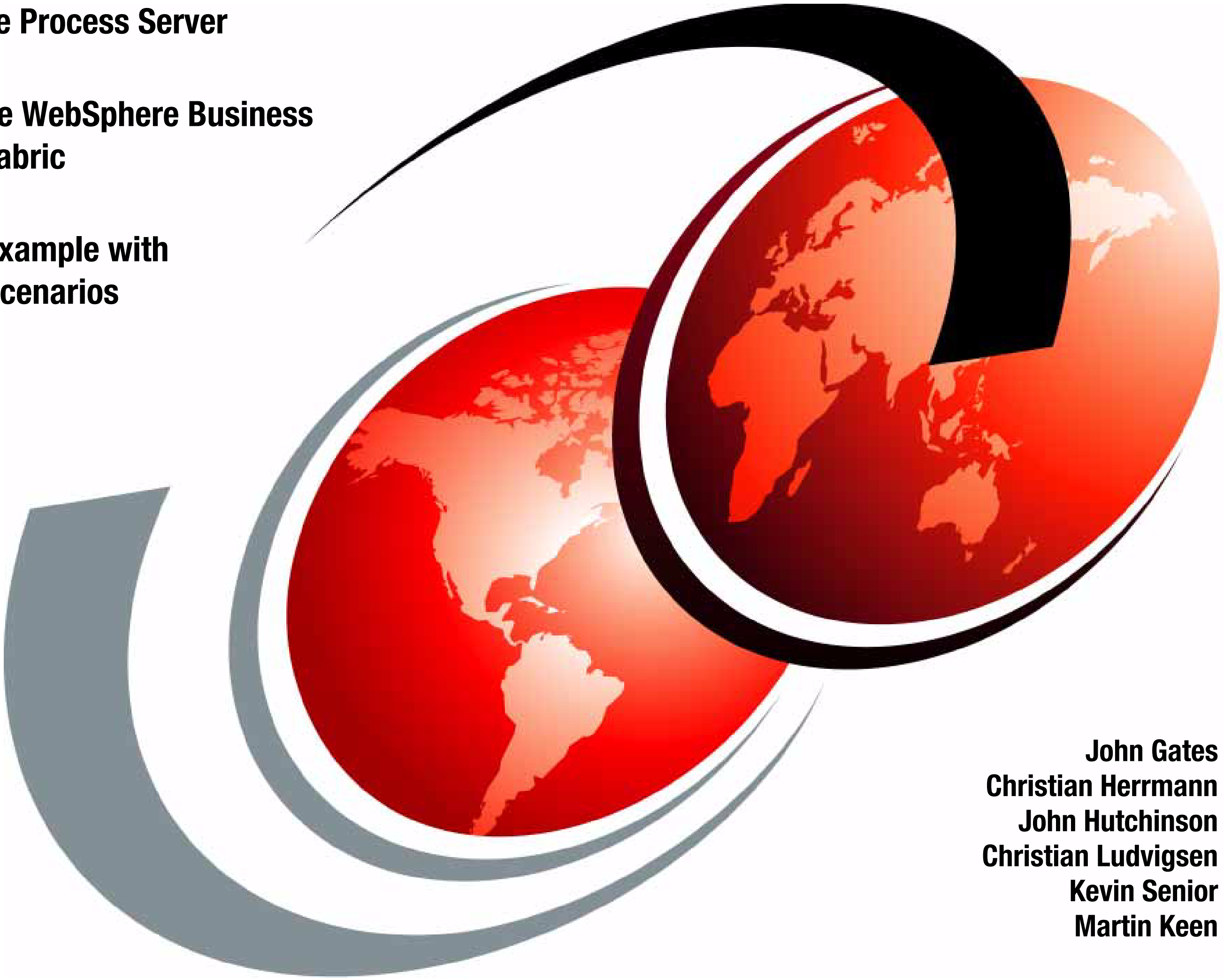
WebSphere® software

z/OS: WebSphere Business Process Management V7 Production Topologies

Create production topologies for WebSphere Process Server

Incorporate WebSphere Business Services Fabric

Learn by example with practical scenarios



John Gates
Christian Herrmann
John Hutchinson
Christian Ludvigsen
Kevin Senior
Martin Keen

Redbooks

ibm.com/redbooks

International Technical Support Organization

z/OS: WebSphere Business Process Management V7 Production Topologies

June 2010

SG24-7831-00

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

First Edition (June 2010)
This edition applies to Version 7 of WebSphere Process Server for z/OS and WebSphere Business Services Fabric for z/OS.

Contents

Notices

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

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

alphaWorks®
CICS®
DB2 Universal Database™
DB2®
developerWorks®
Domino®
IBM®
IMS™
Language Environment®
Lotus®
MVS™
PartnerWorld®
RACF®
Redbooks®
Redpaper™
Redbooks (logo) ®
RMF™
System z10®
System z®
SystemPac®
Tivoli®
WebSphere®
z/OS®
z10™
z9®
zSeries®

The following terms are trademarks of other companies:

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

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

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

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

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

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

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

Preface

In this IBM® Redbooks® publication, we address the configuration, administration, and security of the key runtime environments in business process management: WebSphere® Process Server V7.0 and WebSphere Business Services Fabric V7.0 for z/OS®. This book provides detailed guidance to z/OS system and database administrators who want to configure WebSphere Business Process Management production topologies.

We introduce production topology concepts and terminology and explore the differences between production topologies on distributed platforms and z/OS. Through a series of step-by-step instructions, you will learn how to create and verify a production topology environment for WebSphere Process Server V7 for z/OS.

We extend the production topology concept for WebSphere Process Server by describing step-by-step how to add WebSphere Business Services Fabric V7 for z/OS into the topology. You also get problem diagnosis and prevention guidance to use when you create your own production topologies.

A separate publication that covers distributed platforms is also available: *WebSphere Business Process Management V7 Production Topologies*, SG24-7854.

The team who wrote this book

This book was produced by a team of specialists from around the world.

John Gates is a Senior System z® IT Architect. He has almost 30 years of experience working on large systems in all areas of design, development, and test. In his current position, he works with large customers creating cross platform, complex, integrated solutions to e-business problems. He writes extensively and is a frequent speaker at industry events, such as Impact and the Share User group conference. He attended Marist College in Poughkeepsie, New York and has a BS degree in Computer Information Systems.

Christian Herrmann is a Senior IT Specialist for IBM Software Group Germany. He has a degree in Information Science from the University of Konstanz and a degree in Electrical Engineering from the University of Applied Science of Konstanz. He has over five years of experience in electrical engineering and over 18 years in the IT industry. He joined IBM in 1999 and started working in the WBI-FN development. For the last three years, he works within the WebSphere Solution Center in the development laboratory in Boeblingen, Germany, where he provides expert knowledge about WebSphere Process Server infrastructure on all supported platforms with a special focus on automation. His areas of expertise are WebSphere MQ, WebSphere Message Broker, WebSphere Application Server, and WebSphere Process Server.

John Hutchinson is an Executive Consulting I/T Specialist at the Washington Systems Center (WSC) in Gaithersburg, Maryland. He has over 40 years of experience in mainframe operating systems and the middleware field. He has a degree in Industrial Engineering from the University of California, Berkeley and joined IBM in 1967. He has written extensively on JES2, WebSphere Application Server for z/OS, and WebSphere Process Server for z/OS.

Christian Ludvigsen is an IT Specialist in Integrated Technology Delivery (ITD), Denmark. He has two years of experience in WebSphere Process Server for z/OS.

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

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

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

- Rich Conway and Roy Costa
  IBM ITSO Global Business Technologies
- Russ Heald
  WebSphere Process Server for z/OS Development, IBM Hursley, UK
- Mike Poirier and Dave Bonaccorsi
  IBM Middletown, RI, USA
- Brian Pierce, Mike Kearney, Mike Loos, Mike Cox, Jennie Liang, King AU, and Lee-Win Tai
  IBM Washington Systems Center, Gaithersburg, MD, USA
- Richard Johnson
  WebSphere BPM Business Space Development

Now you can become a published author, too!

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

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

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

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

- Stay current on recent Redbooks publications with RSS Feeds:

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

1

Business Process Management production topologies for z/OS

In this book, we describe a methodology for deploying the IBM Business Process Management (BPM) suite of products in a production, System z environment. In this chapter, we introduce BPM topology patterns and lay out, in broad detail, the plan for the remaining chapters of the book.

We compare the topology patterns for z/OS platforms to the topology patterns for the non-z/OS platforms that are described in the IBM Redbooks publication *WebSphere Business Process Management V7 Production Topologies*, SG24-7854. In most cases, we can follow distributed oriented examples from this book, however, in some areas, we diverge to exploit the additional capabilities of System z.

For this book, we deploy most of the key business functions in a z/OS sysplex. In a few cases, due to the state of the technology, availability of critical system components on z/OS, and the requirement for additional BPM server functions, we augment our z/OS sysplex with additional server capabilities that are deployed on Linux® for System z servers.

In this chapter, we discuss:

- 1.1, “Introduction to WebSphere Process Server for z/OS topology” on page 2
- 1.2, “WebSphere Process Server for z/OS” on page 9
- 1.3, “Deployment environments” on page 12
- 1.4, “Selecting an appropriate topology” on page 18
- 1.5, “BPM reference topology for z/OS and how we use it here” on page 21
- 1.6, “Incorporating WebSphere Business Services Fabric for z/OS into the BPM reference topology for z/OS” on page 23
- 1.7, “Business Space powered by WebSphere” on page 24

We describe two topology patterns in 1.3, “Deployment environments” on page 12:

- Single-cluster topology
- Two-clusters topology

In 1.4, “Selecting an appropriate topology” on page 18, we discuss our reasons for choosing one pattern over another.

For WebSphere Process Server for z/OS on non-z/OS platforms, we gave the topology choices names, such as bronze, silver, or gold. With WebSphere Process Server for z/OS, we avoided using names that imply that one choice is better than another; instead, we discuss the advantages and disadvantages of a single-cluster topology compared to the two-clusters topology, to help you select a production topology that best meets your requirements.

We discuss our recommended topology for a BPM production deployment on z/OS in 1.5, “BPM reference topology for z/OS and how we use it here” on page 21. We refer to this deployment model throughout this book.

1.1 Introduction to WebSphere Process Server for z/OS topology

A WebSphere Process Server for z/OS topology is the physical layout of the deployment environment that is required to meet your business needs for capacity, availability, and scalability.

When we discuss topologies for WebSphere Process Server for z/OS on non-z/OS platforms, the choices are usually described as:

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

The names bronze, silver, and gold come about because performance studies show that the scalability depends on the topology. Therefore gold is better for performance than silver, which in turn, is better than bronze.

With WebSphere Process Server for z/OS, names like this are misleading because it is sometimes unclear what makes one topology better than another on z/OS. The choice of topology depends on the particular circumstances of each customer. In most cases, it is not necessary to split the components of WebSphere Process Server for z/OS across multiple clusters to achieve significant capacity, availability, and scalability.

This difference between WebSphere Process Server for z/OS and WebSphere Process Server for z/OS on non-z/OS platforms stems from the separate architecture of the underlying WebSphere Application Server for z/OS.

It is possible to have a mixed-stack approach with several parts of your infrastructure on a non-z/OS platform and others on z/OS. A project that created such a topology and compared it with a pure z/OS topology is described in the IBM White Paper *The Mixed Platform Stack Project: Deploying a secure SOA solution into z/OS*, WP101300, which you can get from the following Web page:

http://www.ibm.com/support/techdocs

In this Redbooks publication, we primarily describe z/OS topologies. You must not, however, approach your choice of production topology in a dogmatic manner. There are good reasons to deploy on non-z/OS platforms, on z/OS, and on a mixed-stack. The correct choice for your organization depends on your unique requirements. In this book, we introduce several topology components that do not run natively on z/OS. The functions that these components provide are critical to the overall success of an end-to-end BPM solution and are thus included for this reason.

In general, BPM solutions are platform-independent. The servers involved, WebSphere Process Server for z/OS, WebSphere Business Monitor, and WebSphere Business Compass, support the entire BPM programming model, no matter where they are placed. Different server platforms support various qualities of service, and satisfy architectural concerns in terms of non-functional requirements in various ways and with varying levels of success. When choosing a platform on which to deploy your BPM topology, consider all of these factors. Flexibility to choose your deployment platform and topology is the goal we are trying to achieve.

For more information about non-functional requirements and how they relate to server placement and topology choices, refer to 1.1.4, “Non-functional requirements” on page 4.

1.1.1 WebSphere Application Server for z/OS architecture

A WebSphere Application Server for z/OS server has an architecture, shown in Figure 1-1, that includes a controller region, one or more servant regions, and a separate address space called the control region adjunct (or simply, the adjunct), which hosts the message engines.

By providing an adjunct address space with a separate Java™ Virtual Machine (JVM) for the message engines, the architecture of a single-cluster in WebSphere Application Server for z/OS is analogous to the two-clusters silver topology of a non-z/OS WebSphere Application Server for z/OS that is described in the Redbooks publication *WebSphere Business Process Management V7 Production Topologies*, SG24-7854.

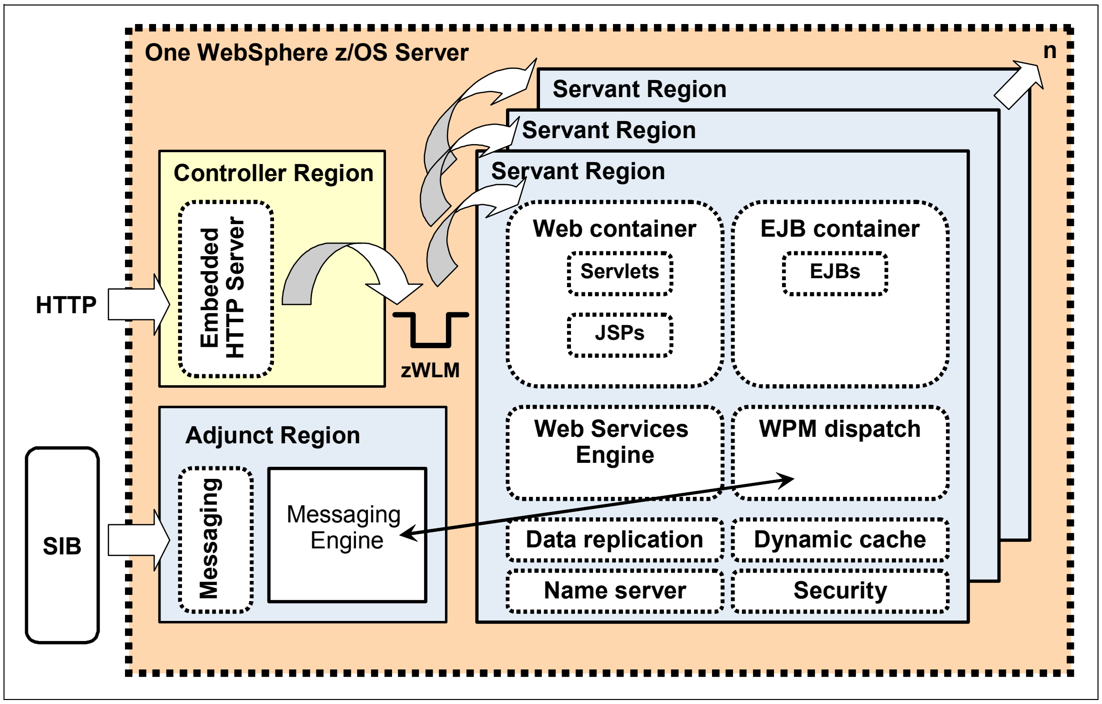


Figure 1-1 Architecture of a server in WebSphere Application Server for z/OS

The simplest WebSphere Process Server for z/OS production topology, comprising one cluster, starts at the silver topology. It is not normally necessary to split the messaging engines into a separate cluster to achieve performance and scalability.

The WebSphere Process Server for z/OS single-cluster topology is described in more detail in 1.3.1, “Single-cluster topology” on page 13.

1.1.2 Hardware considerations

A WebSphere Process Server for z/OS production topology does not normally need to concern itself with the zSeries® hardware, except that more clusters imply more JVMs and greater memory requirements. As documented in our experiences in this book, we use 64-bit WebSphere Application Server for z/OS servers. We do this primarily so that we can have access to more JVM heap memory.

When estimating CPU capacity it is important to remember that WebSphere Process Server for z/OS processing is primarily Java. When running in our WebSphere Application Server for z/OS-based environment where processes were deployed, we measured zAAP offload in the range of 60–70%, depending on workload. If CPU consumption on general purpose processors is a concern for you, consider deploying zAAP processors.

Ensure that the zSeries machines in your parallel sysplex have enough capacity to support the topology that you require. Specific CPU and memory requirements are best achieved through experimentation and rigorous benchmarking activities. The program directories for the products being deployed generally give good advice on where to start.

1.1.3 HTTP, LDAP, and edge servers, and proxies, external security managers, and so forth

A production topology includes other non-WebSphere Process Server services, such as one or more HTTP servers (for Web content) and firewalls.

With WebSphere Process Server for z/OS, load balancing of HTTP requests across an HTTP Server that is running on z/OS is performed by Sysplex Distributor and is not something that must be considered in your topology design. You must, however, define one or more Distributed VIPAs to exploit the Sysplex Distributor function.

If you are using a LocalOS, SAF-based user registry, such as RACF®, the user registry is not something you must consider in your WebSphere Process Server for z/OS production topology. It must already be configured with a RACF database that is shared across the parallel sysplex. However, if you decide to use LDAP, you must plan for high availability and disaster recovery if the LDAP is not running on z/OS.

In an Internet environment, it is normal to install a proxy, such as WebSeal, in the De-Militiarized Zone (DMZ) and to authenticate using an external security manager, such as Tivoli® Access Manager. You must plan for the scalability and availability of any external authentication infrastructure.

1.1.4 Non-functional requirements

In this section, we addresses the following topics:

- Performance
- 64-bit addressability
- Scalability
- Recovery
- Availability
- Servicability

Performance

Performance of the deployed architecture is certainly a factor in designing the topology to be deployed. Some of the factors that are involved are the same as those that drive other aspects of the design process.

CPU

One of the factors to consider is the performance profile of the application to be deployed. Can the application in the best possible configuration make the maximum use of the CPU capacity on the machine under load? In our experience, the answer in the case of WebSphere and Java based products is that it depends. With good application design and implementation, it is possible to see good CPU utilization and overall satisfactory performance. In a z/OS environment, the inherent design of the WebSphere Application Server for z/OS runtime promotes maximum CPU utilization with minimal configuration and manual intervention. More about this in "Scalability" on page 6.

Memory

Memory is also a concern in WebSphere-based processing. Many Java processes are, by their nature, memory intensive. We must therefore pay attention to how much physical and virtual memory we dedicate to a particular LPAR and WebSphere configuration.

In many distributed architectures, it might make sense to base a design around a topology that spans LPARs and servers to fully exploit the installed memory of the servers that are involved. On z/OS, the technology is robust, and memory management is not a concern. Because z/OS is designed to use all configured memory, the primary concern is not whether the machine uses the memory but if there is enough installed memory available for all processes that are involved.

While the major driving force behind memory utilization is Java heap size, we must also consider the memory requirements of the base operating system and address spaces. We must also consider the physical machine architecture.

Most modern WebSphere-based products use large amounts of Java heap. When running in 31-bit addressing mode, unlike distributed platforms, the System z architecture is limited to 31-bits. In this mode, the maximum available memory for a single z/OS address space is 2 GB. All memory for the address space must be served from this 2 GB address range. In practice, available heap in a 31-bit address space is approximately 768 MB.

In many cases, 31-bit servers can be sufficient to run all of the deployed business processes. Additional memory utilization can be achieved through deploying additional servants unless the application sets themselves consume more memory than can fit in a single, 31-bit JVM. In these cases, it might be necessary to deploy 64-bit servants, as described in "64-bit addressability" on page 6.

WLM workload classification is another way to isolate large applications into specific WebSphere servants. This technique allows applications with large Java memory requirements potential to be isolated in fewer 64-bit JVMs.

Application Design

The importance of good application architecture and design cannot be understated because it is just as much a factor when deploying on z/OS as it is on other platforms. Application design can have a direct effect on performance in a number of ways, for example, applications that exploit functions that are provided by large, open source packages must not, by default, include these packages in each application .ear file. This practice results in memory being used to hold multiple copies of the same package. A better practice is to exploit the capabilities of the shared library function within WebSphere to reduce the overall memory footprint of deployed applications.

In our experience, using shared libraries can significantly reduce the overall heap size that a given server uses and therefore the amount of real memory required to support a server instance/ guest/ LPAR. Using shared libraries also has the side effect of reducing server startup time because all packages do not have to be loaded or started for each server.

64-bit addressability

Modern machine architectures are 64 bit. System z is no exception. Modern system z processors use a 64-bit architecture. WebSphere and all of the BPM products that we discuss in this IBM Redbooks publication are fully enabled for 64 bit through their use of a 64-bit JVM.

In various cases, deploying large business processes or large numbers of smaller business processes can force the need for a 64-bit servant to be defined. Because 64-bit processing uses more system resources than 31-bit processing, careful analysis of the deployed workloads must take place before deciding on 64-bit servants.

Note: In this book we use WebSphere Application Server for z/OS V7, which deprecates the use of 31-bit servers and changes to use 64-bit servers by default. In our sysplex environment, we were not constrained by memory; therefore, in our examples, we accept this default and deploy our servers on 64-bit servers.

If memory is a concern in your environment, using 31-bit servers for non-application serving address spaces, such as the WebSphere Application Server for z/OS Deployment Manager controller and servant and the WebSphere Application Server for z/OS node agent controller, is an option.

Because 31-bit servers are deprecated in WebSphere Application Server V7 and might be removed at any time, we strongly recommend that you take the required planning steps to progressively migrate all WebSphere servers to 64-bits as soon as possible.

Scalability

Scalability of WebSphere-based architectures is achieved on distributed platforms by building vertical clusters. By adding additional cluster members on the same physical machine, distributed architectures can approach 100% CPU utilization on a single platform.

Scalability of WebSphere on z/OS is achieved by adding more servants to a single deployed WebSphere Application Server. Rather than adding more cluster members across physical nodes, WebSphere in conjunction with the z/OS operating system automatically adds and subtracts JVMs and servant capacity on demand, which is the normal practice when scaling WebSphere-based processes on a z/OS platform and is an entirely automatic process. Consider this default behavior when designing your WebSphere Process Server for z/OS topology.

Recovery

Recovering architecture components is a factor in designing a BPM topology. The z/OS operating system and installed infrastructure components do a good job of exploiting the underlying hardware and software facilities for recovery and first failure data capture. The design of the WebSphere Application Server for z/OS also helps.

On z/OS, the basic tenant is to keep the operating system alive at all costs. In the name of OS survivability, we sacrifice almost everything.

The BPM portfolio products that run on top of WebSphere Application Server for z/OS inherit all of the QoS of the base WebSphere design:

- WebSphere Application Server for z/OS is designed with a clear separation of concerns with respect to where code runs:
  - Operating system code and code that has access to the operating system runs authorized in a controller address space.
  - Application code, that is WebSphere, WebSphere-based products, and anything that runs in a JVM, runs in its own execution environment, which is called a servant address space.
  - Code that is running in the controller is completely isolated from code that is running in the servants. Communication between the processes is through WLM-managed work queues.
  - In the event of an application error, a particular servant can end abnormally. It does, in most cases, restart automatically by the controller in place. Various additional recovery actions are possible based on individual requirements. Failures that are associated with an application are therefore isolated to the servant where the error occurred.
  - In the event of a controller failure, the z/OS policy-based recovery that is managed by a z/OS component called the Automatic Restart Manager can restart the controller in place or on another system in the z/OS sysplex.
- Subsystems, such as CICS, IMS, DB2, and MQ are designed to integrate well with WebSphere and its stack products, such as WebSphere Process Server for z/OS and WebSphere Business Monitor. In addition, they are built from the ground up to interact closely with key operating system recovery structures and schemantics to ensure that in the event of a failure of a WebSphere operation that involves one or more resource managers, coordinated recovery amongst WebSphere and those subsystems is performed. This coordinated recovery removes the burden of the application having to build complex logic to handle recovery of subsystem resources, such as the database.
- In the unlikely event of a failure of the z/OS operating system, additional recovery options for surviving members of the z/OS sysplex are available.

Robust WebSphere-based architectures consider the factors listed in the previous list in the application of technology in the z/OS environment. The bottom line recommendation is that architectures on z/OS do not have to be overly complex to deal with recovery actions and system failures that might occur on other platforms.

