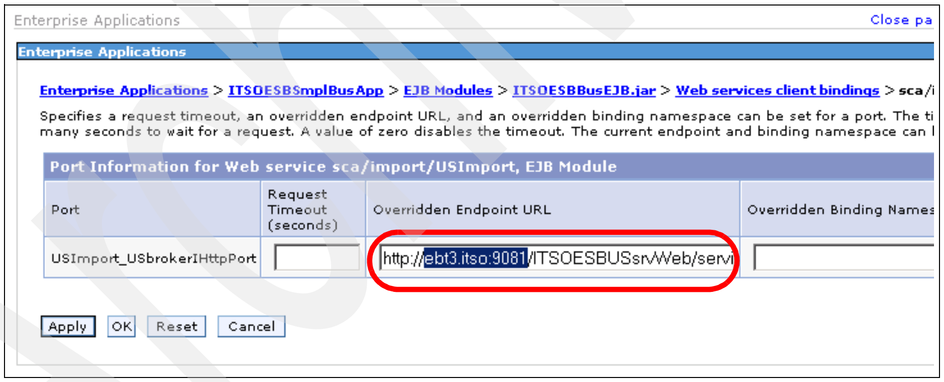
Enterprise Applications > ITSOESBSmplBusApp > EJB Modules > ITSOESBBusEJB.jar > **Web services client bindings**

The client bindings define the Web service WSDL file name, preferred ports and other port information. A Web service can specify the relative path within the module of a compatible WSDL file containing the actual URL to be used for requests. This is needed only if the original WSDL file did not contain a URL or when a different URL is needed. For a service endpoint with multiple ports defined, a preferred port mapping specifies the default port to use for a port type. A timeout, an overridden endpoint URL, and an overridden binding namespace can be set in the port information.

Web Service	EJB	WSDL Filename	Preferred Port Mappings	Port Information
sca/import/USImport	Module	Use default (META-INF/wsdl/USImport_USbrokerIHttp	Edit...	Edit...
sca/import/UKImport	Module	Use default (META-INF/wsdl/Import1_UKbrokerIHttp_S	Edit...	Edit...

Figure 6-51 Editing port information for the remote enterprise services

8. Change the host and port to that of the WebSphere Application Server that is hosting the remote enterprise services, in this example `http://ebt3.itso:9081/` (Figure 6-52). Click **OK**.



Enterprise Applications | Close pa

Enterprise Applications

Enterprise Applications > ITSOESBSmplBusApp > EJB Modules > ITSOESBBusEJB.jar > Web services client bindings > sca/i

Specifies a request timeout, an overridden endpoint URL, and an overridden binding namespace can be set for a port. The ti
many seconds to wait for a request. A value of zero disables the timeout. The current endpoint and binding namespace can l

Port Information for Web service sca/import/USImport, EJB Module

Port	Request Timeout (seconds)	Overridden Endpoint URL	Overridden Binding Names
USImport_USbrokerIHttpPort		http://ebt3.itso:9081/ITSOESBUSsrvWeb/servi	

Figure 6-52 Configuring the host for the remote enterprise services

9. Repeat the previous step for the `sca/import/UKImport` service.

10. Save and synchronize the changes, and click **OK** when complete.

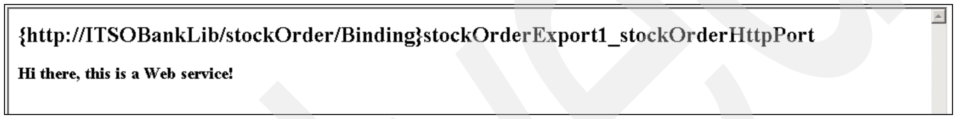
It is possible to test each of the remote enterprise services with a browser that is directed at the following URLs:

- `http://ebt3.itso:9081/ITSOESBUSsrvWeb/services/USbrokerIExport1_USbrokerIHttpPort`
- `http://ebt3.itso:9081/ITSOESBUKsrvWeb/services/UKbrokerIExport1_UKbrokerIHttpPort`

The mediation Web services can be reached directly on the following URLs:

- `http://ebt1.itso:9080/ITSOESBBusWeb/sca/stockOrderExport`
- `http://ebt2.itso:9080/ITSOESBBusWeb/sca/stockOrderExport`

Figure 6-53 shows an example response.

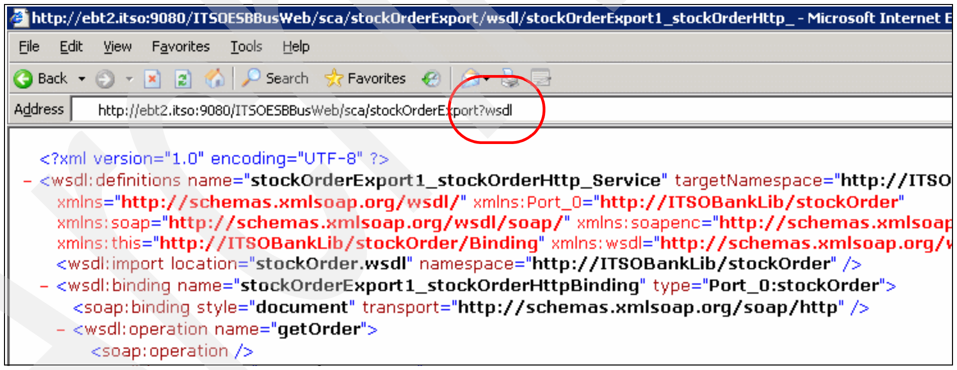


{http://ITSOBankLib/stockOrder/Binding}stockOrderExport1_stockOrderHttpPort

Hi there, this is a Web service!

Figure 6-53 Example response from Web service call

By appending `?wsdl` to these URLs, the actual WSDL definition can be displayed, as shown in Figure 6-54.



http://ebt2.itso:9080/ITSOESBBusWeb/sca/stockOrderExport/wsdl/stockOrderExport1_stockOrderHttp_ - Microsoft Internet E
File Edit View Favorites Tools Help
Back
Search
Favorites
Address

```
http://ebt2.itso:9080/ITSOESBBusWeb/sca/stockOrderExport?wsdl
```

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions name="stockOrderExport1_stockOrderHttp_Service" targetNamespace="http://ITSO
    xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:Port_0="http://ITSOBankLib/stockOrder"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:soapenc="http://schemas.xmlsoap
    xmlns:this="http://ITSOBankLib/stockOrder/Binding" xmlns:wsdl="http://schemas.xmlsoap.org/w
    <wsdl:import location="stockOrder.wsdl" namespace="http://ITSOBankLib/stockOrder" />
  - <wsdl:binding name="stockOrderExport1_stockOrderHttpBinding" type="Port_0:stockOrder">
      <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
    - <wsdl:operation name="getOrder">
        <soap:operation />
```

Figure 6-54 Requesting the WSDL definition

6.4.3 Testing the scenario

This section tests the environment and the application that it hosts and shows both mediation and load-balancing that occurs.

Starting the environment

To start the environment, follow these steps:

1. Ensure that all three layers of the environment (presentation, mediation, and remote enterprise services) are started.
2. Ensure that all applications are running in these layers.
3. Start up IBM HTTP Server with the following commands:

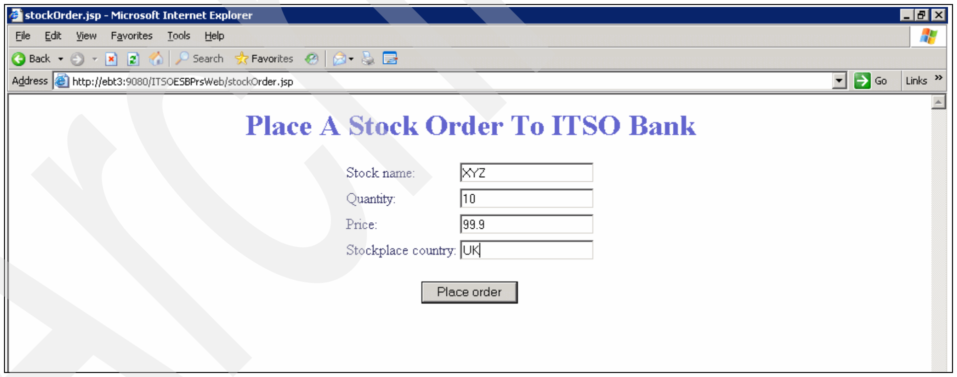
```
cd c:\IBMIHS\bin
Apache.exe -k start
```

Testing the mediation

Testing the application involves accessing the JSP front end and sending in a request:

1. Open a Web browser and launch the presentation logic JSP (Figure 6-55). We used:

```
http://ebt3.itso:9080/ITSOESBPrsWeb/stockOrder.jsp
```


stockOrder.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://ebt3:9080/ITSOESBPrsWeb/stockOrder.jsp
Go Links

Place A Stock Order To ITSO Bank

Stock name:	XYZ
Quantity:	10
Price:	99.9
Stockplace country:	UK

Figure 6-55 Accessing the presentation layer JSP

2. Complete the fields, making sure to specify a Stockplace country value of `UK`. Click **Place order**. The mediation logic places the order with the U. K. broker service and returns a confirmation ID as shown in Figure 6-56.

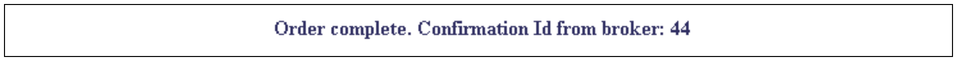

Order complete. Confirmation Id from broker: 44

Figure 6-56 Sample response from U. K. broker

3. Enter a second request, this time specifying a Stockplace country value of `USA`. Click **Place order**. The mediation logic places the order with the USA broker service and returns a confirmation ID as shown in Figure 6-57.

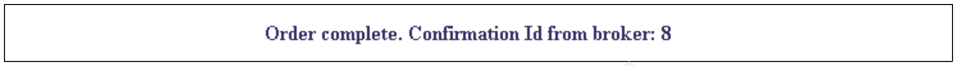

Order complete. Confirmation Id from broker: 8

Figure 6-57 Sample response from U. S. broker

This example shows mediation occurring based on the Stockplace country data that is provided by the JSP in the presentation layer. In one case, the `UKbrokerIExport1_UKbrokerIHttpPort` remote service is called, and in the other case the `USbrokerIExport1_USbrokerIHttpPort` remote service is called.

Testing load-balancing

By default, the WebSphere Plug-in passes requests made to the HTTP server onto the mediation cluster, using the round-robin balancing algorithm.

By changing the WebSphere Plug-in logging level defined in the plugin-cfg.xml file, it is possible to view which mediation cluster member is selected as each request is made. Follow these steps:

1. Open the plugin-cfg.xml file using an appropriate text editor, which in this case is located at c:\IBM\WebSphere\Plugins\config\webserver1\plugin-cfg.xml.
2. In the first `Log` stanza, change the `LogLevel` parameter to `Stats`, as shown in Figure 6-58.

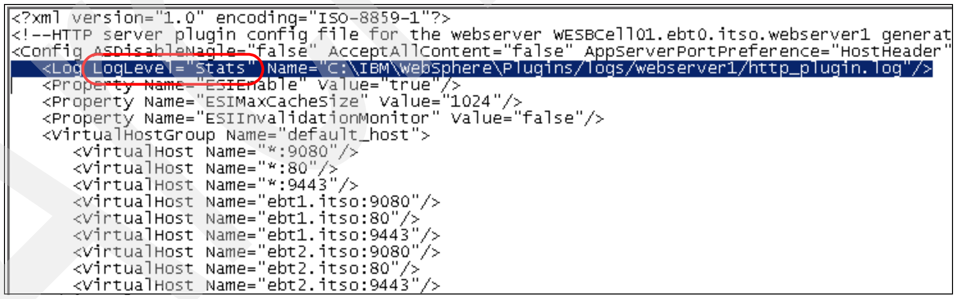

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--HTTP server plugin config file for the webserver WESBCell01.ebt0.itso.webserver1 generat
<Config ASDisableNagle="false" AcceptAllContent="false" AppServerPortPreference="HostHeader"
   <Log LogLevel="Stats" Name="C:\IBM\WebSphere\Plugins/logs/webserver1/http_plugin.log"/>
   <Property Name="ESIEnable" Value="true"/>
   <Property Name="ESIMaxCacheSize" Value="1024"/>
   <Property Name="ESIInvalidationMonitor" Value="false"/>
   <VirtualHostGroup Name="default_host">
      <VirtualHost Name="*:9080"/>
      <VirtualHost Name="*:80"/>
      <VirtualHost Name="*:9443"/>
      <VirtualHost Name="ebt1.itso:9080"/>
      <VirtualHost Name="ebt1.itso:80"/>
      <VirtualHost Name="ebt1.itso:9443"/>
      <VirtualHost Name="ebt2.itso:9080"/>
      <VirtualHost Name="ebt2.itso:80"/>
      <VirtualHost Name="ebt2.itso:9443"/>
```

Figure 6-58 Changing the logging level for the WebSphere Plug-in

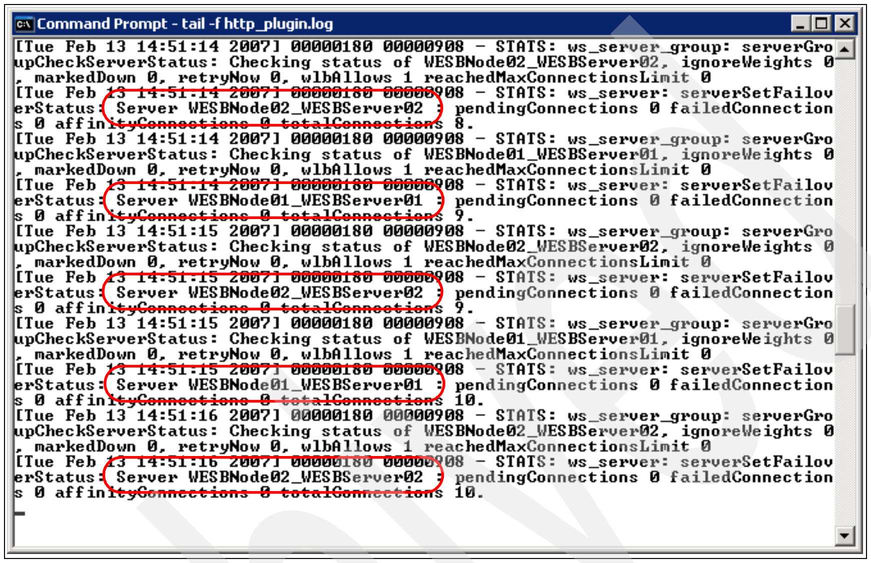
3. Save the plugin-cfg.xml file and restart the HTTP server:

```
cd c:\IBMIHS\bin
Apache -k restart
```

4. Submit a series of requests to the presentation layer JSP, as described in “Testing the mediation” on page 148.

5. Open the WebSphere Plug-in log file in an appropriate text editor, in this case c:\IBM\WebSphere\Plugins\logs\webserver1\http_plugin.log.

6. A series of lines shows each request that is passed to members of the mediation cluster in turn, as shown in Figure 6-59.



```
[Tue Feb 13 14:51:14 2007] 00000180 00000908 - STATS: ws_server_group: serverGro
upCheckServerStatus: Checking status of WESBNode02_WESBServer02, ignoreWeights 0
, markedDown 0, retryNow 0, wlbAllows 1 reachedMaxConnectionsLimit 0
[Tue Feb 13 14:51:14 2007] 00000180 00000908 - STATS: ws_server: serverSetFailov
erStatus: Server WESBNode02_WESBServer02 : pendingConnections 0 failedConnection
s 0 affinityConnections 0 totalConnections 8.
[Tue Feb 13 14:51:14 2007] 00000180 00000908 - STATS: ws_server_group: serverGro
upCheckServerStatus: Checking status of WESBNode01_WESBServer01, ignoreWeights 0
, markedDown 0, retryNow 0, wlbAllows 1 reachedMaxConnectionsLimit 0
[Tue Feb 13 14:51:14 2007] 00000180 00000908 - STATS: ws_server: serverSetFailov
erStatus: Server WESBNode01_WESBServer01 : pendingConnections 0 failedConnection
s 0 affinityConnections 0 totalConnections 9.
[Tue Feb 13 14:51:15 2007] 00000180 00000908 - STATS: ws_server_group: serverGro
upCheckServerStatus: Checking status of WESBNode02_WESBServer02, ignoreWeights 0
, markedDown 0, retryNow 0, wlbAllows 1 reachedMaxConnectionsLimit 0
[Tue Feb 13 14:51:15 2007] 00000180 00000908 - STATS: ws_server: serverSetFailov
erStatus: Server WESBNode02_WESBServer02 : pendingConnections 0 failedConnection
s 0 affinityConnections 0 totalConnections 9.
[Tue Feb 13 14:51:15 2007] 00000180 00000908 - STATS: ws_server_group: serverGro
upCheckServerStatus: Checking status of WESBNode01_WESBServer01, ignoreWeights 0
, markedDown 0, retryNow 0, wlbAllows 1 reachedMaxConnectionsLimit 0
[Tue Feb 13 14:51:15 2007] 00000180 00000908 - STATS: ws_server: serverSetFailov
erStatus: Server WESBNode01_WESBServer01 : pendingConnections 0 failedConnection
s 0 affinityConnections 0 totalConnections 10.
[Tue Feb 13 14:51:16 2007] 00000180 00000908 - STATS: ws_server_group: serverGro
upCheckServerStatus: Checking status of WESBNode02_WESBServer02, ignoreWeights 0
, markedDown 0, retryNow 0, wlbAllows 1 reachedMaxConnectionsLimit 0
[Tue Feb 13 14:51:16 2007] 00000180 00000908 - STATS: ws_server: serverSetFailov
erStatus: Server WESBNode02_WESBServer02 : pendingConnections 0 failedConnection
s 0 affinityConnections 0 totalConnections 10.
_
```

Figure 6-59 Load-balanced requests

7

Implementing the Full Support topology for WebSphere ESB

This chapter details the steps to implement the *Full Support topology* for WebSphere ESB and to deploy a sample application to this topology. We discuss the following topics:

- Selecting and implementing the Full Support topology
- Generic steps for creating WebSphere ESB clusters
- Specific configuration steps for the Full Support topology
- Installing and testing the scenario

For simplicity, this chapter does not implement any security. To be considered a production topology, the addition of security is essential. Chapter 9, "Implementing the Full Support topology for WebSphere Process Server" on page 319 shows how to add security to a production topology.

7.1 Selecting and implementing the Full Support topology

This section includes the following sections:

- Scenario requirements

  Summarizes the fictitious business scenario that we deploy to the Full Support topology we build in this chapter.
- Selecting the appropriate topology

  Demonstrates why we select the Full Support topology for WebSphere ESB to host this business scenario.
- Overview of how to implement the Full Support topology

  Describes the steps that are necessary to build the Full Support topology for WebSphere ESB.

Note: For an overview of the Full Support topology, refer to 4.4, “Full Support topology” on page 53.

7.1.1 Scenario requirements

You must build a production topology to host a WebSphere ESB mediation flow that is built for the fictitious organization ITSO Bank. The mediation flow (shown in Figure 7-1) performs the following functions:

1. Bank customers place stock orders through a traditional trading Web application.
2. The Web application sends a synchronous call to the Stock Purchase mediation flow running in WebSphere ESB.
3. The mediation flow determines by which of two stock brokers the stock order should be fulfilled—either a U. S. broker or a U. K. broker.
4. Under certain conditions, a Common Business Event (CBE) event is logged to CEI before the broker is invoked.
5. A Web service request is sent to the appropriate broker, and a response is returned to the trading Web application (through the mediation response flow).

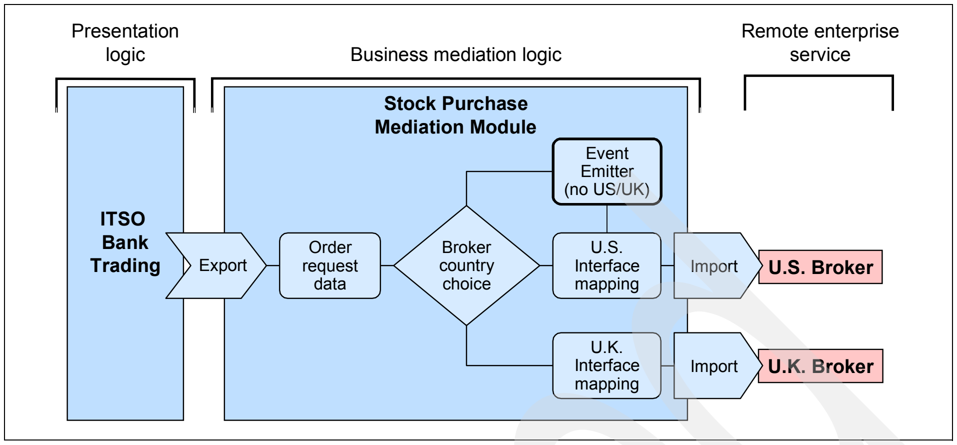


Figure 7-1 Complex mediation scenario for WebSphere ESB

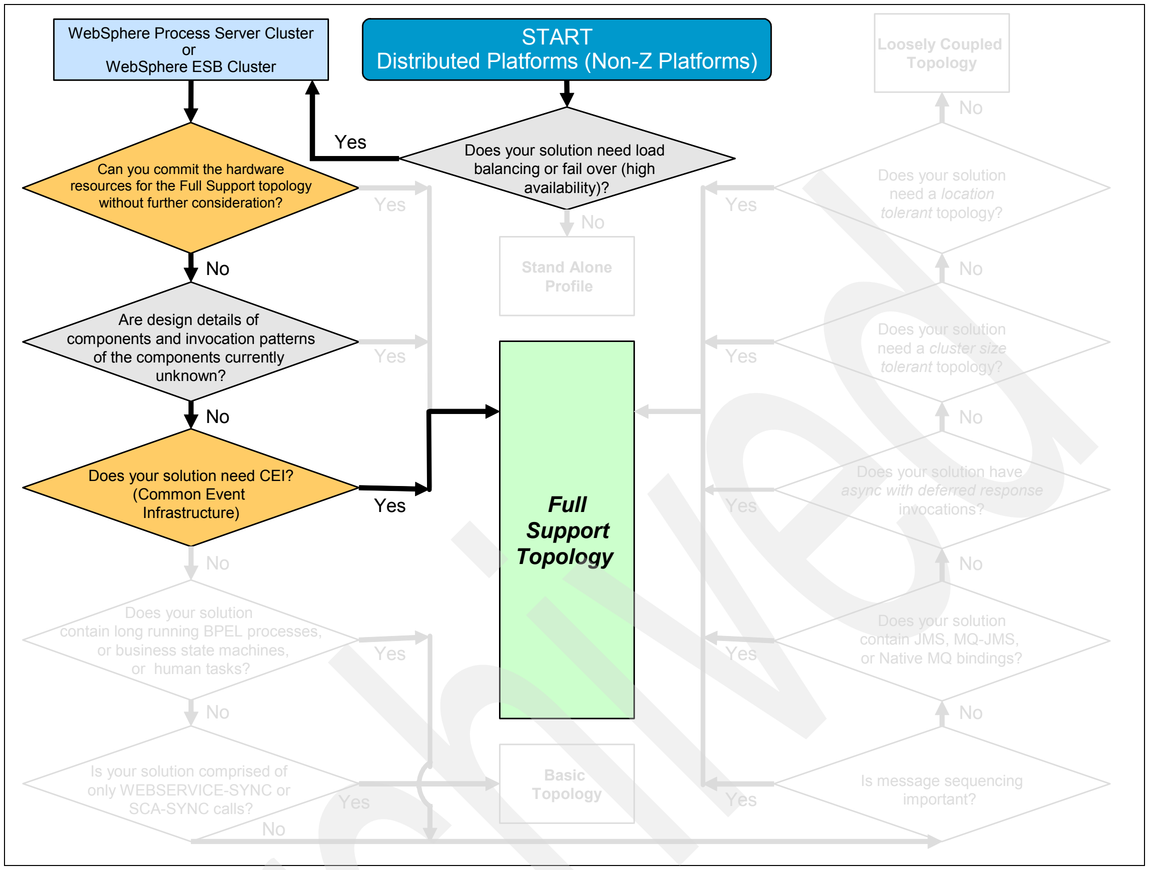
ITSO Bank has the following business and IT requirements for this scenario:

- Failover capability
- The application internals are well understood
- The flow from consumer to provider is synchronous
- Event management services are required

Note: For more detailed information about this scenario, refer to 5.1.3, “Complex mediation scenario” on page 71.

7.1.2 Selecting the appropriate topology

Based on the ITSO Bank requirements for the mediation flow scenario, we can select an appropriate production topology to host the Stock Purchase mediation module. To help us select the appropriate topology, we use the topology selection flowchart, as described in 4.5.2, “Topology selection flow chart” on page 62. Figure 7-2 shows that by using the topology selection flowchart, can select the Full Support topology as the topology of choice for the ITSO Bank mediation flow scenario.



WebSphere Process Server Cluster or WebSphere ESB Cluster
START Distributed Platforms (Non-Z Platforms)
Loosely Coupled Topology
Does your solution need load balancing or fail over (high availability)?
Yes
No
Can you commit the hardware resources for the Full Support topology without further consideration?
Yes
No
Stand Alone Profile
Does your solution need a *location tolerant* topology?
Are design details of components and invocation patterns of the components currently unknown?
Does your solution need a *cluster size tolerant* topology?
Does your solution need CEI? (Common Event Infrastructure)
Full Support Topology
Does your solution have *async with deferred response* invocations?
Does your solution contain long running BPEL processes, or business state machines, or human tasks?
Does your solution contain JMS, MQ-JMS, or Native MQ bindings?
Is your solution comprised of only WEBSERVICE-SYNC or SCA-SYNC calls?
Basic Topology
Is message sequencing important?

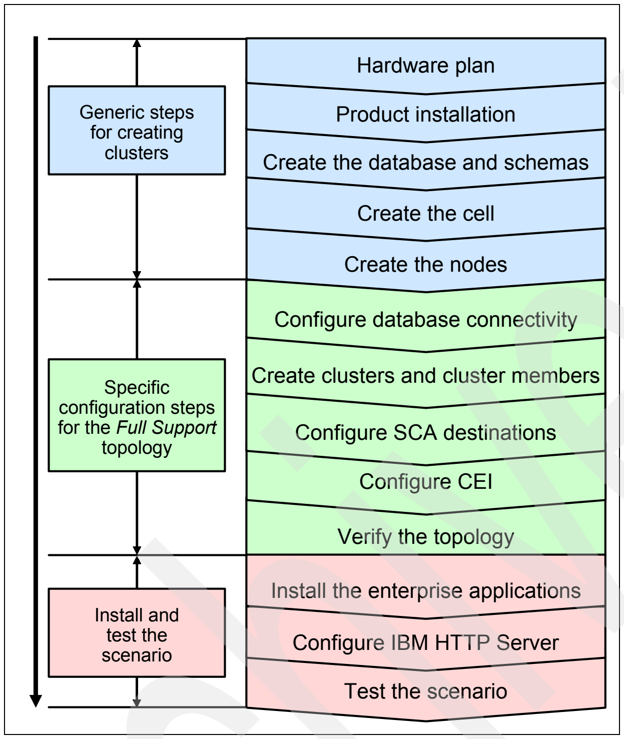
Figure 7-2 Topology selection for the complex mediation flow scenario

The main decision points for ITSO Bank are:

- A highly available solution with load balancing is required, so a clustered topology is needed.
- ITSO Bank do not wish to commit the hardware resources required to implement the Full Support topology unless the applications they are deploying to the topology require it.
- Their application solution uses CEI, therefore there is an application requirement to implement the Full Support topology.

7.1.3 Overview of how to implement the Full Support topology

Figure 7-3 summarizes the steps to build the Full Support topology.



Generic steps for creating clusters
Specific configuration steps for the *Full Support* topology
Install and test the scenario
Hardware plan
Product installation
Create the database and schemas
Create the cell
Create the nodes
Configure database connectivity
Create clusters and cluster members
Configure SCA destinations
Configure CEI
Verify the topology
Install the enterprise applications
Configure IBM HTTP Server
Test the scenario

Figure 7-3 Building the Full Support topology for WebSphere ESB

Figure 7-4 shows the topology that we build in this chapter.

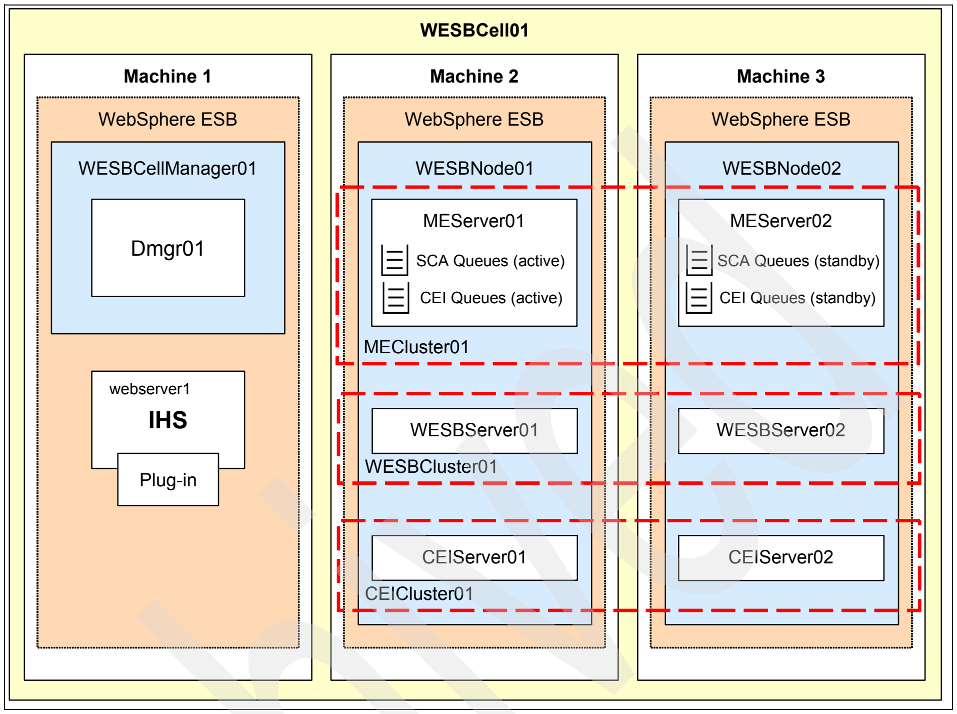


Figure 7-4 Full Support topology implementation

7.2 Generic steps for creating WebSphere ESB clusters

This section describes the hardware plan and describes how to prepare a WebSphere ESB cluster.

7.2.1 Hardware plan for the topology

This section describes the hardware that is required to support the Full Support topology. In this topology (illustrated in Figure 7-5 on page 158), three machines are used to support the cell hosting mediation module and one machine is used to host the client and Web services applications for testing purposes. A final shared machine provides the database functionality. The hardware plan includes:

- The first machine (`egt0`) hosts the deployment manager and the IBM HTTP Server and associated plugin.

- The second and third machines (`egt1` and `egt2`) host three clusters, as well as the node agents, in a horizontally clustered configuration:
  - The mediation cluster
  - A separate messaging cluster
  - The CEI cluster

  These machines need more resources because they are hosting four JVMs each.

  Note: In a production environment with high throughput, the mediation bottleneck might be the messaging engine response. In this case, the messaging engine cluster might be hosted on separate hardware to the mediation cluster. Similarly, if the CEI footprint becomes high, there is a possibility that the monitoring infrastructure (CEI cluster) might impact the mediation application. Again, you can host this cluster on different hardware to alleviate any impact.

- The fourth machine (`egt3`) hosts the client application to access the mediation module and the Web services applications in two stand-alone WebSphere application servers.
- A fifth machine (`db2`), not shown in the diagram, hosts the database that is required to support the messaging engines and CEI.

  Note: Although this is the configuration that we used, you can choose any combination of machines to create this topology.

Table 7-1 describes how we use these machines.

Table 7-1 Systems used for this topology

System name	Host name	Operating system
egt0	egt0.itso	Windows 2003 Server
egt1	egt1.itso	Windows 2003 Server
egt2	egt2.itso	Windows 2003 Server
egt3	egt3.itso	Windows 2003 Server
db2	db2.itso	RedHat Advanced Server 4 U4

Note: You can define a hosts file that maps the host names that are defined in Table 7-1 to the IP address of each of your machines to help when following the steps in this chapter.

You can find the hosts file on Windows platforms at:

<WINDOWS_HOME>\system32\drivers\etc

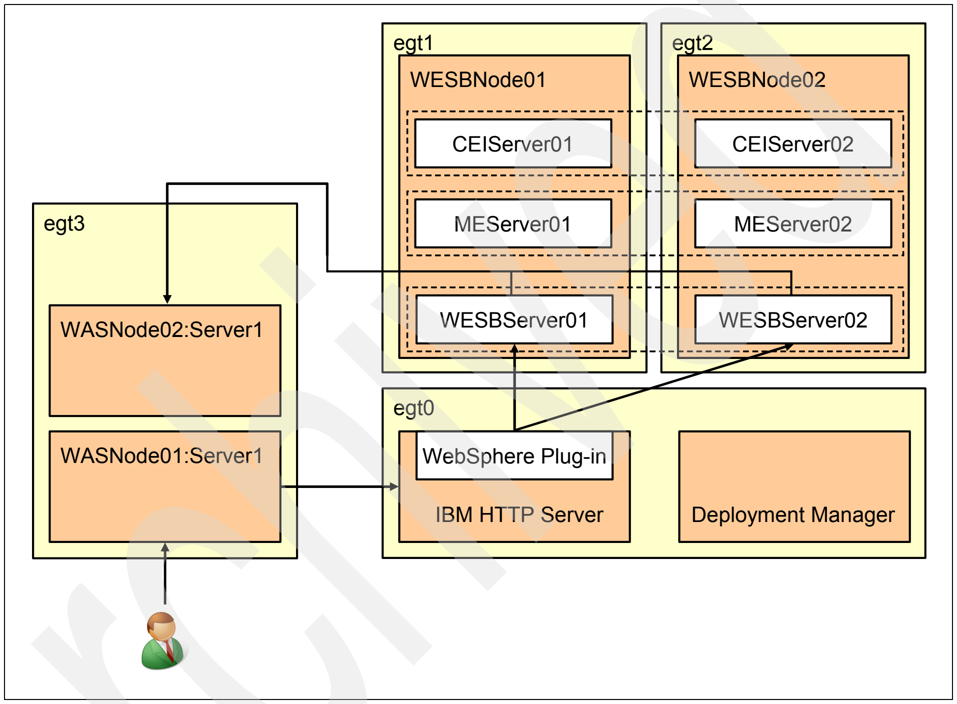


Figure 7-5 Hardware plan for the full support topology

The end-to-end flow for this topology, as shown in Figure 7-5, is:

1. A user makes a request from a browser to the front-end client application, running as a JSP on `WASNode01:server1` on host `egt3`.
2. The client application is configured to forward the request to IBM HTTP Server running on `egt0`.
3. The WebSphere Plugin distributes the request to a cluster member hosting the mediation module. The cluster member is `WESBServer01` or `WESBServer02` hosted on `egt1` or `egt2` respectively.

4. The mediation application running on this cluster member performs the appropriately mediated call to one of the Web services hosted on `WASNode02:server1` on host `egt3`. Any CEI primitive calls are also made here.
5. The return path through the infrastructure is the reverse of this flow.

7.2.2 Product installation

It is assumed that the following products have been installed:

- WebSphere ESB V6.0.2
- IBM HTTP Server V6
- DB2 Universal Database Enterprise Server V8.2

Note: For information about how to install these products, refer to Appendix B, “Product installation” on page 429.

7.2.3 Creating the messaging database

The first task in creating this topology is to create the messaging database. For this chapter, we assume the Relational Database Management System (RDBMS) is DB2, the instance is called `egtinst1`, and the administrative user name is `egtinst1`.

Log in as the database administrative user and issue the following commands from a DB2 command window to create the required database:

```
db2start
db2 create database MEDB
```

If the **`create`** command is successful, you see the following message:

```
DB20000I The CREATE DATABASE command completed successfully
```

7.2.4 Copying the database driver files

The Java driver files for DB2 are located on the database system. In our DB2 example, they are in the directory /opt/IBM/db2/V8.1/java.

On each of the nodes and the deployment manager, copy the database driver files to a standard location. In this example the standard location for the driver files is c:\db2client.

7.2.5 Creating the messaging database schemas

The messaging database includes three schemas that support the messaging engines in this topology:

- SCA Application
- SCA System
- CEI

You can create the DB2 commands for generating the schemas on any system with the WebSphere ESB product installed. On any WebSphere ESB node, create the data definition language (DDL) files for the Service Integration Bus (SIB) SCA schemas as detailed in Table 7-2.

Table 7-2 WebSphere ESB SCA schemas

Schema name	Database	Purpose
SCASYS	MEDB	System bus messaging store
SCAAPP	MEDB	Application bus messaging store
CEIMSG	MEDB	CEI messaging store

In this example:

- The database user name is `egtinst1`.
- The script to generate the schemas is located in the %*WAS_HOME*%\bin directory (for example, C:\IBM\WebSphere\ESB\bin).
- The database target for the DDL generated is running on a Linux system.

To create the messaging database schemas, follow these steps:

1. Open a command prompt and issue the following commands:

   ```
   cd C:\IBM\WebSphere\ESB\bin
   ```

   Note: Each of the commands in this series of steps is a single line although in the format of this book they might wrap. Also, the commands assume that the database is running on a UNIX platform. If you are using Windows, you need to replace -platform unix with `-platform windows`.

2. Issue the following command for the SCASYS DDL generation - SCA System Schema:

   ```
   sibDDLGenerator.bat -system db2 -version 8.1 -platform unix -schema
   SCASYS -statementend ; -user egtinst1 >
   c:\createSIBSchema_SCASYS.ddl
   ```

3. Issue the following command for the SCAAPP DDL generation - SCA Application Schema:

```
sibDDLGenerator.bat -system db2 -version 8.1 -platform unix -schema
SCAAPP -statementend ; -user egtinst1 >
c:\createSIBSchema_SCAAPP.ddl
```

4. Issue the following command for the CEIMSG DDL generation - CEI Messaging Schema:

```
sibDDLGenerator.bat -system db2 -version 8.1 -platform unix -schema
CEIMSG -statementend ; -user egtinst1 >
c:\createSIBSchema_CEIMSG.ddl
```

5. Transfer the DDL files to the DB2 database server and execute them.

   a. Transfer c:\createSIBSchema_SCASYS.ddl, c:\createSIBSchema_SCAAPP.ddl and c:\createSIBSchema_CEIMSG.dd to the DB2 database server. In this example, we transferred the DDL files to /tmp on the database server.

   Attention: If you transfer these files from Windows to UNIX, there might be return characters (`\r`) at the end of each line. To run the DDL scripts, you must remove these characters. A UNIX utility such as `dos2unix` will remove these characters.

   b. Login to the DB2 server as the appropriate user (`egtinst1`) and run the following commands to create the tables:

   ```
   db2 connect to MEDB
   db2 -tf /tmp/createSIBSchema_SCAAPP.ddl
   db2 -tf /tmp/createSIBSchema_SCASYS.ddl
   db2 -tf /tmp/createSIBSchema_CEIMSG.ddl
   db2 connect reset
   ```

Note: These DB2 commands report various informational index and primary key messages, such as:

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

These are not errors. You can ignore these messages.

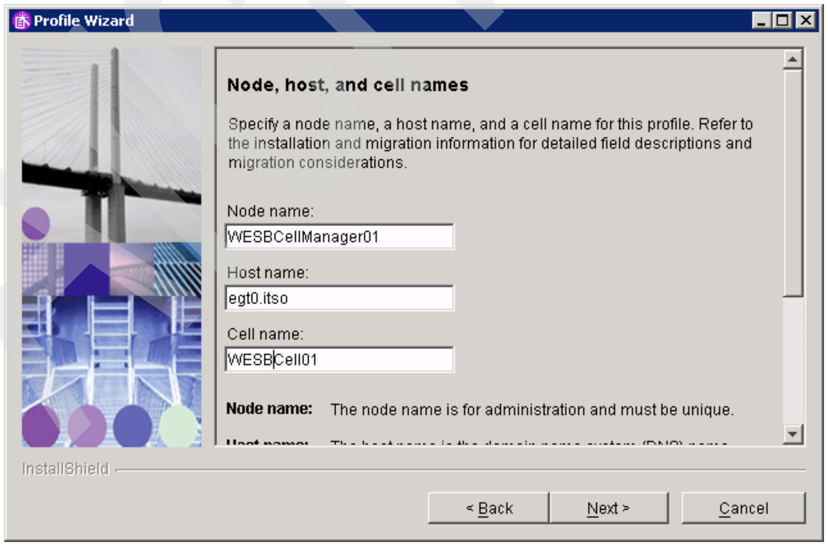
7.2.6 Creating the WebSphere ESB cell

This task creates the WebSphere ESB cell, which is achieved by creating a deployment manager profile. Follow these steps:

1. Login to the `egt0.itso` machine, which becomes the deployment manager.
2. Open the profile creation wizard by selecting **Start** → **All Programs** → **IBM WebSphere** → **Enterprise Service Bus 6.0** → **Profile creation wizard**.

> **Note:** For information about how to launch the Profile creation wizard on other platforms, see:
>
> `http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tpro_instances.html`

3. In the Profile creation wizard, click **Next**.
4. Select **Deployment manager profile** and click **Next**.
5. Leave the default profile name as *Dmgr01* and click **Next**.
6. Leave the default profile directory and click **Next**.
7. Specify the Node name, host name, and cell name according to the following specifications and click **Next**, as shown in Figure 7-6:
   – Node name: `WESBCellManager01`
   – Host name: `egt0.itso`
   – Cell name: `WESBCell01`



Node, host, and cell names

Specify a node name, a host name, and a cell name for this profile. Refer to the installation and migration information for detailed field descriptions and migration considerations.

Node name: WESBCellManager01

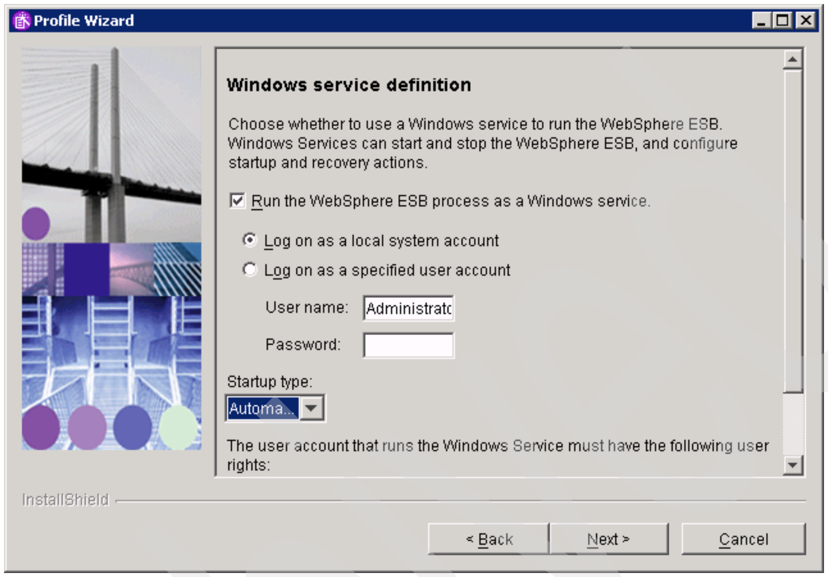
Host name: egt0.itso

Cell name: WESBCell01

Node name: The node name is for administration and must be unique.

Figure 7-6 Node, host and cell name configuration

8. Leave the port numbers at their default settings and click **Next**.
9. In the Windows service definition panel, select **Automatic** and click **Next**, as shown in Figure 7-7.



Windows service definition

Choose whether to use a Windows service to run the WebSphere ESB. Windows Services can start and stop the WebSphere ESB, and configure startup and recovery actions.

[✓] Run the WebSphere ESB process as a Windows service.

(●) Log on as a local system account

( ) Log on as a specified user account

User name: Administratc

Password:

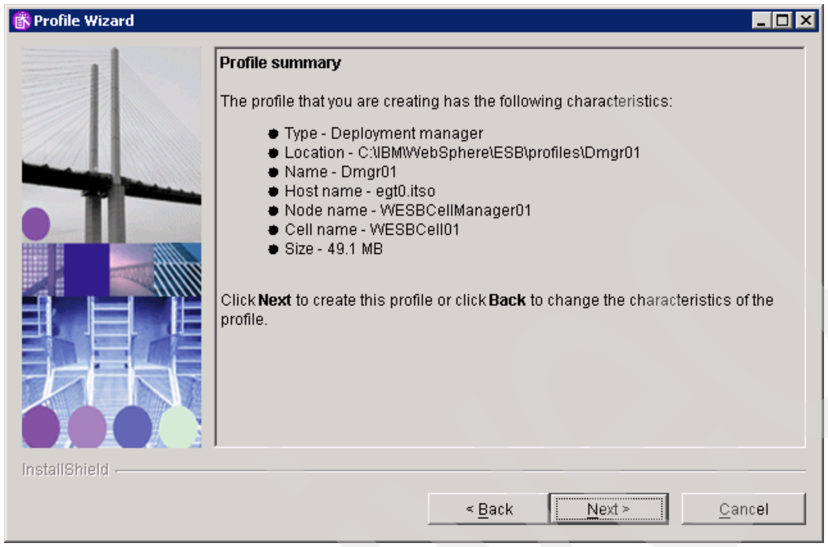
Startup type: Automa...

The user account that runs the Windows Service must have the following user rights:

Figure 7-7 Windows service settings

10. On the Service Component Architecture configuration panel, click **Next**.

11. Confirm that the details are correct on the Profile summary panel, and click **Next**, as shown in Figure 7-8.



Profile summary

The profile that you are creating has the following characteristics:

- Type - Deployment manager
- Location - C:\IBM\WebSphere\ESB\profiles\Dmgr01
- Name - Dmgr01
- Host name - egt0.itso
- Node name - WESBCellManager01
- Cell name - WESBCell01
- Size - 49.1 MB

Click **Next** to create this profile or click **Back** to change the characteristics of the profile.

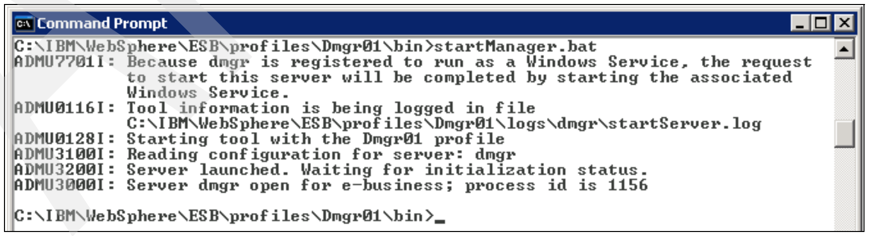
Figure 7-8 Deployment manager creation summary

12. A panel displays the progress. When complete, clear the **Launch the First Steps console** check box and click Finish. The deployment manager now exists.

13. Open a command prompt on `egt0`, change directory to %*WAS_HOME*%\profiles\Dmgr01\bin (for example, C:\IBM\WebSphere\ESB\profiles\Dmgr01\bin), and issue this command:

```
startManager.bat
```

Figure 7-9 shows the output from this command.



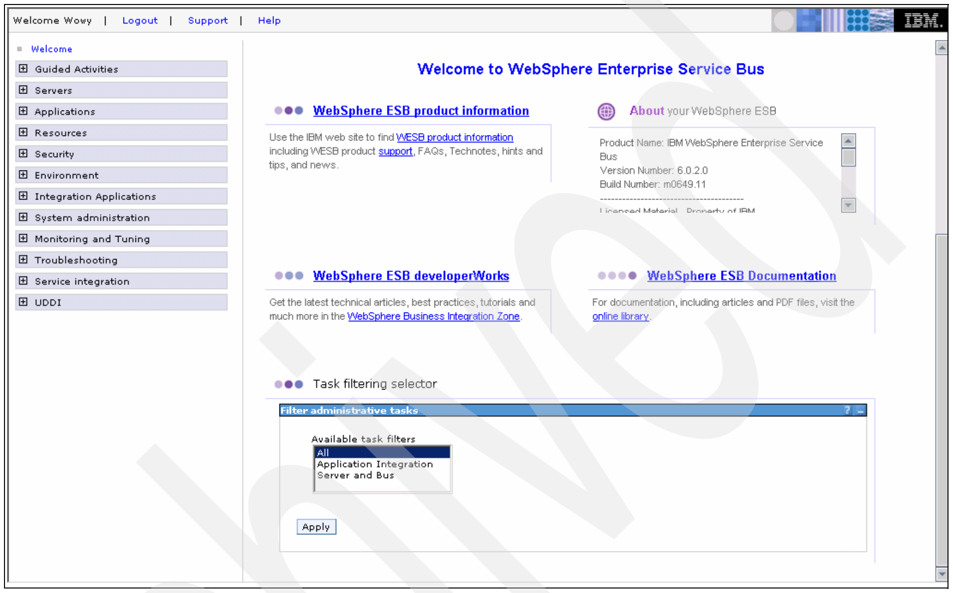
```
C:\IBM\WebSphere\ESB\profiles\Dmgr01\bin>startManager.bat
ADMU7701I: Because dmgr is registered to run as a Windows Service, the request
           to start this server will be completed by starting the associated
           Windows Service.
ADMU0116I: Tool information is being logged in file
           C:\IBM\WebSphere\ESB\profiles\Dmgr01\logs\dmgr\startServer.log
ADMU0128I: Starting tool with the Dmgr01 profile
ADMU3100I: Reading configuration for server: dmgr
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server dmgr open for e-business; process id is 1156

C:\IBM\WebSphere\ESB\profiles\Dmgr01\bin>_
```

Figure 7-9 Starting the deployment manager

It should now be possible to connect to the Administrative Console that is running on the deployment manager by opening a browser to:

```
http://egt0.itso:9060/ibm/console
```

Log in to the Administrative Console (as any user). Figure 7-10 shows the Web that opens. It might be necessary to set the available task filters to *All* in the Filter administrative tasks panel and click **Apply** on the Welcome page of the Administrative Console to override the default filtering of the available options.



Welcome to WebSphere Enterprise Service Bus

WebSphere ESB product information

Use the IBM web site to find WESB product information including WESB product support, FAQs, Technotes, hints and tips, and news.

About your WebSphere ESB

Product Name: IBM WebSphere Enterprise Service Bus
Version Number: 6.0.2.0
Build Number: m0649.11

WebSphere ESB developerWorks

Get the latest technical articles, best practices, tutorials and much more in the WebSphere Business Integration Zone.

WebSphere ESB Documentation

For documentation, including articles and PDF files, visit the online library.

Figure 7-10 First login to the administration console

7.2.7 Creating the nodes

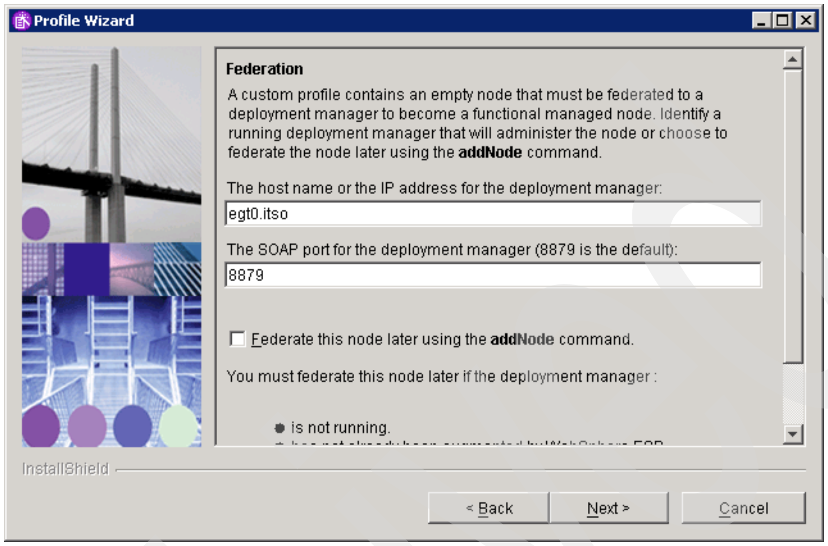
The next step in building this topology is to create the profiles and associated nodes for the topology. The deployment manager must be running on `egt0.itso` to enable node federation, the nodes are created on `egt1.itso` and `egt2.itso`.

Note: Ensure that the time difference between the deployment manager machine and the custom profile machines is within *five* minutes.

To create the nodes, follow these steps:

1. On `egt1`, open the profile creation wizard by selecting **Start** → **All Programs** → **IBM WebSphere** → **Enterprise Service Bus 6.0** → **Profile creation wizard**.
2. In the profile creation wizard, click **Next**.
3. Select **Custom profile** and click **Next**.

4. Enter the host name of the deployment manager, in this example `egt0.itso`, leave the SOAP port as the default (`8879`), ensure that **Federate this node later** is *not* selected and click **Next**, as shown in Figure 7-11.



Federation

A custom profile contains an empty node that must be federated to a deployment manager to become a functional managed node. Identify a running deployment manager that will administer the node or choose to federate the node later using the **addNode** command.

The host name or the IP address for the deployment manager:
egt0.itso

The SOAP port for the deployment manager (8879 is the default):
8879

☐ Federate this node later using the **addNode** command.

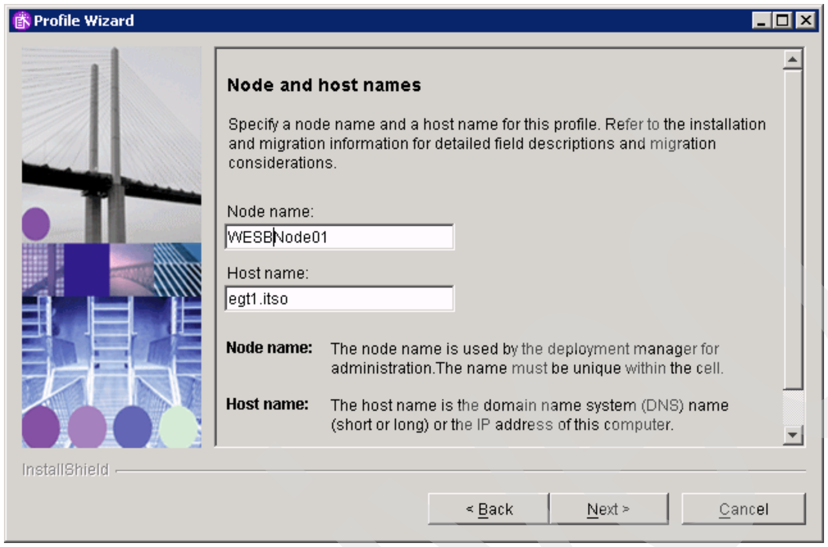
You must federate this node later if the deployment manager :

- is not running.

Figure 7-11 Deployment manager location and federation

5. Set the Profile name appropriately (`WESBNode01`) and click **Next**.
6. Accept the default profile directory and click **Next**.

7. Specify the node name (`WESBNode01`) and host name (`egt1.itso`) and click **Next**, as shown in Figure 7-12.


Profile Wizard

Node and host names

Specify a node name and a host name for this profile. Refer to the installation and migration information for detailed field descriptions and migration considerations.

Node name:
WESBNode01

Host name:
egt1.itso

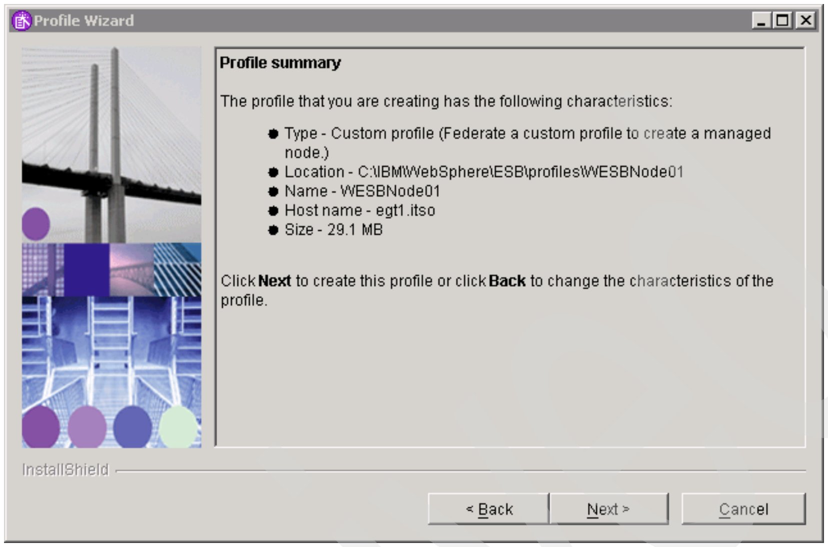
Node name: The node name is used by the deployment manager for administration.The name must be unique within the cell.

Host name: The host name is the domain name system (DNS) name (short or long) or the IP address of this computer.

Figure 7-12 Node and host name

8. Leave the default port numbers and click **Next**.

9. Confirm that the details are correct on the Profile summary panel and click **Next**, as shown in Figure 7-13.



Profile summary

The profile that you are creating has the following characteristics:

- Type - Custom profile (Federate a custom profile to create a managed node.)
- Location - C:\IBM\WebSphere\ESB\profiles\WESBNode01
- Name - WESBNode01
- Host name - egt1.itso
- Size - 29.1 MB

Click **Next** to create this profile or click **Back** to change the characteristics of the profile.

Figure 7-13 Node creation summary

10. A panel displays the progress. When complete, clear the **Launch the First Steps console** check box and click **Finish**.

The node now exists and is federated into the cell. The federation process starts the node automatically.

Repeat these steps for the second node, using the values that are shown in Table 7-3.

Table 7-3 Second node details

Setting	Value
Profile name	WESBNode02
Node name	WESBNode02
Host name	egt2.itso

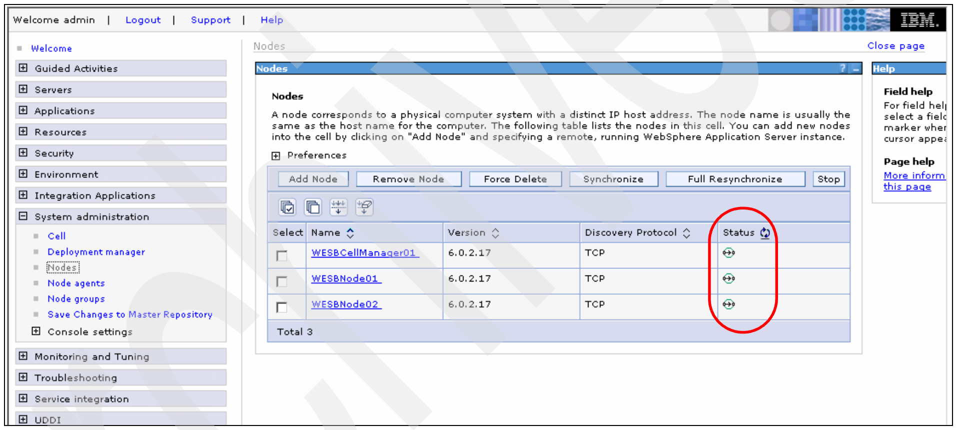
After you have created both nodes, verify that they are running in the Administrative Console by following these steps:

1. Open a browser to `http://egt0.itso:9060/ibm/console`.

   Note: You need to logout from any existing console sessions.

2. Navigate to **System administration** → **Nodes** and verify that all nodes are running as indicated by the green symbol next to each node, as shown in Figure 7-14.

   Note: Upon initial login to the Administrative Console you might need to change the Task filter selection to *All* and click **Apply** to view all available functionality.



Nodes

A node corresponds to a physical computer system with a distinct IP host address. The node name is usually the same as the host name for the computer. The following table lists the nodes in this cell. You can add new nodes into the cell by clicking on "Add Node" and specifying a remote, running WebSphere Application Server instance.

Select	Name	Version	Discovery Protocol	Status
☐	WESBCellManager01	6.0.2.17	TCP	⇔
☐	WESBNode01	6.0.2.17	TCP	⇔
☐	WESBNode02	6.0.2.17	TCP	⇔
Total 3				

Figure 7-14 Successful node installation and startup

You have now constructed the topology shown in Figure 7-15.

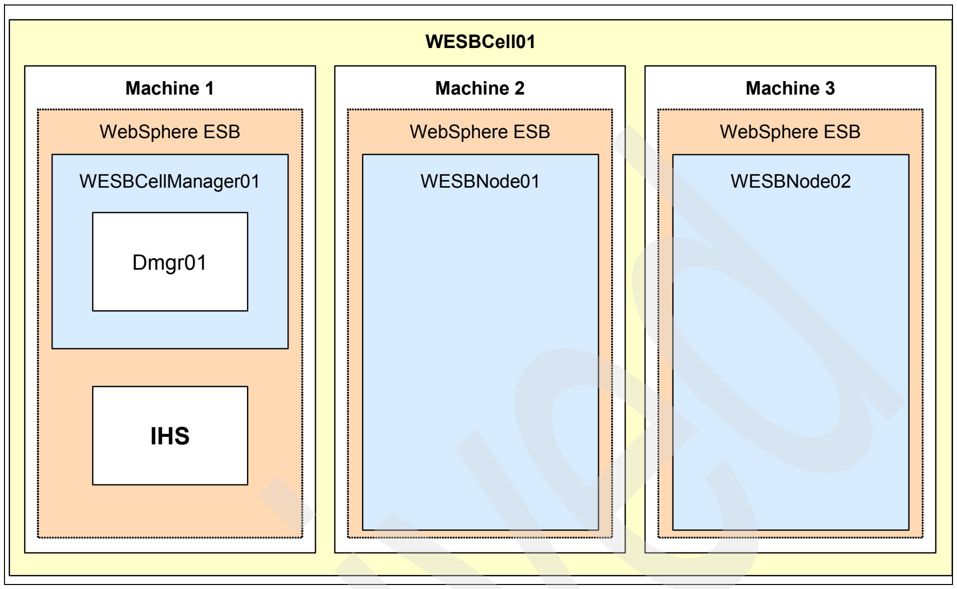


Figure 7-15 Cell and nodes defined

7.3 Specific configuration steps for the Full Support topology

This section describes the following:

- ► Configuring database connectivity
- ► Configuring the cell
- ► Configuring CEI for DB2
- ► Deploying the CEI applications
- ► Verifying the topology

7.3.1 Configuring database connectivity

This section describes how to connect successfully to the database that is required by WebSphere ESB in the Full Support topology (the MEDB - Messaging database).

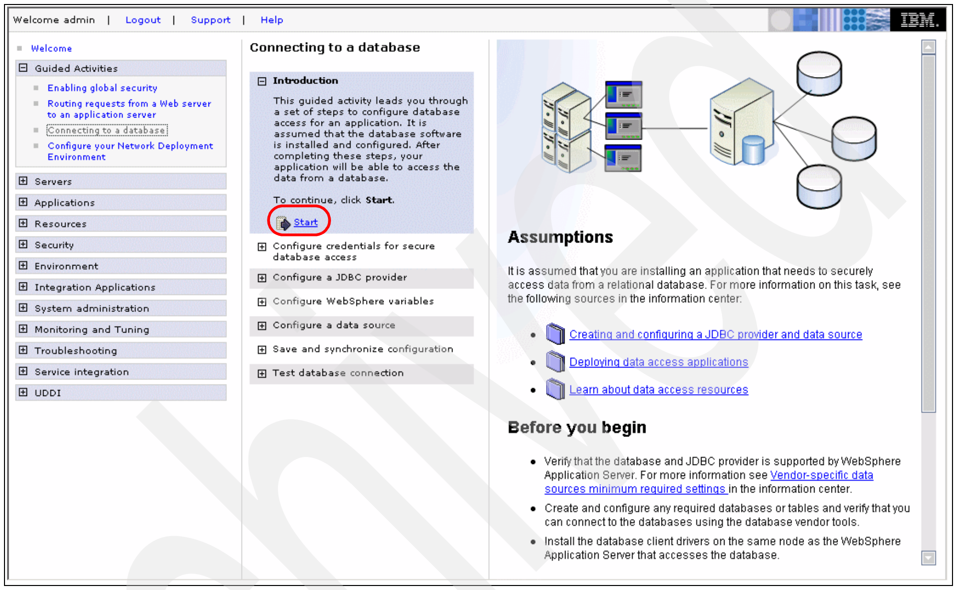
Note: We configure the CEI database later in 7.3.5, "Verifying the topology" on page 206.

To configure database connectivity, follow these steps:

1. On `egt0`, open the deployment manager Administrative Console and login. We used the following URL:

   `http://egt0.itso:9060/ibm/console`

2. Navigate to **Guided Activities** → **Connecting to a database** and click **Start**, as shown in Figure 7-16.



Welcome admin | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Connecting to a database

Introduction

This guided activity leads you through a set of steps to configure database access for an application. It is assumed that the database software is installed and configured. After completing these steps, your application will be able to access the data from a database.

To continue, click **Start**.

Start

- Configure credentials for secure database access
- Configure a JDBC provider
- Configure WebSphere variables
- Configure a data source
- Save and synchronize configuration
- Test database connection

Assumptions

It is assumed that you are installing an application that needs to securely access data from a relational database. For more information on this task, see the following sources in the information center:

- Creating and configuring a JDBC provider and data source
- Deploying data access applications
- Learn about data access resources

Before you begin

- Verify that the database and JDBC provider is supported by WebSphere Application Server. For more information see Vendor-specific data sources minimum required settings in the information center.
- Create and configure any required databases or tables and verify that you can connect to the databases using the database vendor tools.
- Install the database client drivers on the same node as the WebSphere Application Server that accesses the database.

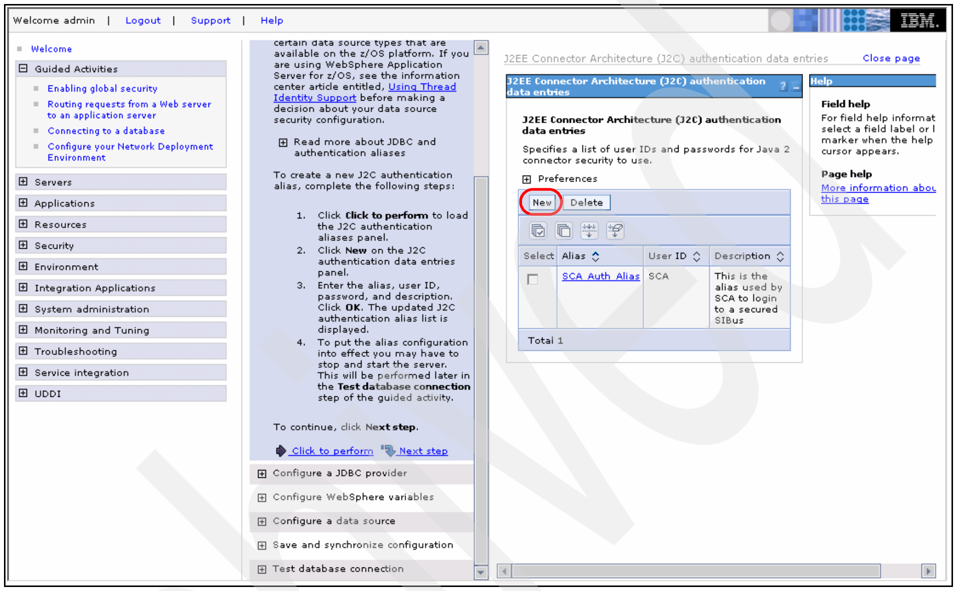
Figure 7-16 Starting the guided database activity

3. Select **Click to perform** in the new panel.

Authenticating the alias

You need to set the user and password values for database access. Follow these steps:

1. Click **New** to create a new authentication alias to connect to the database, as shown in Figure 7-17.



certain data source types that are available on the z/OS platform. If you are using WebSphere Application Server for z/OS, see the information center article entitled, Using Thread Identity Support before making a decision about your data source security configuration.

Read more about JDBC and authentication aliases

To create a new J2C authentication alias, complete the following steps:

1. Click **Click to perform** to load the J2C authentication aliases panel.
2. Click **New** on the J2C authentication data entries panel.
3. Enter the alias, user ID, password, and description. Click **OK**. The updated J2C authentication alias list is displayed.
4. To put the alias configuration into effect you may have to stop and start the server. This will be performed later in the **Test database connection** step of the guided activity.

To continue, click **Next step**.

J2EE Connector Architecture (J2C) authentication data entries

Specifies a list of user IDs and passwords for Java 2 connector security to use.

Select	Alias	User ID	Description
☐	SCA_Auth_Alias	SCA	This is the alias used by SCA to login to a secured SIBus
Total 1			

For field help informat select a field label or l marker when the help cursor appears.

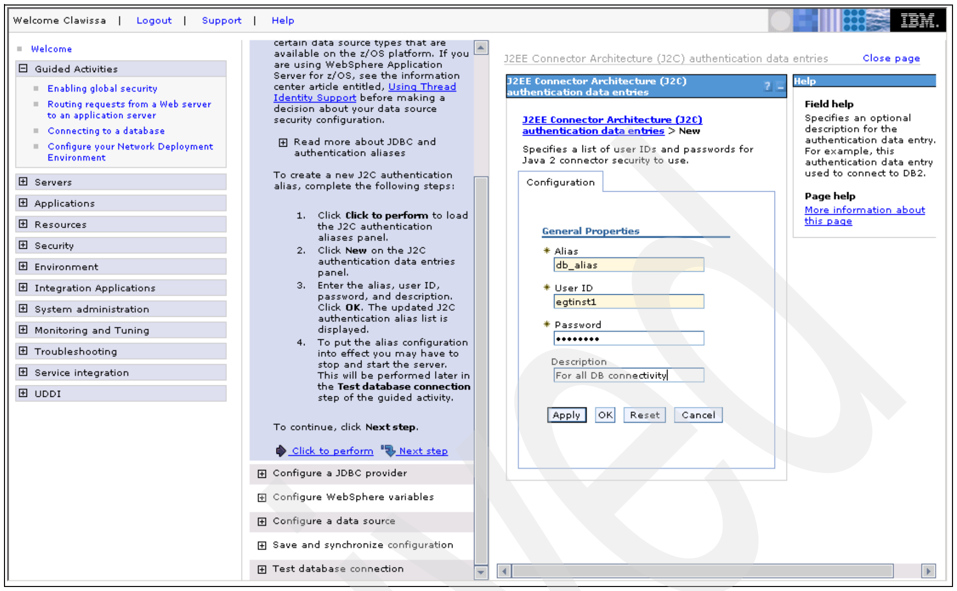
Figure 7-17 Creating a new authentication alias

2. Complete the appropriate information for the database as shown in Table 7-4 and click **OK**.

Table 7-4 Authentication alias parameters

Parameter	Value
Alias	db_alias
User ID	egtinst1
Password	itso4you
Description	For all DB connectivity

Figure 7-18 shows an example.


Welcome Clawissa | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

certain data source types that are available on the z/OS platform. If you are using WebSphere Application Server for z/OS, see the information center article entitled, Using Thread Identity Support before making a decision about your data source security configuration.

Read more about JDBC and authentication aliases

To create a new J2C authentication alias, complete the following steps:

1. Click **Click to perform** to load the J2C authentication aliases panel.
2. Click **New** on the J2C authentication data entries panel.
3. Enter the alias, user ID, password, and description. Click **OK**. The updated J2C authentication alias list is displayed.
4. To put the alias configuration into effect you may have to stop and start the server. This will be performed later in the **Test database connection** step of the guided activity.

To continue, click **Next step**.

Click to perform Next step

- Configure a JDBC provider
- Configure WebSphere variables
- Configure a data source
- Save and synchronize configuration
- Test database connection

J2EE Connector Architecture (J2C) authentication data entries   Close page

J2EE Connector Architecture (J2C) authentication data entries

J2EE Connector Architecture (J2C) authentication data entries > **New**

Specifies a list of user IDs and passwords for Java 2 connector security to use.

Configuration

General Properties

* Alias: db_alias

* User ID: egtinst1

* Password: ••••••••

Description: For all DB connectivity

Apply OK Reset Cancel

Help

Field help

Specifies an optional description for the authentication data entry. For example, this authentication data entry used to connect to DB2.

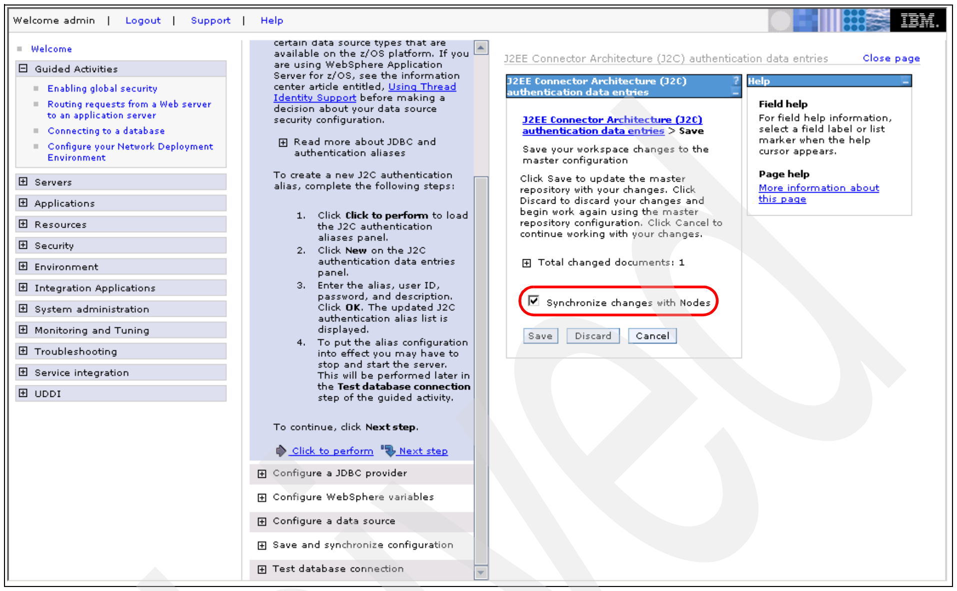
Page help

More information about this page

Figure 7-18 Authentication data entries

3. After the values are configured, click **Save** towards the top of the Web page.

4. Select **Synchronize changes with Nodes** and click **Save**, as shown in Figure 7-19.


Welcome admin | Logout | Support | Help

Welcome
Guided Activities
- Enabling global security
- Routing requests from a Web server to an application server
- Connecting to a database
- Configure your Network Deployment Environment

Servers
Applications
Resources
Security
Environment
Integration Applications
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI

certain data source types that are available on the z/OS platform. If you are using WebSphere Application Server for z/OS, see the information center article entitled, Using Thread Identity Support before making a decision about your data source security configuration.

Read more about JDBC and authentication aliases

To create a new J2C authentication alias, complete the following steps:

1. Click **Click to perform** to load the J2C authentication aliases panel.
2. Click **New** on the J2C authentication data entries panel.
3. Enter the alias, user ID, password, and description. Click **OK**. The updated J2C authentication alias list is displayed.
4. To put the alias configuration into effect you may have to stop and start the server. This will be performed later in the **Test database connection** step of the guided activity.

To continue, click **Next step**.

Click to perform   Next step

Configure a JDBC provider
Configure WebSphere variables
Configure a data source
Save and synchronize configuration
Test database connection

J2EE Connector Architecture (J2C) authentication data entries   Close page

J2EE Connector Architecture (J2C) authentication data entries

J2EE Connector Architecture (J2C) authentication data entries > **Save**

Save your workspace changes to the master configuration

Click Save to update the master repository with your changes. Click Discard to discard your changes and begin work again using the master repository configuration. Click Cancel to continue working with your changes.

Total changed documents: 1

☑ Synchronize changes with Nodes

Save   Discard   Cancel

Help

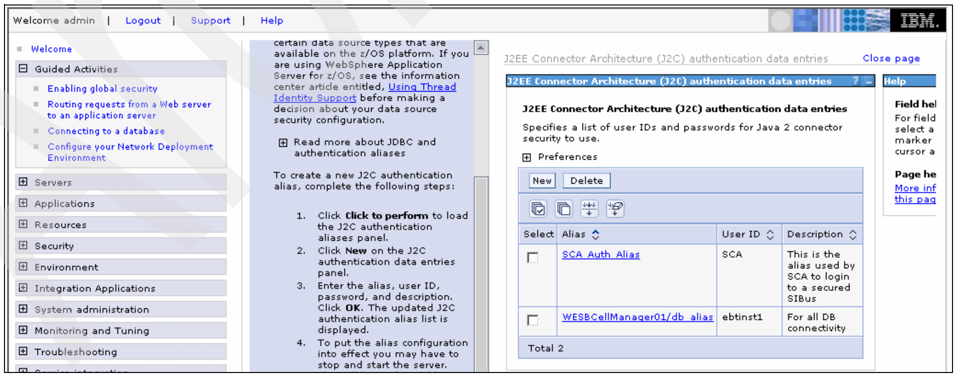
Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

Figure 7-19   Synchronizing the changes

5. When completed, verify that there were no errors with the synchronization and click **OK**.

   A new authentication alias called `WESBCellManager01/db_alias` is created, as shown in Figure 7-20.


Welcome admin | Logout | Support | Help

Welcome
Guided Activities
- Enabling global security
- Routing requests from a Web server to an application server
- Connecting to a database
- Configure your Network Deployment Environment

Servers
Applications
Resources
Security
Environment
Integration Applications
System administration
Monitoring and Tuning
Troubleshooting

certain data source types that are available on the z/OS platform. If you are using WebSphere Application Server for z/OS, see the information center article entitled, Using Thread Identity Support before making a decision about your data source security configuration.

Read more about JDBC and authentication aliases

To create a new J2C authentication alias, complete the following steps:

1. Click **Click to perform** to load the J2C authentication aliases panel.
2. Click **New** on the J2C authentication data entries panel.
3. Enter the alias, user ID, password, and description. Click **OK**. The updated J2C authentication alias list is displayed.
4. To put the alias configuration into effect you may have to stop and start the server.

J2EE Connector Architecture (J2C) authentication data entries   Close page

J2EE Connector Architecture (J2C) authentication data entries

J2EE Connector Architecture (J2C) authentication data entries

Specifies a list of user IDs and passwords for Java 2 connector security to use.

Preferences

New   Delete

Select	Alias	User ID	Description
☐	SCA_Auth_Alias	SCA	This is the alias used by SCA to login to a secured SIBus
☐	WESBCellManager01/db_alias	ebtinst1	For all DB connectivity

Total 2

Help

Field hel
For field select a marker cursor a

Page he
More inf this pag

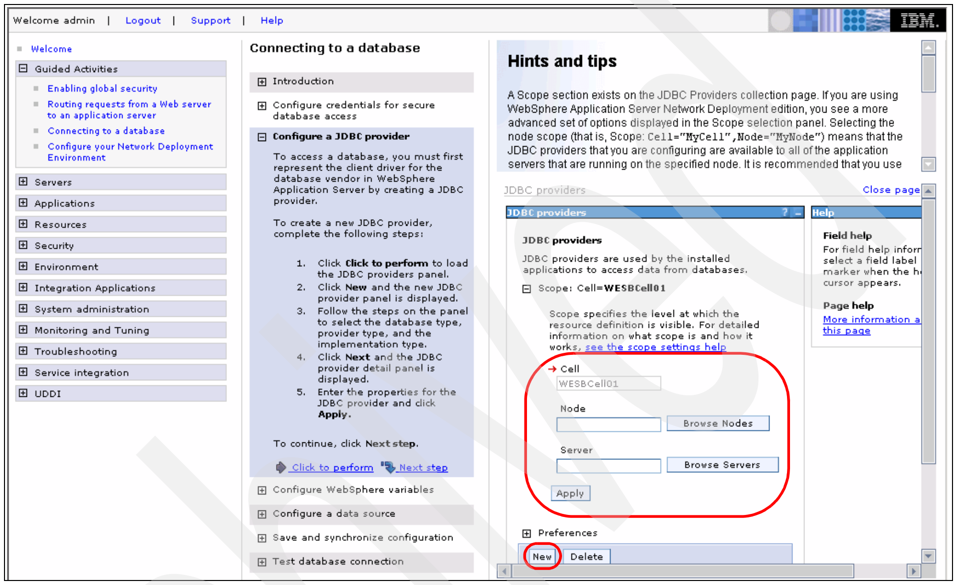
Figure 7-20   New authentication alias created

6. Select **Next step** to continue to the Configure a JDBC provider panel.

Creating a JDBC provider

The next step creates a JDBC provider for use by the data sources later:

1. In this panel, select **Click to perform**.
2. Set the scope to cell level by clearing the Node and Server entry boxes and clicking **Apply**, as shown in Figure 7-21.



To access a database, you must first represent the client driver for the database vendor in WebSphere Application Server by creating a JDBC provider.

To create a new JDBC provider, complete the following steps:

1. Click **Click to perform** to load the JDBC providers panel.
2. Click **New** and the new JDBC provider panel is displayed.
3. Follow the steps on the panel to select the database type, provider type, and the implementation type.
4. Click **Next** and the JDBC provider detail panel is displayed.
5. Enter the properties for the JDBC provider and click **Apply**.

To continue, click **Next step**.

A Scope section exists on the JDBC Providers collection page. If you are using WebSphere Application Server Network Deployment edition, you see a more advanced set of options displayed in the Scope selection panel. Selecting the node scope (that is, Scope: `Cell="MyCell",Node="MyNode"`) means that the JDBC providers that you are configuring are available to all of the application servers that are running on the specified node. It is recommended that you use

JDBC providers

JDBC providers are used by the installed applications to access data from databases.

Scope: Cell=**WESBCell01**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help

Cell: WESBCell01

Node:

Server:

Field help

For field help inforr select a field label marker when the h cursor appears.

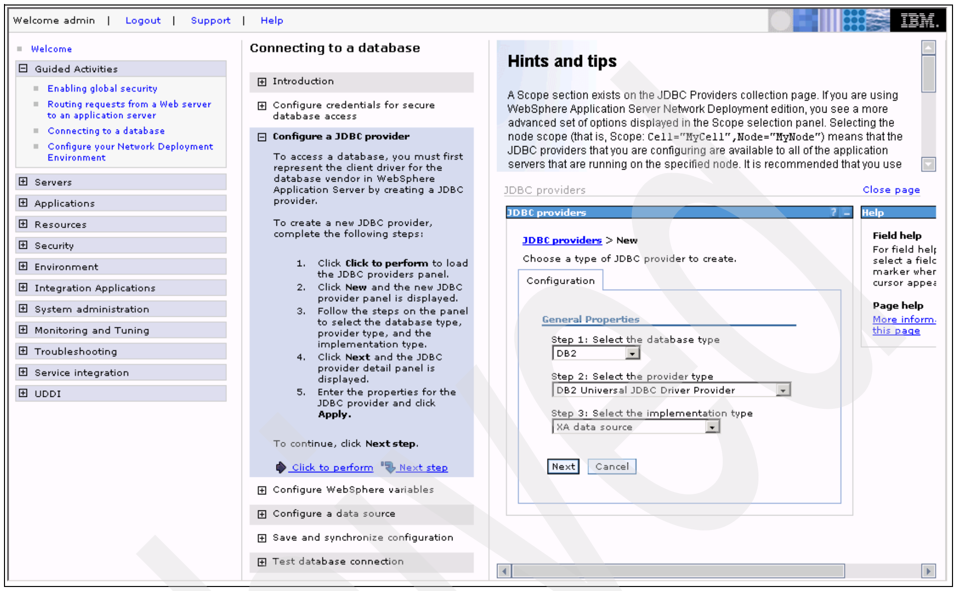
Page help

More information a this page

Figure 7-21 Setting the cell scope

3. In the JDBC providers panel below the scope, click **New**, also highlighted in Figure 7-21.

4. From the **Configuration** panel, select **DB2** as the database type, set the provider type to **DB2 Universal JDBC Driver Provider**, and the implementation type to **XA data source**. Click **Next**, as shown in Figure 7-22.



Connecting to a database

Configure a JDBC provider

To access a database, you must first represent the client driver for the database vendor in WebSphere Application Server by creating a JDBC provider.

To create a new JDBC provider, complete the following steps:

1. Click **Click to perform** to load the JDBC providers panel.
2. Click **New** and the new JDBC provider panel is displayed.
3. Follow the steps on the panel to select the database type, provider type, and the implementation type.
4. Click **Next** and the JDBC provider detail panel is displayed.
5. Enter the properties for the JDBC provider and click **Apply.**

To continue, click **Next step.**

Hints and tips

A Scope section exists on the JDBC Providers collection page. If you are using WebSphere Application Server Network Deployment edition, you see a more advanced set of options displayed in the Scope selection panel. Selecting the node scope (that is, Scope: `Cell="MyCell",Node="MyNode"`) means that the JDBC providers that you are configuring are available to all of the application servers that are running on the specified node. It is recommended that you use

JDBC providers > New

Choose a type of JDBC provider to create.

Configuration

General Properties

Step 1: Select the database type: DB2

Step 2: Select the provider type: DB2 Universal JDBC Driver Provider

Step 3: Select the implementation type: XA data source

Figure 7-22 JDBC provider configuration

5. In the next panel, which names the provider and locates the driver files, accept all the defaults and click **OK**.
6. Save and synchronize these changes and click **OK** when complete.
7. Select **Next step** to open the WebSphere variables panel and then in this panel select **Click to perform**.

Setting the WebSphere variables

The JDBC provider uses WebSphere environment variables to identify the driver file locations. For these variables to resolve, you need to set their vales appropriately. The scope for these variables should be **Cell** or **Node**, as shown in Table 7-5 and Table 7-6.

Note: When setting the scope, the selection is not applied until you click **Apply**.

To set the WebSphere variables:

1. Clear the node and server entries, and click **Apply** to set the active scope to the cell.

2. Create or modify each environment variable, and set the value as defined in Table 7-5.

Table 7-5 Cell scope environment variables

Name	Value	Scope
UNIVERSAL_JDBC_DRIVER_PATH	c:\db2client	Cell
DB2UNIVERSAL_JDBC_DRIVER_PATH	c:\db2client	Cell
DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH	c:\db2client	Cell

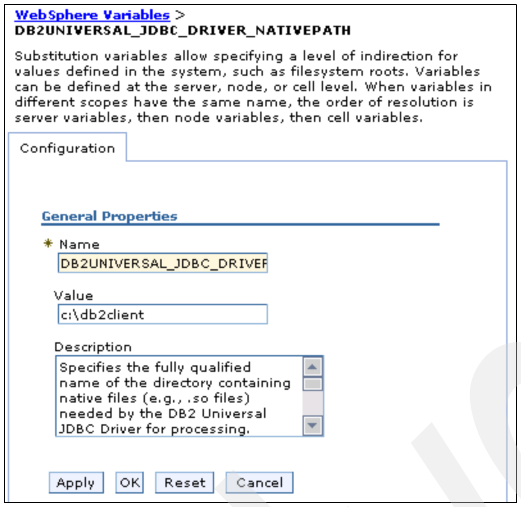
3. When you have defined all three variables at cell level, click **Browse Nodes**. Select **WESBNode01** and click **OK**. Click **Apply** to set the active scope to the node.
4. Define the node scope environment variable DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH, and set the value as shown in Table 7-6. Also delete the other two variables shown in Table 7-6.

Attention: At the node scope, you *must* delete the environment variables UNIVERSAL_JDBC_DRIVER_PATH and DB2UNIVERSAL_JDBC_DRIVER_PATH.

Table 7-6 Node scope environment variables

Name	Value	Scope
DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH	c:\db2client	WESBNode01 WESBNode02
UNIVERSAL_JDBC_DRIVER_PATH	*Delete this variable*	WESBNode01 WESBNode02
DB2UNIVERSAL_JDBC_DRIVER_PATH	*Delete this variable*	WESBNode01 WESBNode02

Figure 7-23 shows an example of the environment variable configuration.


WebSphere Variables > DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH

Substitution variables allow specifying a level of indirection for values defined in the system, such as filesystem roots. Variables can be defined at the server, node, or cell level. When variables in different scopes have the same name, the order of resolution is server variables, then node variables, then cell variables.

* Name: DB2UNIVERSAL_JDBC_DRIVEF

Value: c:\db2client

Description: Specifies the fully qualified name of the directory containing native files (e.g., .so files) needed by the DB2 Universal JDBC Driver for processing.

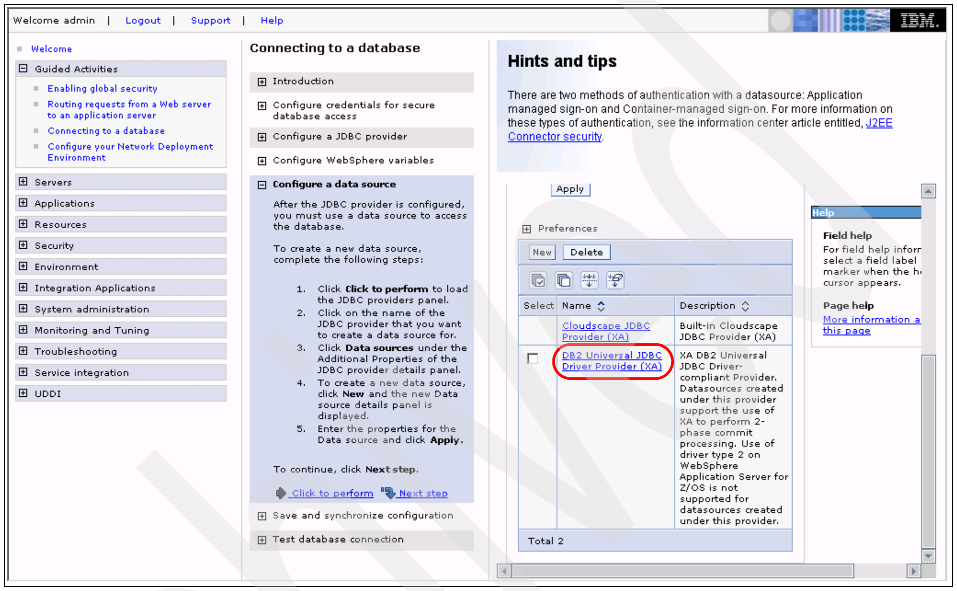
Figure 7-23 Setting an environment variable value

5. Repeat this process for `WESBNode02` to remove the two environment variables and set the third to the value shown in Table 7-6 on page 177.
6. Save and synchronize these changes.
7. Select **Next step** and then **Click to perform**.

Configuring the data sources

The next step is to configure the data sources using the JDBC provider that you just created:

1. In the new panel, click the new JDBC provider **DB2 Universal JDBC Driver Provider (XA)**, as shown in Figure 7-24.



Welcome admin | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Connecting to a database

- Introduction
- Configure credentials for secure database access
- Configure a JDBC provider
- Configure WebSphere variables
- **Configure a data source**

  After the JDBC provider is configured, you must use a data source to access the database.

  To create a new data source, complete the following steps:

  1. Click **Click to perform** to load the JDBC providers panel.
  2. Click on the name of the JDBC provider that you want to create a data source for.
  3. Click **Data sources** under the Additional Properties of the JDBC provider details panel.
  4. To create a new data source, click **New** and the new Data source details panel is displayed.
  5. Enter the properties for the Data source and click **Apply**.

  To continue, click **Next step**.

  Click to perform   Next step
- Save and synchronize configuration
- Test database connection

Hints and tips

There are two methods of authentication with a datasource: Application managed sign-on and Container-managed sign-on. For more information on these types of authentication, see the information center article entitled, J2EE Connector security.

Apply

Preferences

New   Delete

Select	Name	Description
	Cloudscape JDBC Provider (XA)	Built-in Cloudscape JDBC Provider (XA)
☐	DB2 Universal JDBC Driver Provider (XA)	XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on WebSphere Application Server for Z/OS is not supported for datasources created under this provider.
Total 2		

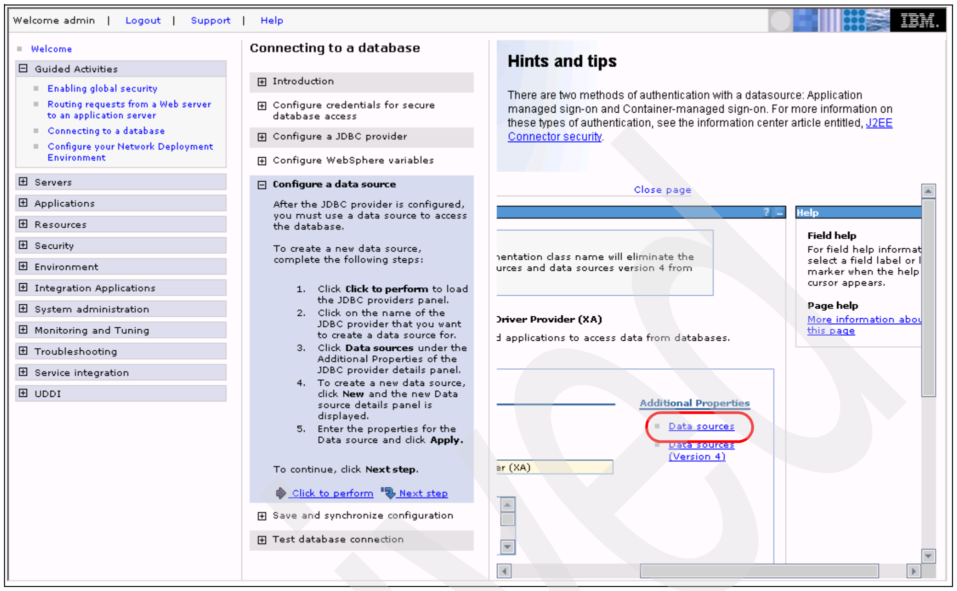
Help

Field help
For field help infor
select a field label
marker when the h
cursor appears.

Page help
More information a
this page

Figure 7-24 JDBC provider selection for data source creation

2. Scroll to the right and click **Data sources**, as shown in Figure 7-25.



Welcome admin | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Connecting to a database

- Introduction
- Configure credentials for secure database access
- Configure a JDBC provider
- Configure WebSphere variables
- **Configure a data source**

  After the JDBC provider is configured, you must use a data source to access the database.

  To create a new data source, complete the following steps:

  1. Click **Click to perform** to load the JDBC providers panel.
  2. Click on the name of the JDBC provider that you want to create a data source for.
  3. Click **Data sources** under the Additional Properties of the JDBC provider details panel.
  4. To create a new data source, click **New** and the new Data source details panel is displayed.
  5. Enter the properties for the Data source and click **Apply.**

  To continue, click **Next step.**

  Click to perform   Next step
- Save and synchronize configuration
- Test database connection

Hints and tips

There are two methods of authentication with a datasource: Application managed sign-on and Container-managed sign-on. For more information on these types of authentication, see the information center article entitled, J2EE Connector security.

Close page

ıentation class name will eliminate the
urces and data sources version 4 from
Driver Provider (XA)
d applications to access data from databases.
er (XA)
Additional Properties
Data sources
Data sources (Version 4)

Help

Field help
For field help informat select a field label or l marker when the help cursor appears.

Page help
More information abou this page

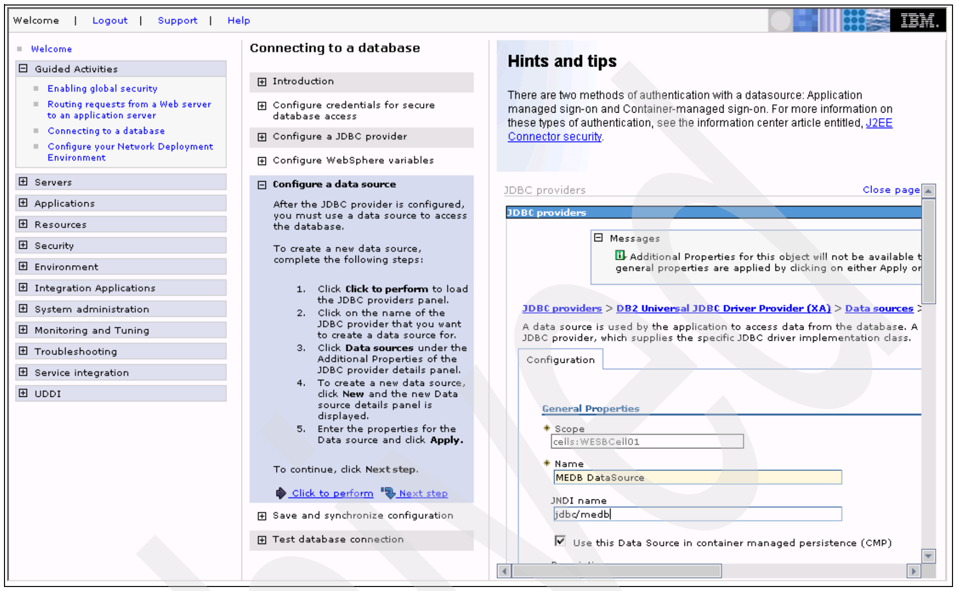
Figure 7-25 Selecting the additional data source property

3. You need to create a data source for the messaging database, MEDB. Click **New** to create a data source and set the parameters, as shown in Table 7-7.

Table 7-7 Data source configuration

Parameter	Messaging Parameters
Name	MEDB DataSource
JNDI name	jdbc/medb
Description	Messaging engine data source
Component-managed authentication alias	WESBCellManager01/db_alias
Authentication alias for XA recovery	Use component-managed authentication alias
Database name	MEDB
Driver type	4
Server name	db2.itso
Port	50004

Figure 7-26 shows the MEDB data source configuration page. You need to enter all the values that are identified in Table 7-7. To enter these values, you need to scroll down in this panel.


Welcome | Logout | Support | Help

- Welcome
- ⊟ Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- ⊞ Servers
- ⊞ Applications
- ⊞ Resources
- ⊞ Security
- ⊞ Environment
- ⊞ Integration Applications
- ⊞ System administration
- ⊞ Monitoring and Tuning
- ⊞ Troubleshooting
- ⊞ Service integration
- ⊞ UDDI

Connecting to a database

- ⊞ Introduction
- ⊞ Configure credentials for secure database access
- ⊞ Configure a JDBC provider
- ⊞ Configure WebSphere variables
- ⊟ **Configure a data source**

After the JDBC provider is configured, you must use a data source to access the database.

To create a new data source, complete the following steps:

1. Click **Click to perform** to load the JDBC providers panel.
2. Click on the name of the JDBC provider that you want to create a data source for.
3. Click **Data sources** under the Additional Properties of the JDBC provider details panel.
4. To create a new data source, click **New** and the new Data source details panel is displayed.
5. Enter the properties for the Data source and click **Apply**.

To continue, click Next step.

Click to perform   Next step

- ⊞ Save and synchronize configuration
- ⊞ Test database connection

Hints and tips

There are two methods of authentication with a datasource: Application managed sign-on and Container-managed sign-on. For more information on these types of authentication, see the information center article entitled, J2EE Connector security.

JDBC providers   Close page

JDBC providers

⊟ Messages

Additional Properties for this object will not be available t general properties are applied by clicking on either Apply or

JDBC providers > DB2 Universal JDBC Driver Provider (XA) > Data sources >

A data source is used by the application to access data from the database. A JDBC provider, which supplies the specific JDBC driver implementation class.

Configuration

General Properties

* Scope
cells:WESBCell01

* Name
MEDB DataSource

JNDI name
jdbc/medb

[✓] Use this Data Source in container managed persistence (CMP)

Figure 7-26 Data source configuration

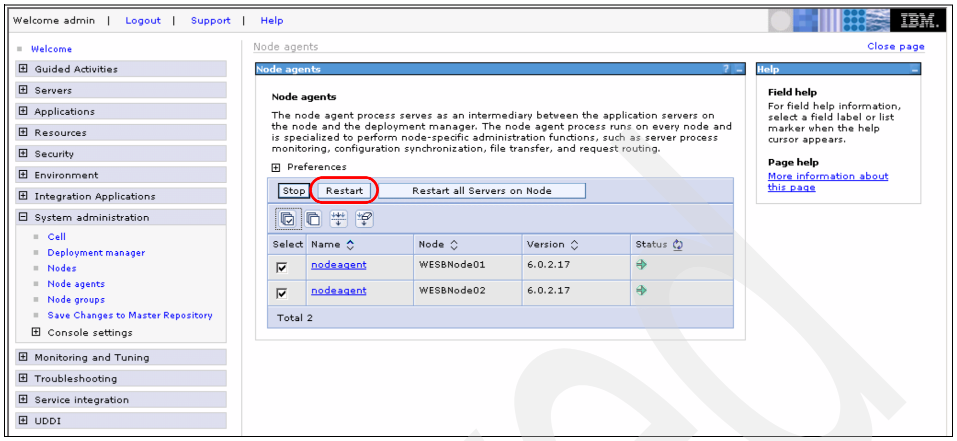
4. Click **OK** to create the new data source. When created, save and synchronize the changes and click **OK** to complete.
5. Click Next step and then **Next step** again in the Save and synchronize configuration panel (because you have already saved and synchronized).

Restarting the environment

You need to restart all nodes and the deployment manager to ensure that the changes that you made are distributed cell wide. Follow these steps:

1. Open the Administrative Console for the deployment manager hosted on `egt0`.
2. Navigate to **System administration** → **Node agents** and select both nodes.

3. Click **Restart** to cause the nodes to stop and start, as shown in Figure 7-27.


Welcome admin | Logout | Support | Help

Welcome
Guided Activities
Servers
Applications
Resources
Security
Environment
Integration Applications
System administration
Cell
Deployment manager
Nodes
Node agents
Node groups
Save Changes to Master Repository
Console settings
Monitoring and Tuning
Troubleshooting
Service integration
UDDI

Node agents | Close page

Node agents

The node agent process serves as an intermediary between the application servers on the node and the deployment manager. The node agent process runs on every node and is specialized to perform node-specific administration functions, such as server process monitoring, configuration synchronization, file transfer, and request routing.

Preferences

Stop | Restart | Restart all Servers on Node

Select	Name	Node	Version	Status
☑	nodeagent	WESBNode01	6.0.2.17	➔
☑	nodeagent	WESBNode02	6.0.2.17	➔
Total 2				

Help

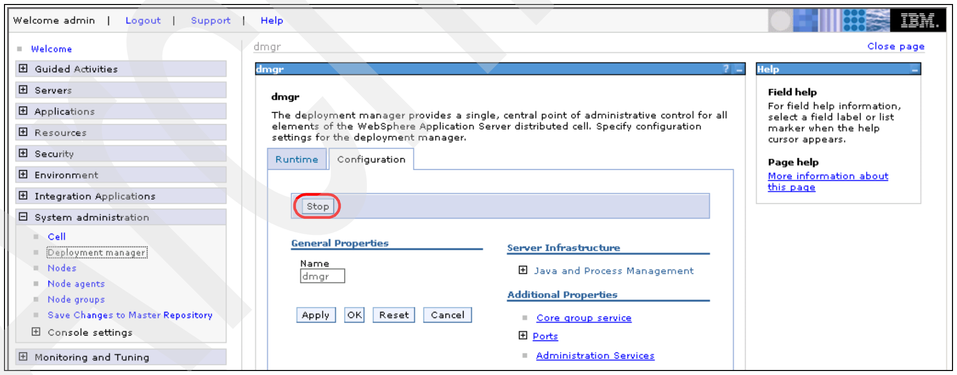
Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

Figure 7-27 Node restart

Note: You need to refresh the Node agents view to see the change in status.

4. Navigate to **System administration** → **Deployment manager** and click **Stop**, as shown in Figure 7-28.


Welcome admin | Logout | Support | Help

Welcome
Guided Activities
Servers
Applications
Resources
Security
Environment
Integration Applications
System administration
Cell
Deployment manager
Nodes
Node agents
Node groups
Save Changes to Master Repository
Console settings
Monitoring and Tuning

dmgr | Close page

dmgr

The deployment manager provides a single, central point of administrative control for all elements of the WebSphere Application Server distributed cell. Specify configuration settings for the deployment manager.

Runtime | Configuration

Stop

General Properties

Name: dmgr

Apply | OK | Reset | Cancel

Server Infrastructure
Java and Process Management

Additional Properties
Core group service
Ports
Administration Services

Help

Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

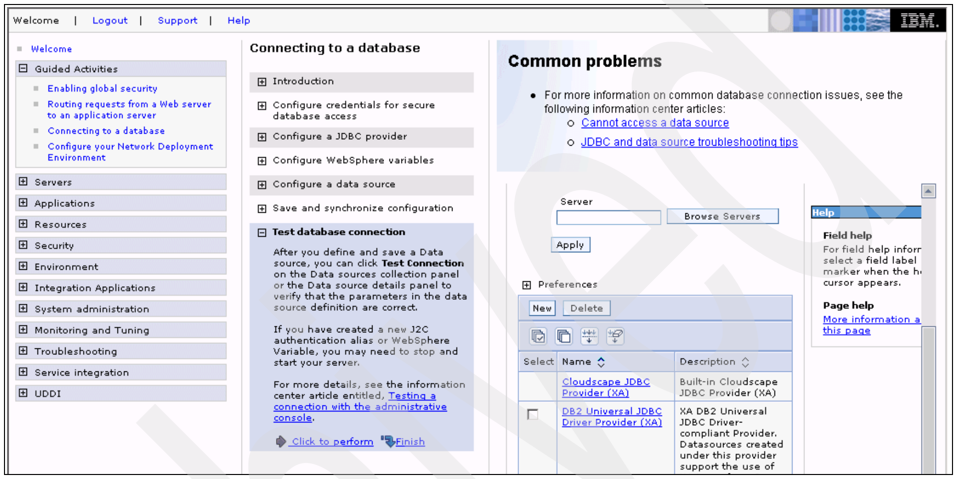
Figure 7-28 Stopping the deployment manager

5. Click **OK** in the verification panel and then start the deployment manager, as described in step 13 of the section 7.2.6, “Creating the WebSphere ESB cell” on page 162.

Testing database connectivity

After the environment has restarted, ensure that the MEDB database is running and test the newly created data sources:

1. Open the `egt0` Administrative Console and login.
2. Navigate to **Guided Activities** → **Connecting to a database** → **Test database connection**, as shown in Figure 7-29.



Welcome | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Connecting to a database

- Introduction
- Configure credentials for secure database access
- Configure a JDBC provider
- Configure WebSphere variables
- Configure a data source
- Save and synchronize configuration
- Test database connection

After you define and save a Data source, you can click **Test Connection** on the Data sources collection panel or the Data source details panel to verify that the parameters in the data source definition are correct.

If you have created a new J2C authentication alias or WebSphere Variable, you may need to stop and start your server.

For more details, see the information center article entitled, Testing a connection with the administrative console.

Click to perform   Finish

Common problems

- For more information on common database connection issues, see the following information center articles:
  - Cannot access a data source
  - JDBC and data source troubleshooting tips

Server

Browse Servers

Apply

Preferences

New   Delete

Select	Name	Description
	Cloudscape JDBC Provider (XA)	Built-in Cloudscape JDBC Provider (XA)
☐	DB2 Universal JDBC Driver Provider (XA)	XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of

Help

Field help

For field help inforr select a field label marker when the h cursor appears.

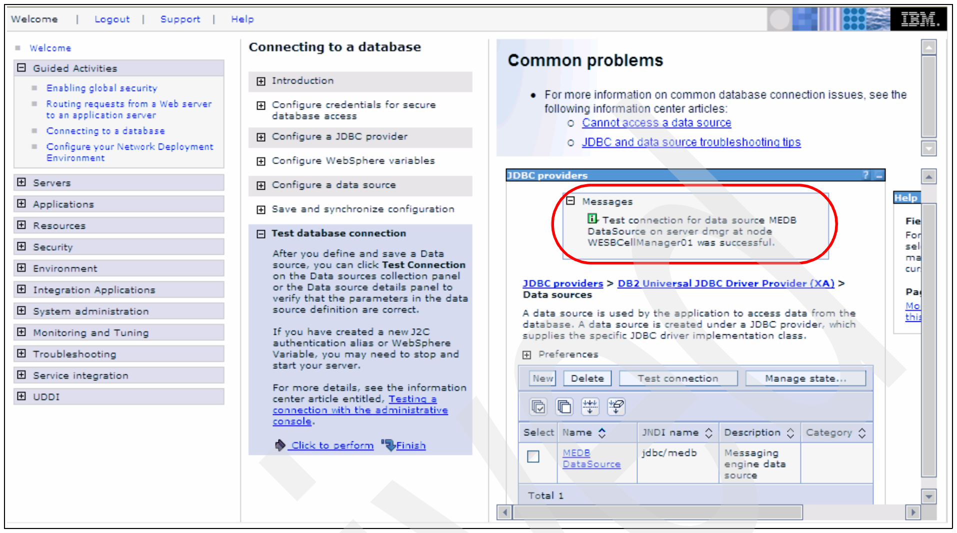
Page help

More information a this page

Figure 7-29 Testing the new data source

3. Select **Click to perform**.
4. Verify the scope is set to **Cell** and click the JDBC provider. In this example, the provide is called **DB2 Universal JDBC Driver Provider (XA)**.
5. In the next panel, scroll right, and under **Additional Properties**, click **Data sources**.

6. From the data sources panel, select **MEDB DataSource** and click **Test connection**, as shown in Figure 7-30.



Connecting to a database

Test database connection

After you define and save a Data source, you can click **Test Connection** on the Data sources collection panel or the Data source details panel to verify that the parameters in the data source definition are correct.

If you have created a new J2C authentication alias or WebSphere Variable, you may need to stop and start your server.

For more details, see the information center article entitled, Testing a connection with the administrative console.

Common problems

- For more information on common database connection issues, see the following information center articles:
  - Cannot access a data source
  - JDBC and data source troubleshooting tips

Messages

Test connection for data source MEDB DataSource on server dmgr at node WESBCellManager01 was successful.

JDBC providers > DB2 Universal JDBC Driver Provider (XA) >
Data sources

A data source is used by the application to access data from the database. A data source is created under a JDBC provider, which supplies the specific JDBC driver implementation class.

Select	Name	JNDI name	Description	Category
☐	MEDB DataSource	jdbc/medb	Messaging engine data source	
Total 1				

Figure 7-30 Successful database connections

7. The Messages dialog box (at the top in Figure 7-30) should indicate that the data source successfully connected to the database.

8. Click **Finish**.

7.3.2 Configuring the cell

To configure the network deployment infrastructure with the options that are required for the Full Support topology, follow these steps:

1. Open the Administrative Console on egt0 and login.

2. Navigate to **Guided Activities** → **Configure your Network Deployment Environment** and click Start.

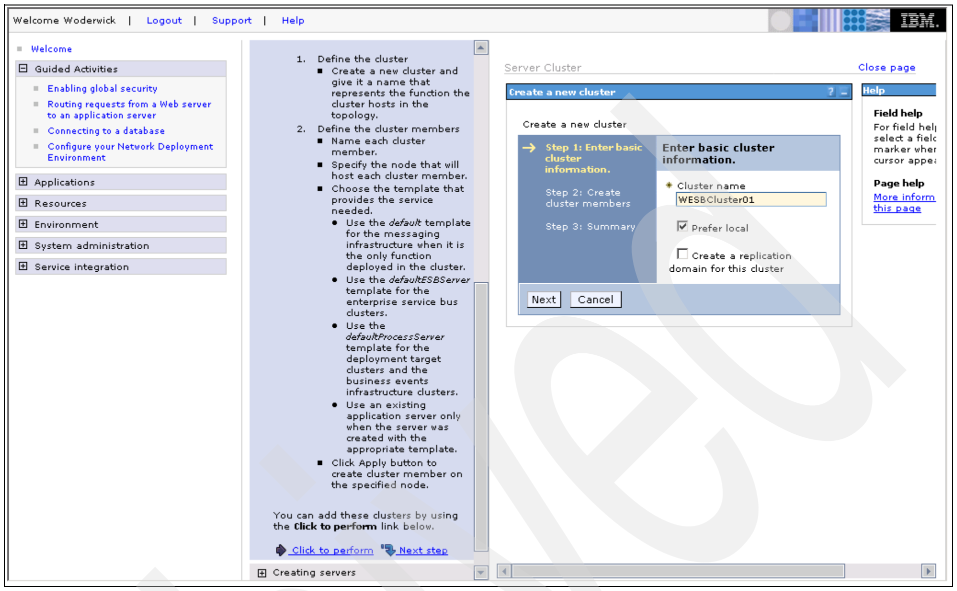
3. Scroll down and select **Click to perform** in the expanded panel to see the list of nodes.

4. The nodes are already federated, so select **Next step**.

5. The database configuration is already complete, so select **Next step**.

6. In the Creating clusters and cluster members panel, select **Click to perform**.

7. Click **New** and enter a Cluster name of `WESBCluster01`, as shown in Figure 7-31.



Welcome Woderwick | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Applications
- Resources
- Environment
- System administration
- Service integration

1. Define the cluster
   - Create a new cluster and give it a name that represents the function the cluster hosts in the topology.
2. Define the cluster members
   - Name each cluster member.
   - Specify the node that will host each cluster member.
   - Choose the template that provides the service needed.
     - Use the *default* template for the messaging infrastructure when it is the only function deployed in the cluster.
     - Use the *defaultESBServer* template for the enterprise service bus clusters.
     - Use the *defaultProcessServer* template for the deployment target clusters and the business events infrastructure clusters.
     - Use an existing application server only when the server was created with the appropriate template.
   - Click Apply button to create cluster member on the specified node.

You can add these clusters by using the **Click to perform** link below.

Click to perform Next step

Creating servers

Server Cluster Close page

Create a new cluster

Create a new cluster

→ **Step 1: Enter basic cluster information.**

Step 2: Create cluster members

Step 3: Summary

Enter basic cluster information.

* Cluster name

WESBCluster01

[✓] Prefer local

[ ] Create a replication domain for this cluster

Next Cancel

Help

Field help

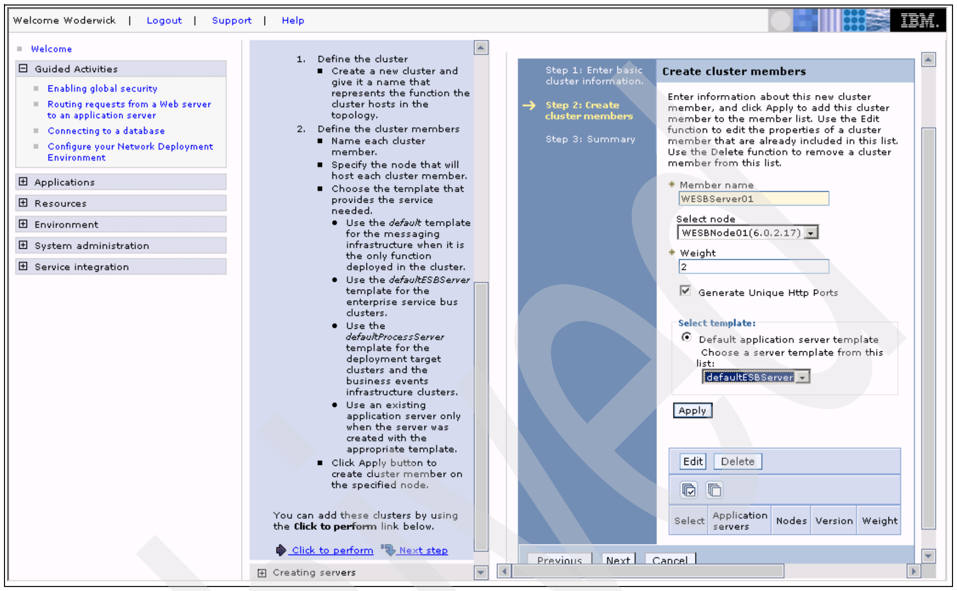
For field hel select a fielc marker wher cursor appe

Page help

More inform this page

Figure 7-31 Cluster creation

8. Click **Next** and then add a first member of the cluster named `WESBServer01` to the node `WESBNode01`. Set the template to `defaultESBServer`, as shown in Figure 7-32.



Welcome Woderwick | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Applications
- Resources
- Environment
- System administration
- Service integration

1. Define the cluster
   - Create a new cluster and give it a name that represents the function the cluster hosts in the topology.
2. Define the cluster members
   - Name each cluster member.
   - Specify the node that will host each cluster member.
   - Choose the template that provides the service needed.
     - Use the *default* template for the messaging infrastructure when it is the only function deployed in the cluster.
     - Use the *defaultESBServer* template for the enterprise service bus clusters.
     - Use the *defaultProcessServer* template for the deployment target clusters and the business events infrastructure clusters.
     - Use an existing application server only when the server was created with the appropriate template.
   - Click Apply button to create cluster member on the specified node.

You can add these clusters by using the **Click to perform** link below.

Click to perform   Next step

Creating servers

Step 1: Enter basic cluster information.
→ Step 2: Create cluster members
Step 3: Summary

Create cluster members

Enter information about this new cluster member, and click Apply to add this cluster member to the member list. Use the Edit function to edit the properties of a cluster member that are already included in this list. Use the Delete function to remove a cluster member from this list.

* Member name: WESBServer01

Select node: WESBNode01(6.0.2.17)

* Weight: 2

[✓] Generate Unique Http Ports

Select template:
● Default application server template
Choose a server template from this list: defaultESBServer

Apply

Edit   Delete

Select	Application servers	Nodes	Version	Weight

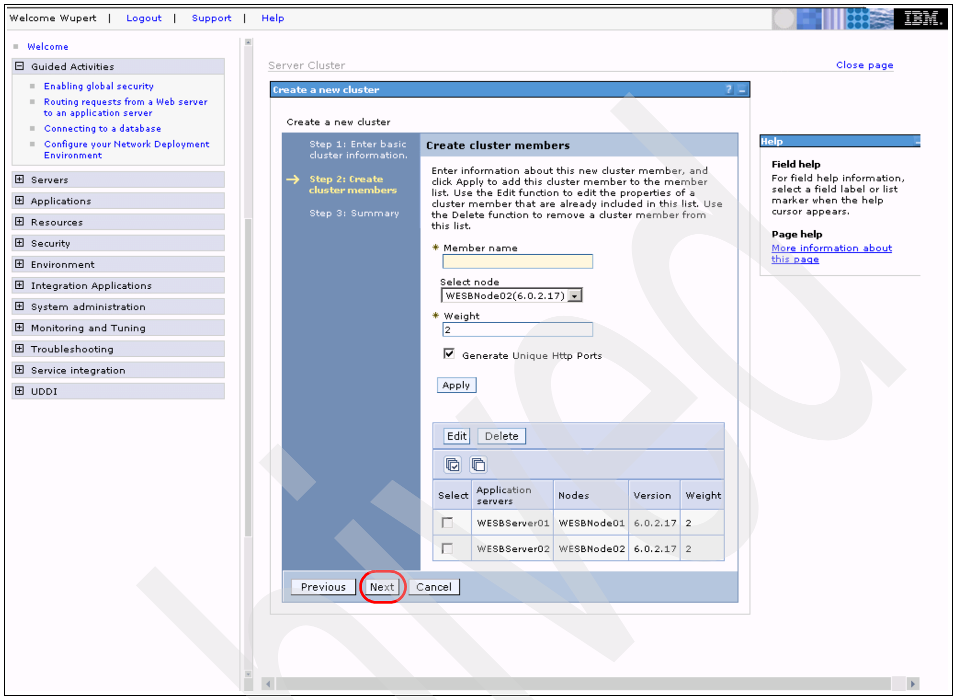
Previous   Next   Cancel

Figure 7-32   Adding the first cluster member

9. Click **Apply** to add this server to the new cluster.

10. Repeat this process for `WESBServer02` on `WESBNode02`. However, the template option will not be available now because all cluster members are of the type that is defined by the first member.

11. Click **Apply** to add the second cluster member and then click **Next**, as shown in Figure 7-33.



Create a new cluster

Create cluster members

Enter information about this new cluster member, and click Apply to add this cluster member to the member list. Use the Edit function to edit the properties of a cluster member that are already included in this list. Use the Delete function to remove a cluster member from this list.

* Member name

Select node: WESBNode02(6.0.2.17)

* Weight: 2

☑ Generate Unique Http Ports

Apply

Edit Delete

Select	Application servers	Nodes	Version	Weight
☐	WESBServer01	WESBNode01	6.0.2.17	2
☐	WESBServer02	WESBNode02	6.0.2.17	2

Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

Figure 7-33 Cluster members added

12. At the summary page, click **Finish**.

13. Repeat this process for the messaging cluster, using the details that are listed in Table 7-8.

Table 7-8 Messaging cluster configuration

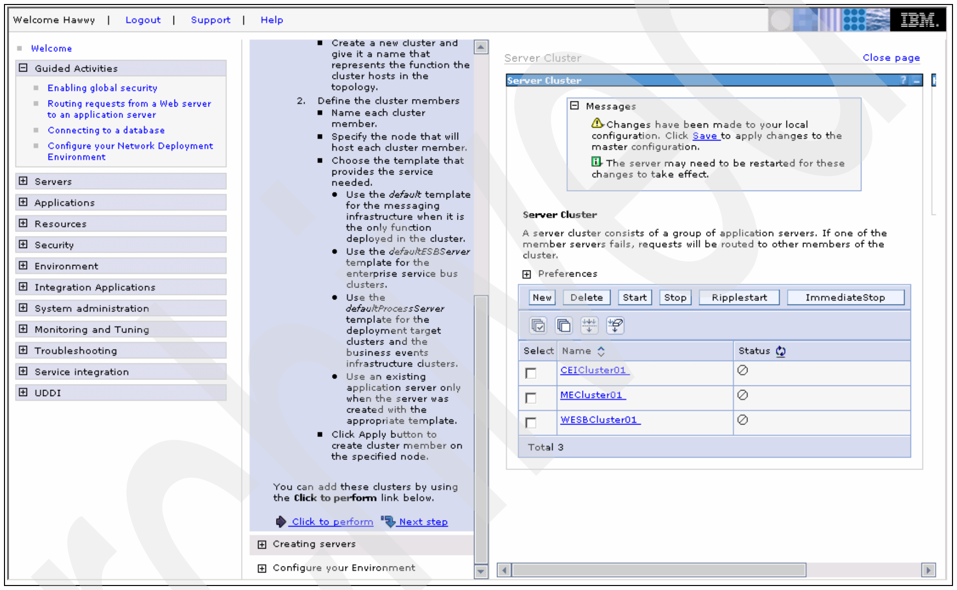
Cluster name	Cluster members	Node name	Template
MECluster01	MEServer01 MEServer02	WESBNode01 WESBNode02	default

14. Repeat this process for the CEI cluster, using the details that are listed in Table 7-9.

Table 7-9 CEI cluster configuration

Cluster name	Cluster members	Node name	Template
CEICluster01	CEIServer01 CEIServer02	WESBNode01 WESBNode02	defaultESBServer

The three clusters are created, as shown in Figure 7-34.



- Create a new cluster and give it a name that represents the function the cluster hosts in the topology.

2. Define the cluster members
   - Name each cluster member.
   - Specify the node that will host each cluster member.
   - Choose the template that provides the service needed.
     - Use the *default* template for the messaging infrastructure when it is the only function deployed in the cluster.
     - Use the *defaultESBServer* template for the enterprise service bus clusters.
     - Use the *defaultProcessServer* template for the deployment target clusters and the business events infrastructure clusters.
     - Use an existing application server only when the server was created with the appropriate template.
   - Click Apply button to create cluster member on the specified node.

You can add these clusters by using the **Click to perform** link below.

Changes have been made to your local configuration. Click Save to apply changes to the master configuration.

The server may need to be restarted for these changes to take effect.

Server Cluster

A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster.

Select	Name	Status
☐	CEICluster01	⊘
☐	MECluster01	⊘
☐	WESBCluster01	⊘
Total 3		

Figure 7-34 Clusters created

15. Save and synchronize the changes.

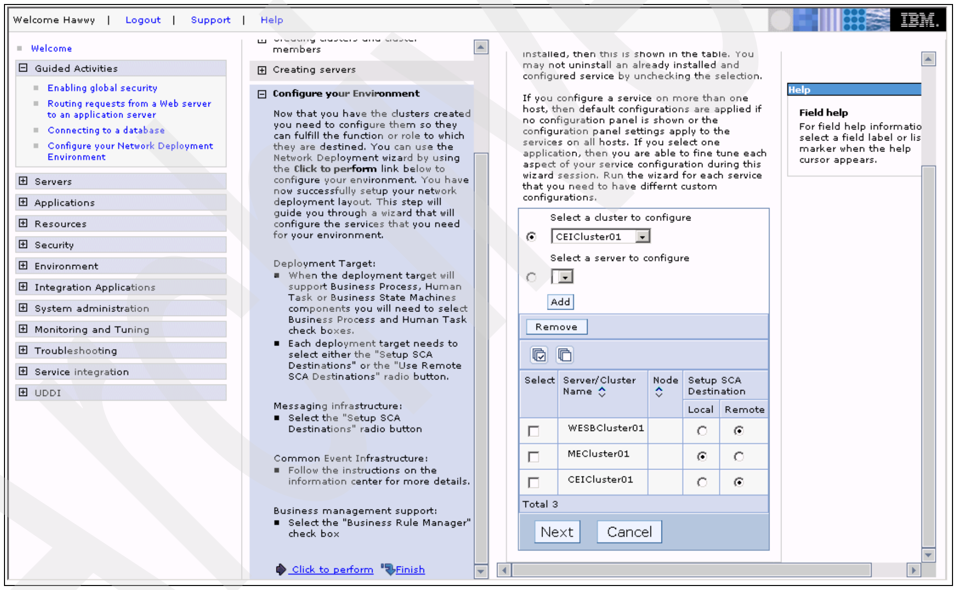
16. Select **Next step** to continue, and then select **Next step** at the **Creating servers** option because you require no more servers.

Configuring the environment

To configure the environment, follow these steps:

1. In the Configure your Environment panel, select **Click to perform**.
2. Click the pull-down menu to configure a cluster, and select the following configuration settings:
   a. Select **WESBCluster01** and click **Add**.
   b. Select **MECluster01** and click **Add**.
   c. Select **CEICluster01** and click **Add**.
   d. Select **Remote** for the **SCA Destination** for `WESBCluster01`.
   e. Select **Local** for the **SCA Destination** for `MECluster01`.
   f. Select **Remote** for the **SCA Destination** for `CEICluster01`.

   Figure 7-35 shows the configuration settings.



Now that you have the clusters created you need to configure them so they can fulfill the function or role to which they are destined. You can use the Network Deployment wizard by using the **Click to perform** link below to configure your environment. You have now successfully setup your network deployment layout. This step will guide you through a wizard that will configure the services that you need for your environment.

Deployment Target:
- When the deployment target will support Business Process, Human Task or Business State Machines components you will need to select Business Process and Human Task check boxes.
- Each deployment target needs to select either the "Setup SCA Destinations" or the "Use Remote SCA Destinations" radio button.

Messaging infrastructure:
- Select the "Setup SCA Destinations" radio button

Common Event Infrastructure:
- Follow the instructions on the information center for more details.

Business management support:
- Select the "Business Rule Manager" check box

installed, then this is shown in the table. You may not uninstall an already installed and configured service by unchecking the selection.

If you configure a service on more than one host, then default configurations are applied if no configuration panel is shown or the configuration panel settings apply to the services on all hosts. If you select one application, then you are able to fine tune each aspect of your service configuration during this wizard session. Run the wizard for each service that you need to have differnt custom configurations.

Field help
For field help informatio select a field label or lis marker when the help cursor appears.

[✓ Select a cluster to configure] CEICluster01
Select a server to configure

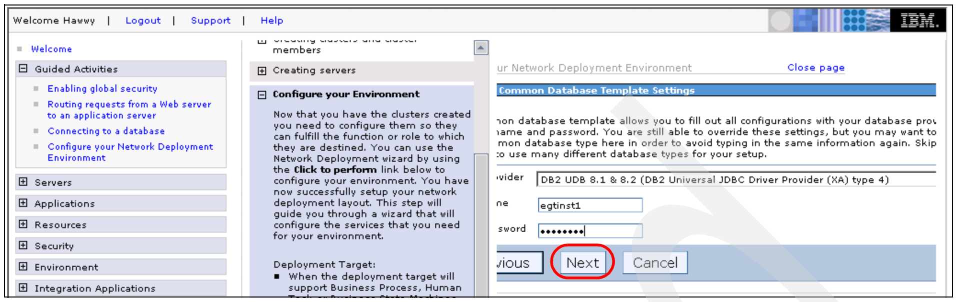
Select	Server/Cluster Name	Node	Setup SCA Destination	
			Local	Remote
☐	WESBCluster01		○	●
☐	MECluster01		●	○
☐	CEICluster01		○	●

Total 3

Figure 7-35 Configuring SCA for each cluster

3. Click **Next**.
4. From the JDBC provider drop-down list, select the appropriate provider. In this example, the provider is **DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC Driver Provider (XA) type 4)**.

5. Set the user name to `egtinst1` and the user password to `itso4you`, and click **Next**, as shown in Figure 7-36.


Welcome Hawwy | Logout | Support | Help

Welcome
Guided Activities
- Enabling global security
- Routing requests from a Web server to an application server
- Connecting to a database
- Configure your Network Deployment Environment

Servers
Applications
Resources
Security
Environment
Integration Applications

members
Creating servers
Configure your Environment

Now that you have the clusters created you need to configure them so they can fulfill the function or role to which they are destined. You can use the Network Deployment wizard by using the **Click to perform** link below to configure your environment. You have now successfully setup your network deployment layout. This step will guide you through a wizard that will configure the services that you need for your environment.

Deployment Target:
- When the deployment target will support Business Process, Human

ur Network Deployment Environment Close page

Common Database Template Settings

non database template allows you to fill out all configurations with your database prov name and password. You are still able to override these settings, but you may want to mon database type here in order to avoid typing in the same information again. Skip to use many different database types for your setup.

vider: DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC Driver Provider (XA) type 4)
ne: egtinst1
sword: ••••••••

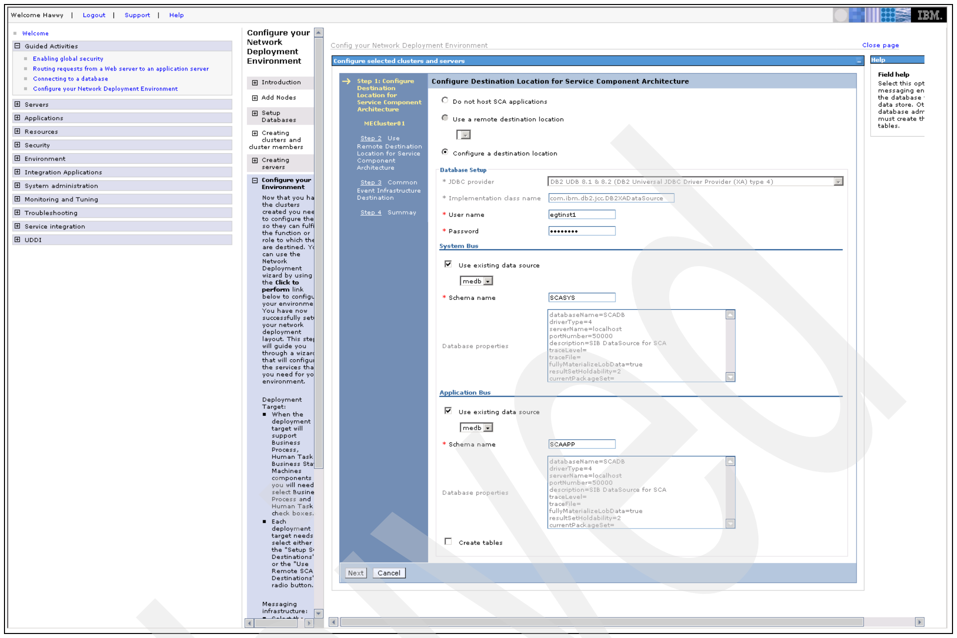
vious | Next | Cancel

Figure 7-36 JDBC provider selection

6. Under System Bus, **select Use existing data source**. In the drop-down menu below, select the data source **medb**.
7. Set the schema name to `SCASYS`.
8. Under Application Bus, select **Use existing data source**. In the drop-down menu below, select the data source **medb**.
9. Set the schema name to `SCAAPP`.
10. Clear **Create tables** because you have already created the tables manually with the `sibDDLgenerator.bat` command.

> **Note:** If the database is DB2, this option creates the SCA database schemas the first time that you start the SCA messaging engines.

Figure 7-37 shows these settings.



Config your Network Deployment Environment

Configure selected clusters and servers

Step 1: Configure Destination Location for Service Component Architecture

MECluster01

Configure Destination Location for Service Component Architecture

( ) Do not host SCA applications

( ) Use a remote destination location

(•) Configure a destination location

Database Setup

* JDBC provider	DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC Driver Provider (XA) type 4)
* Implementation class name	com.ibm.db2.jcc.DB2XADataSource
* User name	egtinst1
* Password	••••••••

System Bus

[✓] Use existing data source: medb

* Schema name	SCASYS

Database properties	databaseName=SCADB
	driverType=4
	serverName=localhost
	portNumber=50000
	description=SIB DataSource for SCA
	traceLevel=
	traceFile=
	fullyMaterializeLobData=true
	resultSetHoldability=2
	currentPackageSet=

Application Bus

[✓] Use existing data source: medb

* Schema name	SCAAPP

Database properties	databaseName=SCADB
	driverType=4
	serverName=localhost
	portNumber=50000
	description=SIB DataSource for SCA
	traceLevel=
	traceFile=
	fullyMaterializeLobData=true
	resultSetHoldability=2
	currentPackageSet=

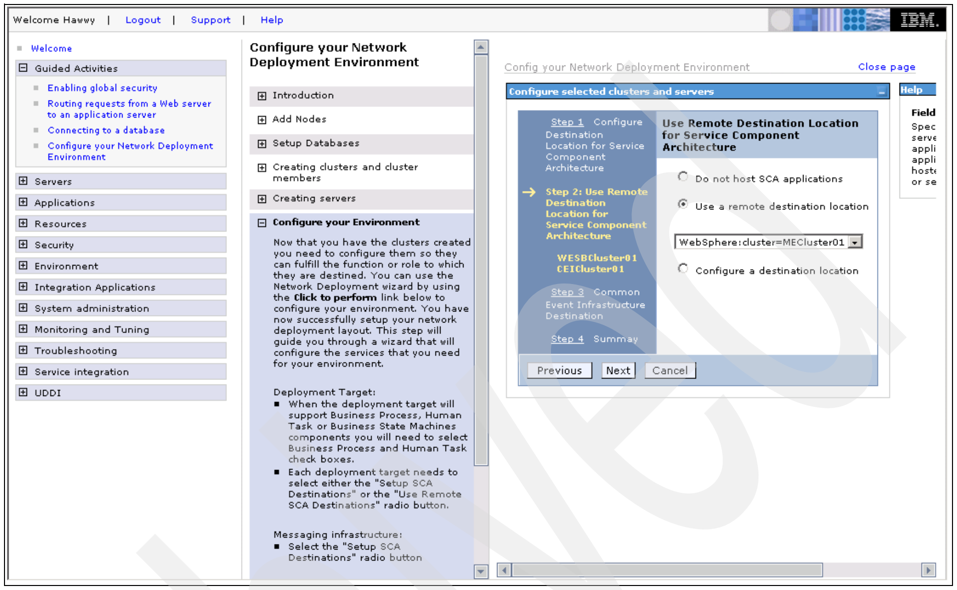
[ ] Create tables

Welcome Hawwy | Logout | Support | Help
Welcome
Guided Activities
Enabling global security
Routing requests from a Web server to an application server
Connecting to a database
Configure your Network Deployment Environment
Servers
Applications
Resources
Security
Environment
Integration Applications
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Configure your Network Deployment Environment
Introduction
Add Nodes
Setup Databases
Creating clusters and cluster members
Creating servers
Configure your Environment
Close page
Step 2 Use Remote Destination Location for Service Component Architecture
Step 3 Common Event Infrastructure Destination
Step 4 Summay
Help
Field help
Next
Cancel

Figure 7-37 SCA Values

11.Click **Next**.

12. This panel configures the WebSphere ESB and CEI clusters to use a remote destination for SCA. The default should set `MECluster01` because of the remote location, as shown in Figure 7-38.



Configure your Network Deployment Environment

Configure your Environment

Now that you have the clusters created you need to configure them so they can fulfill the function or role to which they are destined. You can use the Network Deployment wizard by using the **Click to perform** link below to configure your environment. You have now successfully setup your network deployment layout. This step will guide you through a wizard that will configure the services that you need for your environment.

Deployment Target:
- When the deployment target will support Business Process, Human Task or Business State Machines components you will need to select Business Process and Human Task check boxes.
- Each deployment target needs to select either the "Setup SCA Destinations" or the "Use Remote SCA Destinations" radio button.

Messaging infrastructure:
- Select the "Setup SCA Destinations" radio button

Config your Network Deployment Environment

Configure selected clusters and servers

Step 1 Configure Destination Location for Service Component Architecture
→ **Step 2: Use Remote Destination Location for Service Component Architecture**
WESBCluster01
CEICluster01
Step 3 Common Event Infrastructure Destination
Step 4 Summay

Use Remote Destination Location for Service Component Architecture

( ) Do not host SCA applications
[✓ (•) Use a remote destination location]
WebSphere:cluster=MECluster01
( ) Configure a destination location

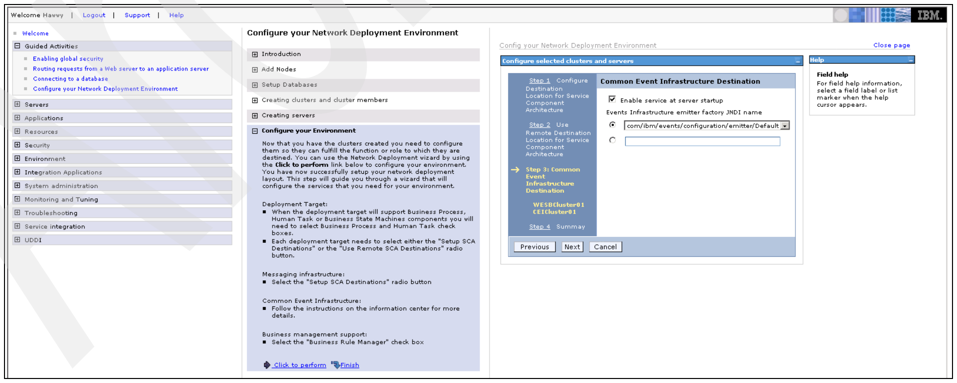
Help

Field Spec serve appli appli hoste or se

Figure 7-38 Remote SCA configuration

13. Click **Next** to move to the next CEI configuration panel.

14. Ensure that **Enable service at server startup** is selected, leave the other default values as is, and click **Next**, as shown in Figure 7-39.



Configure your Network Deployment Environment

Configure your Environment

Now that you have the clusters created you need to configure them so they can fulfill the function or role to which they are destined. You can use the Network Deployment wizard by using the **Click to perform** link below to configure your environment. You have now successfully setup your network deployment layout. This step will guide you through a wizard that will configure the services that you need for your environment.

Deployment Target:
- When the deployment target will support Business Process, Human Task or Business State Machines components you will need to select Business Process and Human Task check boxes.
- Each deployment target needs to select either the "Setup SCA Destinations" or the "Use Remote SCA Destinations" radio button.

Messaging infrastructure:
- Select the "Setup SCA Destinations" radio button

Common Event Infrastructure:
- Follow the instructions on the information center for more details.

Business management support:
- Select the "Business Rule Manager" check box

Config your Network Deployment Environment

Configure selected clusters and servers

Step 1 Configure Destination Location for Service Component Architecture
Step 2 Use Remote Destination Location for Service Component Architecture
→ **Step 3: Common Event Infrastructure Destination**
WESBCluster01
CEICluster01
Step 4 Summay

Common Event Infrastructure Destination

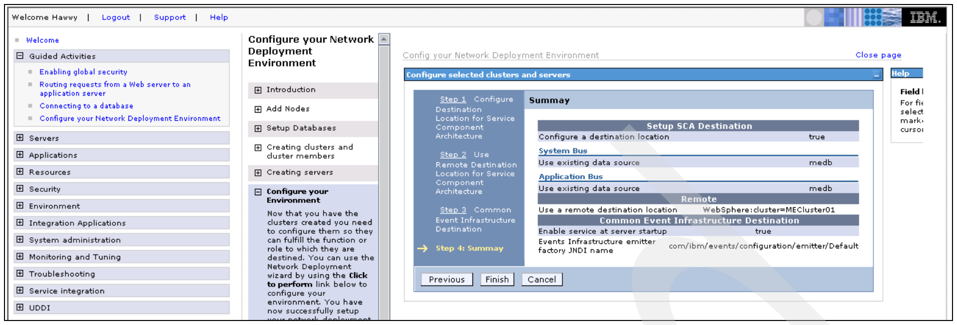
[✓] Enable service at server startup
Events Infrastructure emitter factory JNDI name
[✓ (•) com/ibm/events/configuration/emitter/Default]
( )

Help

Field help
For field help information, select a field label or list marker when the help cursor appears.

Figure 7-39 Startup CEI service settings

15. In the summary panel click **Finish** as shown in Figure 7-40.


Welcome Hawwy | Logout | Support | Help
Welcome
Guided Activities
Enabling global security
Routing requests from a Web server to an application server
Connecting to a database
Configure your Network Deployment Environment
Servers
Applications
Resources
Security
Environment
Integration Applications
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Configure your Network Deployment Environment
Introduction
Add Nodes
Setup Databases
Creating clusters and cluster members
Creating servers
Configure your Environment
Config your Network Deployment Environment
Close page
Configure selected clusters and servers
Help
Step 1 Configure Destination Location for Service Component Architecture
Step 2 Use Remote Destination Location for Service Component Architecture
Step 3 Common Event Infrastructure Destination
Step 4: Summay
Previous
Finish
Cancel

Now that you have the clusters created you need to configure them so they can fulfill the function or role to which they are destined. You can use the Network Deployment wizard by using the **Click to perform** link below to configure your environment. You have now successfully setup

Summay

Setup SCA Destination	
Configure a destination location	true
System Bus	
Use existing data source	medb
Application Bus	
Use existing data source	medb
Remote	
Use a remote destination location	WebSphere:cluster=MECluster01
Common Event Infrastructure Destination	
Enable service at server startup	true
Events Infrastructure emitter factory JNDI name	com/ibm/events/configuration/emitter/Default

Figure 7-40 Cluster configuration summary

16. The network deployment environment is now configured. When it completes save and synchronize the changes.

7.3.3 Configuring CEI for DB2

This section tells you how to create the CEI database (CEIDB) and how to configure the cell that is associated with CEI. In this example, you create the database manually by transferring the creation scripts to the DB2 server.

Generating the CEI database script

You need to generate the scripts to create the CEI database and the associated cell configuration scripts manually. Follow these steps:

1. Open a command prompt on the cell deployment manager system.
2. Change directory to %*WAS_HOME*%\profiles\Dmgr01\event\dbconfig. In this example, the directory is C:\IBM\WebSphere\ESB\profiles\Dmgr01\event\dbconfig.
3. Copy the file DB2ResponseFile.txt to NewDB2ResponseFile.txt.

4. Using an appropriate editor, edit NewDB2ResponseFile.txt with the correct information for the environment. Table 7-10 lists the changes for this example.

Table 7-10 NewDB2Responsfile.txt settings

Property	Value
SCOPE	cell
DB_NAME	CEIDB
DB_NODE_NAME	db2 (look in db2nodes.cfg on the DB2 server)
JDBC_CLASSPATH	c:\db2client
UNIVERSAL_JDBC_CLASSPATH	c:\db2client
JDBC_DRIVER_TYPE	4
DB2_HOST_NAME	db2.itso
DB2_INSTANCE_PORT	50000
EXECUTE_SCRIPT	NO

5. Save the new configuration file.
6. From the same directory, run the following script to generate the CEI configuration scripts:

   ```
   config_event_database.bat NewDB2ResponseFile.txt
   ```

The scripts to configure the cell for CEI are created in the directory %*WAS_HOME*%\profiles\Dmgr01\event\dsscripts\db2. In this example, the directory is C:\IBM\WebSphere\ESB\profiles\Dmgr01\event\dsscripts\db2.

The C:\IBM\WebSphere\ESB\profiles\Dmgr01\event\dbscripts\db2 directory is also generated, and it includes the CEIDB database creation scripts.

Creating the CEI database

This section describes the steps that are necessary to create the DB2 database, CEIDB, on the database server. You need to transfer the scripts from a WebSphere ESB server to the DB2 server because the DB2 windows client is not installed on the members of the cell. It is assumed that the instance is running, in this case the instance name is `egtinst1`.

To create the CEI database, follow these steps:

1. Transfer the CEIDB creation scripts directory for DB2, %*WAS_HOME*%\profiles\Dmgr01\event\dbscripts\db2, to the database server. In this example, the directory is C:\IBM\WebSphere\ESB\profiles\Dmgr01\event\dbscripts\db2.

   > **Attention:** If you transfer these files from Windows to UNIX, there might be return characters (`\r`) at the end of each line. To run the DDL scripts, you must remove these characters. A UNIX utility such as `dos2unix` will remove these characters.

2. Login to the DB2 server `db2.itso` as the instance owner `egtinst1`.
3. Change directory to the /home/egtinst1/db2.
4. Run the CEIDB creation script:

   ```
   ./cr_event_db2.sh
   ```

Table 7-11 lists the command line responses.

Table 7-11 cr_event_db2.sh responses

Response	Description
1	script is running on the DB2 server
egtinst1	DB2 user

This series of steps creates the database and appropriate tables.

Configuring the CEI database cell

To configure the cell for connection to the CEIDB database:

1. On the cell deployment manager where the scripts were generated, open a command line window.
2. Run the JDBC configuration script. In this example, we are running on Windows and the database is DB2. Therefore, the following commands configure the cell:

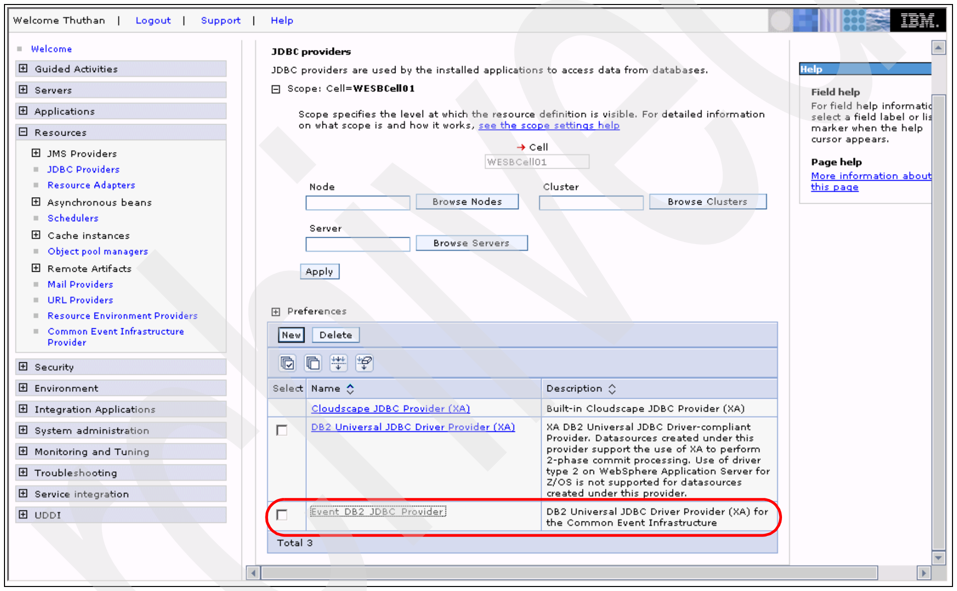
   ```
   cd C:\IBM\WebSphere\ESB\profiles\Dmgr01\event\dsscripts\db2
   cr_db2_jdbc_provider.bat cell
   ```

Table 7-12 shows the command line responses to this script.

Table 7-12 CEI JDBC Provider responses

Response	Description
egtinst1	DB2 user for CEIDB
itso4you	DB2 password for CEIDB

3. Open the Administrative Console, log in, and navigate to **Resources** → **JDBC Providers**. You should see a JDBC provider at the cell scope for CEI called `Event_DB2_JDBC_Provider`, as shown in Figure 7-41.



JDBC providers

JDBC providers are used by the installed applications to access data from databases.

Scope: Cell=**WESBCell01**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help

Cell: WESBCell01

Node:

Cluster:

Server:

Select	Name	Description
	Cloudscape JDBC Provider (XA)	Built-in Cloudscape JDBC Provider (XA)
☐	DB2 Universal JDBC Driver Provider (XA)	XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on WebSphere Application Server for Z/OS is not supported for datasources created under this provider.
☐	Event DB2 JDBC Provider	DB2 Universal JDBC Driver Provider (XA) for the Common Event Infrastructure

Total 3

Field help

For field help information, select a field label or list marker when the help cursor appears.

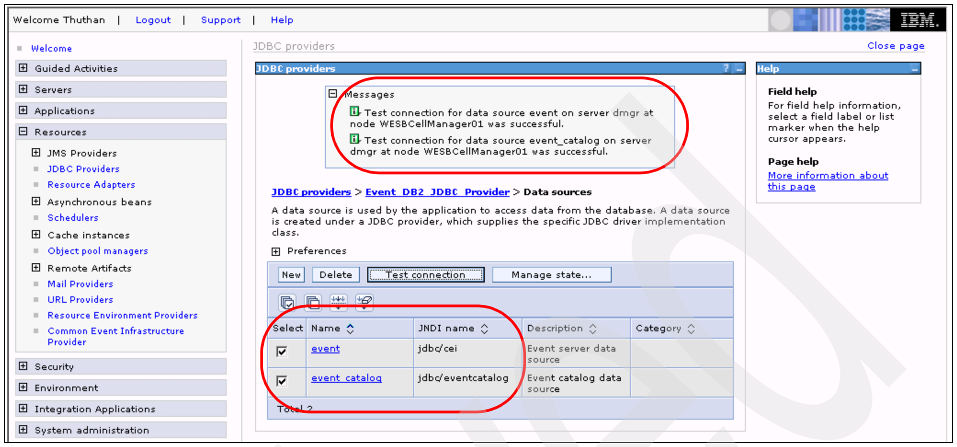
Page help

More information about this page

Figure 7-41 CEI JDBC provider

4. Click the **Event_DB2_JDBC_Provider** that is highlighted in Figure 7-41.
5. In the next panel, scroll right, and under **Additional Properties** click **Data sources**.

6. From the data sources panel, select the **event** and **event_catalog** datasources and click **Test connection**, as shown in Figure 7-42.



Messages

Test connection for data source event on server dmgr at node WESBCellManager01 was successful.

Test connection for data source event_catalog on server dmgr at node WESBCellManager01 was successful.

JDBC providers > Event_DB2_JDBC_Provider > Data sources

A data source is used by the application to access data from the database. A data source is created under a JDBC provider, which supplies the specific JDBC driver implementation class.

Preferences

New | Delete | Test connection | Manage state...

Select	Name	JNDI name	Description	Category
✓	event	jdbc/cei	Event server data source	
✓	event_catalog	jdbc/eventcatalog	Event catalog data source	
Total 2				

Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

Figure 7-42 Successful database connections

The Messages dialog box (at the top in Figure 7-42) should indicate that the data sources successfully connected to the database.

7.3.4 Deploying the CEI applications

This section explains how install the CEI applications into the `CEICluster01` cluster.

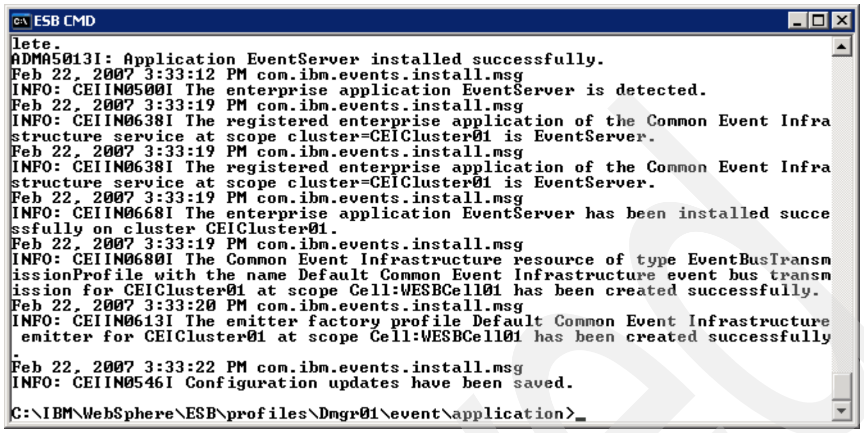
Installing the CEI Event Server application

To install the CEI Event Server application:

1. Open a command prompt on the deployment manager.
2. Change directory to %*WAS_HOME*%\profiles\Dmgr01\event\application. In this example, this directory is C:\IBM\WebSphere\ESB\profiles\Dmgr01\event\application.
3. Execute the command to install the CEI Event Server application. For example:

```
..\..\bin\wsadmin -f event-application.jacl -profile
event-profile.jacl -action install -earfile event-application.ear
-backendid DB2UDBNT_V82_1 -cluster CEICluster01
```

The final line of all the responses should say `Configuration updates have been saved`. Figure 7-43 shows example output.



```
lete.
ADMA5013I: Application EventServer installed successfully.
Feb 22, 2007 3:33:12 PM com.ibm.events.install.msg
INFO: CEIIN0500I The enterprise application EventServer is detected.
Feb 22, 2007 3:33:19 PM com.ibm.events.install.msg
INFO: CEIIN0638I The registered enterprise application of the Common Event Infra
structure service at scope cluster=CEICluster01 is EventServer.
Feb 22, 2007 3:33:19 PM com.ibm.events.install.msg
INFO: CEIIN0638I The registered enterprise application of the Common Event Infra
structure service at scope cluster=CEICluster01 is EventServer.
Feb 22, 2007 3:33:19 PM com.ibm.events.install.msg
INFO: CEIIN0668I The enterprise application EventServer has been installed succe
ssfully on cluster CEICluster01.
Feb 22, 2007 3:33:19 PM com.ibm.events.install.msg
INFO: CEIIN0680I The Common Event Infrastructure resource of type EventBusTransm
issionProfile with the name Default Common Event Infrastructure event bus transm
ission for CEICluster01 at scope Cell:WESBCell01 has been created successfully.
Feb 22, 2007 3:33:20 PM com.ibm.events.install.msg
INFO: CEIIN0613I The emitter factory profile Default Common Event Infrastructure
 emitter for CEICluster01 at scope Cell:WESBCell01 has been created successfully
.
Feb 22, 2007 3:33:22 PM com.ibm.events.install.msg
INFO: CEIIN0546I Configuration updates have been saved.

C:\IBM\WebSphere\ESB\profiles\Dmgr01\event\application>_
```

Figure 7-43 CEI application deployment responses

Adding asynchronous capabilities to the CEI Event Server application

To add asynchronous capabilities to the CEI Event Server application:

1. Open a command prompt on the deployment manager.
2. Change directory to %*WAS_HOME*%\profiles\Dmgr01\event\application. In this example, this directory is C:\IBM\WebSphere\ESB\profiles\Dmgr01\event\application.
3. Execute the command to install the CEI MDB application. For example:

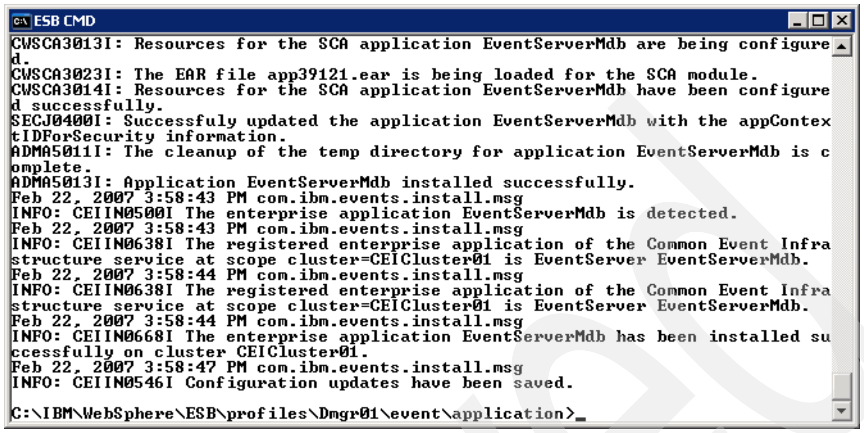
```
..\..\bin\wsadmin -f default-event-message.jacl -profile
event-profile.jacl -earfile event-message.ear -action install
-cluster CEICluster01
```

Table 7-13 shows the command line responses to this script.

Table 7-13 CEI MDB application responses

Response	Description
sca	embedded messaging authentication user
itso4you	embedded messaging authentication password
itso4you	verified embedded messaging authentication password

The final line of the responses should say `Configuration updates have been saved`. Figure 7-43 shows example output.



```
CWSCA3013I: Resources for the SCA application EventServerMdb are being configure
d.
CWSCA3023I: The EAR file app39121.ear is being loaded for the SCA module.
CWSCA3014I: Resources for the SCA application EventServerMdb have been configure
d successfully.
SECJ0400I: Successfuly updated the application EventServerMdb with the appContex
tIDForSecurity information.
ADMA5011I: The cleanup of the temp directory for application EventServerMdb is c
omplete.
ADMA5013I: Application EventServerMdb installed successfully.
Feb 22, 2007 3:58:43 PM com.ibm.events.install.msg
INFO: CEIIN0500I The enterprise application EventServerMdb is detected.
Feb 22, 2007 3:58:43 PM com.ibm.events.install.msg
INFO: CEIIN0638I The registered enterprise application of the Common Event Infra
structure service at scope cluster=CEICluster01 is EventServer EventServerMdb.
Feb 22, 2007 3:58:44 PM com.ibm.events.install.msg
INFO: CEIIN0638I The registered enterprise application of the Common Event Infra
structure service at scope cluster=CEICluster01 is EventServer EventServerMdb.
Feb 22, 2007 3:58:44 PM com.ibm.events.install.msg
INFO: CEIIN0668I The enterprise application EventServerMdb has been installed su
ccessfully on cluster CEICluster01.
Feb 22, 2007 3:58:47 PM com.ibm.events.install.msg
INFO: CEIIN0546I Configuration updates have been saved.

C:\IBM\WebSphere\ESB\profiles\Dmgr01\event\application>_
```

Figure 7-44 CEI MEDB application deployment response

Modifying the CEI configuration

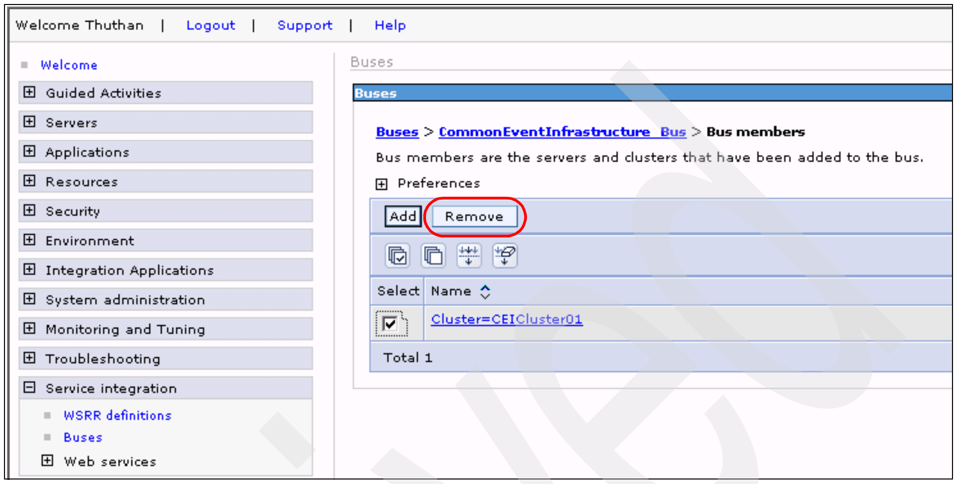
The default configuration for CEI assumes local messaging engines. In this topology, the messaging engines are located on the `MECluster01` cluster members.

Deleting bus member CEICluster01

To delete `CEICluster01`:

1. Open the Administrative Console and login.
2. Navigate to **Service integration** → **Buses**.
3. Click the **CommonEventInfrastructure_Bus** bus.
4. Under the **Topology** heading, click **Bus members**.

5. `CEICluster01` is a member of the bus. However, in the case of a remote messaging environment, this member name should be `MECluster01`. Select **Cluster=CEICluster01** and click **Remove** as shown in Figure 7-45.



Bus members are the servers and clusters that have been added to the bus.

Select	Name
✓	Cluster=CEICluster01
Total 1	

Figure 7-45 Removing CEICluster01 from the bus

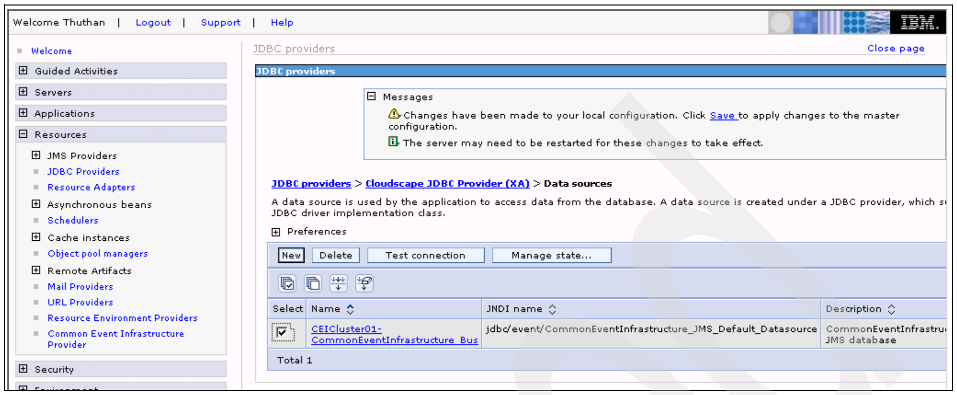
6. Save and synchronize the changes, and click **OK** when complete.

Deleting the Cloudscape data source

To delete the Cloudscape™ data source, follow these steps:

1. Navigate to **Resources** → **JDBC Providers**.
2. Set the scope to the CEICluster01 cluster, and click **Apply**.
3. Click **Cloudscape JDBC Provider (XA)**.
4. Under **Additional Properties**, click **Data sources**.

5. Select **CEICluster01-CommonEventInfrastructure_Bus** and click **Delete**, as shown in Figure 7-46.



Welcome Thuthan | Logout | Support | Help

JDBC providers — Close page

JDBC providers

Messages

⚠ Changes have been made to your local configuration. Click Save to apply changes to the master configuration.

The server may need to be restarted for these changes to take effect.

JDBC providers > Cloudscape JDBC Provider (XA) > Data sources

A data source is used by the application to access data from the database. A data source is created under a JDBC provider, which s JDBC driver implementation class.

Select	Name	JNDI name	Description
✓	CEICluster01-CommonEventInfrastructure_Bus	jdbc/event/CommonEventInfrastructure_JMS_Default_Datasource	CommonEventInfrastru JMS database

Total 1

Figure 7-46 Deletion of the CEICluster01-CommonEventInfrastructure_Bus data source

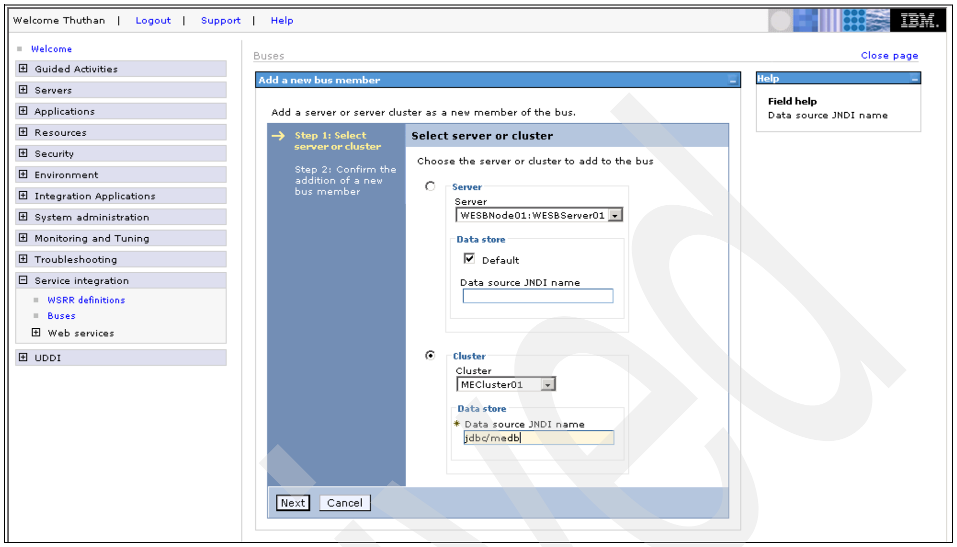
6. Save and synchronize the changes, and click **OK** when complete.

Adding MECluster01 to the CEI bus

To add `MECluster01` to the CEI bus:

1. Navigate to **Service integration** → **Buses**.
2. Click CommonEventInfrastructure_Bus.
3. Under **Topology**, click **Bus members**.
4. Click **Add**.

5. Select **Cluster**, select **MECluster01**, and set the **Data source JNDI name** to `jdbc/medb`, as shown in Figure 7-47.



Add a new bus member

Add a server or server cluster as a new member of the bus.

Step 1: Select server or cluster

Step 2: Confirm the addition of a new bus member

Select server or cluster

Choose the server or cluster to add to the bus

( ) Server
Server: WESBNode01:WESBServer01
Data store
☑ Default
Data source JNDI name:

(•) Cluster
Cluster: MECluster01
Data store
* Data source JNDI name: jdbc/medb

Figure 7-47 Adding the MECluster01 bus member

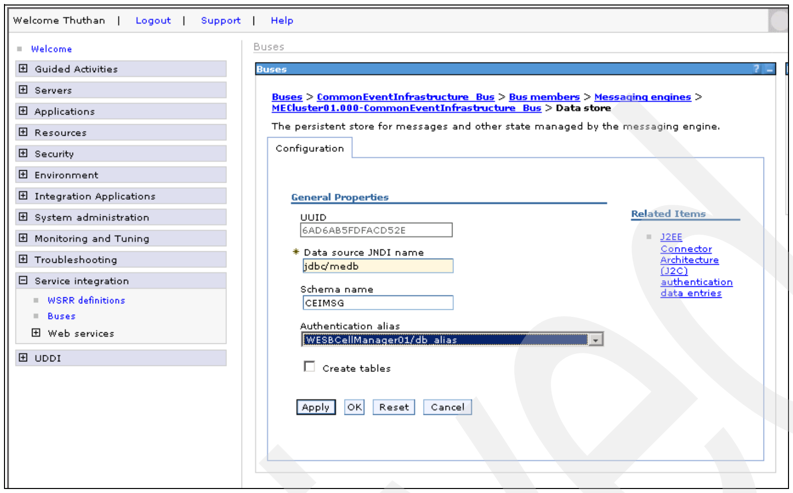
6. Click **Next**.
7. In the confirmation panel, click **Finish**.
8. Save and synchronize the changes, and click **OK** when complete.

Setting the CEIMSG schema

To set the CEIMSG schema:

1. Navigate to **Service integration** → **Buses**.
2. Click **CommonEventInfrastructure_Bus**.
3. Under **Topology**, click **Bus members**.
4. Click bus member **Cluster=MECluster01**.
5. Click the messaging engine **`MECluster01.000-CommonEventInfrastructure_Bus`**.
6. Under Additional Properties, click **Data store**.

7. Set the Schema name to `CEIMSG`, and select **WESBCellManager01/db_alias** from the Authentication alias drop-down list, as shown in Figure 7-48.



Buses > CommonEventInfrastructure_Bus > Bus members > Messaging engines > MECluster01.000-CommonEventInfrastructure_Bus > Data store

The persistent store for messages and other state managed by the messaging engine.

UUID: 6AD6AB5FDFACD52E

* Data source JNDI name: jdbc/medb

Schema name: CEIMSG

Authentication alias: WESBCellManager01/db_alias

☐ Create tables

Figure 7-48 Schema configuration for CEI messaging engine

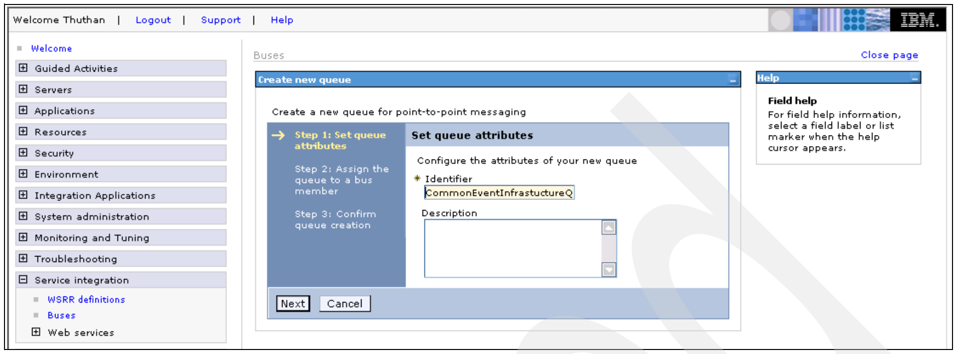
8. Clear the Create tables check box.
9. Click **OK**.
10. Save and synchronize the changes, and click **OK** when complete.

Setting the JMS destinations

To set the JMS destination, follow these steps:

1. Navigate to **Service integration** → **Buses**.
2. Click **CommonEventInfrastructure_Bus**.
3. Under Destination resources, click **Destinations**.
4. Click **New**.
5. Set the destination type to `Queue`, and click **Next**.

6. Set the Identifier to `CommonEventInfrastructureQueueDestination` and click **Next**, as shown in Figure 7-49.


Welcome Thuthan | Logout | Support | Help

Create new queue

Create a new queue for point-to-point messaging

Step 1: Set queue attributes
Step 2: Assign the queue to a bus member
Step 3: Confirm queue creation

Set queue attributes

Configure the attributes of your new queue

Identifier: CommonEventInfrastuctureQ

Description

Help

Field help

For field help information, select a field label or list marker when the help cursor appears.

Figure 7-49   Setting the CEI queue attributes

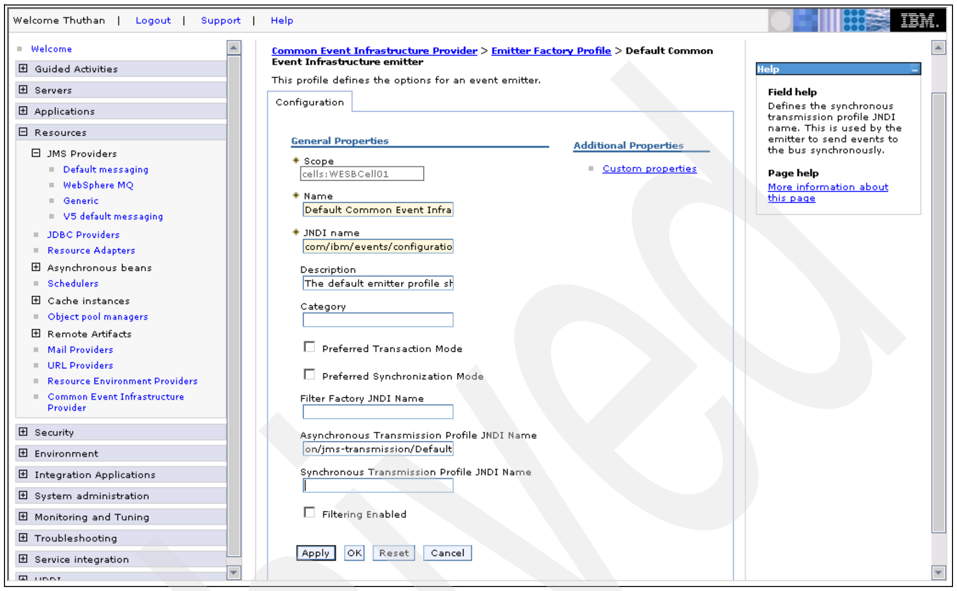
7. Select **MECluster01** from the Bus member drop-down list, and click **Next**.
8. At the confirmation window, click **Finish**.
9. Click **New**.
10. Set the destination type to *Topic space*, and click **Next**.
11. Set the Identifier to `CommonEventInfrastructureTopicDestination`, and click **Next**.
12. At the confirmation window, click **Finish**.
13. Save and synchronize the changes, and click **OK** when complete.

Setting asynchronous event emission

To set asynchronous event emission, follow these steps:

1. Navigate to **Resources** → **Common Event Infrastructure Provider**.
2. Set the scope to the cell, and click **Apply**.
3. Under Additional Properties, click **Emitter Factory Profile**.
4. Click **Default Common Event Infrastructure emitter**.

5. Clear the Preferred synchronization mode check box and clear the Synchronous Transmission Profile JNDI Name entry, as shown in Figure 7-50.


Welcome Thuthan | Logout | Support | Help

- Welcome
- ⊞ Guided Activities
- ⊞ Servers
- ⊞ Applications
- ⊟ Resources
  - ⊟ JMS Providers
    - Default messaging
    - WebSphere MQ
    - Generic
    - V5 default messaging
  - JDBC Providers
  - Resource Adapters
  - ⊞ Asynchronous beans
  - Schedulers
  - ⊞ Cache instances
  - Object pool managers
  - ⊞ Remote Artifacts
  - Mail Providers
  - URL Providers
  - Resource Environment Providers
  - Common Event Infrastructure Provider
- ⊞ Security
- ⊞ Environment
- ⊞ Integration Applications
- ⊞ System administration
- ⊞ Monitoring and Tuning
- ⊞ Troubleshooting
- ⊞ Service integration
- ⊞ UDDI

Common Event Infrastructure Provider > Emitter Factory Profile > Default Common Event Infrastructure emitter

This profile defines the options for an event emitter.

Configuration

General Properties

* Scope: cells:WESBCell01

* Name: Default Common Event Infra

* JNDI name: com/ibm/events/configuratio

Description: The default emitter profile sh

Category:

☐ Preferred Transaction Mode

☐ Preferred Synchronization Mode

Filter Factory JNDI Name:

Asynchronous Transmission Profile JNDI Name: on/jms-transmission/Default

Synchronous Transmission Profile JNDI Name:

☐ Filtering Enabled

Apply | OK | Reset | Cancel

Additional Properties

- Custom properties

Help

Field help

Defines the synchronous transmission profile JNDI name. This is used by the emitter to send events to the bus synchronously.

Page help

More information about this page

Figure 7-50 Configuration of synchronous mode

6. Click **OK**.
7. Click **Default Common Event Infrastructure emitter for CEICluster01** emitter.
8. Clear **Preferred Synchronization Mode**.
9. Click **OK**.
10. Save and synchronize the changes, and click **OK** when completed.

You have now constructed the topology that is shown in Figure 7-51.

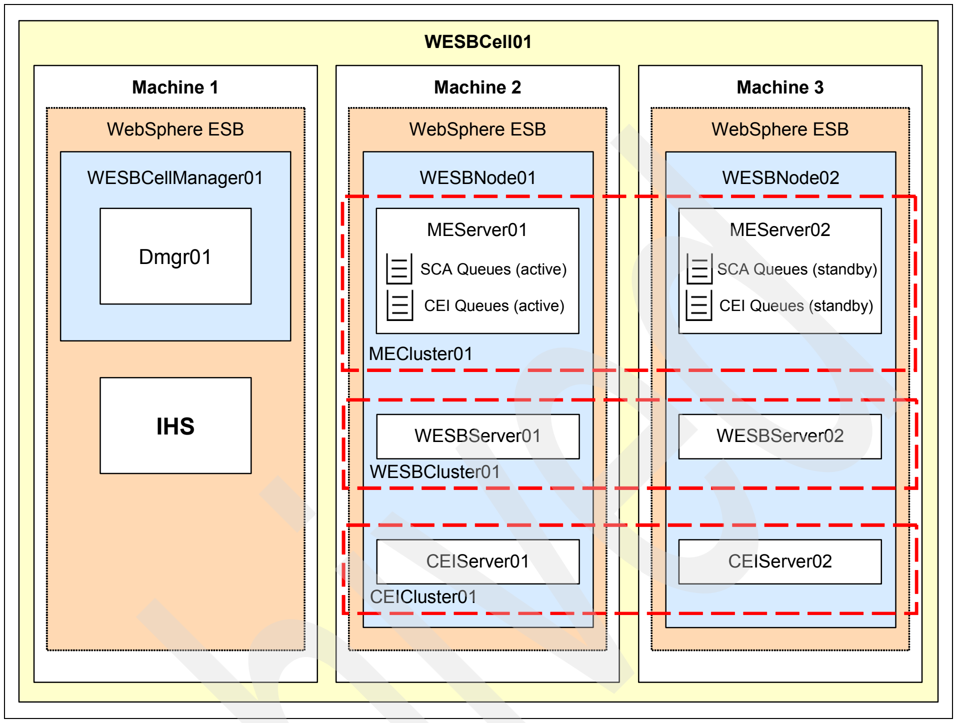


Figure 7-51 Clusters with SCA queues

7.3.5 Verifying the topology

In this section, you verify the database schema and start the cluster.

Checking the messaging engine database schemas

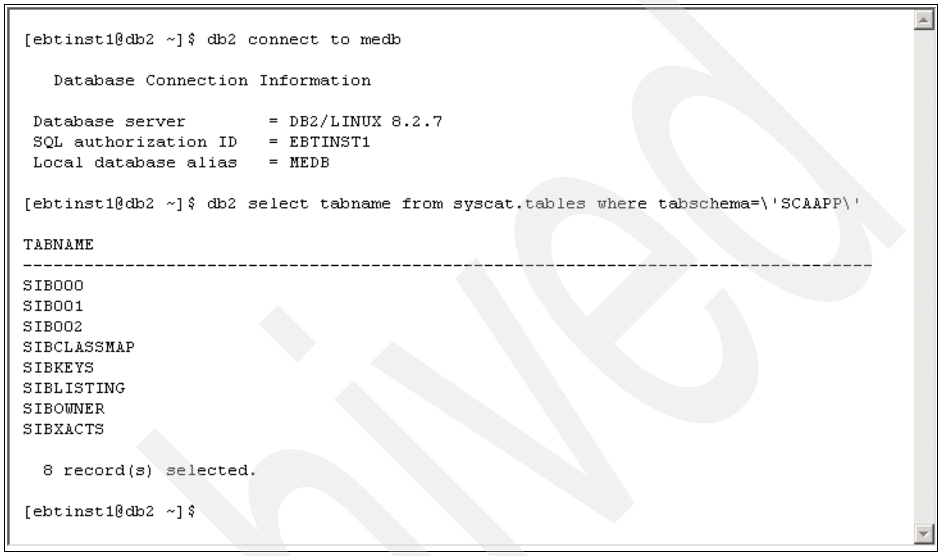
In this example, DB2 is the RDBMS. To check the database schema, follow these steps:

1. Log in to the database server as the instance user. In this example this is `egtinst1`.
2. Run the following commands:

```
db2 connect to MEDB
db2 select tabname from syscat.tables where tabschema=\'SCAAPP\'
db2 select tabname from syscat.tables where tabschema=\'SCASYS\'
db2 select tabname from syscat.tables where tabschema=\'CEIMSG\'
```

Note: On Windows platforms, do not use the backslashes (\) with the **`db2 select`** command.

Each of the schemas should have eight tables. Figure 7-52 shows example output for the SCAAPP schema.



```
[ebtinst1@db2 ~]$ db2 connect to medb

   Database Connection Information

 Database server        = DB2/LINUX 8.2.7
 SQL authorization ID   = EBTINST1
 Local database alias   = MEDB

[ebtinst1@db2 ~]$ db2 select tabname from syscat.tables where tabschema=\'SCAAPP\'

TABNAME
-----------------------------------------------------------------------------------
SIB000
SIB001
SIB002
SIBCLASSMAP
SIBKEYS
SIBLISTING
SIBOWNER
SIBXACTS

  8 record(s) selected.

[ebtinst1@db2 ~]$
```

Figure 7-52 SCAAPP schema

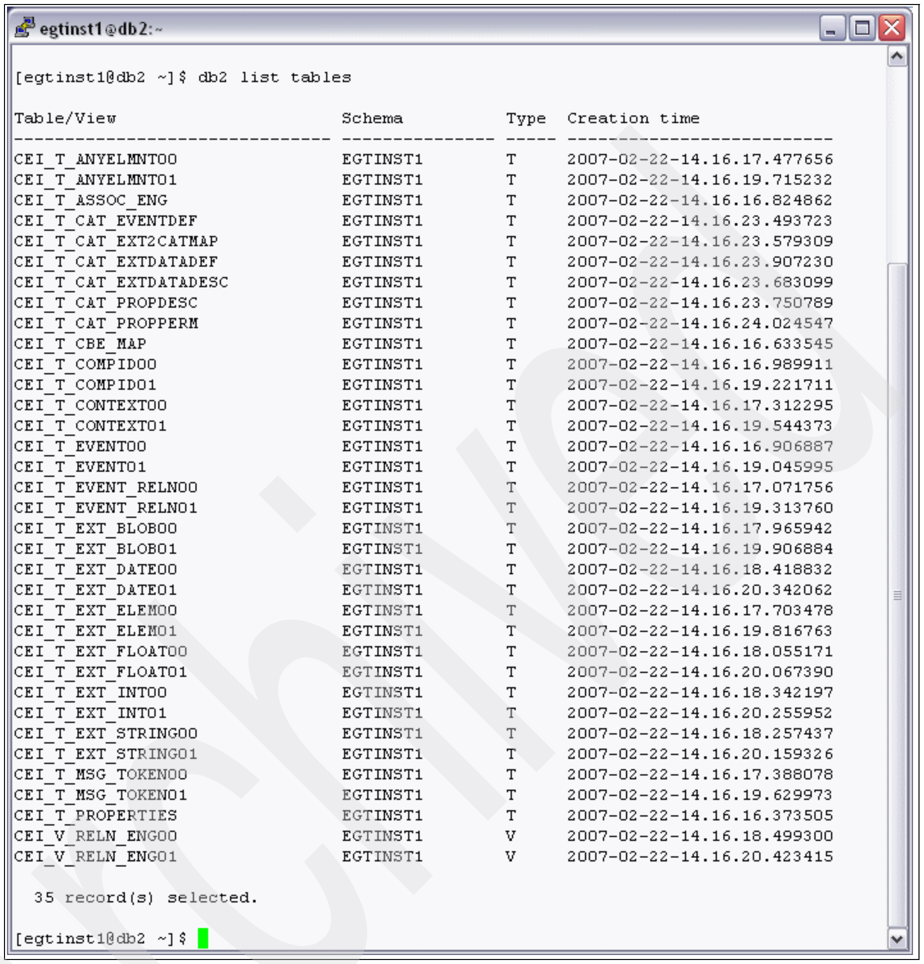
Checking the CEI database schemas

In this example, DB2 is the RDBMS. To check the CEI database schemas, follow these steps:

1. Log in to the database server as the instance user. In this example this is `egtinst1`.
2. Run the following commands:

```
db2 connect to CEIDB
db2 list tables
```

There are 35 tables. Figure 7-53 shows example output.



```
[egtinst1@db2 ~]$ db2 list tables

Table/View                      Schema          Type  Creation time
------------------------------- --------------- ----- --------------------------
CEI_T_ANYELMNT00                EGTINST1        T     2007-02-22-14.16.17.477656
CEI_T_ANYELMNT01                EGTINST1        T     2007-02-22-14.16.19.715232
CEI_T_ASSOC_ENG                 EGTINST1        T     2007-02-22-14.16.16.824862
CEI_T_CAT_EVENTDEF              EGTINST1        T     2007-02-22-14.16.23.493723
CEI_T_CAT_EXT2CATMAP            EGTINST1        T     2007-02-22-14.16.23.579309
CEI_T_CAT_EXTDATADEF            EGTINST1        T     2007-02-22-14.16.23.907230
CEI_T_CAT_EXTDATADESC           EGTINST1        T     2007-02-22-14.16.23.683099
CEI_T_CAT_PROPDESC              EGTINST1        T     2007-02-22-14.16.23.750789
CEI_T_CAT_PROPPERM              EGTINST1        T     2007-02-22-14.16.24.024547
CEI_T_CBE_MAP                   EGTINST1        T     2007-02-22-14.16.16.633545
CEI_T_COMPID00                  EGTINST1        T     2007-02-22-14.16.16.989911
CEI_T_COMPID01                  EGTINST1        T     2007-02-22-14.16.19.221711
CEI_T_CONTEXT00                 EGTINST1        T     2007-02-22-14.16.17.312295
CEI_T_CONTEXT01                 EGTINST1        T     2007-02-22-14.16.19.544373
CEI_T_EVENT00                   EGTINST1        T     2007-02-22-14.16.16.906887
CEI_T_EVENT01                   EGTINST1        T     2007-02-22-14.16.19.045995
CEI_T_EVENT_RELN00              EGTINST1        T     2007-02-22-14.16.17.071756
CEI_T_EVENT_RELN01              EGTINST1        T     2007-02-22-14.16.19.313760
CEI_T_EXT_BLOB00                EGTINST1        T     2007-02-22-14.16.17.965942
CEI_T_EXT_BLOB01                EGTINST1        T     2007-02-22-14.16.19.906884
CEI_T_EXT_DATE00                EGTINST1        T     2007-02-22-14.16.18.418832
CEI_T_EXT_DATE01                EGTINST1        T     2007-02-22-14.16.20.342062
CEI_T_EXT_ELEM00                EGTINST1        T     2007-02-22-14.16.17.703478
CEI_T_EXT_ELEM01                EGTINST1        T     2007-02-22-14.16.19.816763
CEI_T_EXT_FLOAT00               EGTINST1        T     2007-02-22-14.16.18.055171
CEI_T_EXT_FLOAT01               EGTINST1        T     2007-02-22-14.16.20.067390
CEI_T_EXT_INT00                 EGTINST1        T     2007-02-22-14.16.18.342197
CEI_T_EXT_INT01                 EGTINST1        T     2007-02-22-14.16.20.255952
CEI_T_EXT_STRING00              EGTINST1        T     2007-02-22-14.16.18.257437
CEI_T_EXT_STRING01              EGTINST1        T     2007-02-22-14.16.20.159326
CEI_T_MSG_TOKEN00               EGTINST1        T     2007-02-22-14.16.17.388078
CEI_T_MSG_TOKEN01               EGTINST1        T     2007-02-22-14.16.19.629973
CEI_T_PROPERTIES                EGTINST1        T     2007-02-22-14.16.16.373505
CEI_V_RELN_ENG00                EGTINST1        V     2007-02-22-14.16.18.499300
CEI_V_RELN_ENG01                EGTINST1        V     2007-02-22-14.16.20.423415

  35 record(s) selected.

[egtinst1@db2 ~]$
```

Figure 7-53 CEIDB tables

Starting the messaging cluster

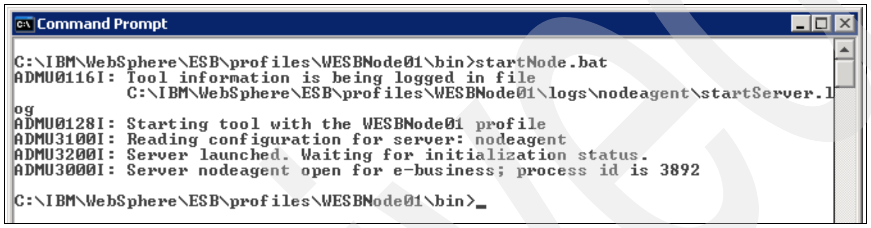
To start the messaging cluster, follow these steps:

1. Ensure that all nodes are running. If a node is not running, start it with the following commands:
   a. Open a command prompt on the node's host machine.
   b. Change directory to the node's bin directory:

      ```
      cd \IBM\WebSphere\ESB\profiles\WESBNode01\bin
      ```

   c. Start the node using this command (Figure 7-54):

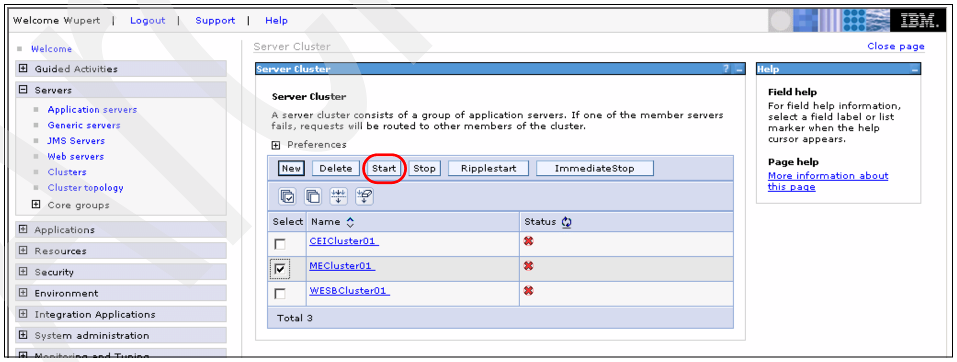
      ```
      startNode.bat
      ```



```
C:\IBM\WebSphere\ESB\profiles\WESBNode01\bin>startNode.bat
ADMU0116I: Tool information is being logged in file
           C:\IBM\WebSphere\ESB\profiles\WESBNode01\logs\nodeagent\startServer.l
og
ADMU0128I: Starting tool with the WESBNode01 profile
ADMU3100I: Reading configuration for server: nodeagent
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server nodeagent open for e-business; process id is 3892

C:\IBM\WebSphere\ESB\profiles\WESBNode01\bin>_
```

Figure 7-54 Starting a node at the command line

2. Start the cluster:
   a. Open the Administrative Console and login.
   b. Navigate to **Servers** → **Clusters**.
   c. Select **MECluster01** and click **Start**, as shown in Figure 7-55.



Server Cluster

A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster.

⊞ Preferences

Select	Name	Status
☐	CEICluster01	✖
☑	MECluster01	✖
☐	WESBCluster01	✖

Total 3

Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

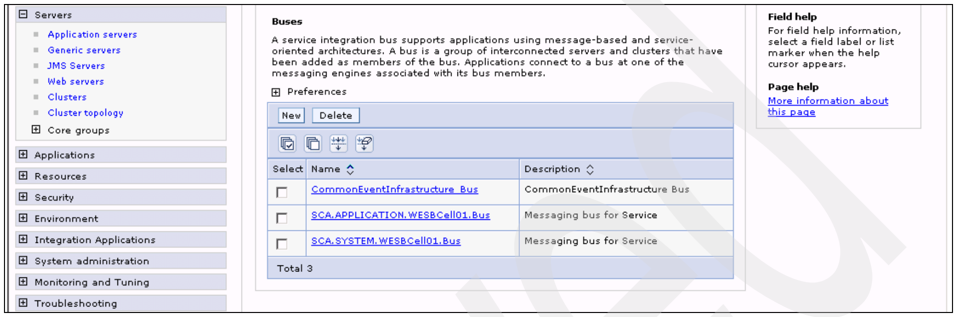
Figure 7-55 Starting the MECluster01 cluster

   d. Verify that the cluster starts successfully. The red cross should change to a solid green arrow. You need to refresh the view to see the change in status.

3. Verify that the messaging engines have started:

   a. Navigate to **Service integration → Buses**.

      There should be three buses listed—two for the SCA Application and SCA System buses and a third CommonEventInfrastructure bus—as shown in Figure 7-56.



- Servers
  - Application servers
  - Generic servers
  - JMS Servers
  - Web servers
  - Clusters
  - Cluster topology
  - Core groups
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting

Buses

A service integration bus supports applications using message-based and service-oriented architectures. A bus is a group of interconnected servers and clusters that have been added as members of the bus. Applications connect to a bus at one of the messaging engines associated with its bus members.

Preferences

New Delete

Select	Name	Description
☐	CommonEventInfrastructure_Bus	CommonEventInfrastructure Bus
☐	SCA.APPLICATION.WESBCell01.Bus	Messaging bus for Service
☐	SCA.SYSTEM.WESBCell01.Bus	Messaging bus for Service
Total 3		

Field help
For field help information, select a field label or list marker when the help cursor appears.

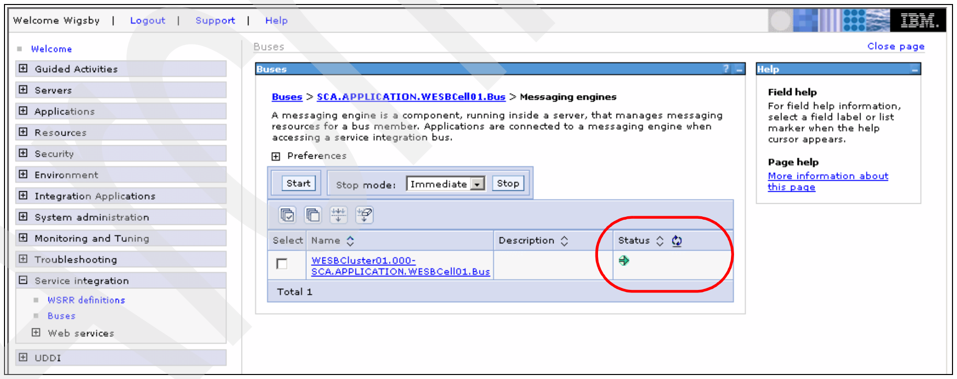
Page help
More information about this page

Figure 7-56 Service integration buses

   b. Click **SCA.APPLICATION.WESBCell01.Bus**.

   c. Under the Topology section, click **Messaging engines**.

   d. The status of the `WESBCluster01.000-SCA.APPLICATION.WESBCell01.Bus` messaging engine should be started, as shown in Figure 7-57.



Welcome Wigsby | Logout | Support | Help

- Welcome
- Guided Activities
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
  - WSRR definitions
  - Buses
  - Web services
- UDDI

Buses Close page

Buses

Buses > SCA.APPLICATION.WESBCell01.Bus > Messaging engines

A messaging engine is a component, running inside a server, that manages messaging resources for a bus member. Applications are connected to a messaging engine when accessing a service integration bus.

Preferences

Start Stop mode: Immediate Stop

Select	Name	Description	Status
☐	WESBCluster01.000-SCA.APPLICATION.WESBCell01.Bus		➔
Total 1			

Help

Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

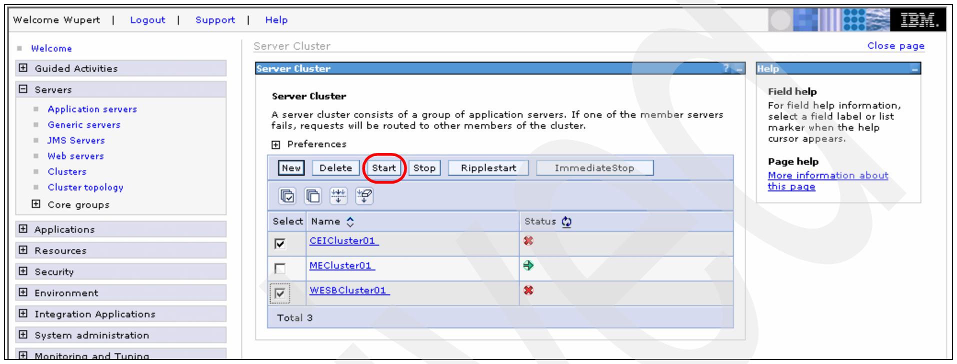
Figure 7-57 Messaging engine in started state

   e. Repeat this process for the `SCA.SYSTEM.WESBCell01.Bus` and `CommonEventInfrastructure_Bus` buses.

Starting the remaining clusters

To start the remaining clusters, follow these steps:

1. Open the Administrative Console and login.
2. Navigate to **Servers** → **Clusters**.
3. Select the cluster check box for each stopped cluster and click **Start**, as shown in Figure 7-55.



Welcome Wupert | Logout | Support | Help

- Welcome
- Guided Activities
- Servers
  - Application servers
  - Generic servers
  - JMS Servers
  - Web servers
  - Clusters
  - Cluster topology
  - Core groups
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning

Server Cluster | Close page

Server Cluster

A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster.

Preferences

New | Delete | Start | Stop | Ripplestart | ImmediateStop

Select	Name	Status
☑	CEICluster01	✗
☐	MECluster01	➔
☑	WESBCluster01	✗
Total 3		

Help

Field help

For field help information, select a field label or list marker when the help cursor appears.

Page help

More information about this page

Figure 7-58 Starting the remaining clusters

4. Verify that the clusters start successfully. The red cross should change to a solid green arrow. You can view detailed information in SystemOut.log on each cluster member.

7.4 Installing and testing the scenario

This section describes how to test the topology that you have built by deploying presentation logic, business module, and enterprise service layers.

7.4.1 Installing the enterprise applications

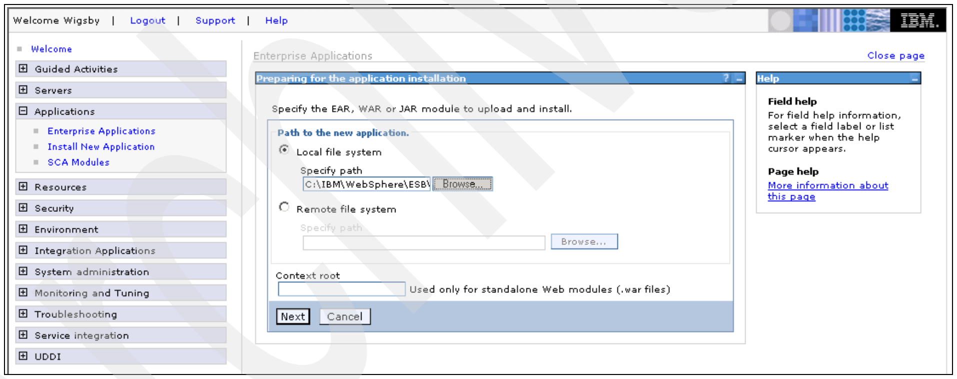
To test the business mediation logic for this topology, you need to create a presentation logic layer and a remote enterprise service layer. The presentation logic layer provides the client for the business module, and the remote enterprise service layer presents services to be mediated.

In this example the presentation logic and remote enterprise service layers are provided by two stand-alone WebSphere Application Server instances. The presentation logic enterprise application, ITSOESBCplxPrsApp.ear, is installed on `egt3` to `WASNode01 on server1`. The enterprise service enterprise application, ITSOESBCplxSrvApp.ear, is installed on `egt3 to WASNode02 on server1`.

Deploying the mediation application

The application supporting the business mediation logic is provided by a single EAR file. You must install this application to the cluster `WESBCluster01`. Follow these steps:

1. Open the `egt0` Administrative Console and log in.
2. Navigate to **Applications** → **Install New Application**.
3. You will install the enterprise application ITSOESBSmplBusApp.ear, which includes the mediation logic. This file is located in the additional material that is supplied with this book. To learn how to obtain it, see Appendix A, “Additional material” on page 427. Copy ITSOESBCplxBusApp.ear from the additional material to c:\IBM\WebSphere\ESB\installableApps.
4. In the Administrative Console, select **Local file system** and click **Browse**.
5. Navigate to the c:\IBM\WebSphere\ESB\installableApps folder.
6. Select the ESB mediation module to be installed, in this example `ITSOESBCplxBusApp.ear`, and click **Open**.
7. From the Administrative Console, click **Next**, as shown in Figure 7-59.


Welcome Wigsby | Logout | Support | Help

Enterprise Applications

Preparing for the application installation

Specify the EAR, WAR or JAR module to upload and install.

Path to the new application.

(●) Local file system

Specify path: C:\IBM\WebSphere\ESB\ [Browse...]

( ) Remote file system

Specify path: [Browse...]

Context root: Used only for standalone Web modules (.war files)

[Next] [Cancel]

Help

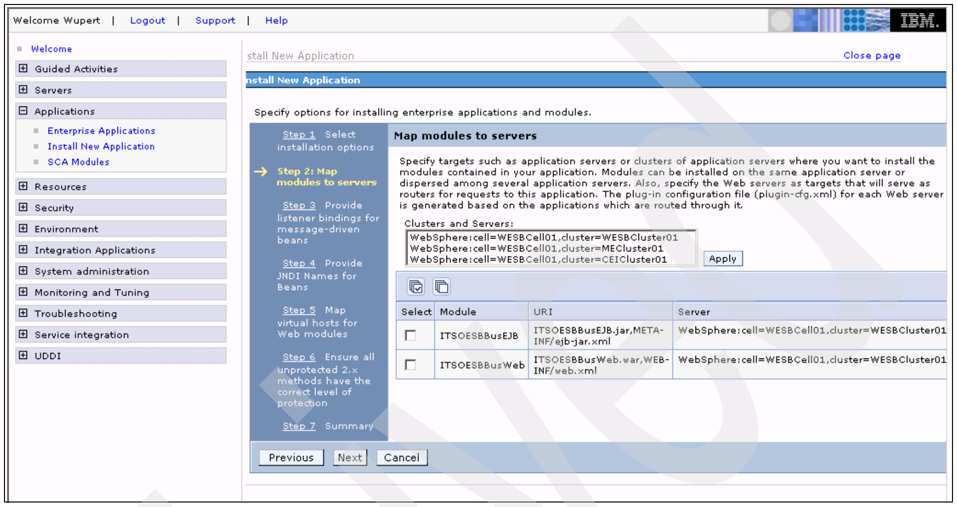
Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

Figure 7-59 Installing the business module

8. In the next panel, leave the settings as default and click **Next**.

   All the other application deployment settings will be left as default, but notably the application must be deployed to the `WESBCluster01` cluster in step 2 of the wizard, ‘Map modules to servers’ as shown in Figure 7-60.



Welcome Wupert | Logout | Support | Help

Welcome
Guided Activities
Servers
Applications
Enterprise Applications
Install New Application
SCA Modules
Resources
Security
Environment
Integration Applications
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI

stall New Application — Close page

nstall New Application

Specify options for installing enterprise applications and modules.

Step 1 Select installation options
→ Step 2: Map modules to servers
Step 3 Provide listener bindings for message-driven beans
Step 4 Provide JNDI Names for Beans
Step 5 Map virtual hosts for Web modules
Step 6 Ensure all unprotected 2.x methods have the correct level of protection
Step 7 Summary

Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install the modules contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that will serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated based on the applications which are routed through it.

Clusters and Servers:
WebSphere:cell=WESBCell01,cluster=WESBCluster01
WebSphere:cell=WESBCell01,cluster=MECluster01
WebSphere:cell=WESBCell01,cluster=CEICluster01
Apply

Select	Module	URI	Server
☐	ITSOESBBusEJB	ITSOESBBusEJB.jar,META-INF/ejb-jar.xml	WebSphere:cell=WESBCell01,cluster=WESBCluster01
☐	ITSOESBBusWeb	ITSOESBBusWeb.war,WEB-INF/web.xml	WebSphere:cell=WESBCell01,cluster=WESBCluster01

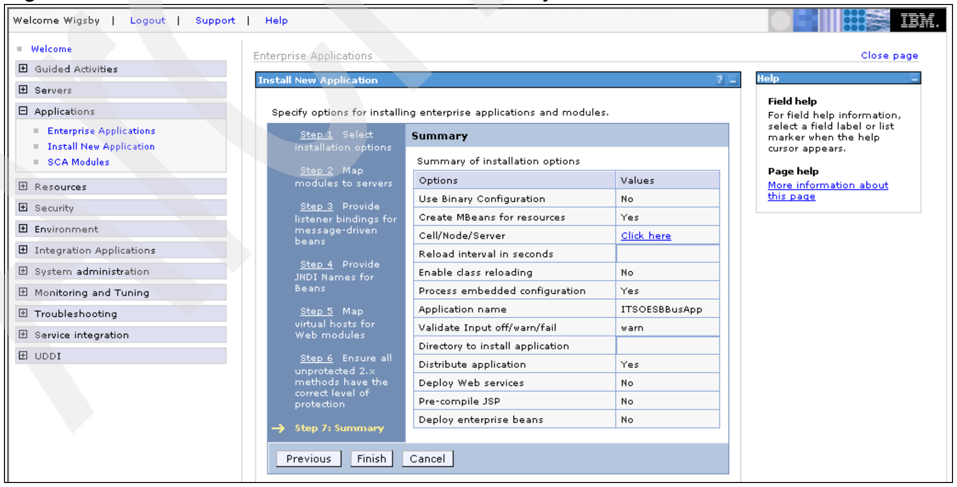
Previous | Next | Cancel

Figure 7-60 Verification of the correct application module mapping

9. Click the link to the summary panel, in this example **Step 7: Summary**.

10. Click **Finish**, as shown in Figure 7-61.

Figure 7-61 Business module installation summary



Welcome Wigsby | Logout | Support | Help

Welcome
Guided Activities
Servers
Applications
Enterprise Applications
Install New Application
SCA Modules
Resources
Security
Environment
Integration Applications
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI

Enterprise Applications — Close page

Install New Application

Specify options for installing enterprise applications and modules.

Step 1 Select installation options
Step 2 Map modules to servers
Step 3 Provide listener bindings for message-driven beans
Step 4 Provide JNDI Names for Beans
Step 5 Map virtual hosts for Web modules
Step 6 Ensure all unprotected 2.x methods have the correct level of protection
→ Step 7: Summary

Summary

Summary of installation options

Options	Values
Use Binary Configuration	No
Create MBeans for resources	Yes
Cell/Node/Server	Click here
Reload interval in seconds	
Enable class reloading	No
Process embedded configuration	Yes
Application name	ITSOESBBusApp
Validate Input off/warn/fail	warn
Directory to install application	
Distribute application	Yes
Deploy Web services	No
Pre-compile JSP	No
Deploy enterprise beans	No

Previous | Finish | Cancel

Help

Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

11. When the application has been installed successfully, click **Save to Master Configuration**.

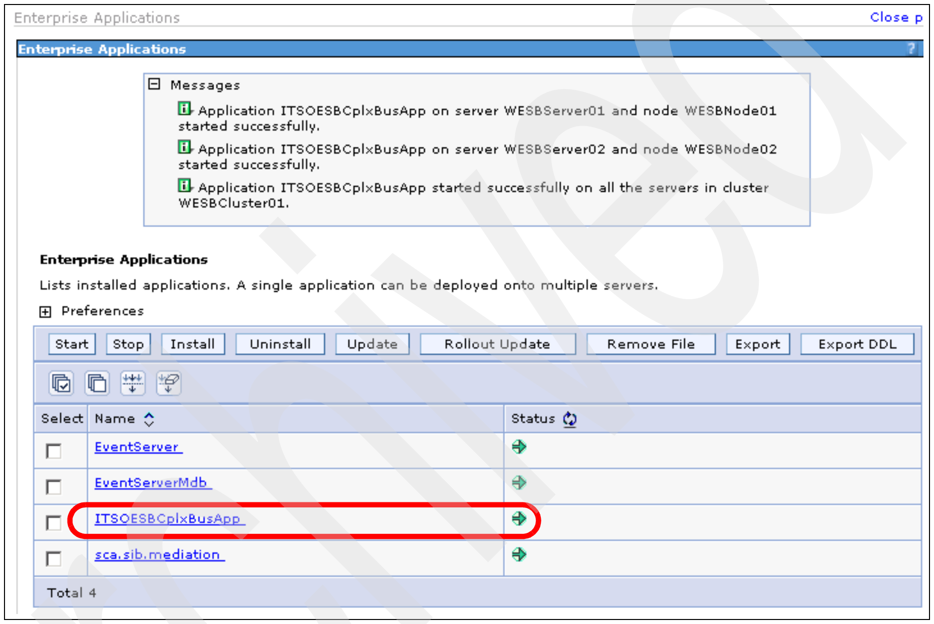
12. Select **Synchronize changes with Nodes**, and click **Save**.