Availability

In general, you only need multiple cluster members on z/OS to provide greater availability. It is natural to think that two is enough. However, it is better to use three than two, which is known as the *Rule of Three.*

When you run multiple cluster members you are protecting against the failure of the entire logical partition (LPAR) or of some singleton address space on which the cluster member depends. If the controller region fails, you lose the capacity of that cluster member. When you lose one cluster member the remaining cluster member must take up the entire load of the failing cluster member in addition to its own. A sudden increase in load from 50% capacity to 100% can cause the remaining cluster member to fail too. When there are three cluster members, however, and one fails, there are two remaining cluster members to absorb 33% of the workload from the failed cluster member, which means a growth of only about 17% in the workload of each cluster member, which might not cause a problem.

On z/OS, when the workload suddenly shifts to another cluster member, the effect is not as bad as it is on a non-z/OS platform because the re-assignment of resources to the remaining cluster by zWLM helps it deal with the extra workload. If you configured the cluster so that the number of servants vary with the workload, the impact of one cluster member failing might not cause a problem for one remaining cluster member.

However there is still one controller region processing all of the communications. You can run into a bottleneck with the number of SSL threads in the controller. For this reason, although you might never encounter a situation where two cluster members is not adequate, if you have the flexibility in your system configuration, we recommend three cluster members for maximum availability and stability, even on z/OS.

> **Two node configuration:** The focus of this IBM Redbooks publication is on configuration and deployment; therefore, to simplify the scenarios in which we deploy, our examples throughout this IBM Redbooks publication use a two-node configuration. Aside from the benefits we describe, a two-node configuration must perform equally as well as a three node configuration.

Servicability

The reality is that in many cases, debugging complex problems requires additional resources and in other cases, problem recreates. A successful architecture is therefore adaptable to service requirements and components and allows for flexible data capture and problem resolution. While not an overriding architectural concern, it is consistent with good design patterns to build topologies that can be *easily* debugged when problems occur.

In this book, we present several techniques that make servicability in this environment easier. While not required, using techniques, such as intermediate symbolic links, can make the service process easier and must therefore be adopted as part of deploying the BPM reference topology for System z.

1.1.5 Summary

When planning your WebSphere Process Server for z/OS production topology, consider the following factors:

- The CPU required and the memory required to support all of the JVMs.
- Consider WebSphere Application Server for z/OS V7 64-bit server default and whether to use 64-bit versus 31-bit to allow for sufficient Java heap.
- The number of clusters and cluster members that are required to support your business and availability requirements. For situations where continuous availability is required, clusters with three cluster members are best.
- The number of servants per cluster member required to support the workload.
- Consider whether WebSphere Process Server for z/OS clusters are configured in distinct nodes from clusters that run other WebSphere workloads. This technique can provide additional flexibility to run WebSphere Process Server for z/OS clusters at various service levels within the same cell.
- If using LDAP, the network communication, high-availability configuration, and disaster recovery requirements of the LDAP server.
- The number and location of any HTTP servers.
- Whether you use WebSphere MQ for z/OS as a JMS provider or if using the WebSphere default messaging provider is sufficient.
- Servicability requirements and additional capacity required to accommodate recreates for data capture.

1.2 WebSphere Process Server for z/OS

In this section, we discuss a number of components that are created and used when configuring a WebSphere Process Server for z/OS topology.

1.2.1 Databases

A production WebSphere Process Server for z/OS cluster must use DB2 for z/OS rather than Derby to hold, store, and track information. You can choose to define all of the tables for all components of WebSphere Process Server for z/OS in one database or create databases for each component.

Unlike WebSphere Process Server for z/OS on non-z/OS platforms, you do not need to use multiple databases for performance reasons. When using DB2 for z/OS, it is more a matter of administrative convenience. There is no need to consider the physical location of databases when selecting your production topology because they are all in DB2 for z/OS.

Whether you choose one or two clusters, you must configure DB2 for z/OS in data sharing mode so that all cluster members can access the same data. For our BPM reference topology for z/OS, in our B7 cell, that databases that we defined are:

- Common database (B7WPSDB)

  Used as a repository for various components in WebSphere Process Server for z/OS, create this database prior to starting WebSphere Process Server for z/OS. It persists information regarding these components:

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

  The WebSphere Process Server for z/OS common database is also used by WebSphere Business Services Fabric for z/OS.

- Business Process Choreographer (BPC) database (B7BPCDB)

  The Business Flow Manager and the Human Task Manager use this database. Create it prior to starting BPC components.

- Messaging engine database (B7SIBDB)

  The SCA system and application buses, the CEI bus, and the BPC bus use these databases. When running WebSphere Business Monitor in a cross-cell configuration, a foreign bus is created for WebSphere Business Monitor for cross cell communication of business events. We created each bus in a separate database because their Data Definition Language (DDL) uses the same tablespace names. It also allowed us to assign specific buffer pools to each bus, if required, more easily.

- Event database (B7CEIDB)

  This database persists information regarding the Event Service, such as Common Based Events and key performance indicators.

- Business Space database (B7BSPDB)

  The Business Space component uses this database for configuration information that is associated with deployed widgets.

Table and schema names: Some BPM topology components use the same table names, which can present problems if a single database is created in support of multiple products. We encountered this problem when creating the tables that support WebSphere Business Monitor in the same database that supports WebSphere Process Server for z/OS. In this situation, we recommend that you use separate schema names. In the case of WebSphere Process Server for z/OS and WebSphere Business Monitor, we additionally recommend using separate databases due to the complexity of the environments to be supported.

1.2.2 Service integration buses

A service integration bus is a managed communication mechanism that supports service integration through synchronous and asynchronous messaging. A bus consists of interconnecting messaging engines. WebSphere Process Server for z/OS uses the following service integration buses:

- SCA system bus

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

- SCA application bus

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

- Common Event Infrastructure (CEI) bus

  This bus transmits common base events asynchronously to a CEI server.

  When WebSphere Business Monitor is running in another cell, an additional foreign CEI bus is constructed to provide cross-cell communication to WebSphere Business Monitor.

- BPC bus

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

On non-z/OS platforms, experience shows that running the message engines in the same cluster as the rest of WebSphere Process Server for z/OS can cause a bottleneck. Therefore, much emphasis is placed on ensuring that the messaging infrastructure runs in its own cluster. However, as we discussed in 1.1.1, “WebSphere Application Server for z/OS architecture” on page 3, because on z/OS, the message engines run in adjunct regions, it is not necessary to configure them in a separate cluster.

1.2.3 Messaging engines

Messaging engines are servers that provide asynchronous messaging support for your applications and for the internal messaging needs of the WebSphere Process Server components, such as the internal navigation queues that long running business processes use.

On non-z/OS systems, there are well-known advantages to configuring messaging engines in a separate cluster, but with WebSphere Process Server for z/OS there is probably no reason to do this. Message engines already run in separate adjunct regions, which is like running them in a separate cluster.

Messaging engines can use two types of message store:

- File-based
- Database

In our implementation, we always uses database message stores backed in DB2 z/OS.

1.2.4 WebSphere MQ for z/OS queue sharing

A unique feature of WebSphere Process Server for z/OS is the ability to use WebSphere MQ for z/OS as a JMS provider. In a clustered environment, using shared queues with WebSphere MQ for z/OS you can run multiple instances of a message engine without the disadvantages of partitioning the queues.

If your messaging workload exceeds the capacity of one adjunct region, you can consider configuring a WebSphere MQ for z/OS JMS provider using the bindings mode for optimal performance and security. In high message rate environments, consider this option *before* resorting to using the WebSphere default message provider in additional, message only, WebSphere clusters.

1.2.5 Business Process Choreographer

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

- Business Flow Manager

  The Business Flow Manager provides services to run business processes within an application server.

- Human Task Manager

  The Human Task Manager provides services to run human tasks within an application server.

Applications that use Business Flow Manager or Human Task functions must be deployed in a cluster where BPC is configured. In the two-clusters topology, BPC is configured in the AppTarget cluster, which was the b7cl02.AppTarget cluster in our B7 cell.

1.2.6 Common Event Infrastructure

Common Event Infrastructure (CEI) is an embeddable technology that is intended to provide event management services to applications that require those services. For service component event points that you monitor, events can be published to the CEI server and stored in the CEI database.

Depending on the extent that applications use CEI, it might be necessary to define CEI in a separate cluster. In the two-clusters topology, CEI is configured in the support and messaging cluster, which is remote from the cluster that hosts the applications, which is probably enough separation.

In the BPM reference topology for z/OS, all buses are defined in the same cluster. For additional capacity, rather than build another cluster, it might make more sense to define the message engines using MQ shared queues.

1.2.7 WebSphere Process Server Applications

WebSphere Process Server for z/OS provides a variety of Web-based application tools, such as:

- BPC Explorer

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

- Business rules manager (BRM)

  BRM assists business analysts in browsing and modifying business rule values.

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

These applications are all support functions, and therefore, in a distributed implementation, are deployed into a support and messaging cluster in the two-clusters topology. On z/OS, we use a single cluster topology and deploy these functions there in the single cluster.

1.3 Deployment environments

When using WebSphere Process Server for z/OS on non-z/OS platforms, you can create a deployment environment using IBM-supplied deployment environment patterns. The deployment environment patterns that are included in the Integrated Solutions Console and the profile management tool represent the most common deployment environments. Each of the patterns centers around the number of WebSphere Process Server clusters and cluster members. Think of deployment environment patterns as wizards that define a chosen topology.

WebSphere Process Server for z/OS contains three basic sets of functions that together form a complete production environment. Each of these functions can be separated into individual, dedicated clusters, but on z/OS it is best to start by assuming they are deployed in the same cluster.

The three sets of functions in the WebSphere Process Server environment are:

- Application deployment target

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

- Supporting infrastructure

  The supporting infrastructure is the CEI and other infrastructure services that support your environment, such as the BPC Explorer, Business Rules Manager, and Business Space.

- Messaging infrastructure

  The messaging infrastructure is the set of servers that provide asynchronous messaging support for your applications and for the internal messaging needs of the WebSphere Process Server components, such as the internal navigation queues that long running business processes use.

On non-z/OS systems, there are well-known advantages to configuring the messaging infrastructure in a separate cluster, but with WebSphere Process Server for z/OS, there is probably no reason to do this. Message engines already run in separate adjunct regions, which is like running them in a separate cluster.

With WebSphere Process Server for z/OS, the main topology decision you face is whether it is worth splitting major application functions, such as WebSphere Business Services Fabric for z/OS, into an AppTarget cluster and leaving the majority of the WebSphere Process Server for z/OS function deployed in the support and messaging cluster.

We describe the single-cluster and two-clusters topologies in more detail in 1.4, “Selecting an appropriate topology” on page 18. We compare the characteristics of these topologies to help you choose between them.

1.3.1 Single-cluster topology

The single-cluster topology pattern provides one cluster for all of the functional components. The user applications, messaging infrastructure, CEI, and support applications are all configured in the same cluster.

On z/OS, a single-cluster topology is a viable production topology because of the variant architecture of WebSphere Application Server for z/OS.

As we mentioned in 1.1.1, “WebSphere Application Server for z/OS architecture” on page 3, because the message engines run in a separate adjunct region means that the drawbacks of this topology do not apply to WebSphere Process Server for z/OS. A single-cluster topology on z/OS has characteristics that are analogous to those of the non-z/OS silver topology. There are other operational advantages, however, such as zWLM-managed servant regions, which make a single-cluster on z/OS what we call a polished silver topology.

Figure 1-2 shows a single-cluster topology sample configuration for WebSphere Process Server.

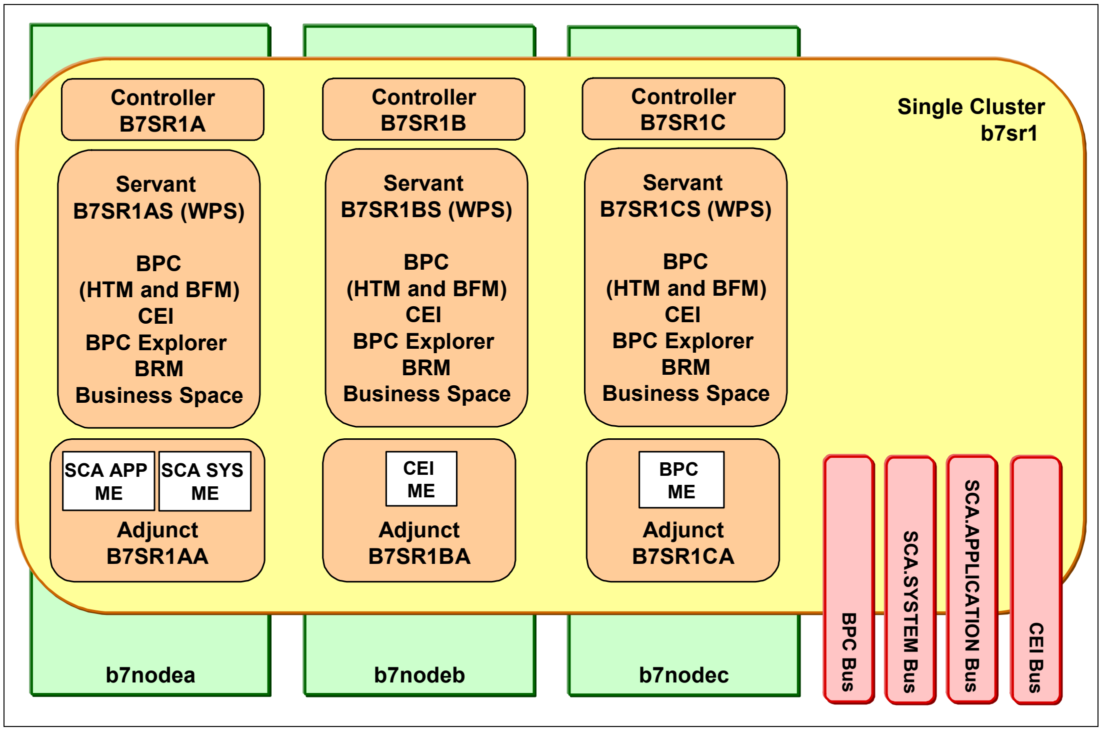


Figure 1-2 WebSphere Process Server for z/OS single-cluster topology

In Figure 1-2 on page 13:

- All of the components are configured in a single cluster. In our single-cluster topology cell B7, this cluster is called *b7sr1*.
- The b7sr1 cluster is a member of all four of the required WebSphere Process Server buses:
  - SCA.SYSTEM bus
  - SCA.APPLICATION bus
  - CEI bus
  - BPC bus

  The message engines execute in the adjunct regions.
- Each cluster member has a business process and Human Task Container.
- All of the supporting infrastructure applications are configured in the cluster:
  - BPC Explorer
  - Business Rules Manager
  - CEI
  - Business Space
- Each cluster member is an application deployment target.
- The messaging engines are shown running in specific adjuncts, as shown in Table 1-1.

Table 1-1 Messaging engines and their adjuncts, as shown in Figure 1-2 on page 13

Adjunct	Messaging engine
B7SR1AA	SCA.SYSTEM and SCA.APPLICATION
B7SR1BA	CEI
B7SR1CA	BPC

  This is not the default configuration. By default, each cluster member can run all four of the messaging engines, and the server that starts first automatically runs all four of the engines.

  You can configure distinct service integration bus policies to control where message engines run. For more information, see the Redbooks publication *WebSphere Business Process Management V7 Production Topologies*, SG24-7854.
- Each server scales by running multiple servants, not by adding more cluster members. More than one cluster member is required for availability reasons (and three is a good number) but not for scalability reasons.

The behavior of the messaging engines in a single-cluster topology is unlike the behavior in a remote cluster. When the messaging engines and the applications are co-located, the default behavior is for message producers and consumers to use a local active messaging engine, if one is available. Thus, if you have two applications deployed to each cluster member that must communicate asynchronously, after each message producer places messages in the queues, the message consumer on the machine where the engine is local consumes all of the messages that are produced. The consuming application only processes messages on the server with the local messaging engine.

Read and write local also creates a unique set of issues if you attempt to partition the destinations. When you create more than one active set of messaging engines, partitioning results. Each server's active messaging engines contain a portion of the queues that are assigned to that engine. Unfortunately, although you can obtain additional throughput if there are active messaging engines on each server, partitioning can create issues for your applications.

If you partition destinations when the applications and messaging engines are in the same cluster, you will not be able to maintain message order, which is true even if you attempt to enable event sequencing in WebSphere Process Server. Partitioned destinations can also create unpredictable behavior if one or more messaging engines fails in a single-cluster topology. If you are prepared to endure unpredictable behavior and loss of message order, partitioning the destinations in a single-cluster topology might be acceptable. However, we discourage this configuration.

Workload sharing: For information about workload sharing with queue destinations, refer to the WebSphere Application Server for z/OS Information Center at:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.pmc.zseries.doc/concepts/cjt0007_.html

1.3.2 Two-cluster topology

The two-clusters topology pattern is analogous to the Remote Messaging and Remote Support topology pattern for WebSphere Process Server for z/OS on non-z/OS platforms, where it is also known as the gold topology.

For non-z/OS platforms, this is the preferred topology for production environments, but on z/OS there might be no need to define multiple clusters for performance reasons. We considered deploying this topology, but ultimately decided not to. Instead, we directed our focus towards the BPM reference topology for z/OS, which we highlight in 1.5, “BPM reference topology for z/OS and how we use it here” on page 21. We present the two-cluster topology here for completeness and to offer you an additional choice of deployment topologies to consider. Refer to 1.4, “Selecting an appropriate topology” on page 18, for more explicit considerations for choosing a deployment topology.

Figure 1-3 on page 16 shows a B7 cell, which is comprised of two clusters:

- b7sr2.WPS_SandM: Provides the support and messaging function.
- b7sr2.AppTarget: Runs BPC and the applications.

Configuring this topology in WebSphere Process Server for z/OS is similar to configuring the single-cluster topology. It is largely a matter of invoking the appropriate wizard in the correct cluster.

For our B7 cell, we only use two nodes. Figure 1-3 on page 16 shows three nodes to illustrate the possible assignment of the buses to specific adjuncts. Also, for availability reasons, we recommend using three nodes with three cluster members in “Availability” on page 7.

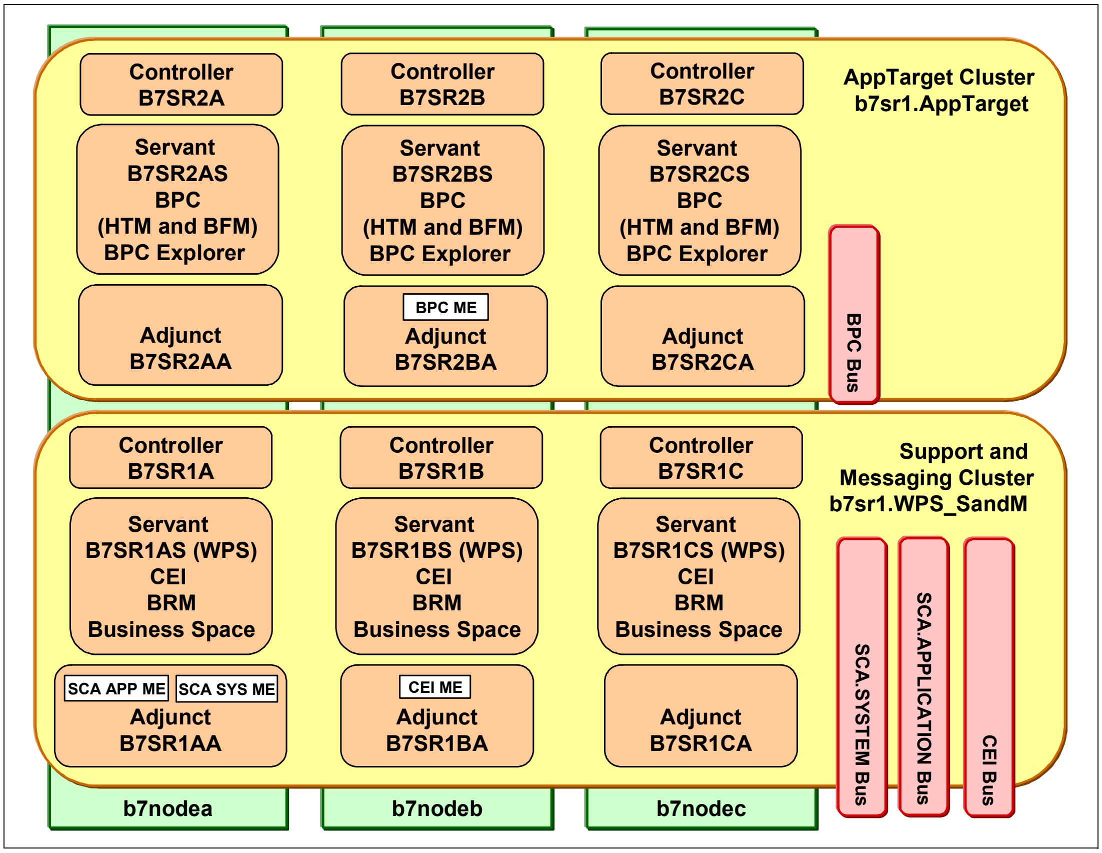


Figure 1-3 WebSphere Process Server for z/OS in the two-clusters topology

In Figure 1-3:

- All of the applications are deployed to the b7sr2.AppTarget cluster.
- The BPC is configured in the b7sr2.AppTarget cluster, so each cluster member has a Business Process Container and a Human Task Container.
- The b7sr2.WPS_SandM cluster is a member of three of the four required WebSphere Process Server for z/OS buses:
  - SCA.SYSTEM bus
  - SCA.APPLICATION bus
  - CEI bus

  The b7sr2.AppTarget cluster is a member of the BPC bus.

  The BPC bus can be defined locally in the same cluster as BPC because the BPC message engine runs in an adjunct, so there is no performance concern with running it in the same cluster.
- The supporting infrastructure applications are configured in the b7sr2.WPS_SandM cluster:
  - BPC Explorer
  - Business Rules Manager
  - CEI
  - Business Space
- Remote SCA must be configured in the b7sr2.AppTarget cluster.
- In Figure 1-3, the messaging engines are shown running in the specific adjuncts shown in Table 1-2 on page 17.

Table 1-2 Messaging engines and their adjuncts, as shown in Figure 1-3 on page 16

Adjunct	Messaging engine
B7SR1AA	SCA.SYSTEM and SCA.APPLICATION
B7SR1BA	CEI
B7SR2BA	BPC

Table 1-2 does not show the default configuration. You can customize a Service Integration Policy to assign a preferred server for the active message engine.

- As with the single-cluster topology, each server scales by running multiple servants, not by adding more cluster members. More than one cluster member is required for availability reasons (and three is a good number) but not for scalability. Using the two-clusters topology you can add servants to cluster members in the AppTarget cluster, which assists in adding capacity to the applications, while keeping the capacity of the support and messaging infrastructure the same (or vice versa).

The behavior of the messaging engines in the two-clusters topology is not like the behavior when the messaging engines are collocated with the applications. Because the messaging engines are in a remote cluster with respect to the applications, there is no preference for the message producers and consumers to use a local messaging engine. Each member of the b7cl02.AppTarget cluster, for example, connects to the appropriate bus and uses the remote messaging engine for that bus.

As with a single-cluster topology, if you partition destinations when the applications and messaging engines are in separate clusters, you no longer have the ability to maintain message order. Any time you partition destinations you lose message order, which is true even if you attempt to enable event sequencing in WebSphere Process Server for z/OS.

In addition, partitioned destinations can create additional issues when the messaging engines are remote. By default, you have no control over which active messaging engine your applications use at run time, which can create situations where two applications on the same server attach to two contrasting messaging engines. If one application produces messages for one engine and the message consumer is using a dissimilar engine, stranded messages can result. For these reasons, we strongly discourage partitioned destinations in the two-clusters scenario.

1.3.3 Custom topologies

If neither of these two topologies suits your needs, you can create a custom topology; however, we do not recommend this unless you have a good reason to depart from one of the two topologies that we presented here. There are performance advantages to gain by placing everything in one cluster, and it is always a good idea to keep it simple.

Creating a custom topology with WebSphere Process Server for z/OS involves defining the number of clusters that you want and deploying the components in the appropriate cluster. You must define remote SCA support in clusters that are remote from the SCA message engines. If you choose to implement a custom topology pattern, it is generally unwise to move support components into non-default locations because this structure will likely impact application performance.

1.4 Selecting an appropriate topology

Selecting an appropriate topology for your production environment depends upon several factors:

- Available hardware resources
- Application invocation patterns
- Types of business processes you plan to implement (interruptible versus non-interruptible)
- How heavily you intend to use the CEI
- Scalability requirements
- Administrative effort involved
- Additional application and server requirements

On non-z/OS platforms, where resources are typically not shared between servers because they are running on separate machines, it is important to consider the characteristics of all the hardware when planning your topology. On z/OS however, CPU, memory, data, message queues, and security profiles can all be shared between the logical partitions of a parallel sysplex. Therefore, even though you must plan to have sufficient total capacity on your zSeries machines to support your workload, the topology of the hardware is not something that greatly influences your WebSphere Process Server for z/OS topology choice.