13. When completed successfully, click **OK**.

14. Start the application (Figure 7-62):

    a. Navigate to **Applications** → **Enterprise Applications**.

    b. Select `ITSOESBCplxBusApp` and click **Start**.



Application ITSOESBCplxBusApp on server WESBServer01 and node WESBNode01 started successfully.
Application ITSOESBCplxBusApp on server WESBServer02 and node WESBNode02 started successfully.
Application ITSOESBCplxBusApp started successfully on all the servers in cluster WESBCluster01.

Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

Preferences
Start Stop Install Uninstall Update Rollout Update Remove File Export Export DDL

Select	Name	Status
☐	EventServer	➜
☐	EventServerMdb	➜
☐	ITSOESBCplxBusApp	➜
☐	sca.sib.mediation	➜
Total 4		

Figure 7-62 Starting the mediation module

Preparing the presentation logic and enterprise service layers

Define a WebSphere Application Server server to host the presentation logic layer, then install the presentation logic application. Follow these steps:

1. Create a new WebSphere Application Server profile using the profile creation wizard on `egt3`. On Windows platforms, launch the Profile Wizard by clicking **Start** → **All Programs** → **IBM WebSphere** → **Application Server Network Deployment V6** → **Profile creation wizard**.

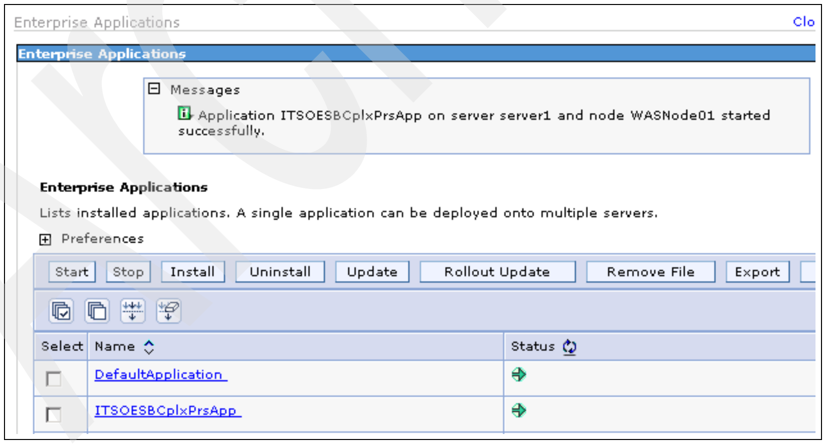
2. Define a profile with the following properties:
   - Profile type: `Application Server profile`
   - Profile name: `WASNode01`
   - Node name: `WASNode01`
   - Host name: `egt3.itso`
3. When the profile is created, navigate to the bin directory of the profile (which on our system was C:\IBM\WebSphere\ESB\profiles\WASNode01\bin) and start the server using the command:

   ```
   startServer server1
   ```

4. When the server is started, launch the Administrative Console:

   ```
   http://egt3.itso:9060/ibm/console
   ```

5. Install the presentation logic enterprise application ITSOESBSmplPrsApp.ear. This file is located in the additional material that is supplied with this book. To learn how to obtain it, see Appendix A, “Additional material” on page 427.
   a. Copy ITSOESBCplxPrsApp.ear from the additional material that is supplied with this book to c:\IBM\WebSphere\ESB\installableApps.
   b. In the Administrative Console, click **Applications** → **Install New Application** and install ITSOESBCplxPrsApp.ear. When the application is installed, save the configuration.
   c. Click **Applications** → **Enterprise Applications** and start **ITSOESBCplxPrsApp** (Figure 7-63).



Application ITSOESBCplxPrsApp on server server1 and node WASNode01 started successfully.

Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

Select	Name	Status
☐	DefaultApplication	➡
☐	ITSOESBCplxPrsApp	➡

Figure 7-63 Installed enterprise application ITSOESBCplxPrsApp

Define a WebSphere Application Server server to host the enterprise service layer, then install the enterprise service application. Follow these steps:

1. Create a new WebSphere Application Server profile using the profile creation wizard on `egt3`. On Windows platforms, launch the Profile Wizard by clicking **Start** → **All Programs** → **IBM WebSphere** → **Application Server Network Deployment V6** → **Profile creation wizard**.
2. Define a profile with the following properties:
   - Profile type: `Application Server profile`
   - Profile name: `WASNode02`
   - Node name: `WASNode02`
   - Host name: `egt3.itso`
3. When the profile is created, navigate to the bin directory of the profile (which on our system was C:\IBM\WebSphere\ESB\profiles\WASNode02\bin) and start the server using the command:

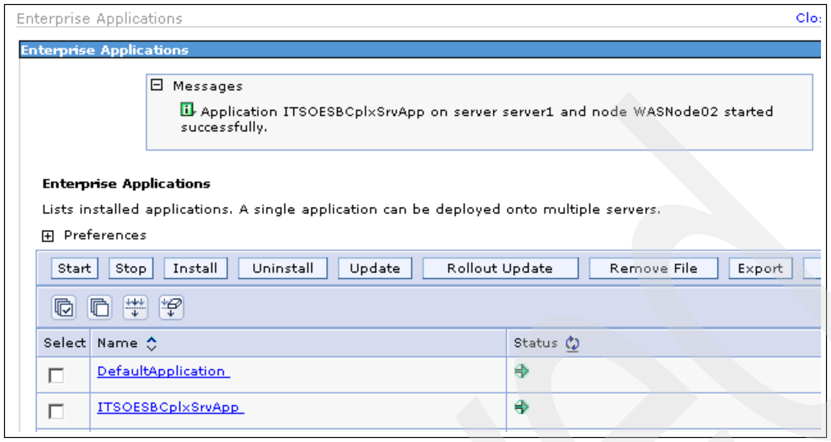
   ```
   startServer server1
   ```
4. When the server is started, launch the Administrative Console:

   ```
   http://egt3.itso:9061/ibm/console
   ```
5. Install the presentation logic enterprise application ITSOESBCplxSrvApp.ear. This file is located in the additional material that is supplied with this book. To learn how to obtain it, see Appendix A, “Additional material” on page 427. Then, follow these steps:

   a. Copy ITSOESBCplxSrvApp.ear from the additional material supplied with this book to c:\IBM\WebSphere\ESB\installableApps.

   b. In the Administrative Console, click **Applications** → **Install New Application** and install ITSOESBCplxSrvApp.ear. When the application is installed, save the configuration.

c. Click **Applications** → **Enterprise Applications** and start **ITSOESBCplxSrvApp** (Figure 7-64).



Application ITSOESBCplxSrvApp on server server1 and node WASNode02 started successfully.

Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

Select	Name	Status
☐	DefaultApplication	➡
☐	ITSOESBCplxSrvApp	➡

Figure 7-64 Installed enterprise application ITSOESBCplxSrvApp

7.4.2 Configuring IBM HTTP Server

You need to install and configure a router in front of the mediation cluster so that all members of the cluster can be accessed by the presentation logic layer.

Client requests from the browser are made to the presentation layer that runs in a stand-alone WebSphere Application Server. The JSP running on this stand-alone application server cannot make requests to multiple destinations hosting the Web service. The IBM HTTP Server and WebSphere Plugin work together to provide a single point of access for the JSP, routing requests from the presentation logic layer to the mediation cluster members, as illustrated in Figure 7-65. Mediation then takes place, and a Web services call is made to the appropriate target. The response is directed back through the original cluster member.

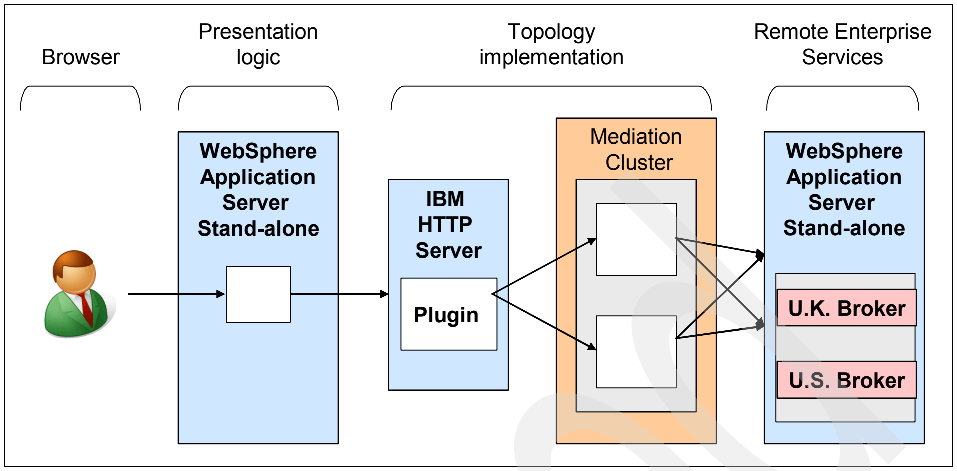


Figure 7-65 Load-balanced and failover-capable mediation cluster

The following products must already be install:

- IBM HTTP Server in c:\IBMIHS
- The WebSphere plugin for IBM HTTP Server in c:\IBM\WebSphere\Plugins

For instructions on how to install these products, refer to Appendix B, "Product installation" on page 429.

Adding the Web server to the cell

To add the Web server to the cell, follow these steps:

1. Open a command prompt on the machine where the Web server plugin has been installed.
2. Change directory to c:\IBM\WebSphere\Plugins\bin.

> **Note:** In this example, the Web server is named `webserver1`. Therefore, the configuration script is named configurewebserver1.bat.

3. Copy the configurewebserver1.bat file to %WAS_HOME%\bin on any node in the cell.
4. Change directory to %*WAS_HOME*%\bin and run configurewebserver1.bat.

This script adds the Web server into the cell, and deploys the applications to the Web server.

Configuring the plug-in

To configure the plug-in, follow these steps:

1. Open the Administrative Console on `egt0` and login.
2. Navigate to **Servers** → **Web servers**.
3. Select **webserver1** and click **Generate Plug-in**.
4. When complete, the message text identifies the location of the generated plug-in file. In this example, it is:

   ```
   c:\IBM\WebSphere\ESB\profiles\Dmgr01\config\cells\WESBCell01\nodes\e
   gt0.itso\servers\webserver1\plugin-cfg.xml
   ```

5. Move the plugin-cfg.xml configuration file.

   The location of plugin-cfg.xml that WebSphere Plug-in uses is specified by the `WebSpherePluginConfig` variable within the httpd.conf configuration file of IBM HTTP Server. Therefore, the generated plugin-cfg.xml must be moved to the path that is defined by this variable. In this example, the command is:

   ```
   move
   c:\IBM\WebSphere\ESB\profiles\Dmgr01\config\cells\WESBCell01\nodes\e
   gt0.itso\servers\webserver1\plugin-cfg.xml
   c:\IBM\WebSphere\Plugins\config\webserver1\plugin-cfg.xml
   ```

Configuring the application deployment

There are two additional steps to configure the application deployment:

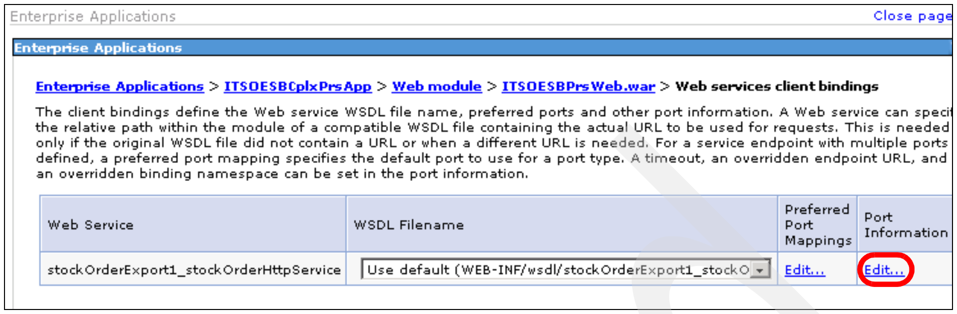
1. Configuring the presentation application for the mediation export
2. Configuring the mediation application for the remote enterprise services

Presentation logic connectivity to mediation layer

To configure the presentation logic connectivity to the mediation layer:

1. Open the presentation logic layer's WebSphere Application Server Administrative Console and login (e.g. `http://egt3.itso:9060/ibm/console`).
2. Navigate to **Applications** → **Enterprise Applications**.
3. Click the presentation application, **ITSOESBCplxPrsApp**.
4. Under Related Items, click **Web modules**.
5. Click the Web Archive (WAR) file, **ITSOESBPrsWeb.war**.
6. Under Additional Properties, click **Web services client bindings**.

7. Under Port Information, click **Edit**, as shown in Figure 7-66.



Enterprise Applications — Close page

Enterprise Applications

Enterprise Applications > ITSOESBCplxPrsApp > Web module > ITSOESBPrsWeb.war > **Web services client bindings**

The client bindings define the Web service WSDL file name, preferred ports and other port information. A Web service can speci the relative path within the module of a compatible WSDL file containing the actual URL to be used for requests. This is needed only if the original WSDL file did not contain a URL or when a different URL is needed. For a service endpoint with multiple ports defined, a preferred port mapping specifies the default port to use for a port type. A timeout, an overridden endpoint URL, and an overridden binding namespace can be set in the port information.

Web Service	WSDL Filename	Preferred Port Mappings	Port Information
stockOrderExport1_stockOrderHttpService	Use default (WEB-INF/wsdl/stockOrderExport1_stockO	Edit...	Edit...

Figure 7-66 Editing port information for the presentation logic layer

8. Change the Overridden Endpoint URL to that of the IBM HTTP Server, in this example:

   `http://egt0.itso:80/ITSOESBBusWeb/sca/stockOrderExport`

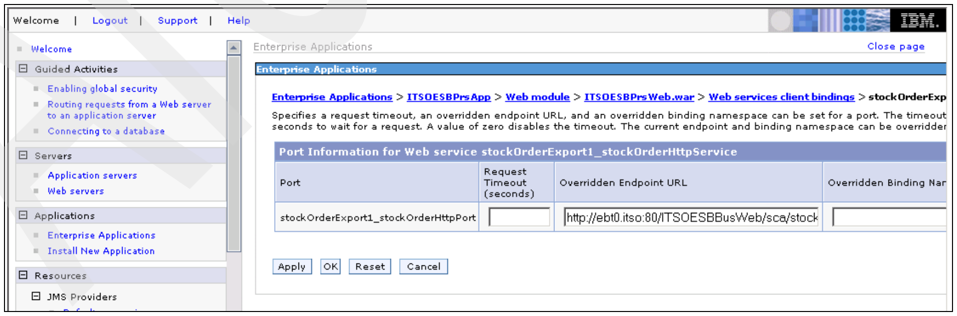
   Note: This Endpoint URL is constructed from:

   – The host name of the IBM HTTP Server and the port on which it listens. In this example `egt0.itso:80`

   – The context-root of the Web module that is defined in the deployment descriptor of the mediation application, `ITSOESBBusWeb`.

   – The Web service URL-pattern that is defined in the deployment descriptor of the WAR file within the mediation application. In this example `sca/stockOrderExport`.

   Figure 7-67 shows this step.



Enterprise Applications — Close page

Enterprise Applications

Enterprise Applications > ITSOESBPrsApp > Web module > ITSOESBPrsWeb.war > Web services client bindings > **stockOrderExp**

Specifies a request timeout, an overridden endpoint URL, and an overridden binding namespace can be set for a port. The timeout seconds to wait for a request. A value of zero disables the timeout. The current endpoint and binding namespace can be overridder

Port Information for Web service stockOrderExport1_stockOrderHttpService

Port	Request Timeout (seconds)	Overridden Endpoint URL	Overridden Binding Nar
stockOrderExport1_stockOrderHttpPort		http://ebt0.itso:80/ITSOESBBusWeb/sca/stock	

Apply | OK | Reset | Cancel

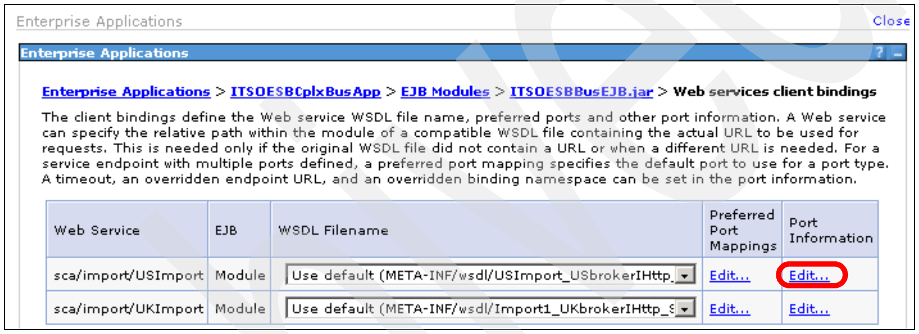
Figure 7-67 Configuring the Endpoint URL

9. Click **OK**. Save and synchronize the changes.

Mediation layer connectivity to remote enterprise services

To configure the mediation layer connectivity to the remote enterprise services:

1. Open the `WESBCell` deployment manager's Administrative Console and login.
2. Navigate to **Applications** → **Enterprise Applications**.
3. Click **ITSOESBCplxBusApp**.
4. Under Related Items, click **EJB Modules**.
5. Click the JAR file that is presented, **ITSOESBBusEJB.jar**.
6. Under Additional Properties, click **Web services client bindings**.
7. For the sca/import/USImport service, click **Edit** under **Port Information**, as shown in Figure 7-68.


Enterprise Applications Close

Enterprise Applications

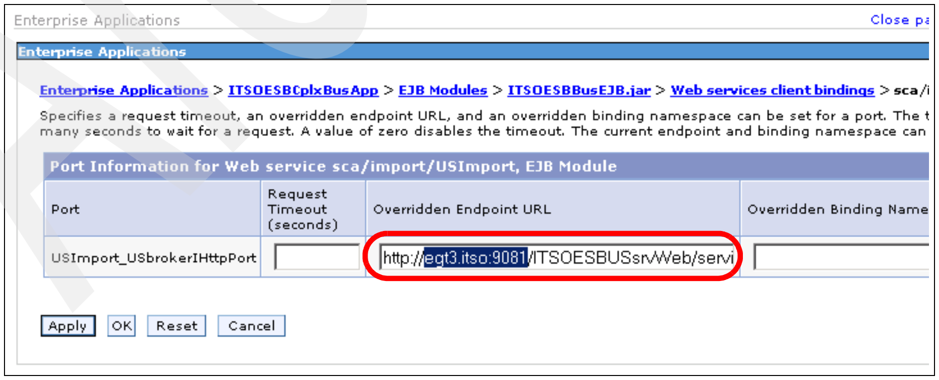
Enterprise Applications > ITSOESBCplxBusApp > EJB Modules > ITSOESBBusEJB.jar > Web services client bindings

The client bindings define the Web service WSDL file name, preferred ports and other port information. A Web service can specify the relative path within the module of a compatible WSDL file containing the actual URL to be used for requests. This is needed only if the original WSDL file did not contain a URL or when a different URL is needed. For a service endpoint with multiple ports defined, a preferred port mapping specifies the default port to use for a port type. A timeout, an overridden endpoint URL, and an overridden binding namespace can be set in the port information.

Web Service	EJB	WSDL Filename	Preferred Port Mappings	Port Information
sca/import/USImport	Module	Use default (META-INF/wsdl/USImport_USbrokerIHttp_	Edit...	Edit...
sca/import/UKImport	Module	Use default (META-INF/wsdl/Import1_UKbrokerIHttp_S	Edit...	Edit...

Figure 7-68 Editing port information for the remote enterprise services

8. Change the host and port to that of the WebSphere Application Server that is hosting the remote enterprise services, in this example `http://egt3.itso:9081/` (Figure 7-69).


Enterprise Applications Close pa

Enterprise Applications

Enterprise Applications > ITSOESBCplxBusApp > EJB Modules > ITSOESBBusEJB.jar > Web services client bindings > sca/i

Specifies a request timeout, an overridden endpoint URL, and an overridden binding namespace can be set for a port. The t many seconds to wait for a request. A value of zero disables the timeout. The current endpoint and binding namespace can

Port Information for Web service sca/import/USImport, EJB Module

Port	Request Timeout (seconds)	Overridden Endpoint URL	Overridden Binding Name
USImport_USbrokerIHttpPort		http://egt3.itso:9081/ITSOESBUSsrvWeb/servi	

Figure 7-69 Configuring the host for the remote enterprise services

9. Repeat the previous step for the `sca/import/UKImport` service.

10. Save and synchronize the changes, and click **OK** when complete.

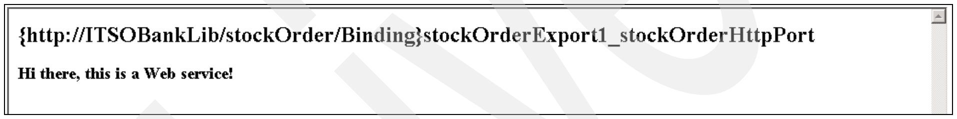
It is possible to test each of the remote enterprise services with a browser directed at the following URLs:

- `http://egt3.itso:9081/ITSOESBUSsrvWeb/services/USbrokerIExport1_USbrokerIHttpPort`
- `http://egt3.itso:9081/ITSOESBUKsrvWeb/services/UKbrokerIExport1_UKbrokerIHttpPort`

The mediation Web services can be reached directly on the following URLs:

- `http://egt1.itso:9080/ITSOESBBusWeb/sca/stockOrderExport`
- `http://egt2.itso:9080/ITSOESBBusWeb/sca/stockOrderExport`

Figure 7-70 shows an example response.

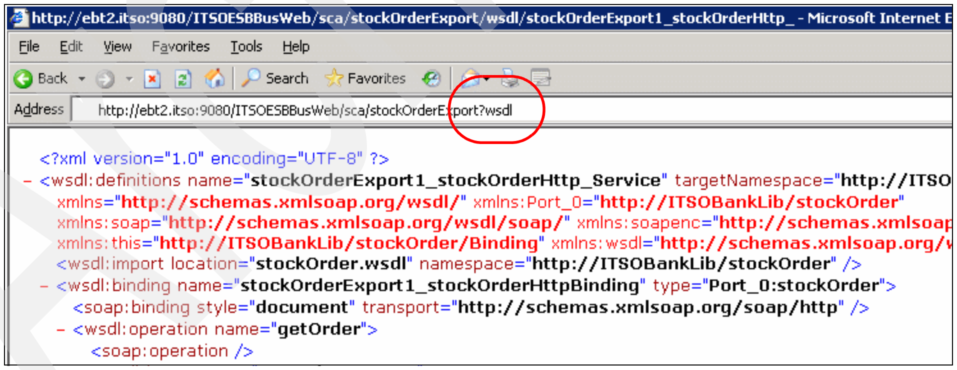


{http://ITSOBankLib/stockOrder/Binding}stockOrderExport1_stockOrderHttpPort

Hi there, this is a Web service!

Figure 7-70 Example response from Web service call

By appending `?wsdl` to these URLs, the actual WSDL definition can be displayed, as shown in Figure 7-71.


http://ebt2.itso:9080/ITSOESBBusWeb/sca/stockOrderExport/wsdl/stockOrderExport1_stockOrderHttp_ - Microsoft Internet E
File Edit View Favorites Tools Help
Back Search Favorites
Address

```
http://ebt2.itso:9080/ITSOESBBusWeb/sca/stockOrderExport?wsdl
```

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions name="stockOrderExport1_stockOrderHttp_Service" targetNamespace="http://ITSO
    xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:Port_0="http://ITSOBankLib/stockOrder"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:soapenc="http://schemas.xmlsoap
    xmlns:this="http://ITSOBankLib/stockOrder/Binding" xmlns:wsdl="http://schemas.xmlsoap.org/v
    <wsdl:import location="stockOrder.wsdl" namespace="http://ITSOBankLib/stockOrder" />
  - <wsdl:binding name="stockOrderExport1_stockOrderHttpBinding" type="Port_0:stockOrder">
      <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
    - <wsdl:operation name="getOrder">
        <soap:operation />
```

Figure 7-71 Requesting the WSDL definition

7.4.3 Testing the scenario

This section tests the environment and the application that it hosts and shows both mediation and load-balancing that occurs. It also demonstrates the logging of CEI messages.

Starting the environment

To start the environment, follow these steps:

1. Ensure that all three layers of the environment (presentation, mediation, and remote enterprise services) are started.
2. Ensure that all applications are running in these layers.
3. Start up IBM HTTP Server with the following commands:

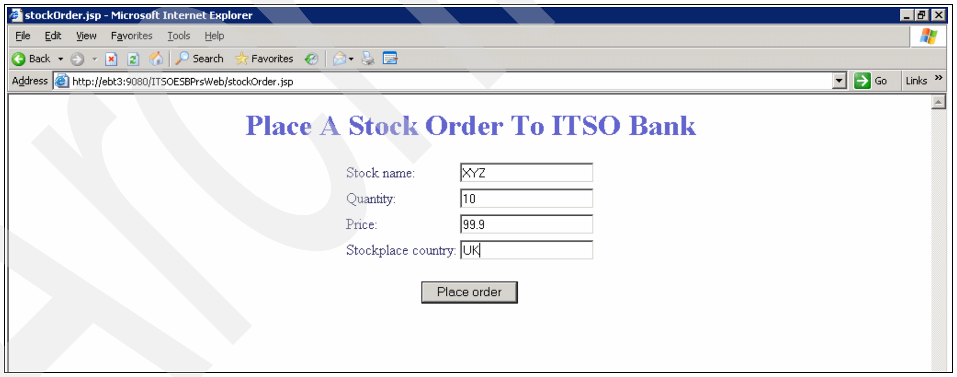
```
cd c:\IBMIHS\bin
Apache.exe -k start
```

Testing the mediation

Testing the application involves accessing the JSP front end and sending in a request:

1. Open a Web browser and launch the presentation logic JSP (Figure 7-72). We used:

```
http://egt3.itso:9080/ITSOESBPrsWeb/stockOrder.jsp
```


stockOrder.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go Links
Place order

http://ebt3:9080/ITSOESBPrsWeb/stockOrder.jsp

Place A Stock Order To ITSO Bank

Stock name:	XYZ
Quantity:	10
Price:	99.9
Stockplace country:	UK

Figure 7-72 Accessing the presentation layer JSP

2. Complete the fields, making sure to specify a Stockplace country value of `UK`. Click **Place order**. The mediation logic places the order with the U. K. broker service and returns a confirmation ID as shown in Figure 7-73.

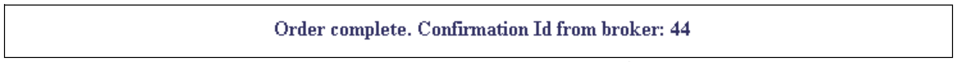

Order complete. Confirmation Id from broker: 44

Figure 7-73 Sample response from U. K. broker

3. Enter a second request, this time specifying a Stockplace country value of `USA`. Click **Place order**. The mediation logic will place the order with the UK broker service, and will return a confirmation id as shown in Figure 7-74 on page 224.

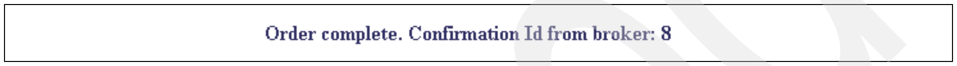

Order complete. Confirmation Id from broker: 8

Figure 7-74 Sample response from U. S. broker

4. Enter a third request, this time specifying a Stockplace country of `Canada`. Click **Place order**. The non-U. K. or non-U. S. stockCountry causes a CEI event to be generated that can be viewed using the Common Base Event Browser (Figure 7-75). The order is then placed with the U. S. broker.

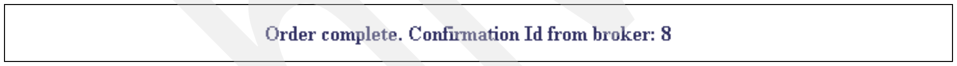

Order complete. Confirmation Id from broker: 8

Figure 7-75 Sample response from US broker

This example shows mediation occurring based on the Stockplace country data that is provided by the JSP in the presentation layer. In one case, the `UKbrokerIExport1_UKbrokerIHttpPort` remote service is called and in the other two cases the `USbrokerIExport1_USbrokerIHttpPort` remote service is called.

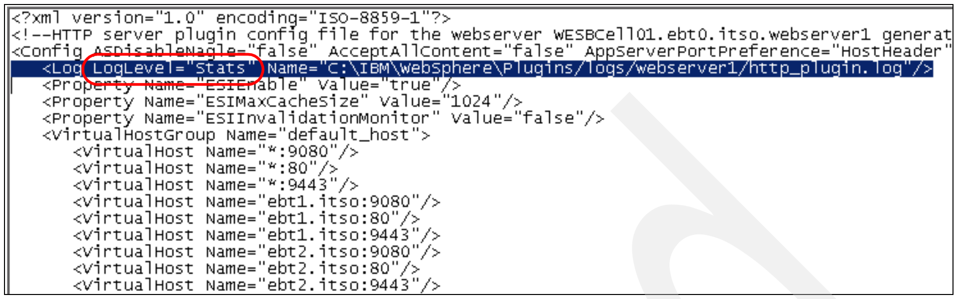
Testing load-balancing

By default, the WebSphere Plug-in passes requests made to the HTTP server onto the mediation cluster, using the round-robin balancing algorithm.

By changing the WebSphere Plug-in logging level defined in the plugin-cfg.xml file, it is possible to view which mediation cluster member is selected as each request is made. Follow these steps:

1. Open the plugin-cfg.xml file using an appropriate text editor. In this case, the directory is c:\IBM\WebSphere\Plugins\config\webserver1\plugin-cfg.xml.

2. In the first `Log` stanza, change the `LogLevel` parameter to `Stats`, as shown in Figure 7-76.



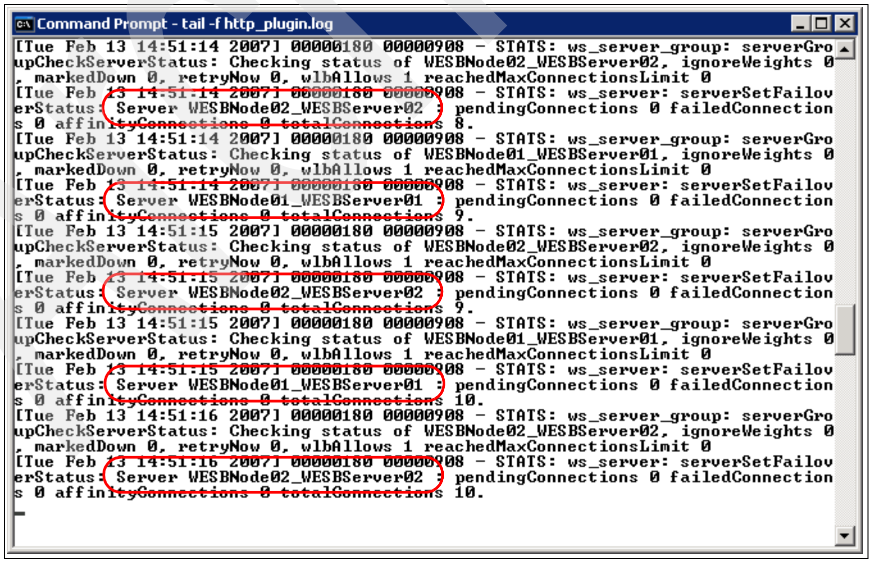
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--HTTP server plugin config file for the webserver WESBCell01.ebt0.itso.webserver1 generat
<Config ASDisableNagle="false" AcceptAllContent="false" AppServerPortPreference="HostHeader"
   <Log LogLevel="Stats" Name="C:\IBM\WebSphere\Plugins/logs/webserver1/http_plugin.log"/>
   <Property Name="ESIEnable" Value="true"/>
   <Property Name="ESIMaxCacheSize" Value="1024"/>
   <Property Name="ESIInvalidationMonitor" Value="false"/>
   <VirtualHostGroup Name="default_host">
      <VirtualHost Name="*:9080"/>
      <VirtualHost Name="*:80"/>
      <VirtualHost Name="*:9443"/>
      <VirtualHost Name="ebt1.itso:9080"/>
      <VirtualHost Name="ebt1.itso:80"/>
      <VirtualHost Name="ebt1.itso:9443"/>
      <VirtualHost Name="ebt2.itso:9080"/>
      <VirtualHost Name="ebt2.itso:80"/>
      <VirtualHost Name="ebt2.itso:9443"/>
```

Figure 7-76 Changing the logging level for the WebSphere Plug-in

3. Save the plugin-cfg.xml file and restart the HTTP server:

```
cd c:\IBMIHS\bin
Apache -k restart
```

4. Submit a series of requests to the presentation layer JSP, as described in “Testing the mediation” on page 223.
5. Open the WebSphere Plug-in log file in an appropriate text editor, in this case c:\IBM\WebSphere\Plugins\logs\webserver1\http_plugin.log.
6. A series of lines shows each request that is passed to members of the mediation cluster in turn, as shown in Figure 7-77.



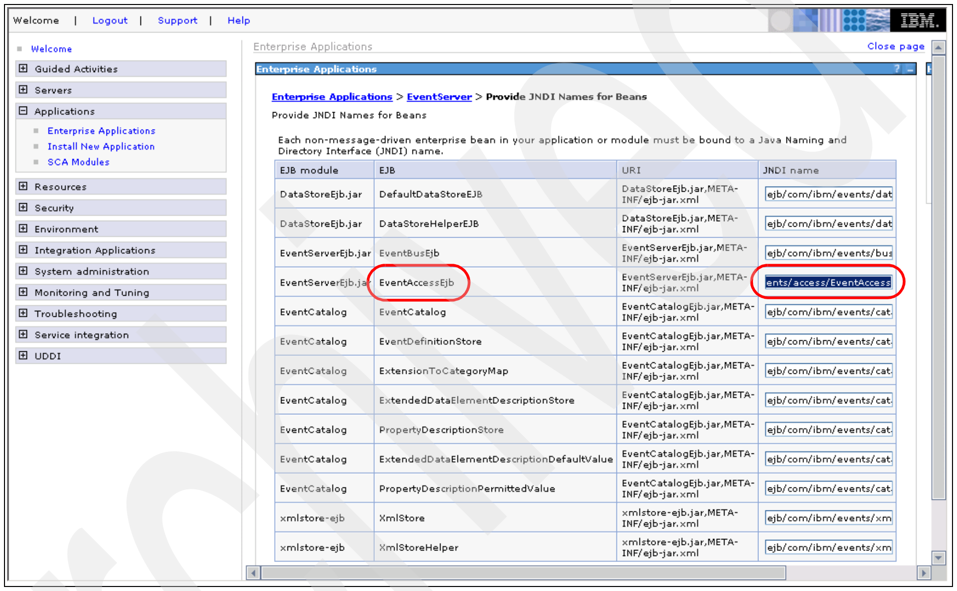
```
[Tue Feb 13 14:51:14 2007] 00000180 00000908 - STATS: ws_server_group: serverGro
upCheckServerStatus: Checking status of WESBNode02_WESBServer02, ignoreWeights 0
, markedDown 0, retryNow 0, wlbAllows 1 reachedMaxConnectionsLimit 0
[Tue Feb 13 14:51:14 2007] 00000180 00000908 - STATS: ws_server: serverSetFailov
erStatus: Server WESBNode02_WESBServer02 : pendingConnections 0 failedConnection
s 0 affinityConnections 0 totalConnections 8.
[Tue Feb 13 14:51:14 2007] 00000180 00000908 - STATS: ws_server_group: serverGro
upCheckServerStatus: Checking status of WESBNode01_WESBServer01, ignoreWeights 0
, markedDown 0, retryNow 0, wlbAllows 1 reachedMaxConnectionsLimit 0
[Tue Feb 13 14:51:14 2007] 00000180 00000908 - STATS: ws_server: serverSetFailov
erStatus: Server WESBNode01_WESBServer01 : pendingConnections 0 failedConnection
s 0 affinityConnections 0 totalConnections 9.
[Tue Feb 13 14:51:15 2007] 00000180 00000908 - STATS: ws_server_group: serverGro
upCheckServerStatus: Checking status of WESBNode02_WESBServer02, ignoreWeights 0
, markedDown 0, retryNow 0, wlbAllows 1 reachedMaxConnectionsLimit 0
[Tue Feb 13 14:51:15 2007] 00000180 00000908 - STATS: ws_server: serverSetFailov
erStatus: Server WESBNode02_WESBServer02 : pendingConnections 0 failedConnection
s 0 affinityConnections 0 totalConnections 9.
[Tue Feb 13 14:51:15 2007] 00000180 00000908 - STATS: ws_server_group: serverGro
upCheckServerStatus: Checking status of WESBNode01_WESBServer01, ignoreWeights 0
, markedDown 0, retryNow 0, wlbAllows 1 reachedMaxConnectionsLimit 0
[Tue Feb 13 14:51:15 2007] 00000180 00000908 - STATS: ws_server: serverSetFailov
erStatus: Server WESBNode01_WESBServer01 : pendingConnections 0 failedConnection
s 0 affinityConnections 0 totalConnections 10.
[Tue Feb 13 14:51:16 2007] 00000180 00000908 - STATS: ws_server_group: serverGro
upCheckServerStatus: Checking status of WESBNode02_WESBServer02, ignoreWeights 0
, markedDown 0, retryNow 0, wlbAllows 1 reachedMaxConnectionsLimit 0
[Tue Feb 13 14:51:16 2007] 00000180 00000908 - STATS: ws_server: serverSetFailov
erStatus: Server WESBNode02_WESBServer02 : pendingConnections 0 failedConnection
s 0 affinityConnections 0 totalConnections 10.
_
```

Figure 7-77 Load-balanced requests

Viewing Common Events

To view Common Events:

1. In the Administrative Console of `egt0`, navigate to **Applications** → **Enterprise Applications**.
2. Click the **EventServer** application.
3. Under Additional Properties, click **Provide JNDI Names for Beans**.
4. Note the contents of the **JNDI name** field for the **EventAccessEjb** EJB, as shown in Figure 7-78.



Enterprise Applications > EventServer > Provide JNDI Names for Beans

Provide JNDI Names for Beans

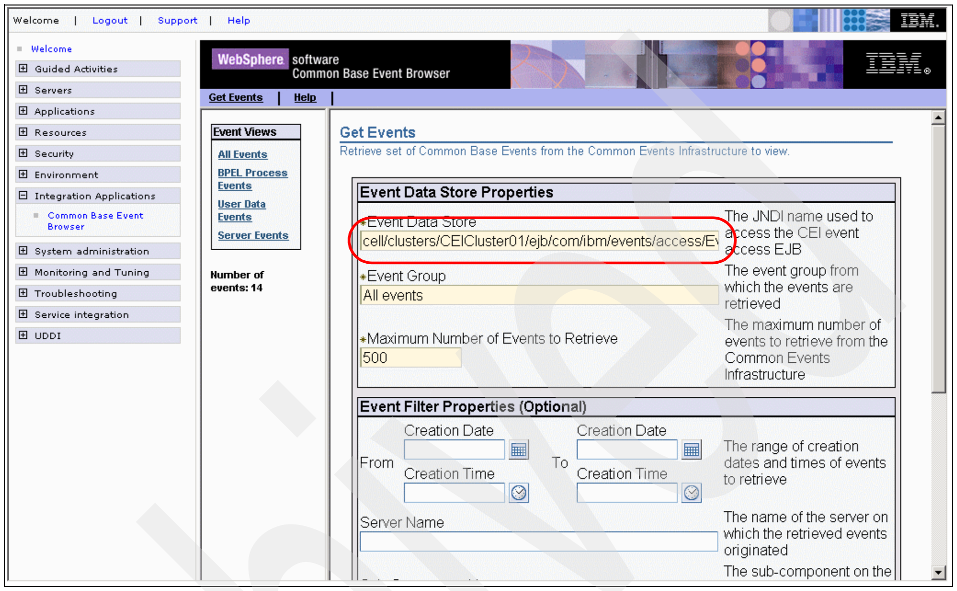
Each non-message-driven enterprise bean in your application or module must be bound to a Java Naming and Directory Interface (JNDI) name.

EJB module	EJB	URI	JNDI name
DataStoreEjb.jar	DefaultDataStoreEJB	DataStoreEjb.jar,META-INF/ejb-jar.xml	ejb/com/ibm/events/dat
DataStoreEjb.jar	DataStoreHelperEJB	DataStoreEjb.jar,META-INF/ejb-jar.xml	ejb/com/ibm/events/dat
EventServerEjb.jar	EventBusEjb	EventServerEjb.jar,META-INF/ejb-jar.xml	ejb/com/ibm/events/bus
EventServerEjb.ja	EventAccessEjb	EventServerEjb.jar,META-INF/ejb-jar.xml	ents/access/EventAccess
EventCatalog	EventCatalog	EventCatalogEjb.jar,META-INF/ejb-jar.xml	ejb/com/ibm/events/cat
EventCatalog	EventDefinitionStore	EventCatalogEjb.jar,META-INF/ejb-jar.xml	ejb/com/ibm/events/cat
EventCatalog	ExtensionToCategoryMap	EventCatalogEjb.jar,META-INF/ejb-jar.xml	ejb/com/ibm/events/cat
EventCatalog	ExtendedDataElementDescriptionStore	EventCatalogEjb.jar,META-INF/ejb-jar.xml	ejb/com/ibm/events/cat
EventCatalog	PropertyDescriptionStore	EventCatalogEjb.jar,META-INF/ejb-jar.xml	ejb/com/ibm/events/cat
EventCatalog	ExtendedDataElementDescriptionDefaultValue	EventCatalogEjb.jar,META-INF/ejb-jar.xml	ejb/com/ibm/events/cat
EventCatalog	PropertyDescriptionPermittedValue	EventCatalogEjb.jar,META-INF/ejb-jar.xml	ejb/com/ibm/events/cat
xmlstore-ejb	XmlStore	xmlstore-ejb.jar,META-INF/ejb-jar.xml	ejb/com/ibm/events/xm
xmlstore-ejb	XmlStoreHelper	xmlstore-ejb.jar,META-INF/ejb-jar.xml	ejb/com/ibm/events/xm

Figure 7-78 Obtaining the JNDI name for the EventAccessEJB

5. Navigate to **Integration Applications** → **Common Base Event Browser**.

6. At the top of the panel, set the Event Data Store entry to `cell/clusters/<CEI cluster>/` and the JNDI name noted earlier. In this example, the entry is `cell/clusters/CEICluster01/ejb/com/ibm/events/access/EventAccess`, as shown in Figure 7-79. This is required where CEI is clustered.



Number of events: 14

Get Events

Retrieve set of Common Base Events from the Common Events Infrastructure to view.

Event Data Store Properties

*Event Data Store: cell/clusters/CEICluster01/ejb/com/ibm/events/access/Ev	The JNDI name used to access the CEI event access EJB
*Event Group: All events	The event group from which the events are retrieved
*Maximum Number of Events to Retrieve: 500	The maximum number of events to retrieve from the Common Events Infrastructure

Event Filter Properties (Optional)

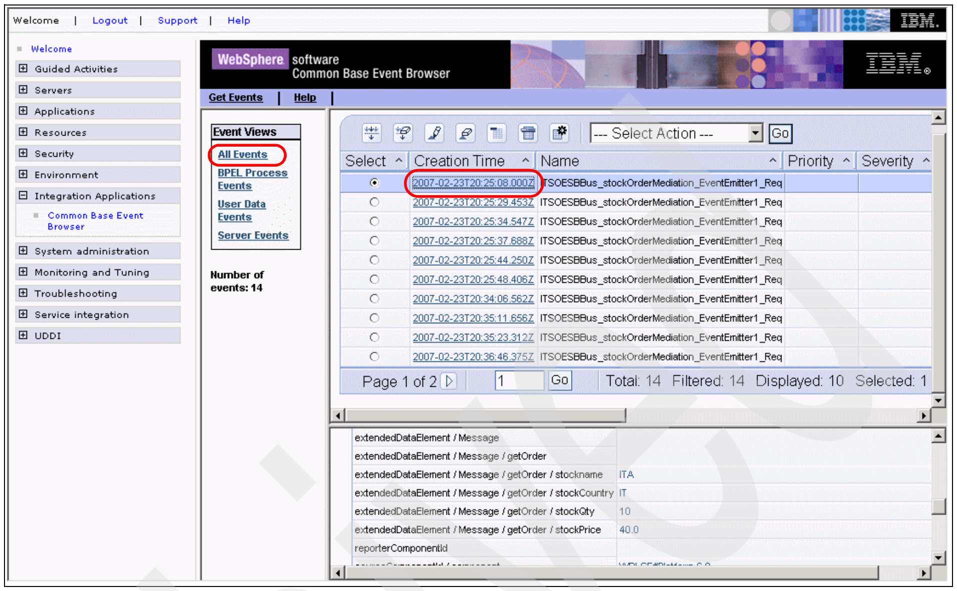
From Creation Date / Creation Time — To Creation Date / Creation Time	The range of creation dates and times of events to retrieve
Server Name	The name of the server on which the retrieved events originated
	The sub-component on the

Figure 7-79 Setting the Event Data Store JNDI name

7. At the bottom of the panel, click **Get Events** to load CEI events from the `CEIDB` database.

8. Under Event Views, click **All Events** to display the list of events.

9. Events can be viewed by clicking the **Creation Time** field, as shown in Figure 7-80.



Number of events: 14

Select	Creation Time	Name	Priority	Severity
●	2007-02-23T20:25:08.000Z	ITSOESBBus_stockOrderMediation_EventEmitter1_Req		
○	2007-02-23T20:25:29.453Z	ITSOESBBus_stockOrderMediation_EventEmitter1_Req		
○	2007-02-23T20:25:34.547Z	ITSOESBBus_stockOrderMediation_EventEmitter1_Req		
○	2007-02-23T20:25:37.688Z	ITSOESBBus_stockOrderMediation_EventEmitter1_Req		
○	2007-02-23T20:25:44.250Z	ITSOESBBus_stockOrderMediation_EventEmitter1_Req		
○	2007-02-23T20:25:48.406Z	ITSOESBBus_stockOrderMediation_EventEmitter1_Req		
○	2007-02-23T20:34:06.562Z	ITSOESBBus_stockOrderMediation_EventEmitter1_Req		
○	2007-02-23T20:35:11.656Z	ITSOESBBus_stockOrderMediation_EventEmitter1_Req		
○	2007-02-23T20:35:23.312Z	ITSOESBBus_stockOrderMediation_EventEmitter1_Req		
○	2007-02-23T20:36:46.375Z	ITSOESBBus_stockOrderMediation_EventEmitter1_Req		

Page 1 of 2 | 1 Go | Total: 14 Filtered: 14 Displayed: 10 Selected: 1

extendedDataElement / Message	
extendedDataElement / Message / getOrder	
extendedDataElement / Message / getOrder / stockname	ITA
extendedDataElement / Message / getOrder / stockCountry	IT
extendedDataElement / Message / getOrder / stockQty	10
extendedDataElement / Message / getOrder / stockPrice	40.0
reporterComponentId	

Figure 7-80 Viewing CEI events

Part 3

Production topologies for WebSphere Process Server

Implementing the Loosely Coupled topology for WebSphere Process Server

This chapter details the steps to implement the *Loosely Coupled topology* for WebSphere Process Server and to deploy a sample business process to this topology. We discuss the following topics:

- Selecting and implementing the Loosely Coupled topology
- Generic steps for creating WebSphere Process Server clusters
- Specific configuration steps for the Loosely Coupled topology
- Installing and testing the scenario

For simplicity, this chapter does not implement any security. To be considered a production topology, the addition of security is essential. Chapter 9, “Implementing the Full Support topology for WebSphere Process Server” on page 319 shows how to add security to a production topology.

8.1 Selecting and implementing the Loosely Coupled topology

This section includes the following sections:

- Scenario requirements

  Summarizes the factious business scenario that we deploy to the Loosely Coupled topology we build in this chapter.
- Selecting the appropriate topology

  Demonstrates why we select the Loosely Coupled topology for WebSphere Process Server to host this business scenario.
- Overview of how to implement the Loosely Coupled topology

  Describes the steps that are necessary to build the Loosely Coupled topology for WebSphere Process Server.

Note: For an overview of the Loosely Coupled topology, refer to 4.3, "Loosely Coupled topology" on page 48.

8.1.1 Scenario requirements

You must build a production topology to host a WebSphere Process Server business process that is built for the fictitious organization ITSO Bank.The business process (shown in Figure 8-1) performs the following functions:

1. A bank employee submits an account closure order to the business process. The interaction between the presentation and the business module is implemented using a Service Component Architecture (SCA) binding export.
2. The *Constraints check* service is invoked. The connection between the process and the service is a Web service binding that is invoked synchronously.
3. The next service is invoked in the same way, with a synchronous Web service binding.
4. The process invokes the *Closure service* with an asynchronous one-way Web service binding.
5. The process terminates successfully and confirmation is returned.

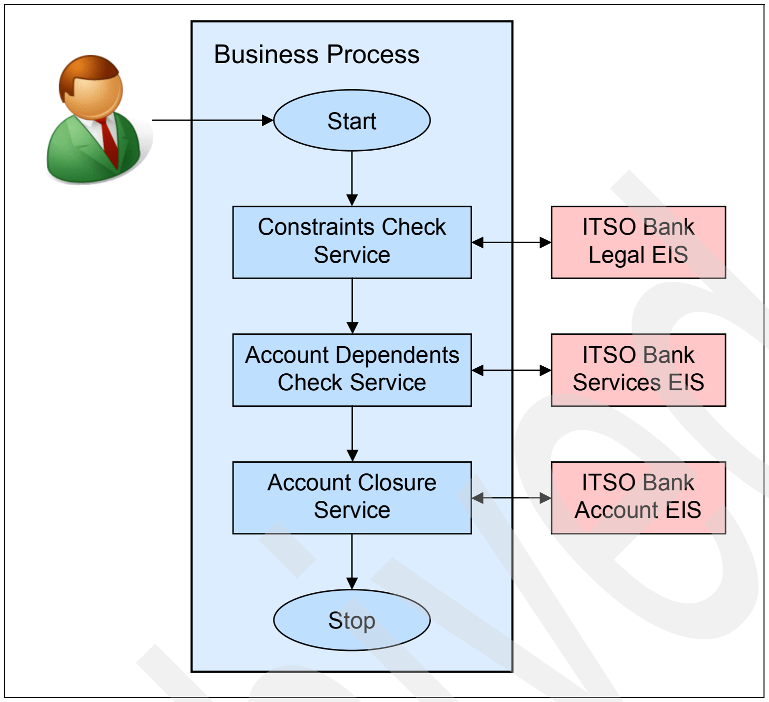


Figure 8-1 Account closure process, simple business scenario

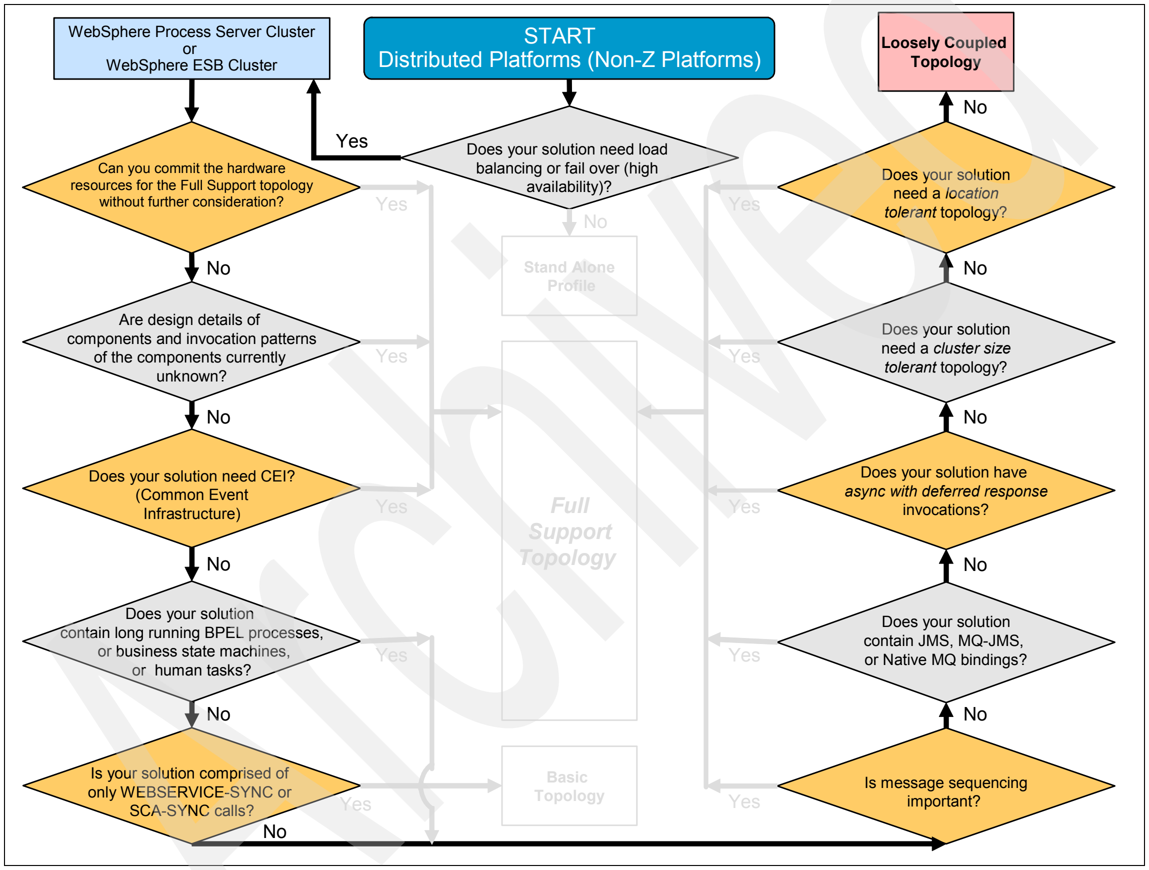
ITSO Bank has the following business and IT requirements for this scenario:

- Failover capability.
- There is insufficient hardware resources available to implement the Full Support topology. In this example, it is assumed that the bank is not investing in new hardware for the process implementation.
- To implement the new process, the bank has decided to use existing services within the enterprise.
- The ITSO Bank Account EIS service is a one-way call and is invoked asynchronously
- Event management services are not required.

Note: For more detailed information about this scenario, refer to 5.2.2, "Simple business process scenario" on page 75.

8.1.2 Selecting the appropriate topology

Based on the ITSO Bank requirements for the business process scenario, we can select an appropriate production topology to host the business process. To help us select the appropriate topology, we use the topology selection flowchart, as described in 4.5.2, “Topology selection flow chart” on page 62. Figure 8-2 shows that by using the topology selection flowchart, we can select the Loosely Coupled topology as the topology of choice for the ITSO Bank simple business process scenario.



START
Distributed Platforms (Non-Z Platforms)
WebSphere Process Server Cluster or WebSphere ESB Cluster
Loosely Coupled Topology
Does your solution need load balancing or fail over (high availability)?
Can you commit the hardware resources for the Full Support topology without further consideration?
Are design details of components and invocation patterns of the components currently unknown?
Does your solution need CEI? (Common Event Infrastructure)
Does your solution contain long running BPEL processes, or business state machines, or human tasks?
Is your solution comprised of only WEBSERVICE-SYNC or SCA-SYNC calls?
Does your solution need a location tolerant topology?
Does your solution need a cluster size tolerant topology?
Does your solution have async with deferred response invocations?
Does your solution contain JMS, MQ-JMS, or Native MQ bindings?
Is message sequencing important?
Stand Alone Profile
Full Support Topology
Basic Topology
Yes
No

Figure 8-2 Topology selection for the simple business process scenario

The main decision points for ITSO Bank are:

- A highly available solution with load balancing is required, so a clustered topology is needed.
- ITSO Bank do not want to commit the hardware resources that are required to implement the Full Support topology unless the applications that they are deploying to the topology require it.
- The business process is short-running and does not include business state machines or human tasks.
- There is a single asynchronous call that does not return a response.
- The Loosely Coupled topology meets the needs of the business process and is topology that we select.

8.1.3 Overview of how to implement the Loosely Coupled topology

Figure 8-3 summarizes the steps to build the Loosely Coupled topology.

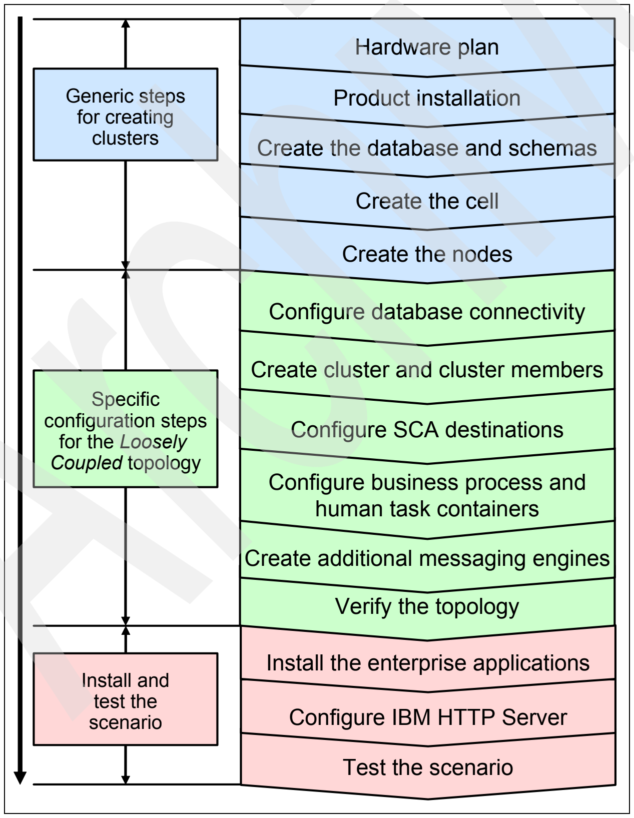

Generic steps for creating clusters
Hardware plan
Product installation
Create the database and schemas
Create the cell
Create the nodes
Specific configuration steps for the *Loosely Coupled* topology
Configure database connectivity
Create cluster and cluster members
Configure SCA destinations
Configure business process and human task containers
Create additional messaging engines
Verify the topology
Install and test the scenario
Install the enterprise applications
Configure IBM HTTP Server
Test the scenario

Figure 8-3 Building the Loosely Coupled topology for WebSphere Process Server

Figure 8-4 shows the topology that we build in this chapter.

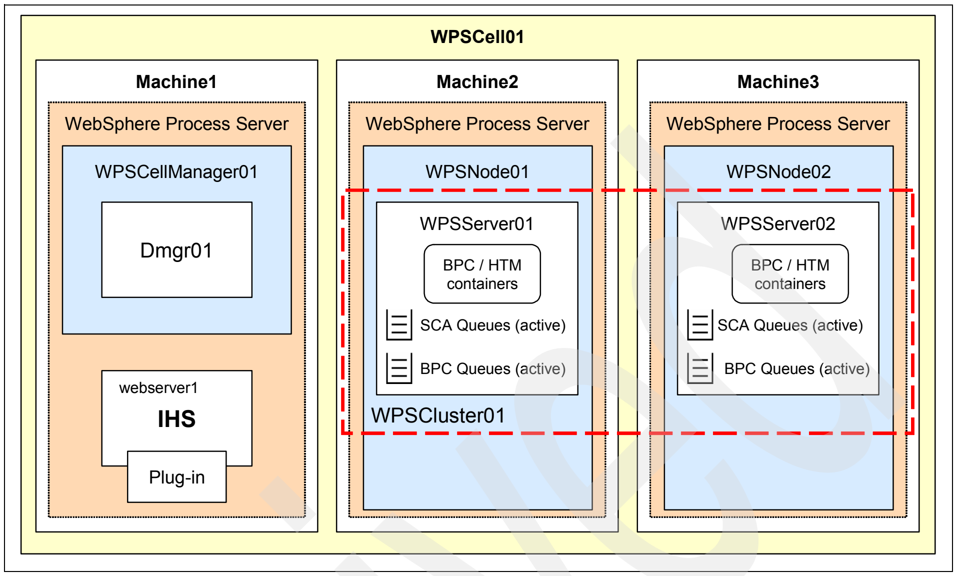


Figure 8-4 Loosely Coupled topology implementation

For each cluster bus member of SCA bus, there are multiple messaging engines. The SCA queue destination is partitioned through the cluster. Each messaging engine running on the cluster member owns one partition of the queue and shares the workload that is delivered to the cluster.

In this topology, there is a single WebSphere Process Server cluster that consists of two or more cluster members. This cluster is configured for SCA with local destinations.

The Business Process Choreographer (BPC) container is configured also within this cluster. If the BPC Explorer is required, you need to configure the Human Task Manager (HTM) container in this cluster because HTM is required to have BPC Explorer work correctly.

8.2 Generic steps for creating WebSphere Process Server clusters

This section describes the hardware plan and how to prepare a WebSphere Process Server cluster.

8.2.1 Hardware plan for the topology

This section describes hardware that is required to support the Loosely Coupled topology. In this topology (illustrated in Figure 8-5 on page 238), three machines are used to host the cell hosting the business process and one machine is used to host the Web services application for testing purposes. A final shared machine provides the database functionality. The hardware plan includes:

- The first machine (`wlt0`) hosts the deployment manager and the IBM HTTP Server and associated plugin.
- The second and third machines (`wlt1 and wlt2`) host the business process cluster, as well as the associated nodes, in a horizontally clustered configuration. These machines also host the client application to invoke the sample business process.
- The fourth machine (`wlt3`) hosts the Web services application in a stand-alone WebSphere Application Server.
- A fifth machine (`db2`), not shown in the diagram, hosts the database that is required to support the messaging engines.

> **Note:** Although this is the configuration that we used, you can choose any combination of machines to create this topology.

Table 8-1 describes how we use these machines. You can choose to use a different combination of machines and operating systems. In total, the topology in this chapter consumes about 2.5 GB of memory.

Table 8-1 Systems used for this topology

System name	Host name	Operating system
wlt0	wlt0.itso	Windows 2003 Server
wlt1	wlt1.itso	Windows 2003 Server
wlt2	wlt2.itso	Windows 2003 Server
wlt3	wlt3.itso	Windows 2003 Server
db2	db2.itso	Red Hat Advanced Server 4 U4

Note: You can define a hosts file that maps the host names that are defined in Table 8-1 to the IP address of each of your machines to help when following the steps in this chapter.

You can find the hosts file on Windows platforms at:

<WINDOWS_HOME>\system32\drivers\etc

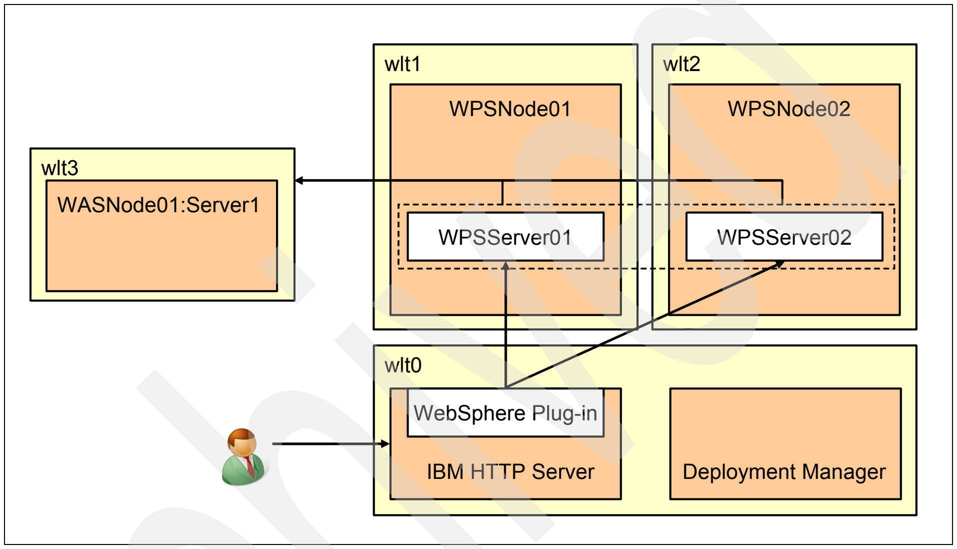


Figure 8-5 Hardware plan

The end-to-end flow for this topology, as shown in Figure 8-5, is:

1. A user makes a request from a browser to the server, which load balances through the plugin.
2. IBM HTTP Server passes the request to the front-end client application, running as a JSP on `WPSNode01:WPSServer01` or `WPSNode02:WPSServer02` on either `wlt1` or `wlt2` respectively.
3. The client application is configured to forward the request to the business application that is running on the same server (`wlt1` or `wlt2`).
4. The business process application starts the account closure process and calls the appropriate Web service (running on `wlt3`) to gather account details.
5. A confirmation is returned to the user either stating that the account has been closed or that the process has been terminated.

8.2.2 Product installation

It is assumed that the following products have been installed:

- WebSphere Process Server V6.0.2
- IBM HTTP Server V6
- DB2 Universal Database Enterprise Server V8.2

Note: For information about how to install these products, refer to Appendix B, “Product installation” on page 429.

8.2.3 Creating the databases

To create the WebSphere Process Server clusters, you need to create the databases. This section explains how to create the databases you need.

For the Loosely Coupled topology, you need three databases. You do not need to use the same names for your databases that we use in our examples, but you must your environment consistent if you use other names for the databases. You can find more information about the databases at the following Web site:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wsps.602.ins.doc/doc/cins_db_specs.html

Creating the messaging database

The first task is to create the messaging database. We assume the Relational Database Management System (RDBMS) is DB2, the instance is called `wltinst1` and the administrative user name is `wltinst1`.

Log in as the database administrative user and issue the following commands from a DB2 command window to create the required database:

```
db2start
db2 create database MEDB
```

If the create command is successful, you see the following message:

```
DB20000I The CREATE DATABASE command completed successfully
```

Creating the WebSphere Process Server common database

The next task is to create the common database for WebSphere Process Server, `WPRCSDB`. This database is the shared database that is used by applications such as the Failed Event Manager, Relationship, and CBE Browser.

Log in as the database administrative user and issue the following command from a DB2 command window to create the required database:

```
db2 create database WPRCSDB
```

If the create command is successful, you see the following message:

```
DB20000I The CREATE DATABASE command completed successfully
```

Creating the WebSphere Process Server BPEDB database

The next task is to create the BPEDB database for WebSphere Process Server business container. This database stores the process-related information.

The DB2 commands for generating the BPEDB database are located in the script %*WAS_HOME*%\dbscripts\ProcessChoreographer\DB2\createDatabase.sql. This script exists on any system with the WebSphere Process Server product installed. To create the BPEDB database:

1. Transfer this file to the DB2 server. On our environment, this script was at C:\IBM\WebSphere\ProcServer\dbscripts\ProcessChoreographer\DB2.

   Attention: If you transfer these files from Windows to UNIX, there might be return characters (\r) at the end of each line. To run the DDL scripts, you must remove these characters. A UNIX utility such as dos2unix will remove these characters.

2. By default, this script creates a database called BPEDB. If you want to create the database using a different name, edit createDatabase.sql and replace BPEDB with the database name of your choice (Example 8-1). We used the default name.

Example 8-1 Modify the database name

```
CREATE DATABASE BPEDB USING CODESET UTF-8 TERRITORY en-us;
CONNECT TO BPEDB;
```

3. Log in as the database administrative user (in this example wltinst1) and issue the following command from a DB2 command window to create the required database:

   ```
   db2 -tf createDatabase.sql
   ```

Note: The default script creates a schema set as the currently logged in user (wltinst1). You can modify the script to your specific requirements.

When completed successfully, the BPEDB database exists.

Listing the created databases

Confirm that all three databases were created successfully in DB2 by following these steps:

1. Run the following command to list the databases that are defined to DB2:

   ```
   db2 list db directory
   ```

2. The three databases should be listed. Example 8-2 shows the first of these (`WPRCSDB`).

Example 8-2 System database directory

```
System Database Directory

Number of entries in the directory = 3

Database 1 entry:

Database alias                       = WPRCSDB
Database name                        = WPRCSDB
```

8.2.4 Copying the database driver files

The Java driver files for DB2 are located on the database system. In our DB2 example, they are in the directory /opt/IBM/db2/V8.1/java.

On each of the nodes and the deployment manager, copy the database driver files to a standard location. In this example, the standard location for the driver files is c:\db2client.

8.2.5 Creating the messaging database schemas

The messaging database includes three schemas that support the messaging engines in this topology:

- SCA Application
- SCA System
- Business processes

You can create the DB2 commands that generate the schemas on any system with the WebSphere Process Server product installed. On any system where WebSphere Process Server is installed, create the data definition language (DDL) files for the Service Integration Bus (SIB) schemas as detailed in Table 8-2.

Table 8-2 WebSphere Process Server SCA schemas

Schema name	Database	Purpose
SCASYS	MEDB	System bus messaging store
SCASYS2	MEDB	System bus messaging store
SCAAPP	MEDB	Application bus messaging store
SCAAPP2	MEDB	Application bus messaging store
BPCMSG	MEDB	Business process messaging store
BPCMSG2	MEDB	Business process messaging store

In this example:

- The database user name is `wltinst1`.
- The script to generate the schemas is located in the %*WAS_HOME*%\bin directory (for example, C:\IBM\WebSphere\ProcServer\bin).
- The database target for the DDL generated is running on a Linux system.

To create the messaging database schemas, follow these steps:

1. Open a command prompt and issue the following command:

   ```
   cd C:\IBM\WebSphere\ProcServer\bin
   ```

   Note: Each of the commands in this series of steps is a single line although in the format of this book they might wrap. Also, the commands assume that the database is running on a UNIX platform. If you are using Windows, you need to replace -platform unix with `-platform windows`.

2. Issue the following command for the SCASYS DDL generation - SCA System Schema:

   ```
   sibDDLGenerator.bat -system db2 -version 8.1 -platform unix -schema
   SCASYS -statementend ; -user wltinst1 >
   c:\createSIBSchema_SCASYS.ddl
   ```

3. Issue the following command for the SCASYS2 DDL generation - SCA System Schema for second messaging engine:

   ```
   sibDDLGenerator.bat -system db2 -version 8.1 -platform unix -schema
   SCASYS2 -statementend ; -user wltinst1 >
   c:\createSIBSchema_SCASYS2.ddl
   ```

4. Issue the following command for the SCAAPP DDL generation - SCA Application Schema:

   ```
   sibDDLGenerator.bat -system db2 -version 8.1 -platform unix -schema
   SCAAPP -statementend ; -user wltinst1 >
   c:\createSIBSchema_SCAAPP.ddl
   ```

5. Issue the following command for the SCAAPP2 DDL generation - SCA Application Schema for second messaging engine:

   ```
   sibDDLGenerator.bat -system db2 -version 8.1 -platform unix -schema
   SCAAPP2 -statementend ; -user wltinst1 >
   c:\createSIBSchema_SCAAPP2.ddl
   ```

6. Issue the following command for the BPCMSG DDL generation - Business Process Schema:

   ```
   sibDDLGenerator.bat -system db2 -version 8.1 -platform unix -schema
   BPCMSG -statementend ; -user wltinst1 >
   c:\createSIBSchema_BPCMSG.ddl
   ```

7. Issue the following command for the BPCMSG2 DDL generation - Business Process Schema for second messaging engine:

   ```
   sibDDLGenerator.bat -system db2 -version 8.1 -platform unix -schema
   BPCMSG2 -statementend ; -user wltinst1 >
   c:\createSIBSchema_BPCMSG2.ddl
   ```

8. Transfer the DDL files to DB2 database server. In this example, we transferred the DDL files to /tmp on the database server.

   Attention: If you transfer these files from Windows to UNIX, there might be return characters (`\r`) at the end of each line. To run the DDL scripts, you must remove these characters. A UNIX utility such as `dos2unix` will remove these characters.

9. Login to the DB2 server as the appropriate user (`wltinst1`) and run the following commands to create the tables:

```
db2 connect to MEDB
db2 -tf /tmp/createSIBSchema_SCAAPP.ddl
db2 -tf /tmp/createSIBSchema_SCAAPP2.ddl
db2 -tf /tmp/createSIBSchema_SCASYS.ddl
db2 -tf /tmp/createSIBSchema_SCASYS2.ddl
db2 -tf /tmp/createSIBSchema_BPCMSG.ddl
db2 -tf /tmp/createSIBSchema_BPCMSG2.ddl
```

Note: These DB2 commands report various informational index and primary key messages, such as:

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

These are not errors. You can ignore these messages.

8.2.6 Configuring the WebSphere Process Server cell

This task creates the WebSphere Process Server cell, which is achieved by creating a deployment manager profile. Follow these steps:

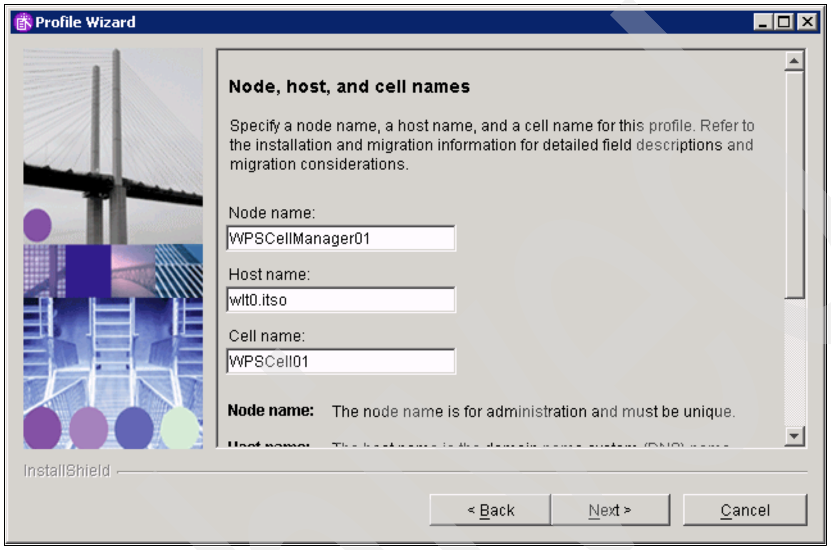
1. Log in to the `wlt0` machine, which becomes the deployment manager.
2. Open the profile creation wizard by selecting **Start** → **All Programs** → **IBM WebSphere** → **Process Server 6.0** → **Profile creation wizard**.

Note: For information about how to launch the Profile creation wizard on other platforms, see:

```
http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.js
p?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tpro_instances.html
```

3. In the Profile creation wizard, click **Next**.
4. Select **Deployment manager profile** and click **Next**.
5. Leave the default profile name as *Dmgr01* and click **Next**.
6. Leave the default profile directory and click **Next**.

7. Specify the node name, host name, and cell name according to the following specifications and click Next, as shown in Figure 8-6:
   - Node name: `WPSCellManager01`
   - Host name: `wlt0.itso`
   - Cell name: `WPSCell01`



Node, host, and cell names

Specify a node name, a host name, and a cell name for this profile. Refer to the installation and migration information for detailed field descriptions and migration considerations.

Node name: WPSCellManager01

Host name: wlt0.itso

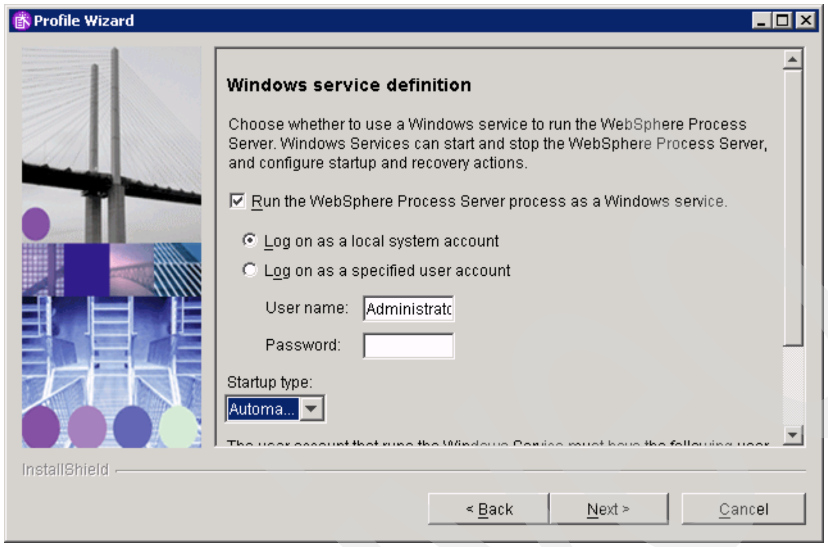
Cell name: WPSCell01

Node name: The node name is for administration and must be unique.

Figure 8-6 Node, host, and cell name configuration

8. Leave port numbers at their default settings and click **Next**.

9. In the Windows service definition panel, select **Automatic** and click **Next**, as shown in Figure 8-7.



Windows service definition

Choose whether to use a Windows service to run the WebSphere Process Server. Windows Services can start and stop the WebSphere Process Server, and configure startup and recovery actions.

[✓] Run the WebSphere Process Server process as a Windows service.

(●) Log on as a local system account

( ) Log on as a specified user account

User name: Administratc

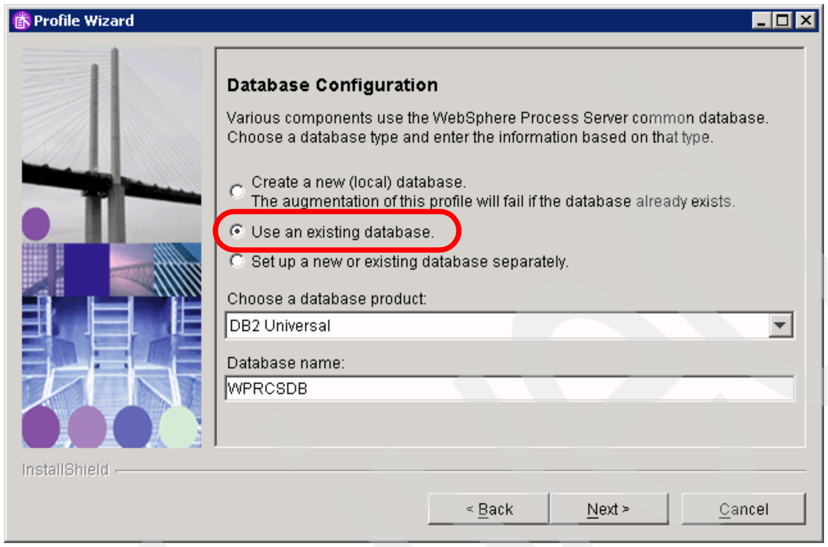
Password:

Startup type: Automa...

Figure 8-7 Windows service settings

10. On the Service Component Architecture configuration panel, click **Next**. You will not secure the SBI™ in this scenario.

11. From the Database Configuration panel, **select Use an existing database**. This example uses DB2 Universal as the database product and WPRCSDB as the database name, as shown in Figure 8-8.



Database Configuration

Various components use the WebSphere Process Server common database. Choose a database type and enter the information based on that type.

( ) Create a new (local) database.
The augmentation of this profile will fail if the database already exists.

[✓ Use an existing database.]

( ) Set up a new or existing database separately.

Choose a database product: DB2 Universal

Database name: WPRCSDB

Figure 8-8 WPRCSDB database settings

12. Click **Next** to continue. In the subsequent database configuration panel, set the values as shown in Table 8-3 then click **Next**.

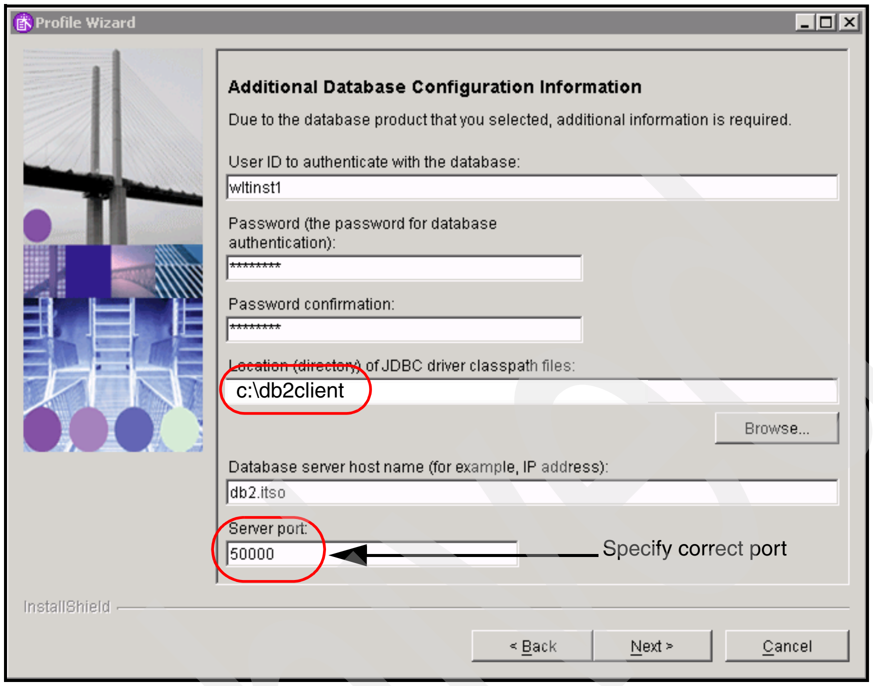
For information about using other database products, refer to:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wsps.602.ins.doc/doc/cins_common_database.html

Table 8-3 Database settings

Name	**Value**
User ID to authenticate with the database	wltinst1
Password (for database authentication)	itso4you
Password confirmation	itso4you
Location (directory) of JDBC driver classpath files	c:\db2client
Database server host name	db2.itso
Server port	50000

Figure 8-9 shows these settings.


Profile Wizard

Additional Database Configuration Information

Due to the database product that you selected, additional information is required.

User ID to authenticate with the database: wltinst1

Password (the password for database authentication): ********

Password confirmation: ********

Location (directory) of JDBC driver classpath files: c:\db2client

Database server host name (for example, IP address): db2.itso

Server port: 50000

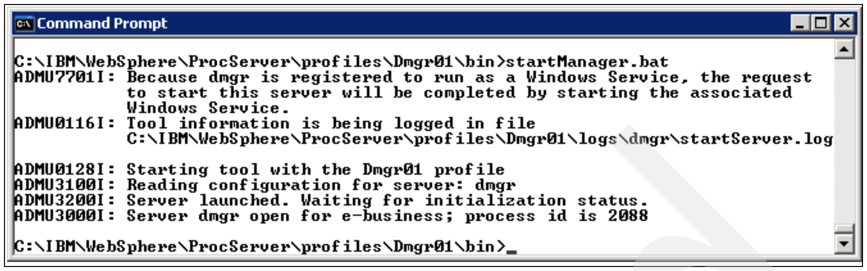
Specify correct port

Figure 8-9 Additional database configuration information

13. Confirm that the details are correct on the Profile summary panel, and click **Next**.
14. A panel displays the progress. When complete, deselect **Launch the First Steps console** and click **Finish**. The deployment manager now exists.
15. Open a command prompt on `wlt0`, change directory to %*WAS_HOME*%\profiles\Dmgr01\bin (for example, C:\IBM\WebSphere\ProcServer\profiles\Dmgr01\bin), and issue the command:

```
startManager.bat
```

Figure 8-10 shows the output from this command.



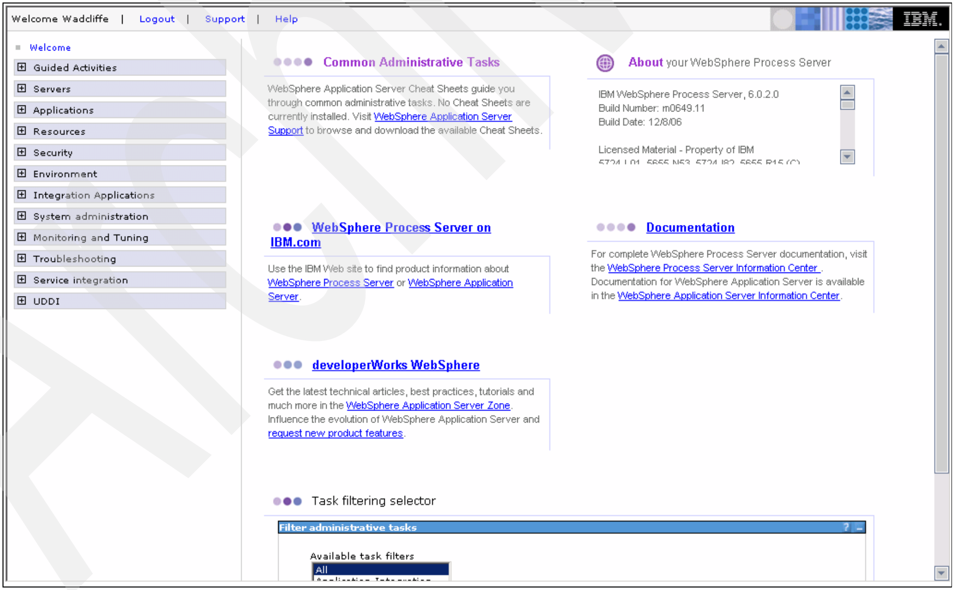
```
C:\IBM\WebSphere\ProcServer\profiles\Dmgr01\bin>startManager.bat
ADMU7701I: Because dmgr is registered to run as a Windows Service, the request
           to start this server will be completed by starting the associated
           Windows Service.
ADMU0116I: Tool information is being logged in file
           C:\IBM\WebSphere\ProcServer\profiles\Dmgr01\logs\dmgr\startServer.log

ADMU0128I: Starting tool with the Dmgr01 profile
ADMU3100I: Reading configuration for server: dmgr
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server dmgr open for e-business; process id is 2088

C:\IBM\WebSphere\ProcServer\profiles\Dmgr01\bin>_
```

Figure 8-10 Starting the deployment manager

It should now be possible to connect to the Administrative Console that is running on the deployment manager by opening a browser to:

```
http://wlt0.itso:9060/ibm/console
```

Log in to the Administrative Console (as any user). Figure 8-11 shows the Web page that opens. It might be necessary to set the available task filters to *All* in the Filter administrative tasks panel and click **Apply** on the Welcome page of the Administrative Console to override the default filtering of the available options.



Common Administrative Tasks

WebSphere Application Server Cheat Sheets guide you through common administrative tasks. No Cheat Sheets are currently installed. Visit WebSphere Application Server Support to browse and download the available Cheat Sheets.

About your WebSphere Process Server

IBM WebSphere Process Server, 6.0.2.0
Build Number: m0649.11
Build Date: 12/8/06

Licensed Material - Property of IBM

WebSphere Process Server on IBM.com

Use the IBM Web site to find product information about WebSphere Process Server or WebSphere Application Server.

Documentation

For complete WebSphere Process Server documentation, visit the WebSphere Process Server Information Center . Documentation for WebSphere Application Server is available in the WebSphere Application Server Information Center.

developerWorks WebSphere

Get the latest technical articles, best practices, tutorials and much more in the WebSphere Application Server Zone. Influence the evolution of WebSphere Application Server and request new product features.

Task filtering selector

Filter administrative tasks

Available task filters: All

Figure 8-11 First login to the Administrative Console

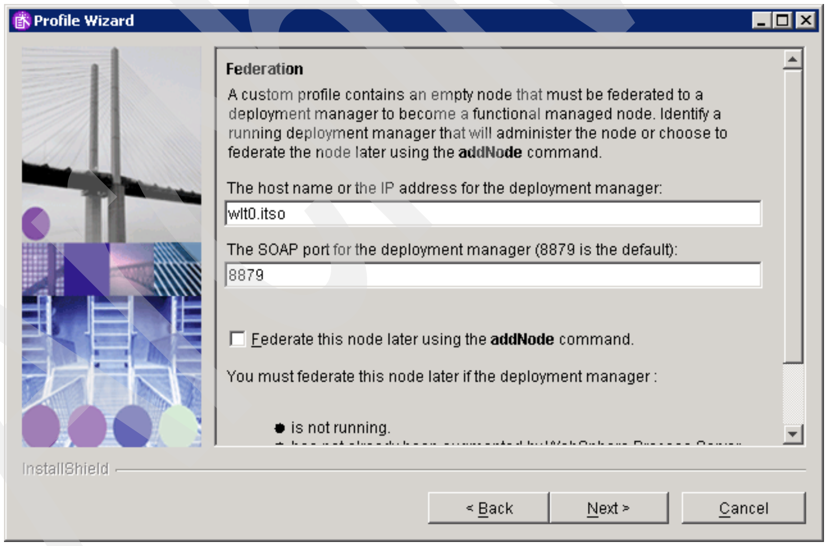
8.2.7 Creating the nodes

The next step in creating the Loosely Coupled topology is to create the profiles and associated nodes for the topology. The deployment manager must be running on `wlt0.itso` to enable node federation, the nodes are created on `wlt1.itso` and `wlt2.itso`.

> **Note:** Ensure that the time difference between the deployment manager machine and the custom profile machines is within *five* minutes.

To create the nodes, follow these steps:

1. On `wlt1` open the profile creation wizard by selecting **Start** → **All Programs** → **IBM WebSphere** → **Process Server 6.0** → **Profile creation wizard**.
2. In the profile creation wizard, click **Next**.
3. Select **Custom profile** and click **Next**.
4. Enter the host name of the deployment manager, in this example `wlt0.itso`, leave the SOAP port as the default (`8879`), ensure that **Federate this node later** is *not* selected, and click **Next,** as shown in Figure 8-12.



Federation

A custom profile contains an empty node that must be federated to a deployment manager to become a functional managed node. Identify a running deployment manager that will administer the node or choose to federate the node later using the **addNode** command.

The host name or the IP address for the deployment manager:
wlt0.itso

The SOAP port for the deployment manager (8879 is the default):
8879

☐ Federate this node later using the **addNode** command.

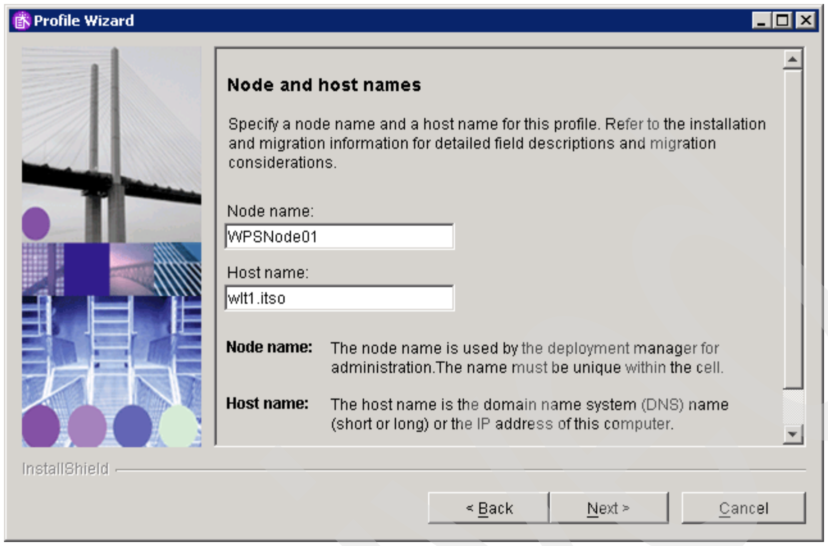
You must federate this node later if the deployment manager :

- is not running.

Figure 8-12 Deployment manager location and federation

5. Set the Profile name appropriately (`WPSNode01`) and click **Next**.
6. Accept the default profile directory and click **Next**.

7. Specify the node name (`WPSNode01`) and host name (`wlt1.itso`) on the subsequent panel and click **Next**, as shown in Figure 8-13.



Node and host names

Specify a node name and a host name for this profile. Refer to the installation and migration information for detailed field descriptions and migration considerations.

Node name: WPSNode01

Host name: wlt1.itso

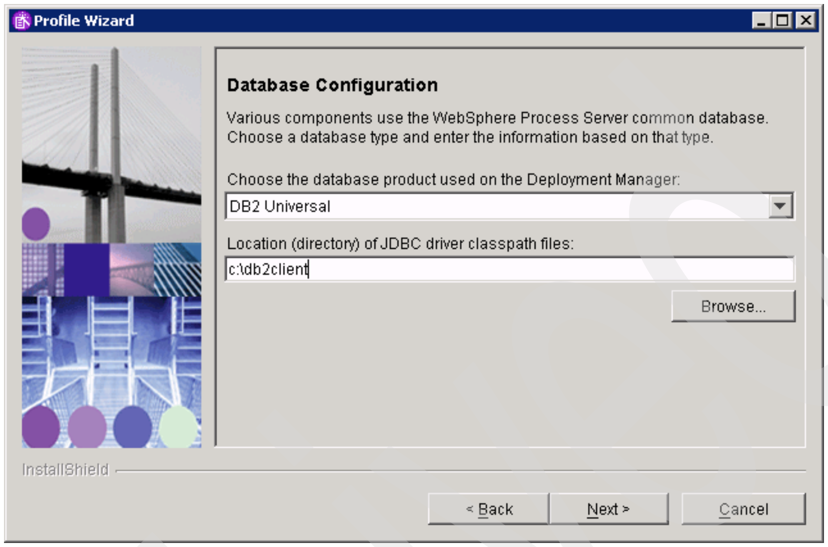
Node name: The node name is used by the deployment manager for administration.The name must be unique within the cell.

Host name: The host name is the domain name system (DNS) name (short or long) or the IP address of this computer.

Figure 8-13 Node and host name

8. Leave the default port numbers and click **Next**.

9. In the Database Configuration panel, set the database product to **DB2 Universal** and the JDBC driver classpath to **c:\db2client** as shown in Figure 8-14. Click **Next**.



Database Configuration

Various components use the WebSphere Process Server common database. Choose a database type and enter the information based on that type.

Choose the database product used on the Deployment Manager: DB2 Universal

Location (directory) of JDBC driver classpath files: c:\db2client

Figure 8-14 Database configuration

10. Confirm that the details are correct on the Profile summary panel and click **Next**.
11. A panel displays the progress. When complete, clear the **Launch the First Steps console** check box and click **Finish**.

The node now exists and is federated into the cell. The federation process starts the node automatically.

Repeat these steps for the second node on `wlt2`, using the values that are shown in Table 8-4.

Table 8-4 Second node details

Setting	Value
Profile name	WPSNode02
Node name	WPSNode02
Host name	wlt2.itso

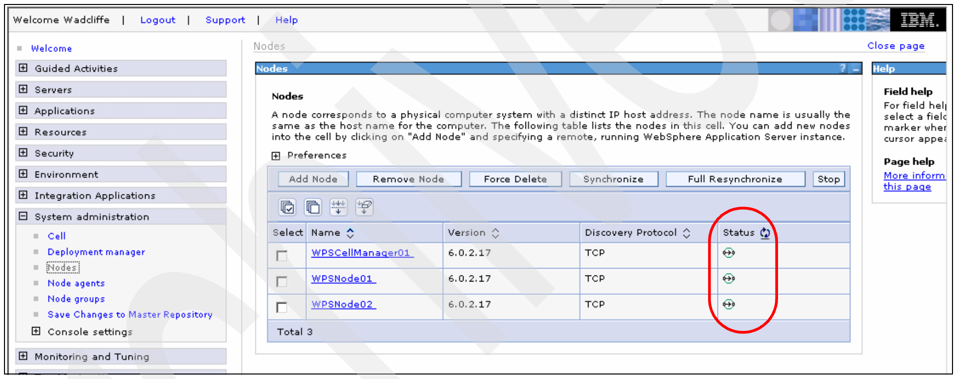
After you have created both nodes, verify that they are running in the Administrative Console by following these steps:

1. Open a browser to `http://wlt0.itso:9060/ibm/console`.

   Note: You need to logout from any existing console sessions.

2. Navigate to **System administration** → **Nodes** and verify that all nodes are running as indicated by the green symbol next to each node, as shown in Figure 8-15.

   Note: Upon initial login to the Administrative Console, you might need to change the Task filter selection to *All* and click **Apply** to view all available functionality.



Nodes

A node corresponds to a physical computer system with a distinct IP host address. The node name is usually the same as the host name for the computer. The following table lists the nodes in this cell. You can add new nodes into the cell by clicking on "Add Node" and specifying a remote, running WebSphere Application Server instance.

Select	Name	Version	Discovery Protocol	Status
☐	WPSCellManager01	6.0.2.17	TCP	●
☐	WPSNode01	6.0.2.17	TCP	●
☐	WPSNode02	6.0.2.17	TCP	●

Total 3

Figure 8-15 Successful node installation and startup

At this point, you have a WebSphere Process Server cell with the configuration as shown in Figure 8-16.

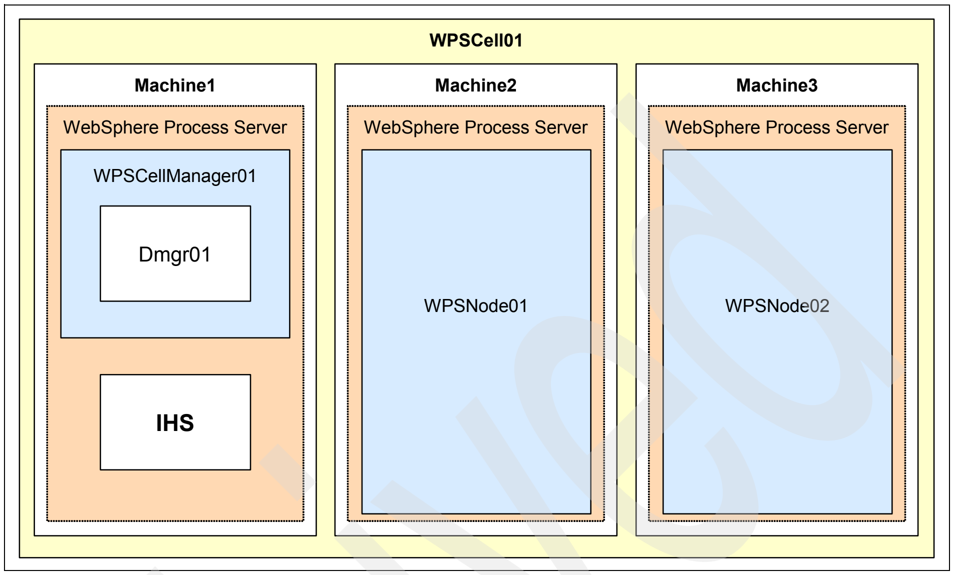


Figure 8-16 Cell and nodes defined

8.3 Specific configuration steps for the Loosely Coupled topology

This section describes the following:

- Configuring database connectivity for the Loosely Coupled topology
- Configuring the cell
- Verifying the topology

8.3.1 Configuring database connectivity for the Loosely Coupled topology

This section describes how to connect successfully to the databases that are required by WebSphere Process Server in the Loosely Coupled topology, including the following databases:

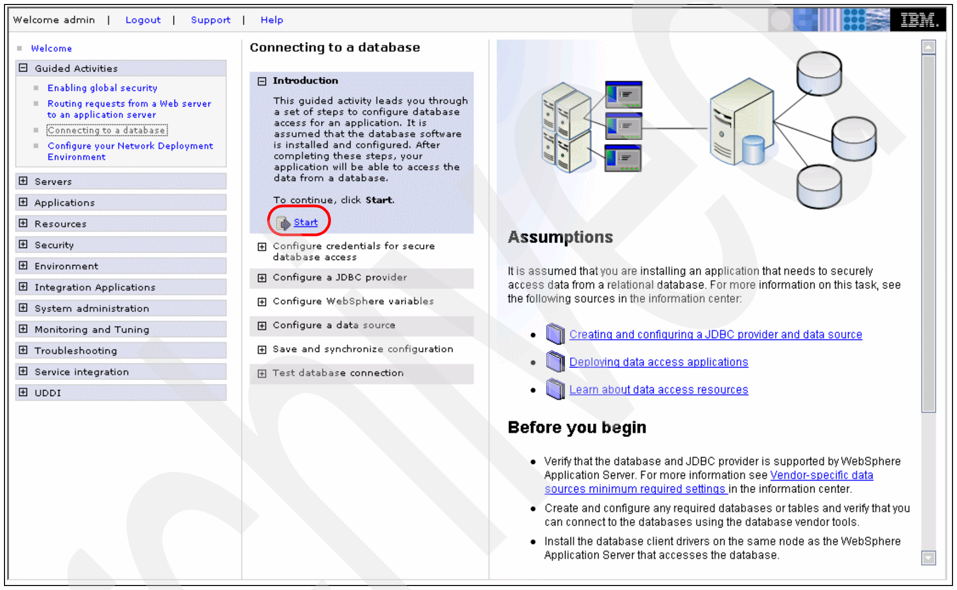
- MEDB - Messaging database
- BPEDB - Business container database

Upon cell creation, the data source jdbc/WPSDB is created automatically, along with the data source provider at cell scope, DB2 Universal JDBC Driver Provider (XA). You can reuse this data source provider. Follow these steps:

1. Open the cell Administrative Console on `wlt0` and login. We used the following URL:

   `http://wlt0.itso:9060/ibm/console`

2. Navigate to **Guided Activities** → **Connecting to a database** and click **Start**, as shown in Figure 8-17.



Welcome admin | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Connecting to a database

Introduction

This guided activity leads you through a set of steps to configure database access for an application. It is assumed that the database software is installed and configured. After completing these steps, your application will be able to access the data from a database.

To continue, click **Start**.

Start

- Configure credentials for secure database access
- Configure a JDBC provider
- Configure WebSphere variables
- Configure a data source
- Save and synchronize configuration
- Test database connection

Assumptions

It is assumed that you are installing an application that needs to securely access data from a relational database. For more information on this task, see the following sources in the information center:

- Creating and configuring a JDBC provider and data source
- Deploying data access applications
- Learn about data access resources

Before you begin

- Verify that the database and JDBC provider is supported by WebSphere Application Server. For more information see Vendor-specific data sources minimum required settings in the information center.
- Create and configure any required databases or tables and verify that you can connect to the databases using the database vendor tools.
- Install the database client drivers on the same node as the WebSphere Application Server that accesses the database.

Figure 8-17 Starting the guided database activity

3. Select **Click to perform**.

Authenticating the alias

You need to set the user and password values for database access. Because you are creating all databases in the same DB2 instance (`wltinst1`) with the same user (`wltinst1`), you can reuse the existing authentication alias. You created this alias when you configured the deployment manager. By default, it is `WPSDBAlias`. To set the user and password value, select **Next step** to continue to the Configure a JDBC provider panel.

Creating a JDBC provider

The next step creates a JDBC provider for use by the data sources later. Again you reuse the existing provider called `DB2 Universal JDBC Driver Provider`

(XA) that you defined at cell scope. To create a JDBC provider, select **Next step** to continue to the Configure WebSphere variables panel.

Setting the WebSphere variables

The JDBC provider uses WebSphere environment variables to identify the driver file locations. For these variables to resolve, you need to set their vales appropriately. The scope for these variables should be **Cell** or **Node**, as shown in Table 8-5 and Table 8-6.

Note: When setting the scope, the selection is not applied until you click **Apply**.

To set the WebSphere variables:

1. Select **Click to perform**.
2. Clear the node and server entries, and click **Apply** to set the active scope to the cell.
3. Create or modify each environment variable, and set the value as defined in Table 8-5.

Table 8-5 Cell scope environment variables

Name	Value	Scope
UNIVERSAL_JDBC_DRIVER_PATH	c:\db2client	Cell
DB2UNIVERSAL_JDBC_DRIVER_PATH	c:\db2client	Cell
DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH	c:\db2client	Cell

4. When you have defined all three variables at cell level, click **Browse Nodes**. Select **WPSNode01** and click **OK**. Click **Apply** to set the active scope to the node.
5. Set the value of the node scope environment variable DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH, as shown in Table 8-6. Also delete the other two variables shown in Table 8-6.

   Attention: At the node scope, you *must* delete the environment variables UNIVERSAL_JDBC_DRIVER_PATH and DB2UNIVERSAL_JDBC_DRIVER_PATH.