If you can use WebSphere MQ as a message provider, the potential bottleneck of running one message engine in an adjunct disappears. For this reason, if your applications make heavy use of messaging, it is more advantageous to use WebSphere MQ as a message provider than to split the message engines into a separate cluster or implement partitioning.

Our recommendation is to keep the number of clusters to a minimum and think carefully before using more than one. We did not describe the two-clusters topology in response to any known performance problem with the single-cluster topology in WebSphere Process Server for z/OS. Our motivation was related to availability and operability, as we explained in 1.3.2, “Two-cluster topology” on page 15.

Although the two-clusters pattern has advantages, they come at the cost of remote access to the SCA buses. We were unable to perform performance tests to quantify this, and it is possible that the overhead of accessing SCA remotely offsets any benefits of having applications in a separate cluster. In addition, multiple clusters mean more address spaces, which means greater CPU and memory requirements.

As you plan for your production environment, consider the advantages and disadvantages of a single-cluster compared to a two-cluster. In the next sections, we compare these topologies in more detail.

1.4.1 The single-cluster pattern

With WebSphere Process Server for z/OS, a single-cluster topology can be suitable for your production environment. Because each server instance must run both the infrastructure and your applications, the heap requirements of the servant JVMs is more likely to necessitate the use of 64-bit mode than the two-clusters topology.

With WebSphere Process Server for z/OS, you can configure zWLM so that asynchronous interactions are routed to separate servants from those that run synchronous requests, which minimizes the impact of asynchronous requests without resorting to a separate cluster. Unlike non-z/OS systems, where a single cluster topology is typically used for proof of concept, development, and testing environments, a single cluster is a practical production topology on z/OS and is in fact the preferred topology choice.

From an administrative and scalability perspective, the single-cluster topology has advantages. A single cluster, where each member runs all of the WebSphere Process Server components, is easier to administer. Instead of several server instances in multiple clusters, you have a single cluster with fewer members. Remember that with WebSphere Process Server for z/OS, when the needs of your environment grow, you scale by running more servants. It is not necessary to configure additional nodes and cluster members. The number of servants can be set to vary dynamically depending on the workload, so it is not necessary to manually add capacity in response to increased workload.

In a non-z/OS environment, when additional nodes or cluster members are added, all components are scaled at the same rate, for example, each additional cluster member adds more CEI processing, whether you need it or not. If you have the messaging engines spread across server members using service integration policies, there is additional administrative effort in creating and maintaining the policies. With WebSphere Process Server for z/OS, however, adding additional servants provides more capacity for the business workload without adding additional infrastructure overhead.

Expanding the cluster beyond three cluster members has no affect on message processing capability because the one-of-*n* high availability policy limits message processing capacity to the adjunct that is running the active message engine. The other adjuncts host message engines that act as stand-bys. Therefore, use service integration policies to spread the active message engines across the three cluster members, as shown in Figure 1-2 on page 13.

1.4.2 The two-cluster pattern

As we discussed in the opening comments of 1.4, “Selecting an appropriate topology” on page 18, we did not describe the two-clusters topology in response to any performance problem with the single-cluster topology. Our motivation is related to availability and operability. The applications will probably have dissimilar availability needs compared to the support and messaging infrastructure. By placing WebSphere Business Services Fabric for z/OS, BPC, and applications in a separate cluster, those cluster members can be restarted separately from the support and messaging infrastructure.

Creating a second cluster, each with its own functions and applications, creates additional administrative burden, but this is not excessive on z/OS. When you add cluster members and clusters on non-z/OS platforms, your performance tuning plan and the troubleshooting burden can expand greatly, but on z/OS you can classify various workloads using zWLM and obtain detailed performance data on each workload. The topology does not affect your ability to monitor your workloads and give them correct priorities.

Spreading messaging engines across the members of the messaging cluster adds an administrative burden that is associated with creating and maintaining policies, but this is mostly a one-off activity.

In WebSphere Process Server for z/OS, you might not need to add a second cluster for scalability reasons, but having two clusters allows you to scale the cluster members’ hosting applications separately from those hosting the support messaging infrastructure.

Expanding the support and messaging cluster beyond three cluster members has no affect on message processing capability. The scalability of message processing is limited by the workload that can be processed by one adjunct, so the best you can do is spread the active message engines across three cluster members.

In summary, the two-clusters topology in WebSphere Process Server for z/OS offers all of the advantages of the Remote Messaging and Remote Support topology on non-z/OS platforms. Because the application target cluster is only responsible for running your business integration applications, performance tuning and diagnostics are much simpler than in a single-cluster topology where one cluster runs everything. The two-clusters topology is also ideal for environments that make extensive use of CEI for monitoring and auditing because CEI processing does not take place in the same cluster as the applications.

1.4.3 Custom patterns

You can create a custom topology by defining as many clusters as you want and launching the wizards that configure each component of WebSphere Process Server for z/OS on the appropriate cluster. The process of creating such a topology is essentially the same as that for the two-clusters topology that we describe in this book.

If your organization has no need for parts of the WebSphere Process Server for z/OS components, you might want to depart slightly from the topologies that we describe. If you know you will not use the CEI, for example, you can create a topology that does not include CEI support. Similarly, if your organization has governance rules that prevent you from taking advantage of the Business Rules Manager, you can remove it from your topology. These decisions do not significantly affect the choice of topology, except that when you configure fewer components it tends to make the single-cluster topology more practical.

When using the Deployment Environment wizard with a custom pattern, the nodes and clusters must be configured before you start the wizard. When you use the custom pattern, you can then map the component to the existing clusters.

In summary, we recommend that you only consider a custom topology if you have very good reasons for doing so. A single-cluster topology with WebSphere Process Server for z/OS is a good choice. It is possible to imagine reasons to adopt the two-clusters topology that we describe in this Redbooks publication; however, topologies with more clusters than this are animals that you must approach with extreme caution.

1.4.4 Condensed pattern selection criteria

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

Table 1-3 Topology selection considerations

Consideration	Single cluster topology	Two-cluster topology
Number of clusters to maintain	One cluster for all components	► One cluster for applications ► One cluster for the support and messaging infrastructure
Hardware requirements	Minimum footprint for a high-availability production environment	More address spaces means more JVMs and more real storage requirements
64-bit addressing	None. Servers running in 64-bit addressing mode have access to more Java heap	Additional servers running in 64-bit addressing mode have access to more Java heap and require more physical memory to be configured to support the topology
Asynchronous interactions	Use z/WLM to isolate synchronous and asynchronous interactions	Use z/WLM to isolate synchronous and asynchronous interactions
Cross-server communication overhead	None. Everything is running in the same cluster members	Some overhead in accessing remote SCA buses from the application cluster

Consideration	Single cluster topology	Two-cluster topology
Long running processes, state machines, and human tasks	No issue. zWLM is used to assign appropriate priorities	No issue. zWLM is used to assign appropriate priorities
Heavy CEI activity	Heavy CEI use can impact applications	Suitable environment for heavy CEI use, which includes remote CEI connections for WebSphere Business Monitor
Scalability	Easy to scale by adding servant regions. zWLM manages the distribution of work across the servants and can dynamically control the number of servants in response to the workload	Easy to scale by adding servant regions. zWLM manages the distribution of work across the servants and can dynamically control the number of servants in response to the workload. Allows selective scaling of either the application cluster or the support and messaging cluster

1.5 BPM reference topology for z/OS and how we use it here

As its base, the BPM reference topology for z/OS uses the single cluster pattern that we described in 1.4.1, “The single-cluster pattern” on page 18. Figure 1-4 illustrates the BPM reference topology for z/OS.

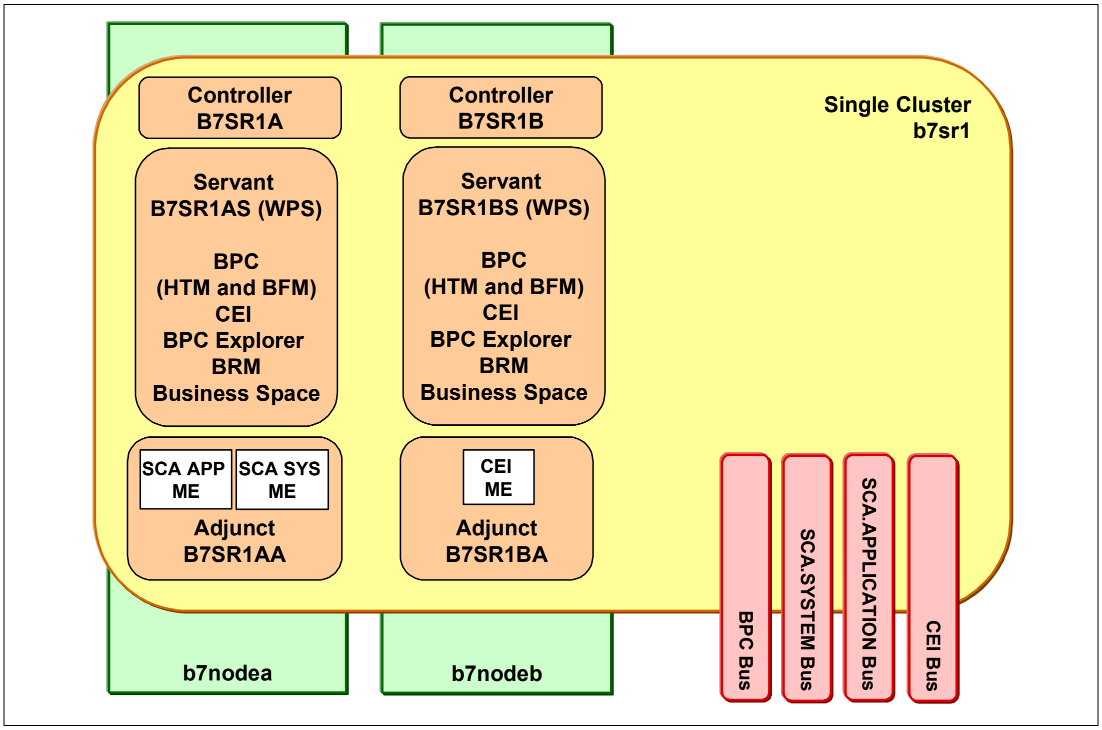


Figure 1-4 BPM reference topology for z/OS

To the pattern in we Figure 1-4 on page 21 add several, additional system components, as shown in Figure 1-5.

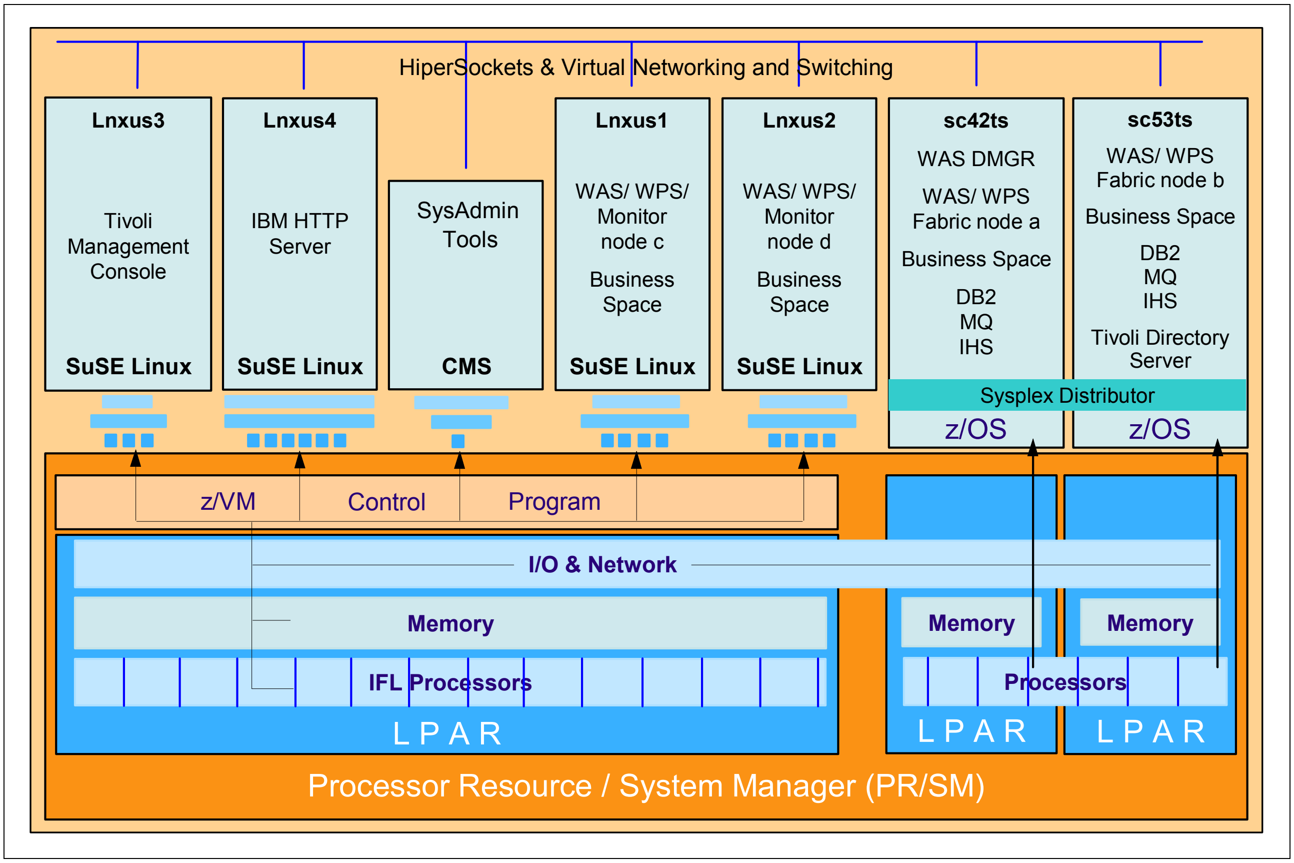


Figure 1-5 BPM reference topology for z/OS

The components and products we use in this book are:

- WebSphere Business Services Fabric for z/OS

  We add the required components and facilities to enable Fabric functions on z/OS. We extend the ITSO-supplied sample application to support business services that are deployed in the Fabric.

- WebSphere MQ shared queues

  We modify the Messaging Engines specified in the two-cluster pattern to be backed by WebSphere MQ shared queues to provide better throughput and scalability.

- WebSphere Business Monitor

  We deploy WebSphere Business Monitor in its own clustered configuration in a pair of Linux for System z LPARs. We federate these nodes and cluster into the cell managed by the WebSphere Application Server for z/OS deployment manager.

- Tivoli Composite Application Manager

  We deploy the Tivoli Management Server in a Linux for System z LPAR. We add ITCAM data collectors to our:

  – WebSphere Process Server for z/OS production nodes
  – WebSphere Business Monitor production nodes
  – WebSphere Business Compass node

The Tivoli Composite Application Manager for WebSphere z/OS 6.1 data collector and the Tivoli Composite Application Manager for Response Time Tracking for z/OS provide management capabilities for our production cell.

- IBM HTTP Server

  We deploy an HTTP server in each of the WebSphere nodes and generate a plug-in configuration so that each node talks to its corresponding HTTP server.

  We configure an additional IHS server in a Linux for System z LPAR to act as a proxy to the HTTP servers on the WebSphere server nodes.

- Sysplex Distributor

  We configure Sysplex Distributor on each of the z/OS nodes to route requests to an available WebSphere node for processing.

- Federated security

  We initially configure security for our cell to be based on RACF and z/OS security. When we add our Linux-based components, we must extend security to them, which we do by deploying a Tivoli Directory Server on z/OS. We use VMM to federate security using Tivoli Directory Server.

1.6 Incorporating WebSphere Business Services Fabric for z/OS into the BPM reference topology for z/OS

A WebSphere Process Server for z/OS production topology can also include other products from the WebSphere business integration portfolio.

1.6.1 WebSphere Business Services Fabric for z/OS modules

WebSphere Business Services Fabric for z/OS contains two software packs:

- Business Services Foundation Pack

  The Business Services Foundation Pack is a set of runtime services and facilities that provide the management and runtime environment for the WebSphere Business Services Fabric for z/OS application set.

- Business Service Tool Pack

  The Business Services Tool Pack is a set of workstation-based tools that provide the design time tools and environment for creating and assembling applications for deployment in the WebSphere Business Services Fabric for z/OS environment. The tool pack provides the necessary tooling for creating and deploying content into WebSphere Business Services Fabric for z/OS. However, because it runs only on the workstation, we only provide limited coverage of it here.

1.6.2 WebSphere Business Services Fabric for z/OS

When WebSphere Business Services Fabric for z/OS is added to the single-cluster topology, the WebSphere Business Services Fabric for z/OS core applications are added to the Single Cluster b7sr1 cluster, as illustrated in Figure 1-6 on page 24.

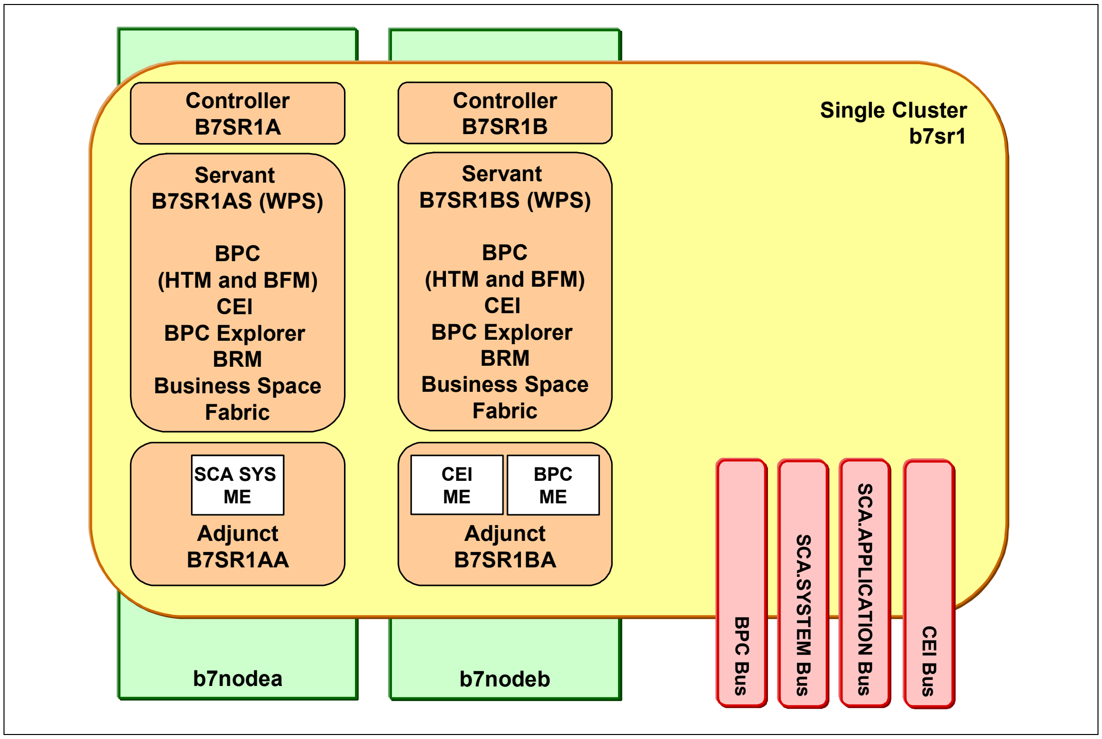


Figure 1-6 WebSphere Business Services Fabric for z/OS in the BPM reference topology for z/OS

Chapter 7, “Adding WebSphere Business Services Fabric for z/OS to an existing z/OS cell” on page 213, provides information about configuring WebSphere Business Services Fabric for z/OS in the BPM reference topology for z/OS.

1.7 Business Space powered by WebSphere

Business Space powered by WebSphere is a component of most of the BPM suite servers, including WebSphere Process Server for z/OS, WebSphere Business Monitor, and WebSphere Business Compass. It provides a common interface for application users to create, manage, and integrate Web interfaces across the IBM WebSphere Business Process Management portfolio.

Business Space powered by WebSphere functions are enabled by default for stand-alone servers, but must be configured manually for clustered implementations, such as the BPM reference topology for System z.

Business Space powered by WebSphere also makes extensive use of REST Services. As part of the configuration process, you must configure the REST Services application, REST services endpoints, and the required Rest Services EJBRoles.

2

Planning for and preparing WebSphere Process Server

In this chapter, we describe how to plan for and prepare a production WebSphere Process Server for z/OS configuration in a network deployment configuration. We provide instructions that describe the decisions and tasks to be made by the z/OS Systems Programmer, the Security Administrator, and the Database Administrator.

In this chapter, we discuss:

In subsequent chapters, we describe how to configure the production topology.

2.1 Overview of configuring WebSphere Process Server

Configuring WebSphere Application Server for z/OS and WebSphere Process Server for z/OS requires that you understand the z/OS runtime and the various components that make up these products.

There are two ways to approach the task of configuring a WebSphere Process Server for z/OS cluster:

- Start with a deployment manager and an empty node. Augment both with WebSphere Process Server for z/OS, federate them, and add your clusters.
- Start with a stand-alone WebSphere Application Server for z/OS server, augment it with WebSphere Process Server for z/OS, create a deployment manager, augment it with WebSphere Process Server for z/OS, and federate them. See *z/OS: WebSphere Business Process Management V6.2 Production Topologies*, SG24-7733 for using this approach.

To configure WebSphere Process Server for z/OS you must understand the z/OS runtime and the various components that make up these products.

2.1.1 Process of building a production WebSphere Process Server for z/OS

In this chapter, we describe the procedure that we recommend when building a production WebSphere Process Server for z/OS cell:

1. Plan.
2. Create the deployment manager and empty node.
3. Augment the deployment manager and empty node with WebSphere Process Server for z/OS.
4. Create DB2 objects.
5. Federate the empty node into the deployment manager's cell
6. Create a cluster with the deployment environment wizard (or the SCA, CEI, and BPC wizards.)
7. Validate, backup, and prepare to extend the cell with other BPM products.

2.1.2 Extending production topologies

After you create the WebSphere Process Server for z/OS cluster, you can extend the cell using:

- WebSphere Business Services Fabric for z/OS
- WebSphere Business Monitor

2.2 Planning the production topology

A great deal of planning is crucial to the successful implementation of a BPM topology. This planning will pay for itself in more efficient configuration, operations, systems management, and troubleshooting. In this section, we describe the following planning areas:

- Configuration topologies
- Hardware requirements, such as CPUs, memory, disk storage, and so on
- Names for cells, nodes, servers, clusters, user IDs, group IDs, and so forth

- z/OS dataset names and space allocations
- Security identities, aliases, and roles
- DB2 names for databases, storage groups, schemas, buffer pools, and GRANTs
- UNIX® file system directories and files (ownerships and permissions)

It is crucial to carefully plan where you are going so that you do not have to re-work components along the way.

2.2.1 Draw a picture

As you plan your topology, make a graphic design of your WebSphere Application Server for z/OS cell with nodes, servers, clusters, and their respective names. The names of your cells, nodes, servers, and clusters deserve careful consideration to identify meaningful use of these components and allow for growth with additional servers, clusters, nodes, and LPARS.

Post this picture or diagram for all to see. As you continue through the steps, you need a spreadsheet to plan out a consistent set of names and server. Use the spreadsheet and diagram as a reference throughout the configuration tasks. Figure 2-1 provides an example of a diagram.

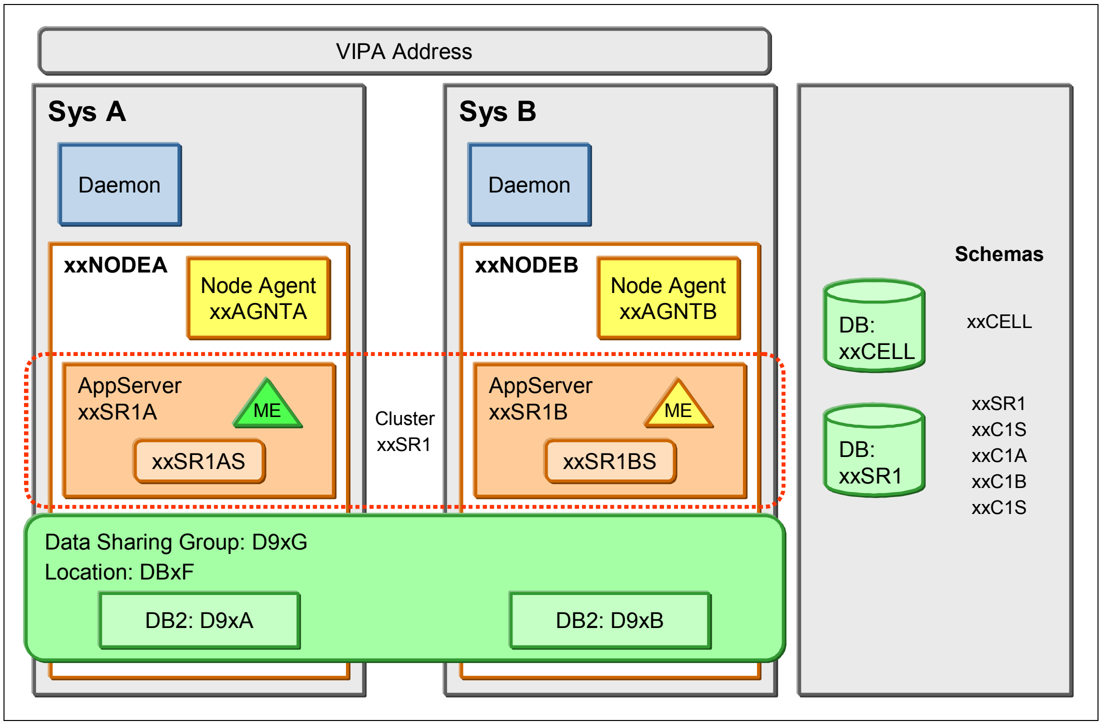


Figure 2-1 Draw a picture

2.2.2 Using a spreadsheet to plan your configuration

The spreadsheet that accompanies this book helps you to enter all of the configuration variables for your environment, and it will also help when you customize your configuration files. The spreadsheet will help enforce consistency between the required configuration tasks and also help to eliminate typographical and consistency errors.

The spreadsheet is divided into distinct worksheets, which are identified by the tabs along the bottom, as illustrated in Figure 2-2 on page 28 and Figure 2-3 on page 28.

Figure 2-2 Spreadsheet worksheet names

The worksheet categories in the spreadsheet are:

Documentation Information about this version of the Spreadsheet and instructions for its use.

Variables Start here by entering the basic variable names for your configuration, including DB2 and LDAP.

Names All of the naming convention rules are implemented in formula on this worksheet. Check these after entering the variables to make sure that you are comfortable with them in your environment. Alter the formula to match your preferred naming convention.

Checklist This is an ordered list of tasks to configure your WebSphere Application Server for z/OS with WebSphere Process Server for z/OS.The Checklist directs you to other worksheets at the right time.

DMGR, and so on Use these response files for use with the WCT/zPMT for deployment manager, empty nodes, standalone server, IHS, LDAP, and Job Managers.

JCL_P and JCL_S Sample jobsteams to help you with the manual steps to create RACF definitions, run shell scripts, DB2 definitions, and more.

Figure 2-3 Spreadsheet worksheet names

WPS_RACF User and group names to be defined in RACF profiles for WebSphere Process Server for z/OS.

WPS_LDAP An ldif file based on the users and groups that you planned on the WPS_RACF worksheet for loading into an LDAP server.

DE Sample xml file for defining the deployment environment cluster.

dbDesign Sample database design document to use with the DbDesignGenerator.sh script.

BPXPRM_P and _S MOUNT statements for the cell configuration datasets to be added to the SYS1.PARMLIB(BPXPRMxx) member.

IHS A checklist for configuring IBM HTTP Servers on z/OS.

Familiarize yourself with these samples to save you the effort of typing them up or copying and customizing from other sources.

Roadmap for using the spreadsheet

The chapters in this book refer to the spreadsheet where necessary and provide detailed information about each task; however, a high-level *roadmap* of how you will use the spreadsheet is useful for the first time that you use the spreadsheet.

All of the worksheets except the Variables worksheet are protected against update by the trivial password WPS, which protects against accidental damage. The spreadsheet generates names based on the naming convention described in Techdoc WP100653; however, the formula to implement that convention are all on the Names worksheet. The other worksheets lookup any ‘names’ that they need from the Names worksheet. Therefore, if you do not like the naming convention that is built-in to the spreadsheet, just un-protect the Names worksheet and alter the formula there.

If you are considering changes to the formula but you are not very experienced with EXCEL, we recommend that you find someone who is, and explain to them what you want.

Let us assume that you accept the names generated on the Names worksheet. All you need to do is complete the dark green fields of the Variables worksheet. Many of the default choices are what we consider to be best practice or a good choice, and if you make different choices, warning messages might appear. These messages alert you when your choice is not the normal one and hopefully help you to think a little more about that choice.

After you complete the Variables worksheet, go to the Checklist. It is customized with instructions for building the cell according to the choices that you made on Variables. The Checklist is divided into sections that contain the Chapter heading in this book for that task.

Using the Variables sheet

This is the starting point to tailor your copy of the spreadsheet by describing your particular environment and cell configuration. Here you specify:

- The WPS configuration scenario and topology
- z/OS sysplex, system, host, and LPAR names
- Product HFS mount points
- General security implementation plans
- TCP/IP Port allocations
- DB2 subsystem configuration names

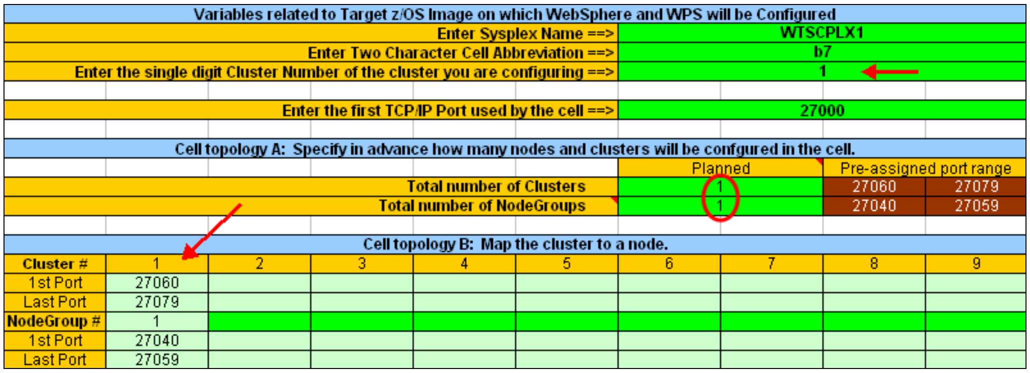
Start by completing the Variables worksheet. Choose from options or enter your names for data set high-level qualifiers, UNIX mount points and networks names, for example. If you already used the spreadsheets provided with Techdocs PRS1331 and PRS3341 to configure a WebSphere Application Server for z/OS cell, you will find this straightforward. However, there are a couple of differences on the Variables worksheet for planning a WebSphere Process Server for z/OS cell that deserves more explanation.

The standard spreadsheets for WebSphere Application Server for z/OS assume that you will configure only one node per LPAR, but your WebSphere Process Server for z/OS cell might have multiple clusters, some with WebSphere Process Server for z/OS and some running J2EE applications only. When you have a cell running several cluster with different workloads, you might want to configure one or more of those clusters in separate nodes because it is the *node* that is configured to run on a particular service level of WebSphere Application Server for z/OS. If you want to run one cluster, one service, and a second cluster on a different service level, you will need two nodes for each LPAR.

Each node must refer to the WebSphere Application Server for z/OS product file system (and to the product file system of Feature Packs or WebSphere Process Server for z/OS) using an intermediate symbolic link, which we discuss in 2.4.4, “Using intermediate symbolic links” on page 48. You can switch a node from one service level to another by re-defining the intermediate symbolic link to point to the mount point of the new service level. This is useful if you think that WebSphere Process Server for z/OS might require a recent WebSphere Application Server for z/OS PTF or a temporary APAR fix, which you do not want to apply to the clusters that are not running WebSphere Process Server for z/OS. If you configure WebSphere Process Server for z/OS in one node and the non-WPS clusters in a different node, you can install the fix on the WebSphere Process Server for z/OS cluster but not on the other clusters.

The spreadsheet helps you to plan the cell with multiple clusters configured in one or more *node-groups*. A node-group is a term that describes the nodes that run related cluster members across the Sysplex, for example, if you have two clusters and two node-groups across three LPARs, there will be two nodes in each LPAR, which makes six nodes in total.

Two tables near the top of the Variables worksheet are where you plan clusters and node groups. Figure 2-4 shows the tables with the default choice of one node group and one cluster.



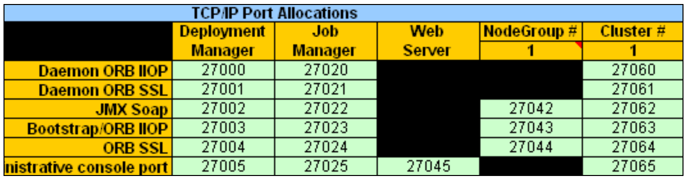
Variables related to Target z/OS Image on which WebSphere and WPS will be Configured			
Enter Sysplex Name ==>	WTSCPLX1		
Enter Two Character Cell Abbreviation ==>	b7		
Enter the single digit Cluster Number of the cluster you are configuring ==>	1		
Enter the first TCP/IP Port used by the cell ==>	27000		
Cell topology A: Specify in advance how many nodes and clusters will be confgured in the cell.			
	Planned	Pre-assigned port range	
Total number of Clusters	1	27060	27079
Total number of NodeGroups	1	27040	27059

Cell topology B: Map the cluster to a node.									
Cluster #	1	2	3	4	5	6	7	8	9
1st Port	27060								
Last Port	27079								
NodeGroup #	1								
1st Port	27040								
Last Port	27059								

Figure 2-4 Node-group and cluster topology with one node-group and one cluster

Use the table Cell Topology A to pre-plan the fact that the cell will contain only one node-group, and there will be only one cluster. In table Cell Topology B, the port ranges of the Node Agents and the cluster members are shown. In the second green field of Figure 2-4, the cluster number that is being configured is 1 because there will be only one cluster. An important difference between the WebSphere Process Server for z/OS spreadsheet and the standard spreadsheet for WebSphere Application Server for z/OS is that when you want to configure multiple clusters, you must use the `Enter the single digit Cluster Number you are configuring field` because this field controls which cluster the other worksheets apply to. If you must configure three clusters in this cell, you start with this field set to 1, complete the configuration of that cluster, and then change that field to 2 and configure the second cluster.

One of the main things that these tables do is to plan out the port allocations for Node Agents and Cluster Members (servers). The port ranges that are assigned here are used in the table of TCP/IP ports, which is located further down the Variables worksheet, as shown in Figure 2-5.



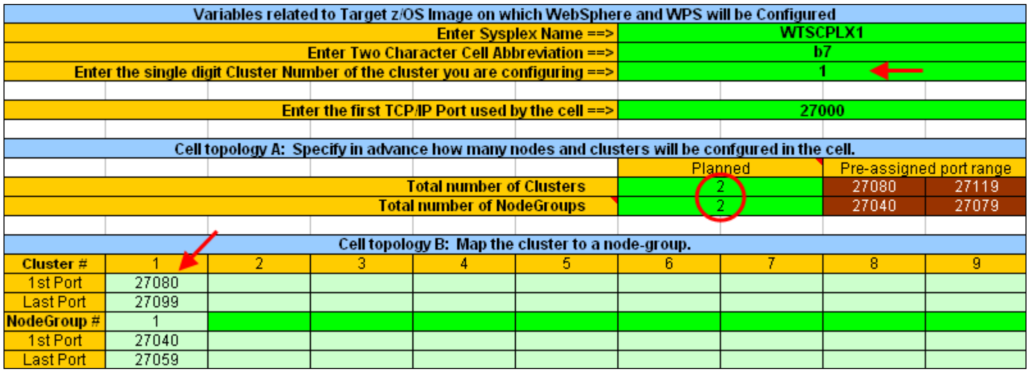
TCP/IP Port Allocations					
	Deployment Manager	Job Manager	Web Server	NodeGroup # 1	Cluster # 1
Daemon ORB IIOP	27000	27020			27060
Daemon ORB SSL	27001	27021			27061
JMX Soap	27002	27022		27042	27062
Bootstrap/ORB IIOP	27003	27023		27043	27063
ORB SSL	27004	27024		27044	27064
nistrative console port	27005	27025	27045		27065

Figure 2-5 Port ranges based on the choices in Cell Topology A and Cell Topology B

The simple case of one node-group and one cluster differs from a topology with more than one node-group in one other way. The naming conventions that are used on the Names worksheet in this simple case follow the same naming conventions as the standard WebSphere Application Server for z/OS spreadsheet. They both use the single character LPAR identifier as the node suffix. So, for example, you would have a Node Agent name B7AGNT**A** that is running in LPAR A, which is not sufficient when there are multiple node-groups in the cell, and two digits must be used to generate unique Node Agent and Cluster Member names.

The easiest way to understand how this works is to make various choices in the Cell Topology A and Cell Topology B tables, and then look at the names that are generated on the Names worksheet.

Let us look at an example where your cell contains two clusters, each in their own Node Group. One cluster might be running WebSphere Process Server for z/OS and the other might be a normal WebSphere Application Server for z/OS cluster. Imagine that you decided that you want them in different node-groups in order to have the flexibility to run them at different service levels. The Cell Topology tables would start off looking like Figure 2-6.



Variables related to Target z/OS Image on which WebSphere and WPS will be Configured

Enter Sysplex Name ==>	WTSCPLX1
Enter Two Character Cell Abbreviation ==>	b7
Enter the single digit Cluster Number of the cluster you are configuring ==>	1
Enter the first TCP/IP Port used by the cell ==>	27000

Cell topology A: Specify in advance how many nodes and clusters will be confgured in the cell.

	Planned	Pre-assigned port range	
Total number of Clusters	2	27080	27119
Total number of NodeGroups	2	27040	27079

Cell topology B: Map the cluster to a node-group.

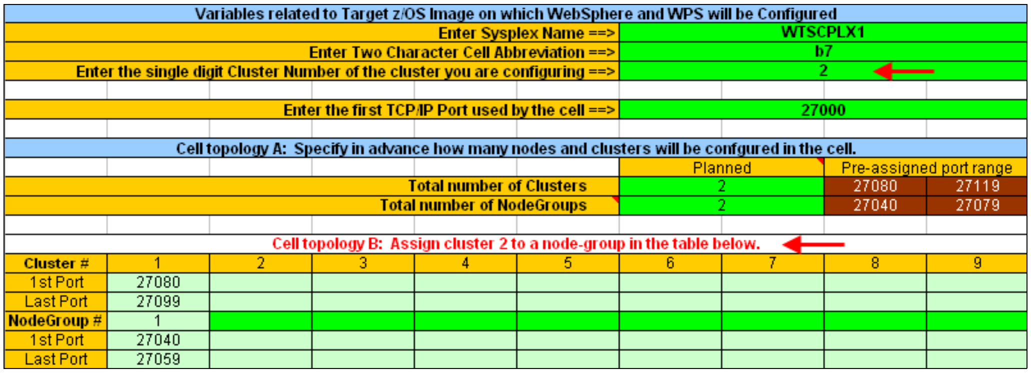
Cluster #	1	2	3	4	5	6	7	8	9
1st Port	27080								
Last Port	27099								
NodeGroup #	1								
1st Port	27040								
Last Port	27059								

Figure 2-6 Planning for two clusters and two node-groups and configuring the first cluster

In Figure 2-6, notice that the brown fields on the right of table Cell Topology A show the total port range that the two node-groups and two clusters use. Initially, ports are assigned in table Cell Topology B for only the first cluster. Notice that because there are two node-groups, the port range for the first cluster starts at 27080 instead of 27060 because of the extra ports used by the Node Agents that manage the second node-group.

Also, if you take a look at the Names worksheet, you will find that the Node Agents now have names, such as B7AGT1A instead of B7AGNTA, to cater for multiple nodes per LPAR.

Now, if you choose to configure the second cluster by entering a 2 in the Enter the single digit Cluster Number you are configuring field, a warning message tells you that you must map the second cluster to a node group in table Cell Topology B, as shown in Figure 2-7.



Variables related to Target z/OS Image on which WebSphere and WPS will be Configured

Enter Sysplex Name ==>	WTSCPLX1
Enter Two Character Cell Abbreviation ==>	b7
Enter the single digit Cluster Number of the cluster you are configuring ==>	2
Enter the first TCP/IP Port used by the cell ==>	27000

Cell topology A: Specify in advance how many nodes and clusters will be confgured in the cell.

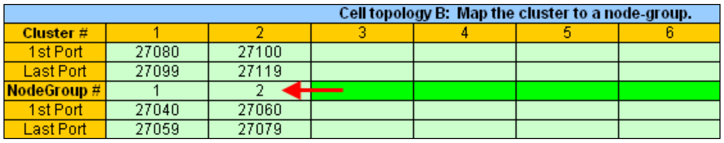
	Planned	Pre-assigned port range	
Total number of Clusters	2	27080	27119
Total number of NodeGroups	2	27040	27079

Cell topology B: Assign cluster 2 to a node-group in the table below.

Cluster #	1	2	3	4	5	6	7	8	9
1st Port	27080								
Last Port	27099								
NodeGroup #	1								
1st Port	27040								
Last Port	27059								

Figure 2-7 Planning for two clusters and two node-groups and configuring the second cluster

Assign the second cluster to a Node-group in Cell Topology B, as shown in Figure 2-8 on page 32.



Cell topology B: Map the cluster to a node-group.

Cluster #	1	2	3	4	5	6
1st Port	27080	27100				
Last Port	27099	27119				
NodeGroup #	1	2				
1st Port	27040	27060				
Last Port	27059	27079				

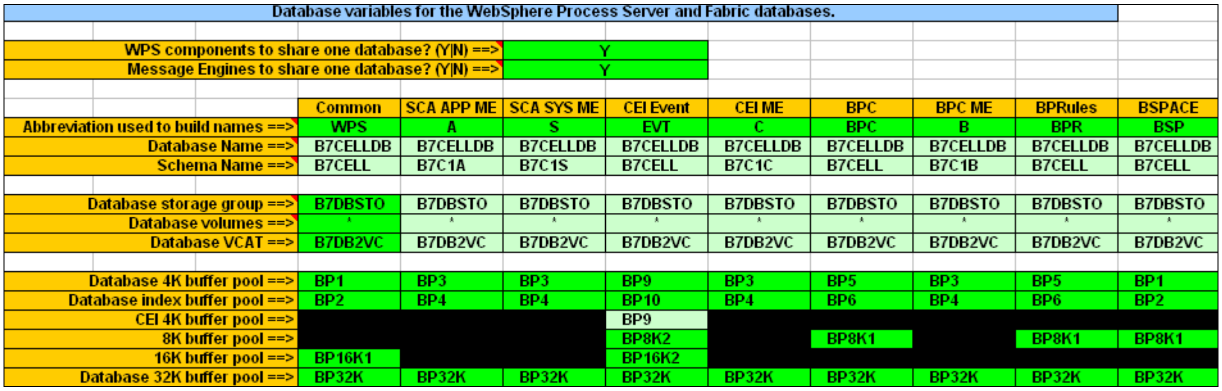
Figure 2-8   The second cluster assigned to node-group 2

If you do not know what you need when you start

What if you do not know what you need when you start? If you know that you will configure multiple clusters in the cell, but you do not know whether you will need more than one node-group, we recommend that you plan to have two node-groups. Remember that a node-group runs on a particular service level WebSphere Application Server for z/OS, so if your cell has four clusters, you are unlikely to want to run each on a different service level. Four clusters in four node-groups is unusual. It is more likely that you would run two clusters on one service level and two on another. If you plan for more than one node-group and end up only using one, all you waste are some ports that the spreadsheet reserves for the unused node-group, so the safe choice is to choose at least two node groups.

Similarly, if you think you will use only one cluster but you are not sure, it might be a good idea to plan the cell now for two node-groups and two clusters. If, in the future, you come to need more than one cluster, it is likely that the business reason for that might imply the need to run that cluster on a different service level to the other. So, in the absence of any solid planning information about the cell topology, a safe decision is to plan for two node-groups and two clusters.

The other table on the Variables worksheet, which deserves more explanation, is the table in Figure 2-9 that plans the names of the objects that WebSphere Process Server for z/OS will create in DB2 for z/OS.



Database variables for the WebSphere Process Server and Fabric databases.

WPS components to share one database? (Y|N) ==>	Y
Message Engines to share one database? (Y|N) ==>	Y

	Common	SCA APP ME	SCA SYS ME	CEI Event	CEI ME	BPC	BPC ME	BPRules	BSPACE
Abbreviation used to build names ==>	WPS	A	S	EVT	C	BPC	B	BPR	BSP
Database Name ==>	B7CELLDB	B7CELLDB	B7CELLDB	B7CELLDB	B7CELLDB	B7CELLDB	B7CELLDB	B7CELLDB	B7CELLDB
Schema Name ==>	B7CELL	B7C1A	B7C1S	B7CELL	B7C1C	B7CELL	B7C1B	B7CELL	B7CELL
Database storage group ==>	B7DBSTO	B7DBSTO	B7DBSTO	B7DBSTO	B7DBSTO	B7DBSTO	B7DBSTO	B7DBSTO	B7DBSTO
Database volumes ==>	*	*	*	*	*	*	*	*	*
Database VCAT ==>	B7DB2VC	B7DB2VC	B7DB2VC	B7DB2VC	B7DB2VC	B7DB2VC	B7DB2VC	B7DB2VC	B7DB2VC
Database 4K buffer pool ==>	BP1	BP3	BP3	BP9	BP3	BP5	BP3	BP5	BP1
Database index buffer pool ==>	BP2	BP4	BP4	BP10	BP4	BP6	BP4	BP6	BP2
CEI 4K buffer pool ==>				BP9					
8K buffer pool ==>				BP8K2		BP8K1		BP8K1	BP8K1
16K buffer pool ==>	BP16K1			BP16K2					
Database 32K buffer pool ==>	BP32K	BP32K	BP32K	BP32K	BP32K	BP32K	BP32K	BP32K	BP32K

Figure 2-9   The table on Variables for planning DB2 object names

An important point to remember about the table in Figure 2-9 is that although the formula generates the default values you will see based on choices that you made further up the worksheet, you can overtype any of the green fields and put in the value that you want. The formula in the spreadsheet is there to generates sensible defaults.

Also, a fundamental decision that you must make is whether all of the WebSphere Process Server for z/OS components will share the database in DB2 or whether the different components will have their tables assigned to different databases. We discuss this choice in detail in 2.6.1, “Designing the database configuration” on page 54. The default choice is to define all components in one database.

Notice also in Figure 2-9 on page 32 that under the ‘Common’ column the field that chooses the characteristics for the DB2 STOGROUP are dark green; whereas, for the other components they are light green. This different coloring is because the STOGROUP values that you set for the ‘Common’ component will become the defaults for the other components, which happens regardless of whether you choose to assign components to multiple databases. This behavior is because in most cases one STOGROUP is sufficient. If you do need different STOGROUPs for each component then type in the names that you want.

The values in this table are used on several spreadsheets, but the most important user of this data is the dbDesign worksheet. That worksheet controls how WebSphere Process Server for z/OS wizards configures JDBC resources in WebSphere and how Data Definition Language (DDL) statements are generated to create all of the necessary objects in DB2.

Using the Names sheet

After you complete the Variables sheet, take a look at the Names sheet to see how the variables affect job names and procedure names, users, groups, UNIX identities, homes, and other names that will be used in the subsequent configuration sheets.

If you do not like the naming convention, you can change the formula by un-protecting the Names worksheet, as shown in Figure 2-10. Enter the password `WPS` when prompted.

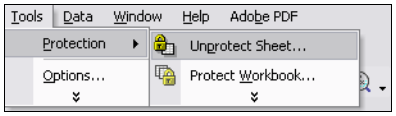


Figure 2-10 Unprotecting the Names worksheet; enter the password ‘WPS’ when prompted.

Although it is always best to change the formula to match your naming convention, if you do not understand the EXCEL formula, you can always just type over the fields on the Names worksheet so that they have the values that you want. We do not recommend typing over, but since the other worksheets will pick up names from the Names worksheet, it does not matter whether the final value is derived from a formula or just typed in; however, changes to other worksheets are likely to have bad consequences somewhere later, so avoid over typing.

Using the Checklist sheet

The Checklist is a step-by-step guide to the tasks that you must perform to complete the configuration. It is an abbreviated form of the instructions that we provide in this book. When you need to use one of the other worksheets, the Checklist tells you what to do, and when it refers to the sample JCL provided on the JCL_P and JCL_S worksheets, there are hyperlinks to take you to the correct JCL for that task. Using the Status column, you can record your progress, which makes it easier to share the work in your team. The Checklist sheet can also be useful as an auditable document that describes what you did. You can make notes in Column B, if necessary. Hyperlinks are provided that take you to the sample JCL when the Checklist refer to it.