Table 8-6 Node scope environment variables

Name	Value	Scope
DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH	c:\db2client	WPSNode01 WPSNode02
UNIVERSAL_JDBC_DRIVER_PATH	*Delete this variable*	WPSNode01 WPSNode02
DB2UNIVERSAL_JDBC_DRIVER_PATH	*Delete this variable*	WPSNode01 WPSNode02

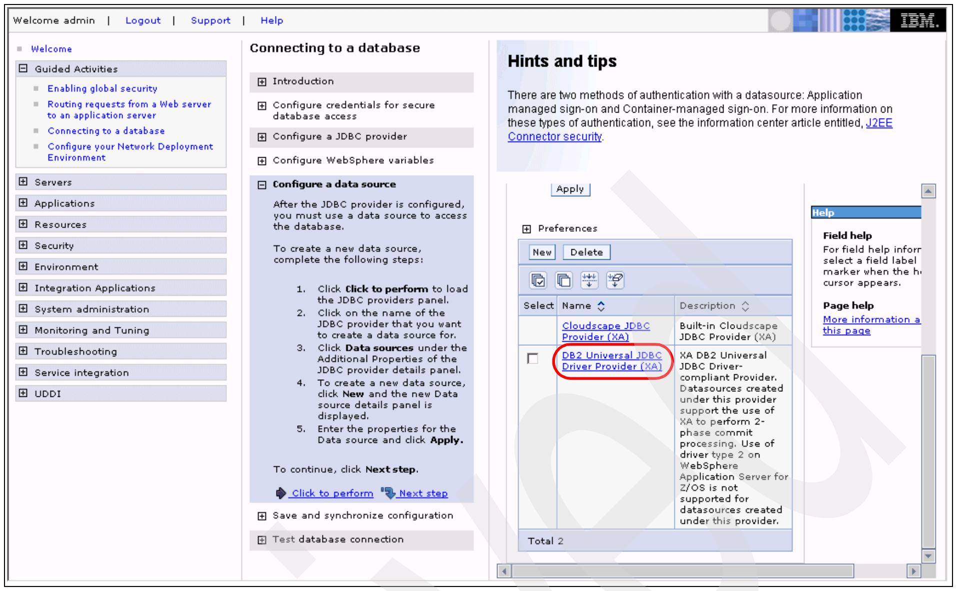
6. Repeat the process for `WPSNode02` to remove the two environment variables and set the third to the value shown in Table 8-6.
7. Save and synchronize these changes.
8. Click **Next step**.

Configuring a data source

The next step is to configure the data sources using the existing JDBC provider that you just created:

1. Select **Click to perform**.
2. In the new panel, click the JDBC provider that is defined at the cell scope **DB2 Universal JDBC Driver Provider (XA)**, as shown in Figure 8-18.

Note: To change the scope to Cell, remove any values in the Node and Server scope text boxes and click **Apply**.


Welcome admin | Logout | Support | Help

Welcome
Guided Activities
- Enabling global security
- Routing requests from a Web server to an application server
- Connecting to a database
- Configure your Network Deployment Environment

Servers
Applications
Resources
Security
Environment
Integration Applications
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI

Connecting to a database

Introduction
Configure credentials for secure database access
Configure a JDBC provider
Configure WebSphere variables
Configure a data source

After the JDBC provider is configured, you must use a data source to access the database.

To create a new data source, complete the following steps:

1. Click **Click to perform** to load the JDBC providers panel.
2. Click on the name of the JDBC provider that you want to create a data source for.
3. Click **Data sources** under the Additional Properties of the JDBC provider details panel.
4. To create a new data source, click **New** and the new Data source details panel is displayed.
5. Enter the properties for the Data source and click **Apply.**

To continue, click **Next step**.

Click to perform   Next step

Save and synchronize configuration
Test database connection

Hints and tips

There are two methods of authentication with a datasource: Application managed sign-on and Container-managed sign-on. For more information on these types of authentication, see the information center article entitled, J2EE Connector security.

Apply

Preferences

New   Delete

Select	Name	Description
	Cloudscape JDBC Provider (XA)	Built-in Cloudscape JDBC Provider (XA)
☐	DB2 Universal JDBC Driver Provider (XA)	XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on WebSphere Application Server for Z/OS is not supported for datasources created under this provider.

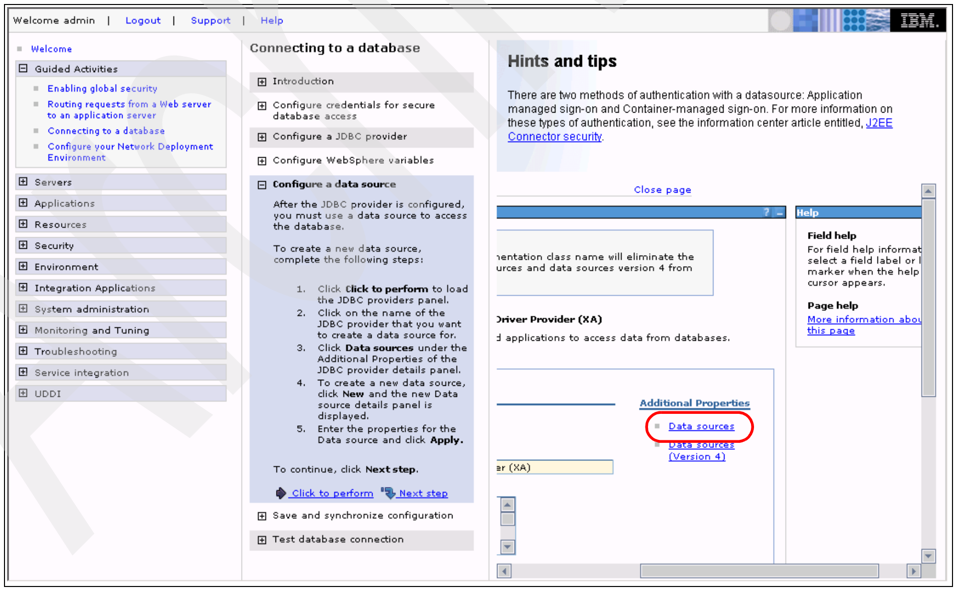
Total 2

Help
Field help
For field help inforr select a field label marker when the h cursor appears.

Page help
More information a this page

Figure 8-18 JDBC provider selection for data source creation

3. Scroll to the right and click **Data sources,** as shown in Figure 8-19.


Welcome admin | Logout | Support | Help

Welcome
Guided Activities
- Enabling global security
- Routing requests from a Web server to an application server
- Connecting to a database
- Configure your Network Deployment Environment

Servers
Applications
Resources
Security
Environment
Integration Applications
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI

Connecting to a database

Introduction
Configure credentials for secure database access
Configure a JDBC provider
Configure WebSphere variables
Configure a data source

After the JDBC provider is configured, you must use a data source to access the database.

To create a new data source, complete the following steps:

1. Click **Click to perform** to load the JDBC providers panel.
2. Click on the name of the JDBC provider that you want to create a data source for.
3. Click **Data sources** under the Additional Properties of the JDBC provider details panel.
4. To create a new data source, click **New** and the new Data source details panel is displayed.
5. Enter the properties for the Data source and click **Apply.**

To continue, click **Next step**.

Click to perform   Next step

Save and synchronize configuration
Test database connection

Hints and tips

There are two methods of authentication with a datasource: Application managed sign-on and Container-managed sign-on. For more information on these types of authentication, see the information center article entitled, J2EE Connector security.

Close page

ıentation class name will eliminate the
urces and data sources version 4 from

)river Provider (XA)
l applications to access data from databases.

Additional Properties
- Data sources
- Data sources (Version 4)

er (XA)

Help
Field help
For field help informat select a field label or l marker when the help cursor appears.

Page help
More information abou this page

Figure 8-19 Selecting the additional data source property

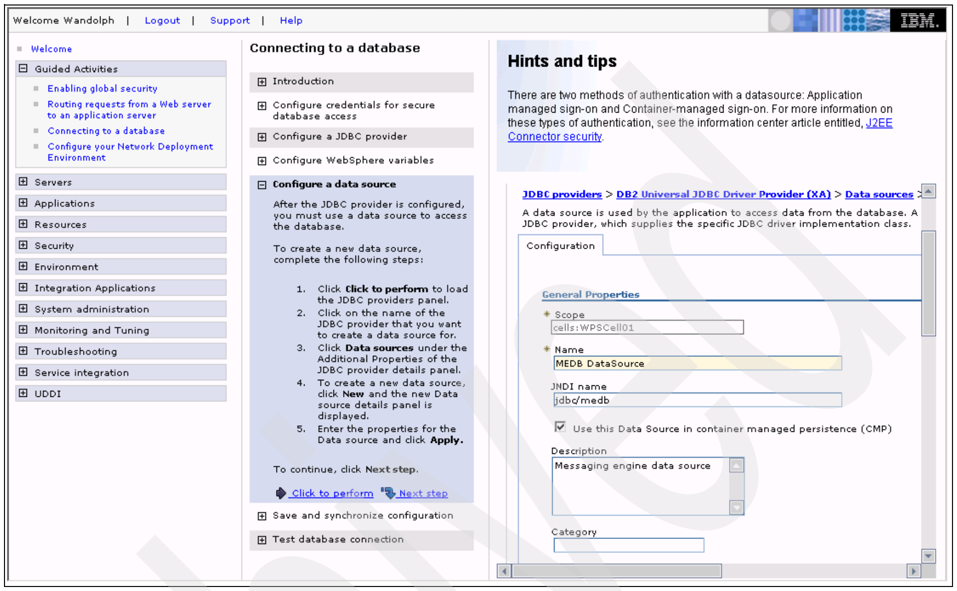
4. You need to create a data source for:
   - Messaging database (MEDB)
   - Business container database (BPEDB)

   Click **New** to create a data source and set the parameters, as shown in Table 8-7.

Table 8-7 Data source configuration

Parameter	**MEDB Data Source Values**	**BPEDB Data Source Values**
Name	MEDB Data Source	BPEDB Data Source
JNDI name	jdbc/medb	jdbc/bpedb
Description	Messaging engine data source	Business process container data source
Component-managed authentication alias	WPSDBAlias	WPSDBAlias
Authentication alias for XA recovery	Use component-managed authentication alias	Use component-managed authentication alias
Database name	MEDB	BPEDB
Driver type	4	4
Server name	db2.itso	db2.itso
Port	50000	50000

Figure 8-20 shows the MEDB data source configuration page. You need to enter all the values that are identified in Table 8-7. To enter these values, you need to scroll down in this panel.



Welcome Wandolph | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Connecting to a database

- Introduction
- Configure credentials for secure database access
- Configure a JDBC provider
- Configure WebSphere variables
- **Configure a data source**

After the JDBC provider is configured, you must use a data source to access the database.

To create a new data source, complete the following steps:

1. Click **Click to perform** to load the JDBC providers panel.
2. Click on the name of the JDBC provider that you want to create a data source for.
3. Click **Data sources** under the Additional Properties of the JDBC provider details panel.
4. To create a new data source, click **New** and the new Data source details panel is displayed.
5. Enter the properties for the Data source and click **Apply.**

To continue, click Next step.

Click to perform   Next step

- Save and synchronize configuration
- Test database connection

Hints and tips

There are two methods of authentication with a datasource: Application managed sign-on and Container-managed sign-on. For more information on these types of authentication, see the information center article entitled, J2EE Connector security.

JDBC providers > DB2 Universal JDBC Driver Provider (XA) > Data sources >

A data source is used by the application to access data from the database. A JDBC provider, which supplies the specific JDBC driver implementation class.

Configuration

General Properties

* Scope: cells:WPSCell01

* Name: MEDB DataSource

JNDI name: jdbc/medb

✓ Use this Data Source in container managed persistence (CMP)

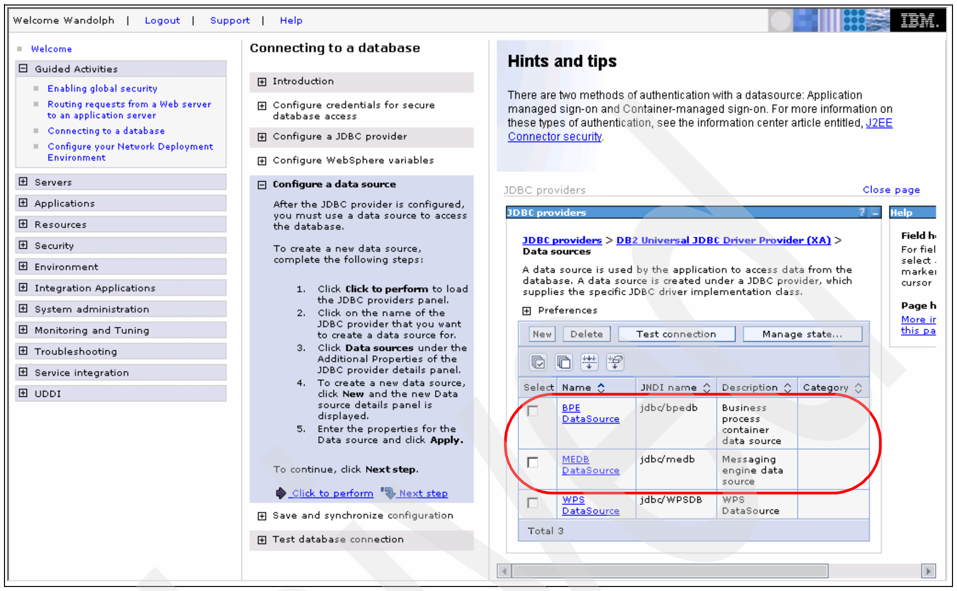
Description: Messaging engine data source

Category:

Figure 8-20   Data source configuration

5. Click **OK** to create the new data source. When created, save and synchronize the changes and click **OK** to complete.

6. Repeat this process for the BPEDB data source. Figure 8-21 shows all the data sources.


Welcome Wandolph | Logout | Support | Help

Connecting to a database

- Introduction
- Configure credentials for secure database access
- Configure a JDBC provider
- Configure WebSphere variables
- Configure a data source

After the JDBC provider is configured, you must use a data source to access the database.

To create a new data source, complete the following steps:

1. Click **Click to perform** to load the JDBC providers panel.
2. Click on the name of the JDBC provider that you want to create a data source for.
3. Click **Data sources** under the Additional Properties of the JDBC provider details panel.
4. To create a new data source, click **New** and the new Data source details panel is displayed.
5. Enter the properties for the Data source and click **Apply**.

To continue, click **Next step**.

Click to perform | Next step

- Save and synchronize configuration
- Test database connection

Hints and tips

There are two methods of authentication with a datasource: Application managed sign-on and Container-managed sign-on. For more information on these types of authentication, see the information center article entitled, J2EE Connector security.

JDBC providers > DB2 Universal JDBC Driver Provider (XA) > Data sources

A data source is used by the application to access data from the database. A data source is created under a JDBC provider, which supplies the specific JDBC driver implementation class.

Select	Name	JNDI name	Description	Category
☐	BPE DataSource	jdbc/bpedb	Business process container data source	
☐	MEDB DataSource	jdbc/medb	Messaging engine data source	
☐	WPS DataSource	jdbc/WPSDB	WPS DataSource	

Total 3

Figure 8-21 Data sources created

7. Click **Next step** and then **Next step** again in the Save and synchronize configuration panel (because you have already saved and synchronized).

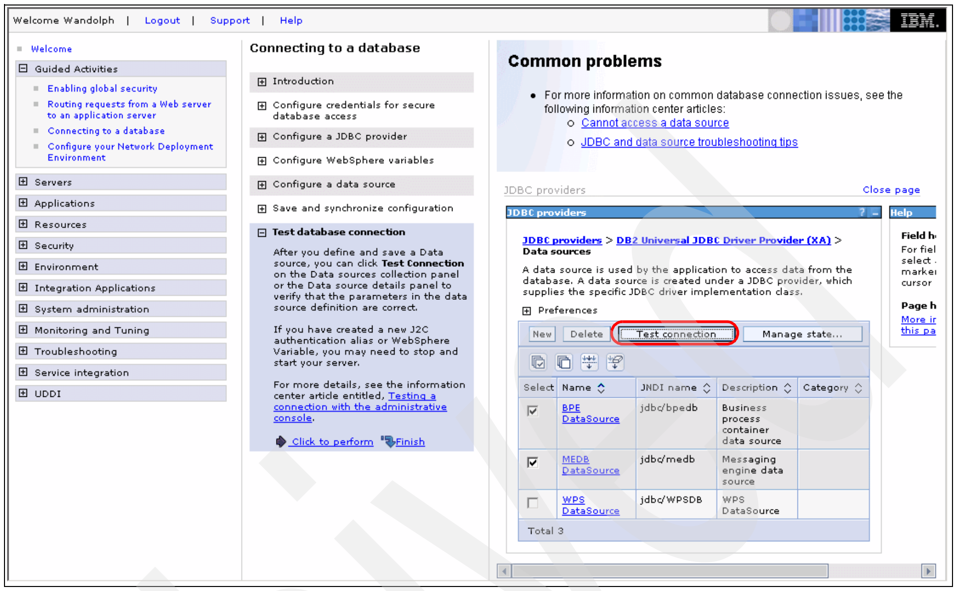
Testing database connectivity

Note: Because the data sources are reusing the existing authentication alias, WPSDBAlias, there is no need to restart the environment. You can test these data sources immediately.

Ensure that the MEDB and BPEDB databases are running and test the newly created data sources:

1. Select **Click to perform**.
2. Click the JDBC provider that is defined at the cell scope **DB2 Universal JDBC Driver Provider (XA)**.
3. Scroll to the right and click **Data sources**.

4. Select the check box next to the new data sources as shown in Figure 8-22 and click **Test connection**.


Welcome Wandolph | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Connecting to a database

- Introduction
- Configure credentials for secure database access
- Configure a JDBC provider
- Configure WebSphere variables
- Configure a data source
- Save and synchronize configuration
- **Test database connection**

After you define and save a Data source, you can click **Test Connection** on the Data sources collection panel or the Data source details panel to verify that the parameters in the data source definition are correct.

If you have created a new J2C authentication alias or WebSphere Variable, you may need to stop and start your server.

For more details, see the information center article entitled, Testing a connection with the administrative console.

Click to perform Finish

Common problems

- For more information on common database connection issues, see the following information center articles:
  - Cannot access a data source
  - JDBC and data source troubleshooting tips

JDBC providers Close page

JDBC providers

JDBC providers > DB2 Universal JDBC Driver Provider (XA) > **Data sources**

A data source is used by the application to access data from the database. A data source is created under a JDBC provider, which supplies the specific JDBC driver implementation class.

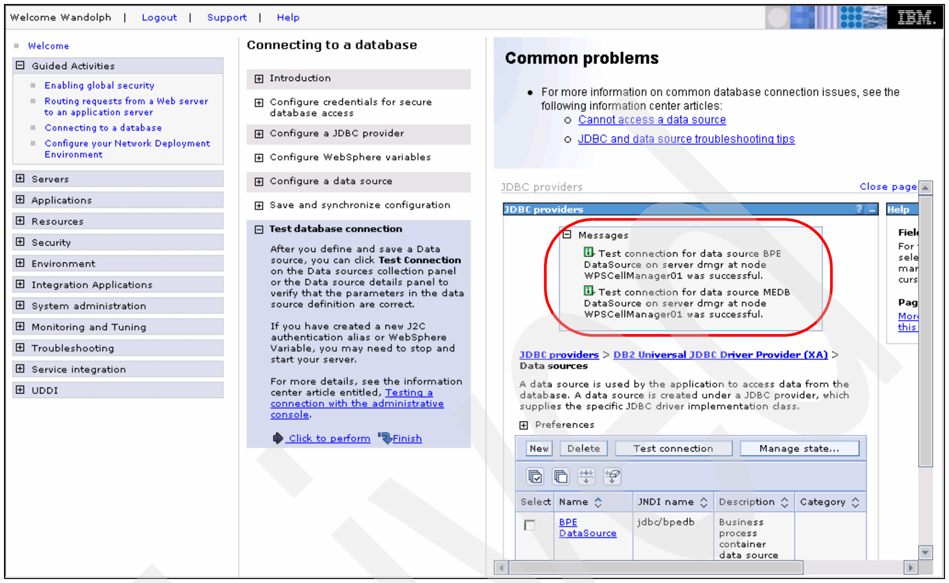
Preferences

Select	Name	JNDI name	Description	Category
✓	BPE DataSource	jdbc/bpedb	Business process container data source	
✓	MEDB DataSource	jdbc/medb	Messaging engine data source	
	WPS DataSource	jdbc/WPSDB	WPS DataSource	

Total 3

Figure 8-22 Testing the new data source

5. The Messages dialog box (at the top in Figure 8-23) should indicate that the data source successfully connected to the database.



Connecting to a database

- Introduction
- Configure credentials for secure database access
- Configure a JDBC provider
- Configure WebSphere variables
- Configure a data source
- Save and synchronize configuration
- **Test database connection**

After you define and save a Data source, you can click **Test Connection** on the Data sources collection panel or the Data source details panel to verify that the parameters in the data source definition are correct.

If you have created a new J2C authentication alias or WebSphere Variable, you may need to stop and start your server.

For more details, see the information center article entitled, Testing a connection with the administrative console.

Click to perform Finish

Common problems

- For more information on common database connection issues, see the following information center articles:
  - Cannot access a data source
  - JDBC and data source troubleshooting tips

JDBC providers Close page

JDBC providers

Messages

Test connection for data source BPE DataSource on server dmgr at node WPSCellManager01 was successful.

Test connection for data source MEDB DataSource on server dmgr at node WPSCellManager01 was successful.

JDBC providers > DB2 Universal JDBC Driver Provider (XA) >
Data sources

A data source is used by the application to access data from the database. A data source is created under a JDBC provider, which supplies the specific JDBC driver implementation class.

Preferences

New | Delete | Test connection | Manage state...

Select	Name	JNDI name	Description	Category
☐	BPE DataSource	jdbc/bpedb	Business process container data source	

Figure 8-23 Successful database connections

6. Click **Finish**.

8.3.2 Configuring the cell

To configure the network deployment infrastructure with the options that are required for the Loosely Coupled topology. Follow these steps:

1. Open the Administrative Console on `w1t0` and login.
2. Navigate to **Guided Activities** → **Configure your Network Deployment Environment** and click **Start**.
3. Scroll down and select **Click to perform** in the expanded panel to see the list of nodes.
4. The nodes are already federated, so click **Next step**.
5. The database configuration is already complete, so click **Next step**.
6. In the Creating clusters and cluster members panel, select **Click to perform**.

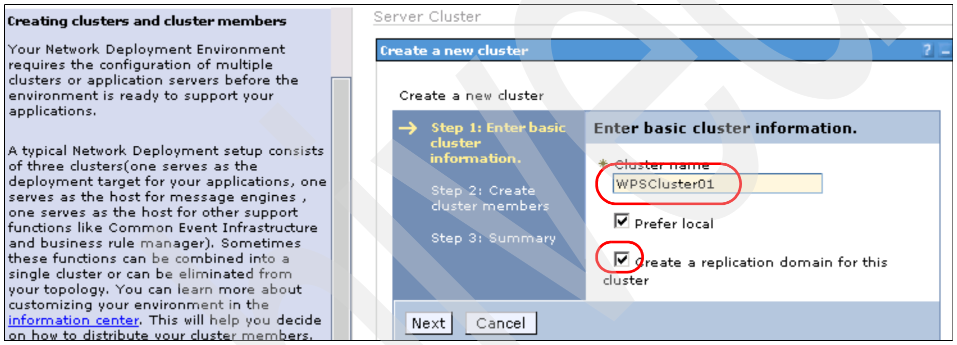
7. Click **New** and define a new cluster (Figure 8-24).

   a. Enter a Cluster name of `WPSCluster01`.

   b. Select **Prefer local**. This indicates that, if possible, Enterprise JavaBean (EJB) requests are routed to the client node. After you enable this feature, performance is improved because client requests are sent to local enterprise beans.

   c. Select **Create a replication domain for this cluster**. Use replication domains to transfer data, objects, or events for session manager, dynamic cache, or stateful session beans among the application servers in a cluster.

   d. Click **Next**.



Creating clusters and cluster members

Your Network Deployment Environment requires the configuration of multiple clusters or application servers before the environment is ready to support your applications.

A typical Network Deployment setup consists of three clusters(one serves as the deployment target for your applications, one serves as the host for message engines , one serves as the host for other support functions like Common Event Infrastructure and business rule manager). Sometimes these functions can be combined into a single cluster or can be eliminated from your topology. You can learn more about customizing your environment in the information center. This will help you decide on how to distribute your cluster members.

Server Cluster

Create a new cluster

Create a new cluster

→ **Step 1: Enter basic cluster information.**

Step 2: Create cluster members

Step 3: Summary

Enter basic cluster information.

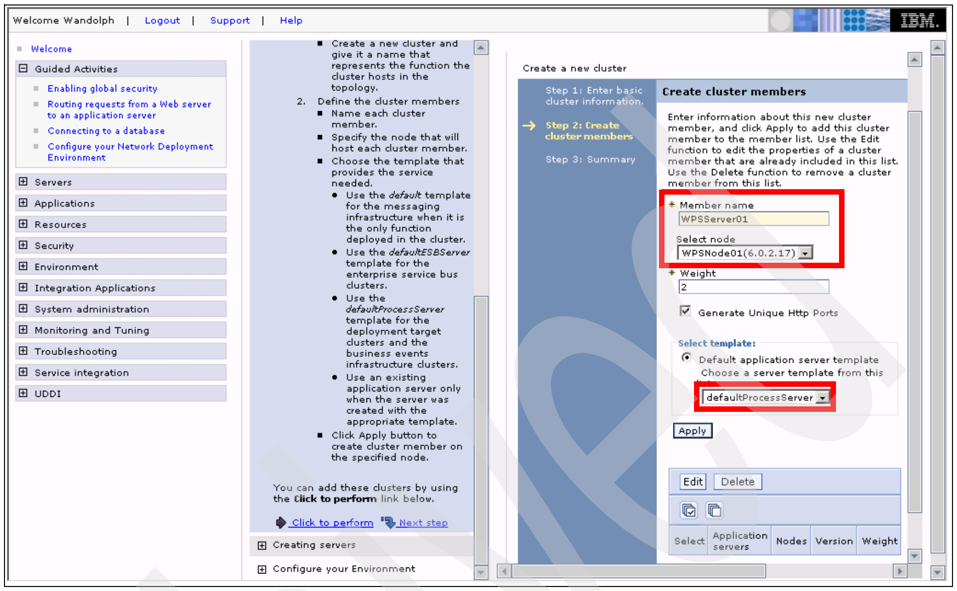
* Cluster name: WPSCluster01

[✓] Prefer local

[✓] Create a replication domain for this cluster

Figure 8-24 Specify the basic cluster information

8. Add a first member of the cluster named `WPSServer01` to the node `WPSNode01`. Set the template to `defaultProcessServer`, as shown in Figure 8-25.



Welcome Wandolph | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

- Create a new cluster and give it a name that represents the function the cluster hosts in the topology.

2. Define the cluster members
   - Name each cluster member.
   - Specify the node that will host each cluster member.
   - Choose the template that provides the service needed.
     - Use the *default* template for the messaging infrastructure when it is the only function deployed in the cluster.
     - Use the *defaultESBServer* template for the enterprise service bus clusters.
     - Use the *defaultProcessServer* template for the deployment target clusters and the business events infrastructure clusters.
     - Use an existing application server only when the server was created with the appropriate template.
   - Click Apply button to create cluster member on the specified node.

You can add these clusters by using the **Click to perform** link below.

Click to perform | Next step

Creating servers

Configure your Environment

Create a new cluster

Step 1: Enter basic cluster information.

→ Step 2: Create cluster members

Step 3: Summary

Create cluster members

Enter information about this new cluster member, and click Apply to add this cluster member to the member list. Use the Edit function to edit the properties of a cluster member that are already included in this list. Use the Delete function to remove a cluster member from this list.

* Member name: WPSServer01

Select node: WPSNode01(6.0.2.17)

* Weight: 2

☑ Generate Unique Http Ports

Select template:

● Default application server template

Choose a server template from this list: defaultProcessServer

Apply

Edit | Delete

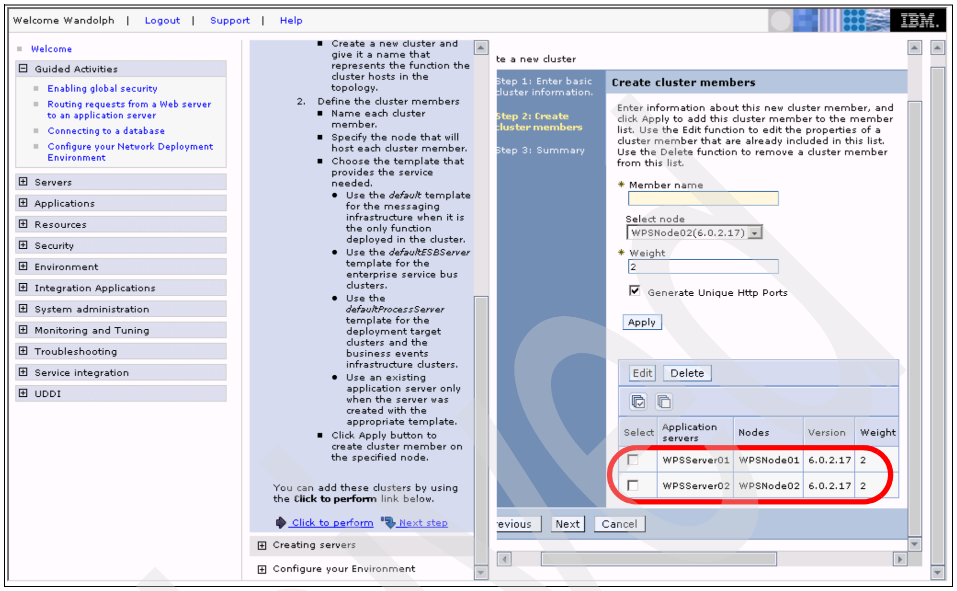
Select	Application servers	Nodes	Version	Weight

Figure 8-25 Adding the first cluster member

9. Click **Apply** to add this server to the new cluster.

10. Repeat this process for `WPSServer02` on `WPSNode02`. However, the template option will not be available now because all cluster members are of the type that is defined by the first member.

11. Click **Apply** to add the second cluster member and then click **Next**, as shown in Figure 8-26.


Welcome Wandolph | Logout | Support | Help

Welcome

Guided Activities
- Enabling global security
- Routing requests from a Web server to an application server
- Connecting to a database
- Configure your Network Deployment Environment

Servers
Applications
Resources
Security
Environment
Integration Applications
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI

- Create a new cluster and give it a name that represents the function the cluster hosts in the topology.

2. Define the cluster members
   - Name each cluster member.
   - Specify the node that will host each cluster member.
   - Choose the template that provides the service needed.
     - Use the *default* template for the messaging infrastructure when it is the only function deployed in the cluster.
     - Use the *defaultESBServer* template for the enterprise service bus clusters.
     - Use the *defaultProcessServer* template for the deployment target clusters and the business events infrastructure clusters.
     - Use an existing application server only when the server was created with the appropriate template.
   - Click Apply button to create cluster member on the specified node.

You can add these clusters by using the **Click to perform** link below.

Click to perform   Next step

Creating servers

Configure your Environment

te a new cluster

Step 1: Enter basic cluster information.

Step 2: Create cluster members

Step 3: Summary

Create cluster members

Enter information about this new cluster member, and click Apply to add this cluster member to the member list. Use the Edit function to edit the properties of a cluster member that are already included in this list. Use the Delete function to remove a cluster member from this list.

* Member name

Select node
WPSNode02(6.0.2.17)

* Weight
2

☑ Generate Unique Http Ports

Apply

Edit   Delete

Select	Application servers	Nodes	Version	Weight
☐	WPSServer01	WPSNode01	6.0.2.17	2
☐	WPSServer02	WPSNode02	6.0.2.17	2

evious   Next   Cancel

Figure 8-26 Cluster members added

12. At the summary page, click **Finish**.

13. Save and synchronize the changes.

You have now created the cluster WPSCluster01. Figure 8-27 shows the current topology.

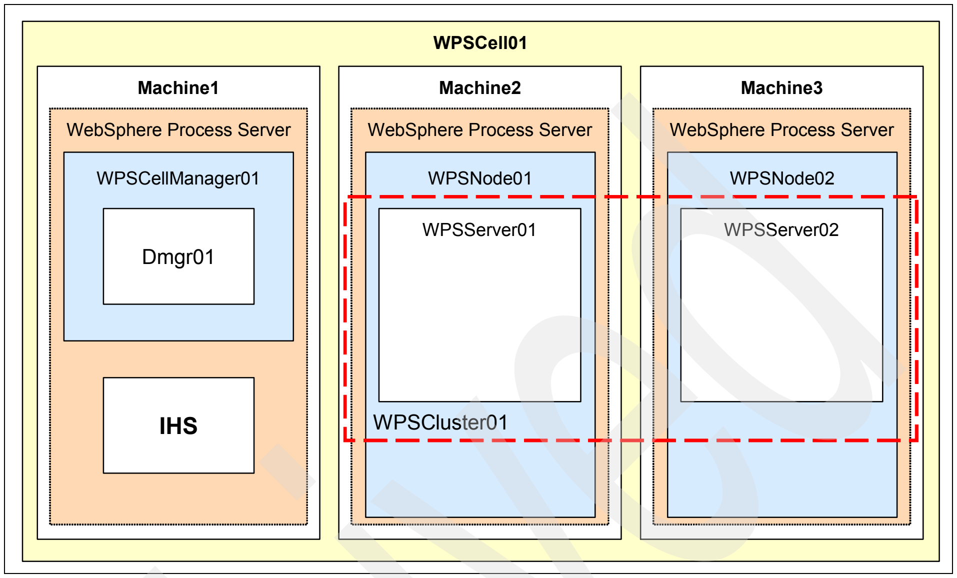


Figure 8-27 The logical view of the cell with cluster created

14. Click **Next step** to continue and then **Next step** at the Creating servers option because this example does not require more servers.

Configuring the environment

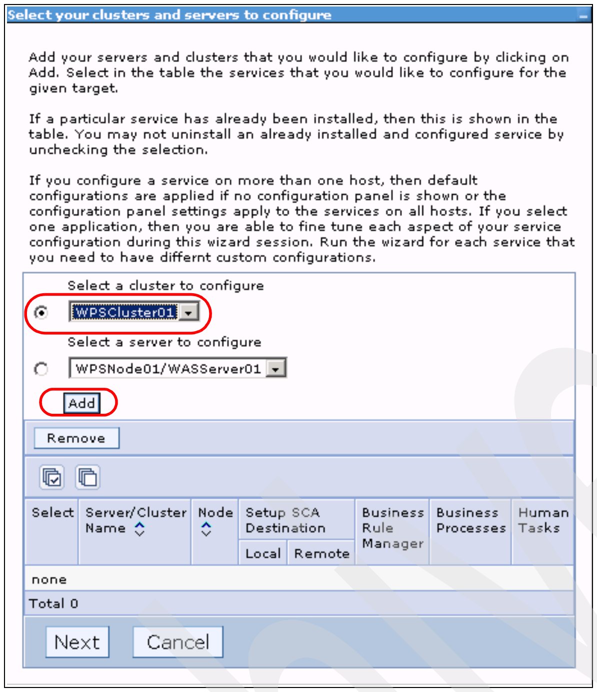
In this section, we describe how to configure SCA, BPC, and HTM for the cluster WPSCluster01.

Configuring the cluster for SCA

To configure the cluster for SCA:

1. In the Configure your Environment panel, select **Click to perform.**
2. Click the pull-down menu to configure a cluster.

3. Select **WPSCluster01** and click **Add** (Figure 8-28).


Select your clusters and servers to configure

Add your servers and clusters that you would like to configure by clicking on Add. Select in the table the services that you would like to configure for the given target.

If a particular service has already been installed, then this is shown in the table. You may not uninstall an already installed and configured service by unchecking the selection.

If you configure a service on more than one host, then default configurations are applied if no configuration panel is shown or the configuration panel settings apply to the services on all hosts. If you select one application, then you are able to fine tune each aspect of your service configuration during this wizard session. Run the wizard for each service that you need to have different custom configurations.

Select a cluster to configure

(●) WPSCluster01

Select a server to configure

( ) WPSNode01/WASServer01

Add

Remove

Select	Server/Cluster Name	Node	Setup SCA Destination		Business Rule Manager	Business Processes	Human Tasks
			Local	Remote			
none							

Total 0

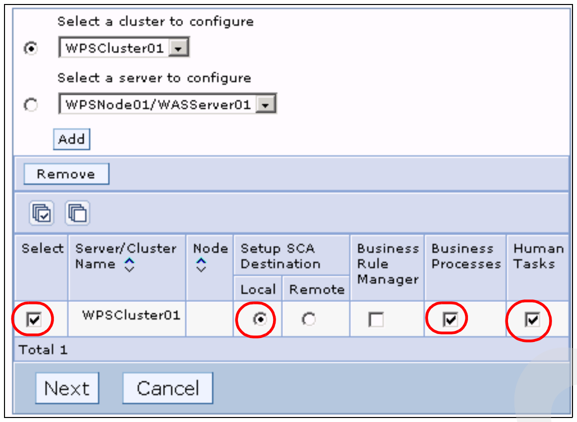
Next Cancel

Figure 8-28 Select the cluster to be configured

4. In the configuration table, select the components that you need to configure for the cluster. You configure SCA, BPC and HTM for cluster WPSCluster01:

   – Select **Local** for the Setup SCA Destination
   – Select **Business Processes**
   – Select **Human Tasks**

Figure 8-29 shows these options. When set, click **Next**.



(●) Select a cluster to configure: WPSCluster01

( ) Select a server to configure: WPSNode01/WASServer01

Add

Remove

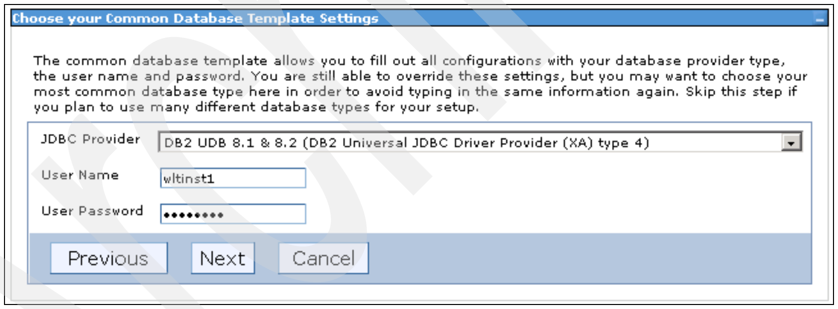
Select	Server/Cluster Name	Node	Setup SCA Destination		Business Rule Manager	Business Processes	Human Tasks
			Local	Remote			
✓	WPSCluster01		●	○	☐	✓	✓

Total 1

Next Cancel

Figure 8-29 Select the component to be configured on the cluster

5. From the JDBC provider drop-down list, select the appropriate provider. In this example, the provider is **DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC Driver Provider (XA) type 4)**.
6. Set the user name to `wltinst1` and the user password, and click **Next**, as shown in Figure 8-30.



Choose your Common Database Template Settings

The common database template allows you to fill out all configurations with your database provider type, the user name and password. You are still able to override these settings, but you may want to choose your most common database type here in order to avoid typing in the same information again. Skip this step if you plan to use many different database types for your setup.

JDBC Provider	DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC Driver Provider (XA) type 4)
User Name	wltinst1
User Password	••••••••

Previous Next Cancel

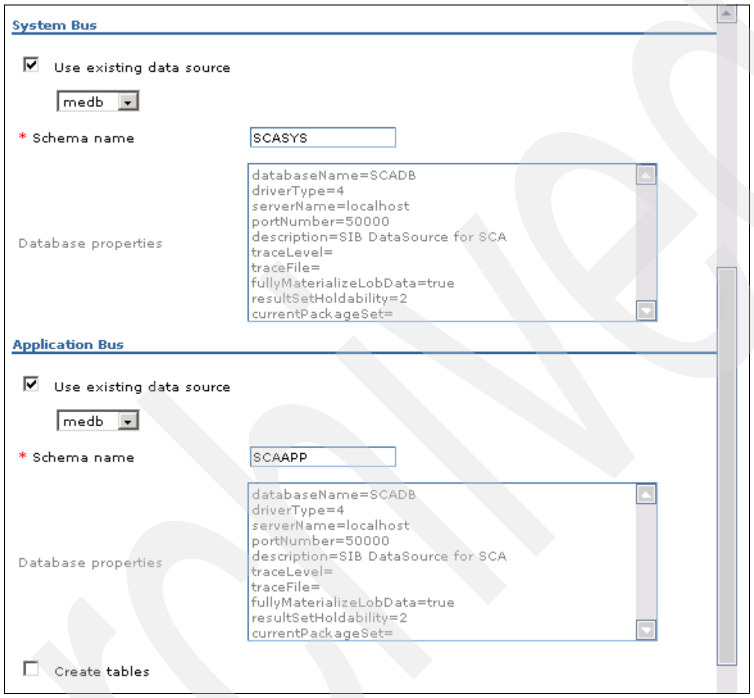
Figure 8-30 JDBC provider selection

7. Under System Bus, select **Use existing data source**. In the drop-down menu, select the data source **medb**.
8. Set the schema name to `SCASYS`.
9. Under Application Bus, select **Use existing data source**. In the drop-down menu, select the data source **medb**.
10. Set the schema name to `SCAAPP`.

11.Clear **Create tables** because you created tables manually with the **`sibDDLgenerator.bat`** command.

Note: If the database is DB2, this option creates the SCA database schemas the first time that you start the SCA messaging engines.

Figure 8-31 shows these settings.


System Bus

[✓] Use existing data source

medb

* Schema name SCASYS

Database properties

```
databaseName=SCADB
driverType=4
serverName=localhost
portNumber=50000
description=SIB DataSource for SCA
traceLevel=
traceFile=
fullyMaterializeLobData=true
resultSetHoldability=2
currentPackageSet=
```

Application Bus

[✓] Use existing data source

medb

* Schema name SCAAPP

Database properties

```
databaseName=SCADB
driverType=4
serverName=localhost
portNumber=50000
description=SIB DataSource for SCA
traceLevel=
traceFile=
fullyMaterializeLobData=true
resultSetHoldability=2
currentPackageSet=
```

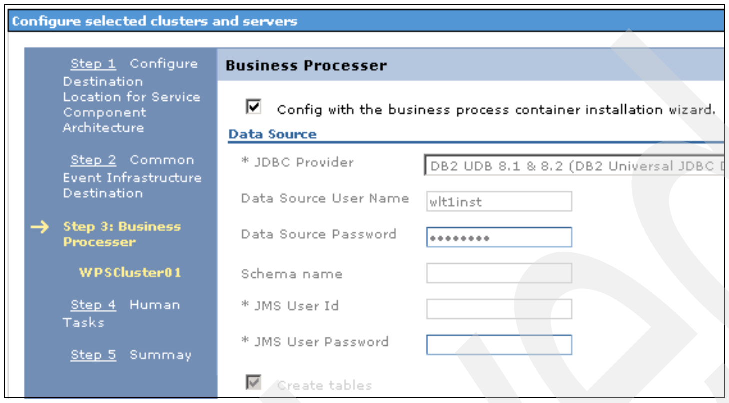
[ ] Create tables

Figure 8-31 SCA values

12.Click **Next**.

13.Specify the **Common Event Infrastructure** (CEI) related configuration information. In this topology, there is no CEI configured, so here you can just click **Next**.

14. Configure the Business Process Container by selecting **Configure with the business process container installation wizard** (Figure 8-32). You use the external installation wizard to configure BPC, because the wizard exposes more options than this panel. Click **Next**.


Configure selected clusters and servers

Step 1 Configure Destination Location for Service Component Architecture
Step 2 Common Event Infrastructure Destination
→ Step 3: Business Processer
WPSCluster01
Step 4 Human Tasks
Step 5 Summay

Business Processer

☑ Config with the business process container installation wizard.

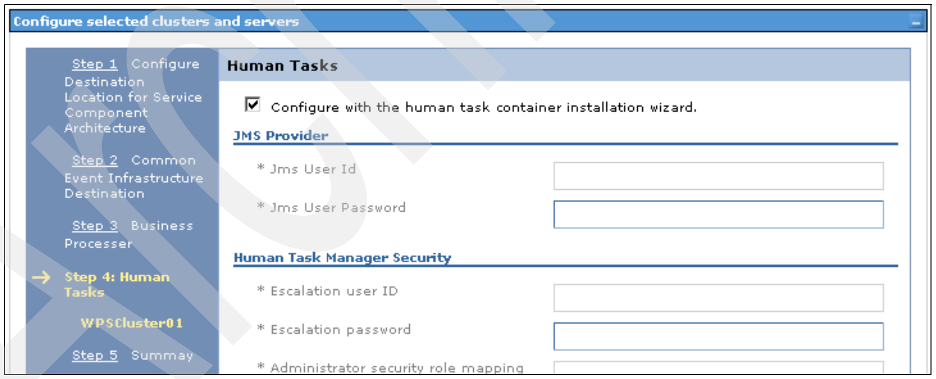
Data Source

* JDBC Provider	DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC D
Data Source User Name	wlt1inst
Data Source Password	••••••••
Schema name	
* JMS User Id	
* JMS User Password	

☑ Create tables

Figure 8-32 Specify the BPC configuration information

15. Configure the Human Task Container by selecting **Configure with the human task container installation wizard** (Figure 8-33). You use the external installation wizard to configure the Human Task Container, because the wizard exposes more options than this panel. Click **Next**.


Configure selected clusters and servers

Step 1 Configure Destination Location for Service Component Architecture
Step 2 Common Event Infrastructure Destination
Step 3 Business Processer
→ Step 4: Human Tasks
WPSCluster01
Step 5 Summay

Human Tasks

☑ Configure with the human task container installation wizard.

JMS Provider

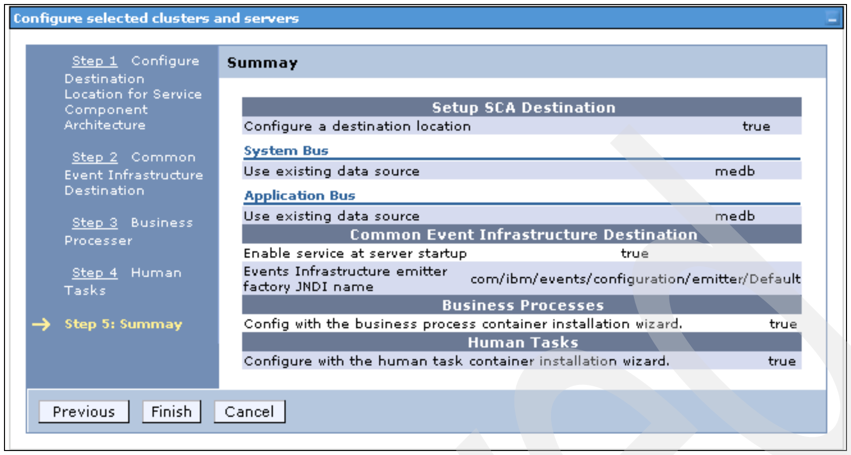
* Jms User Id	
* Jms User Password	

Human Task Manager Security

* Escalation user ID	
* Escalation password	
* Administrator security role mapping	

Figure 8-33 Specifying the human task container configuration information

16. Click review the summary information (Figure 8-34) and click **Finish**.


Configure selected clusters and servers
Step 1 Configure Destination Location for Service Component Architecture
Step 2 Common Event Infrastructure Destination
Step 3 Business Processer
Step 4 Human Tasks
Step 5: Summay
Previous
Finish
Cancel

Summay

Setup SCA Destination	
Configure a destination location	true
System Bus	
Use existing data source	medb
Application Bus	
Use existing data source	medb
Common Event Infrastructure Destination	
Enable service at server startup	true
Events Infrastructure emitter factory JNDI name	com/ibm/events/configuration/emitter/Default
Business Processes	
Config with the business process container installation wizard.	true
Human Tasks	
Configure with the human task container installation wizard.	true

Figure 8-34 Summary information

17. WebSphere Process Server completes the configuration for SCA. Wait for the configuration to complete.
18. Click **Save to Master Configuration**.
19. Select **Synchronize changes with Nodes** and click **Save**.
20. Wait for the synchronization to complete and then click **OK**.

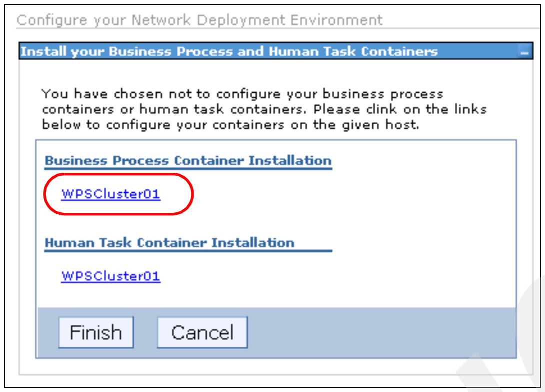
Configuring the cluster for BPC

When configuring the cluster for BPC, you are prompted to Install your Business Process and Human Task Containers.

Note: At this point, do *not* click **Finish.** Otherwise, the process terminates.

To configure the cluster for BPC, follow these steps:

1. Click **WPSCluster01** under **Business Process Container Installation** as shown in Figure 8-35 to launch the external BPC installation wizard.


Configure your Network Deployment Environment

Install your Business Process and Human Task Containers

You have chosen not to configure your business process containers or human task containers. Please clink on the links below to configure your containers on the given host.

Business Process Container Installation

WPSCluster01

Human Task Container Installation

WPSCluster01

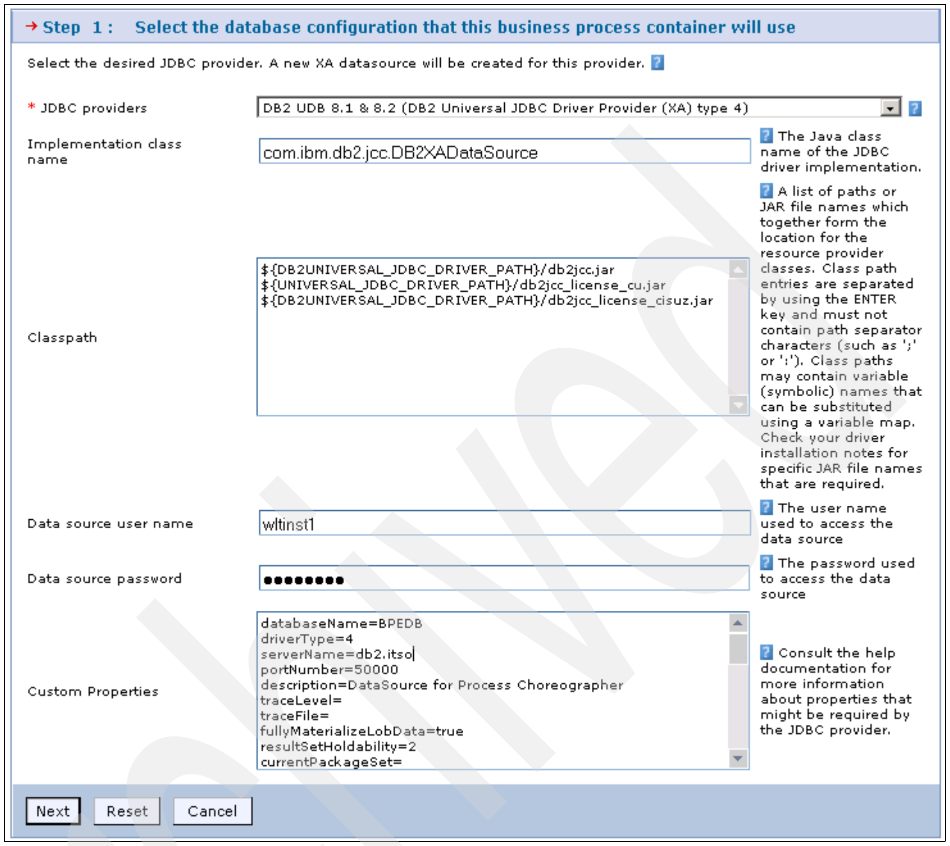
Figure 8-35 Launch the BPC installation wizard

2. Set the values for `BPEDB` as shown in table Table 8-8.

Table 8-8 DB2 settings for BPC wizard

Name	Value
JDBC providers	DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC Driver Provider (XA) type 4)
Data source user name	wltinst1
Data source password	itso4you
databaseName	BPEDB
driverType	4
serverName	db2.itso
portNumber	50000

Figure 8-36 illustrates this configuration.


→ Step 1 : Select the database configuration that this business process container will use

Select the desired JDBC provider. A new XA datasource will be created for this provider.

* JDBC providers	DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC Driver Provider (XA) type 4)	
Implementation class name	com.ibm.db2.jcc.DB2XADataSource	The Java class name of the JDBC driver implementation.
Classpath	${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar ${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar ${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar	A list of paths or JAR file names which together form the location for the resource provider classes. Class path entries are separated by using the ENTER key and must not contain path separator characters (such as ';' or ':'). Class paths may contain variable (symbolic) names that can be substituted using a variable map. Check your driver installation notes for specific JAR file names that are required.
Data source user name	wltinst1	The user name used to access the data source
Data source password	●●●●●●●●	The password used to access the data source

Custom Properties	databaseName=BPEDB
	driverType=4
	serverName=db2.itso
	portNumber=50000
	description=DataSource for Process Choreographer
	traceLevel=
	traceFile=
	fullyMaterializeLobData=true
	resultSetHoldability=2
	currentPackageSet=
		Consult the help documentation for more information about properties that might be required by the JDBC provider.

Figure 8-36 Setting the database configuration from the BPC

3. Click **Next**.

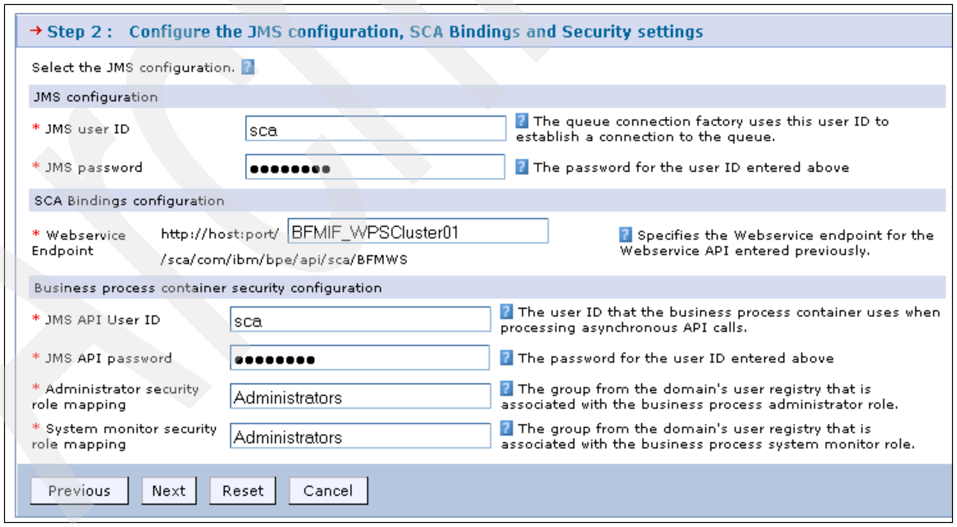
4. Set the values for JMS Configuration, SCA Bindings, and Security, as shown in Table 8-9.

Table 8-9 JMS Configuration settings

Name	Value
JMS user ID	sca
JMS password	itso4you
Webservice Endpoint	BFMIF_WPSCluster01
JMS API User ID	sca
JMS API password	itso4you
Administrator security role mapping	Administrators
System monitor security role mapping	Administrators

Note: You do not enable global security in this scenario, so the `Administrators` group and the `sca` user are dummy values.

Figure 8-37 illustrates the JMS configuration.



→ Step 2 : Configure the JMS configuration, SCA Bindings and Security settings

Select the JMS configuration.

JMS configuration

* JMS user ID	sca	The queue connection factory uses this user ID to establish a connection to the queue.
* JMS password	●●●●●●●●	The password for the user ID entered above

SCA Bindings configuration

* Webservice Endpoint	http://host:port/ BFMIF_WPSCluster01 /sca/com/ibm/bpe/api/sca/BFMWS	Specifies the Webservice endpoint for the Webservice API entered previously.

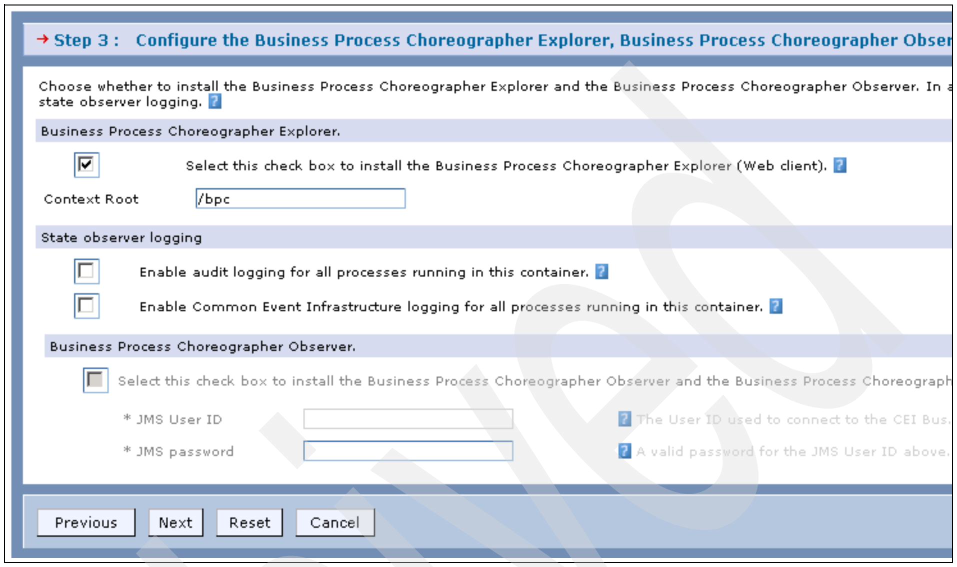
Business process container security configuration

* JMS API User ID	sca	The user ID that the business process container uses when processing asynchronous API calls.
* JMS API password	●●●●●●●●	The password for the user ID entered above
* Administrator security role mapping	Administrators	The group from the domain's user registry that is associated with the business process administrator role.
* System monitor security role mapping	Administrators	The group from the domain's user registry that is associated with the business process system monitor role.

Figure 8-37 Setting the JMS, SCA, and security configuration for the BPC

5. Click **Next**.

6. Provide details for the Business Process Explorer, audit logging, and Common Event Infrastructure. Leave this panel at the defaults, and click **Next** (Figure 8-38).


→ Step 3 : Configure the Business Process Choreographer Explorer, Business Process Choreographer Obser

Choose whether to install the Business Process Choreographer Explorer and the Business Process Choreographer Observer. In a
state observer logging.

Business Process Choreographer Explorer.

[✓] Select this check box to install the Business Process Choreographer Explorer (Web client).

Context Root: /bpc

State observer logging

[ ] Enable audit logging for all processes running in this container.

[ ] Enable Common Event Infrastructure logging for all processes running in this container.

Business Process Choreographer Observer.

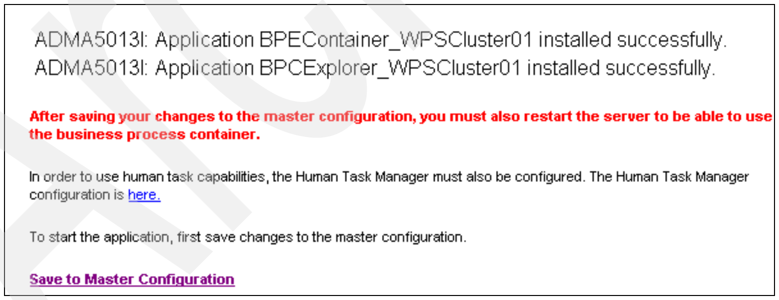
[ ] Select this check box to install the Business Process Choreographer Observer and the Business Process Choreograph

* JMS User ID — The User ID used to connect to the CEI Bus.

* JMS password — A valid password for the JMS User ID above.

Figure 8-38 Other settings for the BPC

7. At the summary panel, click **Finish**.
8. After the installation completes, verify that the configuration is successful (Figure 8-39).


ADMA5013I: Application BPEContainer_WPSCluster01 installed successfully.
ADMA5013I: Application BPCExplorer_WPSCluster01 installed successfully.

After saving your changes to the master configuration, you must also restart the server to be able to use the business process container.

In order to use human task capabilities, the Human Task Manager must also be configured. The Human Task Manager configuration is here.

To start the application, first save changes to the master configuration.

Save to Master Configuration

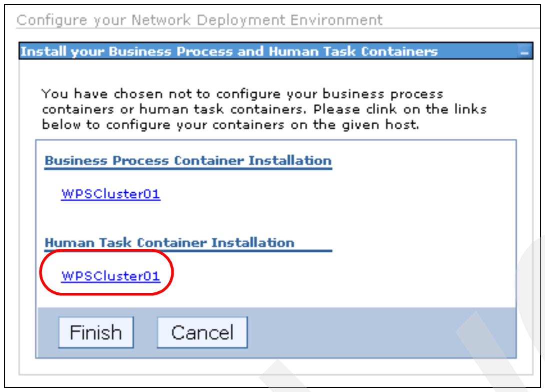
Figure 8-39 Verify the installation of BPC

9. Save and synchronize the changes.
10. The panel returns to Install your Business Process and Human Tasks Container.

Configuring the cluster for the Human Task Manager

To configure the cluster for the Human Task Manager:

1. Under Human Task Container Installation, click **WPSCluster01** to launch the Human Task Manager installation wizard (Figure 8-40).



Configure your Network Deployment Environment

Install your Business Process and Human Task Containers

You have chosen not to configure your business process containers or human task containers. Please clink on the links below to configure your containers on the given host.

Business Process Container Installation

WPSCluster01

Human Task Container Installation

WPSCluster01

Figure 8-40 Launch the HTM installation wizard

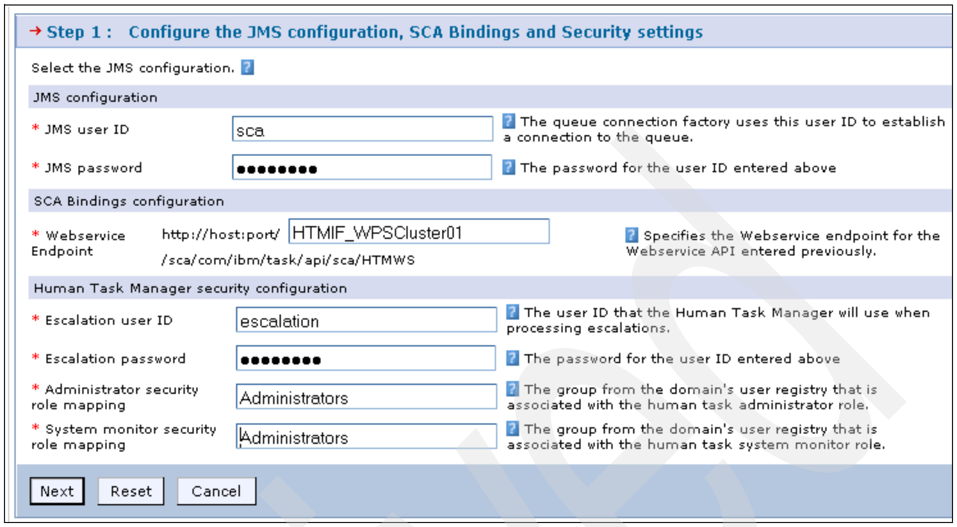
2. Set the configuration for JMS, SCA bindings, and security settings, as detailed in Table 8-10.

Table 8-10 Configuration settings

Name	Value
JMS user ID	sca
JMS password	itso4you
Webservice Endpoint	HTMIF_WPSCluster01(Default)
Escalation user ID	escalation
Escalation password	itso4you
Administrator security role mapping	Administrators
System monitor security role mapping	Administrators

Note: You do not enable global security in this scenario, so the `Administrators` group and the `sca` and `escalation` users are dummy values.

Figure 8-41 illustrates these settings.


→ Step 1 : Configure the JMS configuration, SCA Bindings and Security settings

Select the JMS configuration.

JMS configuration

* JMS user ID	sca	The queue connection factory uses this user ID to establish a connection to the queue.
* JMS password	●●●●●●●●●	The password for the user ID entered above

SCA Bindings configuration

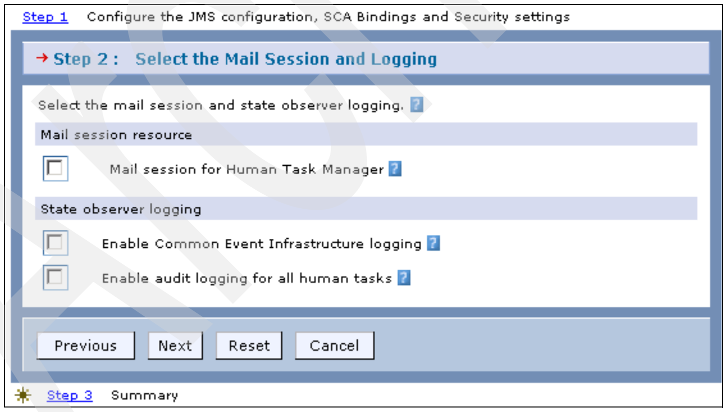
* Webservice Endpoint	http://host:port/ HTMIF_WPSCluster01 /sca/com/ibm/task/api/sca/HTMWS	Specifies the Webservice endpoint for the Webservice API entered previously.

Human Task Manager security configuration

* Escalation user ID	escalation	The user ID that the Human Task Manager will use when processing escalations.
* Escalation password	●●●●●●●●	The password for the user ID entered above
* Administrator security role mapping	Administrators	The group from the domain's user registry that is associated with the human task administrator role.
* System monitor security role mapping	Administrators	The group from the domain's user registry that is associated with the human task system monitor role.

Figure 8-41 Setting JMS, SCA, and security

3. Click **Next**.
4. Select mail session, Common Event Infrastructure logging ,and **audit logging for all human tasks**. You do not use the features for this scenario, so you can click **Next** here (Figure 8-42).


Step 1 Configure the JMS configuration, SCA Bindings and Security settings

→ Step 2 : Select the Mail Session and Logging

Select the mail session and state observer logging.

Mail session resource

☐ Mail session for Human Task Manager

State observer logging

☐ Enable Common Event Infrastructure logging

☐ Enable audit logging for all human tasks

Step 3 Summary

Figure 8-42 Setting mail session and logging

5. On the Summary window, click **Finish**.
6. Save and synchronize the configuration.

At this point, you have installed the BPC and HTM containers, as shown in Figure 8-43.

Note: So far, there is only one active messaging engine for the SCA bus and BPC bus in the cell.

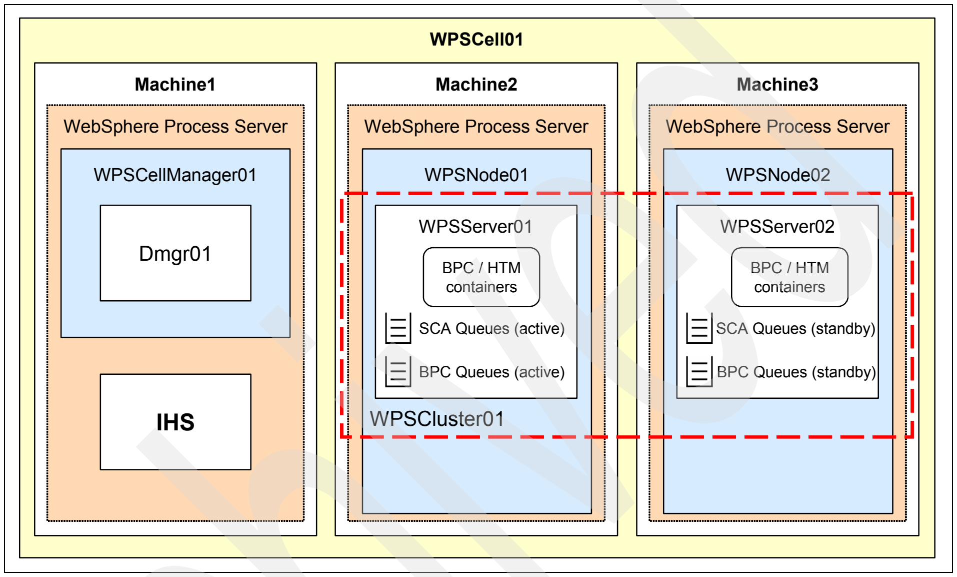


Figure 8-43 SCA and BPC configured with active/standby

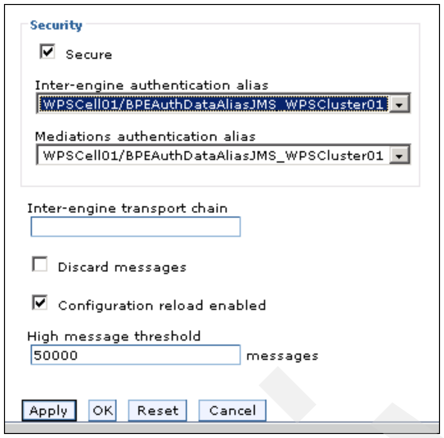
Reconfiguring the BPC Bus and messaging engine

To reconfigure the BPC Bus and messaging engine:

1. Open the Administrative Console on `wlt0` and log in.
2. Navigate to **Service integration** → **Buses**.
3. Click the `BPC.WPSCell01.Bus bus.`
4. Select WPSCell01/BPEAuthDataAliasJMS_WPSCluster01 from the **Inter-engine authentication alias** drop-down list.

5. Select **WPSCell01/BPEAuthDataAliasJMS_WPSCluster01** from the **Mediations authentication alias** drop-down list, as shown in Figure 8-44.

   The application that we use in this example does not require secure mediations; however, we enable the capability as a best-practice.


Security

☑ Secure

Inter-engine authentication alias: WPSCell01/BPEAuthDataAliasJMS_WPSCluster01

Mediations authentication alias: WPSCell01/BPEAuthDataAliasJMS_WPSCluster01

Inter-engine transport chain:

☐ Discard messages

☑ Configuration reload enabled

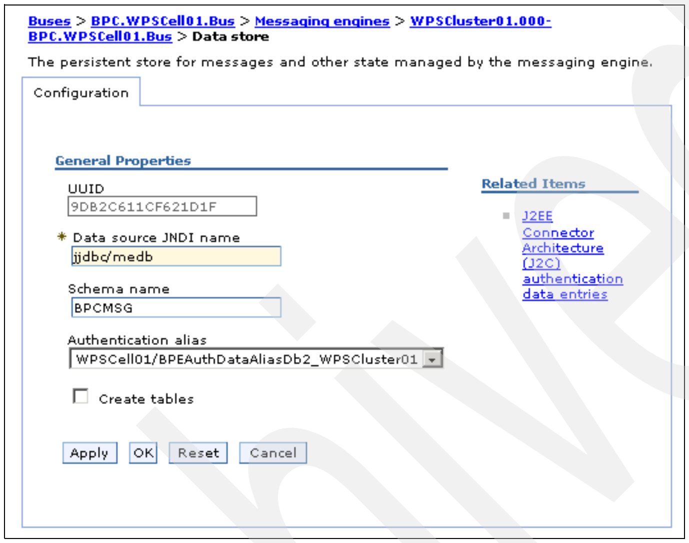
High message threshold: 50000 messages

Figure 8-44 Setting the BPC bus authentication aliases

6. Click **Apply**.
7. Under Topology, click **Messaging engines**.
8. Click the **WPSCluster01.000-BPC.WPSCell01.Bus** messaging engine.
9. Under Additional Properties, click **Data store**.

10. In the Configuration panel (Figure 8-45):

   a. Set the Data source JNDI name to `jdbc/medb`
   b. Set the **Schema name** to `BPCMSG`, as created earlier
   c. Select **WPSCell01/BPEAuthDataAliasDb2_WPSCluster01** from the **Authentication alias**
   d. Clear the **Create tables** check box
   e. Click **OK**.


Buses > BPC.WPSCell01.Bus > Messaging engines > WPSCluster01.000-BPC.WPSCell01.Bus > Data store

The persistent store for messages and other state managed by the messaging engine.

Configuration

General Properties

UUID: 9DB2C611CF621D1F

* Data source JNDI name: jjdbc/medb

Schema name: BPCMSG

Authentication alias: WPSCell01/BPEAuthDataAliasDb2_WPSCluster01

☐ Create tables

Related Items
J2EE Connector Architecture (J2C) authentication data entries
Apply
OK
Reset
Cancel

Figure 8-45   Correcting the BPC messaging engine schema

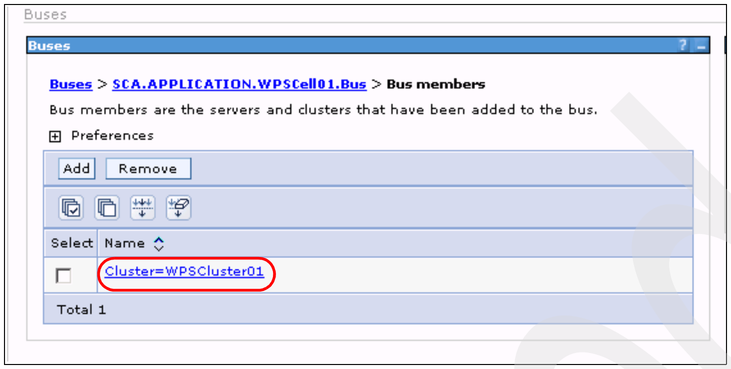
11. Save and synchronize the changes, and click **OK** when complete.

Defining additional messaging engines

To support multiple active SCA and BPC queues, you need to define additional messaging engines. Follow these steps:

1. In the Administrative Console of `wlt0`, navigate to **Service integration → Buses**.
2. Click **SCA.APPLICATION.WPSCell01.Bus**.
3. Click **Bus members** under **Topology**.

4. There is only one bus member displayed, `Cluster=WPSCluster01`. Click this bus member (Figure 8-46).



Buses

Buses

Buses > SCA.APPLICATION.WPSCell01.Bus > Bus members

Bus members are the servers and clusters that have been added to the bus.

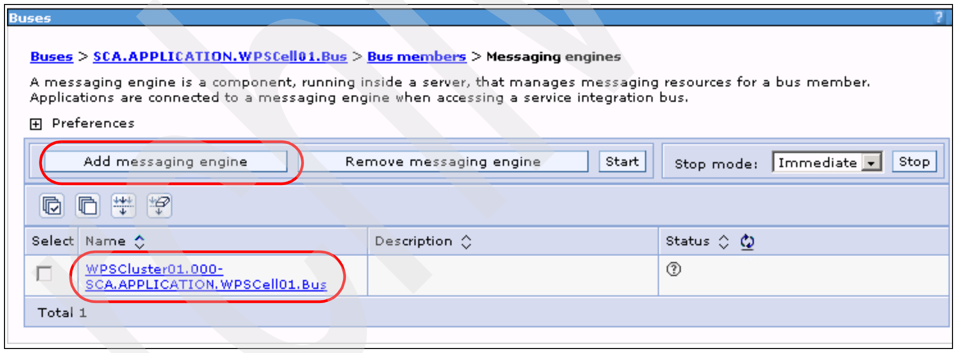
⊞ Preferences

Add | Remove

Select	Name
☐	Cluster=WPSCluster01
Total 1	

Figure 8-46 The cluster bus member

5. This bus member has a single messaging engine defined (`WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus`). Define a new one. Click **Add messaging engine** (Figure 8-47).



Buses

Buses > SCA.APPLICATION.WPSCell01.Bus > Bus members > Messaging engines

A messaging engine is a component, running inside a server, that manages messaging resources for a bus member. Applications are connected to a messaging engine when accessing a service integration bus.

⊞ Preferences

Add messaging engine | Remove messaging engine | Start | Stop mode: Immediate | Stop

Select	Name	Description	Status
☐	WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus		?
Total 1			

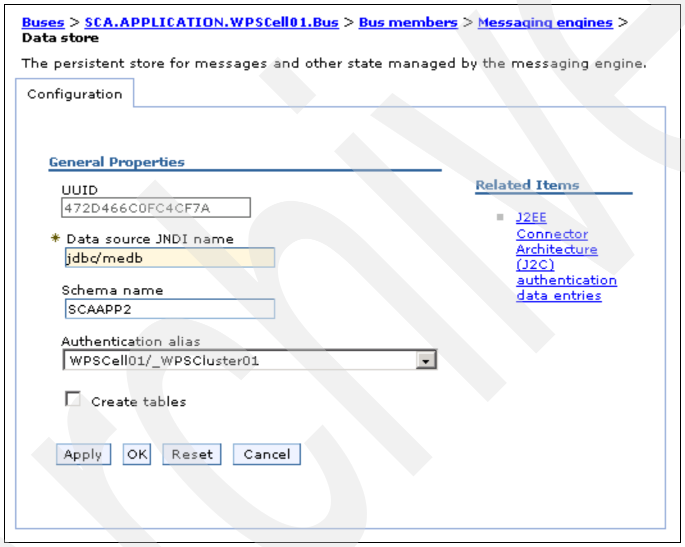
Figure 8-47 Messaging engine

6. Define the messaging engine configuration using the values that are specified in Table 8-11.

Table 8-11 Messaging engine configuration information for SCA application bus

Setting	Value
Data source JNDI name	jdbc/medb
Schema name	SCAAPP2
Authentication alias	WPSCell01/_WPSCluster01
Create tables	No

Figure 8-48 shows these settings.


Buses > SCA.APPLICATION.WPSCell01.Bus > Bus members > Messaging engines >
Data store

The persistent store for messages and other state managed by the messaging engine.

Configuration

General Properties

UUID
472D466C0FC4CF7A

* Data source JNDI name
jdbc/medb

Schema name
SCAAPP2

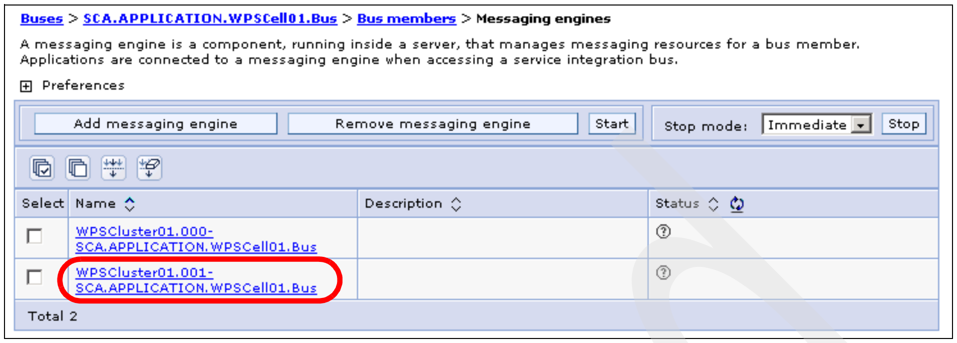
Authentication alias
WPSCell01/_WPSCluster01

☐ Create tables

Related Items
- J2EE Connector Architecture (J2C) authentication data entries

Figure 8-48 Define an additional messaging engine for the SCA application bus

7. Click **OK**. A second messaging engine is now defined (Figure 8-49).


Buses > SCA.APPLICATION.WPSCell01.Bus > Bus members > Messaging engines

A messaging engine is a component, running inside a server, that manages messaging resources for a bus member. Applications are connected to a messaging engine when accessing a service integration bus.

⊞ Preferences

Select	Name	Description	Status
☐	WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus		?
☐	WPSCluster01.001-SCA.APPLICATION.WPSCell01.Bus		?

Total 2

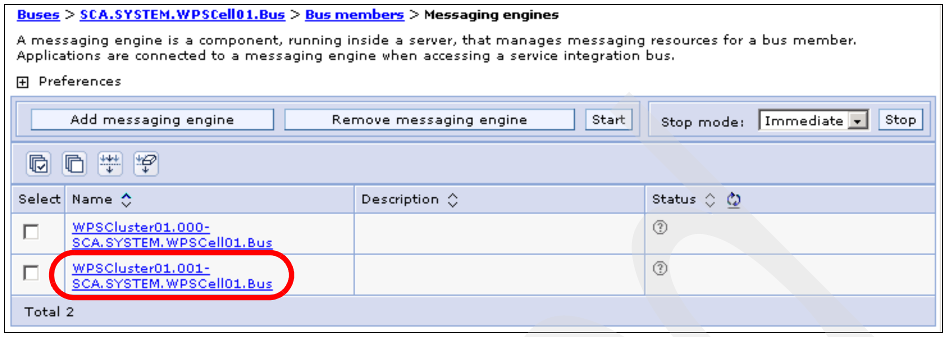
Figure 8-49 Additional messaging engine for the SCA application bus

8. Repeat these steps to define an additional messaging engine for the `SCA.SYSTEM.WPSCell01.Bus` bus.
   a. Click **Service integration** → **Buses**.
   b. Click **SCA.SYSTEM.WPSCell01.Bus** → **Bus members** → **Cluster=WPSCluster01**.
   c. Click **Add messaging engine** and define the messaging engine configuration using the values specified in Table 8-11.

Table 8-12 Messaging engine configuration information for SCA application bus

Setting	Value
Data source JNDI name	jdbc/medb
Schema name	SCASYS2
Authentication alias	WPSCell01/_WPSCluster01
Create tables	No

d. Click **OK**. You should see a second messaging engine defined (Figure 8-50).


Buses > SCA.SYSTEM.WPSCell01.Bus > Bus members > Messaging engines

A messaging engine is a component, running inside a server, that manages messaging resources for a bus member. Applications are connected to a messaging engine when accessing a service integration bus.

⊞ Preferences

Select	Name	Description	Status
☐	WPSCluster01.000-SCA.SYSTEM.WPSCell01.Bus		?
☐	WPSCluster01.001-SCA.SYSTEM.WPSCell01.Bus		?

Total 2

Figure 8-50 Additional messaging engine for the SCA system bus

9. Repeat these steps to define an additional messaging engine for the `SCA.SYSTEM.WPSCell01.Bus` bus.

   a. Click **Service integration** → **Buses**.

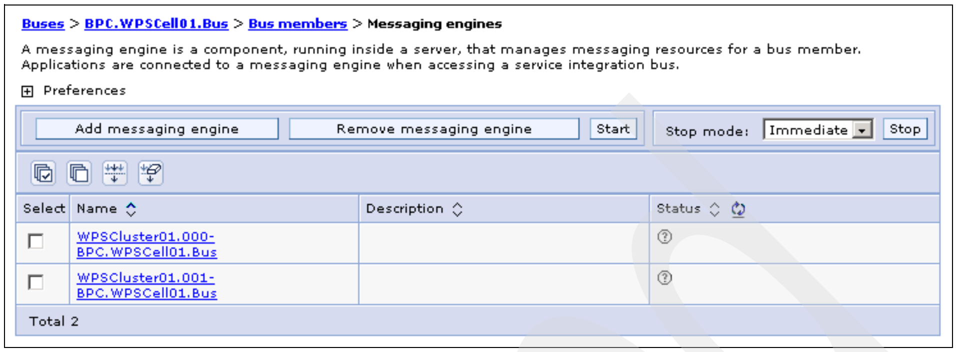
   b. Click **BPC.WPSCell01.Bus** → **Bus members** → **Cluster=WPSCluster01**.

   c. Click **Add messaging engine** and define the messaging engine configuration using the values that are specified in Table 8-11.

Table 8-13 Messaging engine configuration information for SCA application bus

Setting	Value
Data source JNDI name	jdbc/medb
Schema name	BPCMSG2
Authentication alias	WPSCell01/_WPSCluster01
Create tables	No

d. Click **OK**. You should see a second messaging engine defined (Figure 8-50).


Buses > BPC.WPSCell01.Bus > Bus members > Messaging engines

A messaging engine is a component, running inside a server, that manages messaging resources for a bus member. Applications are connected to a messaging engine when accessing a service integration bus.

Select	Name	Description	Status
☐	WPSCluster01.000-BPC.WPSCell01.Bus		?
☐	WPSCluster01.001-BPC.WPSCell01.Bus		?

Total 2

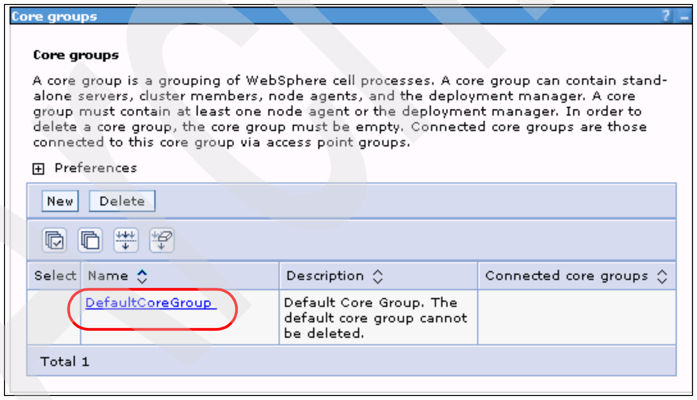
Figure 8-51 Additional messaging engine for the BPC bus

10. Save and synchronize the changes.

Creating HA Manager policies

To create HA Manager policies:

1. In the Administrative Console of `wlt0`, navigate to **Servers** → **Core groups** → **Core group settings**.
2. Click **DefaultCoreGroup** (Figure 8-52).


Core groups

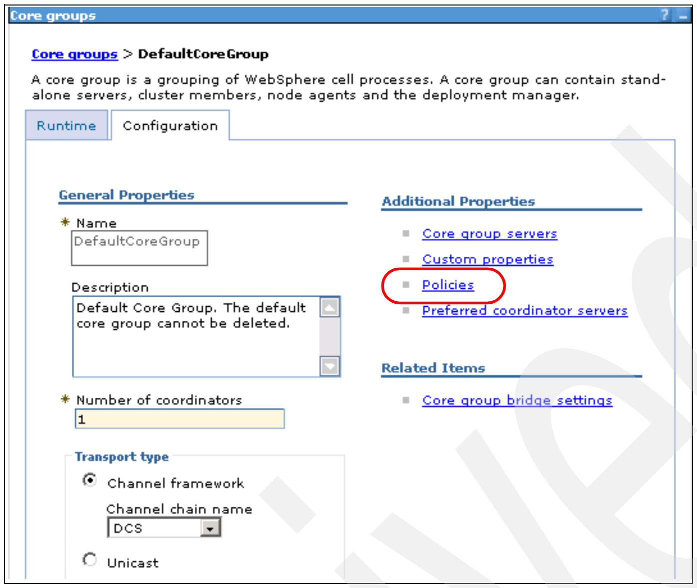
A core group is a grouping of WebSphere cell processes. A core group can contain stand-alone servers, cluster members, node agents, and the deployment manager. A core group must contain at least one node agent or the deployment manager. In order to delete a core group, the core group must be empty. Connected core groups are those connected to this core group via access point groups.

Select	Name	Description	Connected core groups
	DefaultCoreGroup	Default Core Group. The default core group cannot be deleted.	

Total 1

Figure 8-52 DefaultCoreGroup

3. Click **Policies** under **Additional Properties** (Figure 8-53).


Core groups

Core groups > DefaultCoreGroup

A core group is a grouping of WebSphere cell processes. A core group can contain stand-alone servers, cluster members, node agents and the deployment manager.

Runtime | Configuration

General Properties

* Name
DefaultCoreGroup

Description
Default Core Group. The default core group cannot be deleted.

* Number of coordinators
1

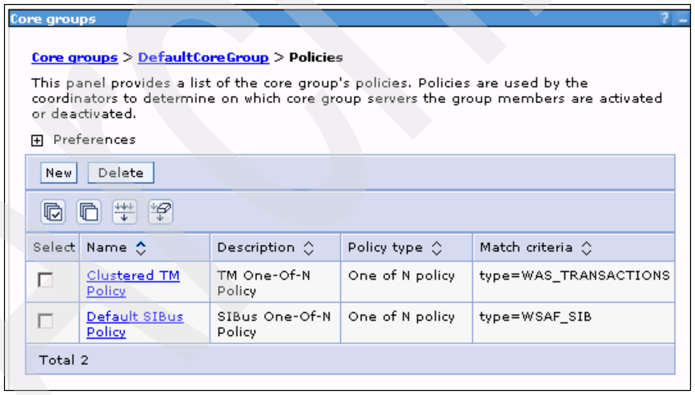
Transport type
(●) Channel framework
Channel chain name
DCS
( ) Unicast

Additional Properties
- Core group servers
- Custom properties
- Policies
- Preferred coordinator servers

Related Items
- Core group bridge settings

Figure 8-53   Policies link

4. A list of existing policies is displayed (Figure 8-54). Click **New** to define a new policy.


Core groups

Core groups > DefaultCoreGroup > Policies

This panel provides a list of the core group's policies. Policies are used by the coordinators to determine on which core group servers the group members are activated or deactivated.

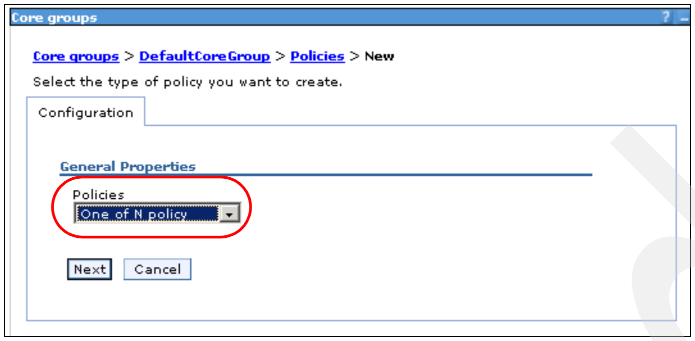
⊞ Preferences

New | Delete

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
Total 2				

Figure 8-54   Create a new policy

5. Select **One of N** policy, then click **Next** (Figure 8-55).


Core groups

Core groups > DefaultCoreGroup > Policies > New

Select the type of policy you want to create.

Configuration

General Properties

Policies: One of N policy

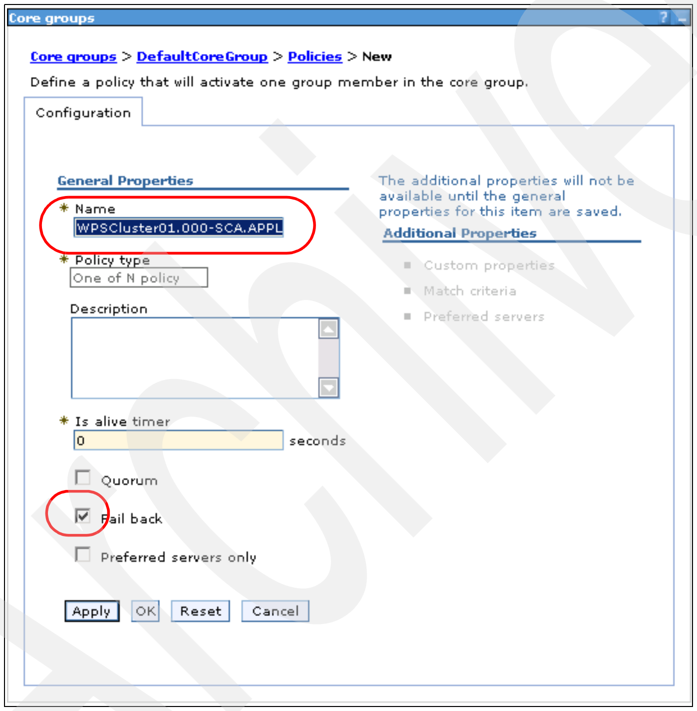
Figure 8-55 Specify the policy type as One of N

6. Specify the configuration information for the policy (Figure 8-56):

   a. Enter `WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus` in the Name field.

   > **Tip:** We set the policy name to match the name of the messaging engine to which this policy applies to help identify easily what each policy does.

   b. Select **Fail back** to ensure that the preferred server that is specified by this policy will always have high priority to run the messaging engine that is matched by the policy.


Core groups

Core groups > DefaultCoreGroup > Policies > New

Define a policy that will activate one group member in the core group.

Configuration

General Properties

* Name: WPSCluster01.000-SCA.APPL

* Policy type: One of N policy

Description:

* Is alive timer: 0 seconds

☐ Quorum

☑ Fail back

☐ Preferred servers only

The additional properties will not be available until the general properties for this item are saved.

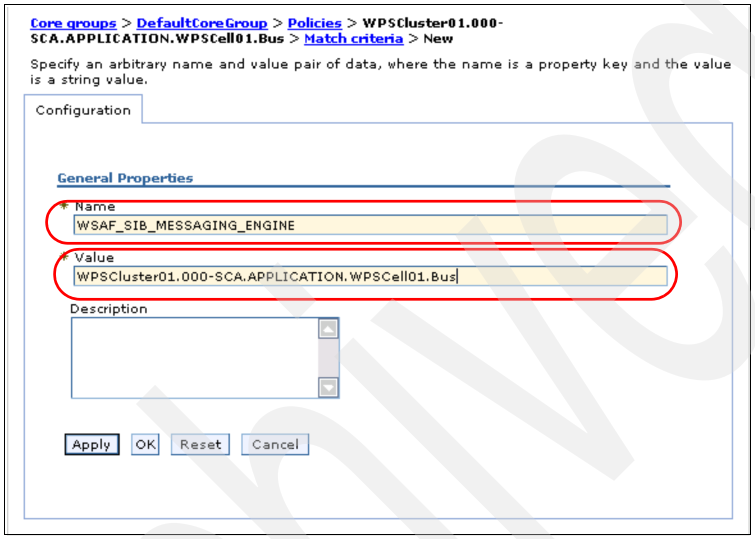
Additional Properties

- Custom properties
- Match criteria
- Preferred servers

Figure 8-56 Specify the policy configuration information

7. Click **Apply**.

8. Click **Match criteria** under Additional Properties. You add two matching criteria. Follow these steps:
   a. Click **New**.
   b. Set the Name to `WSAF_SIB_MESSAGING_ENGINE`.
   c. Set the Value to `WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus`.
   d. Click **OK** (Figure 8-57).


Core groups > DefaultCoreGroup > Policies > WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus > Match criteria > New

Specify an arbitrary name and value pair of data, where the name is a property key and the value is a string value.

Configuration

General Properties

* Name: WSAF_SIB_MESSAGING_ENGINE

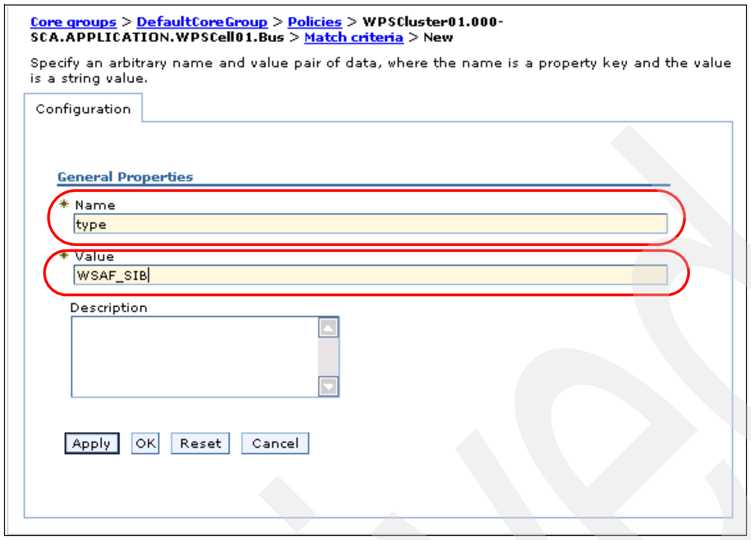
* Value: WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus

Description

Figure 8-57 Specify information for the first matching criteria

   e. Click **New** to create a second matching criteria.
   f. Set the Name to `type`.
   g. Set the Value to `WSAF_SIB`.

h. Click **OK** (Figure 8-58).


Core groups > DefaultCoreGroup > Policies > WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus > Match criteria > New

Specify an arbitrary name and value pair of data, where the name is a property key and the value is a string value.

Configuration

General Properties

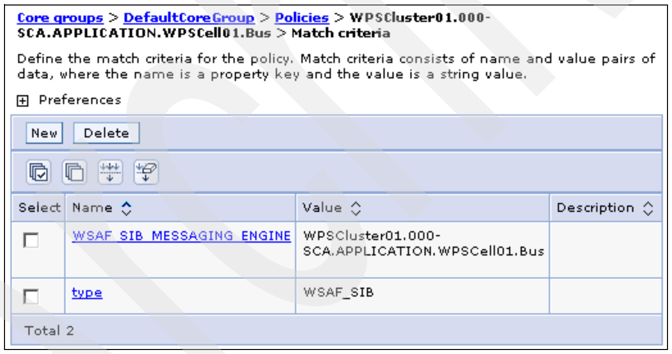
* Name: type

* Value: WSAF_SIB

Description

Figure 8-58 Specify information for the second matching criteria

9. You need to define two match criteria (Figure 8-59).


Core groups > DefaultCoreGroup > Policies > WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus > Match criteria

Define the match criteria for the policy. Match criteria consists of name and value pairs of data, where the name is a property key and the value is a string value.

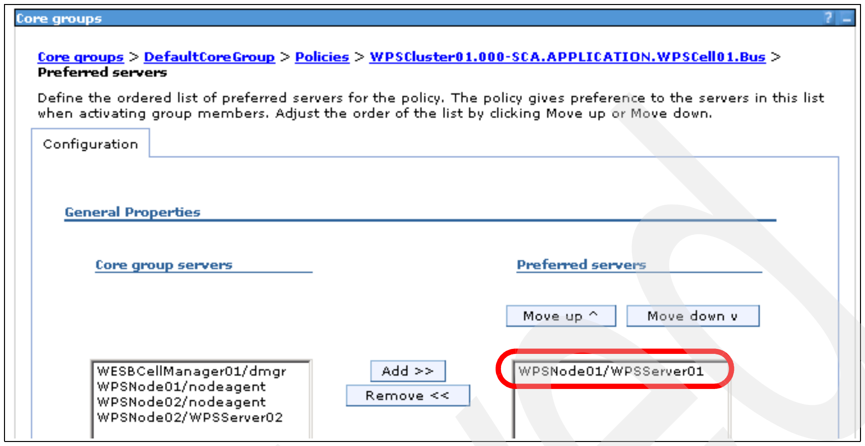
Select	Name	Value	Description
☐	WSAF_SIB_MESSAGING_ENGINE	WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus	
☐	type	WSAF_SIB	

Total 2

Figure 8-59 Match policy for the first SCA application messaging engine

10. Return back to the Policy panel, and **select WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus**.
11. Click **Preferred servers** under **Additional Properties** of the policy panel.

12. Highlight **WPSNode01/WPSServer01** and click **Add** to add it to the preferred servers list (Figure 8-60).


Core groups

Core groups > DefaultCoreGroup > Policies > WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus > Preferred servers

Define the ordered list of preferred servers for the policy. The policy gives preference to the servers in this list when activating group members. Adjust the order of the list by clicking Move up or Move down.

Configuration

General Properties

Core group servers: WESBCellManager01/dmgr, WPSNode01/nodeagent, WPSNode02/nodeagent, WPSNode02/WPSServer02

Preferred servers: WPSNode01/WPSServer01

Figure 8-60 WPSServer01 as the preferred server

13. Click **OK**. Save and synchronize the changes.

14. Create another policy for the second messaging engine on the `SCA.APPLICATION.WPSCell01.Bus` bus using the values specified in Table 8-14.

Table 8-14 Policy for second messaging engine on the SCA application bus

Setting	Value
Policy type	One of N policy
Policy name	WPSCluster01.001-SCA.APPLICATION.WPSCell01.Bus
Fail back	Yes
Match criteria 1 name	WSAF_SIB_MESSAGING_ENGINE
Match criteria 1 value	WPSCluster01.001-SCA.APPLICATION.WPSCell01.Bus
Match criteria 2 name	type
Match criteria 2 value	WSAF_SIB
Preferred server	WPSNode02/WPSServer02

15. Create a policy for the first messaging engine on the `SCA.APPLICATION.WPSCell01.Bus` bus using the values that are specified in Table 8-15.

Table 8-15 Policy for first messaging engine on the SCA system bus

Setting	Value
Policy type	One of N policy
Policy name	WPSCluster01.000-SCA.SYSTEM.WPSCell01.Bus
Fail back	Yes
Match criteria 1 name	WSAF_SIB_MESSAGING_ENGINE
Match criteria 1 value	WPSCluster01.000-SCA.SYSTEM.WPSCell01.Bus
Match criteria 2 name	type
Match criteria 2 value	WSAF_SIB
Preferred server	WPSNode02/WPSServer01

16. Create a policy for the second messaging engine on the `SCA.APPLICATION.WPSCell01.Bus` bus using the values that are specified in Table 8-15.

Table 8-16 Policy for second messaging engine on the SCA system bus

Setting	Value
Policy type	One of N policy
Policy name	WPSCluster01.001-SCA.SYSTEM.WPSCell01.Bus
Fail back	Yes
Match criteria 1 name	WSAF_SIB_MESSAGING_ENGINE
Match criteria 1 value	WPSCluster01.001-SCA.SYSTEM.WPSCell01.Bus
Match criteria 2 name	type
Match criteria 2 value	WSAF_SIB
Preferred server	WPSNode02/WPSServer02

17. Create a policy for the first messaging engine on the BPC.WPSCell01.Bus bus using the values that are specified in Table 8-15.

Table 8-17 Policy for first messaging engine on the BPC bus

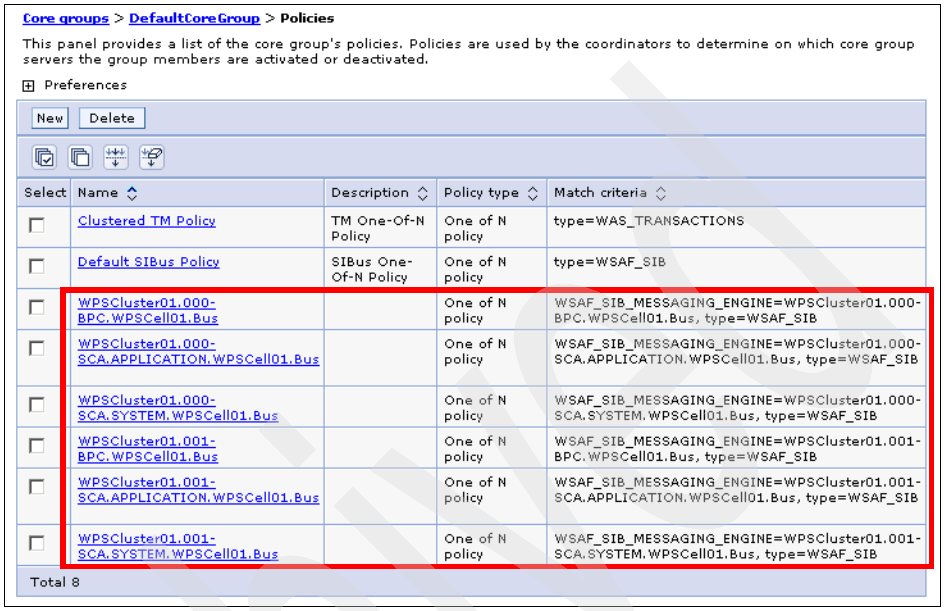
Setting	**Value**
Policy type	One of N policy
Policy name	WPSCluster01.000-BPC.WPSCell01.Bus
Fail back	Yes
Match criteria 1 name	WSAF_SIB_MESSAGING_ENGINE
Match criteria 1 value	WPSCluster01.000-BPC.WPSCell01.Bus
Match criteria 2 name	type
Match criteria 2 value	WSAF_SIB
Preferred server	WPSNode01/WPSServer01

18. Create a policy for the second messaging engine on the BPC.WPSCell01.Bus bus using the values that are specified in Table 8-15.