Using the remaining sheets

In this book, we describe how to use the other sheets that are identified by the *tabs* as you work through the configuration phases.

Several of the worksheets generate *response files*, which are used as input to the z Profile Management Tool (zPDT) that runs in the WorkStation Customization Tool (WCT). This is the tool that you use to generate the jobs that configure the WebSphere Application Server for z/OS cell and generate the jobs that augment the nodes for the Feature Packs and for WebSphere Process Server for z/OS. The Checklist directs you to the appropriate worksheet for each task.

Using sample jobs in the JCL_P sheet

There are dozens of sample jobs provided that can save you a lot of time if you copy them into separate members of a partitioned dataset with the ISPF editor. Then you can use these jobs and other data files that you use to configure your environment with a minimum of typing.

> **Note:** The job names are prefixed with the cell prefix (shown as B7 in the following list) and some jobs contain the node suffix. Therefore the actual job names on the JCL_P and JCL_S worksheets differ from the names in this section depending on your choice of naming convention for the node suffix.

B7RAC1TM Sets up RACF profiles one time in the sysplex: From Techdoc WP101472 at:
`http://www.ibm.com/support/techdocs`

B7RAC700 Defines generic profiles for all servers in cell B7CELL.

B7UNDORA Undoes the B7RAC700 definitions.

B7UNIXPR Issues RACF PERMIT commands to allow the installer to configure WebSphere without needing UID(0).

B7SIGNA Retrieves certificate signers into node when not using SAF security.

B7RACFG Adds groups and user IDs when you decide to use GEJBROLE profiles.

B7HOMEG Creates Home directories for the user IDs that are created by job B7RACFG.

B7GEJB Defines GEJBROLE profiles for WPS. Alternative to B7EJB.

B7RACFE Adds groups and user IDs when you decide to use EJBROLE profiles.

B7HOMEE Creates home directories for the user IDs that are created by job B7RACFE.

B7EJB Defines EJBROLE profiles for WPS. Alternative to B7GEJB.

B7BAKDMG Backs up the zFS for the DMGR.

B7DMINS Runs zWPSInstall.sh for the DMGR's node.

B7DMAUG Runs zWPSConfig.sh for the DMGR's node.

B7BAKNA Backs up the primary empty node.

B7NAINS Runs zWPSInstall.sh for the primary empty node.

B7NAAUG Runs zWPSConfig.sh for the primary empty node.

B7BAKZFS Backs up zFS filesystems.

B7RSTZFS Restores zFS filesystems.

B7CRDB1 Creates the database. The job name is B7CRDBS, if you choose to configure the components into separate databases.

B7GENDDL Generates the DDL/SQL.

B7GENGRA Runs gengrant.sh in batch to generate GRANT statements.

B7DL2PDS Copies generated SQL statements to the MVS™ partitioned dataset.

B7NAPOR Runs updNewServer.py to set the ports for the server.

B7INVARS Initializes WebSphere Application Server variables, as described in Techdoc TD103685, to reduce the amount of messages that are written to SYSLOG and that use local time in messages.

B7CERTS Lists RACF Certificates on keyring for B8ACRU.

B7CLOSGI Clears the OSGI cache.

B7WPSZFS Defines and mounts a small ZFS for the wpswork directory.

@IHSDHA Creates mount point for the IBM HTTP Server configuration.

@IHSDSA Creates the intermediate symbolic link to the IBM HTTP Server product.

@IHSINA Runs install_ihs for the IBM HTTP Server.

@IHSP1A Creates an installation directory for the plug-in.

@IHSP2A Configures the IBM HTTP Server to use the WebSphere Application Server plug-in.

@IHSSTC Creates the started profile for the HTTP Server.

@IHSWSA Runs configureWebserverDefinition.jacl.

Some of the jobs that are listed in this section apply to secondary nodes. The JCL_S worksheet contains sample JCL that is customized for a secondary node.

2.3 Preparing the z/OS system

The latest information about the supported hardware and software configurations for WebSphere Application Server for z/OS are on the following Web page:

http://www.ibm.com/support/docview.wss?rs=2307&uid=swg27016269

2.3.1 Hardware requirements

After you plan the topology of your servers, you need sufficient resources for the configuration that you chose to assure adequate performance, availability, growth, and systems management characteristics.

See techdoc WP101620, *Basic Sizing for a WebSphere Application Server on z/OS*, for an overview of “How much memory and how much DASD do I need for WAS z/OS?” at http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101620

CPU resources

There are capacity planning tools that can help you to size your processor requirements for WebSphere Process Server for z/OS.

Contact your IBM Representative or Business Partner for information about these sizing tools. IBM business partners can also contact PartnerWorld®.

zPSG is a PC-based sizing tool for estimating the IBM System z capacity that is required to support new workloads that are not yet running on System z. Sizing support is provided for both z/OS and Linux operating system environments. The workloads that are currently supported are WebSphere Application Server, WebSphere Portal Server, WebSphere

Process Server, WebSphere Message Broker, WebSphere MQ, DB2 for z/OS OLTP Transactions, and Apache web server.

We recommend a minimum of two processors per LPAR to minimize the overhead of switching between address spaces in a WebSphere Application Server Network Deployment for z/OS environment.

> **Note:** The z/OS operating system is extremely good at assigning CPU to the highest priority task, but it is important that you do not run a WebSphere Process Server for z/OS cell in an LPAR with insufficient CPU or at too low a priority.
>
> The high availability (HA) manager component of WebSphere Application Server for z/OS issues heart-beat messages to members of the HA core group, but if CPU is so short that a core group member does not get dispatched for several seconds, the HA manager can think that component failed; therefore, the HA manager might initiate an unnecessary recovery action that consumes more CPU and will probably fail.
>
> If you see CPU Starvation messages in your controller or Node Agent logs, this is an indication that there is a severe CPU shortage that will generate problems and a lot of unnecessary work for you. Therefore do not starve WebSphere Process Server for z/OS of CPU. Ensure that zWLM classification rules assign sufficient priority to the cell and run controllers in an STC service class that has a greater velocity goal than the service class that is assigned to servants. The adjunct is considered a controller for these purposes.

IBM System z Application Assist Processors (zAAPs)

zAAPs are lower-priced specialized processing units that provide a strategic z/OS Java execution environment for customers who want cost-effective processing engines and traditional qualities of service of the IBM mainframe platform. zAAPs are processors out of the general processor pool that have special licensing terms to allow Java to run on z/OS with lower total cost of ownership (TCO). They run at full-rated processor speed and are not capped according to machine type and rating. They take advantage of the technology dividend (at no cost) as machine designs progress, for example, zAAP on z10™ is 35% faster than comparable zAAP on z9®, at no additional cost to customers.

WebSphere Application Server for z/OS and WebSphere Process Server for z/OS offer full support for zAAPs. zAAPs are designed to operate asynchronously with the general purpose processors when executing Java programming under control of the IBM JVM.

IBM System z Integrated Information Processor (zIIP)

zIIPs are very similar in concept to zAAPs. There are several workloads that are eligible for the zIIP, including DB2, z/OS Communication Server, z/OS XML System Services, and zAAP on zIIP capability. z/OS V1.11 is enhanced with a new capability that can enable zAAP-eligible workloads to run on zIIPs. This new capability is intended to help customers that have only zIIP processors by making Java and XML-based workloads eligible to run on existing zIIPs. Customers who already invested in zAAP, or have invested in both zAAP and zIIP processors, can continue to use them to maximize the new workload potential for the platform. This new capability is not available for z/OS LPARS if zAAPs are installed on the server.

Real memory resources

The real memory requirements for WebSphere Application Server for z/OS and WebSphere Process Server for z/OS depends largely on the size and number of JVM heaps in your servers. They do not tolerate paging very well. If the installation has a very robust paging subsystem, a small amount of demand paging can usually be tolerated.

31-bit versus 64-bit

Virtual storage demands also affect the real storage requirements. z/OS has 31-bit addressing modes and 64-bit addressing modes (not 32-bit and 64-bit modes), which means that when in 31-bit mode, the useful addressability is greatly reduced to less than a 1 GB address space. Most z/OS installations cannot get more than 768 MB of virtual storage for their private address spaces.

Any customer that needs a JVM with a heap greater than 1.0 gigabyte needs a 64-bit JVM because there simply is not enough room in a z/OS address space for a JVM with that heap requirement. Even if you do not have a need for a heap in excess of 1 GB, there might be a need for 64-bit JVMs to alleviate native storage issues.

The default mode in WebSphere Application Server for z/OS Version 7 is 64-bit. 31-bit mode is deprecated. New features in the IBM Developer Kit for Java 6 for 64-bit support can improve performance with *Large Page Support* and *Compressed references*:

- New hardware feature introduced with IBM System z10®
- Allocates and manages memory in 1M chunks
- New method for managing object pointers with the JVM
- Reduces the size of the 64-bit object pointer to 4 bytes
- Designed to use caching within the machine to improve performance

For a description about configuring WebSphere Application Server for z/OS for 64-bit addressing, see *The z/OS: WebSphere Business Process Management V6.2 Production Topologies*, SG24-7733.

Disk storage

In addition to the DASD volumes that are used to hold the product code and file system, you will need additional DASD storage to hold configuration data for application serving environments. The amount of storage depends on the number of environments and the size and complexity of the applications that are being deployed.

A cell that is running WebSphere Process Server for z/OS and WebSphere Business Services Fabric for z/OS requires a large configuration dataset for each node. After configuring WebSphere Business Services Fabric for z/OS in a WebSphere Process Server for z/OS network deployment configuration, each node's configuration zFS (both the deployment manager and the managed node) occupied less than 1000 cylinders of 3390 DASD space when first configured but grew to over 1300 cylinders after WebSphere Process Server and Fabric were added.

Summary of our DASD space requirements

The following list shows roughly how many 3390 cylinders our configurations took. Your numbers will vary depending on the number and size of your applications, servers, activity and persisted data:

Config. zFS with WAS - DMGR	500
Config. zFS with WAS - Node	500
Config. zFS with WPS - DMGR	1200
Config. zFS with WPS - Node	1000
Config. zFS with Fabric - DMGR	1300
Config. zFS with Fabric - Node	1100
Total - DMGR + 2 Nodes	3500
Backup Config. zFS Files	Approx. 90% of the Config. zFSes

Total - Backup DMGR + 2 Nodes	3200
DB2 Tables - Default allocations	5000
Product zFS - WAS	3500
Product zFS - FPs + IHS	200
Product zFS - WPS	1100

We show values from our experiences near the bottom of the Variables worksheet in the spreadsheet that is used to plan the cell.

When you have large datasets like these, dumping them can be tricky because a dump of all the configuration datasets on disk can easily exceed 3500 CYL. We suggest that you speak to your disk space administrator about setting up a pool of 3390-9 disks to hold the configuration datasets and dumps of the WebSphere Process Server for z/OS cell.

You must also request that DFSMS maintain sufficient free space on these volumes to allow the node configuration datasets to extend.

Running out of space on a volume when configuring WebSphere Process Server for z/OS and WebSphere Business Services Fabric for z/OS is the cause of many strange problems and often results in having to restore the configuration to the point before the failure.

Plan the dataset high-level qualifier that the configuration zFS filesystems use because your DFSMS ACS routines must map the high-level qualifier to the storage pool of 3390s.

Workstation tools

You need a workstation to support many of the configuration and administration activities in this book, such as:

- 3270 Emulator (IBM Personal Communications)
- Telnet client (PuTTY, TeraTerm)
- FTP clients (WS_FTP Language Environment®)
- Web browser (Internet Explorer, Firefox)
- PDF Reader (Adobe®)
- Text File Editor (PFE, Notepad, or SlickEdit)
- WCT zPMT (z/OS Profile Management Tool)
- WID (WebSphere Integration Developer)

WebSphere Integration Developer

WebSphere Integration Developer Version 7 is required for use with WebSphere Process Server for z/OS version 7.

> **Note:** It is important to use a level of WebSphere Integration Developer that is compatible with the level of WebSphere Process Server for z/OS that you are running.

For a tutorial about installing WebSphere Integration Developer, see the IBM Education Assistant (IEA) at this Web site:

`http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.wasfpsca/wasfpsca/1.0/Installation/WASv7SCA_zOS_Installation/player.html`

Linux systems

In addition to the tools listed for supporting the z/OS components of the configuration, you might need additional tools for interacting with Linux on System z:

- Remote desktop client (VNC): RealVNC provides remote control software that lets you see and interact with desktop applications across any network. Visit the following Web site:

  `http://www.realvnc.com/`

- FTP clients: FileZilla is a free FTP solution, with both a client and a server version. FileZilla is open source software that is distributed free of charge under the terms of the GNU General Public License. Visit the following Web site:

  `http://filezilla-project.org/`

2.3.2 Software requirements

Several WebSphere products are required to construct a WebSphere Process Server for z/OS configuration in addition to the z/OS operating system, DB2, and so on.

Use the latest service level of WebSphere Application Server for z/OS Version 7. WebSphere Process Server for z/OS Version 7.0.0.1 requires at least WebSphere Application Server for z/OS Version 7.0.0.7 plus ++APAR for AK99267.

If you intend to use more than one servant or a Network Deployment configuration of any kind, plan to use one of the supported releases of DB2 for z/OS. DB2 for z/OS V9.1 improved the performance of LOB management and relaxed some security rules on setting the current schema. Both of these changes benefit WebSphere Process Server for z/OS and for that reason we recommend DB2 for z/OS V9.1.

WebSphere Process Server for z/OS Version 7 also requires the Feature Packs for SCA, XML, and SDO, which you must be install in the WebSphere Application Server for z/OS using the zProfile Management Tool (zPMT). The zPMT is part of the WebSphere Configuration Tool (WCT).

In summary the software levels we used are:

- z/OS 1.9
- DB2 for z/OS 9.1
- WebSphere Application Server for z/OS 7.0.0.7 (plus fix for AK99267)
- Feature Packs for WebSphere Process Server for z/OS:
  - UK52837 for the Feature Pack for SCA V1.0.1.
  - UK52836 for the SDO 2.1.1 Feature of the Feature Pack for SCA V1.0.1.
  - UK52814 for the Feature Pack for XML V1.0.0.
- WebSphere Process Server for z/OS 7.0.0.1
- WebSphere Business Services Fabric for z/OS V7.0
- WebSphere Business Monitor V7.0.0.1
- WebSphere Configuration Tool (WCT)/zPMT V7.0.0.1
- Tivoli Directory Server for z/OS V1.9

See the program directory, PSP, and following technote:
`http://www.ibm.com/support/docview.wss?uid=swg24025352`

z/OS v1.8 or higher

Any supported release of the z/OS operating system can be used with WebSphere Process Server for z/OS, which covers Versions 1.8, 1.9, and 1.10. We recommend using the latest release (1.10) because it contains useful enhancements for your environment.

Preparing the base operating system

WebSphere Application Server for z/OS and WebSphere Process Server for z/OS make extensive use of the z/OS operating system components and Sysplex implementation. See the document “Preparing the base operating system” and the subordinate articles in the Information Center:

- Checklist: Preparing the base operating system.
- Prepare z/OS operating system settings.
- Prepare z/OS sysplex settings.
- Prepare the z/OS job entry subsystem (JES2 or JES3).
- Set up UNIX Systems Services.
- Identify TCP/IP resources you want to use and prepare your network.
- Set up Resource Recovery Services (RRS).
- Set up your SAF-compliant security package such as RACF.

  If you will use another SAF-compliant security product, consult the product's manufacturer for assistance.

- Set up component trace (CTRACE).

WebSphere Application Server for z/OS

The minimum level of WebSphere Application Server for z/OS must be at version 7.0.0.7. In general, we recommend using the latest release level of WebSphere Application Server for z/OS. WebSphere Process Server for z/OS Version 7 does not support and will not run on WebSphere Application Server for z/OS Version 6.

What is the level of my WebSphere products

There are multiple ways to determine the level of the WebSphere products in your runtime:

- For the configured nodes, you can run the `versionInfo.sh` script in the <profile_root>/bin directory.
- The service level also shows up in the Integrated Solutions Console Welcome window and on the System Administration → Nodes panel in the Version column.
- The message in the JOBLOG of the started servers: `BBOM0007I CURRENT CB SERVICE LEVEL IS build level 7.0.0.0 (wps0947.04) release WAS70.ZNATV date 11/25/09 14:01:14.`
- For a product’s SMP/E-maintained ZFS, you can look in the <smpe_root>/properties/version/WAS.product file (in ASCII).

Feature Packs for WebSphere Application Server for z/OS

WebSphere Process Server for z/OS requires three feature packs:

- SCA
- XML
- SDO

Feature pack installation

Feature packs are shipped through PTFs. See:
`http://www.ibm.com/support/docview.wss?uid=swg21417677`

The HFS filesystems for the Feature packs and other optional materials, such as the HTTP server plug-ins, are installed with different mount points but can be in the same or separate file systems that are shared within other optional materials:

- /shared/zWebSphere_OM/V7R0/FPSCA
- /shared/zWebSphere_OM/V7R0/FPXML
- /shared/zWebSphere_OM/V7R0/FPSDO

They can be installed in the same HFS (for example, hlq.SIWOHFS), or they can be installed in separate HFSes because they are in our ITSO system:

- BBO7042.SIWODHFS is mounted at /SC42/zWebSphereKS_OM/V7R0/FPSCA
- BBO7042.SIWOJHFS is mounted at /SC42/zWebSphereKS_OM/V7R0/FPXML
- BBO7042.SIWOEHFS is mounted at /SC42/zWebSphereKS_OM/V7R0/HTTP

WebSphere Application Server V7.0 Feature Pack for SCA V1.0 for z/OS

WebSphere Application Server V7.0 Feature Pack for SCA V1.0 for z/OS provides the Open SOA Collaboration (osoa.org) SCA 1.0 programming model to simplify composite application development and management in a SOA environment. More details are at:

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

FMID: JIWO700

The IBM Education Assistant (IEA) tutorial on installing and configuring this feature pack is available at:

http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.wasfpsca/wasfpsca/1.0/Installation/WASv7SCA_zOS_Installation/player.html

SDO 2.1.1 Feature Pack for SCA V1.0.1 for z/OS

SDO 2.1.1 Feature Pack for SCA V1.0.1 for z/OS is delivered as part of the WebSphere Application Server V7.0 Feature Pack for SCA V1.0 for z/OS. It has the same FMID: JIWO700 and is part of the same HFS file system but mounted at /shared/zWebSphere_OM/V7R0/FPSDO

WebSphere Application Server V7.0 Feature Pack for XML V1.0 for z/OS

The WebSphere Application Server V7 Feature Pack for XML provides application developers with support for the Worldwide Web Consortium (W3C) XML standards. For product information, see:

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

FMID: JIWO700 (Component ID: 5655N0256)

For more information about the Feature Pack for XML, see:

http://www.ibm.com/support/docview.wss?uid=swg21411998

The IBM Education Assistant (IEA) tutorial on installing and configuring this feature pack is available at:

http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.wasfpsca/wasfpsca/1.0/Installation/WASv7SCA_zOS_Installation/player.html

Overview of the product HFSes

Figure 2-11 on page 42 shows a view of the various product HFS filesystems that WebSphere Process Server for z/OS uses.

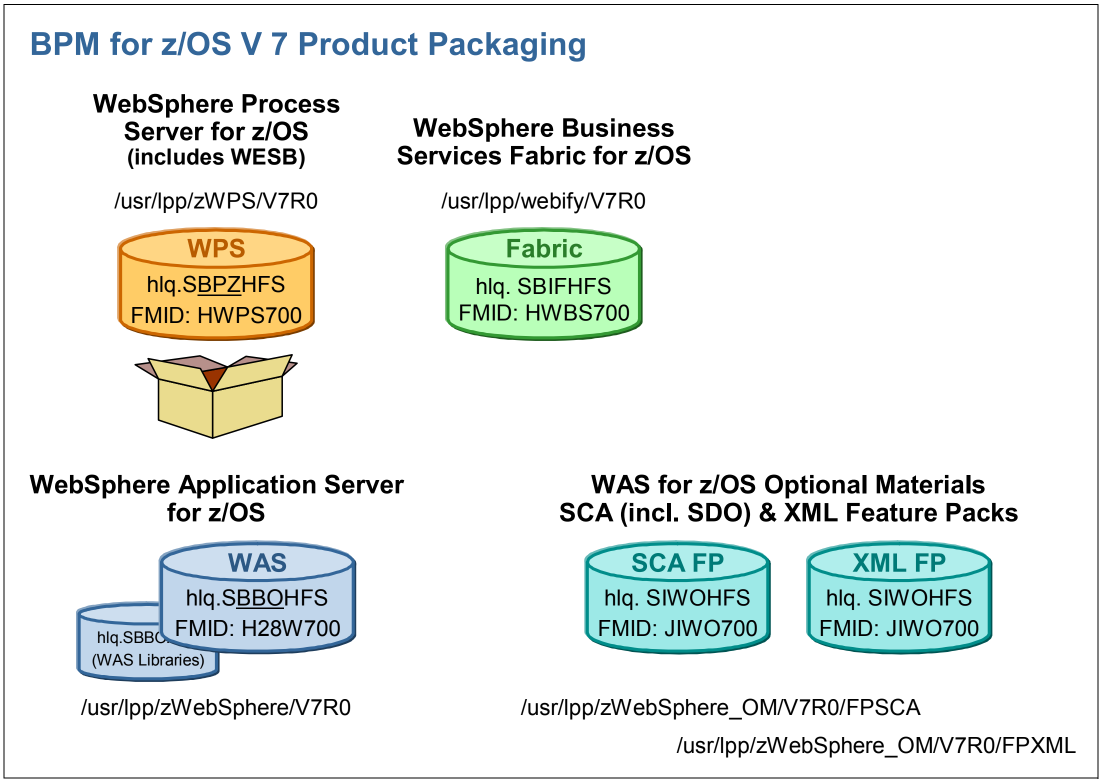

BPM for z/OS V 7 Product Packaging
WebSphere Process Server for z/OS (includes WESB)
/usr/lpp/zWPS/V7R0
WPS
hlq.SBPZHFS
FMID: HWPS700
WebSphere Business Services Fabric for z/OS
/usr/lpp/webify/V7R0
Fabric
hlq. SBIFHFS
FMID: HWBS700
WebSphere Application Server for z/OS
WAS
hlq.SBBOHFS
FMID: H28W700
hlq.SBBO
(WAS Libraries)
/usr/lpp/zWebSphere/V7R0
WAS for z/OS Optional Materials
SCA (incl. SDO) & XML Feature Packs
SCA FP
hlq. SIWOHFS
FMID: JIWO700
XML FP
hlq. SIWOHFS
FMID: JIWO700
/usr/lpp/zWebSphere_OM/V7R0/FPSCA
/usr/lpp/zWebSphere_OM/V7R0/FPXML

Figure 2-11 View of product deliverables

WebSphere Configuration Tool (WCT): zProfile Management Tool

The zProfile Management Tool (zPMT) component of the WebSphere Configuration Tool (WCT) V7.0.0.1 is required to configure your WebSphere cell. It goes on your workstation. We installed WCT Version: 7.0.0.6 (Build id: cf070938.09).

If you already installed earlier copies of these plug-ins, you might need to delete them:

1. Delete any profiles that use these.
2. Restart WCT.
3. Follow the instructions on Pg. 23 of white paper WP101394 at:

   `http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/TechDocs`

4. Install the plug-in extensions from the /util/WCT/ directories in the SMPE HFS:
   - /shared/wasv7drivers/wps7/zWPS/V7R0/util/WCT/zWBI.wct
   - /shared/wasv7drivers/scafp/zWebSphere_OM/V7R0/FPSCA/util/WCT/sca.wct
   - /shared/wasv7drivers/scafp/zWebSphere_OM/V7R0/FPSCA/util/WCT/sdo.wct
   - /shared/wasv7drivers/xmlfp/zWebSphere_OM/V7R0/FPXML/util/WCT/xml.wct
5. After you FTP these files to your workstation, for each wct file:
   - Open the zPMT, and go to **Help** → **Software Updates** → **Install Extension** → **Install new extension location**.

Using SMP/E to install WebSphere Process Server for z/OS

We do not explain how to install WebSphere Process Server for z/OS using SMP/E in this book because it is described in the document, *Program Directory for WebSphere Process Server for z/OS V7*, GI13-0553. A printed copy is shipped with the product, or you can download it from the IBM Publications Center at the following Web page:

ftp://ftp.software.ibm.com/software/websphere/integration/wps/library/pdfz700/

Preventive Service Planning (PSP)

For updated service planning information, review the PSP Buckets at:

http://www.ibm.com/support/docview.wss?uid=isg1_WPSZ_HWPS700

RECEIVE processing WPS

WebSphere Application Server for z/OS and all accumulated maintenance PTFs are included with WebSphere Process Server for z/OS, which can make the space requirements very large when you receive using SMP/E because of the very large RELFILEs.

Even if you already RECEIVEd and APPLYed the PTFs for WebSphere Application Server for z/OS, you must receive these to get a zero return code.

Notes:

- We had space problems with the RELFILEs, one almost 15000 trks, which is considerably larger than the normal installation space defaults and might cause receive problems, such as B37 ABENDs on the SYSUT4 DD and allocation failures for the HWPS700.F2 relfile.
- It contains almost 24000 members. SMPE invokes IEBCOPY to create the Relfiles and, in this case, we updated the SMPE options to add a PARM='WORK=8M' to avoid IEBCOPY failures (misleading IEB177I messages). Also use a large region size.

APPLY processing WebSphere Process Server for z/OS

The sample jobs to allocate the HFS and ZFS might be short on space. We found that after the APPLY, it took actually 1074 Cylinders in four extents, which might be a problem if you are limited on DASD space.

We tried using the ZFS allocation version of the allocation job, but it failed with no error messages. So instead of using the **zfsadm** UNIX System Services command in the JCL, we used a batch program for formatting zFSes, which formatted just fine, as shown in Example 2-1.

Example 2-1 Formatting the zFS file system for WebSphere Process Server for z/OS

```
//CREATE   EXEC   PGM=IOEAGFMT,COND=(0,LT),
// PARM=('-aggregate WAS700.WPS700.SBPZHFS -compat')
//* IOEAGFMT format Linear DS so it can be used for a zFS dataset.
//SYSPRINT DD     SYSOUT=*
//STDOUT   DD     SYSOUT=*
//STDERR   DD     SYSOUT=*
```

ACCEPT processing WPS

The parameter WORK=2M or 4M is also required for the IEBCOPY program on the ACCEPT step; otherwise, you get the same symptoms as the Receive. If you have multiple OPTIONS entries and multiple IEBCOPY Utility entries, this can also be a problem, which is documented in the Program Directory but easy to miss.

Installing and configuring WebSphere Process Server for z/OS

The starting point for this section assumes that you successfully unloaded the products from the installation media, installed them using SMP/E, and mounted the product HFS data sets into your UNIX hierarchical file system (HFS). If you ordered WebSphere Process Server for z/OS using CBPDO, the installation using SMP/E involves the following tasks:

- Obtaining the latest copy of the Program Directory document.
- Obtaining the latest Preventative Service Planning (PSP) document for WebSphere Application Server and WebSphere Process Server. The PSP has the list of the latest important PTFs to install. The name of this document and instructions on obtaining it are in the Program Directory.

If you ordered the products using IBM SystemPac® or ServerPac, follow the instructions in *ServerPac: Installing your Order that ships with your SystemPac or ServerPac* that came with the product. Also, the product Information Centers contain information about the installation using SMP/E.

Feature pack installation

The Feature Packs for SCA, XMP, and SDO for WebSphere Application Server for z/OS are actually shipped using PTFs but must be augmented in the nodes *before* WebSphere Process Server for z/OS can be augmented.

For details, see:

http://www.ibm.com/support/docview.wss?uid=swg21417677

WebSphere Process Server V7.0 for z/OS requires the following PTFs to be applied to WebSphere Application Server Feature Packs for z/OS:

- UK51852: Feature Pack for SCA V1.0.1
- UK51850: SDO 2.1.1 Feature for SCA V1.0.1
- UK51832: Feature Pack for XML V1.0

These MUST be applied before configuring WebSphere Process Server.

Documentation for WebSphere Process Server for z/OS

The Information Center for WebSphere Process Server for z/OS is available online at:

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

You can download the product documentation in PDF form from:

ftp://ftp.software.ibm.com/software/websphere/integration/wps/library/pdfz700/

2.4 UNIX Systems Services planning

In this section, we provide some recommendations for setting up UNIX Systems Services users and components in z/OS.

2.4.1 Controlling use of UID=0

We recommend that the user IDs that you use to install and configure WebSphere Process Server for z/OS not have SuperUser authority (UID=0.) We also advise tight control of users with access to BPX.SUPERUSER.

Using surrogate profiles

Instead of passing around the password for administrator's user ID, you can use the RACF SURROGAT class, as shown in Example 2-2.

Example 2-2 Setting up a surrogate profile for the WebSphere administrator

```
RDEFINE SURROGAT BPX.SRV.B7ADMIN UACC(NONE)
PERMIT BPX.SRV.B7ADMIN CLASS(SURROGAT) ID(FRED) ACCESS(READ)
SETROPTS RACLIST(SURROGAT) REFRESH
```

Authorized users, such as 'FRED', can switch to the administrator's user ID with the following form of the su command without knowing the password for the b7admin ID:

```
su -s b7admin
```

Using file system privileges instead of superuser

Instead of using a super-user ID, use the various UNIXPRIV profiles and RACF PERMISSIONS to the SUPERUSER.FILESYS** profiles in the UNIXPRIV class. See the sample job in the Spreadsheet "JCL_P" tab called "B7UNIXPR" and Example 2-3.

Example 2-3 Sample RACF commands to create profiles to avoid use of UID 0 on installer ID

```
/*   Define the profiles         */
   RDEF UNIXPRIV  SUPERUSER.FILESYS             UACC(NONE) */
   RDEF UNIXPRIV  SUPERUSER.FILESYS.MOUNT       UACC(NONE) */
   RDEF UNIXPRIV  SUPERUSER.FILESYS.CHOWN       UACC(NONE) */
   RDEF UNIXPRIV  SUPERUSER.FILESYS.CHANGEPERMS  UACC(NONE) */
   RDEF UNIXPRIV  SUPERUSER.FILESYS.PFSCTL      UACC(NONE) */

/*   Permit the configration group access and sysprogs    */
 PE   SUPERUSER.FILESYS              CLASS(UNIXPRIV) +
      ID(B7CFG,SYS1)                 ACCESS(CONTROL)
 PE   SUPERUSER.FILESYS.MOUNT        CLASS(UNIXPRIV) +
      ID(B7CFG,SYS1)                 ACCESS(UPDATE)
 PE   SUPERUSER.FILESYS.CHOWN        CLASS(UNIXPRIV) +
      ID(B7CFG,SYS1)                 ACCESS(READ)
 PE   SUPERUSER.FILESYS.CHANGEPERMS  CLASS(UNIXPRIV) +
      ID(B7CFG,SYS1)                 ACCESS(READ)
 PE   SUPERUSER.FILESYS.PFSCTL       CLASS(UNIXPRIV) +
      ID(B7CFG,SYS1)                 ACCESS(READ)
 CONNECT HUTCHJM GROUP(B7CFG)
 CONNECT GATES   GROUP(B7CFG)
      /*activate   */
 SETR RACLIST(UNIXPRIV) REFRESH
```

2.4.2 Customizing .profile and .setup files for users

The default UNIX environment that is specified in /etc/profile might be using old versions of Java and DB2 for z/OS. To ensure that the UNIX command-line environment is customized for the correct version of Java and DB2 Universal JDBC Driver, you must customize a .profile for each user ID that will work with WebSphere Application Server and DB2 z/OS from the command line or from UNIX batch jobs.

For WebSphere Application Server for z/OS, the administrator user ID and the controller user ID share a home directory, for example, the home directory of the administrator user ID of B7CELL was /var/WebSphere/home/B7CFG.

We created a .profile in /var/WebSphere/home/B7CFG that set PATH, LIBPATH, STEPLIB, and CLASSPATH so that the UNIX environment was correctly customized for Java and JDBC, as shown in Example 2-4, which allowed any shell script that we invoked to make JDBC calls and connect to the correct DB2.

JAVA_HOME is set to point to the same Java that the deployment manager uses so that any scripts that are launched under user B7ADMIN will run with the same level of Java that WebSphere Application Server itself uses.

Rather than concatenate the paths to their existing settings, we decided to make sure that the CLASSPATH was how we wanted it by using **unset CLASSPATH** to clear the current CLASSPATH before setting it.

Example 2-4 Sample .profile showing the setting of JAVA and DB2 variables

```
#!/bin/ksh
export HOSTNAME=$(uname -n)
export PS1='$LOGNAME @ $HOSTNAME:$PWD>'
# If _BPX_SHAREAS=YES it can prevent the DMGR from starting
export _BPX_SHAREAS=NO
export ENV=$HOME/.setup
# JAVA
export JAVA_HOME=/wasv7config/b7cell/b7dmnode/DeploymentManager/java
export PATH=$JAVA_HOME/bin:$PATH
####DB2 JDBC and JCC
LIBPATH=/usr/lpp/db2/d9fg/db2910_jdbc/lib:$LIBPATH
LIBPATH=/usr/lpp/db2/d9fg/db2910_base/lib:$LIBPATH
STEPLIB=DB9F9.SDSNEXIT:DB9F9.SDSNLOAD:DB9F9.SDSNLOD2
####
JCC_HOME=/usr/lpp/db2/d9fg/db2910_jdbc
PATH=$JCC_HOME/bin:$PATH
unset CLASSPATH
CLASSPATH=/etc/d9fg/DB2JccConfiguration.properties
CLASSPATH=$CLASSPATH:$JCC_HOME/classes/db2jcc.jar
CLASSPATH=$CLASSPATH:$JCC_HOME/classes/db2jcc_javax.jar
CLASSPATH=$CLASSPATH:$JCC_HOME/classes/sqlj.zip
CLASSPATH=$CLASSPATH:$JCC_HOME/classes/db2jcc_license_cisuz.jar
CLASSPATH=$CLASSPATH:.
export PATH CLASSPATH LIBPATH STEPLIB
####
echo  PATH reset to $PATH
echo -------------------------------------------------------------
echo  CLASSPATH reset to $CLASSPATH
echo -------------------------------------------------------------
echo  JAVA_HOME reset to $JAVA_HOME
echo -------------------------------------------------------------
echo  LIBPATH reset to $LIBPATH
echo -------------------------------------------------------------
echo  STEPLIB reset to $STEPLIB
echo -------------------------------------------------------------
```

Creating a .setup file for users

In addition to setting a .profile for the WebSphere administrator user ID, it is a good idea for most users to have the .setup file shown in Example 2-10 on page 66. To help with typing in telnet sessions, we also created some alias names in the .setup files in the same home directories as the .profile files. Example 2-10 on page 66 shows a sample setup file that you can tailor to your preferences.

Example 2-5 Sample setup file

```
alias ll="ls -lat"
alias dirs="ls -alLD"
set -o vi
alias lookup="find . -name "
```

In Example 2-10 on page 66:

- `ll` allows you to list the files' details with the newest ones at the top.
- The syntax `set -o vi` helps you retrieve commands from the access history. Press the ESC key one time, release it, and press k. The last command that you typed is displayed. Keys 'j' and 'k' help you to scroll back and forth in the history buffer.
- The syntax `lookup` allows you to issue the **`find`** command more easily.

2.4.3 HFS and zFS file system planning

In this section, we provide some recommendations for setting up UNIX System Services filesystems in z/OS.

HFS versus ZFS

We recommend that you use ZFS rather than HFS when allocating the configuration datasets for the nodes. ZFS provides improved performance over HFS and additional functions, such as cloning for making copies and logging for increased availability and recovery. Also, IBM announced the stabilization of the HFS file systems in February, 2004.

Single shared file system versus separate

We recommend that you create separate configuration HFS/ ZFS filesystems per node. This is a good practice and allows for more granular back-up and restore at the HFS dataset level. It also provides a way to avoid shared HFS between MVS images. The performance of write operations to an HFS that is not owned by the MVS operating system is much slower. It is better to have each node's HFS owned by the operating system on which the node's servers run.

Mount points and roots

The terms *root* and *home* are often used to describe where in the UNIX System Services file system an HFS or zFS file system is *mounted*. The abbreviations that we use in this book to describe root directories in the HFS are:

was_smpe_root WebSphere Application Server SMP/E installation root directory: usually /usr/lpp/zWebSphere/V7R0

fpsca_smpe WebSphere SCA Feature Pack installation root directory: usually /usr/lpp/zWebSphere_OM/V7R0/FPSCA

fpxml_smpe WebSphere XML feature pack installation root directory: usually /usr/lpp/zWebSphere_OM/V7R0/FPXML

fpsdo_smpe WebSphere SDO feature pack installation root directory: usually the same as the SCA feature pack at /usr/lpp/zWebSphere_OM/V7R0/FPSCA

wps_smpe_root WPS SMP/E installation root directory: usually /shared/zWPS/V7R0

configuration_root Node's mount point for the configuration file system

app_server_root Application server or deployment manager's home = {configuration_root}/AppServer

profile_root Server's profile directory = {app_server_root}/profiles/default/ on z/OS

We mounted our configuration ZFS filesystems at a mount-point of:

`/wasv7config/<cell_name>/<node_name>` for example:
`/wasv7config/b7node/b7dmnode/ and /wasv7config/b7node/b7nodea/`

The root (/) file system should be mounted Read-Only. Under the root should be a small file system for /wasv7config/ where the various cells can add their *cell root* and must be mounted Read/Write; otherwise, you must change the root to R/W, create the new mount point, and then change it back to R/O.

2.4.4 Using intermediate symbolic links

Symbolic links are files that contains pointers to other files or directories, usually in a different file system. They can be valuable by providing flexibility to not bind your configuration file system to a specific copy of a SMP/E-maintained copy of a product HFS.

When you configure a node, you must specify the paths to the product mount points. We suggest that you create intermediate symbolic links that indirectly reference the physical mount points of the products rather than referring directly to the mount points.

WebSphere products for z/OS delivered as separate HFS files

WebSphere Application Server for z/OS ships a product HFS file system that contains files and code that are needed to support a WebSphere Application Server for z/OS configuration. This product HFS is shipped as hlq.SBBOHFS, where hlq is the high-level qualifier that you give it, for example, WAS700.WAS.SBBOHFS, which is then mounted at a UNIX Systems Services mount point, such as /usr/lpp/zWebSphere/V7R0. See "Overview of the product HFSes" on page 41.

When you configure a node, you must specify the paths to the WebSphere Application Server for z/OS and WebSphere Process Server for z/OS product mount points. We recommend that you create intermediate symbolic links that indirectly reference the physical mount points of the products rather than referring directly to the mount points.

Why use intermediate symbolic links

Intermediate symbolic links allow you to easily switch a node from one service level to another by deleting and redefining the intermediate symbolic links.

> **Note:** For a full discussion on this best practice, refer to White Paper WP100396, *Test, Production and Maintenance: The essential concepts needed to know how to configure isolation between environments*, from the following Web page:
>
> http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100396

We recommend that you use intermediate symbolic links to have the flexibility to upgrade one node at a time. By using intermediate symbolic links, the upgrade granularity is at the level of a node.

The zPMT that created BBOWHFSA, BBOMHFSA, and BBODHFSA jobs, automatically creates the intermediate symbolic links for the WebSphere Application Server for z/OS.

For the Feature Packs, the zPMT created these jobs:

IWODAUGM	SCA Feature Pack
IWOJAUGM	XML Feature Pack
IWOKAUGM	SDO Feature Pack

For WebSphere Process Server for z/OS, the sample job BPZDOLNK created by the zPMT as part of the augmentation of WebSphere Process Server for z/OS generates the symbolic link for WebSphere Process Server for z/OS.

Run this job, which generates a symbolic link to the specified install image from an intermediate directory. This job must be run under a user ID with the UNIX System Services files access privileges:

- CONTROL access to SUPERUSER.FILESYS
- READ access to SUPERUSER.FILESYS.CHOWN
- READ access to SUPERUSER.FILESYS.CHANGEPERMS

If you do not use intermediate symbolic links, an upgrade to a new service level of WebSphere Process Server for z/OS will affect all the nodes that refer to the product's mount point. You must stop all of the nodes on the LPAR that refer to the product ZFS, unmount it, mount the new service level, and then restart all of the nodes on the LPAR. In other words, when you do not use an intermediate symbolic link, the upgrade granularity is that of the LPAR.

You can also define these manually, as is shown in Example 2-6, but you must do it before the job that populates the configuration HFS (BBOWWPFA).

Example 2-6 Creating a symbolic link for WebSphere Process Server for z/OS

```
cd /wasv7config/b7cell/b7nodea
ln -s /usr/lpp/zWPSKS/V7R0 wps_smpe
```

2.5 Security planning and preparation

You must enable Global Security in your production environment. Running with security disabled is NOT an option. There are many decisions you must make about security, and your careful approach and good planning can help prevent problems down the road.

We recommend that you enable security when you first configure the WebSphere Application Servers, which helps you create the necessary user IDs, profiles, keyrings, and certificates at the outset, and is much easier than trying to retro-fit security later on.

When you enable security in a WebSphere Process Server for z/OS cell, you must choose a user registry. You can use a LocalOS, SAF-based user registry, such as RACF, an LDAP user registry, a Custom registry, or a Federated registry.

For LDAP, see 2.5.2, “Using LDAP: Tivoli Directory Server for z/OS” on page 53.

In preparing this Redbooks publication we used a RACF user registry; therefore, there are several steps that help you configure WebSphere Process Server for z/OS and WebSphere Business Services Fabric for z/OS to use RACF.

It is important to have a clear security plan before you start. Both, the previously mentioned techdoc and the zProfile Management Tool (zPMT) generate JCL and RACF commands that configure RACF to support the needs of WebSphere Application Server for z/OS. You must create your own commands to support the needs of WebSphere Process Server for z/OS and WebSphere Business Services Fabric for z/OS.

2.5.1 Planning RACF user IDs, groups, and profiles

When configuring a WebSphere Application Server for z/OS V7 cell using the zProfile Management Tool (zPMT), jobs are generated that configure the RACF user IDs, groups, and profiles that are required by the WebSphere Application Server for z/OS cell. If you use the default configuration scripts that the zPMT creates, the security profiles are created on an “as needed” basis for the deployment manager, empty node, clusters, and servers as you add them to your cell, which makes it difficult when you expand your configuration at a later time.

We recommend that you use the Generic RACF Profiles with WebSphere Application Server for z/OS, as described in Techdoc WP101427, which helps you to define profiles in the CBIND, SERVER, STARTED, and other RACF classes that are more generic and allow expansion of your configuration. This process removes the need to add new profiles each time a new component is created. The WebSphere Application Server for z/OS system administrator can then define new components (nodes, servers, clusters, and so on) without involving the security administrators. In addition, this techdoc provides a REXX EXEC to delete or un-do the RACF definitions that the scripts create, so you can start over.

The spreadsheet provided with this book creates sample jobs B7RAC1TM and B7RAC700 to define these RACF users and profiles in the JCL_P tab of the spreadsheet to help you.

Run the B7RAC1TM job to activate RACF classes that WebSphere Application Server for z/OS uses, and define RACF FACILITY class profiles one-time for the Sysplex. Review this job with your security administrator and change ‘xxxxxxxx’ to the user ID or group that will own the profiles that this job creates.

Before running the xxRAC700 job, make three manual changes:

1. Change `xxxxxxxx` to the user ID or group that will own the B7CFG group created by this job.
2. Change `yourid` to the TSO user ID of the installer, which permits required RACF privileges to the installer and makes the installer a WebSphere Application Server administrator for the cell.
3. Add a STARTED class profile for the adjunct started class, as described in “Adjunct identity” on page 50, in addition to the profiles that are already defined.

Adjunct identity

The WCT in V7 now assigns the controller user ID to the adjunct when using reduced RACF user IDs. Therefore, we made a change to these profiles to assign the controller user ID to the adjunct. The reason this is important is that when using SIBs in cluster, there is outbound communication from one adjunct to the other as part of the HA manager function. When security is enabled, that takes place over SSL and a handshake is required.

When the adjunct runs with the servant user ID, its keyring does not have a personal certificate and the connection cannot succeed because of an SSL handshake failure.

Here is the additional profile we added to the DORAC700 set of commands:

```
RDEFINE STARTED B7%%%%%A.* STDATA(USER(B7ACRU) GROUP(B7CFG)
TRACE(YES))
```

Security registry

The fundamental choice you must make is what user registry to use. Will you use a SAF user registry, such as RACF, or an LDAP user registry, such as Tivoli Directory Server for z/OS. The choice is influenced by many factors, but it is certainly important to know which one you will use before you attempt to configure WebSphere Process Server for z/OS.

Even if you decide to use RACF as the user registry for WebSphere Process Server for z/OS, you must define an LDAP server for the users that access WebSphere Business Services Fabric for z/OS. See 2.5.2, “Using LDAP: Tivoli Directory Server for z/OS” on page 53.

We suggest that you use the spreadsheet together with jobs B7RACF and B7EJBROL that are provided in the additional material that accompanies this publication. For most of the roles that we defined, a RACF group uniquely provides access to that role. This process allows us to give a user ID that role by connecting it to that group. Some of the EJBROLE profiles were set to UACC(READ), which is the same as making the role available to all authenticated users.

UserIDs, GroupIDs, UIDs, and GIDs

Additional RACF user IDs, groups, and profiles are required for WebSphere Process Server for z/OS. The spreadsheet created a sample job called B7RACF to define all of the user IDs and groups we planned for our cell.

The WPS_RACF worksheet on the spreadsheet described in section 2.2.2, “Using a spreadsheet to plan your configuration” on page 27 can be used to help plan all of the user IDs, groups, and EJBROLE profiles that are needed in RACF.

For some roles, we defined a new user ID that was connected to the role group, for example, user ID B7BPEADM was defined in group B7BPEADG, which was on the access list for the EJBROLE B7.BPESystemAdministrator. Obviously you can use your existing user IDs and simply connect them into the correct role groups. However, when you configure BPC, you are prompted for both a user ID and a group that is associated with the roles of the BPESystemAdministrator and HumanTaskSystemAdministrator. We thought it was better to create a specific task user ID for this rather than to use an existing, personal user ID that might not be persistent in the system.

Using EJBROLE profiles

We also created a job called B7EJBROL that defined RACF EJBROLE profiles to support the roles that WebSphere Process Server for z/OS components use. This job creates one profile per role. If you want to group roles together in fewer profiles, see “Using GEJBROLE profiles” on page 52.

It is a good idea to create all of the required user IDs, groups, and EJBROLE profiles before you configure WebSphere Process Server for z/OS. The user IDs, groups, and EJBROLE profiles that we created are summarized in tables on the “WPS_RACF” panel of the spreadsheet.

The roles and groups in these spreadsheet tables reflect a resource-oriented approach to defining RACF profiles, where there is a profile for each role and a RACF group provides access to that role.

Some roles are defined as UACC(READ), meaning that anyone that authenticated will have that role. You might not want to use UACC(READ) for the roles shown in the spreadsheet tables. We did this for the API roles to show you that you can use UACC(READ).

An unauthenticated (guest) user ID cannot access roles that are UACC(READ).

Because the unauthenticated user ID is defined as a RACF RESTRICTED user, you must explicitly PERMIT it to the resources that you want it to have access to. This process is especially important when dealing with Business Space powered by WebSphere functions, which protect several resources with the J2EE Special Subjects Everyone and All authenticated. See the following Web site for considerations when using an authenticated user id with RACF and J2EE Special Subjects:

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

EJBROLE profiles for WebSphere Business Services Fabric for z/OS

There are a number of roles that are associated with WebSphere Application Server for z/OS, WebSphere Process Server for z/OS, and WebSphere Business Services Fabric for z/OS. When using RACF to perform EJB role authorization checking, there are two main ways to approach the management of the RACF profiles:

- The Resource-oriented approach starts by creating profiles to protect all of the resources (in this case EJBROLE profiles), and then considers the access list for each profile.
- The Organization-oriented approach starts by considering how many different types of users there are, and then defines the minimum number of profiles that are required to provide the correct access.

We initially took the granular resource-oriented approach by defining an EJBROLE profile for each role and creating a number of different RACF groups that we permitted access to each role. A drawback of this approach is that it leads to a large number of EJBROLE profiles. The organization-oriented approach results in fewer RACF profiles and is easier to maintain and actually helps performance.

Using GEJBROLE profiles

When following the organization-oriented approach, a useful technique is to define RACF GEJBROLE profiles rather than EJBROLE profiles. When you define a profile in class EJBROLE, it refers to only one role. When you define a profile in class GEJBROLE, it can contain more than role. Roles are added to a GEJBROLE profile using the ADDMEM(<role>) option, for example, suppose that your organization consists of two different organizational groups:

- People responsible for administering WebSphere Process Server for z/OS
- Ordinary users that might need certain roles to use applications

You can create a GEJBROLE profile called WPSAdmins, and add to it all of the roles that provide administrator privileges. Then you can create a second GEJBROLE profile called WPSUsers, and add to it all of the roles that an ordinary user needs to run.