Table 8-18 Policy for second messaging engine on the BPC bus

Setting	**Value**
Policy type	One of N policy
Policy name	WPSCluster01.001-BPC.WPSCell01.Bus
Fail back	Yes
Match criteria 1 name	WSAF_SIB_MESSAGING_ENGINE
Match criteria 1 value	WPSCluster01.001-BPC.WPSCell01.Bus
Match criteria 2 name	type
Match criteria 2 value	WSAF_SIB
Preferred server	WPSNode02/WPSServer02

19. At this point, you have defined six HA Manager policies—four for SCA messaging engines and two for BPC messaging engines—as shown in Figure 8-61.


Core groups > DefaultCoreGroup > Policies

This panel provides a list of the core group's policies. Policies are used by the coordinators to determine on which core group servers the group members are activated or deactivated.

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
☐	WPSCluster01.000-BPC.WPSCell01.Bus		One of N policy	WSAF_SIB_MESSAGING_ENGINE=WPSCluster01.000-BPC.WPSCell01.Bus, type=WSAF_SIB
☐	WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus		One of N policy	WSAF_SIB_MESSAGING_ENGINE=WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus, type=WSAF_SIB
☐	WPSCluster01.000-SCA.SYSTEM.WPSCell01.Bus		One of N policy	WSAF_SIB_MESSAGING_ENGINE=WPSCluster01.000-SCA.SYSTEM.WPSCell01.Bus, type=WSAF_SIB
☐	WPSCluster01.001-BPC.WPSCell01.Bus		One of N policy	WSAF_SIB_MESSAGING_ENGINE=WPSCluster01.001-BPC.WPSCell01.Bus, type=WSAF_SIB
☐	WPSCluster01.001-SCA.APPLICATION.WPSCell01.Bus		One of N policy	WSAF_SIB_MESSAGING_ENGINE=WPSCluster01.001-SCA.APPLICATION.WPSCell01.Bus, type=WSAF_SIB
☐	WPSCluster01.001-SCA.SYSTEM.WPSCell01.Bus		One of N policy	WSAF_SIB_MESSAGING_ENGINE=WPSCluster01.001-SCA.SYSTEM.WPSCell01.Bus, type=WSAF_SIB

Total 8

Figure 8-61 Six HA Manager policies

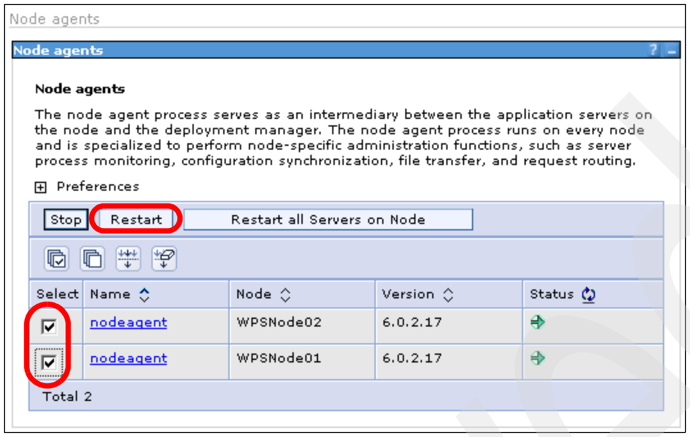
20. Save and synchronize the changes.

Restarting the environment

You need to restart all nodes and the deployment manager to ensure that the changes that you made are distributed cell wide. Follow these steps:

1. Open the Administrative Console for the deployment manager hosted on `wlt0`.
2. Navigate to **System administration** → **Node agents** and select both nodes by clicking the check boxes next to them.

3. Click on the **Restart** button to cause the nodes to stop and start, as shown in Figure 8-62.


Node agents

Node agents

Node agents

The node agent process serves as an intermediary between the application servers on the node and the deployment manager. The node agent process runs on every node and is specialized to perform node-specific administration functions, such as server process monitoring, configuration synchronization, file transfer, and request routing.

⊞ Preferences

Stop | Restart | Restart all Servers on Node

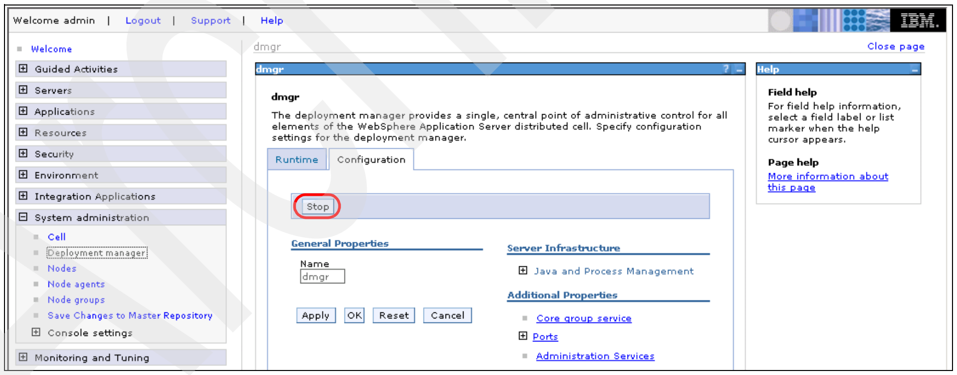
Select	Name	Node	Version	Status
✓	nodeagent	WPSNode02	6.0.2.17	➜
✓	nodeagent	WPSNode01	6.0.2.17	➜

Total 2

Figure 8-62 Restarting the node

Note: You have to refresh the Node agents view to see the change in status.

4. Navigate to **System administration** → **Deployment manager** and click **Stop**, as shown in Figure 8-63.


Welcome admin | Logout | Support | Help

Welcome
⊞ Guided Activities
⊞ Servers
⊞ Applications
⊞ Resources
⊞ Security
⊞ Environment
⊞ Integration Applications
⊟ System administration
- Cell
- Deployment manager
- Nodes
- Node agents
- Node groups
- Save Changes to Master Repository
⊞ Console settings
⊞ Monitoring and Tuning

dmgr | Close page

dmgr

dmgr

The deployment manager provides a single, central point of administrative control for all elements of the WebSphere Application Server distributed cell. Specify configuration settings for the deployment manager.

Runtime | Configuration

Stop

General Properties

Name: dmgr

Apply | OK | Reset | Cancel

Server Infrastructure
⊞ Java and Process Management

Additional Properties
- Core group service
⊞ Ports
- Administration Services

Help

Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

Figure 8-63 Stopping the deployment manager

5. Click **OK** in the verification panel.

6. Start the deployment manager. Open a command prompt on `wlt0`, change directory to %*WAS_HOME*%\profiles\Dmgr01\bin (for example, C:\IBM\WebSphere\ProcServer\profiles\Dmgr01\bin), and issue the following command:

   ```
   startManager.bat
   ```

At this point, the cluster is ready to host the business modules. Figure 8-64 shows the cell infrastructure.

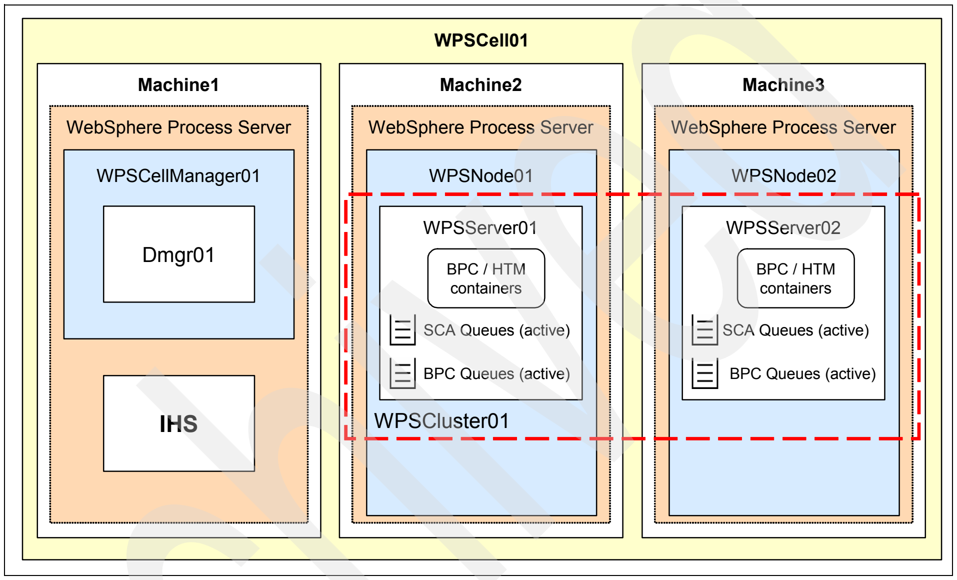


Figure 8-64 The SCA and BPC destinations are partitioned

8.3.3 Verifying the topology

This section describes how to verify the topology by starting the cluster, verifying the messaging engines, verifying the HA Manager policies, and verifying the SCA destinations.

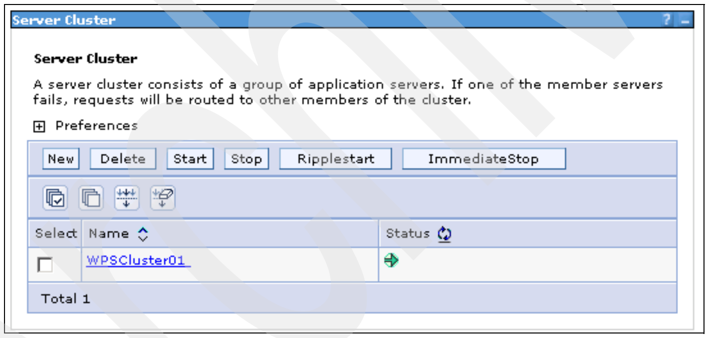
Starting the cluster

To start the cluster:

1. Ensure that all nodes are running. If a node is not running, start it with the following commands:
   a. Open a command prompt on the node's host machine.
   b. Change directory to the node's bin directory:

      cd\IBM\WebSphereProcServer\profiles\WPSNode01\bin
   c. Start the node using the following command:

      ```
      startNode.bat
      ```
2. Start the cluster:
   a. Open the `wlt0` Administrative Console and login.
   b. Navigate to **Servers** → **Clusters.**
   c. Select **WPSCluster01** and click **Start**.
   d. Verify that the cluster starts successfully. The red cross should change to a solid green arrow (Figure 8-65). You need to refresh the server clusters view to see the change in status.



Server Cluster

A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster.

⊞ Preferences

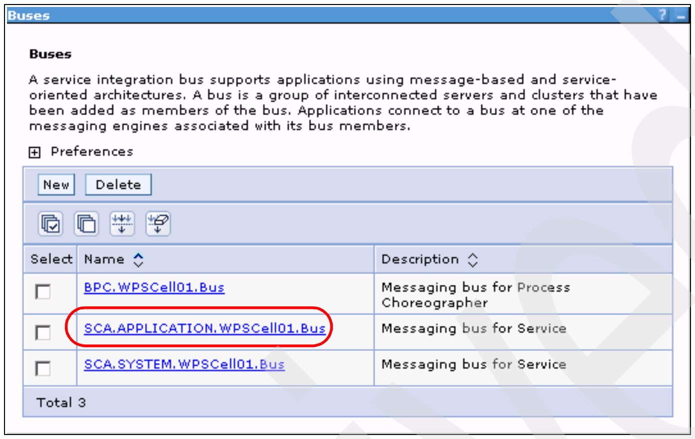
Select	Name	Status
☐	WPSCluster01	➜
Total 1		

Figure 8-65 Cluster in started state

Verifying the messaging engine

Confirm the messaging engine configuration by following these steps:

1. In Administrative Console of `wlt0`, click **Service integration → Buses.**
2. Select **SCA.APPLICATION.WPSCell01.Bus** (Figure 8-66).



Buses

A service integration bus supports applications using message-based and service-oriented architectures. A bus is a group of interconnected servers and clusters that have been added as members of the bus. Applications connect to a bus at one of the messaging engines associated with its bus members.

⊞ Preferences

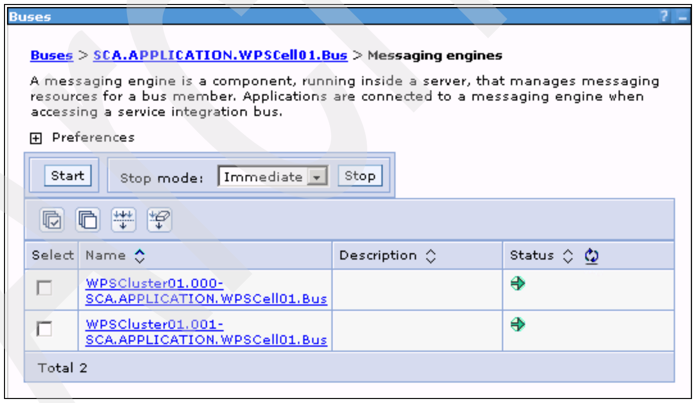
New Delete

Select	Name	Description
☐	BPC.WPSCell01.Bus	Messaging bus for Process Choreographer
☐	SCA.APPLICATION.WPSCell01.Bus	Messaging bus for Service
☐	SCA.SYSTEM.WPSCell01.Bus	Messaging bus for Service

Total 3

Figure 8-66 Select the bus SCA.APPLICATION.WPSCell01.Bus

3. Under **Topology**, click **Messaging engines**.
4. The messaging engines should be in a status of Started, signified by a green arrow (Figure 8-67).



Buses > SCA.APPLICATION.WPSCell01.Bus > Messaging engines

A messaging engine is a component, running inside a server, that manages messaging resources for a bus member. Applications are connected to a messaging engine when accessing a service integration bus.

⊞ Preferences

Start Stop mode: Immediate Stop

Select	Name	Description	Status
☐	WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus		➜
☐	WPSCluster01.001-SCA.APPLICATION.WPSCell01.Bus		➜

Total 2

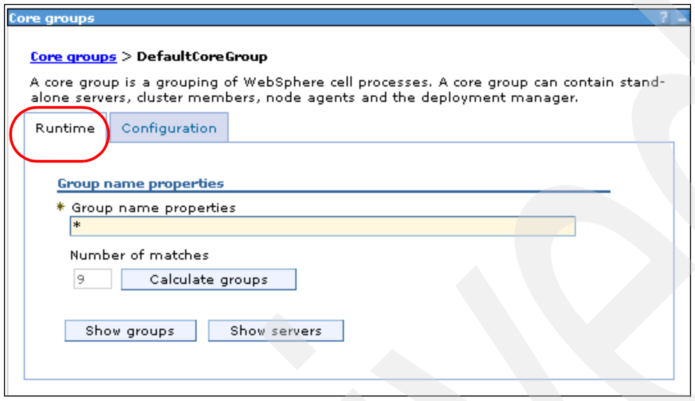
Figure 8-67 Messaging engines for SCA.APPLICATION bus

5. Repeat these steps to verify the status of the other messaging engines that are located in the other buses.

Verifying the HA Manager policy

To verify the HA Manager policy:

1. In Administrative Console of `wlt0`, click **Servers** → **Core groups** → **Core group settings** → **DefaultCoreGroup**.
2. Go to the Runtime tab (Figure 8-68).



A core group is a grouping of WebSphere cell processes. A core group can contain stand-alone servers, cluster members, node agents and the deployment manager.

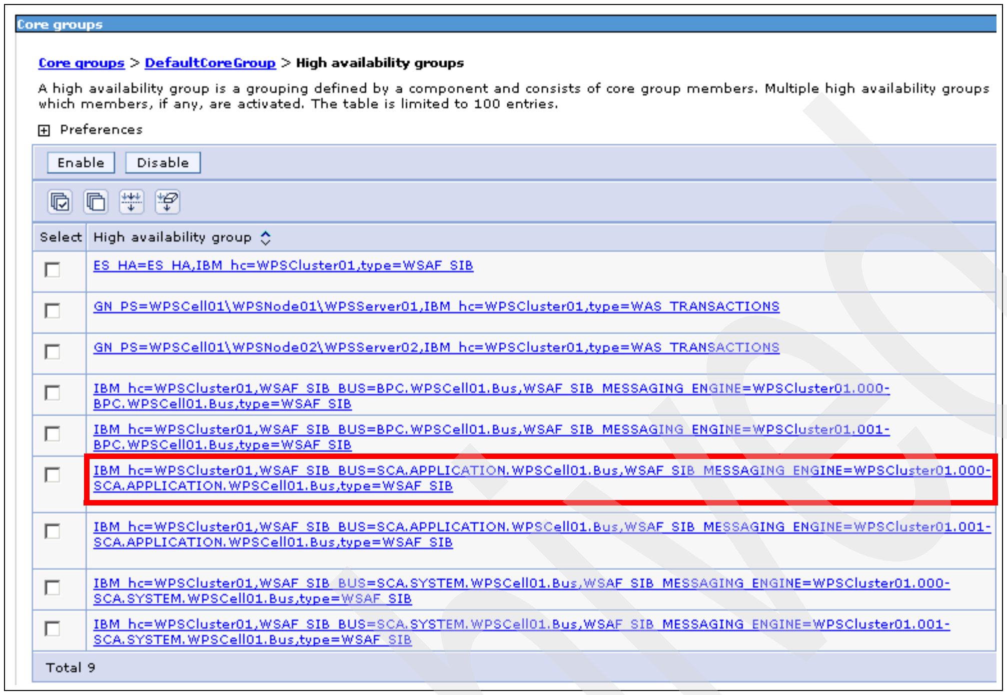
* Group name properties: *

Number of matches: 9

Figure 8-68 Runtime tab

3. Click **Show groups** to delay a list of high availability groups.

4. Click the high availability group for WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus (Figure 8-69).


Core groups

Core groups > DefaultCoreGroup > **High availability groups**

A high availability group is a grouping defined by a component and consists of core group members. Multiple high availability groups which members, if any, are activated. The table is limited to 100 entries.

⊞ Preferences

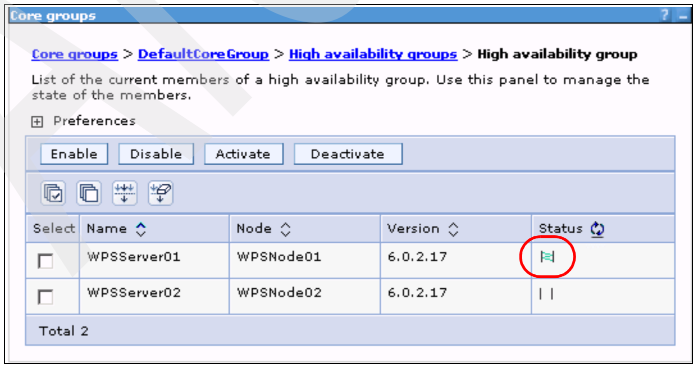
Enable | Disable

Select	High availability group
☐	ES_HA=ES_HA,IBM_hc=WPSCluster01,type=WSAF_SIB
☐	GN_PS=WPSCell01\WPSNode01\WPSServer01,IBM_hc=WPSCluster01,type=WAS_TRANSACTIONS
☐	GN_PS=WPSCell01\WPSNode02\WPSServer02,IBM_hc=WPSCluster01,type=WAS_TRANSACTIONS
☐	IBM_hc=WPSCluster01,WSAF_SIB_BUS=BPC.WPSCell01.Bus,WSAF_SIB_MESSAGING_ENGINE=WPSCluster01.000-BPC.WPSCell01.Bus,type=WSAF_SIB
☐	IBM_hc=WPSCluster01,WSAF_SIB_BUS=BPC.WPSCell01.Bus,WSAF_SIB_MESSAGING_ENGINE=WPSCluster01.001-BPC.WPSCell01.Bus,type=WSAF_SIB
☐	IBM_hc=WPSCluster01,WSAF_SIB_BUS=SCA.APPLICATION.WPSCell01.Bus,WSAF_SIB_MESSAGING_ENGINE=WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus,type=WSAF_SIB
☐	IBM_hc=WPSCluster01,WSAF_SIB_BUS=SCA.APPLICATION.WPSCell01.Bus,WSAF_SIB_MESSAGING_ENGINE=WPSCluster01.001-SCA.APPLICATION.WPSCell01.Bus,type=WSAF_SIB
☐	IBM_hc=WPSCluster01,WSAF_SIB_BUS=SCA.SYSTEM.WPSCell01.Bus,WSAF_SIB_MESSAGING_ENGINE=WPSCluster01.000-SCA.SYSTEM.WPSCell01.Bus,type=WSAF_SIB
☐	IBM_hc=WPSCluster01,WSAF_SIB_BUS=SCA.SYSTEM.WPSCell01.Bus,WSAF_SIB_MESSAGING_ENGINE=WPSCluster01.001-SCA.SYSTEM.WPSCell01.Bus,type=WSAF_SIB

Total 9

Figure 8-69 The core group policy list

5. For this policy, the preferred server is `WPSServer01` and the server is active, so the `WPSCluster01.000-SCA.APPLICATION.WPSCell01.Bus` messaging engine is associated with `WPSServer01`. If you shutdown the server `WPSServer01`, the messaging engine moves to `WPSServer02` automatically (Figure 8-70).


Core groups

Core groups > DefaultCoreGroup > High availability groups > **High availability group**

List of the current members of a high availability group. Use this panel to manage the state of the members.

⊞ Preferences

Enable | Disable | Activate | Deactivate

Select	Name	Node	Version	Status
☐	WPSServer01	WPSNode01	6.0.2.17	▶
☐	WPSServer02	WPSNode02	6.0.2.17	❙❙

Total 2

Figure 8-70 The preferred server is displayed

6. Verify the preferred server for other messaging engines according to the information listed in Table 8-19.

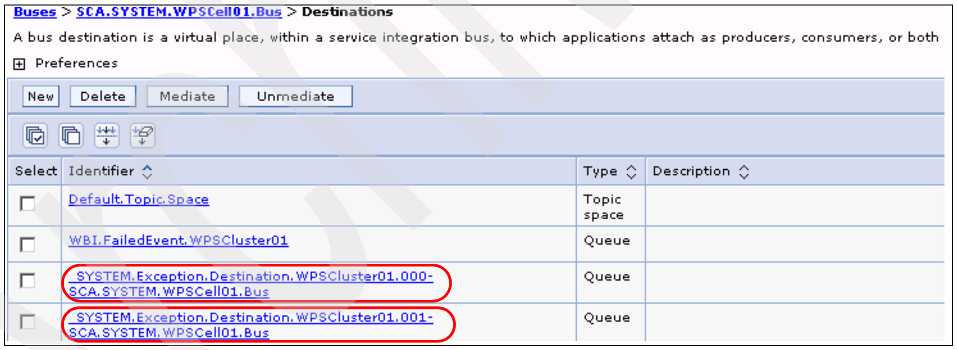
Table 8-19 The map between messaging engines and preferred servers

Messaging Engines	Preferred Server
WPSCluster01.001-SCA.APPLICATION.WPSCell01.Bus	WPSServer02
WPSCluster01.000-SCA.SYSTEM.WPSCell01.Bus	WPSServer01
WPSCluster01.001-SCA.SYSTEM.WPSCell01.Bus	WPSServer02
WPSCluster01.000-BPC.WPSCell01.Bus	WPSServer01
WPSCluster01.000-BPC.WPSCell01.Bus	WPSServer02

Verifying the destinations for SCA buses

To verify the destinations for SCA buses, follow these steps:

1. In Administrative Console of `wlt0`, click **Service integration** → **Buses** → **SCA.SYSTEM.WPSCell01.Bus** → **Destinations**.
2. Notice that the SYSTEM.Exception destination is partitioned across the two messaging engines of SCA system bus (Figure 8-71).



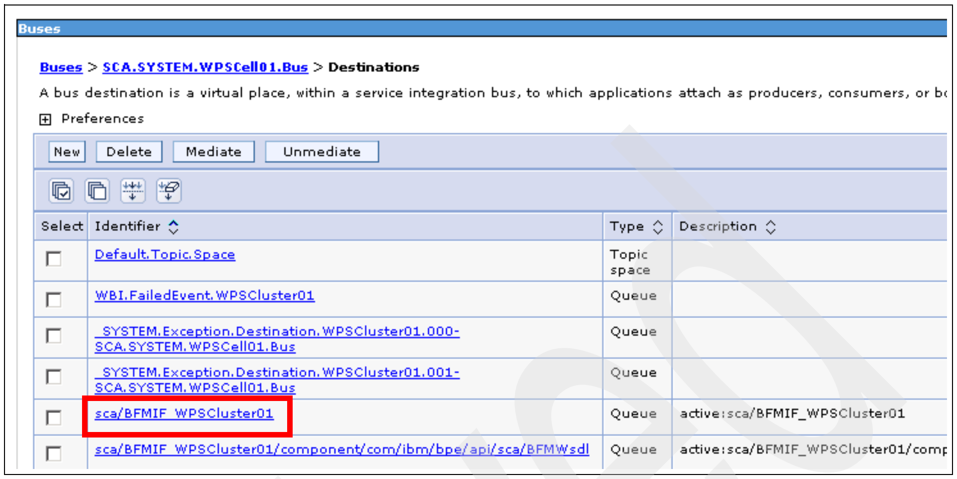
Buses > SCA.SYSTEM.WPSCell01.Bus > Destinations

A bus destination is a virtual place, within a service integration bus, to which applications attach as producers, consumers, or both

Select	Identifier	Type	Description
☐	Default.Topic.Space	Topic space	
☐	WBI.FailedEvent.WPSCluster01	Queue	
☐	SYSTEM.Exception.Destination.WPSCluster01.000-SCA.SYSTEM.WPSCell01.Bus	Queue	
☐	SYSTEM.Exception.Destination.WPSCluster01.001-SCA.SYSTEM.WPSCell01.Bus	Queue	

Figure 8-71 System exception destination is partitioned cross two messaging engines

3. Click the queue destination **sca/BFMIF_WPSCluster01** (Figure 8-72).



Buses

Buses > SCA.SYSTEM.WPSCell01.Bus > Destinations

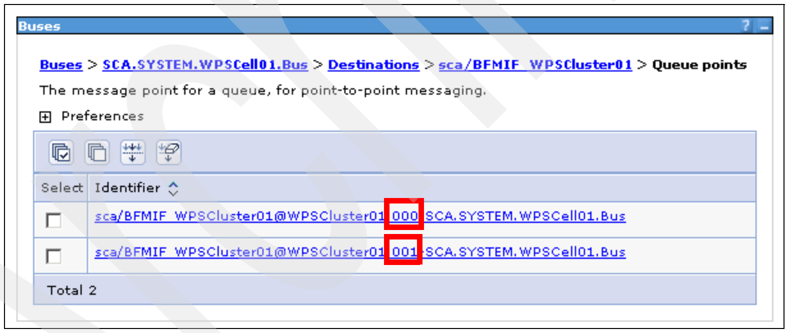
A bus destination is a virtual place, within a service integration bus, to which applications attach as producers, consumers, or bo

⊞ Preferences

Select	Identifier	Type	Description
☐	Default.Topic.Space	Topic space	
☐	WBI.FailedEvent.WPSCluster01	Queue	
☐	_SYSTEM.Exception.Destination.WPSCluster01.000-SCA.SYSTEM.WPSCell01.Bus	Queue	
☐	_SYSTEM.Exception.Destination.WPSCluster01.001-SCA.SYSTEM.WPSCell01.Bus	Queue	
☐	sca/BFMIF_WPSCluster01	Queue	active:sca/BFMIF_WPSCluster01
☐	sca/BFMIF_WPSCluster01/component/com/ibm/bpe/api/sca/BFMWsdl	Queue	active:sca/BFMIF_WPSCluster01/comp

Figure 8-72 Verify a queue destination

4. Click **Queue points** under **Message points**.
5. Two queue points are displayed—one point associates with one messaging engine (Figure 8-73).



Buses

Buses > SCA.SYSTEM.WPSCell01.Bus > Destinations > sca/BFMIF_WPSCluster01 > Queue points

The message point for a queue, for point-to-point messaging.

⊞ Preferences

Select	Identifier
☐	sca/BFMIF_WPSCluster01@WPSCluster01.000-SCA.SYSTEM.WPSCell01.Bus
☐	sca/BFMIF_WPSCluster01@WPSCluster01.001-SCA.SYSTEM.WPSCell01.Bus
Total 2	

Figure 8-73 Queue points configuration

You can also verify the queue points for other destinations in a similar way.

8.4 Installing and testing the scenario

This section describes how to test the topology that you have built by deploying presentation logic, business module, and enterprise service layers.

8.4.1 Installing the enterprise applications

To test the business process logic for this topology, you need to create a presentation logic layer and a remote enterprise service layer. The presentation logic layer provides the client for the business module, and the remote enterprise service layer presents services that are required for the business flows.

In this example the presentation logic layer is hosted on the business cluster alongside the business application.The remote enterprise service layer is hosted on a stand-alone WebSphere Application Server instance.

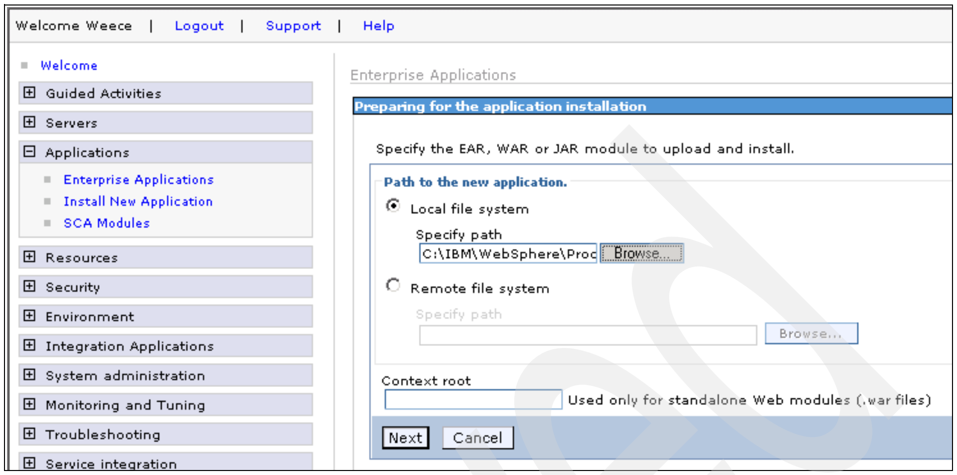
The presentation logic application ITSOWPSSmplPrsApp.ear is installed on `wlt1` and `wlt2` to `WPSCluster01` . The enterprise service application ITSOWPSSmplSrvApp.ear is installed on `wlt3` to `WASNode01 on server1`.

Deploying the business application

The application that supports the business process logic is provided by a single EAR file. This application must be installed to the cluster `WPSCluster01`. Follow these steps:

1. Open the `wlt0` Administrative Console and log in.
2. Navigate to **Applications** → **Install New Application**.
3. You will install the enterprise application ITSOWPSSmplBusApp.ear, which includes the business process logic. This file is located in the additional material that is supplied with this book. To learn how to obtain it, see Appendix A, “Additional material” on page 427. Copy ITSOWPSSmplBusApp.ear from the additional material that is supplied with this book to c:\IBM\WebSphere\ProcServer\installableApps.
4. In the Administrative Console, select **Local file system** and click **Browse**.
5. Navigate to the c:\IBM\WebSphere\ProcServer\installableApps folder.
6. Select the business module to be installed, in this example ITSOWPSCplxBusApp.ear, and click **Open**.

7. From the Administrative Console, click **Next**, as shown in Figure 8-74.


Welcome Weece | Logout | Support | Help

Enterprise Applications

Preparing for the application installation

Specify the EAR, WAR or JAR module to upload and install.

Path to the new application.

(●) Local file system

Specify path

C:\IBM\WebSphere\Proc

( ) Remote file system

Specify path

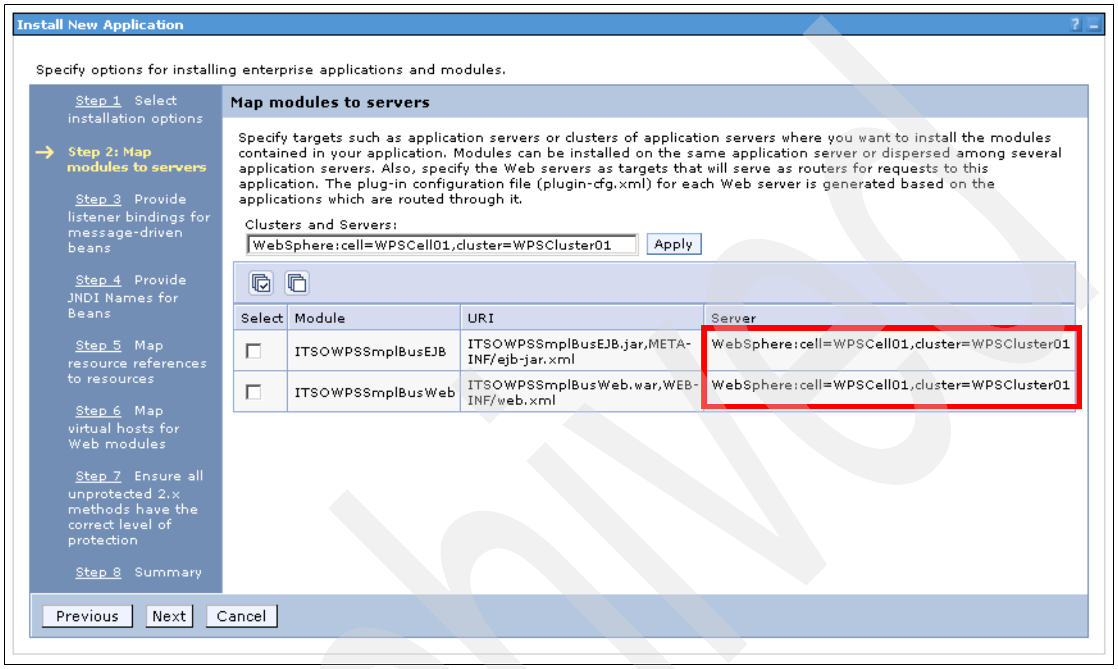
Context root

Used only for standalone Web modules (.war files)

Figure 8-74 Installing the business module

8. In the next panel, leave the settings as default and click **Next**.

   All the other application deployment settings are left as default, but notably the application must be deployed to the `WPSCluster01` cluster in step 2 of the wizard, 'Map modules to servers' as shown in Figure 8-75.


Install New Application
Step 1 Select installation options
Step 2: Map modules to servers
Step 3 Provide listener bindings for message-driven beans
Step 4 Provide JNDI Names for Beans
Step 5 Map resource references to resources
Step 6 Map virtual hosts for Web modules
Step 7 Ensure all unprotected 2.x methods have the correct level of protection
Step 8 Summary
Apply
Previous
Next
Cancel

Specify options for installing enterprise applications and modules.

Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install the modules contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that will serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated based on the applications which are routed through it.

Clusters and Servers: WebSphere:cell=WPSCell01,cluster=WPSCluster01

Select	Module	URI	Server
☐	ITSOWPSSmplBusEJB	ITSOWPSSmplBusEJB.jar,META-INF/ejb-jar.xml	WebSphere:cell=WPSCell01,cluster=WPSCluster01
☐	ITSOWPSSmplBusWeb	ITSOWPSSmplBusWeb.war,WEB-INF/web.xml	WebSphere:cell=WPSCell01,cluster=WPSCluster01

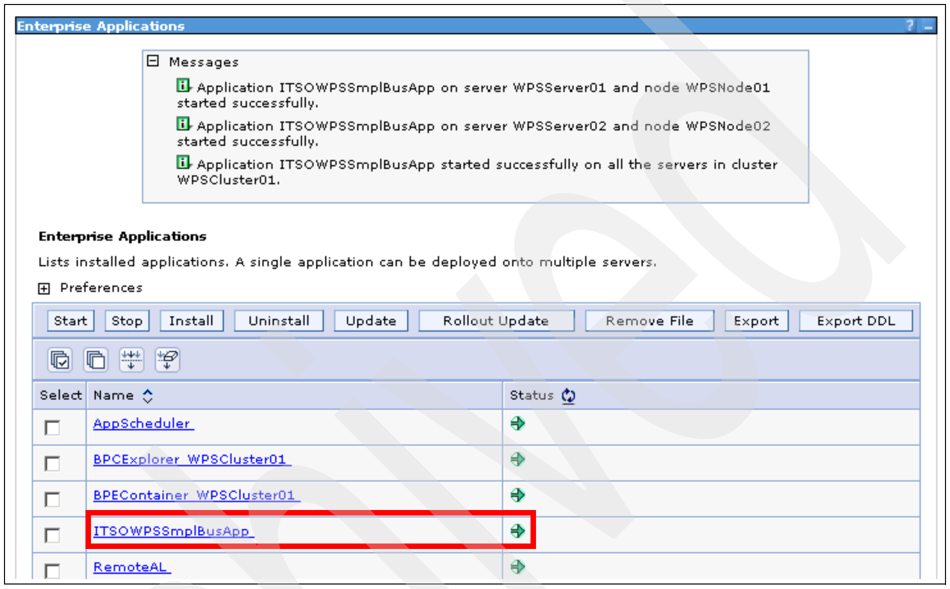
Figure 8-75 Verification of the correct application module mapping

9. Click the link to the summary panel, in this example **Step 8: Summary**.
10. Click **Finish**.
11. When the application has been installed successfully, click **Save to Master Configuration**.
12. Select **Synchronize changes with Nodes**, and click **Save**.
13. When completed successfully, click **OK**.

14. Start the application:

   a. Navigate to **Applications → Enterprise Applications**.

   b. Select **ITSOWPSSmplBusApp** and click **Start**. The status of the application should change to Started (represented by a green arrow), as shown in Figure 8-76.



Application ITSOWPSSmplBusApp on server WPSServer01 and node WPSNode01 started successfully.

Application ITSOWPSSmplBusApp on server WPSServer02 and node WPSNode02 started successfully.

Application ITSOWPSSmplBusApp started successfully on all the servers in cluster WPSCluster01.

Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

Select	Name	Status
☐	AppScheduler	➔
☐	BPCExplorer_WPSCluster01	➔
☐	BPEContainer_WPSCluster01	➔
☐	ITSOWPSSmplBusApp	➔
☐	RemoteAL	➔

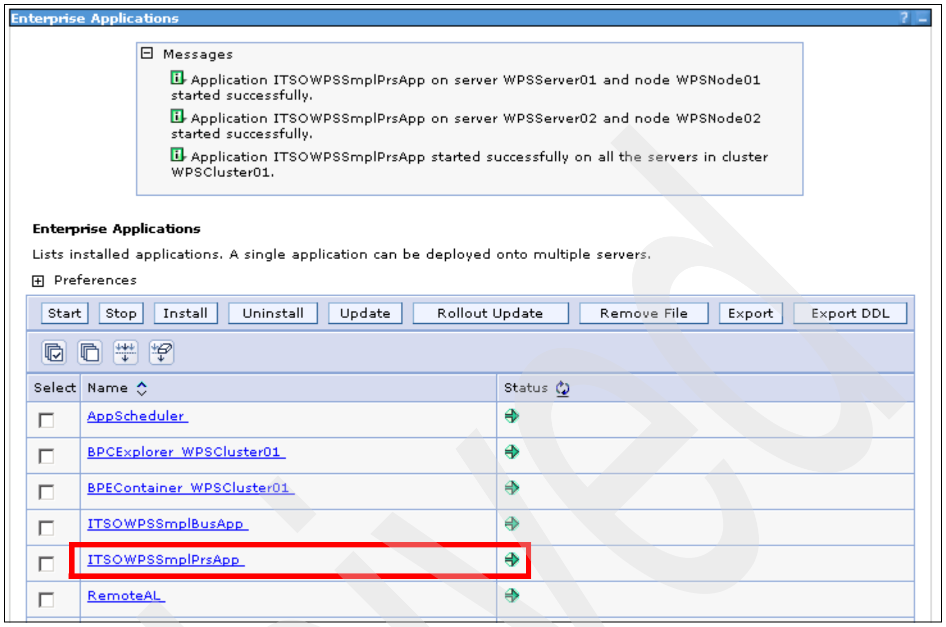
Figure 8-76 Installed enterprise application ITSOWPSSmplBusApp

Deploying the presentation logic application

To deploy the presentation logic application:

1. Repeat the steps that are described in "Deploying the business application" on page 304 to deploy and start the presentation logic application ITSOWPSSmplPrsApp.ear to the cluster `WPSCluster01`. You can also find this file can in the additional material that accompanies this book.

2. Verify that the application starts, as shown in Figure 8-77.


Enterprise Applications

Messages

Application ITSOWPSSmplPrsApp on server WPSServer01 and node WPSNode01 started successfully.

Application ITSOWPSSmplPrsApp on server WPSServer02 and node WPSNode02 started successfully.

Application ITSOWPSSmplPrsApp started successfully on all the servers in cluster WPSCluster01.

Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

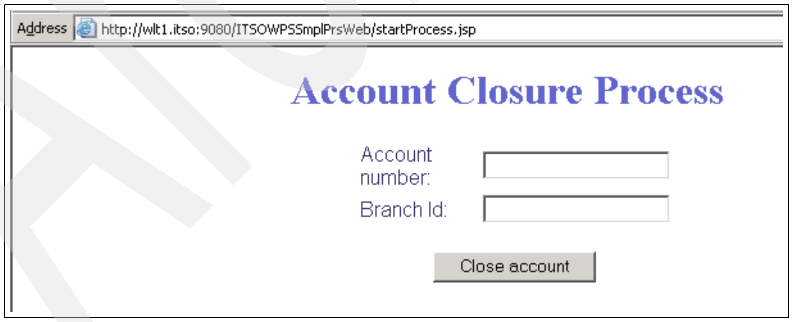
Preferences

Select	Name	Status
☐	AppScheduler	➡
☐	BPCExplorer_WPSCluster01	➡
☐	BPEContainer_WPSCluster01	➡
☐	ITSOWPSSmplBusApp	➡
☐	ITSOWPSSmplPrsApp	➡
☐	RemoteAL	➡

Figure 8-77 Installed enterprise application ITSOWPSSmplPrsApp

3. Verify that the presentation logic application is running on `WPSServer01` and `WPSServer02` within the cluster by entering the following URLs into a browser. You should see the Account Closure Process, as shown in Figure 8-78.

   – `http://wlt1.itso:9080/ITSOWPSSmplPrsWeb/startProcess.jsp`
   – `http://wlt2.itso:9080/ITSOWPSSmplPrsWeb/startProcess.jsp`


Address http://wlt1.itso:9080/ITSOWPSSmplPrsWeb/startProcess.jsp

Account Closure Process

Account number:

Branch Id:

Figure 8-78 Account Closure Process panel

Preparing the enterprise service layer

You need to define a WebSphere Application Server server to host the enterprise service layer, then install the enterprise service application. Follow these steps:

1. Create a new WebSphere Application Server profile using the profile creation wizard on `wlt3`. On Windows platforms, launch the Profile Wizard by clicking **Start** → **All Programs** → **IBM WebSphere** → **Application Server Network Deployment V6** → **Profile creation wizard**.
2. Define a profile with the following properties:
   - Profile type: `Application Server profile`
   - Profile name: `WASNode01`
   - Node name: `WASNode01`
   - Host name: `wlt3.itso`
3. When you have created the profile, navigate to the bin directory of the profile (which on our system was C:\IBM\WebSphere\ProcServer\profiles\WASNode01\bin) and start the server using the following command:

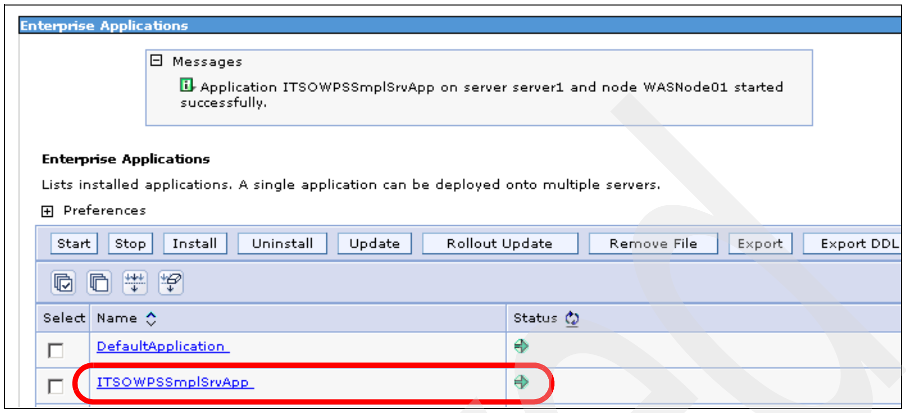
   ```
   startServer server1
   ```
4. When the server is started, launch the Administrative Console:

   ```
   http://wlt3.itso:9060/ibm/console
   ```
5. Install the enterprise service logic enterprise application ITSOWPSSmplSrvApp.ear. This file is located in the additional material that is supplied with this book. To learn how to obtain it, see Appendix A, “Additional material” on page 427. To install this file:

   a. Copy ITSOWPSSmplSrvApp.ear from the additional material that is supplied with this book to c:\IBM\WebSphere\ProcServer\installableApps.

   b. In the Administrative Console, click **Applications** → **Install New Application** and install ITSOWPSSmplSrvApp.ear. When the application is installed, save the configuration.

c. Click **Applications → Enterprise Applications and start ITSOWPSSmplSrvApp** (Figure 8-79).



Enterprise Applications

Messages

Application ITSOWPSSmplSrvApp on server server1 and node WASNode01 started successfully.

Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

Preferences

Select	Name	Status
☐	DefaultApplication	➔
☐	ITSOWPSSmplSrvApp	➔

Figure 8-79 Installed enterprise application ITSOWPSSmplSrvApp

8.4.2 Configuring IBM HTTP Server

You need to install and configure a router in front of the presentation layer running on the business process cluster `WPSCluster01`. This router enables load balancing to the presentation logic layer.

Client requests from the browser are made to the IBM HTTP Server. The IBM HTTP Server passes the request to the WebSphere plugin. The plugin load balances to the presentation applications running in the business cluster.

It is assumed that the following products have been installed:

- IBM HTTP Server in c:\IBMIHS
- The WebSphere plugin for IBM HTTP Server in c:\IBM\WebSphere\Plugins

Adding the Web server to the cell

To add the Web server to the cell:

1. Open a command prompt.
2. Change directory to c:\IBM\WebSphere\Plugins\bin.

Note: In this example, the Web server is named `webserver1`. Therefore, the configuration script is named configurewebserver1.bat.

3. Copy the configurewebserver1.bat file to %*WAS_HOME*%\bin on any node in the cell.

4. The Windows operating system has a limitation for the filename length. In order to avoid the exception caused by that limitation, you should do some additional configuration regarding the working directory of WebSphere Process Server.

   a. In Administrative Console, click **System Administration** → **Deployment Manager** → **Java and Process Management** → **Process Definition** → **Java Virtual Machine** → **Custom Properties**

   b. Click **New** and create a new property with the attributes:

      - Name: `websphere.workspace.root`
      - Value: `C:/w`

   c. Click **OK**.

   d. Save and synchronize the configuration.

   e. Restart the deployment manager.

5. Change directory to %*WAS_HOME*%\bin and run configurewebserver1.bat.

   Note: Ensure that the deployment manager has restarted before issuing this command.

This script adds the Web server into the cell and deploys the applications to the Web server.

Configuring the plug-in

To configure the plug-in:

1. Open the Administrative Console and login.
2. Navigate to **Servers** → **Web servers**.
3. Select the `webserver1` check box and click **Generate Plug-in**.
4. When complete, the message text identifies the location of the generated plug-in file. In this example, it is:

   c:\IBM\WebSphere\ProcServer\profiles\Dmgr01\config\cells\WPSCell01\nodes\wlt0.itso\servers\webserver1\plugin-cfg.xml

5. Move the plugin-cfg.xml configuration file.

The location of plugin-cfg.xml that the WebSphere Plug-in uses is specified by the `WebSpherePluginConfig` variable within the httpd.conf configuration file of IBM HTTP Server. Therefore, the generated plugin-cfg.xml must be moved to the path defined by this variable. In this example, the command is:

```
move
c:\IBM\WebSphere\ProcServer\profiles\Dmgr01\config\cells\WPSCell01\n
odes\wlt0.itso\servers\webserver1\plugin-cfg.xml
c:\IBM\WebSphere\Plugins\config\webserver1\plugin-cfg.xml
```

Configuring application deployment

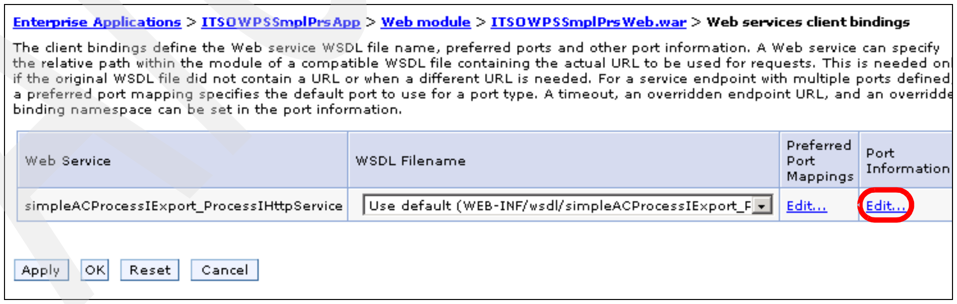
There is are additional steps to configure the end-to-end application deployment:

- Configuring the presentation layer for the business applications
- Configuring the business application for the remote enterprise services

Presentation layer connectivity to the business layer

To configure the presentation layer connectivity to the business layer, follow these steps:

1. Open the cell deployment manager's Administrative Console and login.
2. Navigate to **Applications** → **Enterprise Applications**.
3. Click **ITSOWPSSmplPrsApp**.
4. Under Related Items, click **Web Modules**.
5. Click the JAR file presented, **ITSOWPSSmplPrsWeb.war**.
6. Under Additional Properties, click **Web services client bindings**.
7. For the `simpleACProcessIExport_ProcessIHttpService` service, click **Edit** under **Port Information**, as shown in Figure 8-82.



Enterprise Applications > ITSOWPSSmplPrsApp > Web module > ITSOWPSSmplPrsWeb.war > **Web services client bindings**

The client bindings define the Web service WSDL file name, preferred ports and other port information. A Web service can specify the relative path within the module of a compatible WSDL file containing the actual URL to be used for requests. This is needed on if the original WSDL file did not contain a URL or when a different URL is needed. For a service endpoint with multiple ports defined a preferred port mapping specifies the default port to use for a port type. A timeout, an overridden endpoint URL, and an overridde binding namespace can be set in the port information.

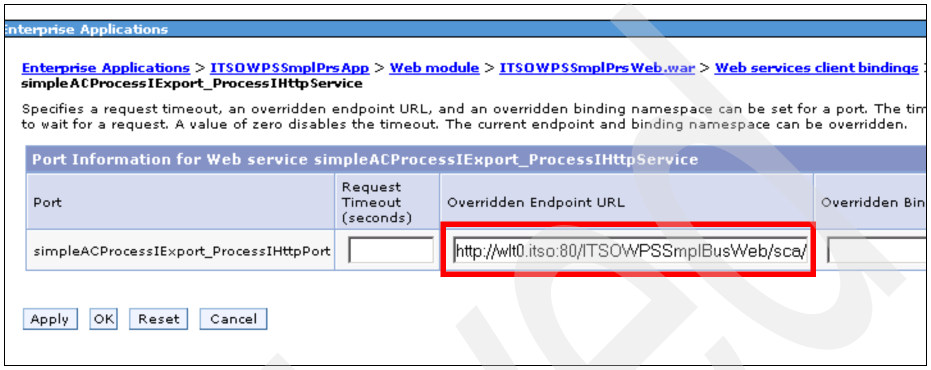
Web Service	WSDL Filename	Preferred Port Mappings	Port Information
simpleACProcessIExport_ProcessIHttpService	Use default (WEB-INF/wsdl/simpleACProcessIExport_F	Edit...	Edit...

Figure 8-80 Editing port information for the business process application

8. Change the Overridden Endpoint URL to that of the IBM HTTP Server, in this example:

```
http://wlt0.itso:80/ITSOWPSSmplBusWeb/sca/simpleACProcessIExport
```

This change is shown in Figure 8-83.



:nterprise Applications

Enterprise Applications > ITSOWPSSmplPrsApp > Web module > ITSOWPSSmplPrsWeb.war > Web services client bindings
simpleACProcessIExport_ProcessIHttpService

Specifies a request timeout, an overridden endpoint URL, and an overridden binding namespace can be set for a port. The tim
to wait for a request. A value of zero disables the timeout. The current endpoint and binding namespace can be overridden.

Port Information for Web service simpleACProcessIExport_ProcessIHttpService

Port	Request Timeout (seconds)	Overridden Endpoint URL	Overridden Bin
simpleACProcessIExport_ProcessIHttpPort		http://wlt0.itso:80/ITSOWPSSmplBusWeb/sca/	

Figure 8-81 Configuring the Endpoint URL

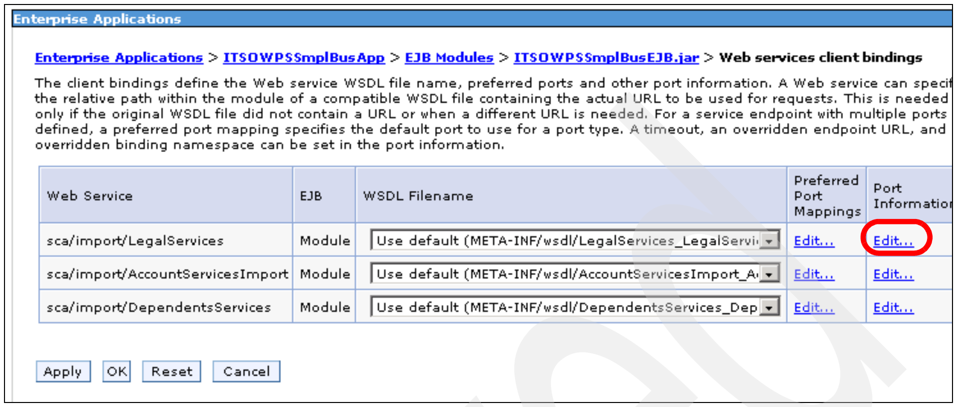
9. Click **OK**.

Business layer connectivity to remote enterprise services

To configure the business layer connectivity to remote enterprise services, follow these steps:

1. Open the cell deployment manager's Administrative Console and login.
2. Navigate to **Applications** → **Enterprise Applications**.
3. Click **ITSOWPSSmplBusApp**.
4. Under Related Items, click **EJB Modules**.
5. Click the JAR file presented, **ITSOWPSSmplBusEJB.jar**.
6. Under Additional Properties, click **Web services client bindings**.

7. For the `sca/import/LegalServices` service, click **Edit** under **Port Information,** as shown in Figure 8-82.



Enterprise Applications

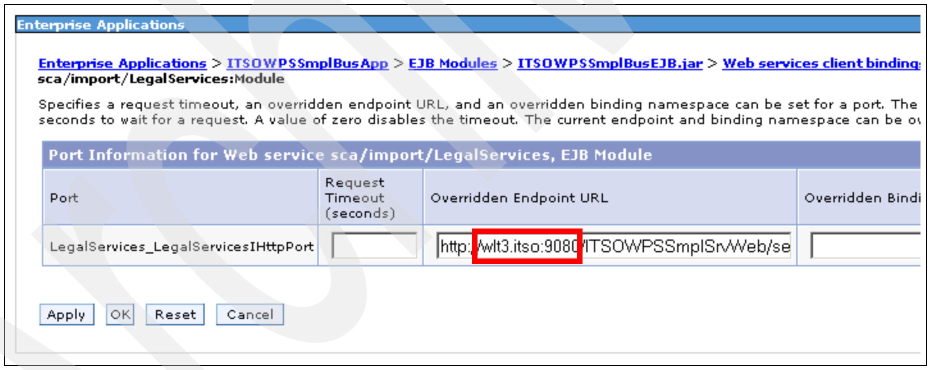
Enterprise Applications > ITSOWPSSmplBusApp > EJB Modules > ITSOWPSSmplBusEJB.jar > **Web services client bindings**

The client bindings define the Web service WSDL file name, preferred ports and other port information. A Web service can specif the relative path within the module of a compatible WSDL file containing the actual URL to be used for requests. This is needed only if the original WSDL file did not contain a URL or when a different URL is needed. For a service endpoint with multiple ports defined, a preferred port mapping specifies the default port to use for a port type. A timeout, an overridden endpoint URL, and overridden binding namespace can be set in the port information.

Web Service	EJB	WSDL Filename	Preferred Port Mappings	Port Informatio
sca/import/LegalServices	Module	Use default (META-INF/wsdl/LegalServices_LegalServi	Edit...	Edit...
sca/import/AccountServicesImport	Module	Use default (META-INF/wsdl/AccountServicesImport_A	Edit...	Edit...
sca/import/DependentsServices	Module	Use default (META-INF/wsdl/DependentsServices_Dep	Edit...	Edit...

Figure 8-82 Editing port information for the remote enterprise services

8. Change the host and port to that of the WebSphere Application Server that is hosting the remote enterprise services, in this example `http://wlt3.itso:9080/` (Figure 8-83).



Enterprise Applications

Enterprise Applications > ITSOWPSSmplBusApp > EJB Modules > ITSOWPSSmplBusEJB.jar > Web services client binding
sca/import/LegalServices:Module

Specifies a request timeout, an overridden endpoint URL, and an overridden binding namespace can be set for a port. The seconds to wait for a request. A value of zero disables the timeout. The current endpoint and binding namespace can be o

Port Information for Web service sca/import/LegalServices, EJB Module

Port	Request Timeout (seconds)	Overridden Endpoint URL	Overridden Bindi
LegalServices_LegalServicesIHttpPort		http://wlt3.itso:9080/ITSOWPSSmplSrvWeb/se	

Figure 8-83 Configuring the host for the remote enterprise services

9. Click **OK**.

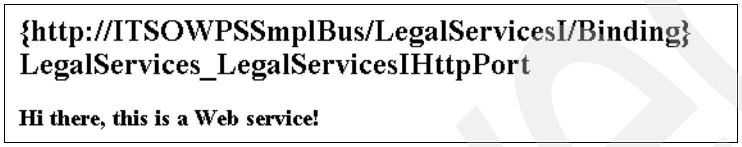
10. Repeat the previous step for the `sca/import/AccountServicesImport` and the `sca/import/DependentsServicesImport` services.

11. Save and synchronize the changes, and click **OK** when complete.

It is possible to test each of the remote enterprise services directly with a browser at the following URLs:

- `http://wlt3.itso:9080/ITSOWPSSmplSrvWeb/services/LegalServices_LegalServicesIHttpPort`
- `http://wlt3.itso:9080/ITSOWPSSmplSrvWeb/services/AccountServicesImport_AccountServicesIHttpPort`
- `http://wlt3.itso:9080/ITSOWPSSmplSrvWeb/services/DependentsServices_DependentsServicesIHttpPort`

Figure 8-84 shows an example response.

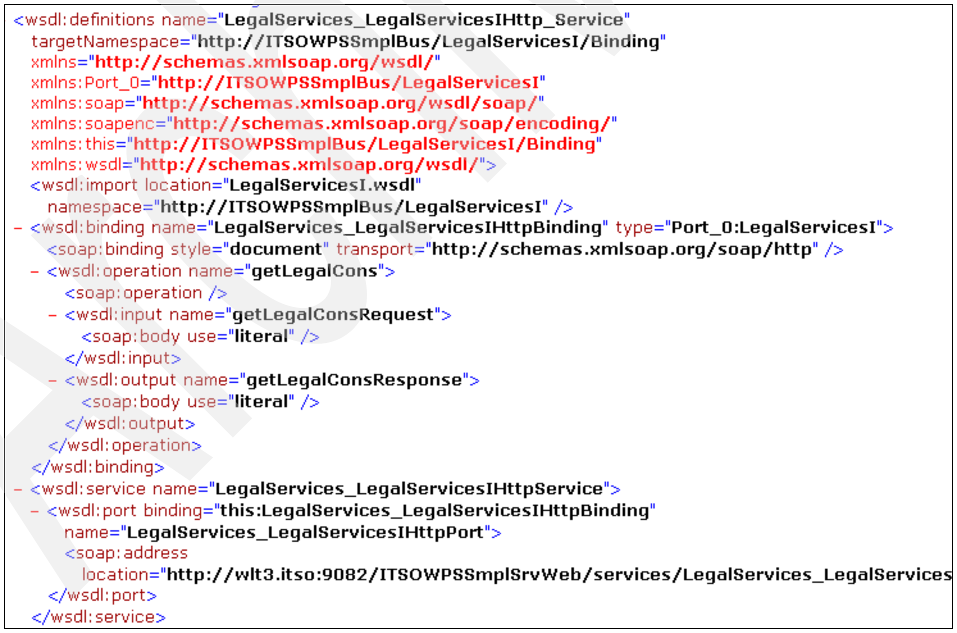


{http://ITSOWPSSmplBus/LegalServicesI/Binding}
LegalServices_LegalServicesIHttpPort

Hi there, this is a Web service!

Figure 8-84 Example response from Web service call

By appending `?wsdl` to these URLs, you can display the actual WSDL definition, as shown in Figure 8-85.



```
<wsdl:definitions name="LegalServices_LegalServicesIHttp_Service"
  targetNamespace="http://ITSOWPSSmplBus/LegalServicesI/Binding"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:Port_0="http://ITSOWPSSmplBus/LegalServicesI"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:this="http://ITSOWPSSmplBus/LegalServicesI/Binding"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:import location="LegalServicesI.wsdl"
    namespace="http://ITSOWPSSmplBus/LegalServicesI" />
- <wsdl:binding name="LegalServices_LegalServicesIHttpBinding" type="Port_0:LegalServicesI">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
  - <wsdl:operation name="getLegalCons">
      <soap:operation />
    - <wsdl:input name="getLegalConsRequest">
        <soap:body use="literal" />
      </wsdl:input>
    - <wsdl:output name="getLegalConsResponse">
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
- <wsdl:service name="LegalServices_LegalServicesIHttpService">
  - <wsdl:port binding="this:LegalServices_LegalServicesIHttpBinding"
      name="LegalServices_LegalServicesIHttpPort">
      <soap:address
        location="http://wlt3.itso:9082/ITSOWPSSmplSrvWeb/services/LegalServices_LegalServices
    </wsdl:port>
  </wsdl:service>
```

Figure 8-85 Requesting the WSDL definition

8.4.3 Testing the scenario

This section tests the environment and the application it hosts.

Starting the environment

To start the environment, follow these steps:

1. Ensure that all layers of the environment are started and that all applications running in these layers are started.
2. Start up IBM HTTP Server with the following commands:

```
cd c:\IBMIHS\bin
Apache.exe -k start
```

Testing the business process

To test the business process:

1. Launch the presentation layer by browsing to the following URL to launch the presentation layer JSP (Figure 8-86):

```
http://wlt0.itso/ITSOWPSSmplPrsWeb/startProcess.jsp
```

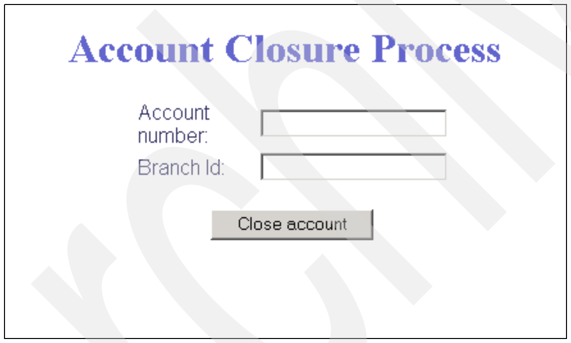
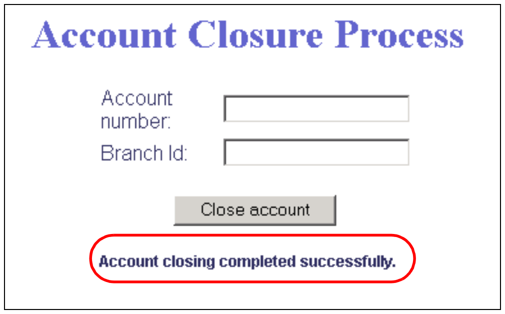


Figure 8-86 Account Closure Process JSP

2. Enter an Account number, for example `888888`.
3. Enter a Branch ID, for example `12345`.
4. Click **Close account**.

5. If the topology is working, you receive the message `Account closing completed successfully` as shown in Figure 8-87.



Account closing completed successfully.

Figure 8-87 The account is closed successfully

6. Try submitting a few more requests.
7. Open the SystemOut.log file of `WPSServer01`. You can find the log files in the *WPS_HOME*\profiles\WPSNode01\logs\WPSServer01 directory (for example in our environment, the `WPSServer01` log is stored in the c:\IBM\WebSphere\ProcServer\profiles\WPSNode01\logs\WPSServer01\ directory).

   You find that the router has dispatched the requests to different cluster members (Example 8-3 and Example 8-4).

Example 8-3 The messages from WPSServer01

```
00000066 SystemOut     O ##### Account closure service called in
asynchronous way for account 666666.#####
00000066 SystemOut     O ##### Account closure service called in
asynchronous way for account 0345678.#####
```

Example 8-4 The messages from WPSServer02

```
0000006a SystemOut     O ##### Account closure service called in
asynchronous way for account 888888.#####
0000006a SystemOut     O ##### Account closure service called in
asynchronous way for account 999999.#####
```

Note: Try sending requests from multiple clients to fully test the load balancing.

9

Implementing the Full Support topology for WebSphere Process Server

This chapter details the steps to implement the *Full Support topology* for WebSphere Process Server and to deploy a sample business process to this topology. We discuss the following topics:

- Selecting and implementing the Full Support topology
- Generic steps for creating WebSphere Process Server clusters
- Specific configuration steps for the Full Support topology
- Installing and testing the scenario

This chapter also describes how to secure this topology using global security with an LDAP registry, and SSL.

9.1 Selecting and implementing the Full Support topology

This section includes the following sections:

- Scenario requirements

  Summarizes the fictitious business scenario that we deploy to the Full Support topology we build in this chapter.
- Selecting the appropriate topology

  Demonstrates why we select the Full Support topology for WebSphere Process Server to host this business scenario.
- Overview of how to implement the Full Support topology

  Describes the steps that are necessary to build the Full Support topology for WebSphere Process Server.

Note: For an overview of the Full Support topology, refer to 4.4, "Full Support topology" on page 53.

9.1.1 Scenario requirements

You must build a production topology to host a WebSphere Process Server business process that is built for the fictitious organization ITSO Bank. The business process (shown in Figure 9-1) performs the following functions:

1. A bank employee submits an account closure order to the business process.
2. The *Constraints check* service is invoked. The connection between the process and the service is a Web service binding invoked synchronously.
3. The next service is invoked in the same way, with a synchronous Web service binding.
4. A service is invoked to calculate the account balance. A human task provides the bank employee with the account balance information. Based on this value (for example the balance exceeds U. S. $10 000), the employee decides whether to proceed with account closure or to take another action. In the second case, the process stops without invocation of the closure service, and the bank employee takes some account retention action.
5. If the account is to be closed, the process invokes the *Closure* service with an asynchronous one-way Web service binding.
6. The process terminates successfully and confirmation is returned.

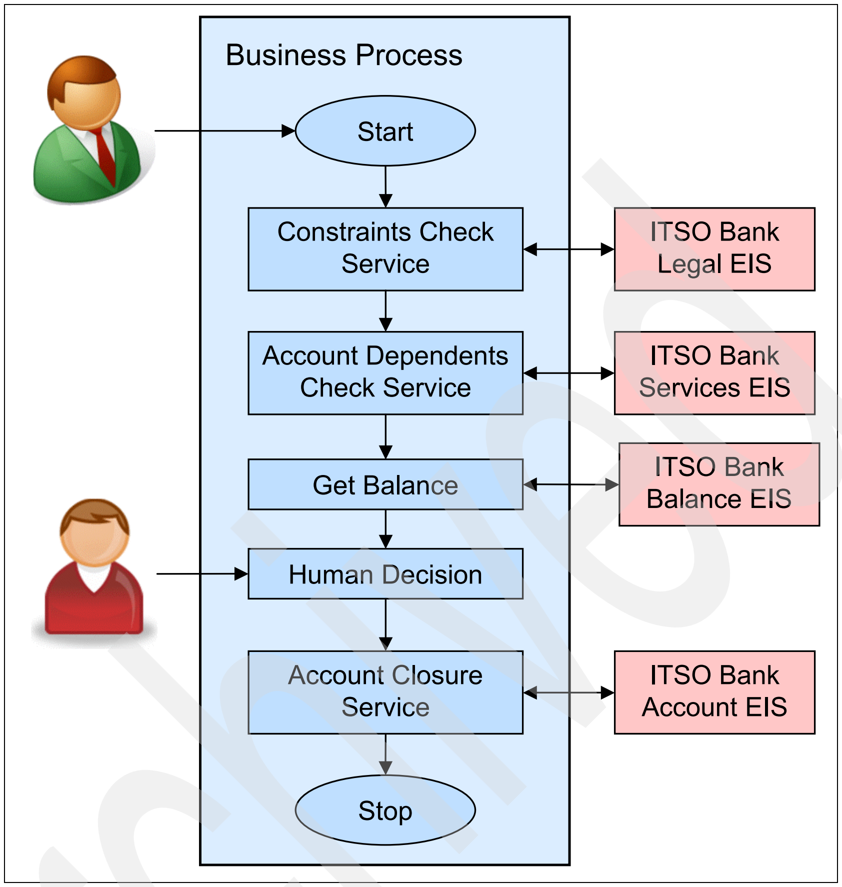


Figure 9-1 Account closure process, complex business process scenario

ITSO Bank has the following business and IT requirements for this scenario:

- Failover capability.
- ITSO Bank has acquired new hardware resources and wants to use them to build this topology
- To implement the process, the bank has decided to use existing services within the enterprise.
- The ITSO Bank Account EIS service is a one-way call and is invoked asynchronously.
- Event management services are not required.

Note: For more detailed information about this scenario, refer to 5.2.3, “Complex business process scenario” on page 79.

9.1.2 Selecting the appropriate topology

Based on the ITSO Bank requirements for the business process scenario, we can select an appropriate production topology to host the business process. To help us select the appropriate topology, we use the topology selection flowchart, as described in 4.5.2, “Topology selection flow chart” on page 62. Figure 9-2 shows that by using the topology selection flowchart, we can select the Full Support topology as the topology of choice for the ITSO Bank complex business process scenario.

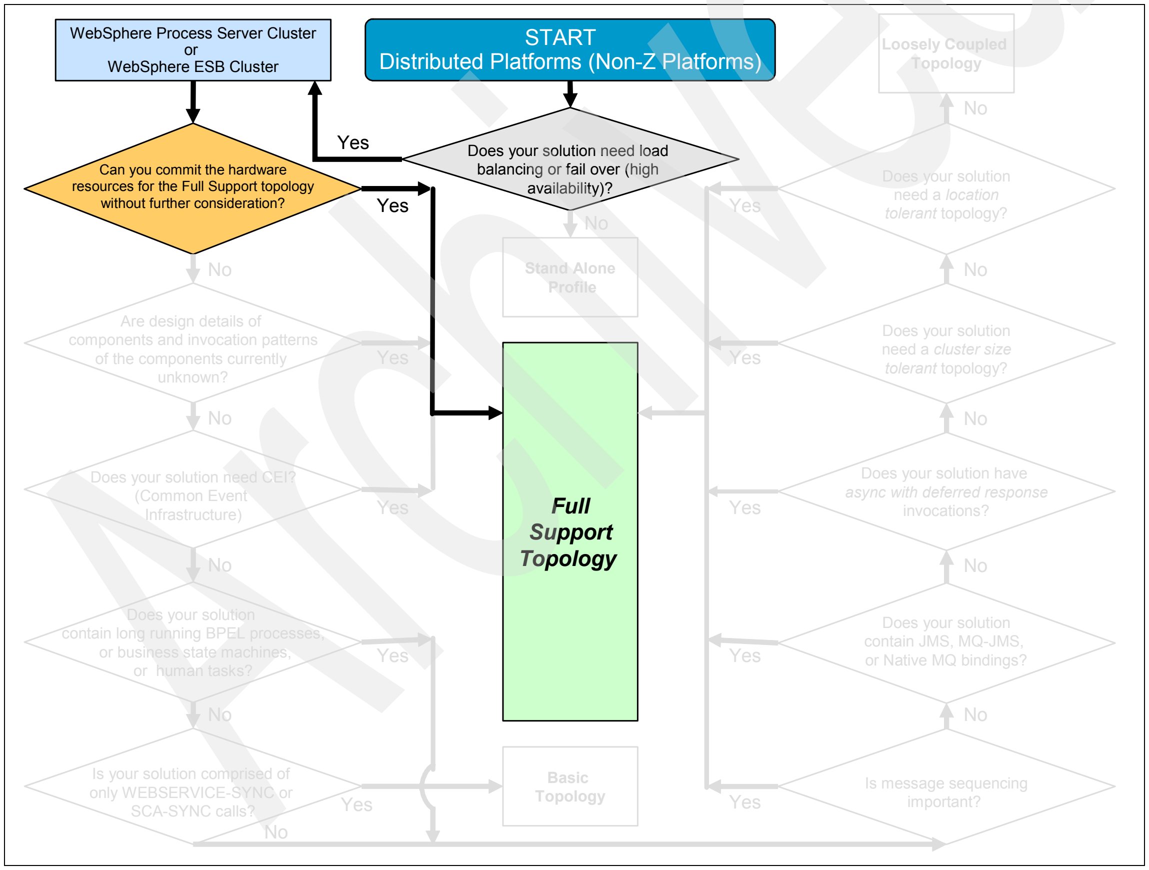

WebSphere Process Server Cluster or WebSphere ESB Cluster
START Distributed Platforms (Non-Z Platforms)
Loosely Coupled Topology
Yes
Does your solution need load balancing or fail over (high availability)?
No
Can you commit the hardware resources for the Full Support topology without further consideration?
Yes
No
Stand Alone Profile
Does your solution need a location tolerant topology?
Yes
No
Are design details of components and invocation patterns of the components currently unknown?
Yes
Does your solution need a cluster size tolerant topology?
Yes
No
Full Support Topology
Does your solution need CEI? (Common Event Infrastructure)
Yes
No
Does your solution have async with deferred response invocations?
Yes
No
Does your solution contain long running BPEL processes, or business state machines, or human tasks?
Yes
No
Does your solution contain JMS, MQ-JMS, or Native MQ bindings?
Yes
No
Is your solution comprised of only WEBSERVICE-SYNC or SCA-SYNC calls?
Yes
No
Basic Topology
Is message sequencing important?
Yes

Figure 9-2 Topology selection for the complex business process scenario

The main decision points for ITSO Bank are:

- A highly available solution with load balancing is required, so a clustered topology is needed.
- ITSO Bank have sufficient hardware resources to host the Full Support topology. They want to implement the Full Support topology (even if the current business process deployed to this topology does not require it) so that they can support the deployment of any type of application in the future.
- The Full Support topology is selected.

9.1.3 Overview of how to implement the Full Support topology

Figure 9-3 summarizes the steps to build this topology.

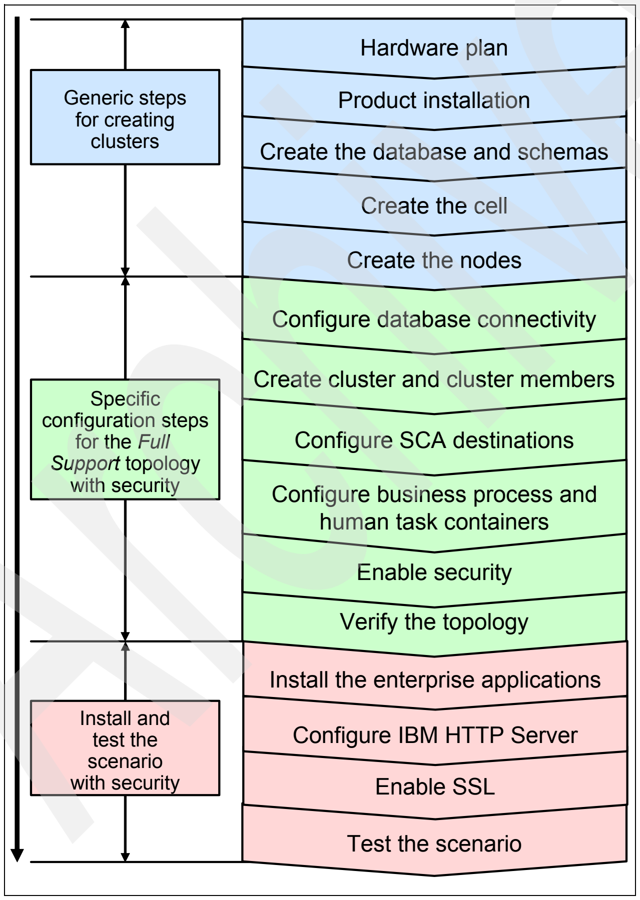

Generic steps for creating clusters
Hardware plan
Product installation
Create the database and schemas
Create the cell
Create the nodes
Specific configuration steps for the *Full Support* topology with security
Configure database connectivity
Create cluster and cluster members
Configure SCA destinations
Configure business process and human task containers
Enable security
Verify the topology
Install and test the scenario with security
Install the enterprise applications
Configure IBM HTTP Server
Enable SSL
Test the scenario

Figure 9-3 Building the Full Support topology for WebSphere Process Server

Figure 9-4 shows the topology that we build in this chapter.

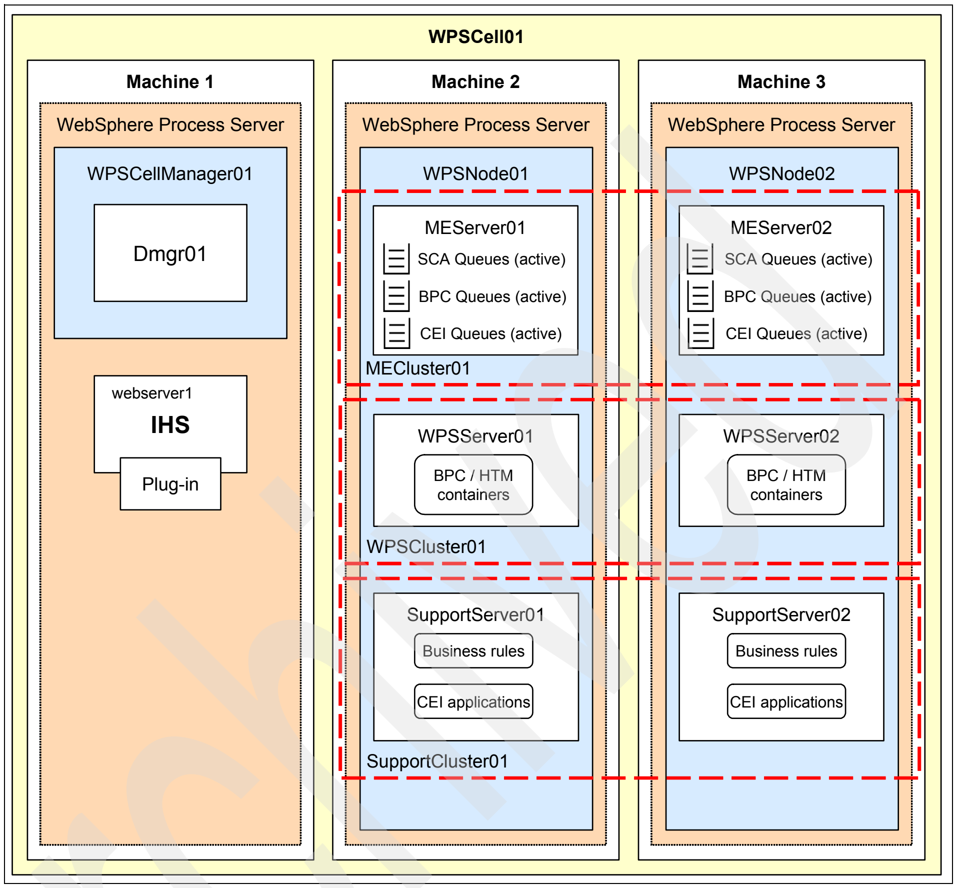


Figure 9-4 Full Support topology implementation

9.2 Generic steps for creating WebSphere Process Server clusters

This section describes the hardware plan and describes how to prepare a WebSphere Process Server cluster.

9.2.1 Hardware plan for the topology

This section describes the hardware that is required to support the Full Support topology. In this topology (illustrated in Figure 9-5 on page 326), three machines are used to support the cell hosting the business module and one machine is used to host the client and Web services applications for testing purposes. A final shared machine provides the database functionality. The hardware plan includes:

- The first machine (`wps0`) hosts the deployment manager and the IBM HTTP Server and associated plug-in.
- The second and third machines (`wps1` and `wps2`) host three clusters, as well as the node agents, in a horizontally clustered configuration:
  - The business cluster (also hosting the presentation logic)
  - A separate messaging cluster
  - The support cluster

  These machines need more resources because they are hosting four JVMs each.

  > **Note:** In a production environment with high throughput, the bottleneck might be the messaging engine response. In this case, the messaging engine cluster might be hosted on separate hardware to the business cluster. Similarly, if the support application (CEI and BRM) footprint becomes high, there is a possibility that this might impact the business application. Again, you can host this cluster on different hardware to alleviate any impact.

- The fourth machine (`wps3`) hosts the Web services application in a stand-alone WebSphere Application Server.
- The fourth machine (`db2`), not shown in the diagram, hosts the databases that are required to support WebSphere Process Server (MEDB, WPRCSDB, BPCDB, and CEIDB).
- The final machine (`ldap`), not shown in the diagram, hosts the LDAP ITDS V6.03 server and an associated DB2 instance.

  > **Note:** Although this is the configuration that we used, you can choose any combination of machines to create this topology.

Table 9-1 describes these machines.

Table 9-1 Systems used for this topology

System name	Host name	Operating system
wps0	wps0.itso	Windows 2003 Server
wps1	wps1.itso	Windows 2003 Server
wps2	wps2.itso	Windows 2003 Server
wps3	wps3.itso	Windows 2003 Server
db2	db2.itso	Red Hat Advanced Server 4 U4
ldap	ldap.itso	Red Hat Advanced Server 4 U2

Note: To help follow along with this chapter, you might want to define a hosts file on each of your machines. The hosts file maps the host names that are defined in Table 9-1 to the IP address of each of your machines.

You can find the hosts file on Windows platforms:

<*WINDOWS_HOME*>\system32\drivers\etc

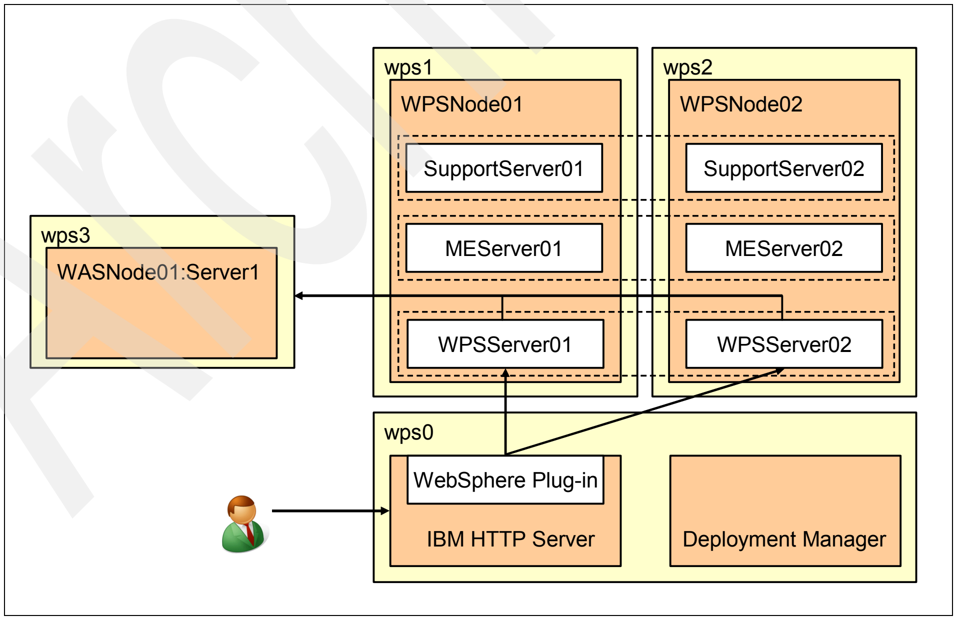


Figure 9-5 Hardware plan for the full support topology

The end-to-end flow for this topology is (Figure 9-5):

1. A user makes a request from a browser to the server, which will load balance through the plug-in.
2. IBM HTTP Server passes the request to the front-end client application, running as a JSP on `WPSNode01:WPSServer01` or `WPSNode02:WPSServer02` on either `wps1` or `wps2` respectively.
3. The client application is configured to forward the request to the business application running on the same server (`wps1` or `wps2`).
4. The business process application starts the account closure process, calls the appropriate Web service (running on `wps3`) to gather account details and generates a human task (specific to this scenario). The user receives a response that the process started successfully.
5. Because the request has now been submitted the continuation of this account closure task requires the user to check for new account closure confirmation instances from the presentation application.
6. If the user confirms an account closure instance, the process continues to completion by means of a call to the Web services closure service that is running in the Web services layer on `wps3`.
7. If the user denies an account closure instance, the process stops and the account is not closed. No call is made to the Web services layer in this case.
8. A confirmation is returned to the user either stating the account has been closed or that the process has been terminated.

9.2.2 Product installation

It is assumed that the following products have been installed:

- WebSphere Process Server V6.0.2
- IBM HTTP Server V6
- Global Security Kit for IBM HTTP Server
- DB2 Universal Database Enterprise Server V8.2
- IBM Tivoli Directory Server V6

Note: For information about how to install these products, refer to Appendix B, “Product installation” on page 429.

9.2.3 Creating the databases

This section describes how to create the messaging engine database, WebSphere Process Server common database, and the WebSphere Process Server BPEDB database.

You do not need to use the same names for your databases that we use in our examples, but your environment must be consistent if you use other names for the databases. You can find more information about the databases at the following Web site:

`http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wsps.602.ins.doc/doc/cins_db_specs.html`

Creating the messaging database

The first task is to create the messaging database. We assume the Relational Database Management System (RDBMS) is DB2, the instance is called `wpsinst1`, and the administrative user name is `wpsinst1`.

Log in as the database administrative user and issue the following commands from a DB2 command window to create the required database:

```
db2start
db2 create database MEDB
```

If the create command is successful, you see this message:

```
DB20000I The CREATE DATABASE command completed successfully
```

Creating the WebSphere Process Server common database

The next task is to create the common database for WebSphere Process Server, `WPRCSDB`. This database is the shared database that is used by applications such as the Failed Event Manager, Relationship, and CBE Browser.Log in as the database administrative user and issue the following commands from a DB2 command window to create the required database:

```
db2start
db2 create database WPRCSDB
```

If the create command is successful, you see the message:

```
DB20000I The CREATE DATABASE command completed successfully
```

Creating the WebSphere Process Server BPEDB database

The next task is to create the `BPEDB` database for WebSphere Process Server business container. This database stores the process-related information.

The DB2 commands for generating the `BPEDB` database is located in the script %*WAS_HOME*%\dbscripts\ProcessChoreographer\DB2\createDatabase.sql. This script exists on any system with the WebSphere Process Server product installed. To create the `BPEDB` database:

1. Transfer this file to the db2 server. On our environment, this script was at C:\IBM\WebSphere\ProcServer\dbscripts\ProcessChoreographer\DB2.

   Attention: If you transfer these files from Windows to UNIX, there might be return characters (`\r`) at the end of each line. To run the DDL scripts, you must remove these characters. A UNIX utility such as `dos2unix` will remove these characters.

2. Log in as the database administrative user (in this example `wpsinst1`) and issue the following commands from a DB2 command window to create the required database:

   ```
   db2start
   db2 -tf createDatabase.sql
   ```

   Note: The default script creates a database called `BPEDB` with the schema set as the currently logged in user (`wpsinst1`). You can modify the script to specific requirements.

When completed successfully, the `BPEDB` database exists.

Listing the created databases

Confirm that all three databases were successfully created in DB2:

1. Run the following command to list the databases that are defined to DB2:

   ```
   db2 list db directory
   ```

2. The three databases should be listed. Example 9-1 shows the first of these (`WPRCSDB`).

Example 9-1 System database directory

```
System Database Directory

Number of entries in the directory = 3

Database 1 entry:

Database alias                       = WPRCSDB
Database name                        = WPRCSDB
```

9.2.4 Copying the database driver files

The Java driver files for DB2 are located on the database system. In our DB2 example, they are in the directory `/opt/IBM/db2/V8.1/java`.

On each of the nodes and the deployment manager, copy the database driver files to a standard location. In this example, the standard location for the driver files is c:\db2client.

9.2.5 Creating the messaging database schemas

The messaging database includes four schemas that support the messaging engines in this topology:

- SCA Application
- SCA System
- Business processes
- CEI

You can create the DB2 commands that generate the schemas on any system with the WebSphere Process Server product installed. On any system where WebSphere Process Server is installed, create the data definition language (DDL) files for the Service Integration Bus (SIB) schemas as detailed in Table 9-2.

Table 9-2 WebSphere Process Server SCA schemas

Schema name	**Database**	**Purpose**
SCASYS	MEDB	System bus messaging store
SCAAPP	MEDB	Application bus messaging store
BPCMSG	MEDB	Business process messaging store
CEIMSG	MEDB	CEI messaging store

In this example:

- The database user name is `wpsinst1`.
- The script to generate the schemas is located in the %*WAS_HOME*%\bin directory (for example, C:\IBM\WebSphere\ProcServer\bin).
- The database target for the DDL generated is running on a Linux system.

To create the messaging database schemas, follow these steps:

1. Open a command prompt and issue the following command:

   ```
   cd C:\IBM\WebSphere\ProcServer\bin
   ```

   Note: Each of the commands in this series of steps is a single line although in the format of this book they might wrap. Also, the commands assume that the database is running on a UNIX platform. If you are using Windows, you need to replace -platform unix with `-platform windows`.

2. Issue the following command for the SCASYS DDL generation - SCA System Schema:

   ```
   sibDDLGenerator.bat -system db2 -version 8.1 -platform unix -schema
   SCASYS -statementend ; -user wpsinst1 >
   c:\createSIBSchema_SCASYS.ddl
   ```

3. Issue the following command for the SCAAPP DDL generation - SCA Application Schema:

   ```
   sibDDLGenerator.bat -system db2 -version 8.1 -platform unix -schema
   SCAAPP -statementend ; -user wpsinst1 >
   c:\createSIBSchema_SCAAPP.ddl
   ```

4. Issue the following command for the BPCMSG DDL generation - Business Process Schema:

```
sibDDLGenerator.bat -system db2 -version 8.1 -platform unix -schema
BPCMSG -statementend ; -user wpsinst1 >
c:\createSIBSchema_BPCMSG.ddl
```

5. Issue the following command for the CEIMSG DDL generation - CEI Messaging Schema:

```
sibDDLGenerator.bat -system db2 -version 8.1 -platform unix -schema
CEIMSG -statementend ; -user wpsinst1 >
c:\createSIBSchema_CEIMSG.ddl
```

6. Transfer the DDL files to DB2 database server. In this example, we transfer the DDL files to /tmp on the database server.

Attention: If you transfer these files from Windows to UNIX, there might be return characters (`\r`) at the end of each line. To run the DDL scripts, you must remove these characters. A UNIX utility such as `dos2unix` will remove these characters.

7. Login to the DB2 server as the appropriate user (`wpsinst1`) and run the following commands to create the tables:

```
db2 connect to MEDB
db2 -tf /tmp/createSIBSchema_SCAAPP.ddl
db2 -tf /tmp/createSIBSchema_SCASYS.ddl
db2 -tf /tmp/createSIBSchema_BPCMSG.ddl
db2 -tf /tmp/createSIBSchema_CEIMSG.ddl
```

Note: These DB2 commands report various informational index and primary key messages, such as:

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

These are not errors. You can ignore these messages.

9.2.6 Configuring WebSphere Process Server cell

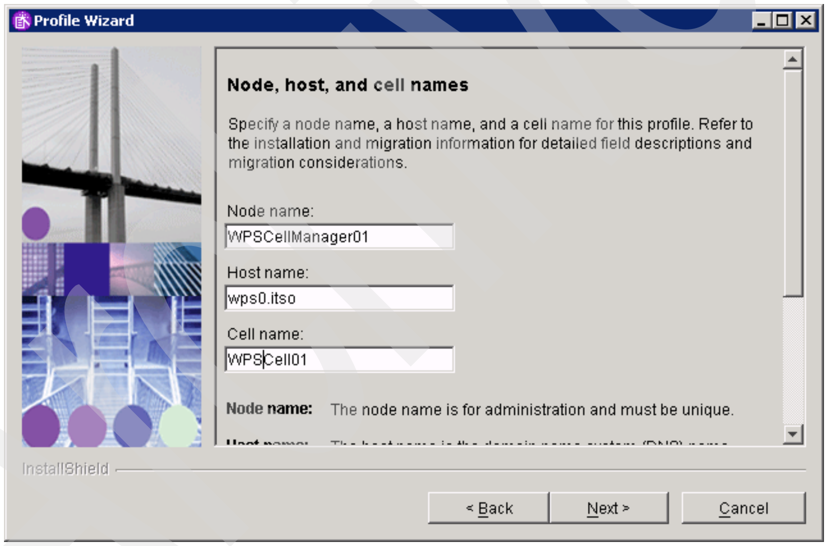
This task creates the WebSphere Process Server cell which is achieved by creating a deployment manager profile.

1. Login to the `wps0.itso` machine which becomes the deployment manager.
2. Open the profile creation wizard by selecting **Start** → **All Programs** → **IBM WebSphere** → **Process Server 6.0** → **Profile creation wizard**.

Note: For information about how to launch the Profile creation wizard on other platforms, see:

`http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tpro_instances.html`

3. In the Profile creation wizard, click **Next**.
4. Select **Deployment manager profile** and click **Next**.
5. Leave the default profile name as *Dmgr01* and click **Next**.
6. Leave the default profile directory and click **Next**.
7. Specify the Node name, Host name and Cell name on the subsequent panel and click **Next**, as shown in Figure 9-6. We used:
   - Node name: `WPSCellManager01`
   - Host name: `wps0.itso`
   - Cell name: `WPSCell01`



Node, host, and cell names

Specify a node name, a host name, and a cell name for this profile. Refer to the installation and migration information for detailed field descriptions and migration considerations.

Node name: WPSCellManager01

Host name: wps0.itso

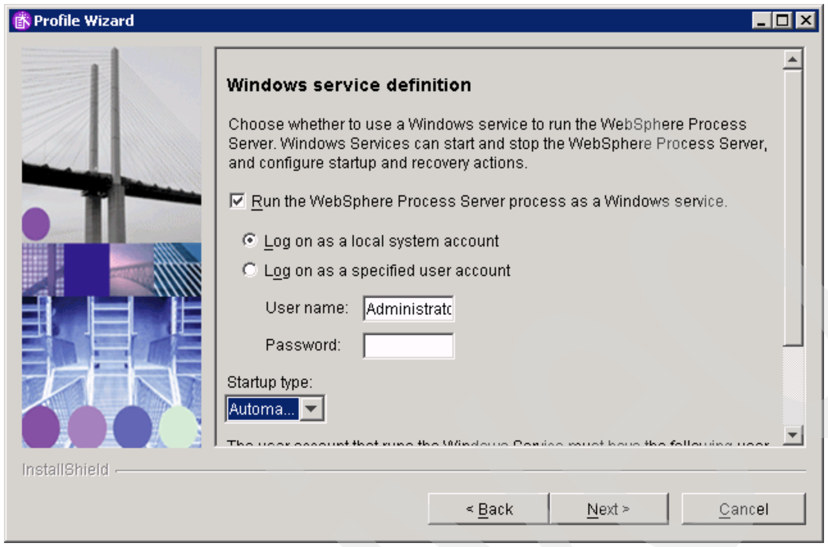
Cell name: WPSCell01

Node name: The node name is for administration and must be unique.

Figure 9-6 Node, host and cell name configuration

8. Leave port numbers at their default settings and click **Next**.

9. In the Windows service definition panel, select **Automatic** and click **Next,** as shown in Figure 9-7.



Windows service definition

Choose whether to use a Windows service to run the WebSphere Process Server. Windows Services can start and stop the WebSphere Process Server, and configure startup and recovery actions.

[✓] Run the WebSphere Process Server process as a Windows service.

(●) Log on as a local system account

( ) Log on as a specified user account

User name: Administratc

Password:

Startup type: Automa...

Figure 9-7 Windows service settings

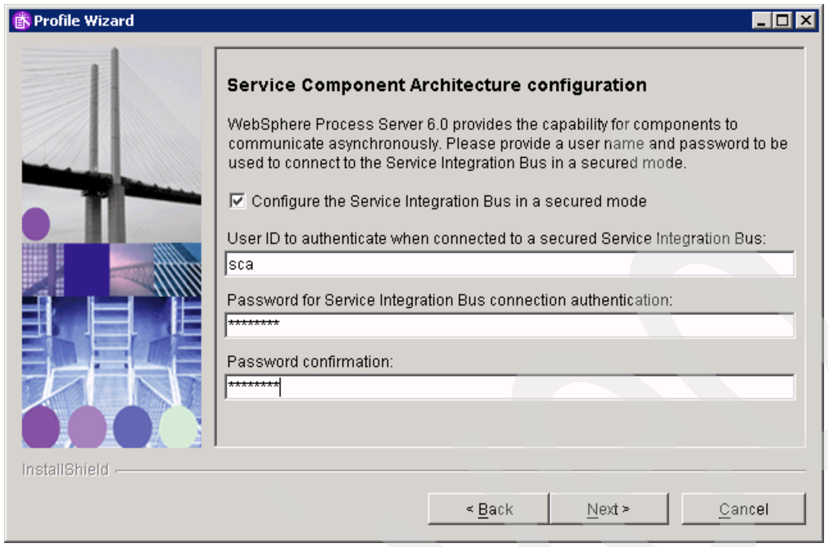
10. On the Service Component Architecture configuration panel, select **Configure the Service Integration Bus in a secured mode** and enter the user ID and password for secured SCA communications as detailed in Table 9-3.

    When you enable global security, this user must exist within the user registry. In this example, we use IBM Tivoli Directory Server. This user is for inter-bus communication and not database access.

Table 9-3 SCA User information

Name	Value
User Id	sca
Password	itso4you

Figure 9-8 shows these security settings.


Profile Wizard

Service Component Architecture configuration

WebSphere Process Server 6.0 provides the capability for components to communicate asynchronously. Please provide a user name and password to be used to connect to the Service Integration Bus in a secured mode.

[✓] Configure the Service Integration Bus in a secured mode

User ID to authenticate when connected to a secured Service Integration Bus: sca

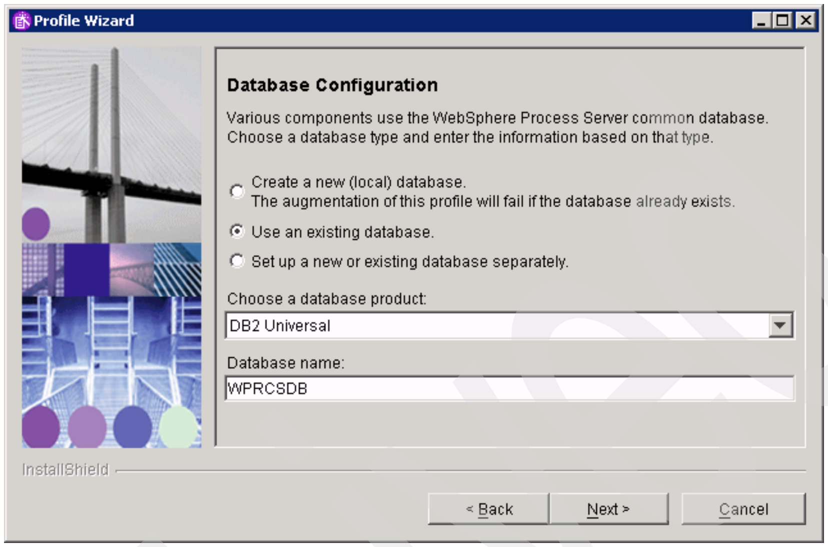
Password for Service Integration Bus connection authentication: ********

Password confirmation: ********

Figure 9-8 SCA security settings

11.Click **Next** to move to the Database Configuration panel.

12. From the Database Configuration panel, select **Use an existing database.** This example uses DB2 Universal as the database product and `WPRCSDB` as the database name as shown in Figure 9-9.



Database Configuration

Various components use the WebSphere Process Server common database. Choose a database type and enter the information based on that type.

- [ ] Create a new (local) database. The augmentation of this profile will fail if the database already exists.
- [✓] Use an existing database.
- [ ] Set up a new or existing database separately.

Choose a database product: DB2 Universal

Database name: WPRCSDB

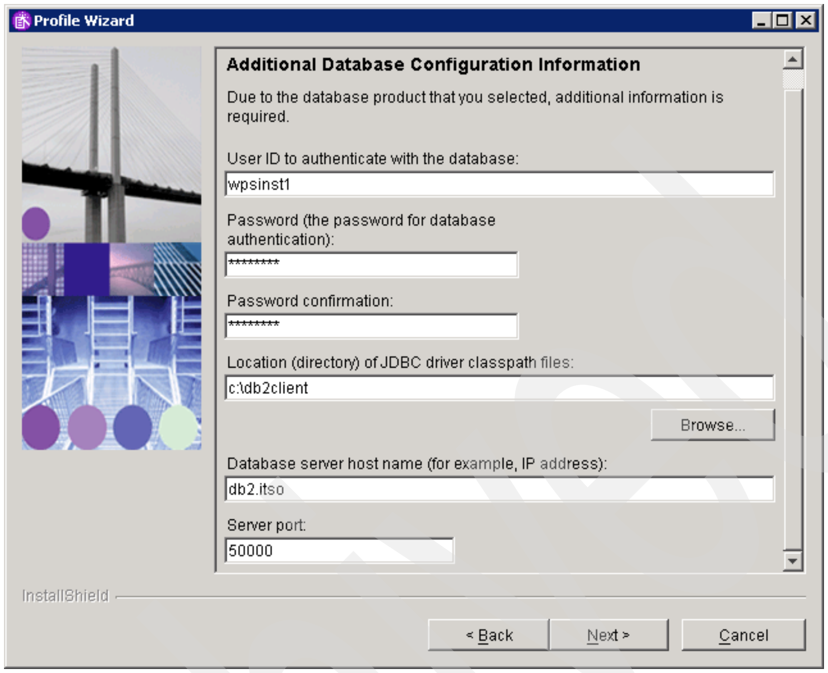
Figure 9-9 WPRCSDB database settings

13. Click **Next** to continue. In the subsequent database configuration panel, set the values as shown in Table 9-4 and click **Next**.

Table 9-4 Database settings

Name	Value
User ID to authenticate with the database	wpsinst1
Password (for database authentication)	itso4you
Password confirmation	itso4you
Location (directory) of JDBC driver classpath files	c:\db2client
Database server host name	db2.itso
Server port	50000

Figure 9-10 shows these settings.



Additional Database Configuration Information

Due to the database product that you selected, additional information is required.