Having only two GEJBROLE profiles like this is at the opposite extreme to having one EJBROLE profile for each role. You will probably find that the correct solution for your organization lies between these two extremes. It is not possible to define only two GEJBROLE profiles for administrators and users, for example, because you must define at least two more profiles to support the JMSAPIUser and EscalationUser runAs roles. Nevertheless, we recommend the organization-oriented approach because it reduces 20 profiles to around five.

For a possible solution, in the spreadsheet under the WPS_RACF tab, see the section labeled `WPS Roles, users and groups when using RACF GEJBROLE profiles (one profile per organizational role)`.

After you decide to use GEJBROLE profiles, define all EJB role profiles in that class. Do not mix EJBROLE and GEJBROLE profiles. Although a GEJBROLE profile contains only one role, it is the same as a single EJBROLE profile. It is best to define the GEJBROLE profile with one member. RACF considers profiles in both classes and merges the access lists of any roles that appear in more than one profile, which can cause confusion.

We created sample JOB GEJBROLE to implement the role assignments shown in the spreadsheet tables for our B7 cell. However, you cannot run this job before the BBOSBRAK job that is created and run as part of the server installation in section 3.2.2, “Executing the installation jobs” on page 74.

2.5.2 Using LDAP: Tivoli Directory Server for z/OS

LDAP servers are available on several platforms, but because this publication is about WebSphere Process Server for z/OS, in this section, we describe some of the reasons to consider running LDAP on z/OS.

We do not describe in detail how to configure an LDAP user registry, but the spreadsheet includes a worksheet called WPS_LDAP, which contains entries in ldif format for loading into Tivoli Directory Server for z/OS. Details about the LDAP server are set on the Variables worksheet. The users and groups are derived from the other worksheets. Most of them come from the WPS_RACF worksheet where the majority of WebSphere Process Server for z/OS and WebSphere Business Services Fabric for z/OS user IDs and groups are planned.

Why use Tivoli Directory Server for z/OS

LDAP servers are available on several platforms but because this publication is about WebSphere Process Server for z/OS, in this section, we describe some of the reasons to consider running LDAP on z/OS.

There are two different LDAP Servers that are available on z/OS since z/OS 1.8:

- z/OS Integrated Security Services (ISS) LDAP
- IBM Tivoli Directory Server for z/OS V6.0 LDAP

IBM Tivoli Directory Server for z/OS is preferred because the ISS LDAP is functionally stabilized and its schema is less compatible with other LDAP servers.

Tivoli Directory Server z/OS schema: The Tivoli Directory Server z/OS schema is compatible with Tivoli Directory Server on other platforms. Products, such as Tivoli Access Manager, for example, can equally use Tivoli Directory Server on z/OS as their user registry. A non-z/OS WebSphere Application Server for z/OS can also use the Tivoli Directory Server z/OS LDAP as its user registry.

Tivoli Directory Server for z/OS offers several advantages compared to a non-z/OS LDAP server that must be taken into account by z/OS customers when selecting an LDAP server. These advantages are:

- Tivoli Directory Server for z/OS is shipped with the base z/OS operating system and is therefore available at no charge to z/OS customers.
- Tivoli Directory Server for z/OS has the LDAP Native Authentication (LNA) feature. Native Authentication is where a user’s password is held in RACF rather than on the

userpassword attribute in the LDAP database. Using LNA can increase the security of powerful system administration user IDs, such as those with access to the Integrated Solutions Console. Such users can be defined as LNA user IDs where their passwords are managed with RACF, and other normal users are held in the LDAP database. Passtickets can be used with LNA user IDs, also.

- Tivoli Directory Server for z/OS has its database in DB2 z/OS, which can be shared across a sysplex, which gives you high availability including load balancing and disaster recovery.
- You can also configure the Tivoli Directory Server z/OS in failover mode, rather than load sharing mode, just as on other platforms. You can even replicate between sysplex nodes.
- Tivoli Directory Server z/OS uses the z/OS UNIX file system, which is more secure than other UNIX file systems because the privileges are controlled by RACF.
- The passwords of users that are held in the LDAP database can be encrypted with the private key that is stored in the z/OS hardware. No keystore files protected by passwords can be hacked.
- The LDAP configuration file can be protected against unauthorized read/write in a z/OS partitioned dataset, which makes it harder for someone to gain access to these files.
- Tivoli Directory Server LDAP z/OS also provides an SDBM schema that maps directly onto the RACF database, which allows a security administrator to use a graphical LDAP client to manage the RACF database.
- Tivoli Directory Server on z/OS can use the hardware crypto cards on the z machine to manage crypto processing related to SSL, giving excellent scalability of logins.
- For scalability in general, the DB2 for z/OS subsystem that provides the database for Tivoli Directory Server is a scalable, available database. Therefore, frequently used LDAP pages end up in memory in the coupling facility (in DB2 global buffer pools). So access to LDAP data is at memory speeds.
- The Tivoli Directory Server for z/OS LDAP server also has a new internal cache management implementation with inter-cache consistency in a sysplex.

For a description of configuring LDAP security, see the scenario described in section 5.3 of *z/OS: WebSphere Business Process Management V6.2 Production Topologies*, SG24-7733.

2.6 Database planning and preparation

WebSphere Process Server for z/OS and other BPM products use database tables that DB2 manages.

It is important to include your Database Administrator (DBA) or an experienced DB2 expert when you plan for and configure WebSphere Process Server for z/OS. In this section, we describe the planning, preparation, and execution of the DB2 database, tables, and other required DB2 artifacts.

2.6.1 Designing the database configuration

Decide on your DB2 database design and configuration from the start, and then decide on your naming convention for the DB2 objects, such as databases, storage groups, schema prefixes, buffer pools, and security IDs.

A key question you must answer is “Do I put all of the WebSphere Process Server for z/OS components in one database, or should I create a separate database for each component?”

We suggest that you plan to create a single databases for all of the WebSphere Process Server components, at least for your first implementation. Another recommendation is to create one database at the cell-level for the data sources that are defined at the cell scope (WBI common and BPC Reporting) and the other at the cluster level for the remaining datasources (CEI, BPC, BS, and the MEs) after you have more experience configuring DB2 for WebSphere Process Server for z/OS.

The planning spreadsheet can assign a single database for all components or separate database names to each component.

Using a single database

The advantages of a single database are:

- Easier to create all of the necessary tables by running the createDB.sh script just one time for all of the DB2 tables.

  The createDB.sh script is a sample script that wraps the DButility.sh and is easier to use than DButility.sh. It is described in Section 4.2.2, "Generating the SQL with createDB.sh" on page 106.
- Simpler to specify and manage GRANTs (DBADM) on just one database.

  The createDB.sh shell script creates objects in one database. You can simply run createDB.sh one time to generate and execute the SQL for all WebSphere Process Server for z/OS components in the server.
- Easier to manage the environment because a single database contains all of the definitions for a given WebSphere Process Server. DBAs can more easily clone the environment and backup its contents.
- If you have one database, the majority of tables and indexes end up in the same buffer pools. If this is what you want, you might choose to use only one database.
- If you have multiple clusters, creating multiple additional databases for each cluster can become quite cumbersome. Using a single database for each cluster simplifies this process.

A good practice is to use your cluster name as your database name, and use the cluster name as the owner of the DB2 resources.

Using multiple databases

The reasons for using multiple databases are:

- More control and easier to tailor the SQL for specific bufferpools, space requirements, and schema names.
- Easier to validate that you have all of the tables and to drop and re-run the CREATE TABLESPACE and TABLE SQL files.

When using DB2 for z/OS, choosing to define multiple databases only has an indirect effect on performance because you can specify the BUFFERPOOL and INDEXBP options on the CREATE DATABASE statements. Much of the SQL that is provided with WebSphere Process Server for z/OS does not specify bufferpools, so they are assigned the default that applies to the database, table, or index in which it is created.

If you want to place tables and indexes in separate, same buffer pools, we suggest that you create separate databases and assign different BUFFERPOOL and INDEXBP values to each component, at least initially.

Using multiple databases makes it easier to manage when the tables and indexes for each component are isolated in their own database. It is possible that you will find a need to delegate some DB2 administration privileges for specific component databases to individuals in your organization, which is much easier to do if you can simply GRANT DBADM to such people.

The planning spreadsheet can assign different database names to each component, if you choose, but the default is currently for the WebSphere Process Server components to share one database.

If you have separate databases and storage groups for each component, their names should be well-organized to avoid confusion during the configuration phase, managing the runtime, and in problem determination, as shown in Table 2-1.

Table 2-1 Alternative layout using separate databases for the MD Cell

Component	DB Name	Schema	Bufferpools
Common WPS & ESB	B7WPSDB	B7WPS	BP1, BP2
BPC	B7BPCDB	B7WPS	BP5, BP6
Business Process Reporting	B7BPRDB	B7WPS	BP7, BP8
CEI	B7CEIDB	B7WPS	BP9, BP10
Business Space	B7BSPDB	B7WPS	BP11, BP12
SCA SYS Bus	MDSIBDB	B701S	BP3, BP4
SCA APP Bus	MDSIBDB	B701A	BP3, BP4
BPC Bus	MDSIBDB	B701B	BP3, BP4
CEI Bus	MDSIBDB	B701C	BP3, BP4

Planning DB2 buffer pools

DB2 data requests for data and indexes might be filled from memory rather than disk to help performance. Buffer pools use fixed page sizes, which must be the same as the page size of any tablespaces they serve. A default buffer pool can be associated with a database and can be used by more than one database and tablespace.

It is better to separate Catalog, Data, Index, and Workfiles into separate bufferpools. Generally, we should have fewer but larger bufferpools and allow DB2 to manage the resources. If we identify performance issues based on the workloads, we can choose to manually isolate tablespaces into their own bufferpool.

You should not use bufferpools BP0, BP8K0, or BP16K0 because they are used by the DB2 catalog and directory and reserved, in most installations, for the DB2 subsystem. Unfortunately some generated SQL (for Business Space ‘BSPBS’ TableSpace) defaults to these values, so review all SQL before executing it, and set the correct bufferpools.

Work with your DB2 database administrator to plan the buffer pools you will use. We used the spreadsheet described in Section 2.2.2, “Using a spreadsheet to plan your configuration” on page 27, to plan all of the names and buffer pools that we used in DB2. The table on the Variables worksheet that plans database names and bufferpools does not allow you to choose BP0, BP8K0, or BP16K0. For each bufferpool, we issued **`-ALTER BUFFERPOOL(BPxxx) VPSIZE(yyyy)`** to define a buffer pool BPxx with yyyy pages.

Because we configure a cluster, which in turn requires DB2 in data-sharing mode, any new buffer pools we add also require cache structures to be defined in the Coupling Facility for Global Buffer pools. For instruction about how to do this, refer to the DB2 for z/OS Information Center article *Defining coupling facility structures* at:

http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db29.doc.dshare/db2z_definingcfstructures.htm

2.6.2 Planning DB2 names

WebSphere Process Server for z/OS involves many components that require many DB2 tables. It is very important to organize and document the DB2 objects to ensure a smooth and successful implementation, for example, you must decide on a naming convention for:

- Databases
- Storage groups
- VCATs
- Tables
- Schema prefixes
- Bufferpools
- Authentication alias IDs

Plan these names with your DB2 system administrator. The spreadsheet that we describe in Section 2.2.2, “Using a spreadsheet to plan your configuration” on page 27, includes a worksheet that can help you choose names for the databases, if you decide to have separate databases for each component.

Naming the databases, storage groups, schemas, and VCATs

If you have one database, the names of these DB2 components is relatively simple. If you have separate databases and storage groups for each component, their names must be well-organized to avoid confusion during the configuration phase, managing the runtime, and in problem determination.

We use the same components for all tables, whether we are using the same database or multiple:

- VCAT (B7DB2VC)
- Storage Group (B7DBSTO)

Using a single database, we used B7WPSDB.

The schema qualifier can vary, but we chose B7WPSDB for all our tables except for the messaging engines. The table names are the same for all messaging engines, so they must be schema qualified with different prefixes to make them unique.

2.6.3 Using CurrentSchema versus SQLID versus RunAs Servant ID

An understanding of schema and how it relates to SQLID is essential to configure WebSphere Process Server for z/OS successfully. The currentSchema and currentSQLID properties can be set on the custom properties of a data source.

WebSphere Process Server for z/OS implications

When an SQL request is made by WebSphere Process Server for z/OS, there are several ways to specify the user ID that the request runs under, which influences the J2C authentication alias when making the request.

Specifying a user ID as the J2C authentication alias

You can specify alias names, which can be mapped to user identities and passwords to authenticate the SQL request to DB2. These identities must be GRANTed authorization to access or manipulate the objects in DB2 according to the defined priviledges or GRANTs.

Using the J2C authentication alias as the schema

One option is to define all of the WebSphere Process Server for z/OS tables with a schema that is equal to the J2C authentication alias user ID. If you choose this option, you do not need to code either currentSchema or currentSQLID.

Using the Started Task ID to connect to the database

Another option is to define all of the WebSphere Process Server for z/OS tables with a schema that is equal to the user ID that is associated with the WebSphere servant Job. If you choose this option, you do not need to code either currentSchema or currentSQLID. You do not need to specify any Authentication Alias on your type2 data sources either.

See 2.6.5, “Using the started task ID to connect to the database” on page 60.

Specifying the schema with currentSchema

In the custom properties of a data source, there is a property called currentSchema. If the application makes an unqualified SQL request, and you specify a value in currentSchema, the SQL request is qualified by the value of currentSchema. This is the recommended setting instead of the ‘currentSQLID’ property. We recommend that you use this in all of the data source custom properties, except for the ME datasources, which have the currentSchema set at the Messaging Engine Data store level.

Using the currentSQLID property indirectly allows WebSphere Process Server for z/OS to use the correct schema but brings additional security issues that are unlikely to be acceptable to a security-conscious DB2 system administrator. When using the currentSQLID property, the J2C authentication alias user ID ends up with more privileges than it needs, including DROP. We suggest that you use currentSchema rather than currentSQLID.

> **Note:** Although we suggest that you not use tht currentSQLID property on a data source, there are times when you will need to use the SET CURRENT SQLID command when you execute the SQL to define all of the WebSphere Process Server for z/OS tables in DB2 for z/OS. You must issue the DB2 command `SET CURRENT SQLID=’<schema>’;` before you execute the `CREATE TABLE` statements because some of the SQL is unqualified.

Your choice of currentSQLID or currrentSchema also affects the GRANT commands that you must use, which we discuss in 2.6.4, “Using GRANT statements in DB2” on page 58.

2.6.4 Using GRANT statements in DB2

The GRANT statements that you must issue differs depending on whether you use currentSQLID or currentSchema. We did not have time to explore all of the possibilities, and your particular requirements might differ, but this section will help you plan what you need to do.

To understand the GRANT statements that are required for WebSphere Process Server for z/OS, look at them in two perspectives:

- GRANT privileges require you to create the database and tables, which you can assign to a WebSphere or Database administrator.

- GRANT privileges are required during the runtime operation. Here is a minimum for the authentication user ID that is assigned to the data source custom properties during the runtime (B7DBU in our case):
  - SELECT on SYSIBM.SYSTABLES
  - USAGE on SEQUENCE objects created for the `governancerepository` component
  - EXECUTE on FUNCTIONs
  - TABLE privileges:
    - DBADM privileges on the databases
    - Select, Insert, Update, Delete, and so forth on the specific DB2 objects (Tables, Views, and so on)

We were not using RACF for authorization checking. If DB2 is using RACF for authorization checking, an alternative to GRANTing DBADM to the alias user ID is to connect the alias user ID to the schema RACF group name (B7CELL).

In the next sections, we provide a short overview of the various ways in which DB2 privileges are used.

Objects protected by GRANTs

Here are some of the DB2 objects that are controlled by GRANT privileges:

- DB2 subsystem
- Databases
- GRANT option
- Storage groups
- Buffer pools
- Tables
- Functions
- Sequences

Users to be granted privileges

Several users and groups must be granted access to the DB2 objects, both when the tables are created and during the start-up and running of WebSphere Process Server for z/OS application servers. Identities are assigned to the following users:

- Database administrators
- WebSphere administrators
- J2C Authentication aliases in the data source properties
- WebSphere server started tasks (Controller, Adjunct, Servant)
- Deployment manager started task (Controller and Servant)

Levels of privileges to be granted

The levels of privelges to be granted are:

- SYSADM
- DBADM
- Select, Insert, Update, Delete
- Drop, or Create Tables
- ALL privileges on Tables,

If this all looks a little complicated, do not worry. There are some scripts provided with WebSphere Process Server for z/OS that will help you execute these GRANTs, which we describe in more detail in section 4.4, “Granting privileges to users in DB2” on page 117.

When using currentSQLID custom property

When using currentSQLID = <schema> the user ID executing the SQL has the same identity as the schema and so has all implicit privileges over that schema. Therefore, there is no need to issue additional grants when using currentSQLID set to the schema. GRANTs only need to be given on the database, schema, and storage groups.

When using currentSchema custom property

When using currentSchema, DB2 objects are accessed under the identity of the J2C authentication alias user ID. That user ID can either be GRANTed DBADM privileges to the database (if this is acceptable) or be given specific privileges to allow it to SELECT, UPDATE, INSERT, and DELETE rows in the tables.

Some of the generated SQL statements create GRANT statements for their DB2 objects, but it is necessary to construct GRANT commands based on the CREATE statements that were issued for the other WebSphere Process Server for z/OS components. Because we did not know exactly what kinds of accesses were being performed by WebSphere Process Server for z/OS we issued GRANT ALL PRIVILEGES ON commands for each table.

We used a tool that we described in section 4.4.2, “Using the GRANT Generator” on page 119.

2.6.5 Using the started task ID to connect to the database

One option is to define all of the WebSphere Process Server for z/OS tables with a schema that is equal to the user ID that is associated with the WebSphere servant Job. If you choose this option, you do not need to code either currentSchema or currentSQLID or specify any Authentication Alias on your type2 datasources either.

Unlike specifying an Authentication Alias, this user ID has no password and the ID cannot be used by real users. It is not subject to the same security rules (for example, password expiration).

Started task users might be allowed additional privileges and granted DBADM authority, which might be a a good reason to create a database with this ID and grant the started task's ID DBADM authority for all additional Schemas existing within the database.

Because these database tables are actually part of the WebSphere Process Server product, the WebSphere Process Server server is a logical owner of these tables.

Database scenario using no Authentication Alias

Here is a sample configuration where we can use the user IDs that are associated with the WebSphere Process Server started tasks as the owner of the schema table definitions.

In this scenario, the deployment manager is the owner of the tables that are defined on the Cell Scope datasources. These datasources have no Authentication Alias definitions specified. These definitions are in a separate database because they are common to all clusters and servers within this cell. The database name is B7CELL as will be the owner of the tables.

The table owner is determined by the servant user ID. In this example, we use the cell name B7CELL:.

```
Cell Name= B7CELL<= DMGR Servant UserID       (zServantUserid=B7CELL)
```

A second database is defined for the cluster. This database B7SR1 contains all of the definitions that are shared across the cluster members as part of the DB2 data sharing group. This database contains six separate schemas (B7SR1,B7C1S, B7C1A, B7C1B, B7C1C, and B7C1F) with a schema that the cluster owns and one for each of the Message Engines that are defined within the cluster. Again, these data sources have no Authentication Alias, and the default Owner is the user ID for the servant tasks (B7SR1):

```
Cluster1= B7SR1<= Servant userID(zServantUserid=B7SR1)
```

In this configuration, there are tables that the cell (B7CELL) owns that the cluster owner (B7SR1) must be able to access. These Cell tables require that ALIAS definitions are created for the owner:

```
CREATE ALIAS B7SR1.CUSTPROPERTIES FOR B7CELL.CUSTPROPERTIES;
```

To create these definitions you can run the SQL command in Example 2-7. It generates the ALIAS definitions for each of the Tables that are currently defined within the cell database. This not necessary for the SIB tables because they are fully qualified by the messaging engine data store.

Example 2-7 DB2 SELECT command to generate the 'CREATE ALIAS' sql command

```
SELECT 'CREATE ALIAS B7SR1.'||STRIP(CHAR(NAME))||' FOR
B7CELL.'||STRIP(CHAR(NAME))||' ;'
 FROM SYSIBM.SYSTABLES WHERE DBNAME='B7CELL';
```

Note: Using this approach changes the user IDs that are assigned to the started tasks when the server is initially created. Currently all servants run with the B7ASRU identity, including the servants for the deployment manager.

Reviewing the user ID assignment for DB2 schemas

Figure 2-12 on page 62 shows the eventual network deployment cell with a single cluster depicting the different schema names.

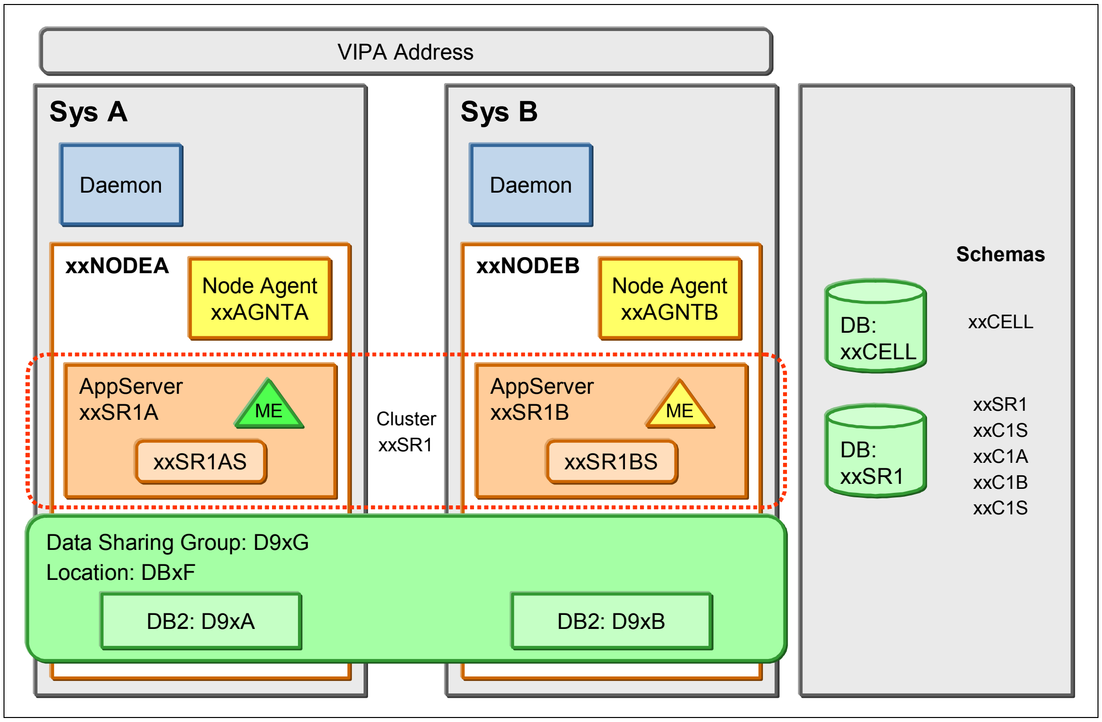


Figure 2-12 Eventual Network Deployment B7CELL single cluster showing schema names

2.6.6 Deciding how to create and execute the SQL

Defining data sources in WebSphere Process Server for z/OS and database objects in DB2 for z/OS involves definitions in both WebSphere Process Server for z/OS and in DB2. Because various names must match in these two places, WebSphere Process Server for z/OS provides shell scripts and Integrated Solutions Console wizards that take various names as input and ensures that the definitions in WebSphere Process Server for z/OS match those in DB2 for z/OS.

Although the shell scripts and wizards hide the details of all of the definitions that are required, an implication is that a user ID is required that has sufficient privileges both to perform administration functions in WebSphere Process Server for z/OS and database administration functions in DB2 for z/OS. If such a user ID does not exist in your organization, avoid creating tables in DB2 for z/OS at the same time that you define the DB2 JDBC provider and data sources in WebSphere Process Server for z/OS.

To avoid granting both WebSphere administration authority and DB2 SYSADM authority to the same user ID, ask your DB2 system administrator to create the databases and stogroups and then issue GRANT statements to delegate authority over the databases, stogroups, and buffer pools to the user ID that executes the SQL. There are two issues to consider:

- Generating and executing the SQL
- Accessing the database objects with SQL, such as SELECT

WebSphere Process Server offers several alternative methods for generating and executing the SQL statements that create the necessary tables and indexes in DB2. One of the early decisions that you must make is the method that you plan to use to generate and execute the SQL statements to create the objects and the method you plan to use to access them.

See Chapter 4, “Configuring the database resources” on page 99 for this discussion.