User ID to authenticate with the database:
wpsinst1

Password (the password for database authentication):
****

Password confirmation:
****

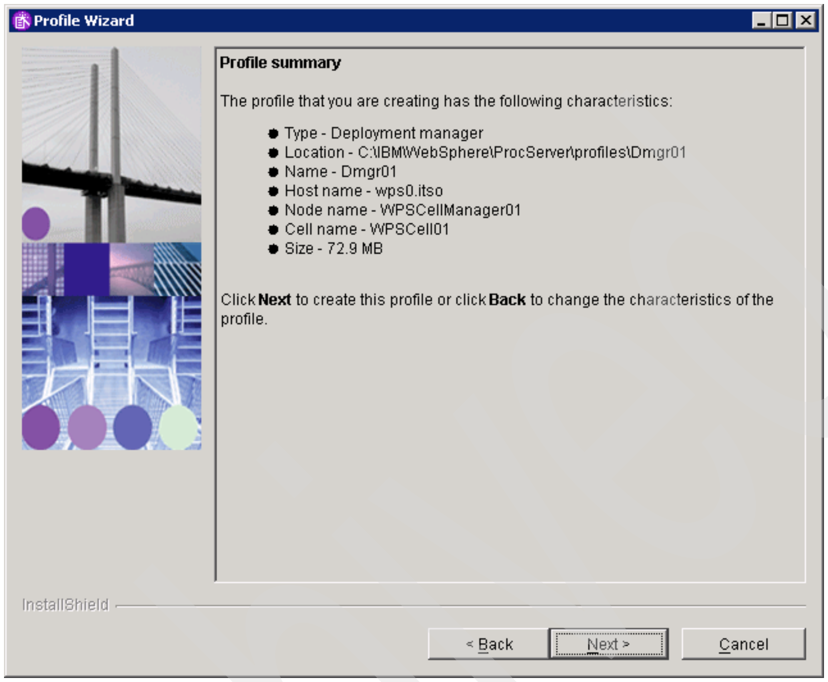
Location (directory) of JDBC driver classpath files:
c:\db2client

Database server host name (for example, IP address):
db2.itso

Server port:
50000

Figure 9-10 Additional Database Configuration

14. Confirm that the details are correct on the Profile summary panel, and click **Next**, as shown in Figure 9-11.



Profile summary

The profile that you are creating has the following characteristics:

- Type - Deployment manager
- Location - C:\IBM\WebSphere\ProcServer\profiles\Dmgr01
- Name - Dmgr01
- Host name - wps0.itso
- Node name - WPSCellManager01
- Cell name - WPSCell01
- Size - 72.9 MB

Click **Next** to create this profile or click **Back** to change the characteristics of the profile.

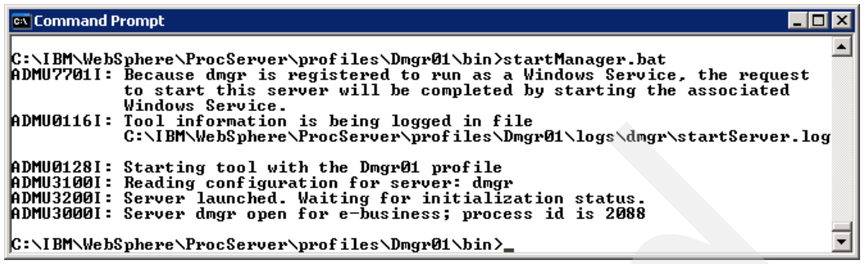
Figure 9-11 Deployment manager creation summary

15. A panel displays the progress. When complete, clear the **Launch the First Steps console** check box and click **Finish**. The deployment manager now exists.

16. Open a command prompt on wps0, change directory to %*WAS_HOME*%\profiles\Dmgr01\bin (for example, C:\IBM\WebSphere\ProcServer\profiles\Dmgr01\bin), and issue the following command:

```
startManager.bat
```

Figure 9-12 shows the output from this command.



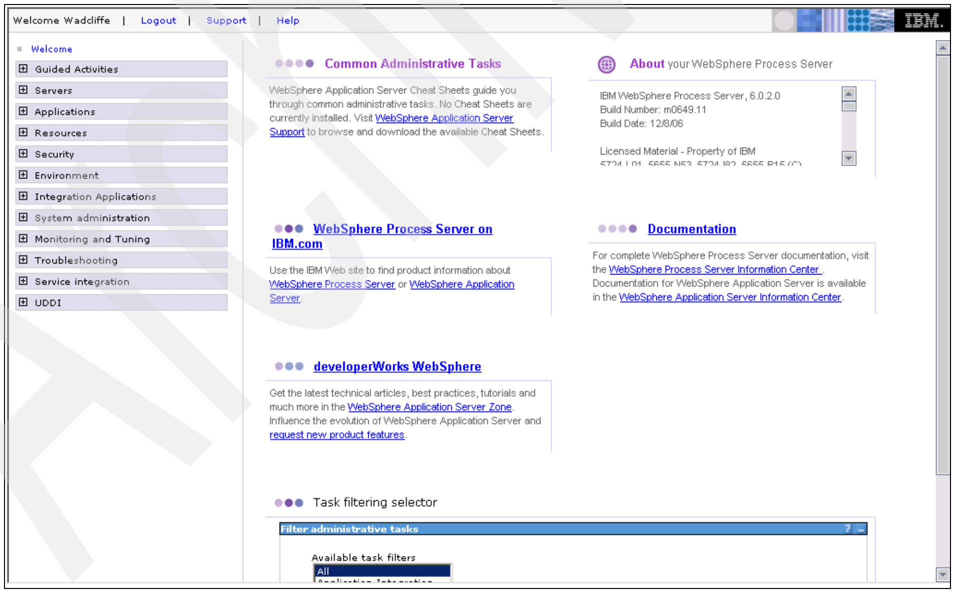
```
C:\IBM\WebSphere\ProcServer\profiles\Dmgr01\bin>startManager.bat
ADMU7701I: Because dmgr is registered to run as a Windows Service, the request
           to start this server will be completed by starting the associated
           Windows Service.
ADMU0116I: Tool information is being logged in file
           C:\IBM\WebSphere\ProcServer\profiles\Dmgr01\logs\dmgr\startServer.log

ADMU0128I: Starting tool with the Dmgr01 profile
ADMU3100I: Reading configuration for server: dmgr
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server dmgr open for e-business; process id is 2088

C:\IBM\WebSphere\ProcServer\profiles\Dmgr01\bin>_
```

Figure 9-12 Starting the deployment manager

It should now be able to connect to the Administrative Console running on the deployment manager by opening a browser to:

```
http://wps0.itso:9060/ibm/console
```

Log in to the Administrative Console (as any user). Figure 9-13 shows the Web page that opens. It might be necessary to set the available task filters to *All* in the Filter administrative tasks panel and click **Apply** on the Welcome page of the Administrative Console to override the default filtering of the available options.



Common Administrative Tasks

WebSphere Application Server Cheat Sheets guide you through common administrative tasks. No Cheat Sheets are currently installed. Visit WebSphere Application Server Support to browse and download the available Cheat Sheets.

About your WebSphere Process Server

IBM WebSphere Process Server, 6.0.2.0
Build Number: m0649.11
Build Date: 12/8/06

Licensed Material - Property of IBM

WebSphere Process Server on IBM.com

Use the IBM Web site to find product information about WebSphere Process Server or WebSphere Application Server.

Documentation

For complete WebSphere Process Server documentation, visit the WebSphere Process Server Information Center . Documentation for WebSphere Application Server is available in the WebSphere Application Server Information Center.

developerWorks WebSphere

Get the latest technical articles, best practices, tutorials and much more in the WebSphere Application Server Zone. Influence the evolution of WebSphere Application Server and request new product features.

Task filtering selector

Filter administrative tasks

Available task filters: All

Figure 9-13 First login to the Administrative Console

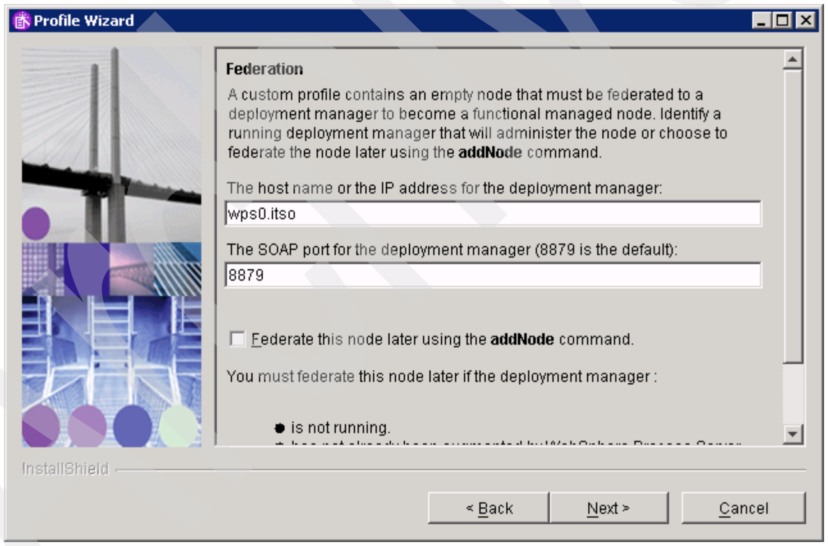
9.2.7 Creating the nodes

The next step in creating this topology is to create the profiles and associated nodes for the topology. The deployment manager must be running on `wps0.itso` to enable node federation, the nodes are created on `wps1.itso` and `wps2.itso`.

Note: Ensure that the time difference between the deployment manager machine and the custom profile machines is within *five* minutes.

To create the nodes, follow these steps:

1. On `wps1` open the profile creation wizard from the **Start** → **All Programs** → **IBM WebSphere** → **Process Server 6.0** → **Profile creation wizard**.
2. In the profile creation wizard, click **Next**.
3. Select **Custom profile** and click **Next**.
4. Enter the host name of the deployment manager, in this example `wps0.itso`, leave the SOAP port as the default (`8879`), ensure that **Federate this node later** is *not* selected, and click **Next,** as shown in Figure 9-14.



Federation

A custom profile contains an empty node that must be federated to a deployment manager to become a functional managed node. Identify a running deployment manager that will administer the node or choose to federate the node later using the **addNode** command.

The host name or the IP address for the deployment manager:
wps0.itso

The SOAP port for the deployment manager (8879 is the default):
8879

☐ Federate this node later using the **addNode** command.

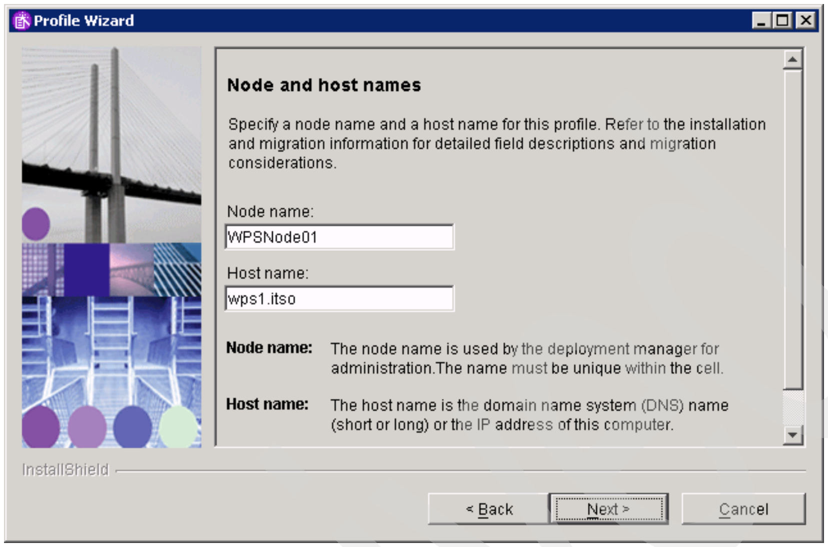
You must federate this node later if the deployment manager :

- is not running.

Figure 9-14 Deployment manager location and federation

5. Set the Profile name appropriately (`WPSNode01`) and click **Next**.
6. Accept the default profile directory and click **Next**.

7. Specify the node name (`WPSNode01`) and host name (`wps1.itso`) on the subsequent panel and click **Next**, as shown in Figure 9-15.



Node and host names

Specify a node name and a host name for this profile. Refer to the installation and migration information for detailed field descriptions and migration considerations.

Node name:
WPSNode01

Host name:
wps1.itso

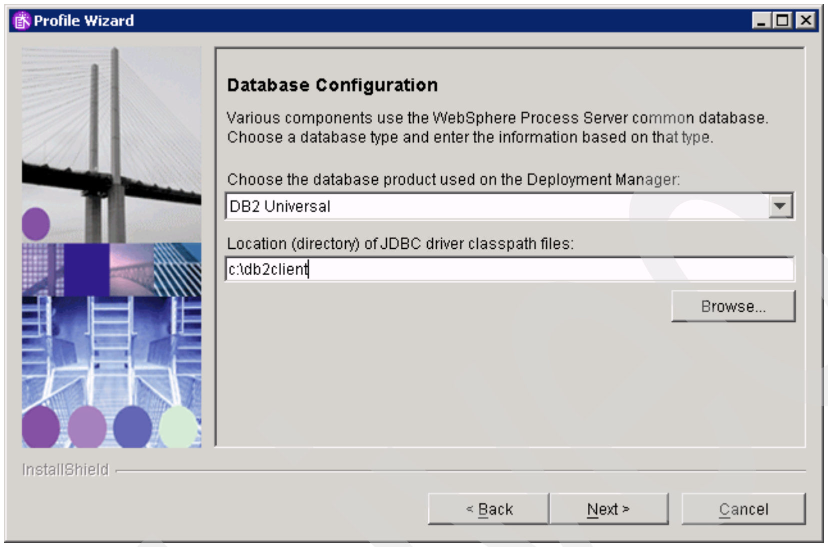
Node name: The node name is used by the deployment manager for administration.The name must be unique within the cell.

Host name: The host name is the domain name system (DNS) name (short or long) or the IP address of this computer.

Figure 9-15 Node and host name

8. Leave the default port numbers and click **Next**. Specify the node name (`WPSNode01`) and host name (`wps1.itso`) on the subsequent panel and click **Next**.

9. In the Database Configuration panel, set the database product to **DB2 Universal** and the JDBC driver classpath to **c:\db2client** as shown in Figure 9-16. Click **Next**.



Database Configuration

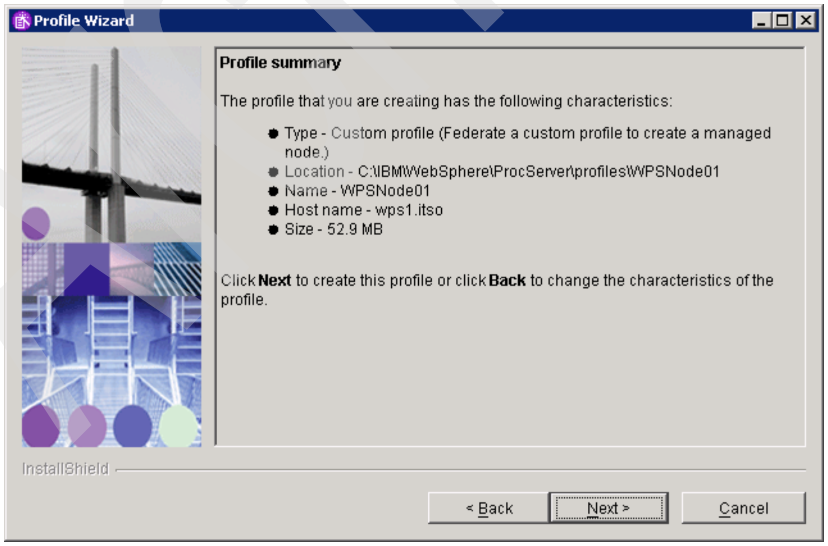
Various components use the WebSphere Process Server common database. Choose a database type and enter the information based on that type.

Choose the database product used on the Deployment Manager: DB2 Universal

Location (directory) of JDBC driver classpath files: c:\db2client

Figure 9-16 Database configuration

10. Confirm that the details are correct on the Profile summary panel and click **Next**, as shown in Figure 9-17.



Profile summary

The profile that you are creating has the following characteristics:

- Type - Custom profile (Federate a custom profile to create a managed node.)
- Location - C:\IBM\WebSphere\ProcServer\profiles\WPSNode01
- Name - WPSNode01
- Host name - wps1.itso
- Size - 52.9 MB

Click **Next** to create this profile or click **Back** to change the characteristics of the profile.

Figure 9-17 Node creation summary

11. A panel displays the progress. When complete, clear the **Launch the First Steps console** check box and click **Finish**.

The node now exists and is federated into the cell. The federation process starts the node automatically.

Repeat these steps for the second node using the values shown in Table 9-5.

Table 9-5 Second node details

Setting	Value
Profile name	WPSNode02
Node name	WPSNode02
Host name	wps2.itso

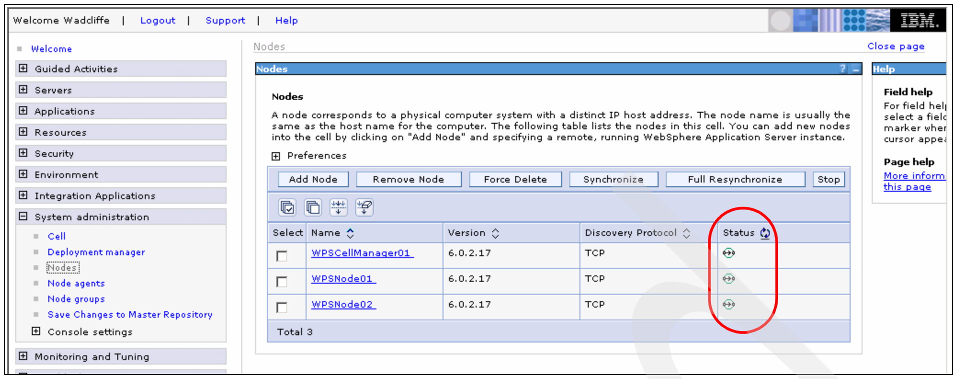
After you have created both nodes, verify that they are running in the Administrative Console by following these steps:

1. Open a browser to `http://wps0.itso:9060/ibm/console`.

   Note: You need to logout from any existing console sessions.

2. Navigate to **System administration** → **Nodes** and verify that all nodes are running as indicated by the green symbol next to each node, as shown in Figure 9-18.

   Note: Upon initial login to the Administrative Console, you might need to change the Task filter selection to *All* and click **Apply** to view all available functionality.



Nodes

A node corresponds to a physical computer system with a distinct IP host address. The node name is usually the same as the host name for the computer. The following table lists the nodes in this cell. You can add new nodes into the cell by clicking on "Add Node" and specifying a remote, running WebSphere Application Server instance.

Select	Name	Version	Discovery Protocol	Status
☐	WPSCellManager01	6.0.2.17	TCP	⟷
☐	WPSNode01	6.0.2.17	TCP	⟷
☐	WPSNode02	6.0.2.17	TCP	⟷
Total 3				

Figure 9-18 Successful node installation and startup

At this point, you have a WebSphere Process Server cell with the configuration as shown in Figure 9-19.

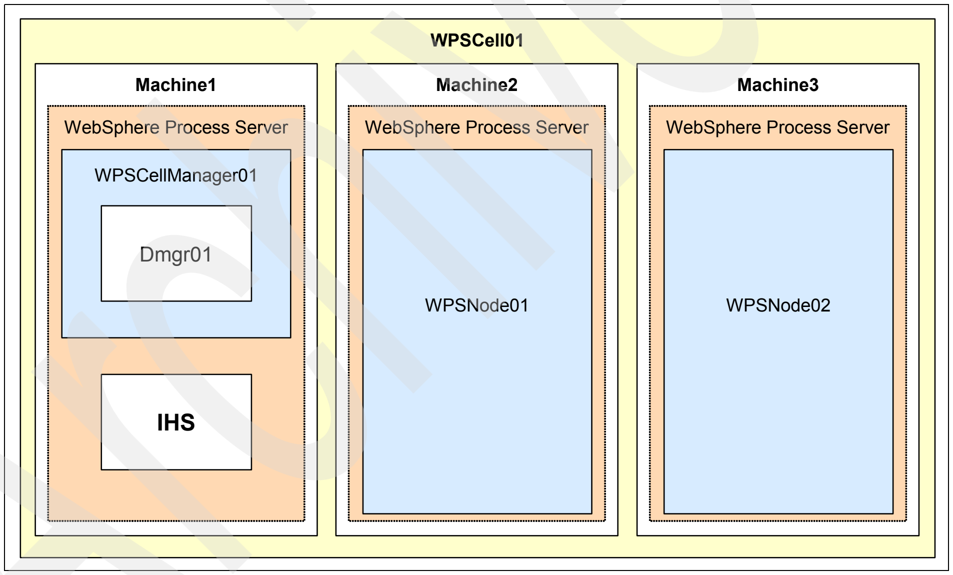


Figure 9-19 Cell and nodes defined

9.3 Specific configuration steps for the Full Support topology

This section describes the following:

- ► Configuring database connectivity for the Full Support topology
- ► Configuring the cell
- ► Setting the preferred server for messaging engines
- ► CEI configuration for DB2
- ► Deploying the CEI applications
- ► Enabling security
- ► Configuring IBM HTTP Server
- ► Enabling SSL
- ► Configuring the remote enterprise service
- ► Verifying the topology

9.3.1 Configuring database connectivity for the Full Support topology

This section describes how to connect successfully to the databases that are required by WebSphere Process Server in the Full Support topology, including the following databases:

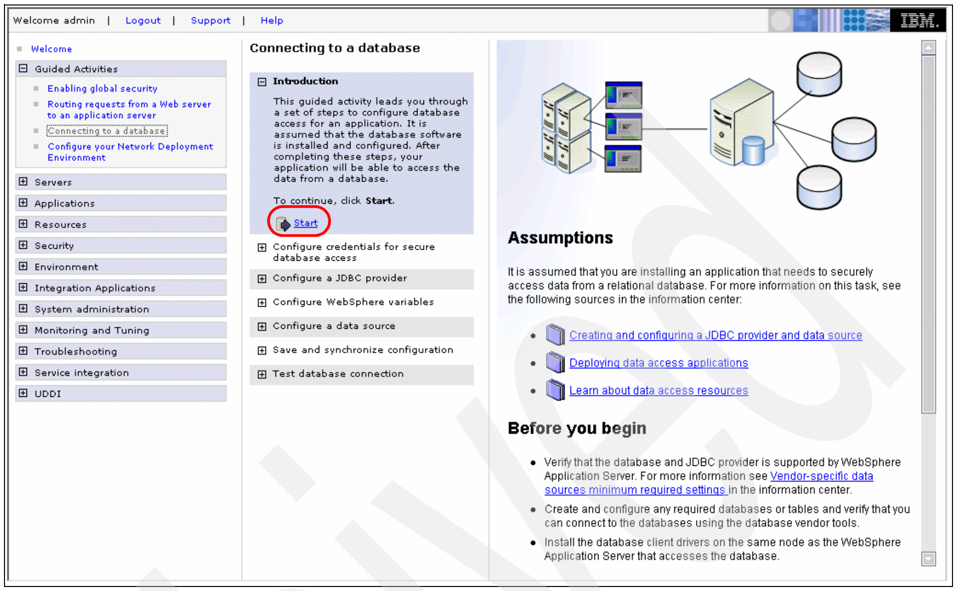
- ► MEDB - Messaging database
- ► BPEDB - Business container database

Upon cell creation the data source jdbc/WPSDB is created automatically, along with the data source provider at cell scope, DB2 Universal JDBC Driver Provider (XA). You can reuse this data source provider. Follow these steps:

1. Open the cell Administrative Console on `wps0` and login. We used the following URL:

   ```
   http://wps0.itso:9060/ibm/console
   ```

2. Navigate to **Guided Activities** → **Connecting to a database** and click **Start**, as shown in Figure 9-20.



Welcome admin | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Connecting to a database

Introduction

This guided activity leads you through a set of steps to configure database access for an application. It is assumed that the database software is installed and configured. After completing these steps, your application will be able to access the data from a database.

To continue, click **Start**.

Start

- Configure credentials for secure database access
- Configure a JDBC provider
- Configure WebSphere variables
- Configure a data source
- Save and synchronize configuration
- Test database connection

Assumptions

It is assumed that you are installing an application that needs to securely access data from a relational database. For more information on this task, see the following sources in the information center:

- Creating and configuring a JDBC provider and data source
- Deploying data access applications
- Learn about data access resources

Before you begin

- Verify that the database and JDBC provider is supported by WebSphere Application Server. For more information see Vendor-specific data sources minimum required settings in the information center.
- Create and configure any required databases or tables and verify that you can connect to the databases using the database vendor tools.
- Install the database client drivers on the same node as the WebSphere Application Server that accesses the database.

Figure 9-20 Starting the guided database activity

3. Select **Click to perform**.

Authenticating the alias

You need to set the user and password values for database access. Because you are creating all databases in the same DB2 instance (`wpsinst1`) with the same user (`wpsinst1`), you can reuse the existing authentication alias. You created this alias when you configured the deployment manager. By default, it is `WPSDBAlias`.

You can find step-by-step instructions for the configuration of an authentication alias in 9.3.1, “Configuring database connectivity for the Full Support topology” on page 345.

To set the user and password value, select **Next step** to continue to the Configure a JDBC provider panel.

Creating a JDBC provider

The next step creates a JDBC provider for use by the data sources later. Again you reuse the existing provider called `DB2 Universal JDBC Driver Provider`

(XA) that you defined at cell scope. To create a JDBC provider, select **Next step** to continue to the Configure WebSphere variables panel.

Setting the WebSphere variables

The JDBC provider uses WebSphere environment variables to identify the driver file locations. For these variables to resolve, you need to set their vales appropriately. The scope for these variables should be **Cell** or **Node**, as shown in Table 9-6 and Table 9-7.

Note: When setting the scope, the selection is not applied until you click **Apply**.

To set the WebSphere variables:

1. Select **Click to perform**.
2. Clear the node and server entries, and click **Apply** to set the active scope to the cell.
3. Create or modify each environment variable, and set the value as defined in Table 9-6 on page 347.

Table 9-6 Cell scope environment variables

Name	Value	Scope
UNIVERSAL_JDBC_DRIVER_PATH	c:\db2client	Cell
DB2UNIVERSAL_JDBC_DRIVER_PATH	c:\db2client	Cell
DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH	c:\db2client	Cell

4. When you have defined all three variables at cell level, click **Browse Nodes**. Select **WPSNode01** and click **OK**. Click **Apply** to set the active scope to the node.
5. Set the value of the node scope environment variable DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH, as shown in Table 9-7. Also delete the other two variables shown in Table 9-7.

   Attention: At the node scope, you *must* delete the environment variables UNIVERSAL_JDBC_DRIVER_PATH and DB2UNIVERSAL_JDBC_DRIVER_PATH.

Table 9-7 Node scope environment variables

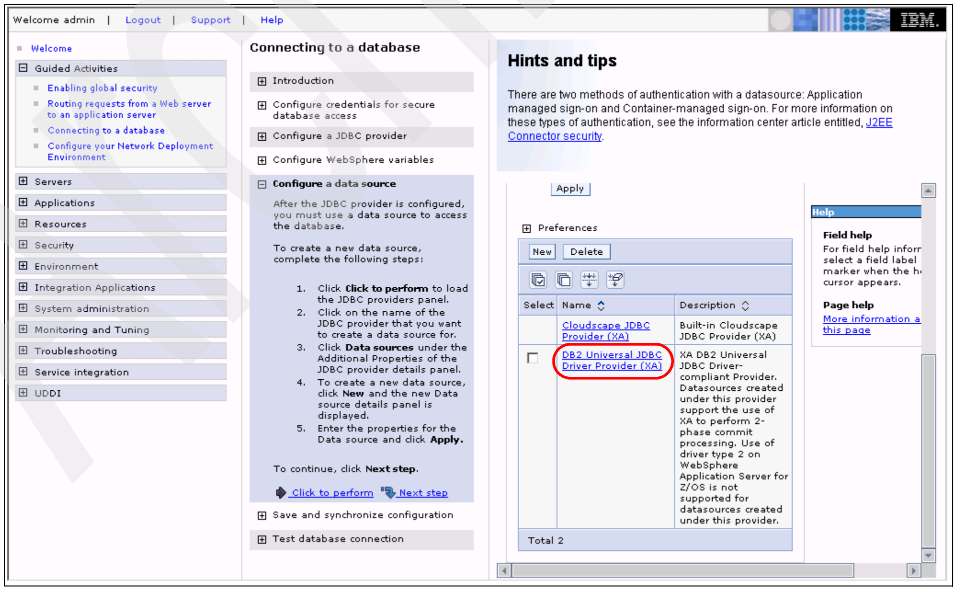
Name	Value	Scope
DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH	c:\db2client	WPSNode01 WPSNode02
UNIVERSAL_JDBC_DRIVER_PATH	*Delete this variable*	WPSNode01 WPSNode02
DB2UNIVERSAL_JDBC_DRIVER_PATH	*Delete this variable*	WPSNode01 WPSNode02

6. Repeat the process for `WPSNode02` to remove the two environment variables and set the third to the value shown in Table 9-7.
7. Save and synchronize these changes.
8. Click Next step.

Configuring a data source

The next step is to configure the data sources using the existing JDBC provider that you just created:

1. Select **Click to perform**.
2. In the new panel, select the JDBC provider that is defined at the cell scope **DB2 Universal JDBC Driver Provider (XA)**, as shown in Figure 9-21.



Connecting to a database

Configure a data source

After the JDBC provider is configured, you must use a data source to access the database.

To create a new data source, complete the following steps:

1. Click **Click to perform** to load the JDBC providers panel.
2. Click on the name of the JDBC provider that you want to create a data source for.
3. Click **Data sources** under the Additional Properties of the JDBC provider details panel.
4. To create a new data source, click **New** and the new Data source details panel is displayed.
5. Enter the properties for the Data source and click **Apply**.

To continue, click **Next step**.

Click to perform   Next step

Hints and tips

There are two methods of authentication with a datasource: Application managed sign-on and Container-managed sign-on. For more information on these types of authentication, see the information center article entitled, J2EE Connector security.

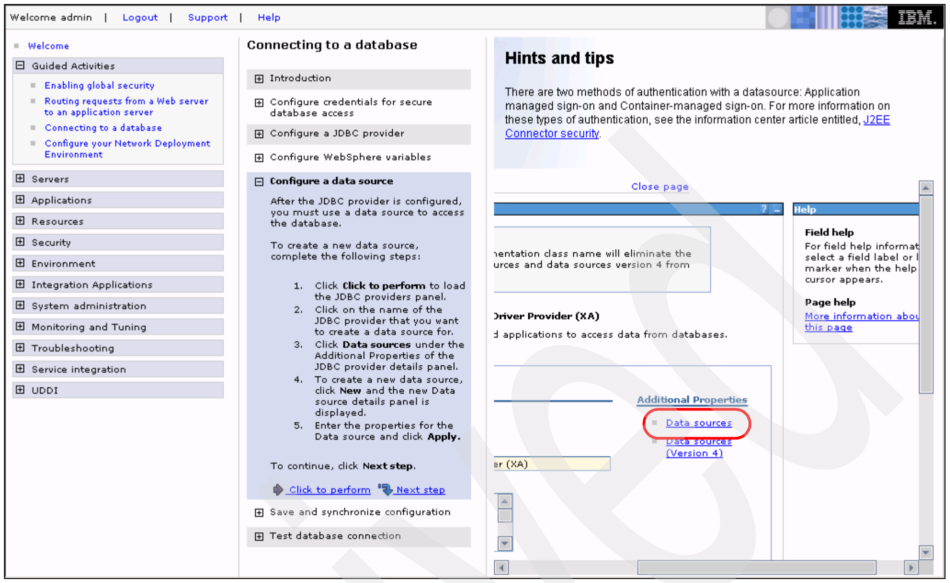
Select	Name	Description
	Cloudscape JDBC Provider (XA)	Built-in Cloudscape JDBC Provider (XA)
☐	DB2 Universal JDBC Driver Provider (XA)	XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on WebSphere Application Server for Z/OS is not supported for datasources created under this provider.

Total 2

For field help infor
select a field label
marker when the h
cursor appears.

Figure 9-21 JDBC provider selection for data source creation

3. Scroll to the right and select **Data sources,** as shown in Figure 9-22.



Connecting to a database

Configure a data source

After the JDBC provider is configured, you must use a data source to access the database.

To create a new data source, complete the following steps:

1. Click **Click to perform** to load the JDBC providers panel.
2. Click on the name of the JDBC provider that you want to create a data source for.
3. Click **Data sources** under the Additional Properties of the JDBC provider details panel.
4. To create a new data source, click **New** and the new Data source details panel is displayed.
5. Enter the properties for the Data source and click **Apply.**

To continue, click **Next step.**

Hints and tips

There are two methods of authentication with a datasource: Application managed sign-on and Container-managed sign-on. For more information on these types of authentication, see the information center article entitled, J2EE Connector security.

nentation class name will eliminate the
urces and data sources version 4 from

Driver Provider (XA)

d applications to access data from databases.

Field help

For field help informat
select a field label or l
marker when the help
cursor appears.

Page help

More information abou
this page

Figure 9-22 Selecting the additional data source property

4. You need to create a data source for:

   – Messaging database (MEDB)
   – Business container database (BPEDB)

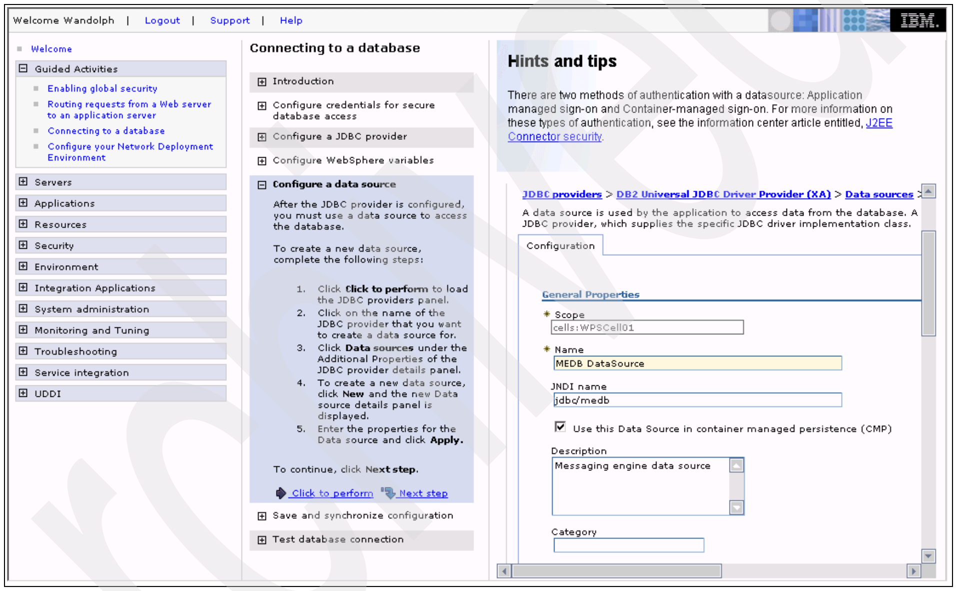
   Click **New** to create a data source and set the parameters, as shown in Table 9-8 under the MEDB Data Source Values column.

Table 9-8 Data source configuration

Parameter	**MEDB Data Source Values**	**BPEDB Data Source Values**
Name	MEDB DataSource	BPE DataSource
JNDI name	jdbc/medb	jdbc/bpedb
Description	Messaging engine data source	Business process container data source
Component-managed authentication alias	WPSDBAlias	WPSDBAlias
Authentication alias for XA recovery	Use component-managed authentication alias	Use component-managed authentication alias

Parameter	MEDB Data Source Values	BPEDB Data Source Values
Database name	MEDB	BPEDB
Driver type	4	4
Server name	db2.itso	db2.itso
Port	50000	50000

Figure 9-23 shows the MEDB data source configuration page. To enter these values, you need to scroll down in this panel.


Welcome Wandolph | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Connecting to a database

- Introduction
- Configure credentials for secure database access
- Configure a JDBC provider
- Configure WebSphere variables
- **Configure a data source**

  After the JDBC provider is configured, you must use a data source to access the database.

  To create a new data source, complete the following steps:

  1. Click **Click to perform** to load the JDBC providers panel.
  2. Click on the name of the JDBC provider that you want to create a data source for.
  3. Click **Data sources** under the Additional Properties of the JDBC provider details panel.
  4. To create a new data source, click **New** and the new Data source details panel is displayed.
  5. Enter the properties for the Data source and click **Apply.**

  To continue, click **Next step.**

  Click to perform Next step
- Save and synchronize configuration
- Test database connection

Hints and tips

There are two methods of authentication with a datasource: Application managed sign-on and Container-managed sign-on. For more information on these types of authentication, see the information center article entitled, J2EE Connector security.

JDBC providers > DB2 Universal JDBC Driver Provider (XA) > Data sources >

A data source is used by the application to access data from the database. A JDBC provider, which supplies the specific JDBC driver implementation class.

Configuration

General Properties

* Scope: cells:WPSCell01

* Name: MEDB DataSource

JNDI name: jdbc/medb

[✓] Use this Data Source in container managed persistence (CMP)

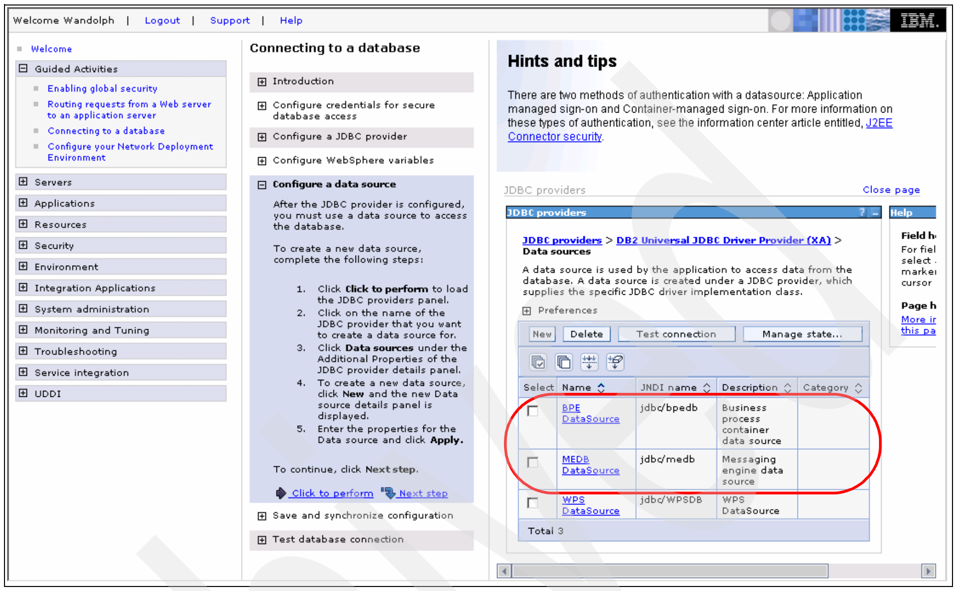
Description: Messaging engine data source

Category:

Figure 9-23 Data source configuration

5. Click **OK** to create the new data source. When created, save and synchronize the changes and click **OK** to complete.

6. Repeat this process for the `BPEDB` data source, as shown in Table 9-8 under the BPEDB DataSource Values column. Figure 9-24 shows all the data sources.



Connecting to a database

Introduction

Configure credentials for secure database access

Configure a JDBC provider

Configure WebSphere variables

Configure a data source

After the JDBC provider is configured, you must use a data source to access the database.

To create a new data source, complete the following steps:

1. Click **Click to perform** to load the JDBC providers panel.
2. Click on the name of the JDBC provider that you want to create a data source for.
3. Click **Data sources** under the Additional Properties of the JDBC provider details panel.
4. To create a new data source, click **New** and the new Data source details panel is displayed.
5. Enter the properties for the Data source and click **Apply.**

To continue, click **Next step**.

Click to perform   Next step

Save and synchronize configuration

Test database connection

Hints and tips

There are two methods of authentication with a datasource: Application managed sign-on and Container-managed sign-on. For more information on these types of authentication, see the information center article entitled, J2EE Connector security.

JDBC providers   Close page

JDBC providers

JDBC providers > **DB2 Universal JDBC Driver Provider (XA)** > **Data sources**

A data source is used by the application to access data from the database. A data source is created under a JDBC provider, which supplies the specific JDBC driver implementation class.

Preferences

Select	Name	JNDI name	Description	Category
☐	BPE DataSource	jdbc/bpedb	Business process container data source	
☐	MEDB DataSource	jdbc/medb	Messaging engine data source	
☐	WPS DataSource	jdbc/WPSDB	WPS DataSource	
Total 3				

Figure 9-24 Data sources created

7. Click **Next step** and then **Next step** again in the Save and synchronize configuration panel (because you have already saved and synchronized).

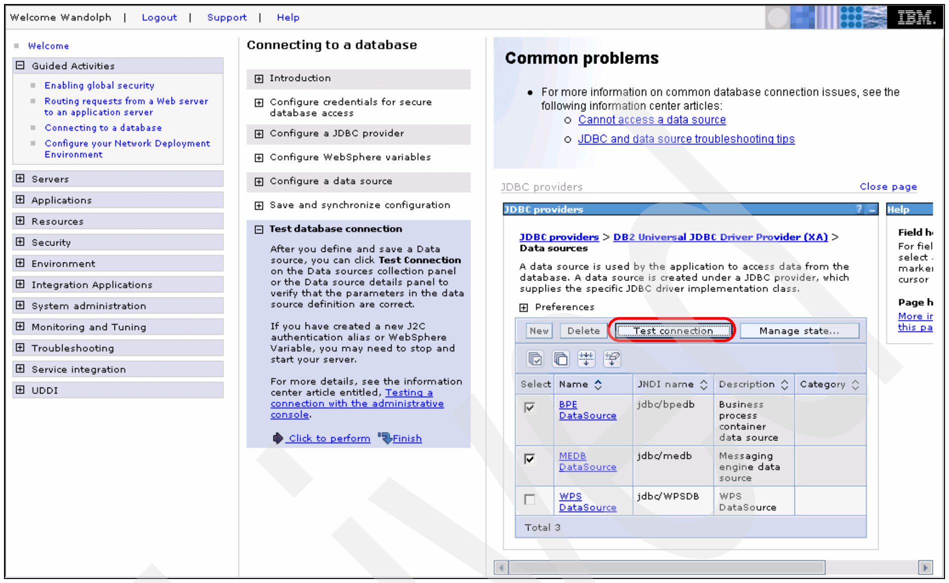
Testing database connectivity

Note: Because the data sources are reusing the existing authentication alias, `WPSDBAlias`, there is no need to restart the environment. You can test these data sources immediately.

Ensure that the `MEDB` and `BPEDB` databases are running and test the newly created data sources:

1. Select **Click to perform**.
2. Click the JDBC provider that is defined at the cell scope **DB2 Universal JDBC Driver Provider (XA)**.
3. Scroll to the right and select **Data sources**.

4. Select the check box next to the new data sources as shown in Figure 9-25 and click **Test connection**.



Connecting to a database

- Introduction
- Configure credentials for secure database access
- Configure a JDBC provider
- Configure WebSphere variables
- Configure a data source
- Save and synchronize configuration
- **Test database connection**

After you define and save a Data source, you can click **Test Connection** on the Data sources collection panel or the Data source details panel to verify that the parameters in the data source definition are correct.

If you have created a new J2C authentication alias or WebSphere Variable, you may need to stop and start your server.

For more details, see the information center article entitled, Testing a connection with the administrative console.

Click to perform Finish

Common problems

- For more information on common database connection issues, see the following information center articles:
  - Cannot access a data source
  - JDBC and data source troubleshooting tips

JDBC providers Close page

JDBC providers

JDBC providers > **DB2 Universal JDBC Driver Provider (XA)** > **Data sources**

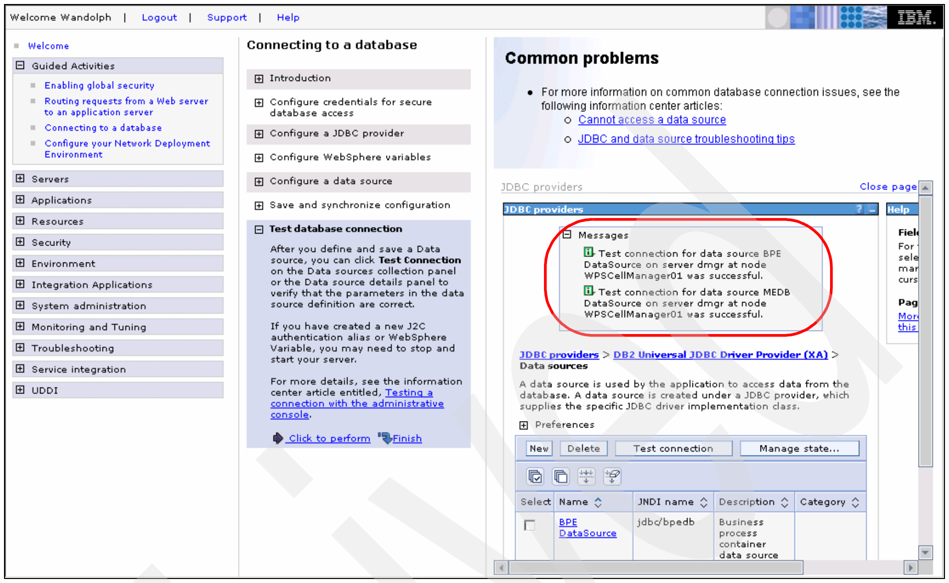
A data source is used by the application to access data from the database. A data source is created under a JDBC provider, which supplies the specific JDBC driver implementation class.

Preferences

Select	Name	JNDI name	Description	Category
✓	BPE DataSource	jdbc/bpedb	Business process container data source	
✓	MEDB DataSource	jdbc/medb	Messaging engine data source	
	WPS DataSource	jdbc/WPSDB	WPS DataSource	
Total 3				

Figure 9-25 Testing the new data source

5. The Messages dialog box (at the top in Figure 9-26) should indicate that the data source connected successfully to the database.



Welcome Wandolph | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Connecting to a database

- Introduction
- Configure credentials for secure database access
- Configure a JDBC provider
- Configure WebSphere variables
- Configure a data source
- Save and synchronize configuration
- **Test database connection**

After you define and save a Data source, you can click **Test Connection** on the Data sources collection panel or the Data source details panel to verify that the parameters in the data source definition are correct.

If you have created a new J2C authentication alias or WebSphere Variable, you may need to stop and start your server.

For more details, see the information center article entitled, Testing a connection with the administrative console.

Click to perform Finish

Common problems

- For more information on common database connection issues, see the following information center articles:
  - Cannot access a data source
  - JDBC and data source troubleshooting tips

JDBC providers Close page

JDBC providers

Messages

Test connection for data source BPE DataSource on server dmgr at node WPSCellManager01 was successful.

Test connection for data source MEDB DataSource on server dmgr at node WPSCellManager01 was successful.

JDBC providers > DB2 Universal JDBC Driver Provider (XA) > **Data sources**

A data source is used by the application to access data from the database. A data source is created under a JDBC provider, which supplies the specific JDBC driver implementation class.

Preferences

New | Delete | Test connection | Manage state...

Select	Name	JNDI name	Description	Category
☐	BPE DataSource	jdbc/bpedb	Business process container data source	

Figure 9-26 Successful database connections

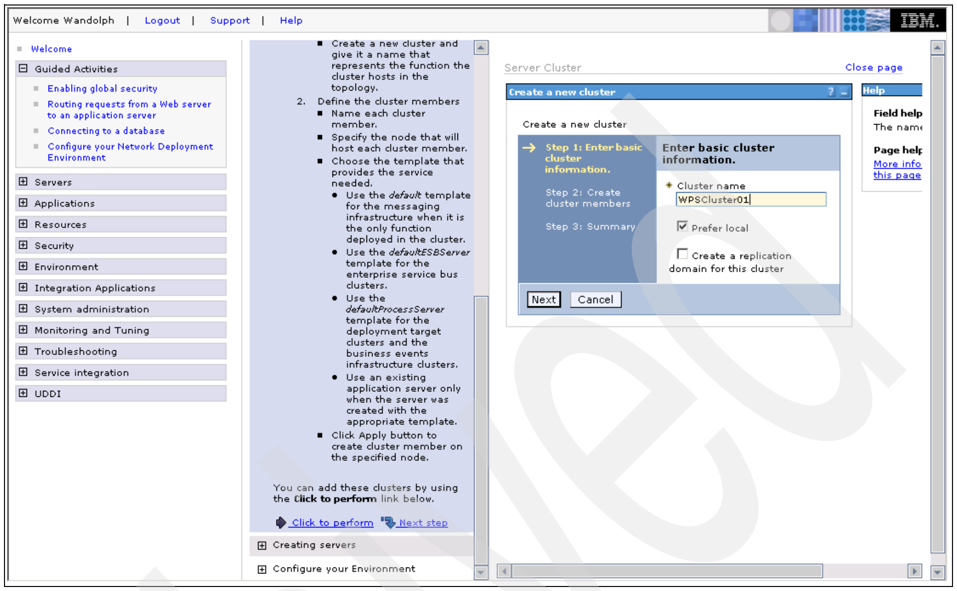
6. Select **Finish**.

9.3.2 Configuring the cell

To configure the network deployment infrastructure with the options that are required for the Full Support topology. Follow these steps:

1. Open the Administrative Console on `wps0` and login.
2. Navigate to **Guided Activities** → **Configure your Network Deployment Environment** and click Start.
3. Scroll down and select **Click to perform** in the expanded panel to see the list of nodes.
4. The nodes are already federated, so click **Next step**.
5. The database configuration is already complete, so click **Next step**.
6. In the Creating clusters and cluster members panel , select **Click to perform**.

7. Click **New** and enter a Cluster name of `WPSCluster01`, as shown in Figure 9-27. Click **Next**.


Welcome Wandolph | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

- Create a new cluster and give it a name that represents the function the cluster hosts in the topology.

2. Define the cluster members
   - Name each cluster member.
   - Specify the node that will host each cluster member.
   - Choose the template that provides the service needed.
     - Use the *default* template for the messaging infrastructure when it is the only function deployed in the cluster.
     - Use the *defaultESBServer* template for the enterprise service bus clusters.
     - Use the *defaultProcessServer* template for the deployment target clusters and the business events infrastructure clusters.
     - Use an existing application server only when the server was created with the appropriate template.
   - Click Apply button to create cluster member on the specified node.

You can add these clusters by using the **Click to perform** link below.

Click to perform | Next step

- Creating servers
- Configure your Environment

Server Cluster | Close page

Create a new cluster

Create a new cluster

→ Step 1: Enter basic cluster information.

Step 2: Create cluster members

Step 3: Summary

Enter basic cluster information.

* Cluster name: WPSCluster01

☑ Prefer local

☐ Create a replication domain for this cluster

Next | Cancel

Help

Field help

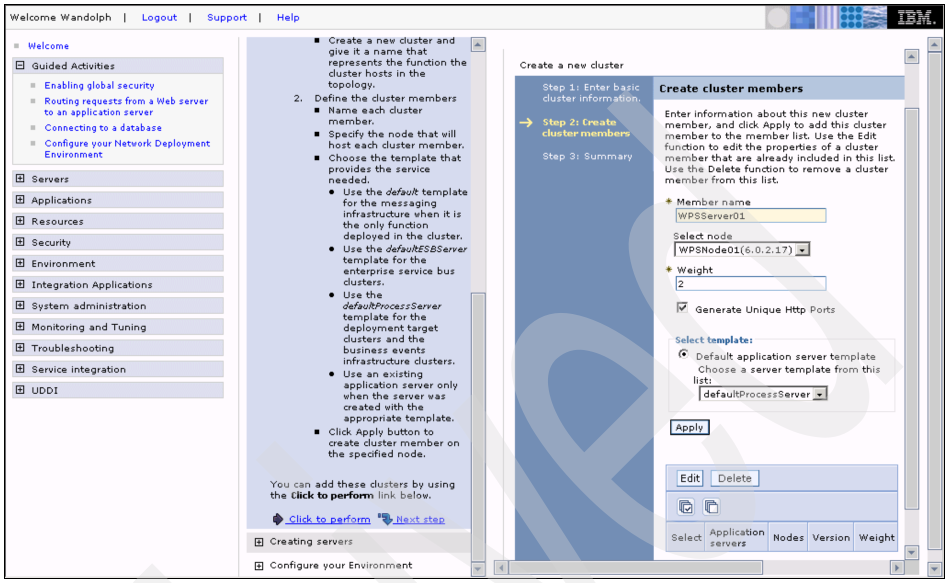
The name

Page help

More info this page

Figure 9-27 Cluster creation

8. Add a first member of the cluster named `WPSServer01` to the node `WPSNode01`. Set the template to `defaultProcessServer`, as shown in Figure 9-28.



Welcome Wandolph | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

- Create a new cluster and give it a name that represents the function the cluster hosts in the topology.

2. Define the cluster members
   - Name each cluster member.
   - Specify the node that will host each cluster member.
   - Choose the template that provides the service needed.
     - Use the *default* template for the messaging infrastructure when it is the only function deployed in the cluster.
     - Use the *defaultESBServer* template for the enterprise service bus clusters.
     - Use the *defaultProcessServer* template for the deployment target clusters and the business events infrastructure clusters.
     - Use an existing application server only when the server was created with the appropriate template.
   - Click Apply button to create cluster member on the specified node.

You can add these clusters by using the **Click to perform** link below.

Click to perform | Next step

- Creating servers
- Configure your Environment

Create a new cluster

Step 1: Enter basic cluster information.
→ Step 2: Create cluster members
Step 3: Summary

Create cluster members

Enter information about this new cluster member, and click Apply to add this cluster member to the member list. Use the Edit function to edit the properties of a cluster member that are already included in this list. Use the Delete function to remove a cluster member from this list.

* Member name: WPSServer01

Select node: WPSNode01(6.0.2.17)

* Weight: 2

[✓] Generate Unique Http Ports

Select template:
(●) Default application server template
Choose a server template from this list: defaultProcessServer

Apply

Edit | Delete

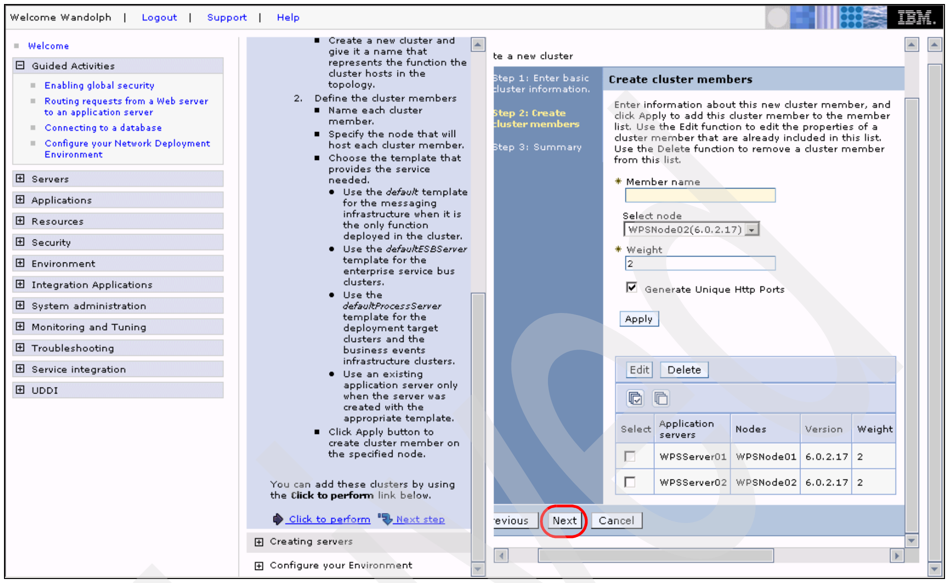
Select	Application servers	Nodes	Version	Weight

Figure 9-28 Adding the first cluster member

9. Click **Apply** to add this server to the new cluster.

10. Repeat this process for `WPSServer02` on `WPSNode02`. However, the template option will not be available now because all cluster members are of the type that is defined by the first member.

11. Click **Apply** to add the second cluster member and then select **Next**, as shown in Figure 9-29.



- Create a new cluster and give it a name that represents the function the cluster hosts in the topology.

2. Define the cluster members
   - Name each cluster member.
   - Specify the node that will host each cluster member.
   - Choose the template that provides the service needed.
     - Use the *default* template for the messaging infrastructure when it is the only function deployed in the cluster.
     - Use the *defaultESBServer* template for the enterprise service bus clusters.
     - Use the *defaultProcessServer* template for the deployment target clusters and the business events infrastructure clusters.
     - Use an existing application server only when the server was created with the appropriate template.
   - Click Apply button to create cluster member on the specified node.

You can add these clusters by using the **Click to perform** link below.

Create cluster members

Enter information about this new cluster member, and click Apply to add this cluster member to the member list. Use the Edit function to edit the properties of a cluster member that are already included in this list. Use the Delete function to remove a cluster member from this list.

* Member name

Select node: WPSNode02(6.0.2.17)

* Weight: 2

[✓] Generate Unique Http Ports

Select	Application servers	Nodes	Version	Weight
☐	WPSServer01	WPSNode01	6.0.2.17	2
☐	WPSServer02	WPSNode02	6.0.2.17	2

Figure 9-29 Cluster members added

12. At the summary page, click **Finish**.

13. Repeat this process for the messaging cluster using the details in Table 9-9.

Table 9-9 Messaging cluster configuration

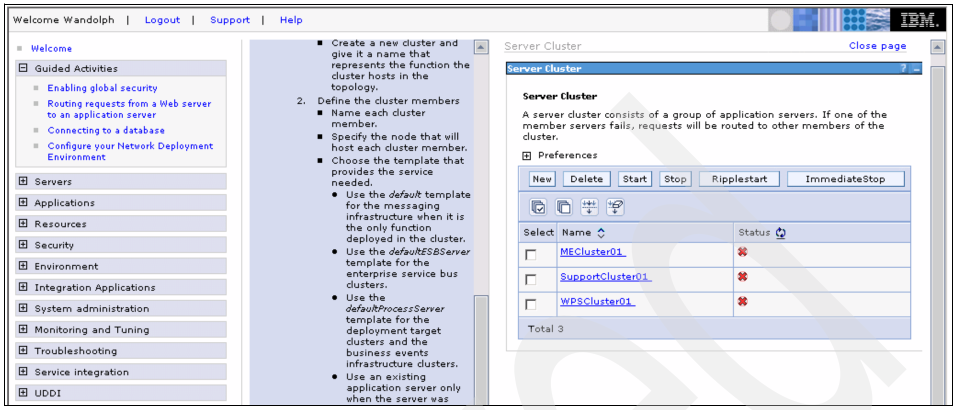
Cluster name	Cluster members	Node name	Template
MECluster01	MEServer01 MEServer02	WPSNode01 WPSNode02	default

14. Repeat this process for the support cluster using the details in Table 9-10.

Table 9-10 Support cluster configuration

Cluster name	Cluster members	Node name	Template
SupportCluster01	SupportServer01 SupportServer02	WPSNode01 WPSNode02	defaultProcessServer

15. Save and synchronize the changes. The three clusters are created as shown in Figure 9-30.


Welcome Wandolph | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

- Create a new cluster and give it a name that represents the function the cluster hosts in the topology.

2. Define the cluster members
   - Name each cluster member.
   - Specify the node that will host each cluster member.
   - Choose the template that provides the service needed.
     - Use the *default* template for the messaging infrastructure when it is the only function deployed in the cluster.
     - Use the *defaultESBServer* template for the enterprise service bus clusters.
     - Use the *defaultProcessServer* template for the deployment target clusters and the business events infrastructure clusters.
     - Use an existing application server only when the server was

Server Cluster   Close page

Server Cluster

Server Cluster

A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster.

⊞ Preferences

New | Delete | Start | Stop | Ripplestart | ImmediateStop

Select	Name	Status
☐	MECluster01	✖
☐	SupportCluster01	✖
☐	WPSCluster01	✖

Total 3

Figure 9-30 Clusters created

16. Click **Next step** to continue and then **Next step** at the Creating servers option because you require no more servers.

The environment now includes three clusters (as illustrated in Figure 9-31). You will create a fourth cluster later for CEI.

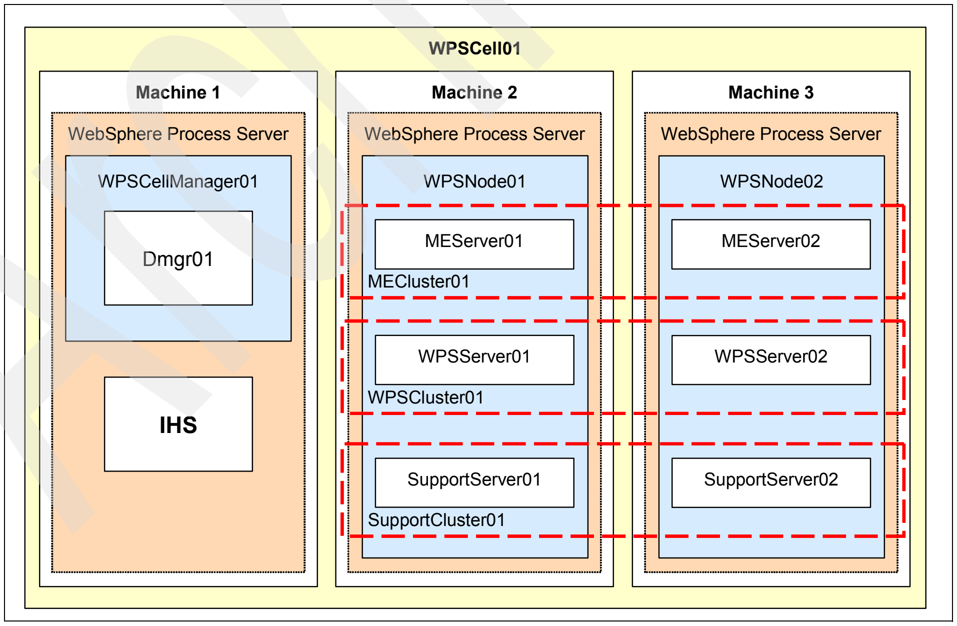


Figure 9-31 Clusters defined

Configuring the environment

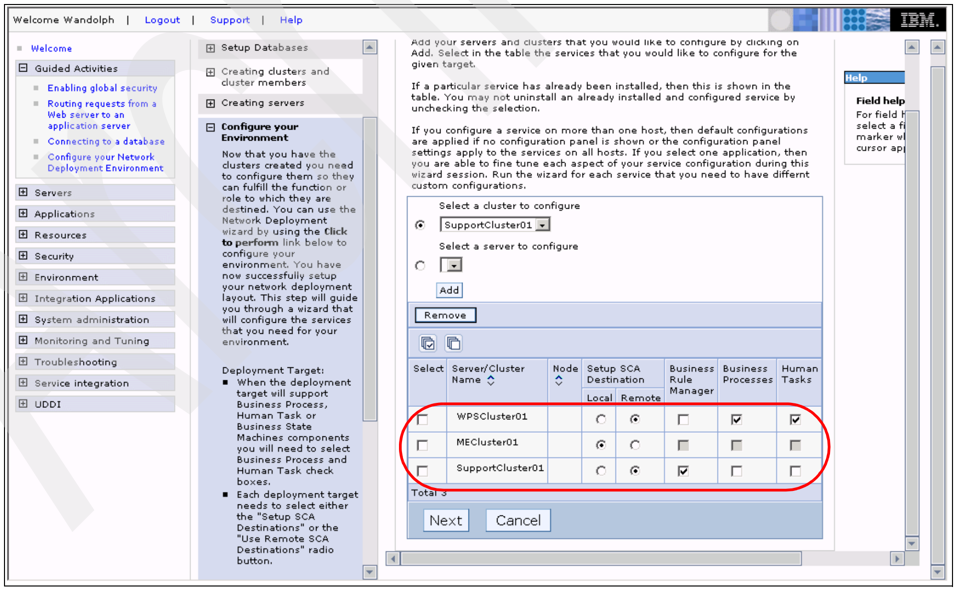
In this section, we describe how to configure SCA, BPC, and HTM for the cluster WPSCluster01.

Configuring the cluster for SCA

To configure the cluster for SCA:

1. In the Configure your Environment panel, select **Click to perform**.
2. Click the pull-down menu to configure a cluster.
3. Select **WPSCluster01** and click **Add**.
4. Select **MECluster01** and click **Add**.
5. Select **SupportCluster01** and click **Add**.
6. Select **Remote** for the SCA Destination for `WPSCluster01`.
7. Select **Local** for the SCA Destination for `MECluster01`.
8. Select **Remote** for the SCA Destination for `SupportCluster01`.
9. Select **Business Processes** and **Human Tasks** for `WPSCluster01`.
10. Check the check boxes under **Business Rule Manager** for `SupportCluster01`.

    Figure 9-32 shows the configuration settings.



Now that you have the clusters created you need to configure them so they can fulfill the function or role to which they are destined. You can use the Network Deployment wizard by using the **Click to perform** link below to configure your environment. You have now successfully setup your network deployment layout. This step will guide you through a wizard that will configure the services that you need for your environment.

Deployment Target:

- When the deployment target will support Business Process, Human Task or Business State Machines components you will need to select Business Process and Human Task check boxes.
- Each deployment target needs to select either the "Setup SCA Destinations" or the "Use Remote SCA Destinations" radio button.

Add your servers and clusters that you would like to configure by clicking on Add. Select in the table the services that you would like to configure for the given target.

If a particular service has already been installed, then this is shown in the table. You may not uninstall an already installed and configured service by unchecking the selection.

If you configure a service on more than one host, then default configurations are applied if no configuration panel is shown or the configuration panel settings apply to the services on all hosts. If you select one application, then you are able to fine tune each aspect of your service configuration during this wizard session. Run the wizard for each service that you need to have diffent custom configurations.

Select a cluster to configure: [✓ SupportCluster01]

Select a server to configure

Add

Remove

Select	Server/Cluster Name	Node	Setup SCA Destination		Business Rule Manager	Business Processes	Human Tasks
			Local	Remote			
☐	WPSCluster01		○	●	☐	☑	☑
☐	MECluster01		●	○	☐	☐	☐
☐	SupportCluster01		○	●	☑	☐	☐

Total 3

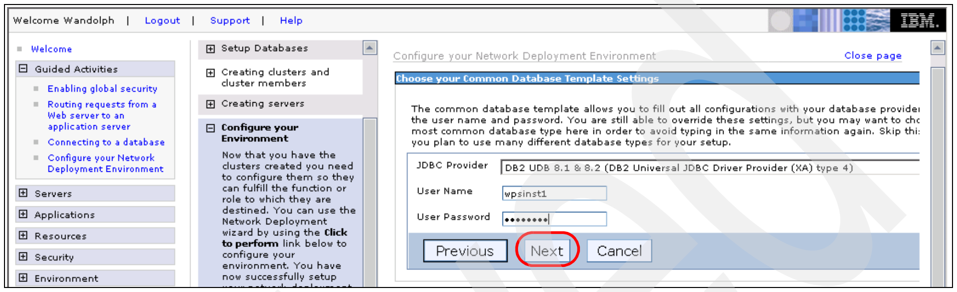
Next Cancel

Figure 9-32 Configuring SCA for each cluster

11. Click **Next**.

12. From the JDBC provider drop-down list, select the appropriate provider. In this example the provider is **DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC Driver Provider (XA) type 4)**.

13. Set the user name to `wpsinst1` and the user password to `itso4you`, and click **Next**, as shown in Figure 9-33.


Welcome Wandolph | Logout | Support | Help
IBM.
Welcome
Guided Activities
Enabling global security
Routing requests from a Web server to an application server
Connecting to a database
Configure your Network Deployment Environment
Servers
Applications
Resources
Security
Environment
Setup Databases
Creating clusters and cluster members
Creating servers
Configure your Environment
Configure your Network Deployment Environment
Close page
Previous
Next
Cancel

Now that you have the clusters created you need to configure them so they can fulfill the function or role to which they are destined. You can use the Network Deployment wizard by using the **Click to perform** link below to configure your environment. You have now successfully setup

Choose your Common Database Template Settings

The common database template allows you to fill out all configurations with your database provide the user name and password. You are still able to override these settings, but you may want to chc most common database type here in order to avoid typing in the same information again. Skip thi: you plan to use many different database types for your setup.

JDBC Provider: DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC Driver Provider (XA) type 4)
User Name: wpsinst1
User Password: ••••••••

Figure 9-33 JDBC provider selection

14. Under System Bus, select **Use existing data source**. In the drop-down menu, select the data source **medb**.

15. Set the schema name to `SCASYS`.

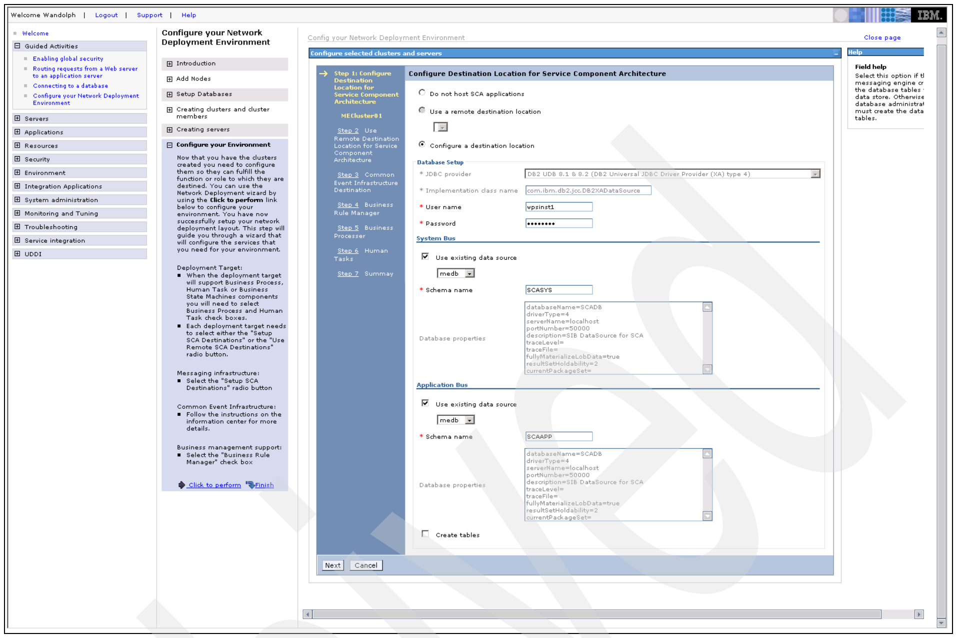
16. Under Application Bus, select **Use existing data source**. In the drop-down menu, select the data source **medb**.

17. Set the schema name to `SCAAPP`.

18. Clear the **Create tables** check box because you created tables manually with the **`sibDDLgenerator.bat`** command.

> **Note:** If the database is DB2, this option creates the SCA database schemas the first time that you start the SCA messaging engines.

Figure 9-34 shows these settings.


Welcome Wandolph | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Configure your Network Deployment Environment

- Introduction
- Add Nodes
- Setup Databases
- Creating clusters and cluster members
- Creating servers
- **Configure your Environment**

Now that you have the clusters created you need to configure them so they can fulfill the function or role to which they are destined. You can use the Network Deployment wizard by using the **Click to perform** link below to configure your environment. You have now successfully setup your network deployment layout. This step will guide you through a wizard that will configure the services that you need for your environment.

Deployment Target:
- When the deployment target will support Business Process, Human Task or Business State Machines components you will need to select Business Process and Human Task check boxes.
- Each deployment target needs to select either the "Setup SCA Destinations" or the "Use Remote SCA Destinations" radio button.

Messaging infrastructure:
- Select the "Setup SCA Destinations" radio button

Common Event Infrastructure:
- Follow the instructions on the information center for more details.

Business management support:
- Select the "Business Rule Manager" check box

Click to perform Finish

Config your Network Deployment Environment Close page

Configure selected clusters and servers

→ **Step 1: Configure Destination Location for Service Component Architecture**

MECluster01

Step 2 Use Remote Destination Location for Service Component Architecture

Step 3 Common Event Infrastructure Destination

Step 4 Business Rule Manager

Step 5 Business Processer

Step 6 Human Tasks

Step 7 Summay

Configure Destination Location for Service Component Architecture

○ Do not host SCA applications

○ Use a remote destination location

● Configure a destination location

Database Setup

* JDBC provider	DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC Driver Provider (XA) type 4)
* Implementation class name	com.ibm.db2.jcc.DB2XADataSource
* User name	wpsinst1
* Password	••••••••

System Bus

☑ Use existing data source

medb

* Schema name	SCASYS

Database properties

```
databaseName=SCADB
driverType=4
serverName=localhost
portNumber=50000
description=SIB DataSource for SCA
traceLevel=
traceFile=
fullyMaterializeLobData=true
resultSetHoldability=2
currentPackageSet=
```

Application Bus

☑ Use existing data source

medb

* Schema name	SCAAPP

Database properties

```
databaseName=SCADB
driverType=4
serverName=localhost
portNumber=50000
description=SIB DataSource for SCA
traceLevel=
traceFile=
fullyMaterializeLobData=true
resultSetHoldability=2
currentPackageSet=
```

☐ Create tables

Next Cancel

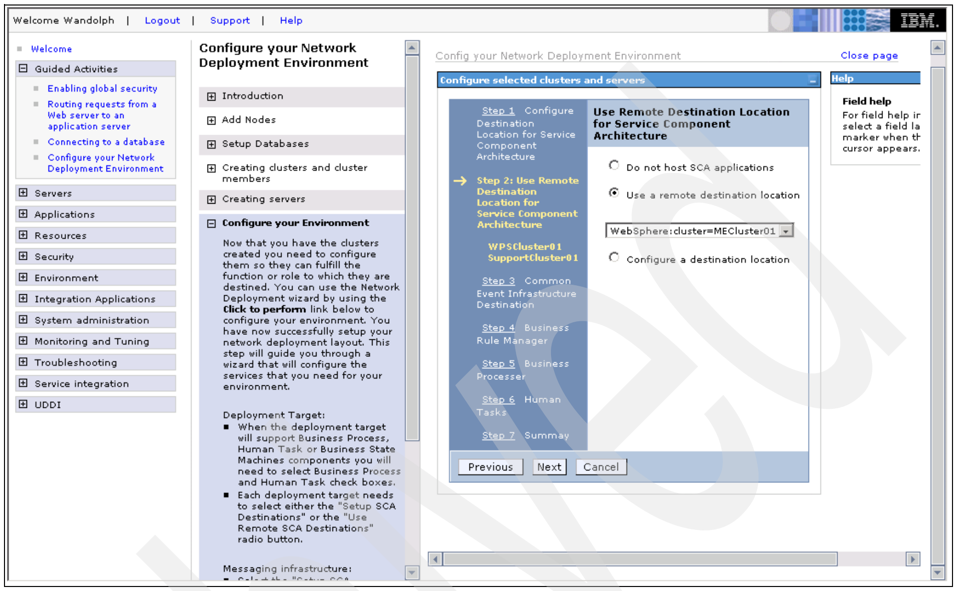
Help

Field help
Select this option if tl messaging engine cr the database tables data store. Otherwise database administrat must create the data tables.

Figure 9-34 SCA Values

19. Click **Next**.

20. This panel configures the WPS and Support clusters to use a remote destination for SCA. The default should set `MECluster01` as the remote location (Figure 9-35).



Configure your Network Deployment Environment

Now that you have the clusters created you need to configure them so they can fulfill the function or role to which they are destined. You can use the Network Deployment wizard by using the **Click to perform** link below to configure your environment. You have now successfully setup your network deployment layout. This step will guide you through a wizard that will configure the services that you need for your environment.

Deployment Target:
- When the deployment target will support Business Process, Human Task or Business State Machines components you will need to select Business Process and Human Task check boxes.
- Each deployment target needs to select either the "Setup SCA Destinations" or the "Use Remote SCA Destinations" radio button.

Messaging infrastructure:

Config your Network Deployment Environment

Configure selected clusters and servers

Step 1 Configure Destination Location for Service Component Architecture
→ **Step 2: Use Remote Destination Location for Service Component Architecture**
WPSCluster01
SupportCluster01
Step 3 Common Event Infrastructure Destination
Step 4 Business Rule Manager
Step 5 Business Processer
Step 6 Human Tasks
Step 7 Summay

Use Remote Destination Location for Service Component Architecture

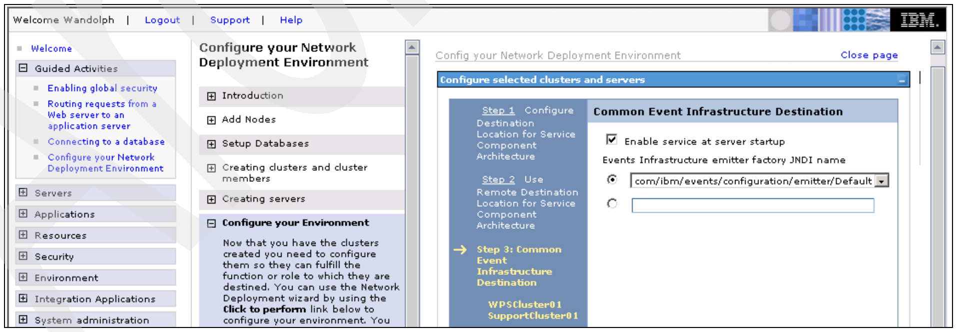
( ) Do not host SCA applications
(●) Use a remote destination location
WebSphere:cluster=MECluster01
( ) Configure a destination location

Field help
For field help ir select a field la marker when th cursor appears.

Figure 9-35 Remote SCA configuration

21. Click **Next**.

22. Ensure that **Enable service at server startup** is selected, leave the other default values, and click **Next** as shown in Figure 9-36.



Configure your Network Deployment Environment

Now that you have the clusters created you need to configure them so they can fulfill the function or role to which they are destined. You can use the Network Deployment wizard by using the **Click to perform** link below to configure your environment. You

Config your Network Deployment Environment

Configure selected clusters and servers

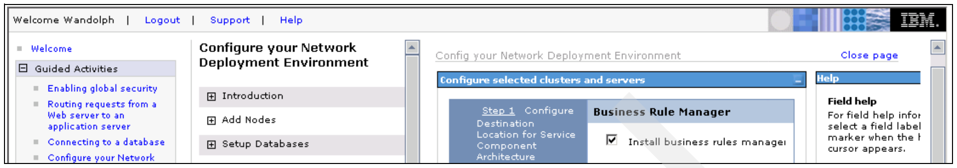
Step 1 Configure Destination Location for Service Component Architecture
Step 2 Use Remote Destination Location for Service Component Architecture
→ **Step 3: Common Event Infrastructure Destination**
WPSCluster01
SupportCluster01

Common Event Infrastructure Destination

[✓] Enable service at server startup
Events Infrastructure emitter factory JNDI name
(●) com/ibm/events/configuration/emitter/Default
( )

Figure 9-36 Startup CEI service settings

23. In the Business Rule Manager panel, ensure that **Install business rules manager** is selected, as shown in Figure 9-37.


Welcome Wandolph | Logout | Support | Help
IBM.
Welcome
Guided Activities
Enabling global security
Routing requests from a Web server to an application server
Connecting to a database
Configure your Network
Configure your Network Deployment Environment
Introduction
Add Nodes
Setup Databases
Config your Network Deployment Environment
Close page
Configure selected clusters and servers
Step 1 Configure Destination Location for Service Component Architecture
Business Rule Manager
Help
Field help

[✓] Install business rules manager

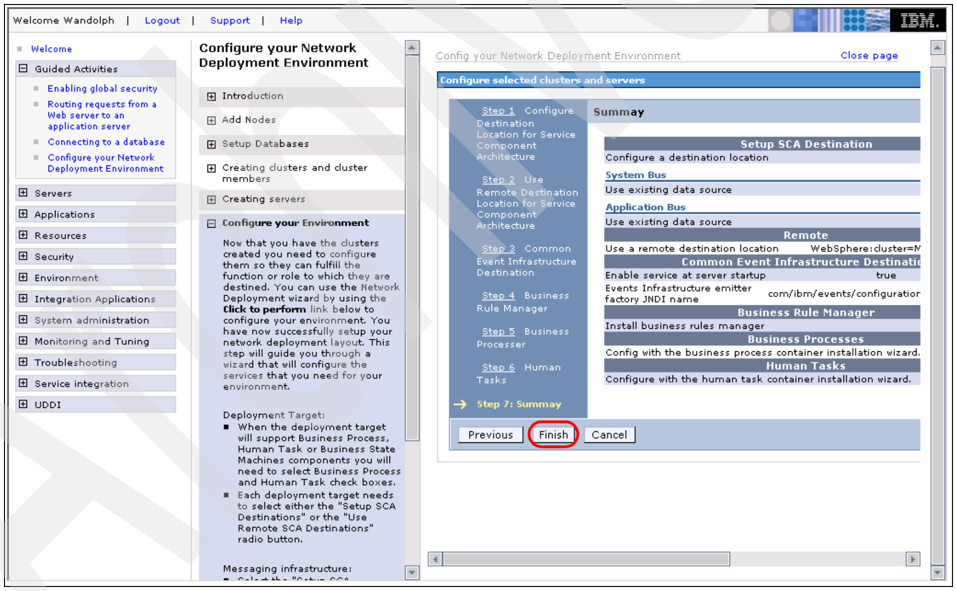
For field help infor
select a field label
marker when the h
cursor appears.

Figure 9-37   Business Rule Manager selected

24. Click **Next**, and in the Business Processer (sic) panel, select **Configure with the business process container wizard installation**, because the wizard exposes more options than this panel.

25. Click **Next**, and in the Human Tasks panel, select **Configure with the human task container wizard installation**, because the wizard exposes more options than this panel.

26. Click **Next**, and in the summary panel, click Finish as shown in Figure 9-38.


Welcome Wandolph | Logout | Support | Help
IBM.
Welcome
Guided Activities
Enabling global security
Routing requests from a Web server to an application server
Connecting to a database
Configure your Network Deployment Environment
Servers
Applications
Resources
Security
Environment
Integration Applications
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Configure your Network Deployment Environment
Introduction
Add Nodes
Setup Databases
Creating clusters and cluster members
Creating servers
Configure your Environment
Config your Network Deployment Environment
Close page
Configure selected clusters and servers
Step 1 Configure Destination Location for Service Component Architecture
Step 2 Use Remote Destination Location for Service Component Architecture
Step 3 Common Event Infrastructure Destination
Step 4 Business Rule Manager
Step 5 Business Processer
Step 6 Human Tasks
Step 7: Summay
Summay
Previous
Finish
Cancel

Now that you have the clusters created you need to configure them so they can fulfill the function or role to which they are destined. You can use the Network Deployment wizard by using the **Click to perform** link below to configure your environment. You have now successfully setup your network deployment layout. This step will guide you through a wizard that will configure the services that you need for your environment.

Deployment Target:

- When the deployment target will support Business Process, Human Task or Business State Machines components you will need to select Business Process and Human Task check boxes.
- Each deployment target needs to select either the "Setup SCA Destinations" or the "Use Remote SCA Destinations" radio button.

Messaging infrastructure:

Setup SCA Destination	
Configure a destination location	
System Bus	
Use existing data source	
Application Bus	
Use existing data source	
Remote	
Use a remote destination location	WebSphere:cluster=M
Common Event Infrastructure Destinatio	
Enable service at server startup	true
Events Infrastructure emitter factory JNDI name	com/ibm/events/configuratior
Business Rule Manager	
Install business rules manager	
Business Processes	
Config with the business process container installation wizard.	
Human Tasks	
Configure with the human task container installation wizard.	

Figure 9-38   Cluster configuration summary

27. The network deployment environment is now configured. When completed, click **Save to Master Configuration**.

28. Save and synchronize the changes, and click **OK** when complete.

Configuring the cluster for BPC

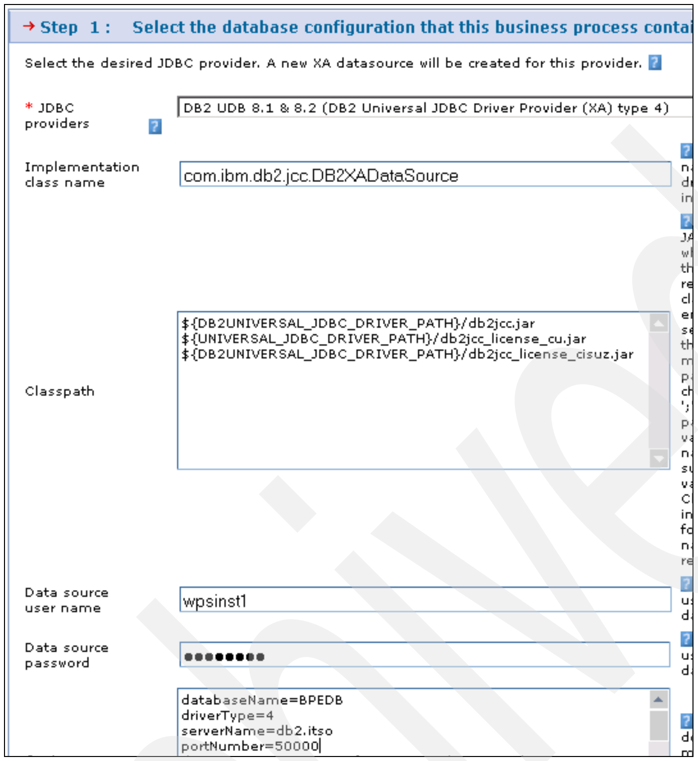
To confirm the cluster for BPC, follow these steps:

1. The next panel allows configuration of the Business Process and Human Task Containers. Select **WPSCluster01** under the Business Process Container Installation heading to start the wizard.
2. Set the values as shown in table Table 9-11.

Table 9-11 DB2 settings for BPC wizard

Name	Value
JDBC providers	DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC Driver Provider (XA) type 4)
Data source user name	wpsinst1
Data source password	itso4you
server Name	db2.itso

Figure 9-39 shows this configuration.


→ Step 1 : Select the database configuration that this business process contai

Select the desired JDBC provider. A new XA datasource will be created for this provider.

* JDBC providers	DB2 UDB 8.1 & 8.2 (DB2 Universal JDBC Driver Provider (XA) type 4)
Implementation class name	com.ibm.db2.jcc.DB2XADataSource
Classpath	${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar ${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar ${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar
Data source user name	wpsinst1
Data source password	●●●●●●●●

	databaseName=BPEDB
	driverType=4
	serverName=db2.itso
	portNumber=50000

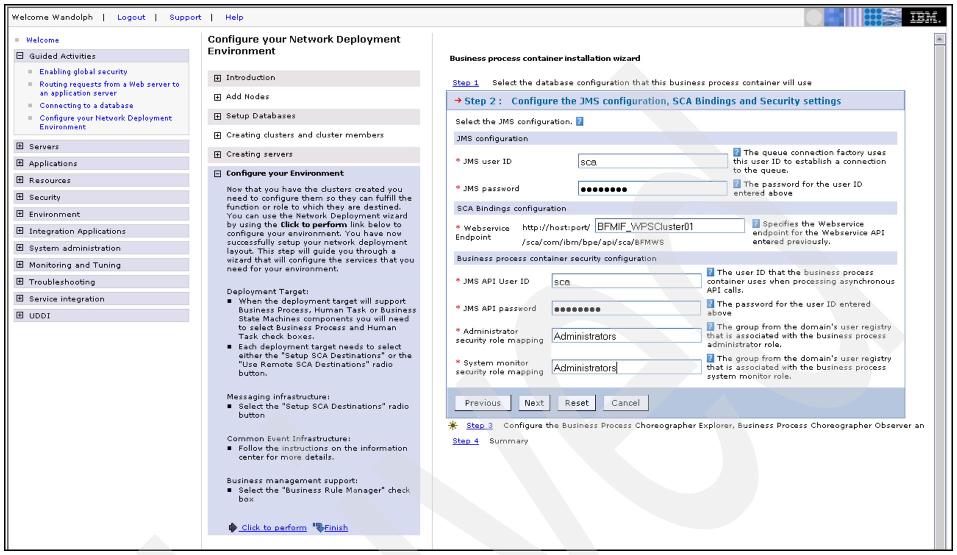
Figure 9-39 BPC database configuration

3. Click **Next**.
4. Set the JMS configuration as detailed in Table 9-12.

Table 9-12 JMS Configuration settings

Name	Value
JMS user ID	sca
JMS password	itso4you
JMS API User ID	sca
JMS API password	itso4you
Administrator security role mapping	Administrators
System monitor security role mapping	Administrators

The `Administrators` group and the `sca` user must exist in the LDAP registry which will be configure in a later step. Figure 9-40 shows the JMS configuration.


Welcome Wandolph | Logout | Support | Help

- Welcome
- Guided Activities
  - Enabling global security
  - Routing requests from a Web server to an application server
  - Connecting to a database
  - Configure your Network Deployment Environment
- Servers
- Applications
- Resources
- Security
- Environment
- Integration Applications
- System administration
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Configure your Network Deployment Environment

- Introduction
- Add Nodes
- Setup Databases
- Creating clusters and cluster members
- Creating servers
- **Configure your Environment**

Now that you have the clusters created you need to configure them so they can fulfill the function or role to which they are destined. You can use the Network Deployment wizard by using the **Click to perform** link below to configure your environment. You have now successfully setup your network deployment layout. This step will guide you through a wizard that will configure the services that you need for your environment.

Deployment Target:
- When the deployment target will support Business Process, Human Task or Business State Machines components you will need to select Business Process and Human Task check boxes.
- Each deployment target needs to select either the "Setup SCA Destinations" or the "Use Remote SCA Destinations" radio button.

Messaging infrastructure:
- Select the "Setup SCA Destinations" radio button

Common Event Infrastructure:
- Follow the instructions on the information center for more details.

Business management support:
- Select the "Business Rule Manager" check box

Click to perform   Finish

Business process container installation wizard

Step 1   Select the database configuration that this business process container will use

→ **Step 2 : Configure the JMS configuration, SCA Bindings and Security settings**

Select the JMS configuration.

JMS configuration

* JMS user ID	sca	The queue connection factory uses this user ID to establish a connection to the queue.
* JMS password	●●●●●●●●	The password for the user ID entered above

SCA Bindings configuration

* Webservice Endpoint	http://host:port/ BFMIF_WPSCluster01 /sca/com/ibm/bpe/api/sca/BFMWS	Specifies the Webservice endpoint for the Webservice API entered previously.

Business process container security configuration

* JMS API User ID	sca	The user ID that the business process container uses when processing asynchronous API calls.
* JMS API password	●●●●●●●●	The password for the user ID entered above
* Administrator security role mapping	Administrators	The group from the domain's user registry that is associated with the business process administrator role.
* System monitor security role mapping	Administrators	The group from the domain's user registry that is associated with the business process system monitor role.

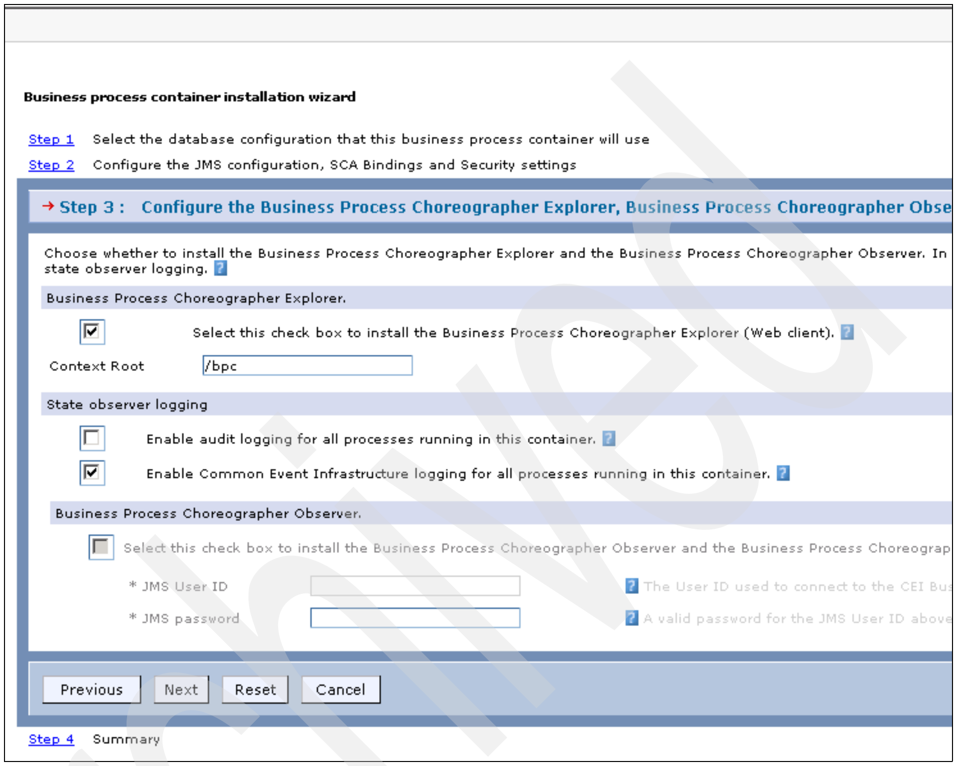
Step 3   Configure the Business Process Choreographer Explorer, Business Process Choreographer Observer an

Step 4   Summary

Figure 9-40 JMS configuration settings

5. Click **Next**.

6. In the BPC Explorer configuration, select **Enable Common Event Infrastructure logging** and leave the other values as the defaults, as shown in Figure 9-41.



Business process container installation wizard

Step 1 Select the database configuration that this business process container will use

Step 2 Configure the JMS configuration, SCA Bindings and Security settings

→ Step 3 : Configure the Business Process Choreographer Explorer, Business Process Choreographer Obse

Choose whether to install the Business Process Choreographer Explorer and the Business Process Choreographer Observer. In
state observer logging.

Business Process Choreographer Explorer.

[✓] Select this check box to install the Business Process Choreographer Explorer (Web client).

Context Root: /bpc

State observer logging

[ ] Enable audit logging for all processes running in this container.

[✓] Enable Common Event Infrastructure logging for all processes running in this container.

Business Process Choreographer Observer.

[ ] Select this check box to install the Business Process Choreographer Observer and the Business Process Choreograp

* JMS User ID — The User ID used to connect to the CEI Bus

* JMS password — A valid password for the JMS User ID above

Step 4 Summary

Figure 9-41 BPC Explorer Configuration

7. Click **Next**. In the summary panel click **Finish**.
8. When complete, click **Save to Master Configuration**.
9. Save and synchronize the changes and click **OK** when complete.

Configuring the cluster for the Human Task Manager

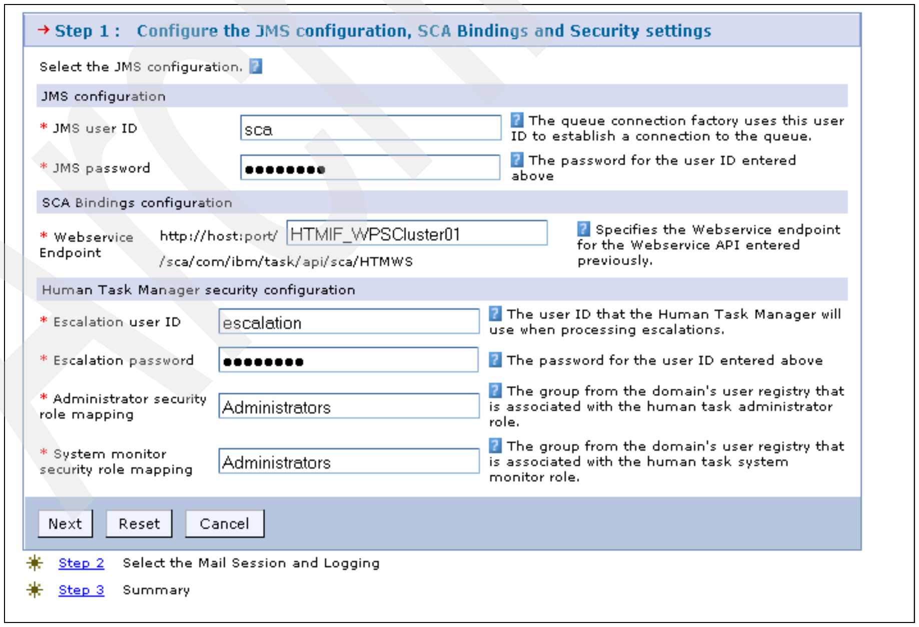
To confirm the cluster for the Human Task Manager, follow these steps:

1. Select the cluster WPSCluster01 under the **Human Task Manager Installation** heading to start the wizard.
2. Set the JMS configuration as detailed in Table 9-13.

Table 9-13 JMS Configuration settings

Name	Value
JMS user ID	sca
JMS password	itso4you
Escalation user ID	escalation
Escalation password	itso4you
Administrator security role mapping	Administrators
System monitor security role mapping	Administrators

The `Administrators` group and the `escalation` user must exist in the LDAP registry which you will configure later. Figure 9-40 shows the JMS configuration.


→ Step 1 : Configure the JMS configuration, SCA Bindings and Security settings

Select the JMS configuration.

JMS configuration

* JMS user ID	sca	The queue connection factory uses this user ID to establish a connection to the queue.
* JMS password	●●●●●●●●	The password for the user ID entered above

SCA Bindings configuration

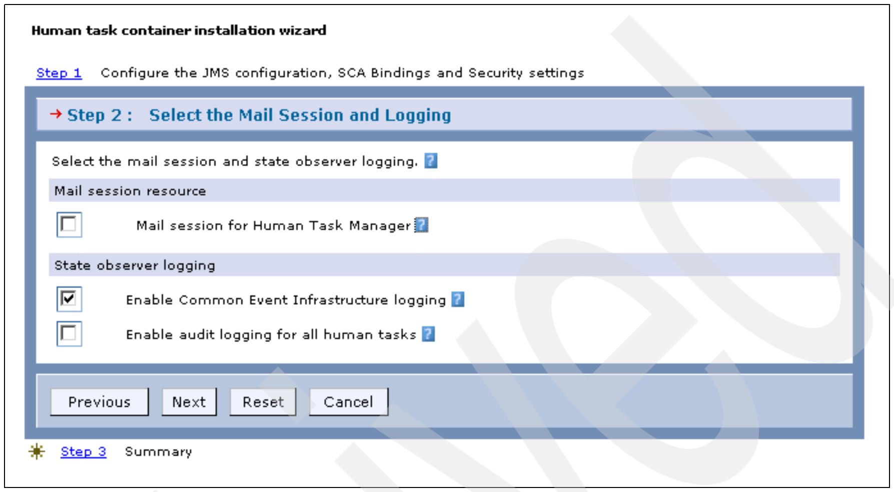
* Webservice Endpoint	http://host:port/ HTMIF_WPSCluster01 /sca/com/ibm/task/api/sca/HTMWS	Specifies the Webservice endpoint for the Webservice API entered previously.

Human Task Manager security configuration

* Escalation user ID	escalation	The user ID that the Human Task Manager will use when processing escalations.
* Escalation password	●●●●●●●●	The password for the user ID entered above
* Administrator security role mapping	Administrators	The group from the domain's user registry that is associated with the human task administrator role.
* System monitor security role mapping	Administrators	The group from the domain's user registry that is associated with the human task system monitor role.

Figure 9-42 JMS configuration settings

3. Click **Next**.
4. In the **Mail Session and Logging** panel configuration, select **Enable Common Event Infrastructure logging** and leave the remaining options clear (Figure 9-43).


Human task container installation wizard

Step 1 Configure the JMS configuration, SCA Bindings and Security settings

→ Step 2 : Select the Mail Session and Logging

Select the mail session and state observer logging.

Mail session resource

☐ Mail session for Human Task Manager

State observer logging

☑ Enable Common Event Infrastructure logging

☐ Enable audit logging for all human tasks

Step 3 Summary

Figure 9-43 BPC Explorer Configuration

5. Click **Next**.
6. In the summary panel click **Finish**.
7. When complete, click **Save to Master Configuration**. Save and synchronize the changes, and click **OK** when complete.
8. Click **Finish** and click **Finish** again.

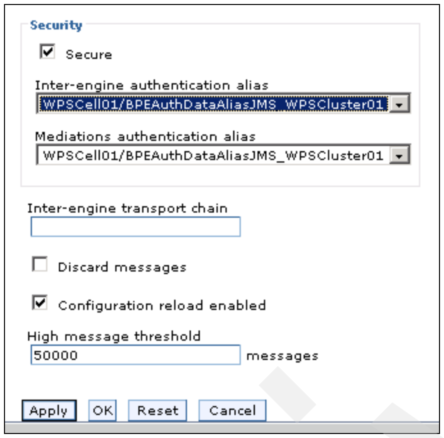
Reconfiguring the BPC Bus and messaging engine

To reconfigure the BPC Bus and messaging engine:

1. Open the `wps0` Administrative Console and log in.
2. Navigate to **Service integration** → **Buses**.
3. Select the **BPC.WPSCell01.Bus bus**.
4. Select WPSCell01/BPEAuthDataAliasJMS_WPSCluster01 from the **Inter-engine authentication alias drop-down list**.

5. Select **WPSCell01/BPEAuthDataAliasJMS_WPSCluster01** from the **Mediations authentication alias** drop-down list, as shown in Figure 9-44.

   The application that we use in this example does not require secure mediations; however, we enable the capability as a best practice.



Security

☑ Secure

Inter-engine authentication alias: WPSCell01/BPEAuthDataAliasJMS_WPSCluster01

Mediations authentication alias: WPSCell01/BPEAuthDataAliasJMS_WPSCluster01

Inter-engine transport chain:

☐ Discard messages

☑ Configuration reload enabled

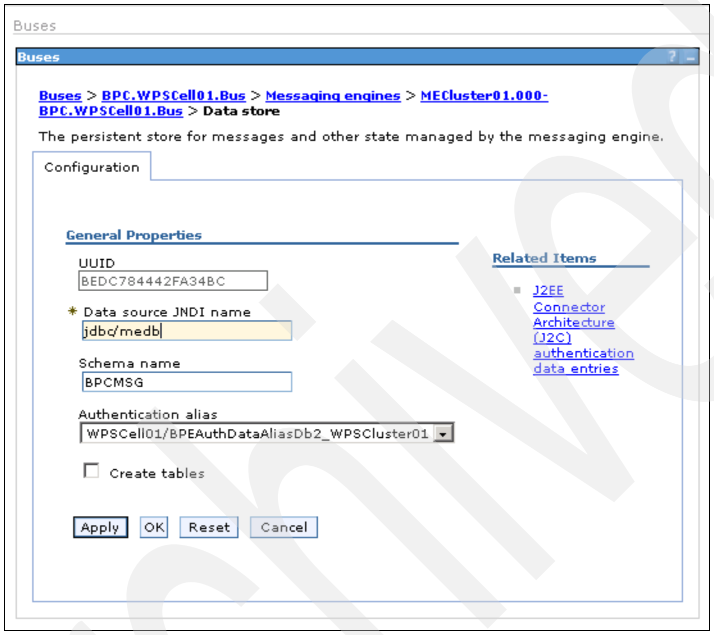
High message threshold: 50000 messages

Figure 9-44 Setting the BPC bus authentication aliases

6. Click **Apply**.
7. Under Topology, click **Messaging engines**.
8. Click the **MECluster01.000-BPC.WPSCell01.Bus** messaging engine.
9. Under Additional Properties, click **Data store**.

10.In the Configuration panel (Figure 9-45):

a. Set the Data source JNDI name to `jdbc/medb`
b. Set the **Schema name** to `BPCMSG`, as created earlier
c. Select **WPSCell01/BPEAuthDataAliasDb2_WPSCluster01** from the **Authentication alias**
d. Clear the **Create tables** check box
e. Click **OK**.


Buses

Buses

Buses > BPC.WPSCell01.Bus > Messaging engines > MECluster01.000-BPC.WPSCell01.Bus > Data store

The persistent store for messages and other state managed by the messaging engine.

Configuration

General Properties

UUID: BEDC784442FA34BC

* Data source JNDI name: jdbc/medb

Schema name: BPCMSG

Authentication alias: WPSCell01/BPEAuthDataAliasDb2_WPSCluster01

☐ Create tables

Apply
OK
Reset
Cancel
Related Items
J2EE Connector Architecture (J2C) authentication data entries

Figure 9-45 Correcting the BPC messaging engine schema

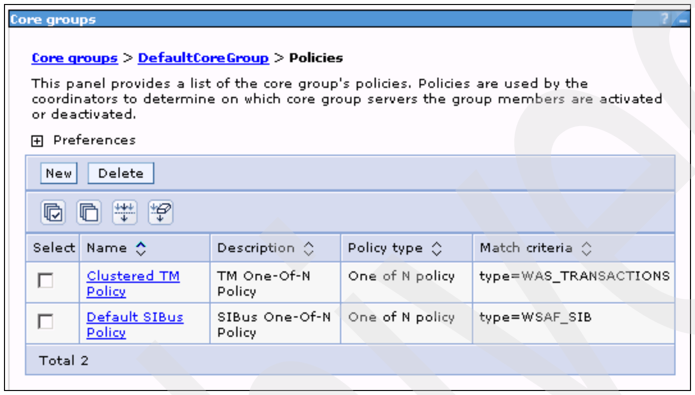
11.Save and synchronize the changes, and click **OK** when complete.

9.3.3 Setting the preferred server for messaging engines

By default, all messaging engines are hosted on the first cluster member in the `MECluster01` cluster that starts. To balance the workload of the messaging engines, you need to modify the HA Manager policy to distribute the load. The messaging engines for `SCAAPP` and `SCASYS` are configured to have the `MEServer01` cluster member as their preferred server, and the messaging engines for `CEIMSG` and `BPCMSG` are configured to have the `MEServer02` cluster member as their preferred server.

To set the preferred server, follow these steps:

1. Open the `wps0` Administrative Console and login.
2. Navigate to **Servers** → **Core groups** → **Core group settings**.
3. Select **DefaultCoreGroup**.
4. Under Additional Properties select **Policies**.
5. A list of existing policies displays (Figure 9-46). Click **New** to define a new policy.


Core groups

Core groups > DefaultCoreGroup > Policies

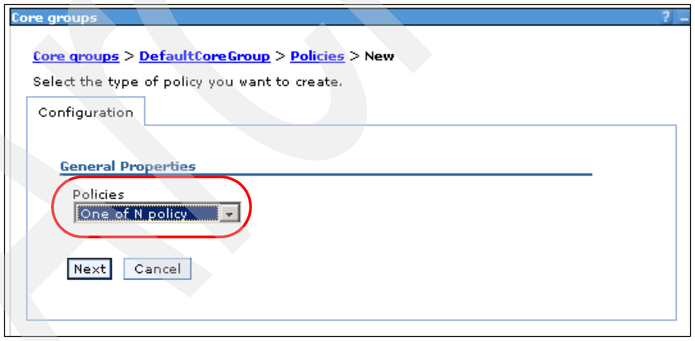
This panel provides a list of the core group's policies. Policies are used by the coordinators to determine on which core group servers the group members are activated or deactivated.

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

Total 2

Figure 9-46 Create a new policy

6. Select **One of N policy** in the Policies pull-down menu**,** then click **Next** (Figure 9-47).


Core groups

Core groups > DefaultCoreGroup > Policies > New

Select the type of policy you want to create.

Policies: One of N policy

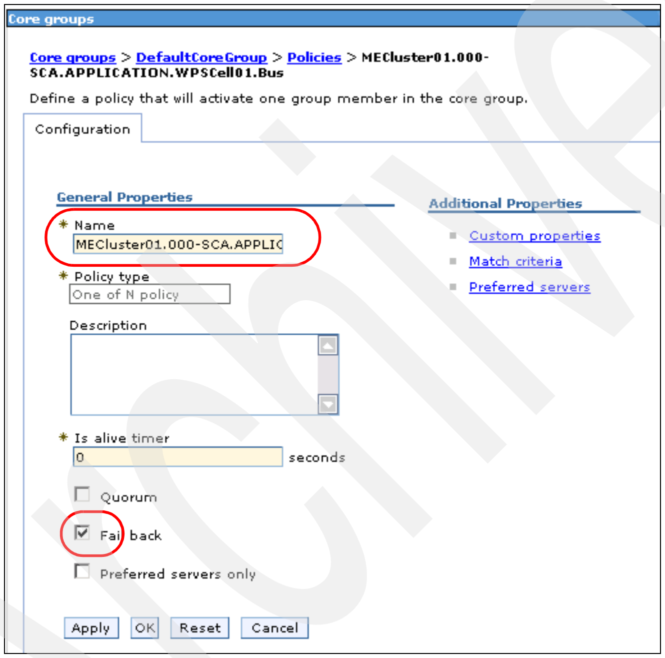
Figure 9-47 Specify the policy type as One of N

7. Specify the configuration information for the policy (Figure 9-48 on page 372).

   a. Enter `MECluster01.000-SCA.APPLICATION.WPSCell01.Bus` in the Name field.

   > **Tip:** We have set the policy name to match the name of the messaging engine to which this policy applies. This helps to identify easily what each policy does.

   b. Select **Fail back** to ensure that the preferred server that is specified by this policy will always have high priority to host the messaging engine that is matched by the policy.


Core groups

Core groups > DefaultCoreGroup > Policies > MECluster01.000-SCA.APPLICATION.WPSCell01.Bus

Define a policy that will activate one group member in the core group.

Configuration

General Properties

* Name: MECluster01.000-SCA.APPLIC

* Policy type: One of N policy

Description:

* Is alive timer: 0 seconds

☐ Quorum

☑ Fail back

☐ Preferred servers only

Additional Properties

- Custom properties
- Match criteria
- Preferred servers

Figure 9-48 Specify the policy configuration information

8. Click **Apply**.

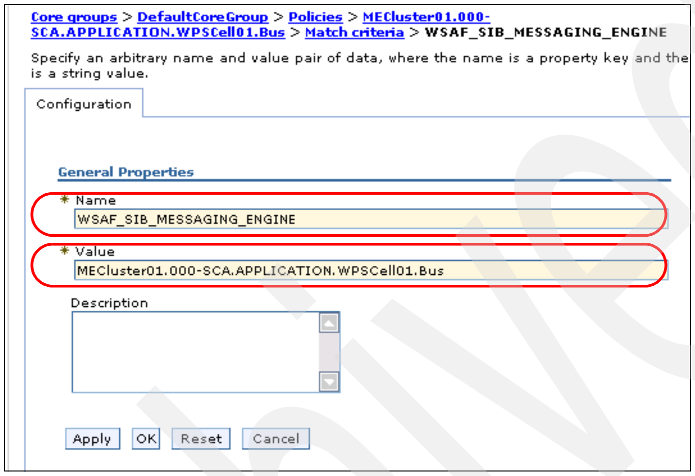
9. Click **Match criteria** under Additional Properties**.** You add two matching criteria.

   a. Click **New**.

   b. Set the Name to `WSAF_SIB_MESSAGING_ENGINE`.

   c. Set the Value to `MECluster01.000-SCA.APPLICATION.WPSCell01.Bus`.

   d. Click **OK** (Figure 9-49).



Core groups > DefaultCoreGroup > Policies > MECluster01.000-SCA.APPLICATION.WPSCell01.Bus > Match criteria > WSAF_SIB_MESSAGING_ENGINE

Specify an arbitrary name and value pair of data, where the name is a property key and the is a string value.

Configuration

General Properties

* Name: WSAF_SIB_MESSAGING_ENGINE

* Value: MECluster01.000-SCA.APPLICATION.WPSCell01.Bus

Description

Figure 9-49 Specify information for the first matching criteria

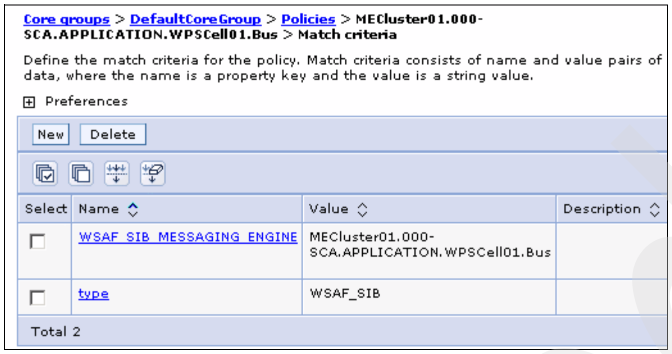
   e. Click **New** to create a second matching criteria.

   f. Set the Name to `type`.

   g. Set the Value to `WSAF_SIB`.

   h. Click **OK**.

10. You need to define two match criteria (Figure 9-50).


Core groups > DefaultCoreGroup > Policies > MECluster01.000-SCA.APPLICATION.WPSCell01.Bus > Match criteria

Define the match criteria for the policy. Match criteria consists of name and value pairs of data, where the name is a property key and the value is a string value.

Select	Name	Value	Description
☐	WSAF_SIB_MESSAGING_ENGINE	MECluster01.000-SCA.APPLICATION.WPSCell01.Bus	
☐	type	WSAF_SIB	

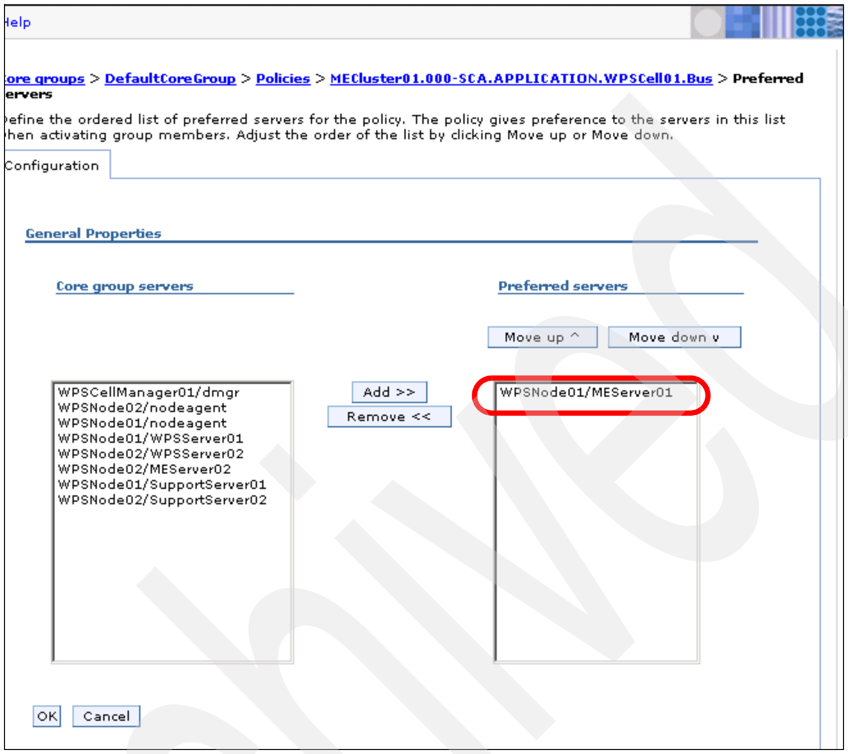
Total 2

Figure 9-50 Match policy for the first SCA application messaging engine

11. From the Policies panel select the **MECluster01.000-SCA.APPLICATION.WPSCell01.Bus** policy.

12. Click **Preferred servers** under **Additional Properties**.

13. Highlight **WPSNode01/MEServer01** and click **Add** to add it to the preferred servers list (Figure 9-51).


ore groups > DefaultCoreGroup > Policies > MECluster01.000-SCA.APPLICATION.WPSCell01.Bus > Preferred ervers

efine the ordered list of preferred servers for the policy. The policy gives preference to the servers in this list hen activating group members. Adjust the order of the list by clicking Move up or Move down.

Core group servers:

```
WPSCellManager01/dmgr
WPSNode02/nodeagent
WPSNode01/nodeagent
WPSNode01/WPSServer01
WPSNode02/WPSServer02
WPSNode02/MEServer02
WPSNode01/SupportServer01
WPSNode02/SupportServer02
```

Preferred servers:

```
WPSNode01/MEServer01
```

Figure 9-51 MEServer01 as the preferred server

14. Click **OK** and save and synchronize the changes.

15. Create another policy for the messaging engine on the `SCA.SYSTEM.WPSCell01.Bus` bus using the values specified in Table 9-14.

Table 9-14 Policy for the SCA system bus

Setting	Value
Policy type	One of N policy
Policy name	MECluster01.000-SCA.SYSTEM.WPSCell01.Bus
Fail back	Yes
Match criteria 1 name	WSAF_SIB_MESSAGING_ENGINE

Setting	Value
Match criteria 1 value	MECluster01.000-SCA.SYSTEM.WPSCell01.Bus
Match criteria 2 name	type
Match criteria 2 value	WSAF_SIB
Preferred server	WPSNode01/MEServer01

You have now created the HA Manager policy for the messaging engine on the `SCA.APPLICATION.WPSCell01.Bus` and `SCA.SYSTEM.WPSCell01.Bus` buses.

16. Create a policy for the messaging engine on the BPC.WPSCell01.Bus bus using the values specified in Table 9-15

Table 9-15 Policy for the BPC messaging engine

Setting	Value
Policy type	One of N policy
Policy name	MECluster01.000-BPC.WPSCell01.Bus
Fail back	Yes
Match criteria 1 name	WSAF_SIB_MESSAGING_ENGINE
Match criteria 1 value	MECluster01.000-BPC.WPSCell01.Bus
Match criteria 2 name	type
Match criteria 2 value	WSAF_SIB
Preferred server	WPSNode02/MEServer02

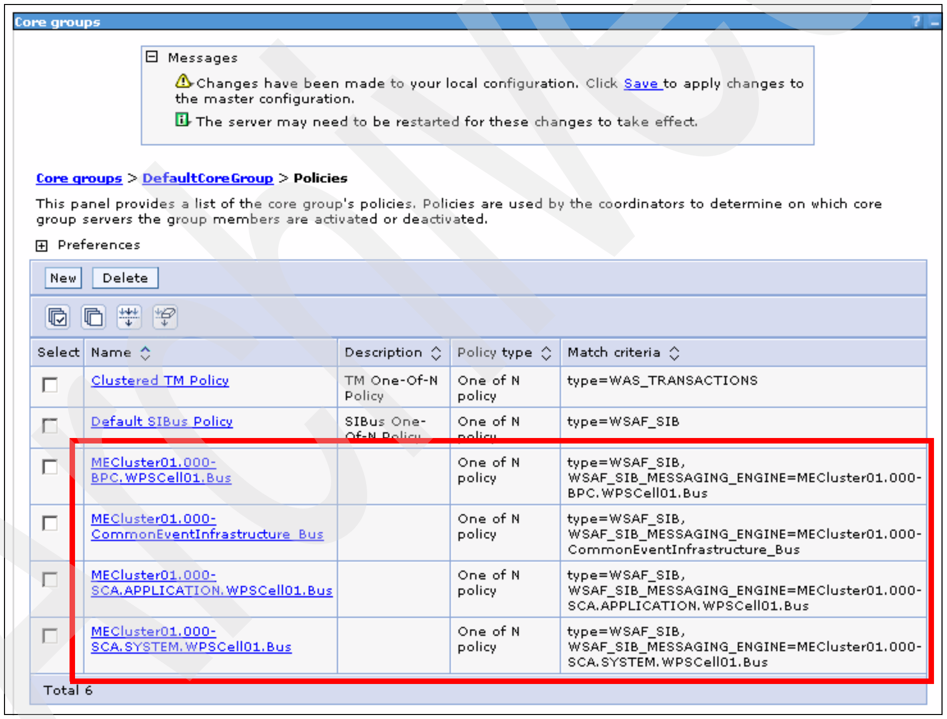
17. Create a policy for the messaging engine on the `CommonEventInfrastructure.WPSCell01.Bus` bus using the values that are specified in Table 9-16.

Table 9-16 Policy for second messaging engine on the BPC bus

Setting	Value
Policy type	One of N policy
Policy name	MECluster01.000-CommonEventInfrastructure.WPSCell01.Bus
Fail back	Yes

Setting	Value
Match criteria 1 name	WSAF_SIB_MESSAGING_ENGINE
Match criteria 1 value	MECluster01.000-CommonEventInfrastructure.WPSCell01.Bus
Match criteria 2 name	type
Match criteria 2 value	WSAF_SIB
Preferred server	WPSNode02/MEServer02

18. At this point, you have defined four HA Manager policies—two for SCA messaging engines and one each for BPC and CEI messaging engines—as shown in Figure 9-52.


Core groups

Messages

Changes have been made to your local configuration. Click Save to apply changes to the master configuration.

The server may need to be restarted for these changes to take effect.

Core groups > DefaultCoreGroup > Policies

This panel provides a list of the core group's policies. Policies are used by the coordinators to determine on which core group servers the group members are activated or deactivated.

Preferences

New | Delete

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
☐	MECluster01.000-BPC.WPSCell01.Bus		One of N policy	type=WSAF_SIB, WSAF_SIB_MESSAGING_ENGINE=MECluster01.000-BPC.WPSCell01.Bus
☐	MECluster01.000-CommonEventInfrastructure_Bus		One of N policy	type=WSAF_SIB, WSAF_SIB_MESSAGING_ENGINE=MECluster01.000-CommonEventInfrastructure_Bus
☐	MECluster01.000-SCA.APPLICATION.WPSCell01.Bus		One of N policy	type=WSAF_SIB, WSAF_SIB_MESSAGING_ENGINE=MECluster01.000-SCA.APPLICATION.WPSCell01.Bus
☐	MECluster01.000-SCA.SYSTEM.WPSCell01.Bus		One of N policy	type=WSAF_SIB, WSAF_SIB_MESSAGING_ENGINE=MECluster01.000-SCA.SYSTEM.WPSCell01.Bus

Total 6

Figure 9-52 Six HA Manager policies

19. Save and synchronize the changes.

9.3.4 CEI configuration for DB2

This section details the creation of the CEI database (`CEIDB`) and the cell configuration that is associated with CEI. In this example, we create the database manually by transferring the creation scripts to the DB2 server.

Generating the CEI database script

You need to generate the scripts to create the CEI database and the associated cell configuration scripts manually for the CEI database. Follow these steps:

1. Open a command prompt on the `wps0`.
2. Change directory to %*WAS_HOME*%\profiles\Dmgr01\event\dbconfig. In this example, the directory is C:\IBM\WebSphere\ProcServer\profiles\Dmgr01\event\dbconfig.
3. Copy the file DB2ResponseFile.txt to NewDB2ResponseFile.txt.
4. Using an appropriate editor, edit NewDB2ResponseFile.txt with the correct information for the environment. Table 9-17 lists the changes for this example.

Table 9-17 NewDB2Responsfile.txt settings

Setting	Value
SCOPE	cell
DB_NAME	CEIDB
DB_NODE_NAME	db2 (look in db2nodes.cfg on the DB2 server)
JDBC_CLASSPATH	c:\db2client
UNIVERSAL_JDBC_CLASSPATH	c:\db2client
JDBC_DRIVER_TYPE	4
DB2_HOST_NAME	db2.itso
DB2_INSTANCE_PORT	50000
EXECUTE_SCRIPTS	NO

5. Save the new configuration file.
6. From the same directory, run the following script to generate the CEI configuration scripts:

```
config_event_database.bat NewDB2ResponseFile.txt
```

The scripts to configure the cell for CEI are created in the directory %*WAS_HOME*%\profiles\Dmgr01\event\dsscripts\db2. In this example, the directory is C:\IBM\WebSphere\ProcServer\profiles\Dmgr01\event\dsscripts\db2.

The C:\IBM\WebSphere\ProcServer\profiles\Dmgr01\event\dbscripts\db2 directory is also generated, and it includes the CEIDB database creation scripts.

Creating the CEI database

This section describes the steps that are necessary to create the DB2 database, `CEIDB`, on the database server. You need to transfer the scripts from a WebSphere Process Server server to the DB2 server because the DB2 windows client is not installed on the members of the cell. It is assumed that the instance is running, in this case the instance name is `wpsinst1`.

To create the CEI database, follow these steps:

1. Transfer the CEIDB creation scripts directory for DB2, %*WAS_HOME*%\profiles\Dmgr01\event\dbscripts\db2, to the database server. In this example, this directory is C:\IBM\WebSphere\ProcServer\profiles\Dmgr01\event\dbscripts\db2.

   Attention: If you transfer these files from Windows to UNIX, there might be return characters (`\r`) at the end of each line. To run the DDL scripts, you must remove these characters. A UNIX utility such as `dos2unix` will remove these characters.

2. Log in to the DB2 server `db2.itso` as the instance owner `wpsinst1`.
3. Run the CEIDB creation script as follows:

```
chmod 755 *
./cr_event_db2.sh
```

Table 9-18 shows the command line responses.

Table 9-18 cr_event_db2.sh responses

Response	Description
1	script is running on the DB2 server
wpsinst1	DB2 user

This series of steps creates the database and appropriate tables.

Configuring the CEI database cell

To configure the cell for connection to the CEIDB database:

1. On the cell deployment manager, where the scripts were generated, open a command line prompt.
2. Run the JDBC configuration script. In this example, we are running on Windows and the database is DB2. Therefore, the following commands configure the cell:

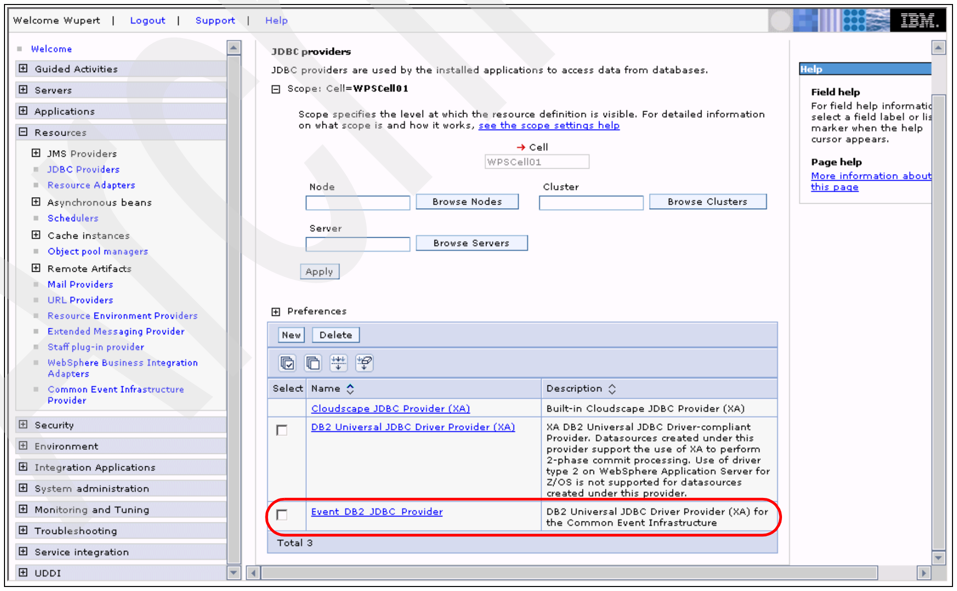
```
cd C:\IBM\WebSphere\ProcServer\profiles\Dmgr01\event\dsscripts\db2
cr_db2_jdbc_provider.bat cell
```

   Table 9-19 shows the command line responses to this script.

Table 9-19 CEI JDBC Provider responses

Response	Description
wpsinst1	DB2 user for CEIDB
itso4you	DB2 password for CEIDB

3. Open the `wps0` Administrative Console, log in, and navigate to **Resources** → **JDBC Providers**. You should see a JDBC provider at the cell scope for CEI called `Event_DB2_JDBC_Provider`, as shown in Figure 9-53.



JDBC providers

JDBC providers are used by the installed applications to access data from databases.

Scope: Cell=WPSCell01

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help

→ Cell: WPSCell01

Node:

Cluster:

Server:

Select	Name	Description
	Cloudscape JDBC Provider (XA)	Built-in Cloudscape JDBC Provider (XA)
☐	DB2 Universal JDBC Driver Provider (XA)	XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on WebSphere Application Server for Z/OS is not supported for datasources created under this provider.
☐	Event DB2 JDBC Provider	DB2 Universal JDBC Driver Provider (XA) for the Common Event Infrastructure

Total 3

Field help

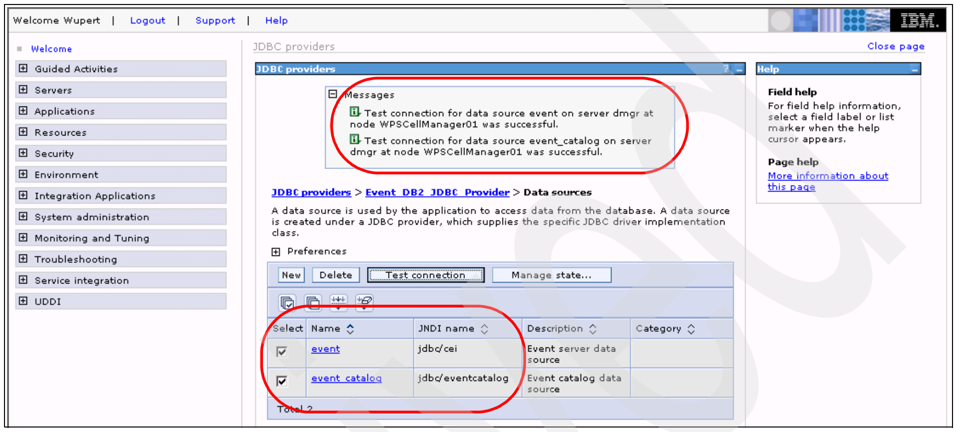
For field help informatio select a field label or lis marker when the help cursor appears.

Page help

More information about this page

Figure 9-53 CEI JDBC provider

4. Click the Event_DB2_JDBC_Provider that is highlighted in Figure 9-53.
5. In the next panel, scroll right and under **Additional Properties** click **Data sources**.
6. From the data sources panel, select the **event** and **event_catalog** data sources and click **Test connection**, as shown in Figure 9-54.



Messages

Test connection for data source event on server dmgr at node WPSCellManager01 was successful.

Test connection for data source event_catalog on server dmgr at node WPSCellManager01 was successful.

JDBC providers > Event_DB2_JDBC_Provider > **Data sources**

A data source is used by the application to access data from the database. A data source is created under a JDBC provider, which supplies the specific JDBC driver implementation class.

Select	Name	JNDI name	Description	Category
✓	event	jdbc/cei	Event server data source	
✓	event_catalog	jdbc/eventcatalog	Event catalog data source	

Total 2

Figure 9-54 Successful database connections

The Messages dialog box (at the top in Figure 9-54) should indicate that the data sources connected successfully to the database.

9.3.5 Deploying the CEI applications

This section explains the steps that are required to install the CEI applications into the `SupportCluster01` cluster.

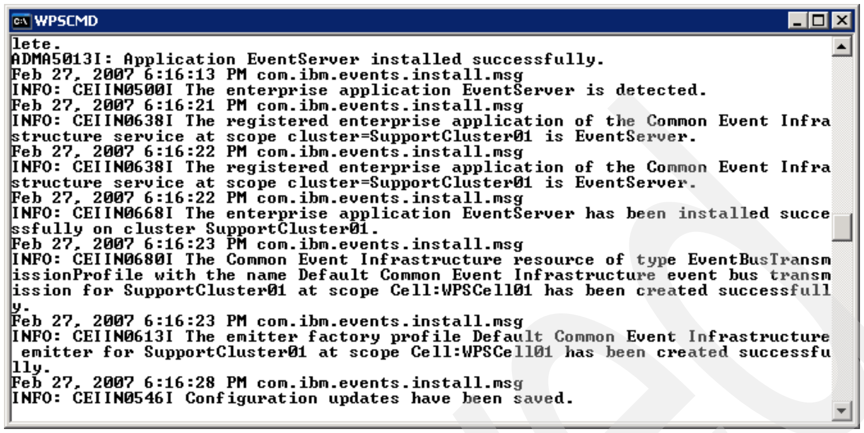
Installing the CEI Event Server application

To install the CEI Event Server application:

1. Open a command prompt on the deployment manager.
2. Change directory to %*WAS_HOME*%\profiles\Dmgr01\event\application. In this example, the directory is C:\IBM\WebSphere\ProcServer\profiles\Dmgr01\event\application.
3. Execute the command to install the CEI Event Server application. For example:

```
..\..\bin\wsadmin -f event-application.jacl -profile
event-profile.jacl -action install -earfile event-application.ear
-backendid DB2UDBNT_V82_1 -cluster SupportCluster01
```

The final line of all the responses should say `Configuration updates have been saved`. Figure 9-55 shows example output.



```
lete.
ADMA5013I: Application EventServer installed successfully.
Feb 27, 2007 6:16:13 PM com.ibm.events.install.msg
INFO: CEIIN0500I The enterprise application EventServer is detected.
Feb 27, 2007 6:16:21 PM com.ibm.events.install.msg
INFO: CEIIN0638I The registered enterprise application of the Common Event Infra
structure service at scope cluster=SupportCluster01 is EventServer.
Feb 27, 2007 6:16:22 PM com.ibm.events.install.msg
INFO: CEIIN0638I The registered enterprise application of the Common Event Infra
structure service at scope cluster=SupportCluster01 is EventServer.
Feb 27, 2007 6:16:22 PM com.ibm.events.install.msg
INFO: CEIIN0668I The enterprise application EventServer has been installed succe
ssfully on cluster SupportCluster01.
Feb 27, 2007 6:16:23 PM com.ibm.events.install.msg
INFO: CEIIN0680I The Common Event Infrastructure resource of type EventBusTransm
issionProfile with the name Default Common Event Infrastructure event bus transm
ission for SupportCluster01 at scope Cell:WPSCell01 has been created successfull
y.
Feb 27, 2007 6:16:23 PM com.ibm.events.install.msg
INFO: CEIIN0613I The emitter factory profile Default Common Event Infrastructure
 emitter for SupportCluster01 at scope Cell:WPSCell01 has been created successfu
lly.
Feb 27, 2007 6:16:28 PM com.ibm.events.install.msg
INFO: CEIIN0546I Configuration updates have been saved.
```

Figure 9-55 CEI application deployment responses

Adding asynchronous capabilities to the CEI Event Server application

To add asynchronous capabilities to the CEI Event Server application:

1. Open a command prompt on the deployment manager.
2. Change directory to %*WAS_HOME*%\profiles\Dmgr01\event\application. In this example, the directory is C:\IBM\WebSphere\ProcServer\profiles\Dmgr01\event\application.
3. Execute the command to install the CEI MDB application. For example:

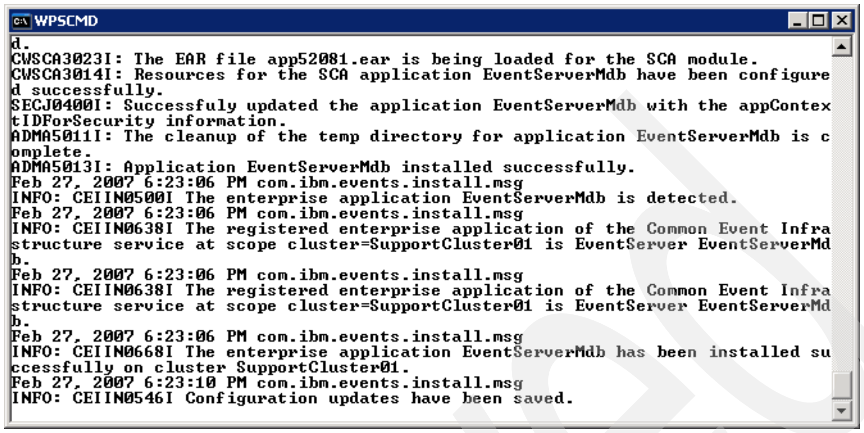
```
..\..\bin\wsadmin -f default-event-message.jacl -profile
event-profile.jacl -earfile event-message.ear -action install
-cluster SupportCluster01
```

Table 9-20 shows the command line responses to this script.

Table 9-20 CEI MDB application responses

Response	Description
sca	Embedded messaging authentication user
itso4you	Embedded messaging authentication password
itso4you	Verified embedded messaging authentication password

The final line of the responses should say `Configuration updates have been saved`. Figure 9-55 shows example output.



```
d.
CWSCA3023I: The EAR file app52081.ear is being loaded for the SCA module.
CWSCA3014I: Resources for the SCA application EventServerMdb have been configure
d successfully.
SECJ0400I: Successfuly updated the application EventServerMdb with the appContex
tIDForSecurity information.
ADMA5011I: The cleanup of the temp directory for application EventServerMdb is c
omplete.
ADMA5013I: Application EventServerMdb installed successfully.
Feb 27, 2007 6:23:06 PM com.ibm.events.install.msg
INFO: CEIIN0500I The enterprise application EventServerMdb is detected.
Feb 27, 2007 6:23:06 PM com.ibm.events.install.msg
INFO: CEIIN0638I The registered enterprise application of the Common Event Infra
structure service at scope cluster=SupportCluster01 is EventServer EventServerMd
b.
Feb 27, 2007 6:23:06 PM com.ibm.events.install.msg
INFO: CEIIN0638I The registered enterprise application of the Common Event Infra
structure service at scope cluster=SupportCluster01 is EventServer EventServerMd
b.
Feb 27, 2007 6:23:06 PM com.ibm.events.install.msg
INFO: CEIIN0668I The enterprise application EventServerMdb has been installed su
ccessfully on cluster SupportCluster01.
Feb 27, 2007 6:23:10 PM com.ibm.events.install.msg
INFO: CEIIN0546I Configuration updates have been saved.
```

Figure 9-56 CEI MEDB application deployment response

Modifying the CEI configuration

The default configuration for CEI assumes local messaging engines. In this topology, the messaging engines are located on the `MECluster01` cluster members.

Deleting bus member SupportCluster01

To delete SupportCluster01:

1. Close any existing browsers to the cell Administrative Console.
2. Open the cell administration console Administrative Console and login.
3. Navigate to **Service integration** → **Buses**.
4. Click CommonEventInfrastructure_Bus.
5. Under the Topology heading, select **Bus members**.

6. `SupportCluster01` is a member of the bus; however, in the case of a remote messaging environment, this member should be `MECluster01`. Select **Cluster=SupportCluster01** and click **Remove** as shown in Figure 9-57.

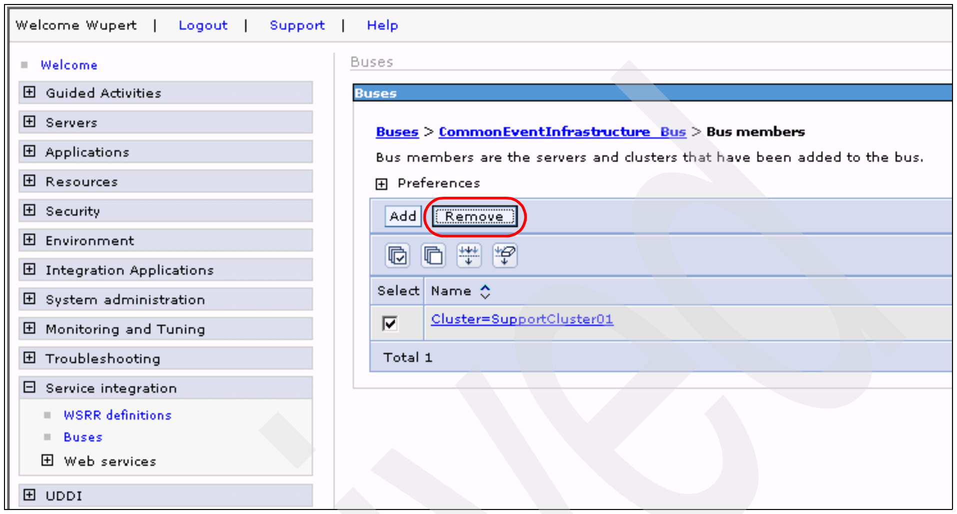


Figure 9-57 SupportCluster01 removal from the bus

7. Save and synchronize the changes, and click **OK** when complete.

Deleting the Cloudscape data source

To delete the Cloudscape data source, follow these steps:

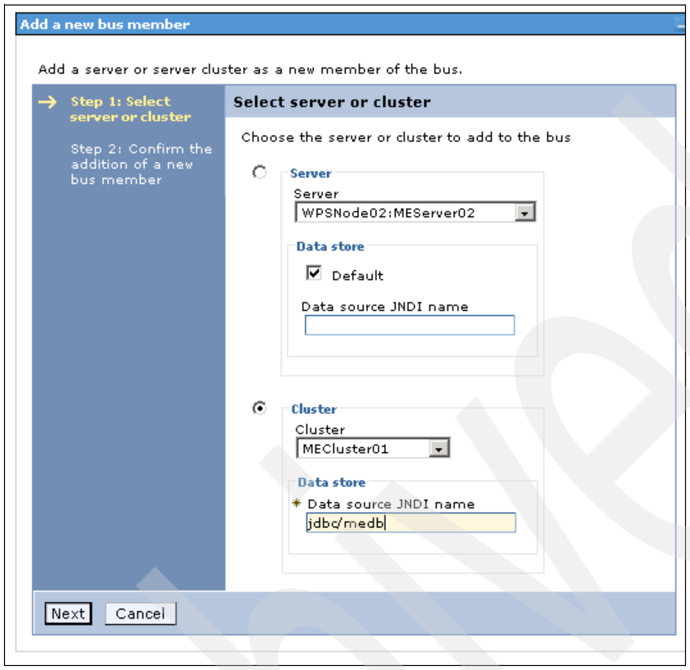
1. Navigate to **Resources** → **JDBC Providers**.
2. Set the scope to the **SupportCluster01** cluster, and click **Apply**.
3. Select the **Cloudscape JDBC Provider (XA)**.
4. Under Additional Properties, click **Data sources**.
5. Select **SupportCluster01-CommonEventInfrastructure_Bus** and click **Delete**.
6. Save and synchronize the changes, and click **OK** when complete.

Adding MECluster01 to the CEI bus

To add `MECluster01` to the CEI bus:

1. Navigate to **Service integration** → **Buses**.
2. Click **CommonEventInfrastructure_Bus**.
3. Under Topology, click **Bus members**.
4. Click **Add**.

5. Select **Cluster**, select the **MECluster01** cluster from the drop-down list, and set the Data source JNDI name to `jdbc/medb`, as shown in Figure 9-58.



Add a server or server cluster as a new member of the bus.

Choose the server or cluster to add to the bus

( ) Server
Server: WPSNode02:MEServer02
Data store: [✓] Default
Data source JNDI name:

(•) Cluster
Cluster: MECluster01
Data store:
* Data source JNDI name: jdbc/medb

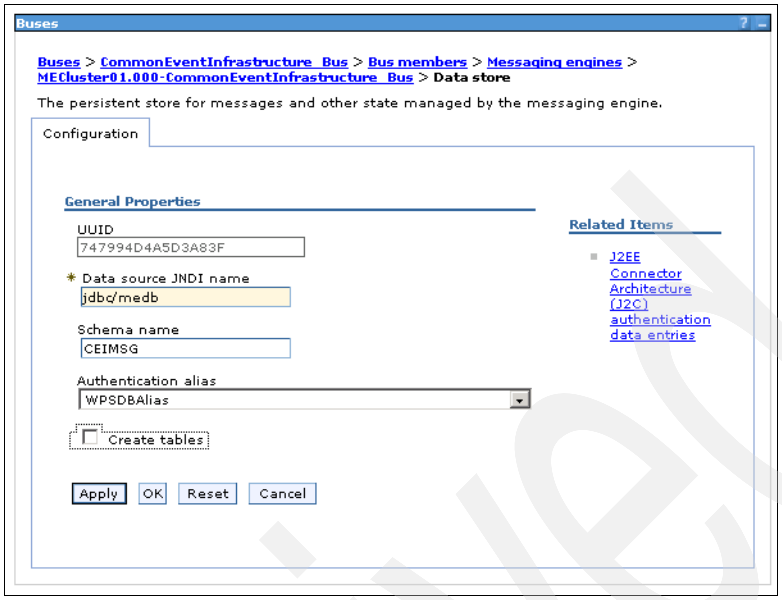
Figure 9-58 Adding the MECluster01 bus member

6. Click **Next**.
7. In the confirmation panel, click **Finish**.
8. Save and synchronize the changes, and click **OK** when complete.

Setting the CEIMSG schema

To set the CEIMSG schema:

1. Navigate to **Service integration** → **Buses**.
2. Click the **CommonEventInfrastructure_Bus**.
3. Under Topology, select **Bus members**.
4. Select bus member **Cluster=MECluster01**.
5. Select the messaging engine **MECluster01.000-CommonEventInfrastructure_Bus**.
6. Under Additional Properties, click **Data store**.
7. Set the Schema name to `CEIMSG`, and select **WPSDBAlias** from the Authentication alias drop-down list (Figure 9-59).


Buses

Buses > CommonEventInfrastructure_Bus > Bus members > Messaging engines > MECluster01.000-CommonEventInfrastructure_Bus > Data store

The persistent store for messages and other state managed by the messaging engine.

Configuration

General Properties

UUID
747994D4A5D3A83F

* Data source JNDI name
jdbc/medb

Schema name
CEIMSG

Authentication alias
WPSDBAlias

☐ Create tables

Related Items
- J2EE Connector Architecture (J2C) authentication data entries

Figure 9-59 Schema configuration for CEI messaging engine

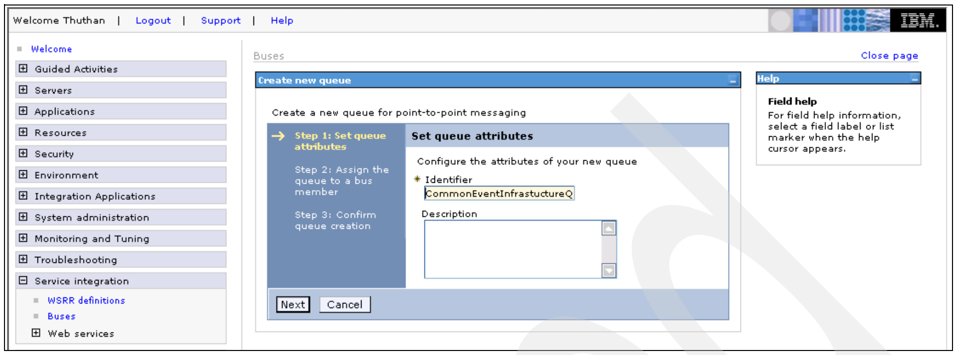
8. Clear the **Create tables** check box.
9. Click **OK**.
10. Save and synchronize the changes, and click **OK** when complete.

Setting the JMS destinations

To set the JMS destination, follow these steps:

1. Navigate to **Service integration** → **Buses**.
2. Click the **CommonEventInfrastructure_Bus**.
3. Under Destination resources, select **Destinations**.
4. Click **New**.
5. Set the destination type to **Queue**, and click **Next**.

6. Set the Identifier to `CommonEventInfrastructureQueueDestination` and click **Next**, as shown in Figure 9-60.



Create a new queue for point-to-point messaging

Configure the attributes of your new queue

Identifier: CommonEventInfrastuctureQ

Field help
For field help information, select a field label or list marker when the help cursor appears.

Figure 9-60 Setting the CEI queue attributes

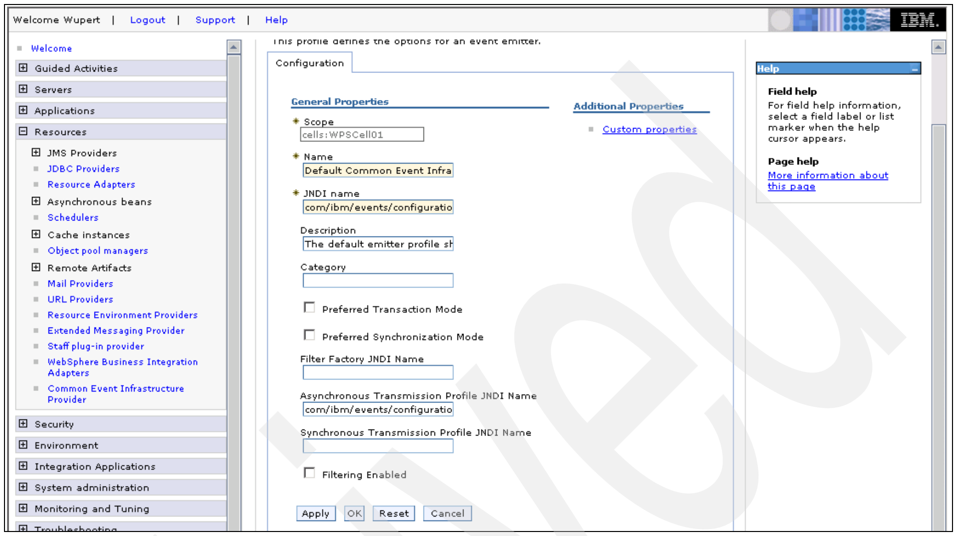
7. Select MECluster01 from the Bus member drop-down list, and click **Next**.
8. At the confirmation panel, click **Finish**.
9. Click **New**.
10. Set the destination type to **Topic space**, and click **Next**.
11. Set the Identifier to `CommonEventInfrastructureTopicDestination`, and click **Next**.
12. At the confirmation panel, click **Finish**.
13. Save and synchronize the changes, and click **OK** when complete.

Setting asynchronous event emission

To set asynchronous even emission, follow these steps:

1. Navigate to **Resources** → **Common Event Infrastructure Provider**.
2. Set the scope to the cell, and click **Apply**.
3. Under Additional Properties, click **Emitter Factory Profile**.
4. Click **Default Common Event Infrastructure emitter**.

5. Clear the Preferred synchronization mode check box and clear the Synchronous Transmission Profile JNDI Name entry, as shown in Figure 9-61.



This profile defines the options for an event emitter.

General Properties

* Scope: cells:WPSCell01
* Name: Default Common Event Infra
* JNDI name: com/ibm/events/configuratio

Description: The default emitter profile sh

Category:

☐ Preferred Transaction Mode

☐ Preferred Synchronization Mode

Filter Factory JNDI Name:

Asynchronous Transmission Profile JNDI Name: com/ibm/events/configuratio

Synchronous Transmission Profile JNDI Name:

☐ Filtering Enabled

Help

Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

Figure 9-61 Configuration of synchronous mode

6. Click **OK**.
7. Click **Default Common Event Infrastructure emitter for SupportCluster01**.
8. Clear the **Preferred Synchronization Mode** check box.
9. Click **OK**.
10. Save and synchronize the changes, and click **OK** when completed.

9.3.6 Enabling security

This section discusses the following:

- Enabling security with LDAP
- LTPA password
- Configuring LDAP registry
- Enabling Global Security

Enabling security with LDAP

Enabling security is the same mechanism used with WebSphere Application Server. The difference lies in the users in LDAP. You need to ensure that all nodes are running before enabling security.

When you ran the Business Process Container wizard, you added the following users:

- JMS user ID: sca (`itso4you`)
- JMS API User ID: sca (`itso4you`)
- Admin role: Administrators
- Monitor role: Administrators

When you created the cell initially, you set the SCA authentication user to be `sca` with password `itso4you`. These users and groups must exist in LDAP for the environment to work when security is enabled.

To enable security, follow these steps:

1. Ensure that the DB2 databases are started and start the LDAP server as the `idsldap user` with the command:

   ```
   ibmdirctl -D "cn=root" -w itso4you start
   ```

2. Start the deployment manager if it is not already running:

   ```
   c:\IBM\WebSphere\ProcServer\profiles\Dmgr01\bin\startManager.bat
   ```

3. Start the node on `wps1` if it is not running:

   ```
   c:\IBM\WebSphere\ProcServer\profiles\WPSNode01\bin\startNode.sh
   ```

4. Start the node on `wps2` if it is not running:

   ```
   c:\IBM\WebSphere\ProcServer\profiles\WPSNode02\bin\startNode.sh
   ```

When all the nodes are running, you can configure and enable security, and the nodes synchronize with these changes. If one of the nodes is down, then after a restart of the deployment manager, that node is no longer able to connect as part of the cell because it is unaware of the security changes.

LTPA password

You need to set up an LTPA password that will be used to encrypt and decrypt the LTPA keys. Follow these steps:

1. Open the `wps0` Administrative Console and navigate to **Security** → **Global security** → **Authentication mechanisms** → **LTPA**.

   It is necessary to set up an initial password. We set the Password and Confirm password fields to `itso4you`.

   The timeout period is the number of minutes that the LTPA token remains valid. By default, single sign-on (SSO) is enabled. If SSO is disabled, cookies are not stored on the authenticating client and every action requiring authentication will incur the overhead of a user authentication.

2. Click **OK**, save and synchronize the changes.

Configuring LDAP registry

WebSphere must be configured to locate the existing LDAP Registry.

Note: For information about how to configure an LDAP Registry, refer to Appendix C, "IBM Tivoli Directory Server (LDAP)" on page 439.

To configure the LDAP registry:

1. Open the `wps0` Administrative Console and navigate to **Security** → **Global security**.
2. Select **LDAP** under **User registries**.
3. Select **Advanced Lightweight Directory Access Protocol (LDAP) user registry settings** under Additional Properties.
4. The LDAP directory that was created uses an object class of `inetOrgPerson` and therefore the `User filter` value needs to be changed (the default is *ePerson*). The User filter should be set to:

   ```
   (&(uid=%v)(objectclass=inetOrgPerson))
   ```

   This filter allows the translation of the user short name to an LDAP entry (in conjunction with the User ID map, the LDAP structure includes a parameter `uid` to match this setting). This parameter enables the user name (`wsadmin` in this example) to be entered when logging in as opposed to the complete LDAP entry (for example, `cn=wsadmin,cn=users,cn=security,o=customer`).
5. When the user filter has been set to match the LDAP directory, select **OK**, save and synchronize.
6. Navigate to **Security** → **Global security**.
7. Select **LDAP** under **User registries**. The configuration settings in this panel identify how the LDAP server is contacted. Table 9-21 shows the settings.

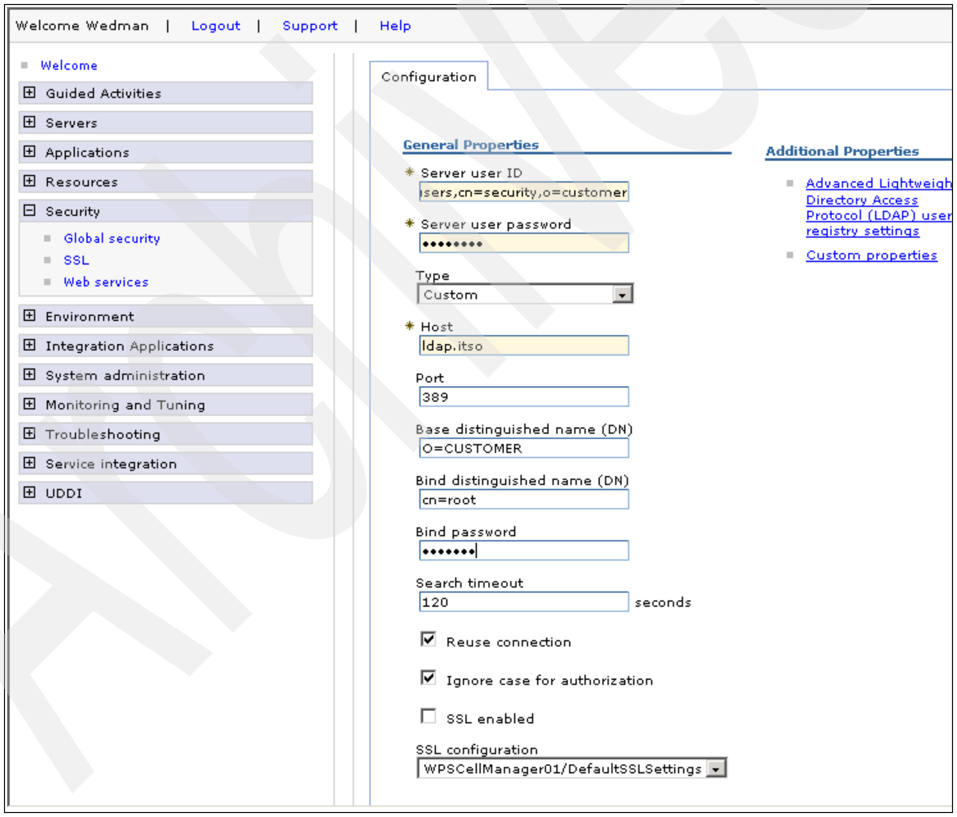
Table 9-21 LDAP settings

Property	**Value**	**Comment**
Server user ID	cn=wsadmin,cn=users,cn=security,o=customer	User ID for WebSphere security
Server user password	itso4you	
Type	Custom	Because we changed the User filter
Host	ldap.itso	Host name or IP of the LDAP server
Port	389	LDAP server port (no SSL)

Property	Value	Comment
Base distinguished name	O=CUSTOMER	Starting point for LDAP searches
Bind distinguished name	cn=root	Bind user for LDAP
Bind password	itso4you	

Note: By default *Ignore case for authorization* is enabled. This option is required because group data that is acquired from the user object is not guaranteed to be identical to group data that is acquired directly from LDAP.

Figure 9-62 lists the appropriate values.



Server user ID: isers,cn=security,o=customer
Server user password: ••••••••
Type: Custom
Host: ldap.itso
Port: 389
Base distinguished name (DN): O=CUSTOMER
Bind distinguished name (DN): cn=root
Bind password: •••••••
Search timeout: 120 seconds
[✓] Reuse connection
[✓] Ignore case for authorization
[ ] SSL enabled
SSL configuration: WPSCellManager01/DefaultSSLSettings

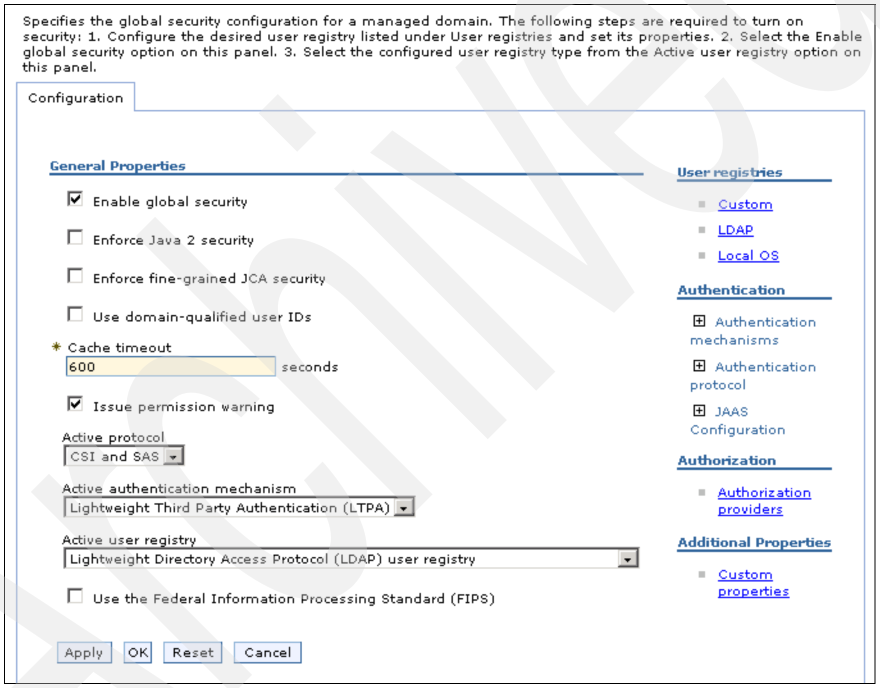
Figure 9-62 LDAP Configuration in WebSphere

8. When completed, select **OK**, save and synchronize.

Enabling Global Security

The final configuration step to enable security checking through the LDAP registry is to enable global security. Follow these steps:

1. Navigate to **Security** → **Global security**, and select **Enable global security**.
2. In this example, you will not enforce the more restrictive Java 2 security because the applications are not Java 2 security enabled. So, clear this check box.
3. In the same panel, under Active authentication mechanism ensure that **Lightweight Third Party Authentication (LTPA)** is selected.
4. Under Active user registry, select **Lightweight Directory Access Protocol (LDAP) user registry** as shown in Figure 9-63.


Specifies the global security configuration for a managed domain. The following steps are required to turn on security: 1. Configure the desired user registry listed under User registries and set its properties. 2. Select the Enable global security option on this panel. 3. Select the configured user registry type from the Active user registry option on this panel.

Configuration

General Properties

[✓] Enable global security

[ ] Enforce Java 2 security

[ ] Enforce fine-grained JCA security

[ ] Use domain-qualified user IDs

* Cache timeout: 600 seconds

[✓] Issue permission warning

Active protocol: CSI and SAS

Active authentication mechanism: Lightweight Third Party Authentication (LTPA)

Active user registry: Lightweight Directory Access Protocol (LDAP) user registry

[ ] Use the Federal Information Processing Standard (FIPS)

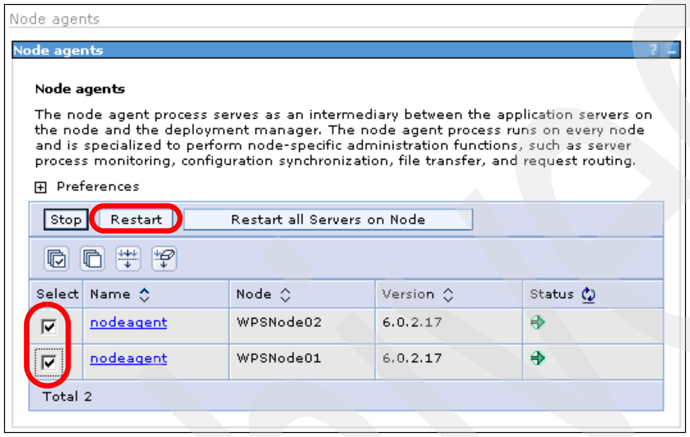
Figure 9-63 Enabling global security: settings

5. When complete, select **OK**, save and synchronize the changes.

Restarting the environment

To enable the security settings, it is essential that you stop and restart all servers, nodes, and the deployment manager. Follow these steps:

1. Open the Administrative Console for the deployment manager hosted on `wps0`.
2. Navigate to **System administration** → **Node agents** and select both nodes.
3. Click **Restart** to cause the nodes to stop and start, as shown in Figure 9-64.



Node agents

Node agents

Node agents

The node agent process serves as an intermediary between the application servers on the node and the deployment manager. The node agent process runs on every node and is specialized to perform node-specific administration functions, such as server process monitoring, configuration synchronization, file transfer, and request routing.

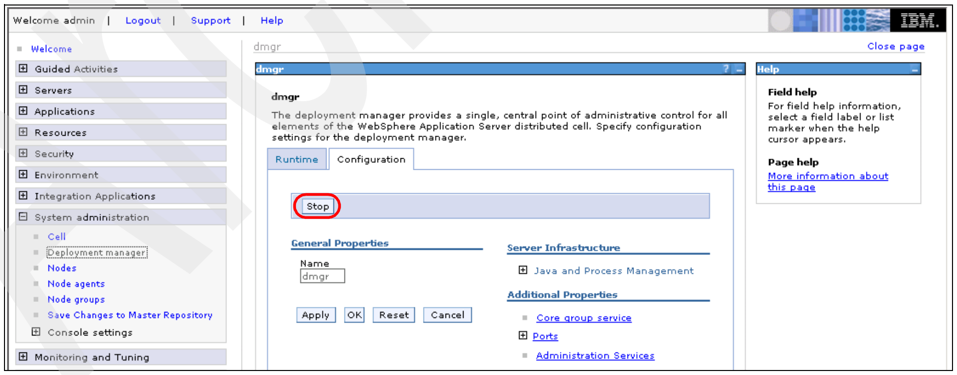
⊞ Preferences

Select	Name	Node	Version	Status
✓	nodeagent	WPSNode02	6.0.2.17	➔
✓	nodeagent	WPSNode01	6.0.2.17	➔

Total 2

Figure 9-64 Node restart

4. Navigate to **System administration** → **Deployment manager** and click **Stop**, as shown in Figure 9-65.



dmgr

dmgr

The deployment manager provides a single, central point of administrative control for all elements of the WebSphere Application Server distributed cell. Specify configuration settings for the deployment manager.

General Properties

Name: dmgr

Server Infrastructure

⊞ Java and Process Management

Additional Properties

- Core group service
- ⊞ Ports
- Administration Services

Help

Field help

For field help information, select a field label or list marker when the help cursor appears.

Page help

More information about this page

Figure 9-65 Stopping the deployment manager

5. Click **OK** in the verification panel. Wait for the deployment manager to stop.

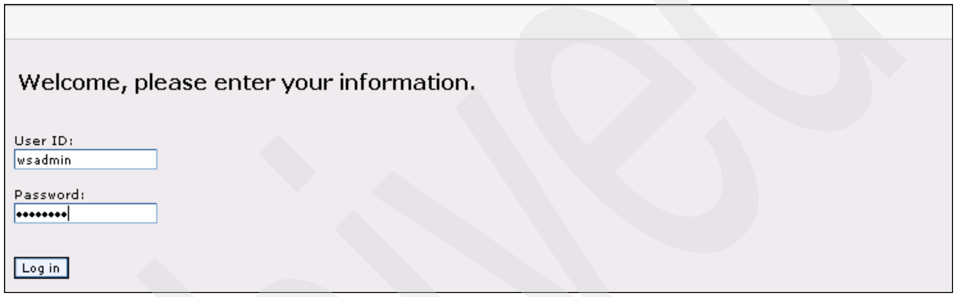
6. Start the deployment manager. Open a command prompt on `wps0`, change directory to %*WAS_HOME*%\profiles\Dmgr01\bin (for example, C:\IBM\WebSphere\ProcServer\profiles\Dmgr01\bin) and issue the command:

   ```
   startManager.bat
   ```

7. When restarted, launch the Administrative Console using:

   ```
   https://wps0.itso:9043/ibm/console
   ```

8. After a security certificate is accepted, the prompt requests a user ID and password, as shown in Figure 9-66. Enter a user ID of `wsadmin` and a password of `itso4you`.



Welcome, please enter your information.

User ID: wsadmin

Password: ••••••••

Figure 9-66 Secure Administrative Console access

Note: If there is a problem with the security settings and access is no longer available, you can disable it from a **`wsadmin`** command line with the following commands:

```
cd c:\IBM\WebSphere\ProcServer\bin
wsadmin.bat -conntype NONE
securityoff
quit
```

Access to applications such as BPC Explorer now require an authenticated user. The BPC Explorer application is set to *All authenticated*. So, any valid user from LDAP allows access.

9.3.7 Verifying the topology

This section includes the following:

- Checking the messaging engine database schemas
- Checking the CEI database schemas
- Starting the messaging cluster
- Starting the remaining clusters

Checking the messaging engine database schemas

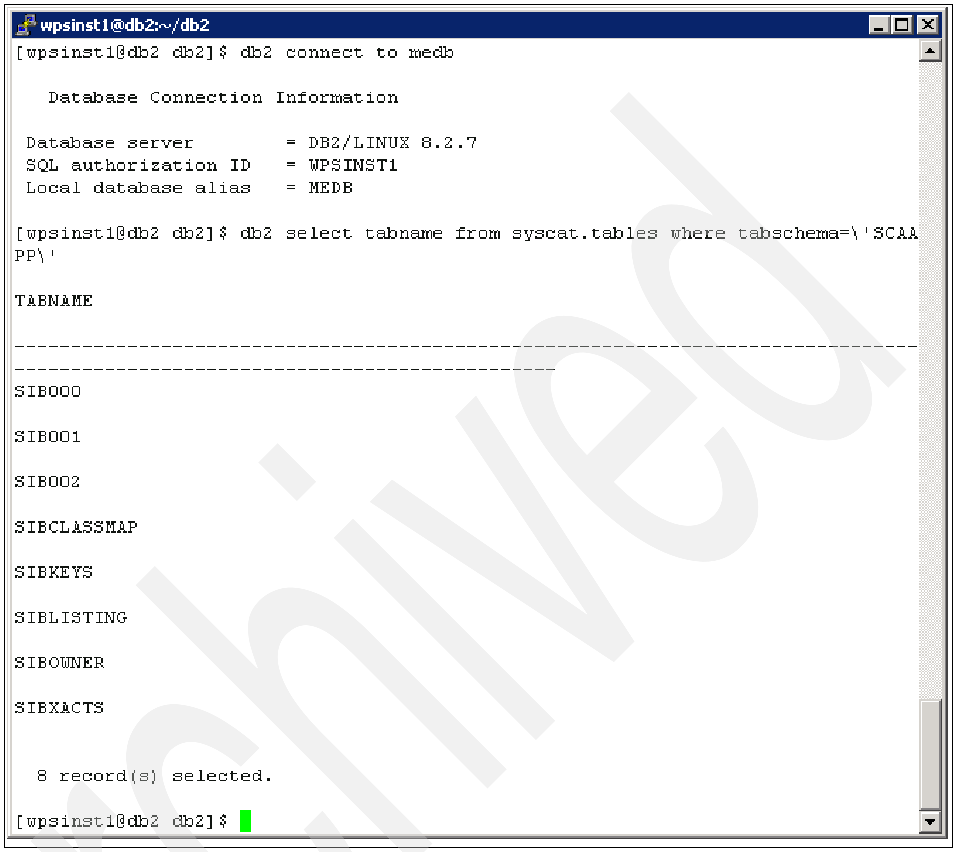
In this example, DB2 is the RDBMS. To check the messaging engine database schemas, follow these steps:

1. Log in to the database server as the instance user. In this example this is `wpsinst1`.
2. Run the following commands:

```
db2 connect to MEDB
db2 select tabname from syscat.tables where tabschema=\'SCAAPP\'
db2 select tabname from syscat.tables where tabschema=\'SCASYS\'
db2 select tabname from syscat.tables where tabschema=\'CEIMSG\'
```

> **Note:** On Windows platforms, do not use the backslashes (\) with the **`db2 select`** command.

Each of the schemas should have eight tables. Figure 9-67 shows example output for the SCAAPP schema.


wpsinst1@db2:~/db2

```
[wpsinst1@db2 db2]$ db2 connect to medb

   Database Connection Information

 Database server        = DB2/LINUX 8.2.7
 SQL authorization ID   = WPSINST1
 Local database alias   = MEDB

[wpsinst1@db2 db2]$ db2 select tabname from syscat.tables where tabschema=\'SCAA
PP\'

TABNAME

--------------------------------------------------------------------------------
------------------------------------------------
SIB000

SIB001

SIB002

SIBCLASSMAP

SIBKEYS

SIBLISTING

SIBOWNER

SIBXACTS


  8 record(s) selected.

[wpsinst1@db2 db2]$
```

Figure 9-67 SCAAPP schema

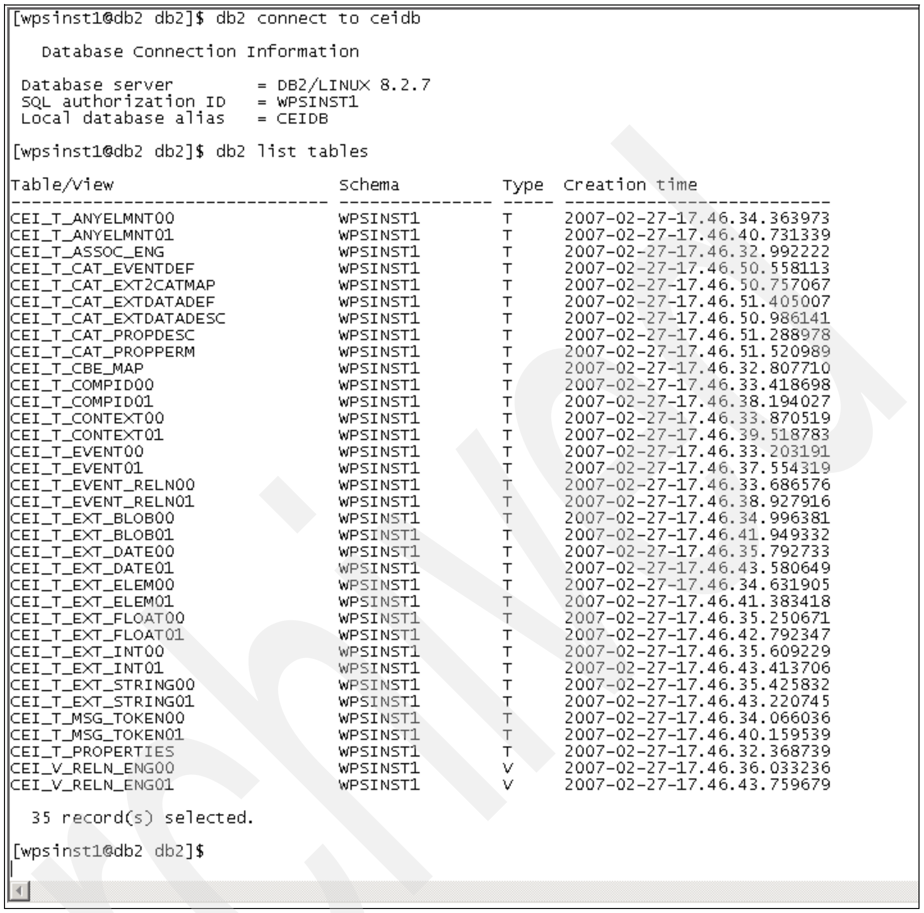
Checking the CEI database schemas

In this example, DB2 is the RDBMS. To check the CEI database schemas, follow these steps:

1. Log in to the database server as the instance user. In this example this is `wpsinst1`.
2. Run the following commands:

```
db2 connect to CEIDB
db2 list tables
```

There are 35 tables. Figure 9-68 shows example output.



```
[wpsinst1@db2 db2]$ db2 connect to ceidb

   Database Connection Information

 Database server        = DB2/LINUX 8.2.7
 SQL authorization ID   = WPSINST1
 Local database alias   = CEIDB

[wpsinst1@db2 db2]$ db2 list tables

Table/View                      Schema          Type  Creation time
------------------------------- --------------- ----- --------------------------
CEI_T_ANYELMNT00                WPSINST1        T     2007-02-27-17.46.34.363973
CEI_T_ANYELMNT01                WPSINST1        T     2007-02-27-17.46.40.731339
CEI_T_ASSOC_ENG                 WPSINST1        T     2007-02-27-17.46.32.992222
CEI_T_CAT_EVENTDEF              WPSINST1        T     2007-02-27-17.46.50.558113
CEI_T_CAT_EXT2CATMAP            WPSINST1        T     2007-02-27-17.46.50.757067
CEI_T_CAT_EXTDATADEF            WPSINST1        T     2007-02-27-17.46.51.405007
CEI_T_CAT_EXTDATADESC           WPSINST1        T     2007-02-27-17.46.50.986141
CEI_T_CAT_PROPDESC              WPSINST1        T     2007-02-27-17.46.51.288978
CEI_T_CAT_PROPPERM              WPSINST1        T     2007-02-27-17.46.51.520989
CEI_T_CBE_MAP                   WPSINST1        T     2007-02-27-17.46.32.807710
CEI_T_COMPID00                  WPSINST1        T     2007-02-27-17.46.33.418698
CEI_T_COMPID01                  WPSINST1        T     2007-02-27-17.46.38.194027
CEI_T_CONTEXT00                 WPSINST1        T     2007-02-27-17.46.33.870519
CEI_T_CONTEXT01                 WPSINST1        T     2007-02-27-17.46.39.518783
CEI_T_EVENT00                   WPSINST1        T     2007-02-27-17.46.33.203191
CEI_T_EVENT01                   WPSINST1        T     2007-02-27-17.46.37.554319
CEI_T_EVENT_RELN00              WPSINST1        T     2007-02-27-17.46.33.686576
CEI_T_EVENT_RELN01              WPSINST1        T     2007-02-27-17.46.38.927916
CEI_T_EXT_BLOB00                WPSINST1        T     2007-02-27-17.46.34.996381
CEI_T_EXT_BLOB01                WPSINST1        T     2007-02-27-17.46.41.949332
CEI_T_EXT_DATE00                WPSINST1        T     2007-02-27-17.46.35.792733
CEI_T_EXT_DATE01                WPSINST1        T     2007-02-27-17.46.43.580649
CEI_T_EXT_ELEM00                WPSINST1        T     2007-02-27-17.46.34.631905
CEI_T_EXT_ELEM01                WPSINST1        T     2007-02-27-17.46.41.383418
CEI_T_EXT_FLOAT00               WPSINST1        T     2007-02-27-17.46.35.250671
CEI_T_EXT_FLOAT01               WPSINST1        T     2007-02-27-17.46.42.792347
CEI_T_EXT_INT00                 WPSINST1        T     2007-02-27-17.46.35.609229
CEI_T_EXT_INT01                 WPSINST1        T     2007-02-27-17.46.43.413706
CEI_T_EXT_STRING00              WPSINST1        T     2007-02-27-17.46.35.425832
CEI_T_EXT_STRING01              WPSINST1        T     2007-02-27-17.46.43.220745
CEI_T_MSG_TOKEN00               WPSINST1        T     2007-02-27-17.46.34.066036
CEI_T_MSG_TOKEN01               WPSINST1        T     2007-02-27-17.46.40.159539
CEI_T_PROPERTIES                WPSINST1        T     2007-02-27-17.46.32.368739
CEI_V_RELN_ENG00                WPSINST1        V     2007-02-27-17.46.36.033236
CEI_V_RELN_ENG01                WPSINST1        V     2007-02-27-17.46.43.759679

  35 record(s) selected.

[wpsinst1@db2 db2]$
```

Figure 9-68 CEIDB tables

Starting the messaging cluster

To start the cluster:

1. Ensure that all nodes are running. If a node is not running, start it with the following commands:
   a. Open a command prompt on the node's host machine.
   b. Change directory to the node's bin directory:

      cd\IBM\WebSphere\ProcServer\profiles\WPSNode01\bin

   c. Start the node:

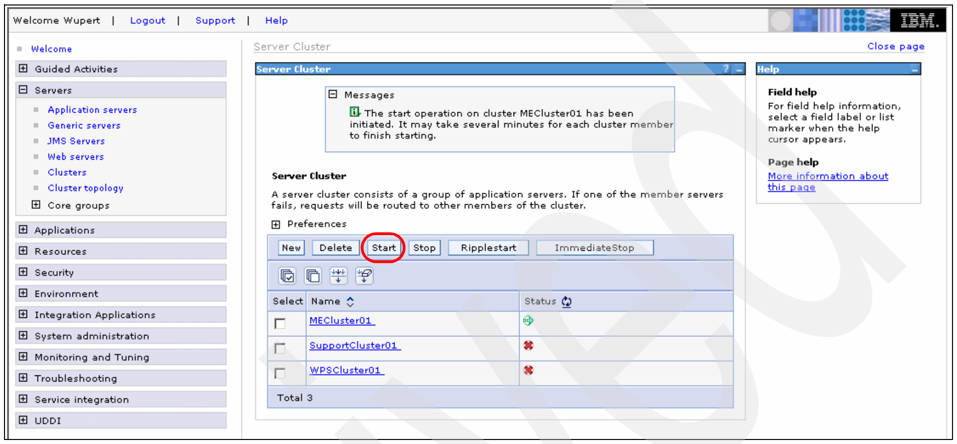
      `startNode.bat`

2. Start the messaging cluster:

   a. Open the cell Administrative Console and login.

   b. Navigate to **Servers** → **Clusters**.

   c. Select **MECluster01** and click **Start**, as shown in Figure 9-69.



Server Cluster

Messages

The start operation on cluster MECluster01 has been initiated. It may take several minutes for each cluster member to finish starting.

Server Cluster

A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster.

Select	Name	Status
☐	MECluster01	➔
☐	SupportCluster01	✖
☐	WPSCluster01	✖

Total 3

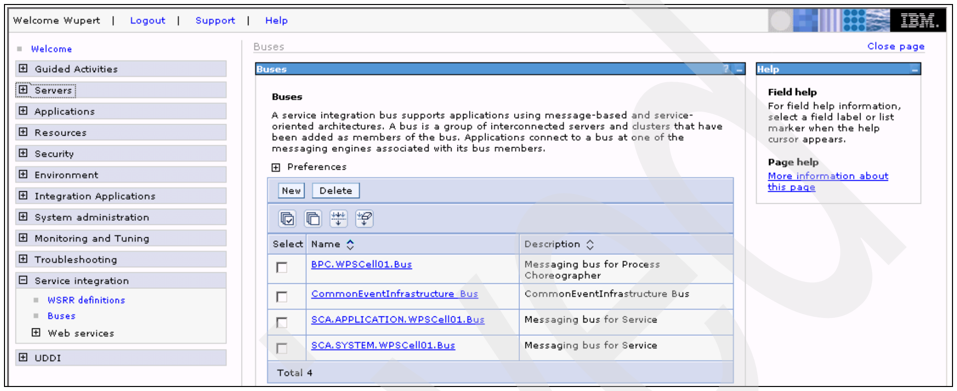
Figure 9-69 Starting the MECluster01 cluster

   d. Verify that the cluster starts successfully. The red cross should change to a solid green arrow. You have to refresh the view to see the change in status.

3. Verify that the messaging engines have started:

   a. Navigate to **Service integration → Buses**.

      There should be four buses listed—two for the SCA Application and SCA System buses, a Common Event Infrastructure bus, and the BPC bus—as shown in Figure 9-70.



Welcome Wupert | Logout | Support | Help

Welcome
Guided Activities
Servers
Applications
Resources
Security
Environment
Integration Applications
System administration
Monitoring and Tuning
Troubleshooting
Service integration
WSRR definitions
Buses
Web services
UDDI

Buses

Buses

A service integration bus supports applications using message-based and service-oriented architectures. A bus is a group of interconnected servers and clusters that have been added as members of the bus. Applications connect to a bus at one of the messaging engines associated with its bus members.

Preferences

New | Delete

Select	Name	Description
☐	BPC.WPSCell01.Bus	Messaging bus for Process Choreographer
☐	CommonEventInfrastructure_Bus	CommonEventInfrastructure Bus
☐	SCA.APPLICATION.WPSCell01.Bus	Messaging bus for Service
☐	SCA.SYSTEM.WPSCell01.Bus	Messaging bus for Service
Total 4		

Help

Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

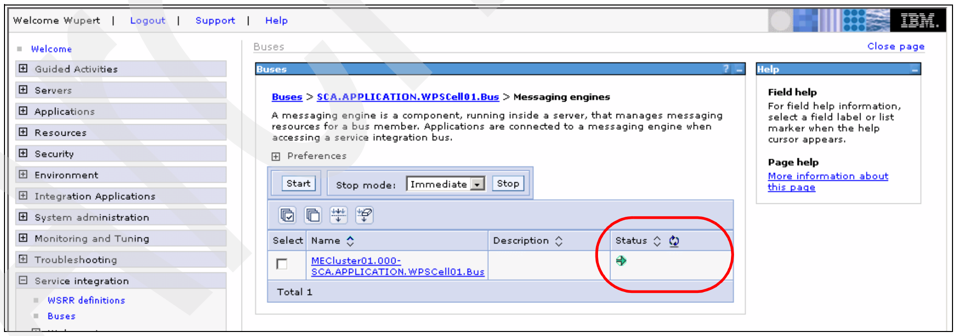
Close page

Figure 9-70 Service integration buses

   b. Click **SCA.APPLICATION.WPSCell01.Bus**.

   c. Under the Topology section, click **Messaging engines**.

   d. The status of the `MECluster01.000-SCA.APPLICATION.WPSCell01.Bus` messaging engine should be started, as shown in Figure 9-71.



Welcome Wupert | Logout | Support | Help

Welcome
Guided Activities
Servers
Applications
Resources
Security
Environment
Integration Applications
System administration
Monitoring and Tuning
Troubleshooting
Service integration
WSRR definitions
Buses

Buses

Buses > SCA.APPLICATION.WPSCell01.Bus > **Messaging engines**

A messaging engine is a component, running inside a server, that manages messaging resources for a bus member. Applications are connected to a messaging engine when accessing a service integration bus.

Preferences

Start | Stop mode: Immediate | Stop

Select	Name	Description	Status
☐	MECluster01.000-SCA.APPLICATION.WPSCell01.Bus		➜
Total 1			

Help

Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

Close page

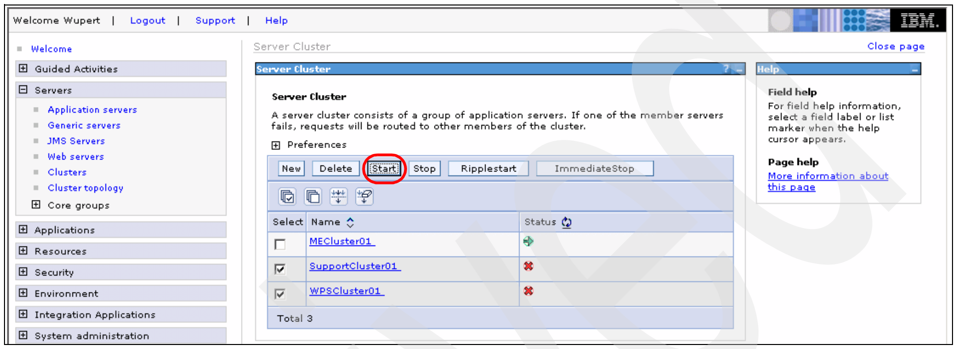
Figure 9-71 Messaging engine in started state

Repeat this process for the `SCA.SYSTEM.WPSCell01.Bus`, `BPC.WPSCell01.Bus` and `CommonEventInfrastructure_Bus` buses.

Starting the remaining clusters

To start the remaining clusters, follow these steps:

1. Open the Administrative Console and login.
2. Navigate to **Servers** → **Clusters**.
3. Select the cluster check box for each stopped cluster and click **Start**, as shown in Figure 9-69.



Server Cluster

A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster.

For field help information, select a field label or list marker when the help cursor appears.

Select	Name	Status
☐	MECluster01	➜
☑	SupportCluster01	✖
☑	WPSCluster01	✖
Total 3		

Figure 9-72 Starting the remaining clusters

4. Verify that the clusters start successfully. The red cross should change to a solid green arrow. You can view detailed information in the SystemOut.log log file on each cluster member.

9.4 Installing and testing the scenario

This section describes how to install the applications to test the scenario, how to configure IBM HTTP Server, how to enable SSL, and how to test the scenario business process.

9.4.1 Installing the enterprise applications

To test the business process logic for this topology, you need to create a presentation logic layer and a remote enterprise service layer. The presentation logic layer provides the client for the business module, and the remote enterprise service layer presents services that are required for the business flows.

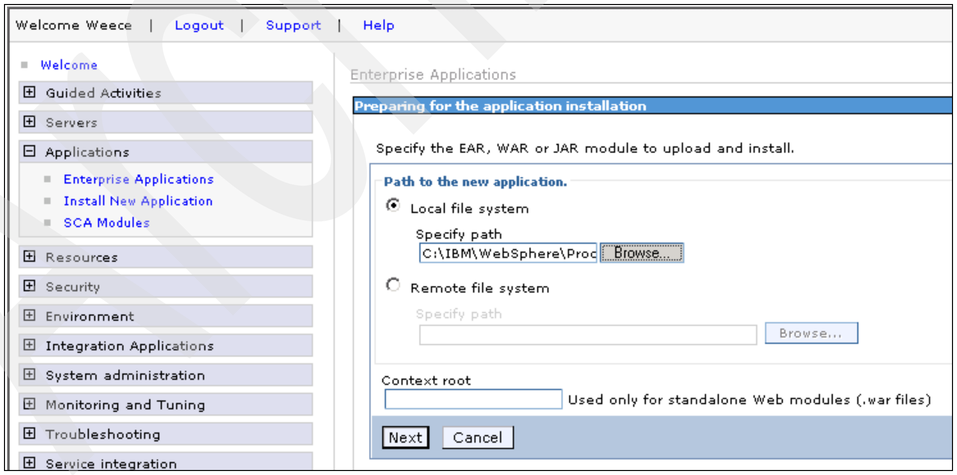
In this example, the presentation logic layer is hosted on the business cluster alongside the business application.The remote enterprise service layer is hosted on a stand-alone WebSphere Application Server instance.

The presentation logic application ITSOWPSCplxPrsApp.ear is installed on `wps1` and `wps2` to `WPSCluster01`. The enterprise service application ITSOWPSCplxSrvApp.ear is installed on `wps3` to `WASNode01` on `server1`.

Deploying the business application

The application that supports the business process logic is provided by a single EAR file. This application must be installed to the cluster `WPSCluster01`. Follow these steps:

1. Open the `wps0` Administrative Console and log in.
2. Navigate to **Applications** → **Install New Application**.
3. You install the enterprise application ITSOWPSCplxBusApp.ear, which includes the business process logic. This file is located in the additional material that is supplied with this book. To learn how to obtain it, see Appendix A, "Additional material" on page 427. Copy ITSOWPSCplxBusApp.ear from the additional material supplied with this book to c:\IBM\WebSphere\ProcServer\installableApps.
4. In the Administrative Console, select **Local file system** and click **Browse**.
5. Navigate to the c:\IBM\WebSphere\ProcServer\installableApps folder.
6. Select the business module to be installed, in this example ITSOWPSCplxBusApp.ear, and click **Open**.
7. From the Administrative Console, click **Next**, as shown in Figure 9-73.



Specify the EAR, WAR or JAR module to upload and install.

Path to the new application.

[● Local file system]

Specify path
C:\IBM\WebSphere\Proc
Browse...

○ Remote file system

Specify path
Browse...

Context root
Used only for standalone Web modules (.war files)

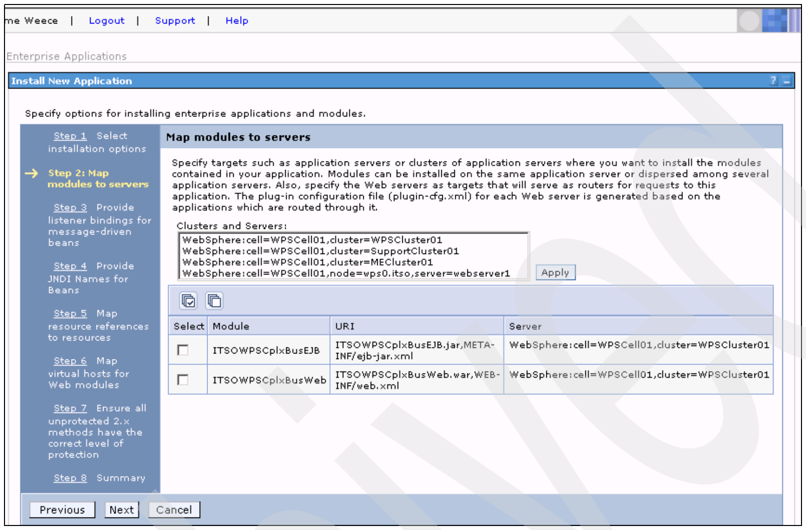
Next Cancel

Figure 9-73 Installing the business module

8. In the next panel, leave the settings as default and click **Next**.

9. In the next panel, leave the settings as default and click **Next**.

   All the other application deployment settings are left as default, but notably the application must be deployed to the `WPSCluster01` cluster in step 2 of the wizard, 'Map modules to servers' as shown in Figure 9-74.



me Weece | Logout | Support | Help

Enterprise Applications

Install New Application

Specify options for installing enterprise applications and modules.

Step 1 Select installation options
→ Step 2: Map modules to servers
Step 3 Provide listener bindings for message-driven beans
Step 4 Provide JNDI Names for Beans
Step 5 Map resource references to resources
Step 6 Map virtual hosts for Web modules
Step 7 Ensure all unprotected 2.x methods have the correct level of protection
Step 8 Summary

Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install the modules contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that will serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated based on the applications which are routed through it.

Clusters and Servers:

```
WebSphere:cell=WPSCell01,cluster=WPSCluster01
WebSphere:cell=WPSCell01,cluster=SupportCluster01
WebSphere:cell=WPSCell01,cluster=MECluster01
WebSphere:cell=WPSCell01,node=wps0.itso,server=webserver1
```

Apply

Select	Module	URI	Server
☐	ITSOWPSCplxBusEJB	ITSOWPSCplxBusEJB.jar,META-INF/ejb-jar.xml	WebSphere:cell=WPSCell01,cluster=WPSCluster01
☐	ITSOWPSCplxBusWeb	ITSOWPSCplxBusWeb.war,WEB-INF/web.xml	WebSphere:cell=WPSCell01,cluster=WPSCluster01

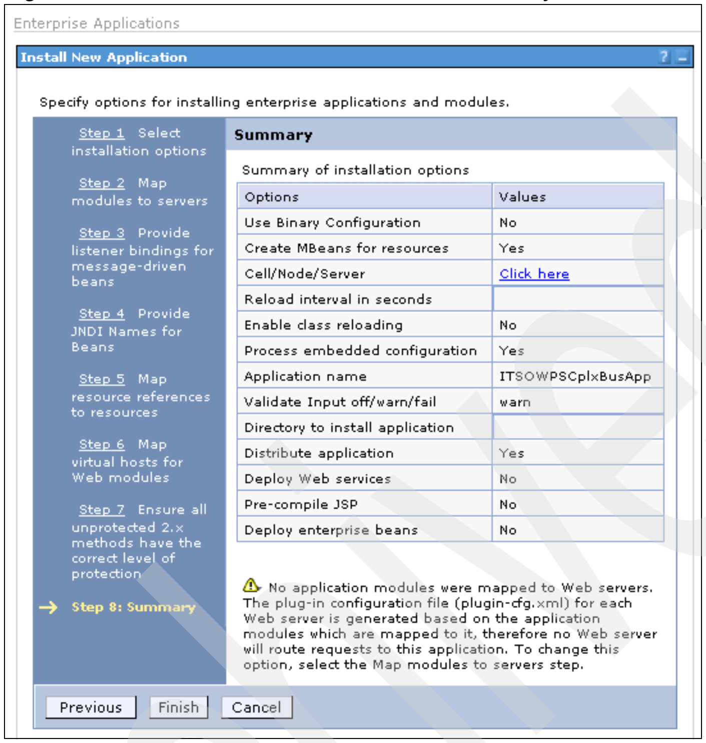
Previous | Next | Cancel

Figure 9-74 Verification of the correct application module mapping

10. Click the link to the summary panel, in this example **Step 8: Summary**.

11. Click **Finish**, as shown in Figure 9-75. You can ignore any warnings about activation specifications.

Figure 9-75 Business module installation summary


Enterprise Applications
Install New Application
Step 1 Select installation options
Step 2 Map modules to servers
Step 3 Provide listener bindings for message-driven beans
Step 4 Provide JNDI Names for Beans
Step 5 Map resource references to resources
Step 6 Map virtual hosts for Web modules
Step 7 Ensure all unprotected 2.x methods have the correct level of protection
Step 8: Summary
Previous
Finish
Cancel

Specify options for installing enterprise applications and modules.

Summary

Summary of installation options

Options	Values
Use Binary Configuration	No
Create MBeans for resources	Yes
Cell/Node/Server	Click here
Reload interval in seconds	
Enable class reloading	No
Process embedded configuration	Yes
Application name	ITSOWPSCplxBusApp
Validate Input off/warn/fail	warn
Directory to install application	
Distribute application	Yes
Deploy Web services	No
Pre-compile JSP	No
Deploy enterprise beans	No

No application modules were mapped to Web servers. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated based on the application modules which are mapped to it, therefore no Web server will route requests to this application. To change this option, select the Map modules to servers step.

12. When the application has been installed successfully, click **Save to Master Configuration**.

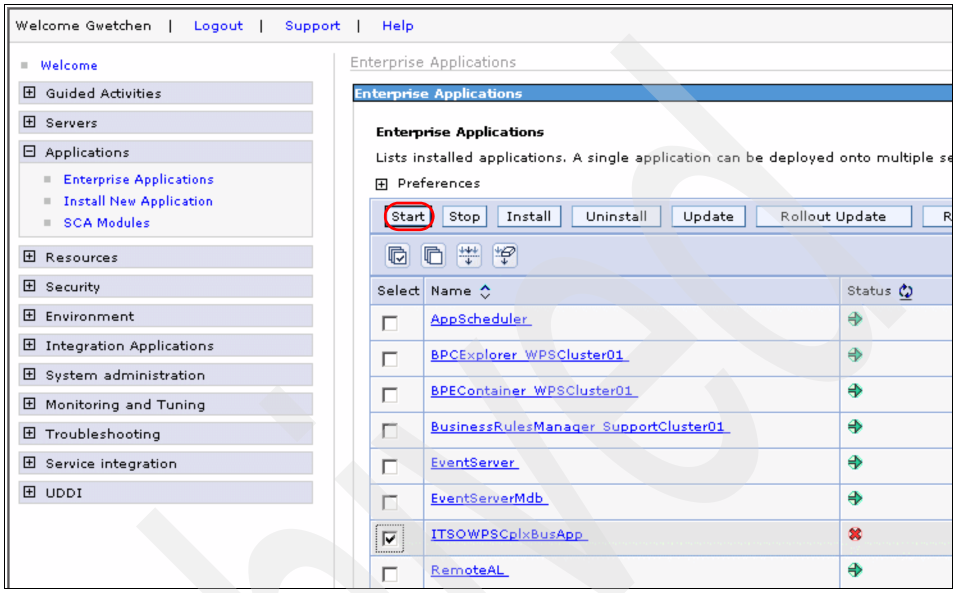
13. Select **Synchronize changes with Nodes**, and click **Save**.

14. When completed successfully, click **OK**.

15. Start the application:

a. Navigate to **Applications** → **Enterprise Applications**.

b. Select **ITSOWPSCplxBusApp** and click **Start**, as shown in Figure 9-76.



Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple se

Select	Name	Status
☐	AppScheduler	➔
☐	BPCExplorer_WPSCluster01	➔
☐	BPEContainer_WPSCluster01	➔
☐	BusinessRulesManager_SupportCluster01	➔
☐	EventServer	➔
☐	EventServerMdb	➔
☑	ITSOWPSCplxBusApp	✖
☐	RemoteAL	➔

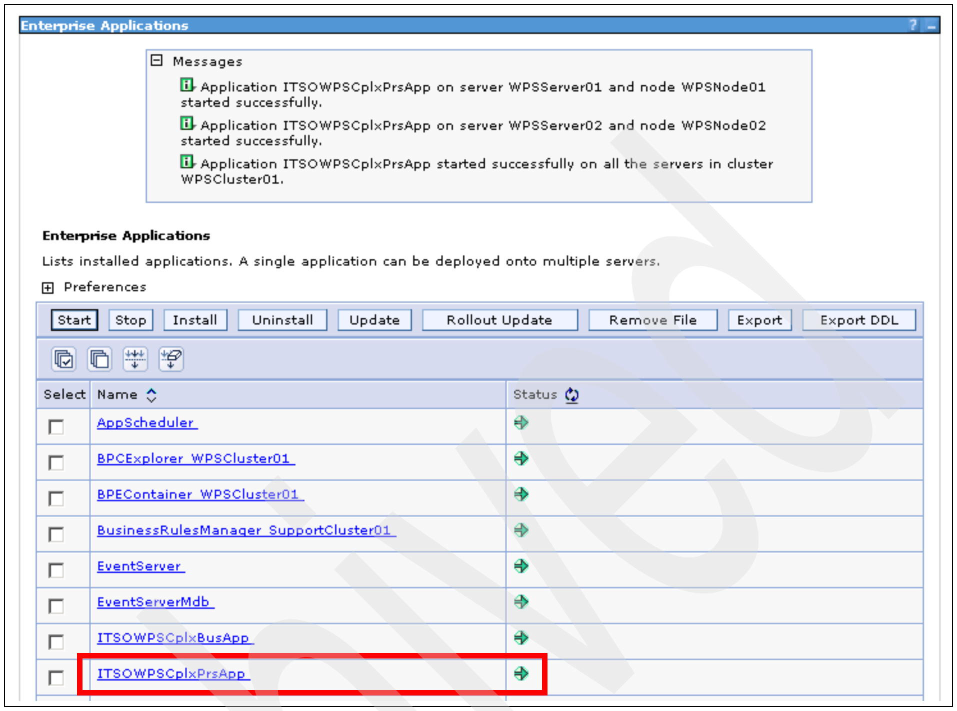
Figure 9-76 Starting the business module

Deploying the presentation logic application

To deploy the presentation logic application:

1. Repeat the steps described in “Deploying the business application” on page 401 to deploy and start the presentation logic application ITSOWPSCplxPrsApp.ear to the cluster `WPSCluster01`. You can also find this file in the additional material that accompanies this book.

2. Verify that the application starts, as shown in Figure 9-77.


Enterprise Applications

Messages

Application ITSOWPSCplxPrsApp on server WPSServer01 and node WPSNode01 started successfully.

Application ITSOWPSCplxPrsApp on server WPSServer02 and node WPSNode02 started successfully.

Application ITSOWPSCplxPrsApp started successfully on all the servers in cluster WPSCluster01.

Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

⊞ Preferences

Select	Name	Status
☐	AppScheduler	➔
☐	BPCExplorer_WPSCluster01	➔
☐	BPEContainer_WPSCluster01	➔
☐	BusinessRulesManager_SupportCluster01	➔
☐	EventServer	➔
☐	EventServerMdb	➔
☐	ITSOWPSCplxBusApp	➔
☐	ITSOWPSCplxPrsApp	➔

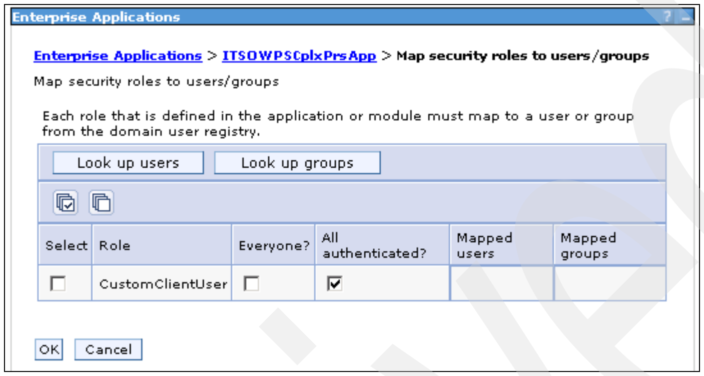
Figure 9-77 Installed enterprise application ITSOWPSCplxPrsApp

3. The presentation logic application should be set to use *All* authenticated rather than *Everyone*.

   a. In the Enterprise Applications view select **ITSOWPSCplxPrsApp**.

   b. Under Additional Properties, click **Map security roles to users/groups**.

   c. Select **All authenticated** (Figure 9-78).



Enterprise Applications

Enterprise Applications > ITSOWPSCplxPrsApp > Map security roles to users/groups

Map security roles to users/groups

Each role that is defined in the application or module must map to a user or group from the domain user registry.

Select	Role	Everyone?	All authenticated?	Mapped users	Mapped groups
☐	CustomClientUser	☐	☑		

Figure 9-78 Setting the security roles to all authenticated

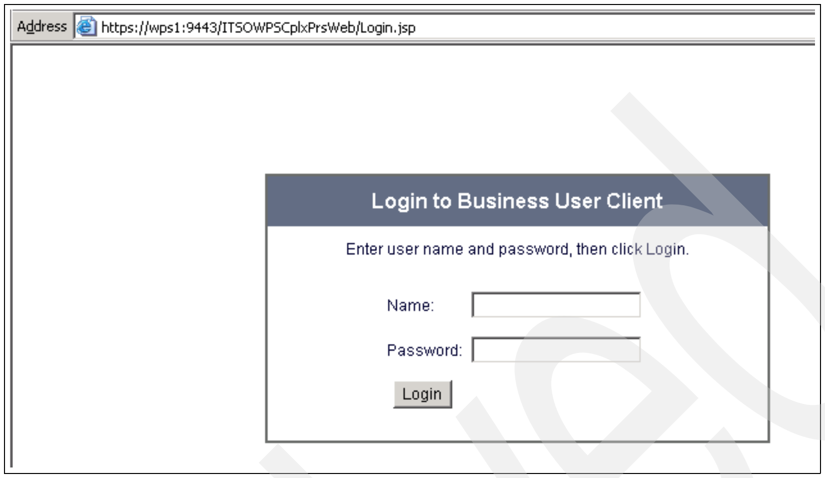
   d. Click **OK**. Save and synchronize the changes.

4. Verify that the presentation logic application is running on `WPSServer01` and `WPSServer02` within the cluster and can connect to the business layer.

   a. Enter one of the following URLs into a browser.

```
https://wps1.itso:9443/ITSOWPSCplxPrsWeb/faces/Index.jsp
https://wps2.itso:9443/ITSOWPSCplxPrsWeb/faces/Index.jsp
```

You should see a panel titled *Login to Business User Client*, as shown in Figure 9-79.


Address https://wps1:9443/ITSOWPSCplxPrsWeb/Login.jsp

Login to Business User Client

Enter user name and password, then click Login.

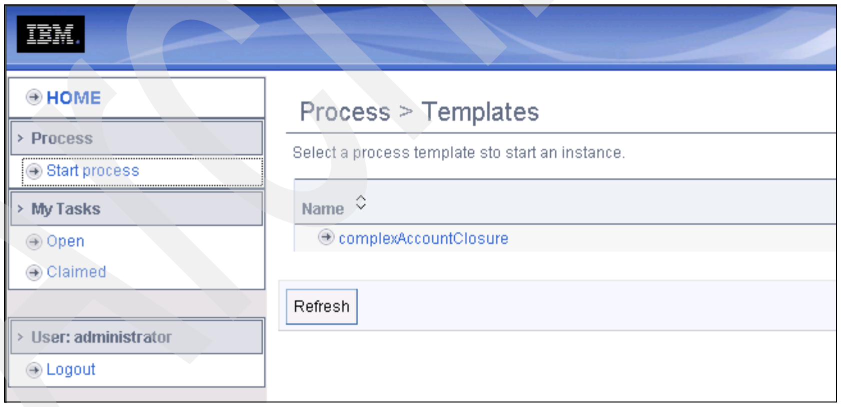
Name:

Password:

Figure 9-79 Presentation application logon panel

b. Log in by entering a name of `administrator` and a password of `itso4you`. This name and password is authenticated using the LDAP server.

c. In the Business User Client, click **Start process**. You should see the process `complexAccountClosure` defined (Figure 9-80).



Process > Templates

Select a process template sto start an instance.

Name
complexAccountClosure

Figure 9-80 complexAccountClosure process template

d. Click **Logout**.

Preparing the enterprise service layer

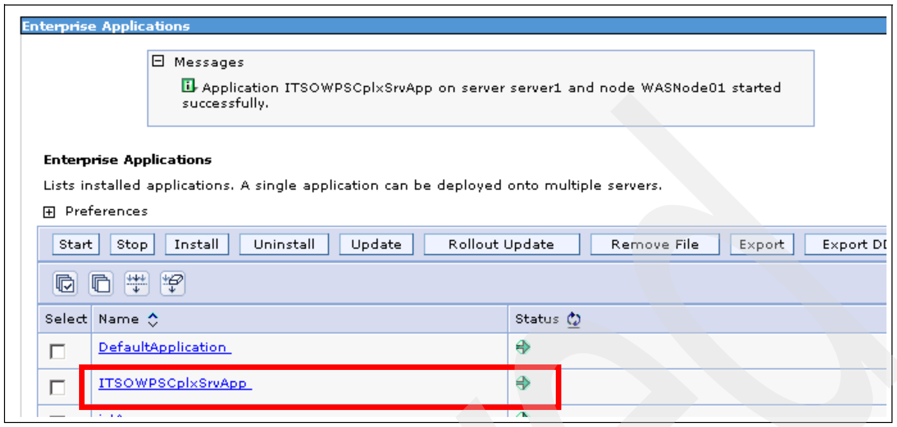
Define a WebSphere Application Server server to host the enterprise service layer, then install the enterprise service application. Follow these steps:

1. Create a new WebSphere Application Server profile using the profile creation wizard on `wps3`. On Windows platforms, launch the Profile Wizard by clicking **Start → All Programs → IBM WebSphere → Application Server Network Deployment V6 → Profile creation wizard**.
2. Define a profile with the following properties:
   - Profile type: `Application Server profile`
   - Profile name: `WASNode01`
   - Node name: `WASNode01`
   - Host name: `wps3.itso`
3. When the profile is created, navigate to the bin directory of the profile (which on our system was `C:\IBM\WebSphere\ProcServer\profiles\WASNode01\bin`) and start the server using the command:

   ```
   startServer server1
   ```
4. When the server is started, launch the Administrative Console:

   ```
   http://wps3.itso:9060/ibm/console
   ```
5. Enable Global Security using LTPA and LDAP. Refer to the following sections to complete this task:
   a. "LTPA password" on page 389
   b. "Configuring LDAP registry" on page 390
   c. "Enabling Global Security" on page 392
6. Restart the server.
7. When the server is started, log in to the Administrative Console using a user ID of `wsadmin` and a password of `itso4you`. This will be authenticated against the LDAP registry.
8. Install the enterprise service logic enterprise application ITSOWPSCplxSrvApp.ear. This file is located in the additional material that is supplied with this book. To learn how to obtain it, see Appendix A, "Additional material" on page 427.
   a. Copy ITSOWPSCplxSrvApp.ear from the additional material that is supplied with this book to c:\IBM\WebSphere\ProcServer\installableApps.
   b. In the Administrative Console, click **Applications → Install New Application** and install ITSOWPSCplxSrvApp.ear. When the application is installed, save the configuration.

c. Click **Applications** → **Enterprise Applications and start ITSOWPSCplxSrvApp** (Figure 9-81).


Enterprise Applications

Messages

Application ITSOWPSCplxSrvApp on server server1 and node WASNode01 started successfully.

Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

Preferences

Select	Name	Status
☐	DefaultApplication	➔
☐	ITSOWPSCplxSrvApp	➔

Figure 9-81 Installed enterprise application ITSOWPSCplxSrvApp

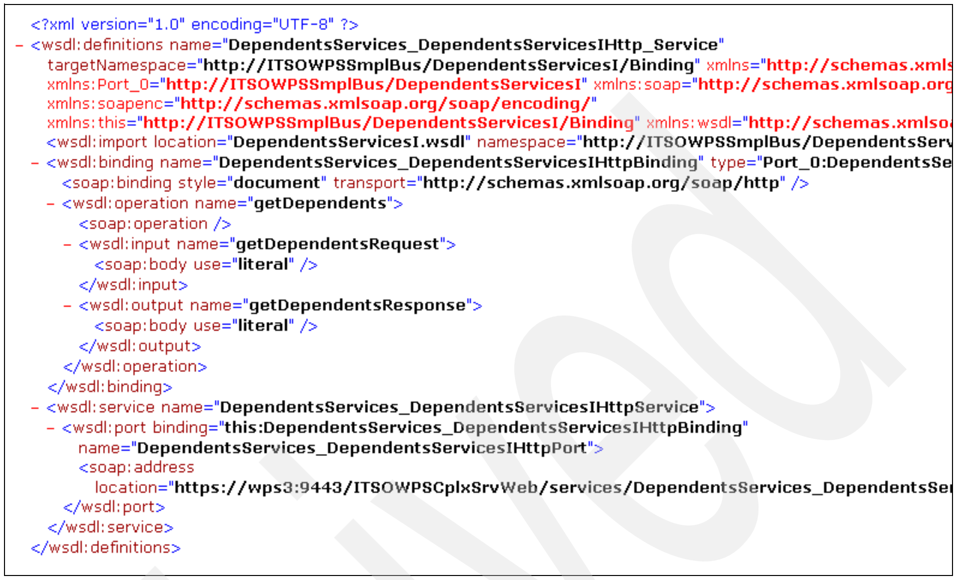
9. It is possible to test each of the remote enterprise services directly with a browser at the following URLs:

   – `https://wps3.itso:9443/ITSOWPSCplxSrvWeb/services/AccountingServices_AccountingServicesIHttpPort`

   – `https://wps3.itso:9443/ITSOWPSCplxSrvWeb/services/LegalServices_LegalServicesIHttpPort`

   – `https://wps3.itso:9443/ITSOWPSCplxSrvWeb/services/DependentsServices_DependentsServicesIHttpPort`

By appending `?wsdl` to these URLs, you can display the actual WSDL definition, as shown in Figure 9-82.



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions name="DependentsServices_DependentsServicesIHttp_Service"
    targetNamespace="http://ITSOWPSSmplBus/DependentsServicesI/Binding" xmlns="http://schemas.xmls
    xmlns:Port_0="http://ITSOWPSSmplBus/DependentsServicesI" xmlns:soap="http://schemas.xmlsoap.org
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:this="http://ITSOWPSSmplBus/DependentsServicesI/Binding" xmlns:wsdl="http://schemas.xmlso
    <wsdl:import location="DependentsServicesI.wsdl" namespace="http://ITSOWPSSmplBus/DependentsServ
  - <wsdl:binding name="DependentsServices_DependentsServicesIHttpBinding" type="Port_0:DependentsSe
      <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
    - <wsdl:operation name="getDependents">
        <soap:operation />
      - <wsdl:input name="getDependentsRequest">
          <soap:body use="literal" />
        </wsdl:input>
      - <wsdl:output name="getDependentsResponse">
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
  - <wsdl:service name="DependentsServices_DependentsServicesIHttpService">
    - <wsdl:port binding="this:DependentsServices_DependentsServicesIHttpBinding"
        name="DependentsServices_DependentsServicesIHttpPort">
        <soap:address
          location="https://wps3:9443/ITSOWPSCplxSrvWeb/services/DependentsServices_DependentsSe
      </wsdl:port>
    </wsdl:service>
  </wsdl:definitions>
```

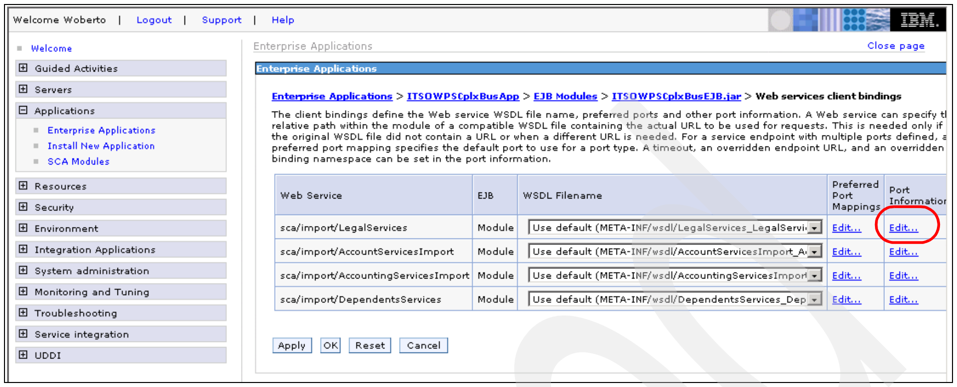
Figure 9-82 Requesting the WSDL definition

Configuring application deployment

There are additional steps to configure the end-to-end application deployment. You need to configure the business application for the remote enterprise services. Follow these steps:

1. Open the cell deployment manager's Administrative Console and login.
2. Navigate to **Applications** → **Enterprise Applications**.
3. Select **ITSOWPSCplxBusApp**.
4. Under Related Items, click **EJB Modules**.
5. Select the JAR file presented, **ITSOWPSCplxBusEJB.jar**.
6. Under Additional Properties, click **Web services client bindings**.

7. For the sca/import/LegalServices service, click **Edit** under **Port Information**, as shown in Figure 9-83.



Enterprise Applications > ITSOWPSCplxBusApp > EJB Modules > ITSOWPSCplxBusEJB.jar > Web services client bindings

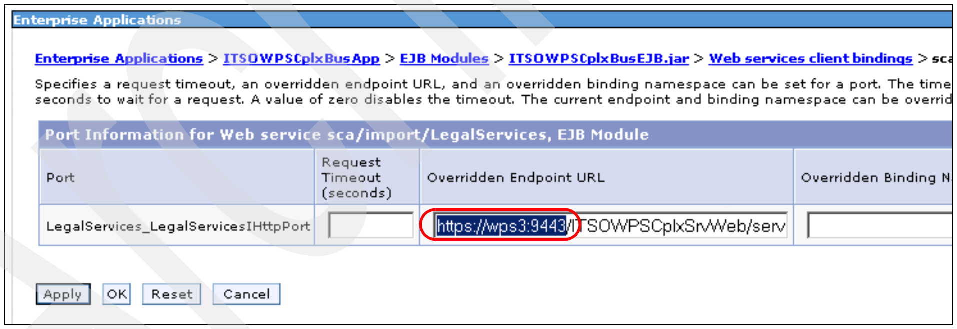
The client bindings define the Web service WSDL file name, preferred ports and other port information. A Web service can specify tl relative path within the module of a compatible WSDL file containing the actual URL to be used for requests. This is needed only if the original WSDL file did not contain a URL or when a different URL is needed. For a service endpoint with multiple ports defined, a preferred port mapping specifies the default port to use for a port type. A timeout, an overridden endpoint URL, and an overridden binding namespace can be set in the port information.

Web Service	EJB	WSDL Filename	Preferred Port Mappings	Port Informatior
sca/import/LegalServices	Module	Use default (META-INF/wsdl/LegalServices_LegalServi	Edit...	Edit...
sca/import/AccountServicesImport	Module	Use default (META-INF/wsdl/AccountServicesImport_A	Edit...	Edit...
sca/import/AccountingServicesImport	Module	Use default (META-INF/wsdl/AccountingServicesImpor	Edit...	Edit...
sca/import/DependentsServices	Module	Use default (META-INF/wsdl/DependentsServices_Dep	Edit...	Edit...

Figure 9-83   Editing port information for the remote enterprise services

8. Change the host and port to that of the WebSphere Application Server hosting the remote enterprise services, in this example `https://wps3.itso:9443/` (Figure 9-84).

Note: Ensure you set the protocol to `https` and not `http`.



Enterprise Applications > ITSOWPSCplxBusApp > EJB Modules > ITSOWPSCplxBusEJB.jar > Web services client bindings > sca

Specifies a request timeout, an overridden endpoint URL, and an overridden binding namespace can be set for a port. The time seconds to wait for a request. A value of zero disables the timeout. The current endpoint and binding namespace can be overrid

Port Information for Web service sca/import/LegalServices, EJB Module

Port	Request Timeout (seconds)	Overridden Endpoint URL	Overridden Binding N
LegalServices_LegalServicesIHttpPort		https://wps3:9443/ITSOWPSCplxSrvWeb/serv	

Figure 9-84   Configuring the host for the remote enterprise services

9. Repeat the previous step for the `sca/import/AccountServicesImport`, `sca/import/AccountingServicesImport` and the `sca/import/DependentsServicesImport` services.

10. Save and synchronize the changes, and click **OK** when complete.

9.4.2 Configuring IBM HTTP Server

You need to install and configure a router in front of the presentation layer running on the business process cluster `WPSCluster01` to load balancing to the presentation logic layer.

Client requests from the browser are made to the IBM HTTP Server. The IBM HTTP Server server passes the requests to the WebSphere plug-in. The plug-in load balances to the presentation applications running in the business cluster as illustrated in Figure 9-85.

The presentation application forwards the request to the business application. Multiple requests (dependent upon the business flow) are sent from the business application to the standalone WebSphere Application Server which hosts the external Web Services application (service provider).

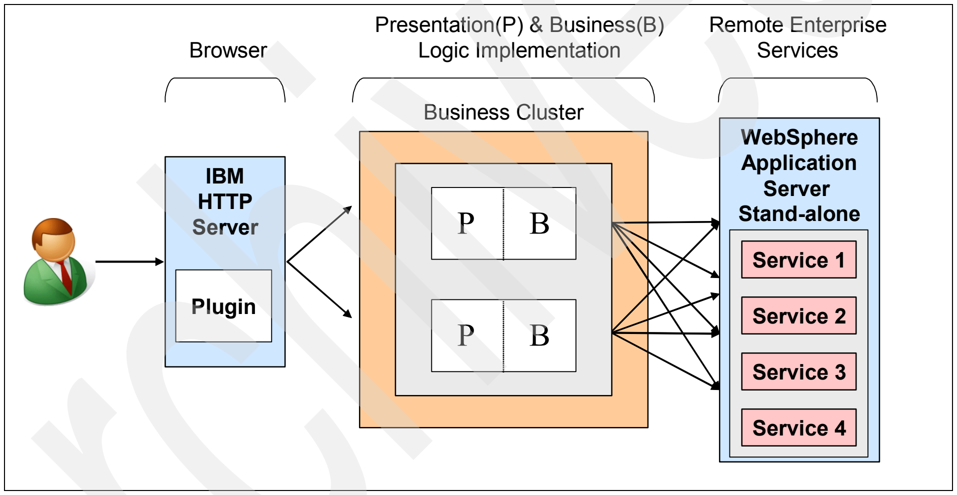

Browser
Presentation(P) & Business(B) Logic Implementation
Remote Enterprise Services
Business Cluster
IBM HTTP Server
Plugin
P
B
P
B
WebSphere Application Server Stand-alone
Service 1
Service 2
Service 3
Service 4

Figure 9-85 Load-balanced and failover-capable business cluster

It is assumed that the following products have been installed:

- IBM HTTP Server in c:\IBMIHS
- The WebSphere plug-in for IBM HTTP Server in c:\IBM\WebSphere\Plugins

Adding the Web server to the cell

To add the Web server to the cell, follow these steps:

1. Open a command prompt.
2. Change directory to c:\IBM\WebSphere\Plugins\bin.

   > **Note:** In this example, the Web server is named `webserver1`. Therefore the configuration script is named `configurewebserver1.bat`.

3. Copy the configurewebserver1.bat file to %*WAS_HOME*%\bin on any node in the cell.
4. The Windows operating system has a limitation for the file name length. To avoid the exception caused by that limitation, you need to do some additional configuration regarding the working directory of WebSphere Process Server.
   a. In Administrative Console, click **System Administration** → **Deployment Manager** → **Java and Process Management** → **Process Definition** → **Java Virtual Machine** → **Custom Properties**.
   b. Click **New** and create a new property with the attributes:
      - Name: `websphere.workspace.root`
      - Value: `C:/w`
   c. Click **OK**.
   d. Save and synchronize the configuration.
   e. Restart the deployment manager.
5. Change directory to %*WAS_HOME*%\bin and open configurewebserver1.bat in a text editor.
6. Add user ID and password credentials as shown in Example 9-2 .

Example 9-2 Modify configurewebserver1.bat to add user credentials

```
wsadmin.bat -user wsadmin -password itso4you -f
configureWebserverDefinition.jacl
...
```

7. Run configurewebserver1.bat.

This script adds the Web server into the cell and deploys the applications to the Web server.

Configuring the plug-in

Note: Before generating the plug-in to load balance to the presentation application, you must deploy this application to the `webserver1` IBM HTTP Server. This deployment can be set in the Administrative Console by navigating to **Enterprise Applications** → **ITSOWPSCplxPrsApp** → **Map modules to servers** and setting the servers to both `WPSCluster01` and `webserver1`.

To configure the plug-in, follow these steps:

1. Open the Administrative Console and login.
2. Navigate to **Servers** → **Web servers**.
3. Select **webserver1** and click **Generate Plug-in**.
4. When complete, the message text identifies the location of the generated plug-in file. In this example, it is:

   ```
   c:\IBM\WebSphere\ProcServer\profiles\Dmgr01\config\cells\WPSCell01\n
   odes\wps0.itso\servers\webserver1\plugin-cfg.xml
   ```

5. Move the plugin-cfg.xml configuration file.

   The location of plugin-cfg.xml that WebSphere Plug-in uses is specified by the `WebSpherePluginConfig` variable within the httpd.conf configuration file of IBM HTTP Server. Therefore, you must move the generated plugin-cfg.xml to the path that is defined by this variable. In this example, the command is:

   ```
   move
   c:\IBM\WebSphere\ProcServer\profiles\Dmgr01\config\cells\WPSCell01\n
   odes\wps0.itso\servers\webserver1\plugin-cfg.xml
   c:\IBM\WebSphere\Plugins\config\webserver1\plugin-cfg.xml
   ```

9.4.3 Enabling SSL

This section details the steps that are required to enable SSL encryption. It is assumed that you have installed Global Security Kit (GSK), in C:\Program Files\ibm\gsk7\.

External access to IBM HTTP Server

In this example only SSL access is allowed to the IBM HTTP Server.

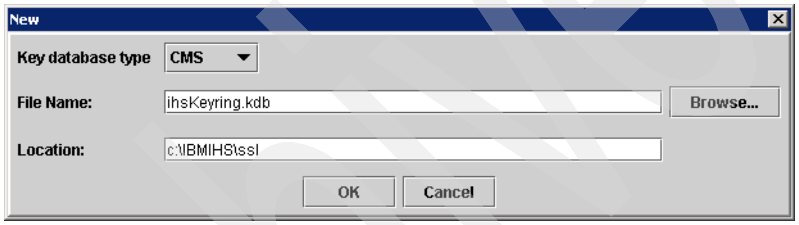
If trust store certificates are not available each web server will require a self certified certificate created. The key stores need to be created from every web server in the environment. This process will need to run on wps0 in this example.

The corresponding key store files will be saved in <*IHS_HOME*>\ssl (for example, C:\IBMIHS\ssl).

Creating the PKI artefacts needed for IBM HTTP Server

Follow these steps:

1. Start the IBM HTTP Server Key Management Utility in a command prompt with the JAVA_HOME environment variable set.
2. Run the ikeyman utility C:\Program Files\ibm\gsk7\bin\gsk7ikm.exe.
3. Select **Key Database File** → **New**.
4. Specify the following settings (Figure 9-86):
   - Key Database Type: **CMS**
   - File Name: `ihsKeyring.kdb`
   - Location: <*IHS_HOME*>\ssl (create this directory manually before selecting **OK**).

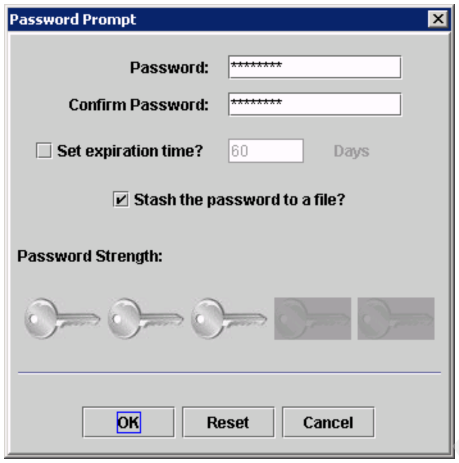


Key database type: CMS
File Name: ihsKeyring.kdb
Location: c:\IBMIHS\ssl

Figure 9-86 Setting the keystore filename

5. Select **OK** and specify the following settings when prompted:
   - Password: `itso4you`
   - Check **Stash the password to a file**

6. Select **OK** to complete the key store creation as shown in Figure 9-87.


Password Prompt

Password: *********

Confirm Password: *********

☐ Set expiration time? 60 Days

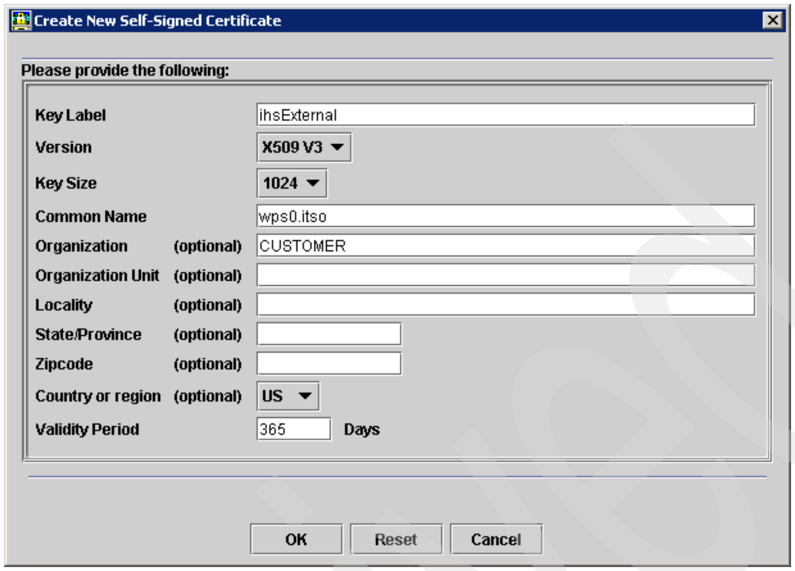
☑ Stash the password to a file?

Password Strength:

Figure 9-87 Setting keystore password

7. Select **Create** → **New Self-Signed Certificate**.
8. Specify the following parameters:
   – Key Label: `ihsExternal`
   – Organization: `CUSTOMER`
   – Validity Period: `365` Days

9. Click **OK** to create the certificate (Figure 9-88).



Create New Self-Signed Certificate

Please provide the following:

Key Label		ihsExternal
Version		X509 V3
Key Size		1024
Common Name		wps0.itso
Organization	(optional)	CUSTOMER
Organization Unit	(optional)	
Locality	(optional)	
State/Province	(optional)	
Zipcode	(optional)	
Country or region	(optional)	US
Validity Period		365 Days

Figure 9-88 Creating the certificate

10. Exit the IBM Key Management utility.

Reconfiguring httpd.conf for the new SSL certificate

You need to add the following entries to the httpd.conf for each Web server to enable SSL encryption with the keystore just created:

1. Edit the file C:\IBMIHS\conf\httpd.conf on `wps0`:
   a. Remove the line:
   ```
   Listen 80
   ```
   b. Append the following lines to the end of the file:
   ```
   Listen 443
   LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
   <VirtualHost wps0.itso:443>
   SSLEnable
   Keyfile ssl/ihsKeyring.kdb
   </VirtualHost>
   ```
   c. Save and close the file.

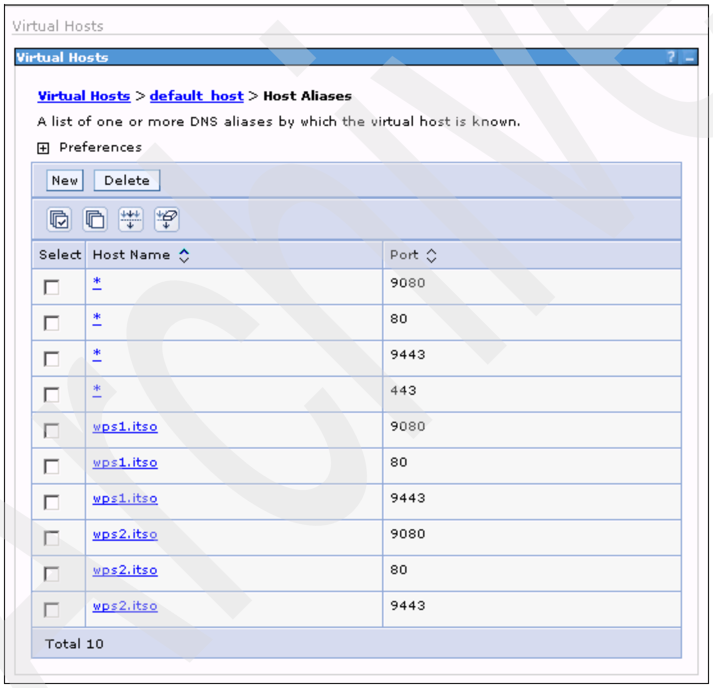
2. Restart the IBM HTTP Server server with the following commands:

```
cd c:\IBMIHS\bin
apache.exe -k stop
apache.exe -k start
```

Modifying virtual hosts

You need to add a virtual host entry for the SSL default port, `443`, for the application server cluster running the presentation logic to respond to requests. Follow these steps:

1. Open the cell Administrative Console and log in.
2. Navigate to **Environment** → **Virtual Hosts**.
3. Click the **default_host** virtual host.
4. Under Additional Properties, click **Host Aliases** (Figure 9-89).



Virtual Hosts

Virtual Hosts

Virtual Hosts > default_host > Host Aliases

A list of one or more DNS aliases by which the virtual host is known.

⊞ Preferences

New Delete

Select	Host Name	Port
☐	*	9080
☐	*	80
☐	*	9443
☐	*	443
☐	wps1.itso	9080
☐	wps1.itso	80
☐	wps1.itso	9443
☐	wps2.itso	9080
☐	wps2.itso	80
☐	wps2.itso	9443

Total 10

Figure 9-89 Host aliases

5. Click **New**.
6. Set the **Host Name** to *.
7. Set the **Port** to `443`.

Note: The Host Name entry can be specific to the application server host name. In this case, you must make an entry for every individual application server within the cluster, for example, wps1.itso, wps2.itso, and so forth.

8. Click **OK**.
9. Save and synchronize the changes. Click **OK** when complete.
10. Navigate to **Servers** → **Clusters**.
11. Select **WPSCluster01**.
12. Click **Stop**.
13. When complete, select **WPSCluster01** and click **Start**.

Note: You must restart the application server cluster, in this case `WPSCluster01`, for the virtual hosts directives to take effect.

14. Navigate to **Servers** → **Web servers**.
15. Select **webserver1** and click **Generate Plug-in**.
16. When complete, the message text identifies the location of the generated plug-in file. In this example, it is:

```
c:\IBM\WebSphere\ProcServer\profiles\Dmgr01\config\cells\WPSCell01\n
odes\wps0.itso\servers\webserver1\plugin-cfg.xml
```

17. Move the plugin-cfg.xml configuration file.

    The location of plugin-cfg.xml that WebSphere Plug-in uses is specified by the `WebSpherePluginConfig` variable within the httpd.conf configuration file of IBM Http Server . Therefore, you must move the generated plugin-cfg.xml to the path that is defined by this variable. In this example, the command is:

```
move
c:\IBM\WebSphere\ProcServer\profiles\Dmgr01\config\cells\WPSCell01\n
odes\wps0.itso\servers\webserver1\plugin-cfg.xml
c:\IBM\WebSphere\Plugins\config\webserver1\plugin-cfg.xml
```

Reconfiguring plugin-cfg.xml

You need to modify the plugin-cfg.xml file, which is located in the C:\IBM\WebSphere\Plugins\config\webserver1\ directory, to point to the keyring and stash files on the WebSphere server if the Web server is not on a WebSphere node. If this is the case, you need to generate or copy these files from a WebSphere node.

In this example, we use the default keyring and stash files that are provided with WebSphere. Because the IBM HTTP Server is on a WebSphere node the automatically generated `plugin-cfg.xml` file points to the correct files.

For each setting of `keyring` in the plugin-cfg.xml file set the `Value` to:

```
C:\IBM\WebSphere\ProcServer\etc\plugin-key.kbd
```

For each setting of `stashfile` in the plugin-cfg.xml file set the `Value` to:

```
C:\IBM\WebSphere\ProcServer\etc\plugin-key.sth
```

The `<VirtualHostGroup Name="default_host">` must be listening on port 443.

Restarting IBM HTTP Server

Restart IBM HTTP Server with the following commands:

```
cd c:\IBMIHS\bin
Apache.exe -k stop
Apache.exe -k start
```

9.4.4 Configuring the remote enterprise service

To configure the remote enterprise service:

1. Open the wps3.itso Administrative Console and log in.
2. Navigate to **Environment** → **Virtual Hosts**.
3. Click the **default_host** virtual host.
4. Under Additional Properties, click **Host Aliases**.
5. Select ***:80** and ***:9080** host aliases.
6. Click **Delete**.

> **Note:** This represents a remote web service that only allows SSL connections.

7. Open a command prompt, and restart the server with the following commands:

```
cd c:\IBM\WebSphere\AppServer\profiles\WASNode01\bin
stopServer.bat server1 -username wsadmin -password itso4you
startServer.bat server1 -username wsadmin -password itso4you
```

9.4.5 Testing the scenario

This section tests the environment and the application it hosts, and shows load-balancing occurring through the business flow.

Starting the environment

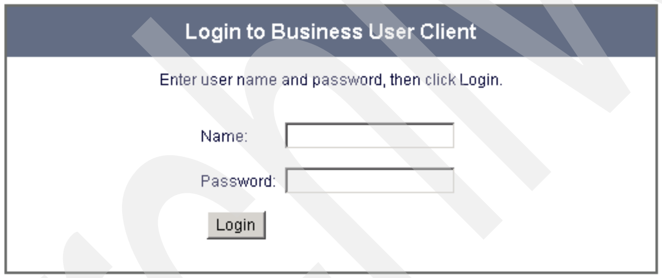
To start the environment:

1. Ensure that all layers of the environment: presentation, business and remote enterprise services, are started.
2. Ensure that all applications are running in these layers.

Business testing

Testing the application involves accessing the JSP front end from the IBM HTTP Server server. Follow these steps:

1. Open a browser to `https://wps0.itso/ITSOWPSCplxPrsWeb/`. Because security is enabled, the initial panel is a login prompt, as shown in Figure 9-90.



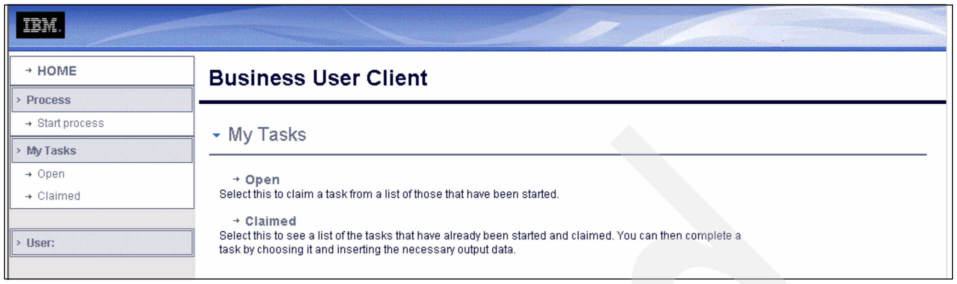
Login to Business User Client

Enter user name and password, then click Login.

Figure 9-90 Accessing the business

2. Log in by entering a name of `wsadmin` and a password of `itso4you`. This name and password is authenticated using the LDAP server.

3. You are presented with the window shown in Figure 9-91.



Business User Client

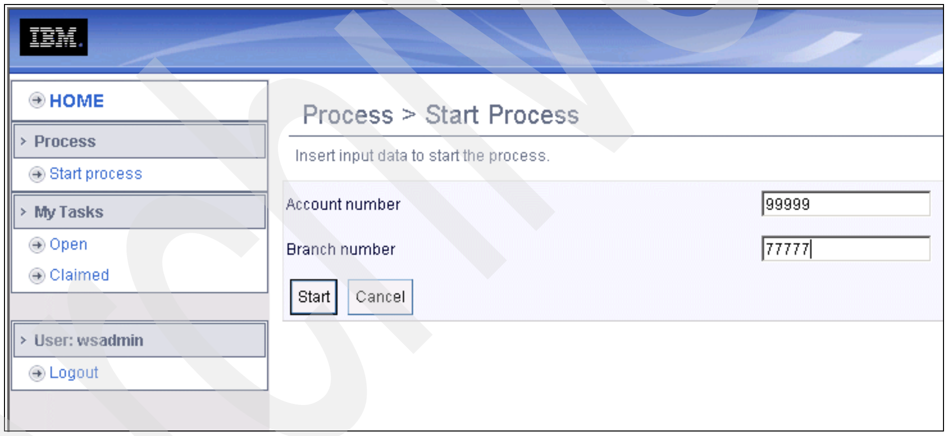
My Tasks

Open
Select this to claim a task from a list of those that have been started.

Claimed
Select this to see a list of the tasks that have already been started and claimed. You can then complete a task by choosing it and inserting the necessary output data.

Figure 9-91 Accessing the presentation layer JSP through IBM HTTP Server

4. Under the left-hand menu, click **Start process**.
5. Select **complexAccountClosure**.
6. Enter an arbitrary account number and branch number, as shown in Figure 9-92, then click **Start.**



Process > Start Process

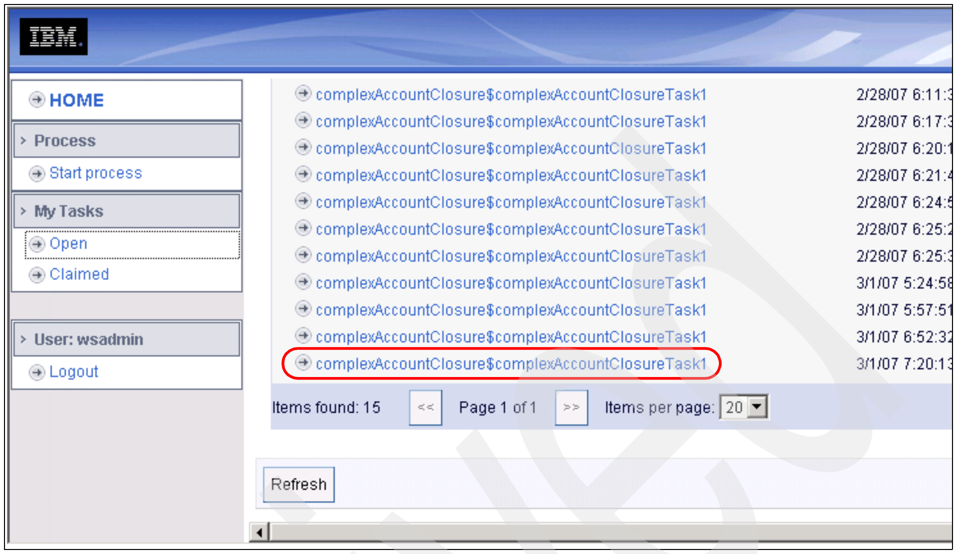
Insert input data to start the process.

Account number	99999
Branch number	77777

Figure 9-92 Starting the account closure process

7. A message displays stating that the process has started correctly.
8. Under the left-hand menu, click **Open**.

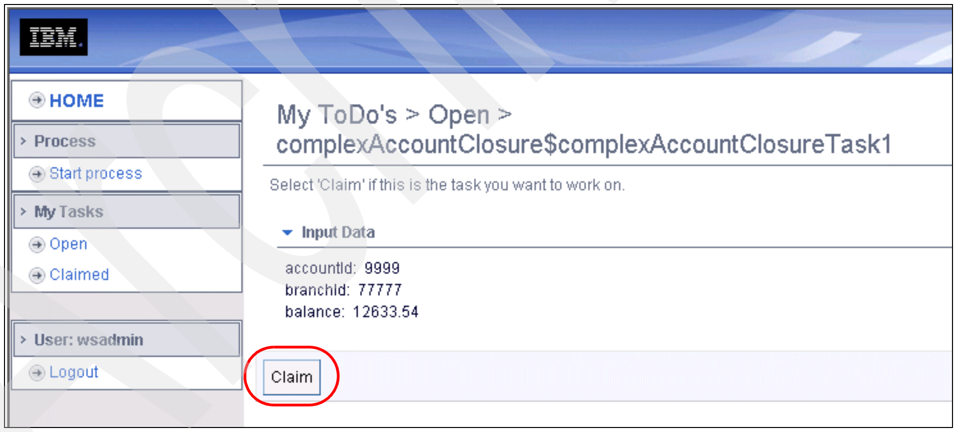
9. Scroll down to select the open task just created, as shown in Figure 9-93.



```
complexAccountClosure$complexAccountClosureTask1    2/28/07 6:11:3
complexAccountClosure$complexAccountClosureTask1    2/28/07 6:17:3
complexAccountClosure$complexAccountClosureTask1    2/28/07 6:20:1
complexAccountClosure$complexAccountClosureTask1    2/28/07 6:21:4
complexAccountClosure$complexAccountClosureTask1    2/28/07 6:24:5
complexAccountClosure$complexAccountClosureTask1    2/28/07 6:25:2
complexAccountClosure$complexAccountClosureTask1    2/28/07 6:25:3
complexAccountClosure$complexAccountClosureTask1    3/1/07 5:24:58
complexAccountClosure$complexAccountClosureTask1    3/1/07 5:57:51
complexAccountClosure$complexAccountClosureTask1    3/1/07 6:52:32
complexAccountClosure$complexAccountClosureTask1    3/1/07 7:20:13
Items found: 15   <<   Page 1 of 1   >>   Items per page: 20
Refresh
```

Figure 9-93   Opening a task

10.Click **Claim**, as shown in Figure 9-94.



My ToDo's > Open > complexAccountClosure$complexAccountClosureTask1

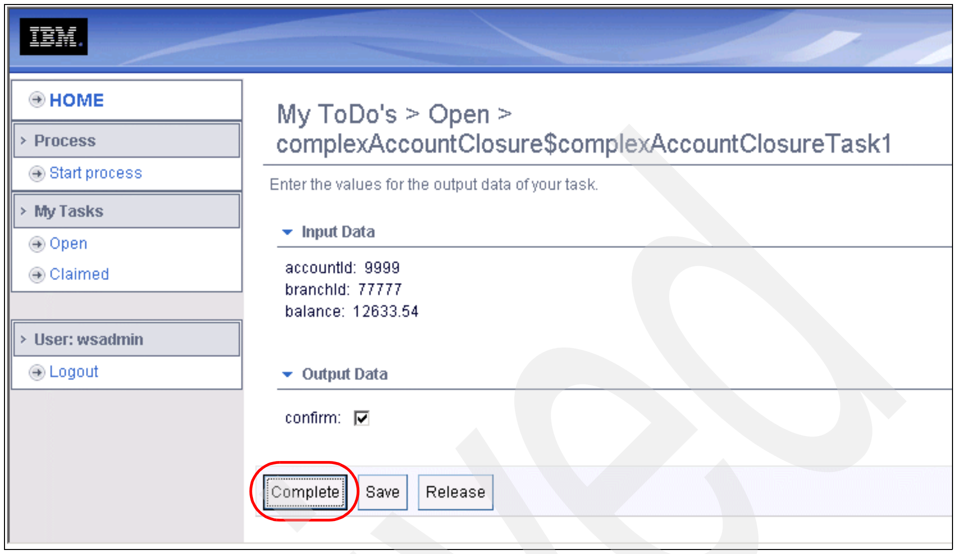
Select 'Claim' if this is the task you want to work on.

Input Data

accountId: 9999
branchId: 77777
balance: 12633.54

Figure 9-94   Claiming a task

11.Select **confirm**, and click **Complete**, as shown in Figure 9-95.



My ToDo's > Open > complexAccountClosure$complexAccountClosureTask1

Enter the values for the output data of your task.

Input Data

accountId: 9999
branchId: 77777
balance: 12633.54

Output Data

confirm: ✓

Figure 9-95 Confirming an account closure request

This end-to-end process has sent a request to close an account through the following path:

1. The client browser sent a request to IBM HTTP Server over HTTPS.
2. The associated WebSphere Plug in passed this request to one of the WebSphere Process Server cluster members running the presentation application over HTTPS.
3. The presentation application called the local business application that is running on the same server, with RMI/IIOP.
4. The business application EJB called the remote enterprise service using SOAP/HTTPS.

Part 4

Appendixes

A

Additional material

This book refers to additional material that you can download from the Internet as described in this appendix.

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 to:

ftp://www.redbooks.ibm.com/redbooks/SG247413

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, SG24-7413.

Using the Web material

The additional Web material that accompanies this book includes the following files:

File name	*Description*
SG247413.zip	Zipped code samples

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zipped file into this folder.

The ZIP file includes the following directory structure:

- `\EAR files`

  Include the enterprise applications for deployment to the WebSphere ESB and WebSphere Process Server topologies that we describe in this book.

- `\ProjectInterchange`

  Includes the project files that are used in WebSphere Integration Developer V6.0.2 to build the sample applications that we use in this book.

- `\LDAP`

  Includes an LDIF file that stores an LDAP structure that is used by a sample application in this book.

Naming convention used for the EAR files

The EAR files supplied in the `\EAR files` directory use the following naming convention:

```
ITSO[WPS|ESB][Smpl|Cplx][Prs|Bus|Srv]App.ear
```

Where:

- `WPS` or `ESB` is the name of the product (WebSphere Process Server or WebSphere ESB) for which this application has been developed.
- `Smpl` or `Cplx` indicates whether this is the simple or complex scenario.
- `Prs`, `Bus`, or `Srv` indicates which layer this application is intended to be deployed to: the Presentation layer, Business layer, or Enterprise Service layer.

For example, the enterprise application to be deployed to the presentation layer of the WebSphere ESB simple scenario is called `ITSOESBSmplPrsApp.ear`.

B

Product installation

This appendix describes the installation of the following products that we use throughout this book:

- IBM WebSphere Process Server V6.0.2 (and WebSphere ESB V6.0.2)
- IBM DB2 Universal Database Enterprise Server V8.2
- IBM HTTP Server V6
- IBM Web server plug-ins
- Global Security Kit (GSK)

WebSphere Process Server installation

Note: This book assumes the use of WebSphere Process Server V6.0.2 or WebSphere ESB V6.0.2. WebSphere ESB is a subset of WebSphere Process Server. Therefore, this section applies equally to WebSphere Process Server and WebSphere ESB.

You can either install a new WebSphere Process Server, or upgrade an existing configuration:

- You can install a new WebSphere Process Server v6.0.2 using either the GUI Installation Wizard or you can install it silently with a response file. For more information see:

  http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp?topic=/com.ibm.wsps.602.ins.doc/welcome_wps_ins.html

- If you have a prior version installed of WebSphere Process Server (such as WebSphere Process Server v6.0.1), you can create a WebSphere Process Server v6.0.2 environment by applying Refresh Pack 2 for WebSphere Process Server Version 6.0. This refresh pack is available from the IBM support Web site for WebSphere Process Server:

  http://www.ibm.com/software/integration/wps/support/

For step-by-step information about installing WebSphere Process Server, refer to the WebSphere Process Server Information Center:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp?topic=/com.ibm.wsps.602.ins.doc/welcome_wps_ins.html

When performing the installation, consider the following information:

- Install the full product of WebSphere Process Server v6.0.2.0.

  Note: The Windows operating system has a limitation for the length of the file path. You are recommended to use a short value for *WPS_HOME* to avoid encountering this problem.

  - Perform a custom installation. This will give the option of whether to launch the Profile Wizard.
  - When the installation is complete, the Installation Wizard will ask if you wish to launch the Profile Wizard. Do not create any profiles at this time. Each chapter in this book specifies when you need to create a profile.

- Apply any necessary maintenance to the product.
  - After the installation of the product, it is always a good practice to apply the required maintenance for the product. The information of the maintenance is published on the IBM support Web site for WebSphere Process Server. You can find more information here:

    `http://www.ibm.com/products/finder/us/finders?Ne=5000000&finderN=1000100&trac=SU1&pg=ddfinder&collectionN=0&sid=494345711170684003038&cc=us&lc=en&oldC1=5000832&tmpl=%2Fproducts%2Ffinder%2Fus%2Fen%2Ffinders&C1=5000832&C2=5429715`
- Validate the installation

  Upon installation, one of the following messages can be found in *WAS_HOME*\logs\wbi\log.txt.

  - `INSTCONFSUCCESS`

    Indicates the installation was successful
  - `INSTCONFPARTIALSUCCESS`

    Indicates the installation was partly successful. Some installation actions failed but can be retried.
  - `INSTCONFFAILED`

    Indicates installation was not successful. Recovery is not possible.

  If the result is `INSTCONFPARTIALSUCCESS` or `INSTCONFFAILED`, the installation did not complete successfully, and you will need to diagnose the problem. You can find more information about the troubleshooting installation problems at:

  `http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wsps.602.ins.doc/doc/tins_trouble.html`

  Example 9-3 shows a successful installation.

Example 9-3 Successful installation

```
com.ibm.ws.install.ni.ismp.actions.ISMPLogSuccessMessageAction, msg1,
INSTCONFSUCCESS
```

DB2 Universal Database Enterprise Server installation

This book assumes the use of DB2 Universal Database Enterprise Server V8.2 with Fix Pack 14 applied. For information about installing DB2, refer to the DB2 Information Center:

> `http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/core/c0008880.htm`

DB2 is used with WebSphere Process Server and WebSphere ESB. However the DB2 server does not need to reside on the same machine as these products. A DB2 Universal database driver is included in the WebSphere Process Server (or WebSphere ESB) installation at WPS_HOME\universalDriver_wbi. This driver may be used for any resources that are configured to utilize DB2.

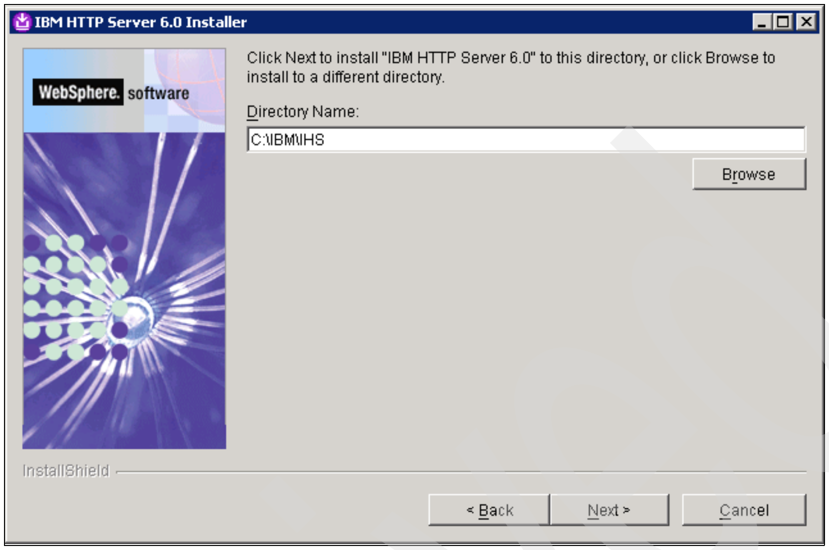
Each chapter in this book describes how to create specific DB2 databases that are required for each topology.

IBM HTTP Server installation

The IBM HTTP Server installation is shipped with the WebSphere Process Server and WebSphere ESB installation image under the `IHS` folder. Installation of IBM HTTP Server is straightforward. Perform the following:

1. Launch the Install Shield Wizard of IBM HTTP Server 6.0 (on Windows environments launch **install.exe**).
2. On the Welcome panel, click **Next.**
3. On the Software License Agreement panel, accept the terms, click **Next.**

4. Set the IBM HTTP Server installation directory, then click **Next** (Figure B-1).



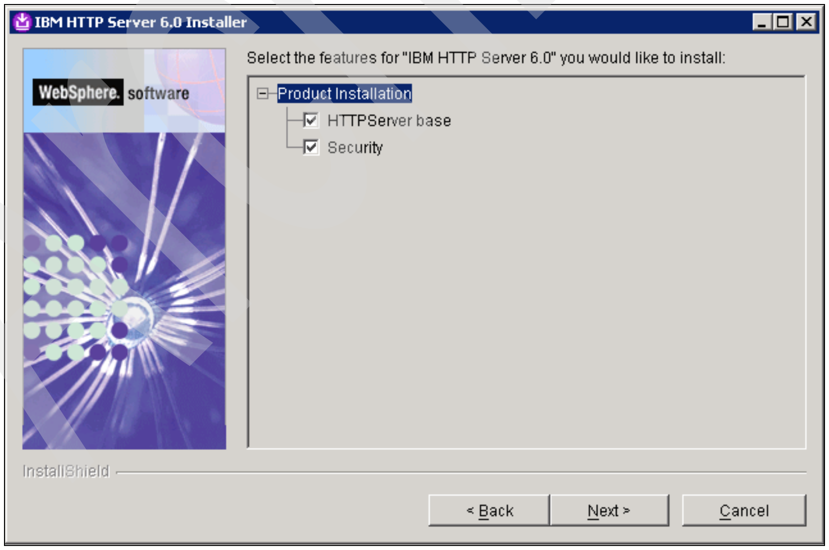
Click Next to install "IBM HTTP Server 6.0" to this directory, or click Browse to install to a different directory.

Directory Name:

C:\IBM\IHS

Figure B-1 Select the IBM HTTP Server installation directory

5. Select **Custom** installation, then click **Next.**
6. On the Select features panel, accept the default values, then click **Next** (Figure B-2).

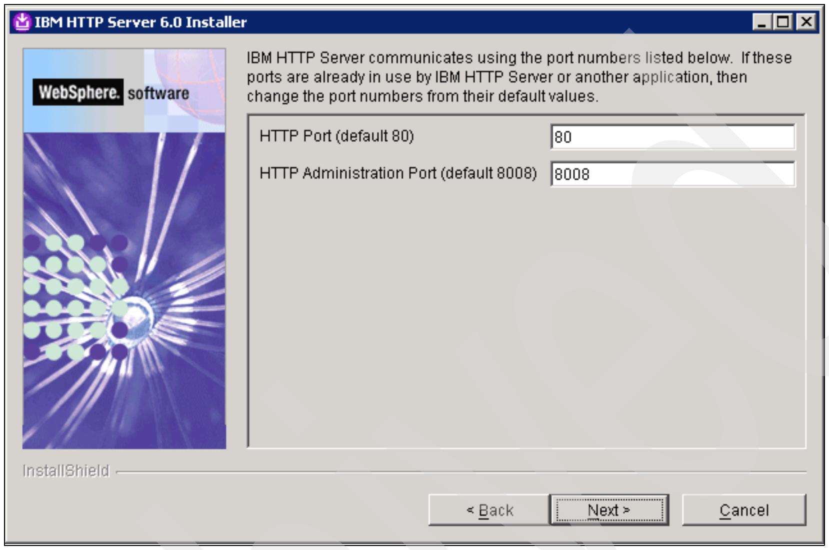


Select the features for "IBM HTTP Server 6.0" you would like to install:

Product Installation
- [✓] HTTPServer base
- [✓] Security

Figure B-2 Select the installation features

7. Specify the TCP/IP ports to use. You must ensure that the assigned ports are available, and are not in use by other applications. Make a note of the HTTP port you use as you will need this value when you configure the HTTP server plug-ins. Click **Next** (Figure B-3).

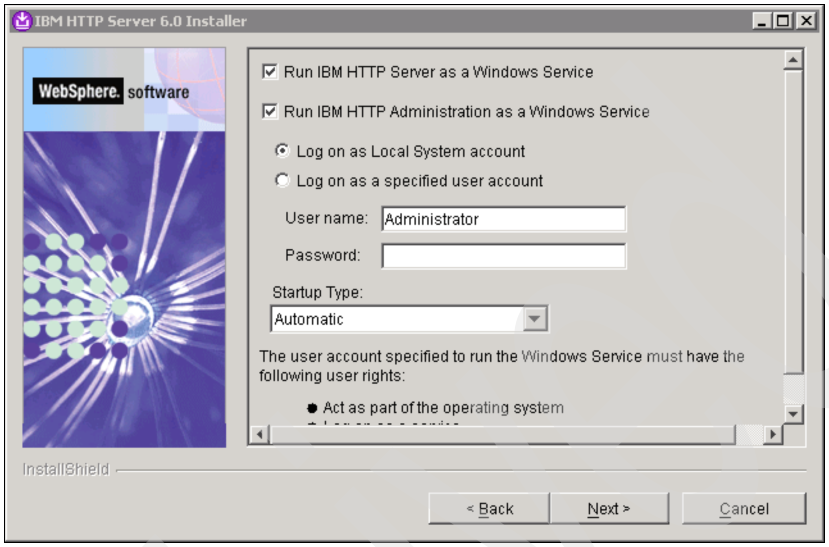


IBM HTTP Server communicates using the port numbers listed below. If these ports are already in use by IBM HTTP Server or another application, then change the port numbers from their default values.

HTTP Port (default 80)	80
HTTP Administration Port (default 8008)	8008

Figure B-3 Set the value for the default port and the administration port

8. On Windows platforms, decide whether to run IBM HTTP Server as a Windows service. We elected to define a service, so selected **Log on as Local System account**, and clicked **Next** (Figure B-4).



☑ Run IBM HTTP Server as a Windows Service

☑ Run IBM HTTP Administration as a Windows Service

(●) Log on as Local System account

( ) Log on as a specified user account

User name: Administrator

Password:

Startup Type: Automatic

The user account specified to run the Windows Service must have the following user rights:

- Act as part of the operating system

Figure B-4 Run IBM HTTP Server as a Windows service

9. On the summary panel, click **Next**. The installation begins.
10. At the last panel, clear the option of **Launch the WebSphere Application Server - Plugin install** and click **Finish**.

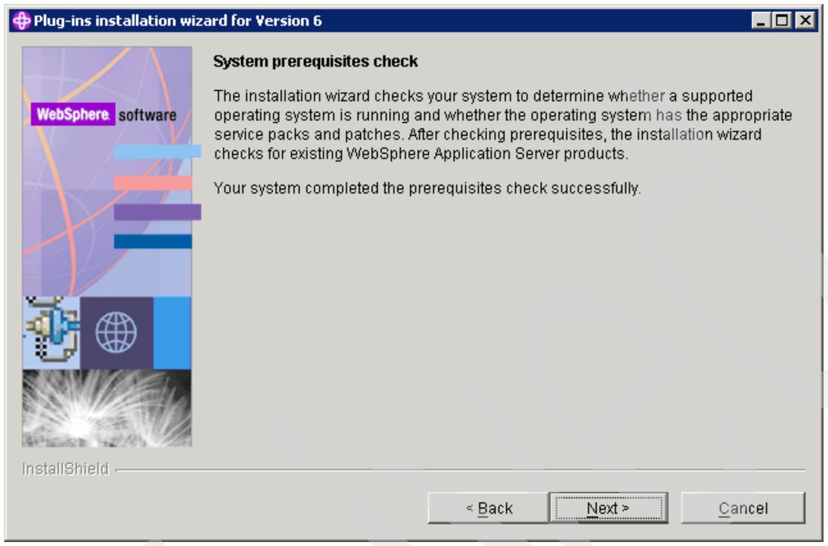
At this point, IBM HTTP Server will be installed. There is additional work to use IBM HTTP Server with WebSphere Process Server and WebSphere ESB (such as editing the httpd.conf file and running a generated script to create a Web server). We describe the steps to perform these tasks in each of the relevant chapters in this book.

IBM Web server plug-ins installation

Web server plug-ins for WebSphere Application Server are shipped with the WebSphere Process Server and WebSphere ESB installation image under the `plugin` folder. Follow these steps:

1. Launch the Install Shield Wizard for Web server plug-ins (on Windows environments launch **install.exe**).
2. On the Welcome panel, clear **Installation roadmap** and click **Next**.
3. On the Software License Agreement panel, accept the terms, click **Next.**

4. On the System prerequisites check panel, the system checks your operating system, and whether a WebSphere Application Server based product is installed. If this completes successfully, click **Next** (Figure B-5).



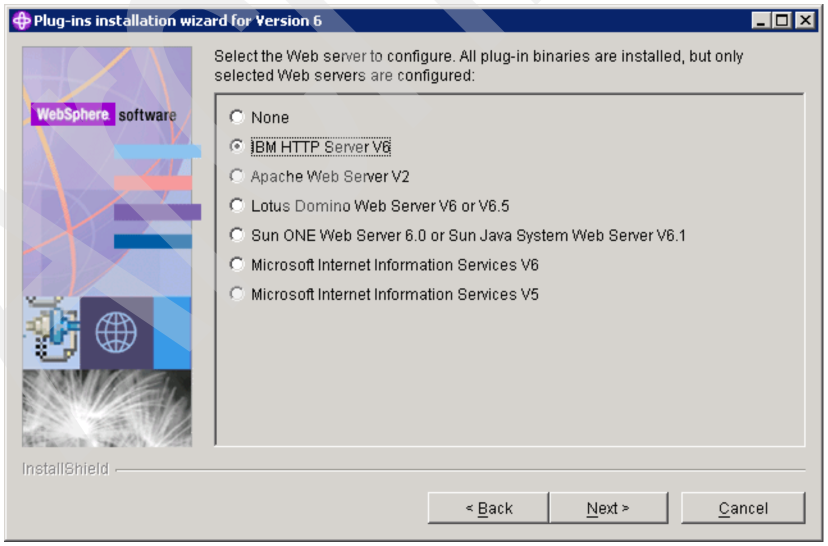
System prerequisites check

The installation wizard checks your system to determine whether a supported operating system is running and whether the operating system has the appropriate service packs and patches. After checking prerequisites, the installation wizard checks for existing WebSphere Application Server products.

Your system completed the prerequisites check successfully.

Figure B-5 System prerequisite check

5. You will be asked to select a Web server to configure. Select **IBM HTTP Server V6** and click **Next** (Figure B-6).



Select the Web server to configure. All plug-in binaries are installed, but only selected Web servers are configured:

- None
- [✓ IBM HTTP Server V6]
- Apache Web Server V2
- Lotus Domino Web Server V6 or V6.5
- Sun ONE Web Server 6.0 or Sun Java System Web Server V6.1
- Microsoft Internet Information Services V6
- Microsoft Internet Information Services V5

Figure B-6 Select Web server to configure

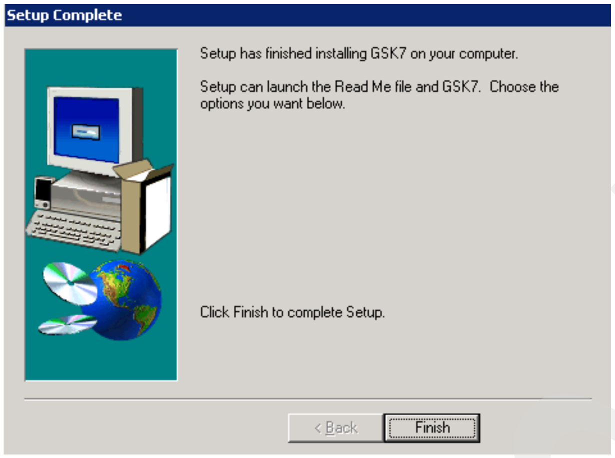
6. Set the installation scenario to **Web server machine (remote)**. Click **Next**.
7. Specify an installation directory and click **Next**.
8. Locate the httpd.conf file from your IBM HTTP Server installed, that you installed in “IBM HTTP Server installation” on page 432. The httpd.conf file is located in the `conf` directory of the IBM HTTP Server installation. Click **Next**.
9. Lave the unique Web server definition name as default, and click **Next**.
10. Accept the default for the location of the plugin-cfg.xml file and click **Next**.
11. Specify the host name of the application server. This will be specific to your scenario. Click **Next** and the **Next** again.
12. On the summary panel, click **Next** to begin the installation.
13. When the installation completes, click **Next** and then **Finish**.

Global Security Kit installation

Global Security Kit (GSK) is shipped with the WebSphere Process Server and WebSphere ESB installation image under the `GSKit` folder. Follow these steps:

1. Launch the Install Shield Wizard. From a command prompt navigate to the `GSKit` folder and issue the command:

   ```
   setup.exe gsk7bas
   ```
2. On the Welcome panel, click **Next**.
3. Specify a directory to install GSK. Click **Next** and the installation begins.
4. When the installation completes, you are presented with the panel shown in Figure B-7. Click **Finish**.


Setup Complete

Setup has finished installing GSK7 on your computer.

Setup can launch the Read Me file and GSK7. Choose the options you want below.

Click Finish to complete Setup.

Figure B-7 Setup complete

C

IBM Tivoli Directory Server (LDAP)

An LDAP server is used to enable security within WebSphere Application Server and related products. WebSphere Application Server related products offer three security enablement options:

- Local OS
- LDAP
- Custom registry

Local OS is rarely used in production topologies as it does not scale. For example you cannot use Local OS across two separate servers as the security tokens used include both the hostname and the OS user in their creation so host1:user1 is not the same as host2:user1.

The Custom registry option is viable, but increases complexity. The most common security enablement mechanism is an LDAP server of some kind.

This appendix describes how to install IBM Tivoli Directory Server V6.0 (ITDS) and configure an LDAP directory for use with WebSphere Application Server based products.

Installing Tivoli Directory Server

Before starting an LDAP installation on UNIX it is necessary to switch to an X session. Typically, this is achieved using the `startx` command as the `root` user.

By default a Linux installation creates a new user, `idsldap`, with shell `/bin/ksh` and `.profile` configuration to include the path and environment variable settings for LDAP usage. The default for DB2 is the bash shell (using `.bashrc`). When this user needs to run DB2 commands, it is necessary to change the shell to bash by typing `bash` at the `ksh` shell prompt.

By default, the DB2 instance for LDAP (the idsldap instance) will be listening for connections on port `3700`. The following example installs ITDS V6.0 onto a Red Hat EL 4 Update 2 server. The DB2 supporting database is already installed locally.

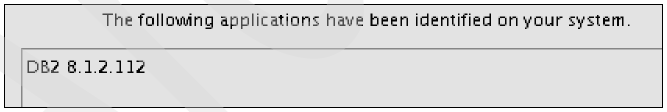
1. Insert the DVD with the installation media on it. As the `root` user, mount the drive and copy the files (or extract them) to /opt/ldapsource:

   ```
   mount /dev/dvd
   mkdir /opt/ldapsource
   cp -r /media/cdrecorder/* /opt/ldapsource
   cd /opt/ldapsource
   ```

2. Start the installation from /opt/ldapsource by running the setup command.

   ```
   ./setup
   ```

3. Select the appropriate language and click **OK**.
4. Select **Next** at the welcome panel, **Accept** the license, and select **Next** again.
5. The installation should detect the DB2 product as already installed, select **Next** at the panel as shown in Figure C-1.

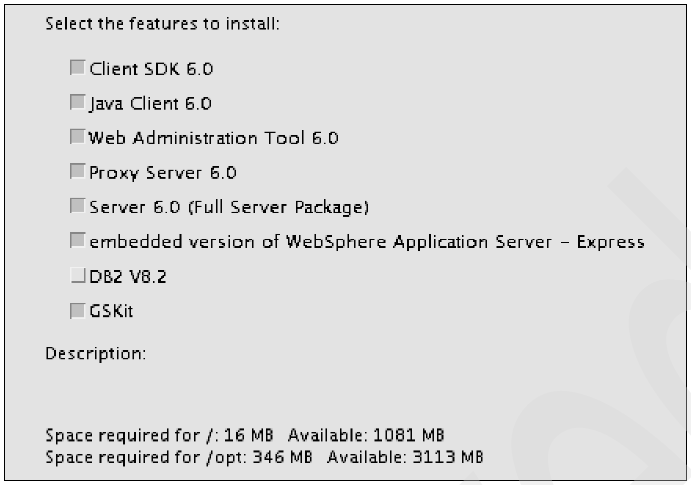

The following applications have been identified on your system.

DB2 8.1.2.112

Figure C-1 Preinstalled software

   The products shown in Figure C-2 are installed. If the Web Admin Tool and WebSphere Application Server Express are not needed, this part of the product can be removed as there are other mechanisms for performing LDAP administration.

6. Select **Next** when the correct features are selected.


Select the features to install:

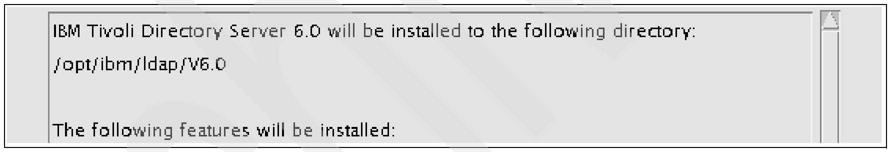
- Client SDK 6.0
- Java Client 6.0
- Web Administration Tool 6.0
- Proxy Server 6.0
- Server 6.0 (Full Server Package)
- embedded version of WebSphere Application Server – Express
- DB2 V8.2
- GSKit

Description:

Space required for /: 16 MB   Available: 1081 MB
Space required for /opt: 346 MB   Available: 3113 MB

Figure C-2 Components for installation

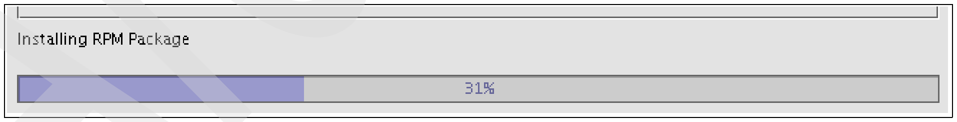
Note the location of the installation as shown in Figure C-3. The location of the installation is under the /opt file system, which was previously created for product installations.


IBM Tivoli Directory Server 6.0 will be installed to the following directory:
/opt/ibm/ldap/V6.0

The following features will be installed:

Figure C-3 Installation directory

7. Select **Next** and the product installation begins. The status of the installation displays at the bottom of the panel, as shown in Figure C-4.


Installing RPM Package

31%

Figure C-4 Installation status bar

8. Select **Yes** and **Next** if prompted about existing links, this occurs if you have previously uninstalled the product but not tidied up or there are existing LDAP components installed (by default with this install of Red Hat).

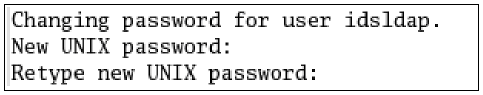
User and Group modifications

After the installation is complete and before creating a new directory instance set the password for the newly created idsldap user:

1. Log in as `root` and issue the following command:

   ```
   passwd idsldap
   ```

   You must enter the new password twice. Set the new password to `itso4you`. Refer to Figure C-5.



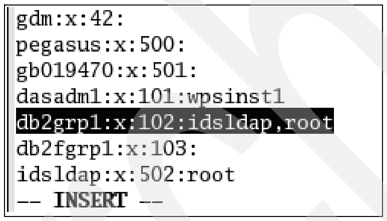
```
Changing password for user idsldap.
New UNIX password:
Retype new UNIX password:
```

Figure C-5 Setting the password for idsldap

2. Next, you add the users `root` and `idsldap` to the group `db2grp1`. Issue the following command to configure the groups:

   ```
   vi /etc/group
   ```

3. Scroll down and add `root,idsldap` to the entry for `db2grp1` as shown in Figure C-6:



```
gdm:x:42:
pegasus:x:500:
gb019470:x:501:
dasadm1:x:101:wpsinst1
db2grp1:x:102:idsldap,root
db2fgrp1:x:103:
idsldap:x:502:root
-- INSERT --
```

Figure C-6 /etc/group modifications

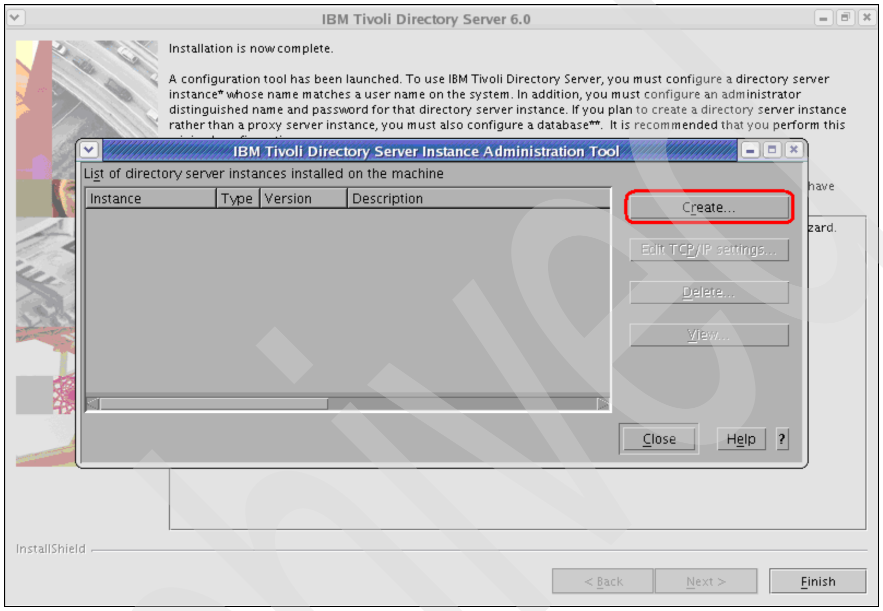
4. Save the changes in vi.
5. Finally, the /home/idsldap directory needs to be group writeable. As the **root** user issue the following command:

   ```
   chmod -R g+w /home/idsldap
   ```

Creating the LDAP Instance

When installation is complete, the instance administration tool starts. You need to create a new instance for the LDAP directory. Follow these steps:

1. Click **Create** as shown in Figure C-7.



IBM Tivoli Directory Server 6.0

Installation is now complete.

A configuration tool has been launched. To use IBM Tivoli Directory Server, you must configure a directory server instance* whose name matches a user name on the system. In addition, you must configure an administrator distinguished name and password for that directory server instance. If you plan to create a directory server instance rather than a proxy server instance, you must also configure a database**. It is recommended that you perform this

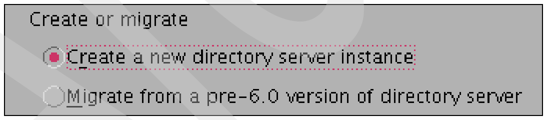
IBM Tivoli Directory Server Instance Administration Tool

List of directory server instances installed on the machine

Instance	Type	Version	Description

Figure C-7 Creating an LDAP instance

2. In the next panel, ensure that you are creating a new directory server instance as shown in Figure C-8 and select **Next**.



Create or migrate

[✓ Create a new directory server instance]

Migrate from a pre-6.0 version of directory server

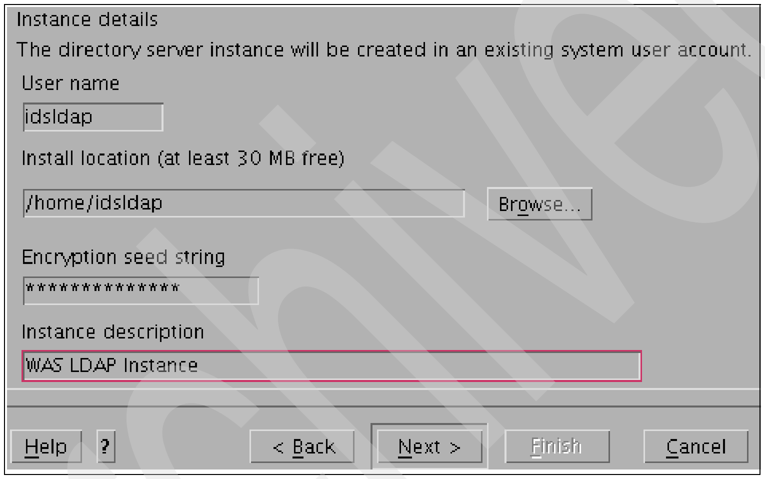
Figure C-8 Creating a new instance

3. Enter the instance details in the next panel. The details are shown in Table C-1.

Table C-1 LDAP configuration settings

User	idsldap
Install location	/home/idsldap
Encryption seed	itso4youitso Needs to be at least 12 chars long
Description	WAS LDAP Instance

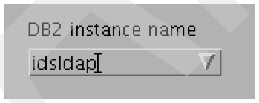
4. Select **Next** when you have added these entries (Figure C-9).



Instance details
The directory server instance will be created in an existing system user account.
User name
idsldap
Install location (at least 30 MB free)
/home/idsldap
Encryption seed string
****
Instance description
WAS LDAP Instance

Figure C-9 LDAP configuration panel

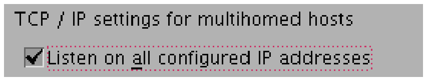
5. The next panel sets the new DB2 instance name. In this example, you create an instance called `idsldap`, as shown in Figure C-10. Select **Next** when this is correct.



DB2 instance name
idsldap

Figure C-10 DB2 database instance name

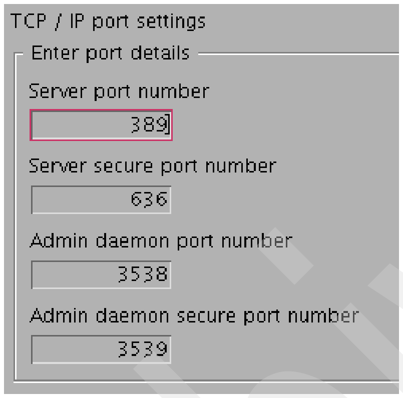
6. In the next panel, accept the default to listen on all IPs as shown in Figure C-11 and select **Next**.


TCP / IP settings for multihomed hosts

☑ Listen on all configured IP addresses

Figure C-11 Listen on all IP addresses for multihome hosts

7. Accept the default ports as shown in Figure C-12 and select **Next**.


TCP / IP port settings

Enter port details

Server port number: 389

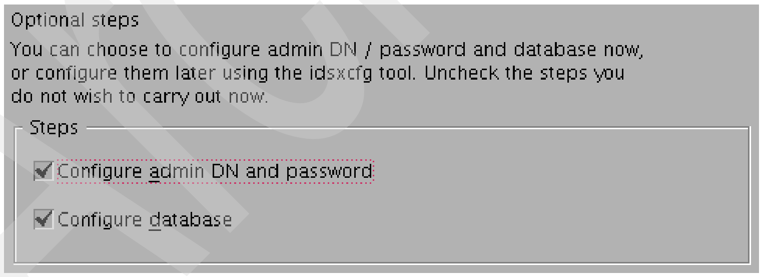
Server secure port number: 636

Admin daemon port number: 3538

Admin daemon secure port number: 3539

Figure C-12 Default port settings

8. Ensure that both optional settings for users and database are selected because you will configure these next. (Figure C-13). Click **Next** when the settings are correct.


Optional steps

You can choose to configure admin DN / password and database now, or configure them later using the idsxcfg tool. Uncheck the steps you do not wish to carry out now.

Steps

☑ Configure admin DN and password

☑ Configure database

Figure C-13 Additional steps

9. The user `cn=root` is the default, set the password (twice) to `itso4you` as shown in Figure C-14 and click **Next** to continue.

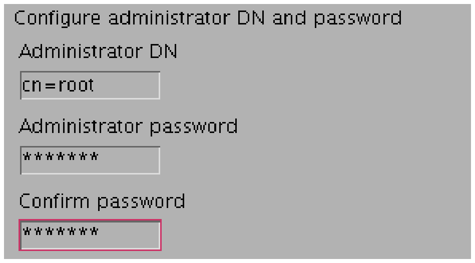

Configure administrator DN and password
Administrator DN
cn=root
Administrator password
****
Confirm password
****

Figure C-14   LDAP Admin DN

10. Use the `idsldap` user to configure the database, set the password to `itso4you` and give the database a name: `ldap`. (Figure C-15). Select **Next** to continue.

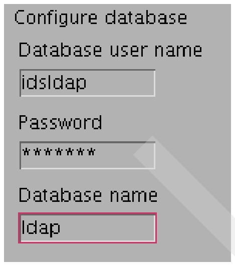
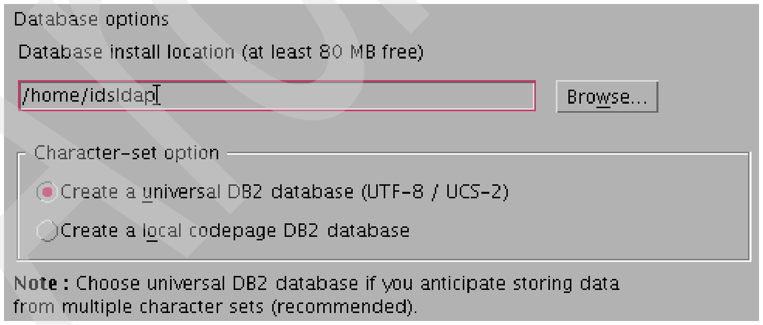

Configure database
Database user name
idsldap
Password
****
Database name
ldap

Figure C-15

11. Select the defaults on the next panel, as shown in Figure C-16 and click **Next**.

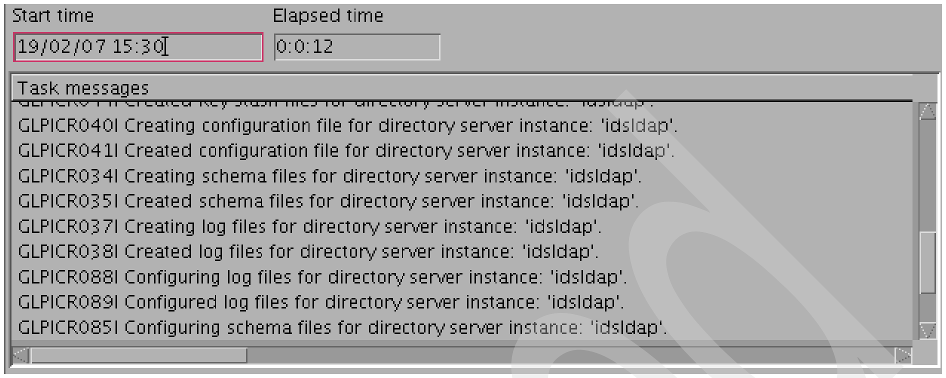

Database options
Database install location (at least 80 MB free)
/home/idsldap

Character-set option
[✓ Create a universal DB2 database (UTF-8 / UCS-2)]
Create a local codepage DB2 database
Note : Choose universal DB2 database if you anticipate storing data from multiple character sets (recommended).

Figure C-16   Database options

12. On the summary page, verify the settings and click **Finish** to create the LDAP instance and the associated DB2 instance and database.

The installation shows the current progress. Figure C-17 shows example output.



Start time: 19/02/07 15:30

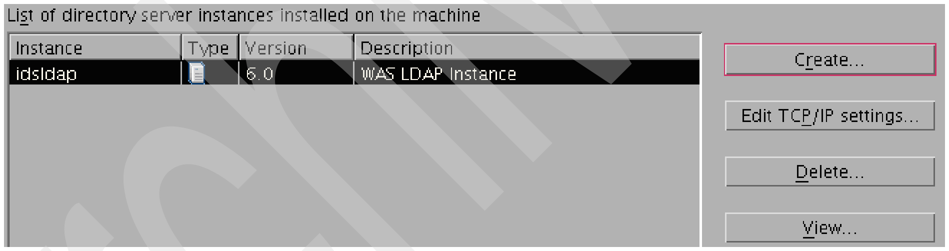
Elapsed time: 0:0:12

Task messages

```
GLPICR040I Creating configuration file for directory server instance: 'idsldap'.
GLPICR041I Created configuration file for directory server instance: 'idsldap'.
GLPICR034I Creating schema files for directory server instance: 'idsldap'.
GLPICR035I Created schema files for directory server instance: 'idsldap'.
GLPICR037I Creating log files for directory server instance: 'idsldap'.
GLPICR038I Created log files for directory server instance: 'idsldap'.
GLPICR088I Configuring log files for directory server instance: 'idsldap'.
GLPICR089I Configured log files for directory server instance: 'idsldap'.
GLPICR085I Configuring schema files for directory server instance: 'idsldap'.
```

Figure C-17 LDAP creation - output

13. When completed, click **OK** at the Task Completed message and then close the status window. The new instance is shown in the original instance admin tool window (Figure C-18).

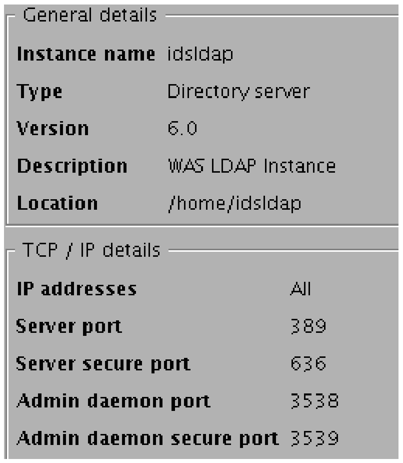


List of directory server instances installed on the machine

Instance	Type	Version	Description
idsldap		6.0	WAS LDAP Instance

Figure C-18 Newly created LDAP instance

14. Click **View** to see the details of this instance (Figure C-19).


General details

Instance name	idsldap
Type	Directory server
Version	6.0
Description	WAS LDAP Instance
Location	/home/idsldap

TCP / IP details

IP addresses	All
Server port	389
Server secure port	636
Admin daemon port	3538
Admin daemon secure port	3539

Figure C-19 Instance details

15. Click **OK** to close the view window and **Close** then **Yes** to finish the installation. Click **Finish** in the product installation window.

Note: If you need to rerun the create instance wizard, the command you need to use is **`idsxinst`**.

ITDS is now installed and running.

Verification and start up

To verity the installation:

1. As the `idsldap` user issue the following command to verify the status of the server:

```
ibmdirctl -D "cn=root" -w itso4you status
```

2. If the status is *not* running, then issue the following command when logged in as the `idsldap` user from the `ksh` shell to start it up (this might take about a minute and the DB2 `idsldap` instance must be running):

```
ibmdirctl -D "cn=root" -w itso4you start
```

You can ignore any `rc=-1` error. This error is fixed with ITDS V6.0 Fix Pack 3.

Configuring Tivoli Directory Server

Figure C-20 illustrates the directory structure to be created.

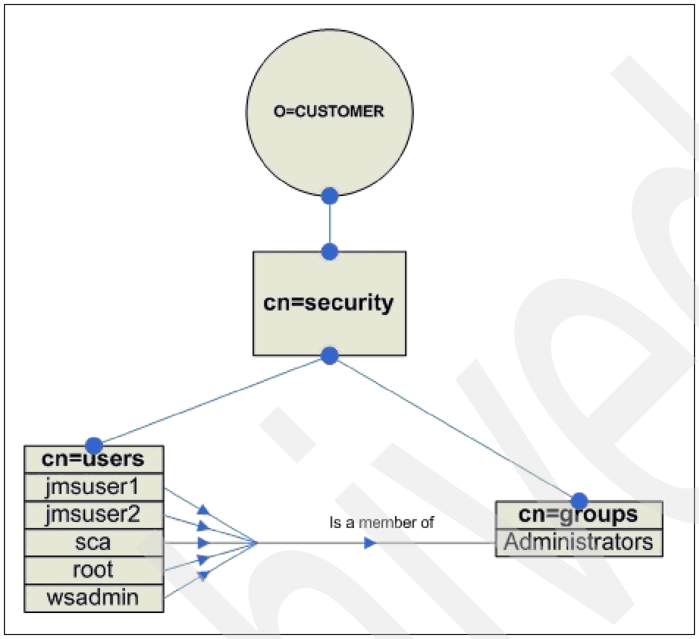


Figure C-20 LDAP directory structure

To create a starting suffix for the schema before loading an LDIF into the directory, follow these steps:

1. Log in as `root` and issue the command **`xhost +`** to allow connection to the root X windows session.
2. From the same window change to the `idsldap` user and verify the DISPLAY variable is set.

```
su - idsldap
echo $DISPLAY
```

3. To add a new suffix the server must be stopped. Again as the `idsldap` user issue the command:

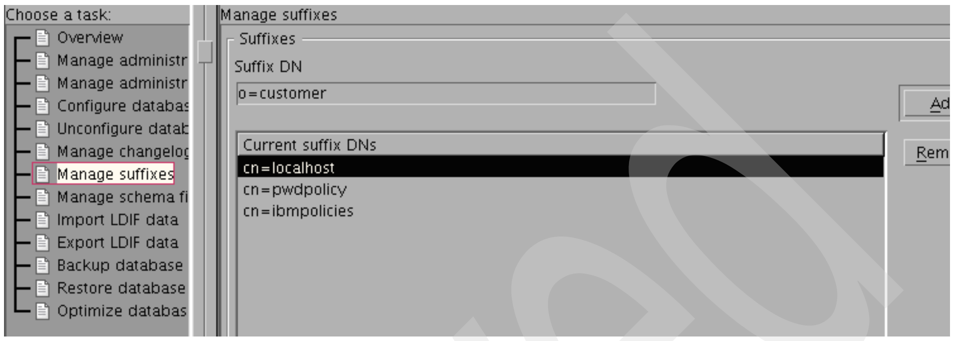
```
ibmdirctl -D "cn=root" -w itso4you stop
```

4. Start up the ITDS configuration tool

```
idsxcfg
```

5. From the tasks panel (left-hand side), click **Manage suffixes**.

6. On the right-hand side set the **Suffix DN** to be `o=customer` and click **Add**. This creates the top level under which the rest of the directory structure can be created.
7. Click **OK** to complete this change (Figure C-21).



Suffix DN

o=customer

Current suffix DNs

cn=localhost
cn=pwdpolicy
cn=ibmpolicies

Figure C-21 Adding the new suffix

8. When the suffix is created, exit the **`idsxcfg`** tool.

Creating the directory values

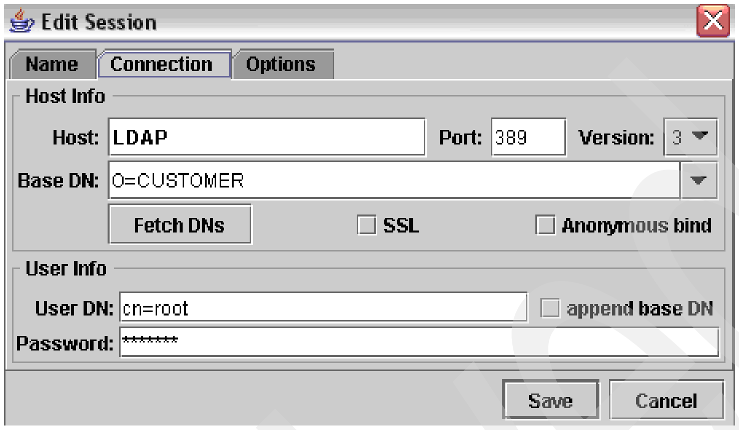
The directory structure can be created manually from the idsxcfg tool, we could import an existing LDIF with the tool, or use another tool to import an existing LDIF file. You will choose the latter. Follow these steps:

1. Download and unzip an LDAP browser. You can download an example from:

   `http://www-unix.mcs.anl.gov/~gawor/ldap/`

2. You can launch this browser by running the `lde.bat` script that comes with it. There is a requirement for JAVA to be installed and in the system path.

3. Create a connection to the LDAP server as shown in Figure C-22 (password is itso4you). Ensure that **Anonymous bind** is unchecked, save the connection, and **Open** it.


Edit Session

Host Info

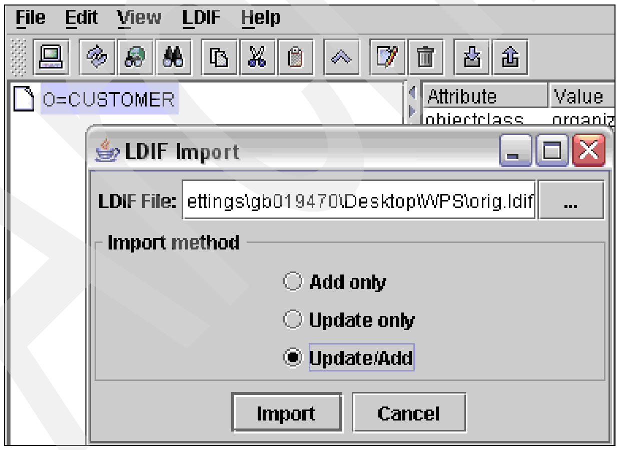
Host: LDAP
Port: 389
Version: 3
Base DN: O=CUSTOMER
☐ SSL
☐ Anonymous bind

User Info

User DN: cn=root
☐ append base DN
Password: *******

Figure C-22 Configuration for connecting to LDAP

4. Select the **O=CUSTOMER** suffix and click **LDIF** → **Import**.
5. Browse to the LDIF file `orig.ldif` which can be found in the \LDAP\ directory of the additional material supplied with this book. To obtain this additional material, refer to Appendix A, “Additional material” on page 427. Verify that **Update/Add** is selected and click **Import**. Refer to Figure C-23.


LDIF Import

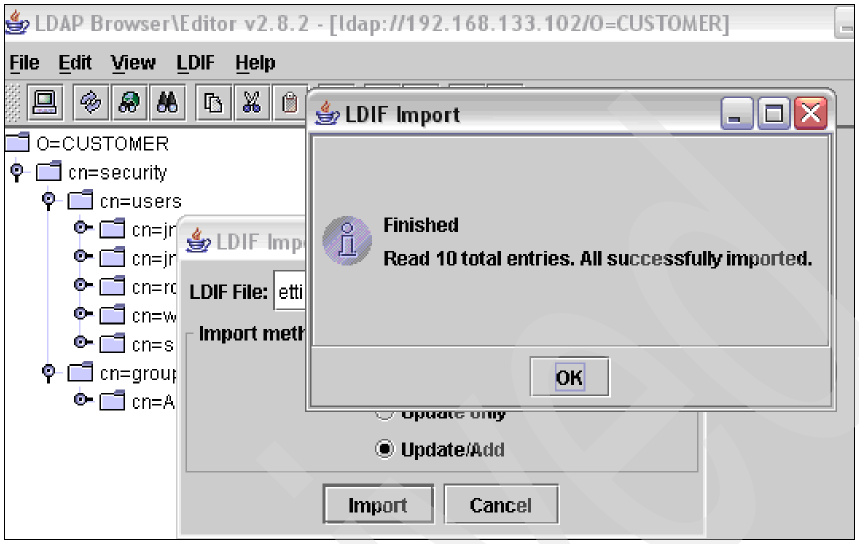
LDIF File: ettings\gb019470\Desktop\WPS\orig.ldif

Import method

( ) Add only
( ) Update only
(●) Update/Add

Figure C-23 Importing an LDAP directory structure via an LDIF

6. The directory structure will be imported, the example LDIF adds 10 new entries. Click **OK** to close the import windows (Figure C-24).



LDIF Import

Finished

Read 10 total entries. All successfully imported.

Figure C-24 LDIF successfully imported

It is now possible to browse through the structure that you have created for use in WebSphere security.

7. Close the LDAP browser.

ITDS Configuration is now complete.

Tivoli Directory Server useful commands

This section includes a few useful commands for control of the LDAP environment:

- ► To start the LDAP server the LDAP database must first be running. Login as the `idsldap` user, switch to the bash shell and start the database.

```
bash
db2start
```

- ► To start the ITDS Admin process, login as the idsldap user and issue the following commands once DB2 has started (this should be done from the `ksh` SHELL not `bash`):

```
ibmdiradm -I idsldap # Start the ITDS Admin process
ibmdirctl -D "cn=root" -w itso4you start
```

Note: If this start command returns an `rc=-1 error`, you can ignore this error. If you apply the ITDS V6.0 Fix Pack 3, this harmless error can be removed.

- The following command also starts the `idsldap` instance (if run as the `root` user):

  ```
  ibmslapd -I idsldap -n
  ```

- To get the server status use:

  ```
  ibmdirctl -D "cn=root" -w itso4you status
  ```

- To verify the server is responding use the `ldapsearch` command:

  ```
  ldapsearch -b "o=customer" "cn=*"
  ```

  This will display all container entries below o=customer

- To stop the LDAP server issue the command:

  ```
  ibmdirctl -D "cn=root" -w itso4you stop
  ```

- To stop the ITDS Admin process issue the command:

  ```
  ibmdirctl -D "cn=root" -w itso4you admstop
  ```

D

New product features used in the sample applications

This chapter provides information relating to the development of the sample applications that we describe in Chapter 5, “Scenarios that we use in this book” on page 65.

In our sample application, we use two new features that are available with WebSphere Integration Developer 6.0.2:

- Automatic client generation for business processes and human tasks
- Event Emitter mediation primitive to emit business events from within a mediation module

How to use these new features is the topic of this appendix.

Generating human task interfaces

The presentation layer application for the complex business scenario has been developed using new functionality available in WebSphere Integration Developer V6.0.2. This functionality simplifies the building the graphical user interface for the business process.

It is possible to generate a client from a module, business process, or a human task. To generate a client for a human task, proceed as follows:

1. In the Business Integration view of WebSphere Integration Developer, right-click the human task or tasks that requires a user-interface, and select **Generate User Interfaces**. To generate a client for multiple tasks in different modules, select each module in the business integration view while holding the Ctrl key. The User Interface Wizard for Human Tasks launches. See Figure D-1.

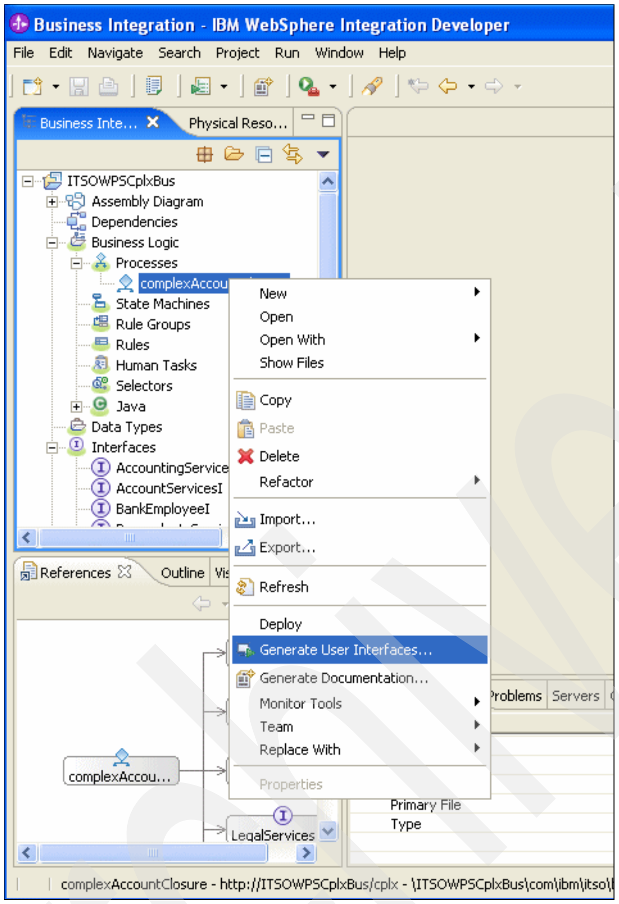
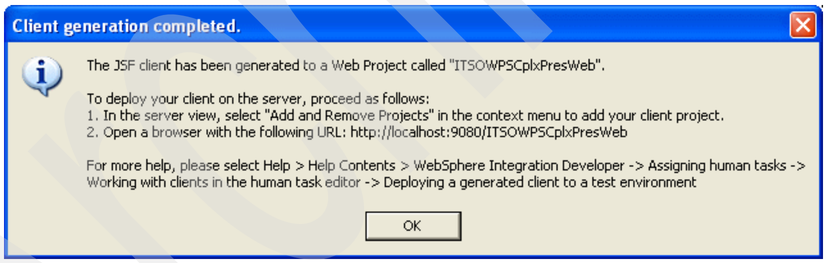


Figure D-1 Generate User Interface launch

Note: if the generated client is intended to be able to start processes, then this process must have a human task defined for the initial receive activity / activities.

2. On the Client Generator Selection page, proceed as follows:
   a. In the Generator type field, choose the generator that you will use to create this client. By default, the JSF Custom Client is selected.
   b. Use this list to choose the human task or tasks for which you want to generate the client. Expand the tree until you find the required human task, and then enable the associated check box for each task.

3. On the JSF client configuration page, proceed as follows:
   a. In the Name of dynamic Web project field assign the client name.
   b. In the Company logo field, specify the file location of a graphic file containing a company logo. This logo will appear as a banner on the top of the generated client Web page. Only GIF or JPG files with an image height of 50 pixels are accepted.
   c. In the Client view area, there are two choices. Choose **Local** if a single application server will host both the generated client and the related processes and tasks. Choose **Remote** if, in the same cell, the generated client will be deployed on one server, and the related processes and tasks on another.
   d. In the Style area, you can choose from the two styles that are provided for your generated client. If neither of these meet your needs, then you can adjust the CSS file accordingly, without needing to regenerate the client. The CSS file is located in the generated web project in `WebContent\theme\styles.css`.
4. On the second JSF client configuration page, select the custom properties (as defined in your business process) that need to be generated with the client. The custom properties can be used as filter criteria for search activities.
5. When you are done, click **Finish**. The application is created by the tool, and a confirmation message displays. See Figure D-2.



The JSF client has been generated to a Web Project called "ITSOWPSCplxPresWeb".

To deploy your client on the server, proceed as follows:
1. In the server view, select "Add and Remove Projects" in the context menu to add your client project.
2. Open a browser with the following URL: http://localhost:9080/ITSOWPSCplxPresWeb

For more help, please select Help > Help Contents > WebSphere Integration Developer -> Assigning human tasks -> Working with clients in the human task editor -> Deploying a generated client to a test environment

Figure D-2 Generate User Interfaces *confirmation* message

For our scenario, we added the start process functionality to the application provided by the Generate User Interfaces.

Using the Event Emitter mediation primitive

This section discusses the Event Emitter mediation primitive that was introduced with WebSphere Enterprise Service Bus V6.0.2.

Introduction

The Event Emitter mediation primitive sends out business events during a mediation flow. The events are generated in the form of Common Base Events (CBE) and are sent to a Common Event Infrastructure (CEI) server.

You can generate events that include all or part of the message being processed, by setting the Root property. The Root property is an XPath expression that specifies the section of the Service Message Object (SMO) that is included in the CBE application specific elements (extended data elements). The Root property can specify the complete SMO, a subtree or a leaf element.

If you do not set the Root property then you generate events that do not contain any part of the message being processed. You also generate events that do not include the message in another way: if you specify the Root property, but its value does not match any of the elements in the SMO when the message is mediated.

The Event Emitter mediation primitive has one input terminal and two output terminals. One output terminal is for successful output and one for failure output. The input terminal is wired to accept a message and the output terminals are wired to propagate a message. The successful output terminal propagates the original message. If an exception occurs during the processing of the input message, then the fail terminal propagates the original message, together with any exception information.

Usage

The recommended best practice is to use the Event Emitter mediation primitive to indicate an unusual event, such as notification of a failure within the flow or an unusual path executed in a flow. In our scenario we use the Event Emitter for logging stock orders when the marketplace is neither U. S. nor U. K.

When you place event emitters within a flow you should consider the possible number of events that may be generated as well as the size of the message that will be stored within the event. Placing an event emitter within the normal flow execution path will generate many events compared to placing it within an unusual event or failure path. In addition, consider configuring the emitter to store only significant message data rather than the complete message to reduce the overall size of the event.

You can use the Event Emitter mediation primitive to record a failure in another mediation primitive, and then continue processing. To do this you wire the fail terminal of the previous mediation primitive to the input terminal of the Event Emitter mediation primitive; and wire the output terminal of the Event Emitter mediation primitive to the next step in the flow.

Figure D-3 shows how the Event Emitter is used inside the mediation flow from the complex business process scenario that we use in this book.

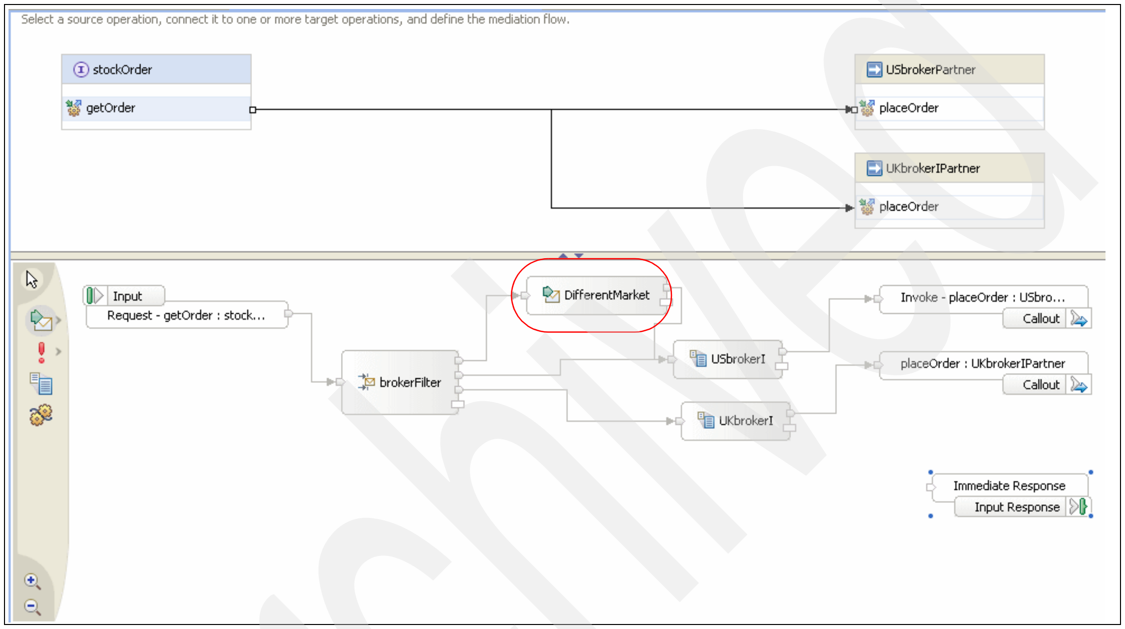

Select a source operation, connect it to one or more target operations, and define the mediation flow.
stockOrder
getOrder
USbrokerPartner
placeOrder
UKbrokerIPartner
placeOrder
Input
Request - getOrder : stock...
DifferentMarket
brokerFilter
USbrokerI
UKbrokerI
Invoke - placeOrder : USbro...
Callout
placeOrder : UKbrokerIPartner
Callout
Immediate Response
Input Response

Figure D-3 Event Emitter in mediation flow

In this case, we use the Event Emitter mediation primitive in combination with the `brokerFilter` Message Filter mediation primitive to generate business events based on message content. We wire one of the output terminals of the Message Filter mediation primitive to the input terminal of the `DifferentMarket` Event Emitter mediation primitive and wire the output terminal of the Event Emitter mediation primitive to the `USbrokerI XSL` Transformation, the next step in the flow.

Abbreviations and acronyms

BPC	Business Process Choreographer
BPEL	Business Process Execution Language
BRM	Business Rules Manager
CBE	Common Business Event
CEI	Common Event Infrastructure
CRM	Customer Relationship Management
CSI	Common Secure Interoperability
DDL	Data Definition Language
EIS	Enterprise Information System
EJB	Enterprise JavaBean
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
GSK	Global Security Kit
HA	High Availability
HTM	Human Task Manager
IBM	International Business Machines Corporation
IHS	IBM HTTP Server
ITSO	International Technical Support Organization
J2EE	Java 2 Platform, Enterprise Edition
JAAS	Java Authentication and Authorization Service
JMS	Java Message Service
LDAP	Lightweight Directory Access Protocol
LTPA	Lightweight Third Party Authentication
MDB	Message Driven Bean
POJO	Plain Old Java Object
QoS	Qualities of Service
RDBMS	Relational Database Management System
SCA	Service Component Architecture
SCDL	Service Component Definition Language
SDO	Service Data Objects
SIB	Service Integration Bus
SMO	Service Message Object
SOA	Service-oriented Architecture
SSL	Secure Sockets Layer
SSO	Single Sign-on
SWAM	Simple WebSphere Authentication Mechanism
UML	Unified Modeling Language
WAR	Web Archive

Related publications

We consider the publications that we list in this section particularly suitable for a more detailed discussion of the topics that we cover in this book.

Online resources

These Web sites are relevant as further information sources:

- WebSphere Process Server product overview

  `http://www.ibm.com/software/integration/wps/`
- WebSphere Application Server product overview

  `http://www.ibm.com/software/webservers/appserv/was/`
- J2EE 1.4 specification

  `http://java.sun.com`
- DB2 Universal Database product overview

  `http://www.ibm.com/software/data/db2/udb`
- WebSphere Process Server security overview

  `http://www.ibm.com/developerworks/websphere/library/techarticles/0602_khangoankar/0602_khangaonkar.html`
- Java 2 security with WebSphere Application Server based products

  `http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/csec_rsecmgr2.html`

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IBM

Redbooks

Production Topologies for WebSphere Process Server and WebSphere ESB V6

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

Production Topologies for WebSphere Process Server and WebSphere ESB V6

Discusses messaging, CEI, business process container, and security

Includes selection criteria to pick the right topology

Features examples that include step-by-step instructions

WebSphere Process Server and WebSphere ESB are highly available and scalable products for running enterprise-level SOA solutions. But does every production topology that uses these products follow the same basic pattern? This book answers that question by providing guidance on how to select and build production topologies based on the applications that you deploy to that topology. It also helps you to discern which production topology is suitable for your environment and provides step-by-step guidance on how to build that topology. The intended audience of this book is IT architects and administrators.

Part one of the book introduces some of the basic concepts and discusses three types of production topologies for WebSphere ESB and for WebSphere Process Server. It provides guidance on how to select the appropriate topology, discusses security considerations, and introduces the sample scenarios that we built for this book.

Part two focuses on building production topologies for WebSphere ESB. Using sample applications, this part provides step-by-step instructions for building WebSphere ESB V6.0.2 production topologies.

Part three uses a similar format but focuses on production topologies for WebSphere Process Server V6.0.2. This part includes step-by-step instructions on how to build a full support production topology that implements security.

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

SG24-7413-00 ISBN 0738489255