Note: The Deployment Environment wizard can also create SQL statements for CEI in the following location, but we recommend it not be used because createDB.sh does a better job of customizing it correctly for DB2 z/OS:

```
<profile_root>/databases/event/<cluster_name>/dbscripts/db2zos/ddl
```

2.6.7 Preparing the z/OS environment for DB2

Here are a few items that you must address early while you prepare the environment for configuring WebSphere Process Server for z/OS.

Creating a DB2JccConfiguration.properties file

A DB2JccConfiguration.properties file might be required when using a type 2 data source to connect to DB2 for z/OS. If the default DB2 subsystem ID is not specified in the DSNHDECP load module on your current STEPLIB, this file must contain at least the property db2.jcc.ssid, which specifies the subsystem ID.

The SSID or group sharing name of the DB2 system to which your WebSphere Process Server for z/OS cell will connect, for example, we specified `db2.jcc.ssid=D9FG`, where D9FG is the group-sharing name of our DB2 for z/OS system.

To understand how the DB2JccConfiguration.properties file is used and why we use a file with that long name, in the next section, we provide background information about JDBC driver properties.

DB2 properties file background

When accessing DB2 for z/OS through the DB2 JDBC Universal driver there are several ways to specify the DB2 subsystem to which you want to connect. See *DB2 Universal Database for z/OS Application Programming Guide and Reference for Java*, SC18-7414, for a full discussion of the options. The options are:

- Pass system property -Ddb2.jcc.ssid when invoking Java. For Java-stored procedures, you can code such properties in file referred to by a //JVMPROPS DD card.
- Specify the location of a properties file using the JVM system property db2.jcc.propertiesFile, which is how WebSphere Application Server for z/OS does it.
- Add properties to a file called DB2JccConfiguration.properties, and make that file available to the classpath. It is possible to include the directory that holds the DB2JccConfiguration.properties file on the CLASSPATH, or include the file in a jar file that is included on the CLASSPATH.
- Specify the DB2 subsystem name in the customized DSNHDECP module. Ensure that the customized SDSNEXIT that holds that module is above the SDSNLOAD in any JOBLIB or STEPLIB concatenation of any address space accessing DB2.

When using the second or third options, specify the name of the DB2 subsystem to which you want to connect on the property db2.jcc.ssid, which you code in the properties file. The options previously listed take precedence in the order listed, for example, if the db2.jcc.propertiesFile is not set, DB2 searches the classpath for a file called DB2JccConfiguration.properties. If DB2 also fails to find a file of that name, DB2 tries to contact the DB2 whose SSID is coded in the DSNHEDCP module that is located in the customized SDSNEXIT library.

Using -D JVM system properties

It is not common to pass the JDBC properties individually as JVM system properties. Typically, only the -Ddb2.jcc.propertiesFile is passed to specify the location of a file that contains the JDBC properties.

Using the db2.jcc.propertiesFile system property

In WebSphere Application Server for z/OS, the db2.jcc.propertiesFile system property is normally the method to use to specify the location of the DB2 Universal JDBC driver properties file. The zWPSConfig.sh sets this property when you augment nodes to support WebSphere Process Server.

When you use the db2.jcc.propertiesFile to set the location of the JCC properties file, you can call the properties file anything that you want to call it. We decided to call this file DB2JccConfiguration.properties, so it can also be located through a CLASSPATH search.

Using the DB2JccConfiguration.properties file

When the db2.jcc.propertiesFile is not set, DB2 will search CLASSPATH for a DB2JccConfiguration.properties file. When configuring WebSphere Process Server, batch jobs and shell scripts normally obtain their JCC properties using this method.

The group-sharing name of our DB2 is D9FG, so we created a directory called /etc/d9fg, and created a DB2JccConfiguration.properties file in that directory. We coded db2.jcc.ssid=D9FG in the DB2JccConfiguration.properties file.

> **User IDs require read access to the file:** Both the user ID that runs the configuration jobs and the user ID of the WebSphere Application Server for z/OS Control Region require read access to the db2.jcc.propertiesFile configuration file.
>
> Because /etc is not a shared mount point, we created an /etc/d9fg directory on each LPAR that hosts a node in our cluster. Alternatively, you can create the DB2JccConfiguration.properties file in a directory under a shared mount point, so there is only one copy for the whole sysplex. We put it in the deployment manager's mount point at /wasv7config/b7cell/b7dmnode, so all servers in the cell can access it.

Customizing .profile and .setup files for DB2 users

See 2.4.2, "Customizing .profile and .setup files for users" on page 45 for recommendations about tailoring the .profile files for your UNIX users. In this section, we discuss additions that might be needed for users that must run DB2 scripts.

The default UNIX environment that is specified in /etc/profile might be using old versions of Java and DB2 for z/OS. To ensure that the UNIX command-line environment is customized for the correct version of Java and DB2 Universal JDBC Driver, customize a .profile for each user ID that will work with WebSphere Application Server and DB2 z/OS from the command line or from UNIX batch jobs.

For WebSphere Application Server for z/OS, the administrator user ID and the controller user ID share a home directory, for example, the home directory of the administrator user ID of B7CELL was /var/WebSphere/home/B7CFG.

We created a .profile in /var/WebSphere/home/B7CFG that set PATH, LIBPATH, STEPLIB, and CLASSPATH so that the UNIX environment was correctly customized for Java and JDBC, as shown in Example 2-8 on page 65, which allowed any shell script that we invoked to make JDBC calls and connect to the correct DB2.

JAVA_HOME is set to point to the same Java that the deployment manager uses so that any scripts that are launched under user B7ADMIN will run with the same level of Java as the level of Java that WebSphere Application Server itself uses.

Example 2-8 Part of a .profile showing JAVA_HOME and paths necessary for DB2

```
#
# DB2 setup
#
export DB2_PATH=/usr/lpp/db2/d9fg/db2910_jdbc
STEPLIB=DB9FU.SDSNLOAD:$STEPLIB
STEPLIB=DB9FU.SDSNLOD2:$STEPLIB
STEPLIB=DB9FU.SDSNEXIT:$STEPLIB
PATH=$DB2_PATH/bin:$DB2_PATH/bin:$PATH
LIBPATH=$DB2_PATH/lib:$DB2_PATH/lib:$LIBPATH
CLASSPATH=$DB2_PATH/classes/sqlj.zip:$CLASSPATH
CLASSPATH=$DB2_PATH/classes/db2jcc_javax.jar:$CLASSPATH
CLASSPATH=$DB2_PATH/classes/db2jcc_license_cisuz.jar:$CLASSPATH
CLASSPATH=$DB2_PATH/classes/db2jcc.jar:$CLASSPATH
CLASSPATH=/etc/d9fg/DB2JccConfiguration.properties:$CLASSPATH
export PATH STEPLIB LIBPATH CLASSPATH
#
echo 'JAVA_HOME='$JAVA_HOME
echo 'DB2_PATH='$DB2_PATH
echo 'PATH='$PATH
echo 'STEPLIB='$STEPLIB
echo 'CLASSPATH='$CLASSPATH
echo 'LIBPATH='$LIBPATH
```

In addition to setting a .profile for the WebSphere administrator user ID, it is a good idea for the WebSphere systems programmer to use the same .profile for their personal TSO user ID.

Customizing .profile to use the DB2 command-line processor

Example 2-9 contains some additional lines that you can add to your .profile if you want to use the DB2 command-line processor.

Example 2-9 Additional statements in .profile to use the DB2 command line processor

```
# Set up to use the DB2 command line processor:
export JAVA5=/usr/lpp/java/J5.0/bin/java
export JAVA5_HOME=/usr/lpp/java/J5.0
export JAVA5_64=/usr/lpp/java/J5.0_64/bin/java
export JAVA5_64_HOME=/usr/lpp/java/J5.0_64
#
#  Setup to use the Java level I like.
#
export JAVA_HOME=$JAVA5_HOME
export PATH=$JAVA_HOME/bin:$PATH
export CLPHOME=/usr/lpp/db2/d9fg/db2910_base
export CLASSPATH=$CLASSPATH:$CLPHOME/lib/clp.jar
export CLPPROPERTIESFILE=$HOME/clp.properties
. ./jcc/jcc3_env.sh
```

Creating a .setup file for users

To help with typing in telnet sessions, we also created some alias names in the .setup files in the same home directories as the .profile files. Example 2-10 shows a sample setup file that you can tailor to your preferences.

Example 2-10 Sample addition to setup file

```
alias db2="java com.ibm.db2.clp.db2"
```

Using the syntax db2, you can issue DB2 command-line processor commands. See 2.6.8, "Using the DB2 command line processor" on page 66.

2.6.8 Using the DB2 command line processor

You can use the command line processor (CLP) to run sql statements in the UNIX Systems Services environment from the command-line prompt. For more information, see the following topic and sub-topics in the DB2 V.9 Information Center at:

http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db29.doc.comref/db2z_commanddescriptions.htm

http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db29.doc.inst/db2z_useclp.htm

Setting up to use the command line processor

You must set up several environmental variables and settings to use the CLP:

- Export the following variables, which you can do in the user's .profile, as described in 2.4.2, "Customizing .profile and .setup files for users" on page 45:
  - JAVA_HOME
  - PATH
  - CLASSPATH
  - CLPHOME
  - CLPPROPERTIESFILE
- Create an alias for db2 in the .setup file to invoke the command line processor.
- Create a clp.properties file to specify the semi-colon (;) as the termination character for sql statements, as shown in Example 2-11.

Example 2-11 Sample clp.properties file.

```
#Specify the value as ON/OFF or leave them blank
DisplaySQLCA=ON
AutoCommit=ON
InputFilename=
OutputFilename=
DisplayOutput=
StopOnError=
TerminationChar=;
Echo=ON
StripHeaders=
MaxLinesFromSelect=
MaxColumnWidth=20
IsolationLevel=

#Create your own alias name for DB2 servers. Here we connect to D9F1
```

```
#SERVER1=<URL>,<username>,<password>
D9F1=wtsc42.itso.ibm.com:37894/DB9F,sysadm0,sysadm0
```

- Create a shell script to set environment variables so that the operating system can locate the IBM Data Server Driver for JDBC and SQLJ:

  a. Set the location of the sdk home, jcc home, DB2 OS data set high-level qualifier, and the current directory.

  b. Add the jcc-related jar files to the CLASSPATH.

  c. Add jars from JCC_HOME/classes for jcc.

  d. Add the current directory to pick up classes that are anchored here.

  e. Add the jcc-related directories to the PATH and LIBPATH.

  f. Add the java directory to the PATH, if needed.

  Example 2-12 is a sample shell script to set these variables.

Example 2-12 Sample jcc3_env.sh script.

```
#!/bin/sh
## set the location of the sdk home, jcc home, DB2 OS data set
# highlevel qualifier, and the current directory
#
JAVA5_HOME=/usr/lpp/java/J5.0
JCC_HOME=/usr/lpp/db2/d9fg/db2910_jdbc
DB2_HLQ=DB9F9
BASE_DIR=`pwd`
#
# Add the jcc related jar files to the CLASSPATH
JCC_CLASSPATH=
#
# Add jars from JCC_HOME/classes for jcc
JCC_CLASSPATH=$JCC_HOME/classes/db2jcc.jar
JCC_CLASSPATH=$JCC_HOME/classes/sqlj.zip:$JCC_CLASSPATH
JCC_CLASSPATH=$JCC_HOME/classes/db2jcc_javax.jar:$JCC_CLASSPATH
JCC_CLASSPATH=$JCC_HOME/classes/db2jcc_license_cisuz.jar:$JCC_CLASSPATH
#
# Add the current directory to pick up classes anchored here.
export CLASSPATH="$BASE_DIR":"$JCC_CLASSPATH":"$CLASSPATH"
#
# Add the jcc related directories to the PATH and LIBPATH
export PATH="${JCC_HOME}"/bin:$PATH
export LIBPATH="${JCC_HOME}"/lib:$LIBPATH
#
# Add the java directory to the PATH if needed
whence java | grep -q -e "J5.0" -e "J6.0"
rc=$?
if [ rc -eq 1 ]; then
   export PATH="${JAVA5_HOME}"/bin:$PATH
fi
#
# If SDSNLOAD, SDSNLOD2 and SDSNEXIT are not in the linklist,
# fixup the following statements and uncomment the export statement.

export STEPLIB=${DB2_HLQ}SDSNEXIT:${DB2_HLQ}.SDSNLOAD:${DB2_HLQ}.SDSNLOD2
```

Run this jcc3_env.sh shell script with the . (dot) command to set the environment variables for this session.

- You must also add the following statement to the front of your sql files to connect to the correct DB2 subsystem or group-attach name:

  CONNECT TO D9FG;

Running the command line processor

After you update your .profile, .setup, create the properties file and shell script. You can now run the DB2 command line processor that points to the updated sql file, as shown here:

```
db2 -f <profile_root>/dbscripts/CommonDB/DB2zOSV9/xxx.sql
```

Verifying Java and the DB2 Universal JDBC Driver

After modifying the .profile, we ran a couple of simple tests to check that the command line environment for Java and the DB2 Universal JDBC driver were correct. These tests are not a full verification of JDBC function.

Checking that the Java environment is correct

To check that the Java environment is correct, start an OMVS session, and issue the commands in Example 2-13.

Example 2-13 Testing the Java environment

```
> su b7admin
FSUM5019 Enter the password for b7admin: <enter the password>
> java -fullversion
java full version "J2RE 1.5.0 IBM z/OS build pmz31devifx-20080502 (SR7 + IZ15683 +
IZ21286)"
```

Verifying the DB2 JDBC Universal driver check

To verify the DB2 JDBC Universal driver check, use the **db2jcc -version** command, as shown in Example 2-14.

Example 2-14 Testing that db2jcc can be invoked

```
> db2jcc -version
IBM DB2 JDBC Universal Driver Architecture 3.6.67
```

Using the **db2jcc** command is not a comprehensive test of the JDBC driver; instead, it only confirms that the UNIX command-line environment is good enough to launch db2jcc. If you want to verify that the DB2 Universal Driver is configured correctly, refer to the Infocenter article *Verify the installation of the IBM Data Server Driver for JDBC and SQLJ* at the following web page:

http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db29.doc.java/db2z_jccverifyinstall.htm

2.7 Summary: Looking ahead at the configuration steps

Now that we planned and prepared for the configuration of the BPM topology, let us take a look at the tasks at a high level. Figure 2-13 on page 69 shows where we are headed.

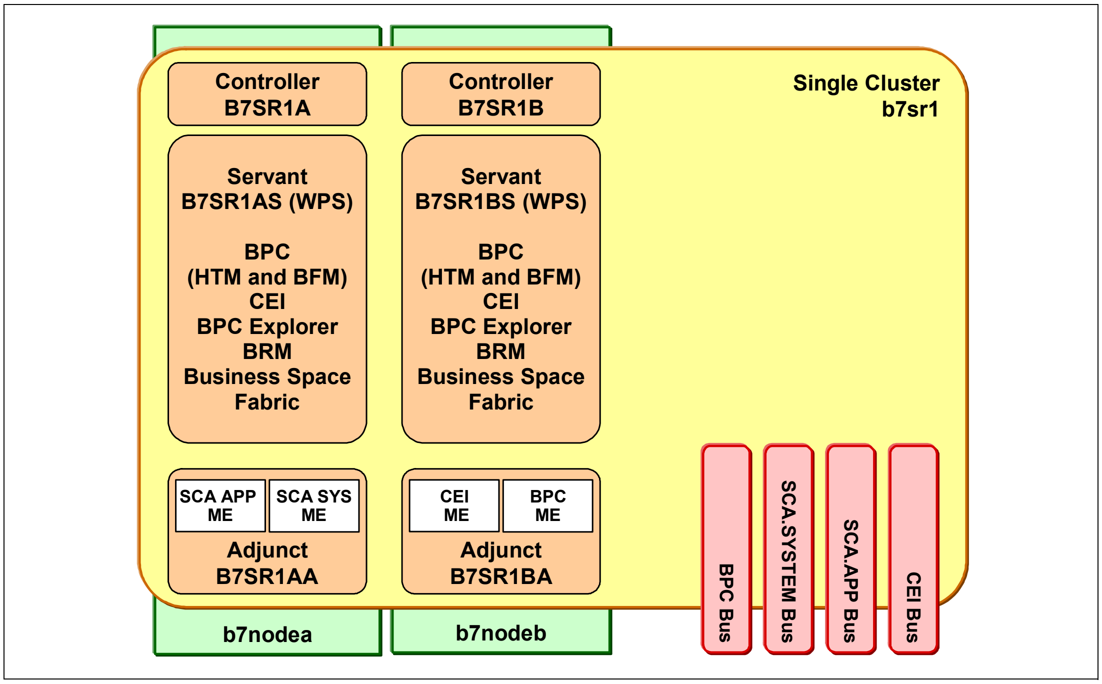


Figure 2-13 Our Goal: A single-cluster, two-node cell with WebSphere Process Server for z/OS V7 configured

2.7.1 Tying the configuration steps back to the spreadsheet

Figure 2-14 on page 70 shows the interaction of the tasks that are involved to configure the cell and how everything comes from the spreadsheet.

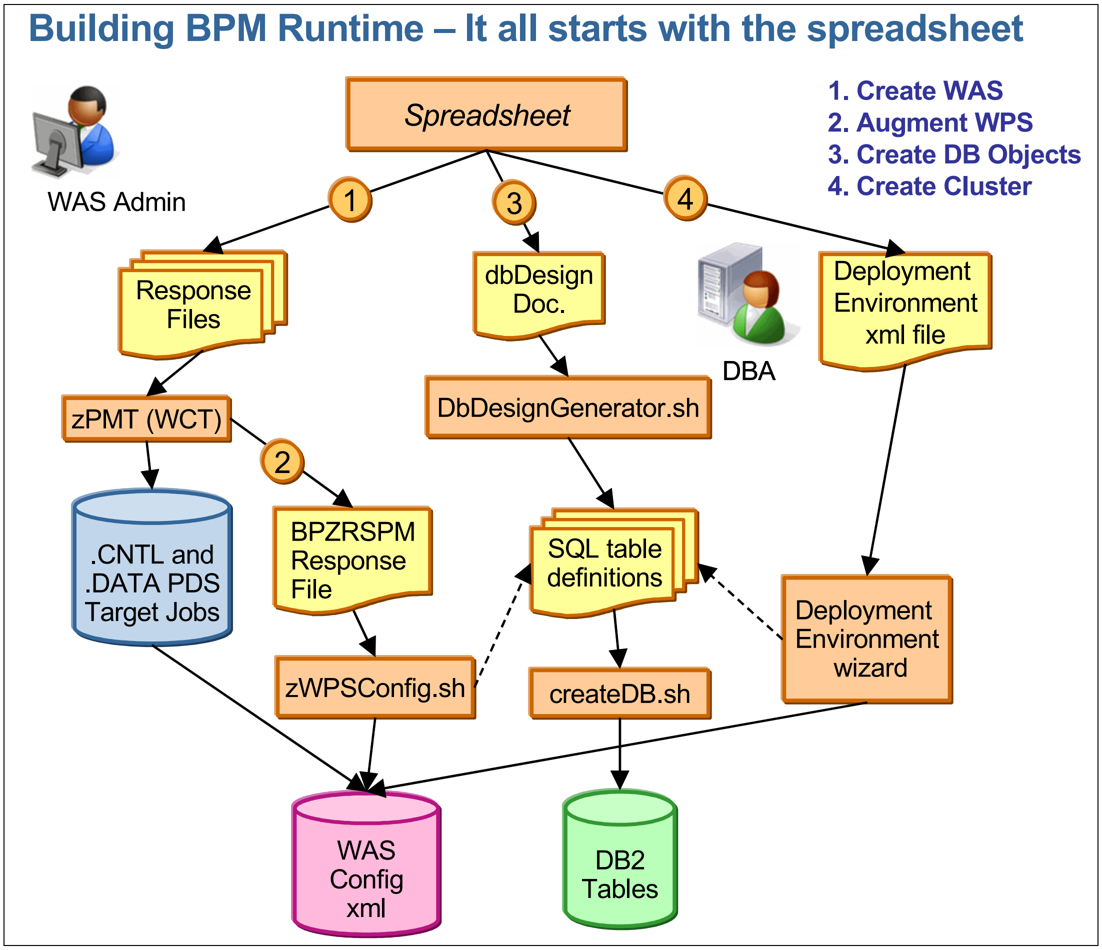


Figure 2-14 Showing the installation steps discussed

The remainder of Part 2 describes how we configure WebSphere Process Server for z/OS:

- Chapter 3, “Creating the WebSphere Process Server deployment manager and two empty nodes” on page 71.
- Chapter 4, “Configuring the database resources” on page 99
- Chapter 5, “Configuring the WebSphere Process Server cluster” on page 129.
- Chapter 6, “Verifying the configuration” on page 195.

3

Creating the WebSphere Process Server deployment manager and two empty nodes

In this chapter, we describe how to set up a WebSphere Process Server for z/OS network deployment environment using an empty node approach. This environment will become a functional WebSphere Process Server for z/OS environment.

In this chapter, we discuss:

3.1 Overview of WebSphere Application Server for z/OS environment

After you finish planning your environment using the guidelines in Chapter 2, “Planning for and preparing WebSphere Process Server” on page 25, you can begin the process of creating a WebSphere Application Server for z/OS environment. The first thing to do, is to set up the WCT on a workstation with FTP access to the target z/OS system.

The latest version of WCT is available for download at:

http://www.ibm.com/support/docview.wss?rs=180&uid=swg24020368

Before using the WCT for the first time, we suggest that you obtain the PowerPoint presentation in the Techdoc *WebSphere Application Server for z/OS V7.0: Introducing the WCT for z/OS*, PRS3357, which you can download from the following web page:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS3357

To update WCT to support WebSphere Process Server for z/OS, the Feature Packs for SCA, XML, and SDO:

1. Download zWBI.wct to your workstation in binary. It comes with WebSphere Process Server for z/OS, and is located in /usr/lpp/zWPS/V7R0/util/WCT/zWBI.wct.
2. Start up your WCT and go to **Help** → **Software Updates** → **Install Extension.**
3. Click **Install new extension location**.
4. Enter the path to your downloaded copy of zWBI.wct, and click **next.**
5. Click **next**, and make sure that you copy the path in the Location field. Copy it into a text editor to verify that it was actually picked up.
6. Click **Finish.**
7. Go to **Help** → **Software Updates** → **Product Configuration**, and click **Add an Extension Location.**
8. Paste the path that you copied in step 5 into the Folder: field.
9. Click **OK.**
10. When prompted for a restart of WCT, click **No.**
11. Repeat steps 1 through 10 for the Feature Packs for SCA, XML, and SDO.
12. After you add all four extensions, restart WCT.

The locations for the .wct files of the Feature Packs are:

- SCA is located at /usr/lpp/zWebSphere_OM/V7R0/FPSCA/util/WCT/sca.wct
- XML is located at /usr/lpp/zWebSphere_OM/V7R0/FPXML/util/WCT/xml.wct
- SDO is located at /usr/lpp/zWebSphere_OM/V7R0/FPSCA/util/WCT/sdo.wct

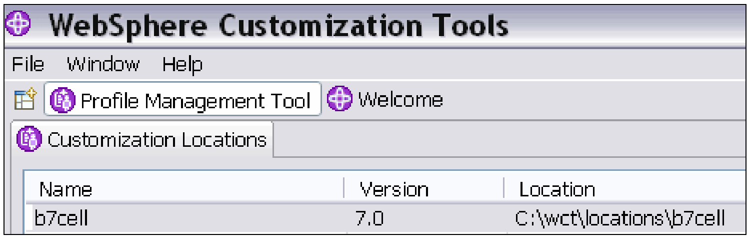
In this chapter, we create the WebSphere Process Server for z/OS Network Deployment environment by first creating the deployment manager. We then create the empty node on the same LPAR, and finish off by creating an empty node on the second LPAR in the SYSPLEX.

3.2 Creating a deployment manager

In this section, we describe how to create a deployment manager.

3.2.1 Generating and uploading the installation jobs using the WebSphere Customization Tool

In the WebSphere Customization Tool, add a version 7 location, as shown in Figure 3-1. Think of the locations as a separation of the environments that you are setting up. In our example, we called it b7cell. All of your definitions are created in your locations folder under your specific location. This folder is fully transportable and can be added to a WebSphere Customization Tool installation on another workstation.



Name	Version	Location
b7cell	7.0	C:\wct\locations\b7cell

Figure 3-1 Profile Management Tool V7, added location

1. Using the planning spreadsheet from Chapter 2, “Planning for and preparing WebSphere Process Server” on page 25, verify that the Variables and Names worksheets are correct. Extract the response file for the deployment manager, as described in the DMGR worksheet. We use this response file as input for the definition creation process.
2. Click **Create** to add a WebSphere Application Server for z/OS Management node definition to your location. Make sure that the WebSphere Application Server for z/OS: Management environment is selected, as shown in Figure 3-2.

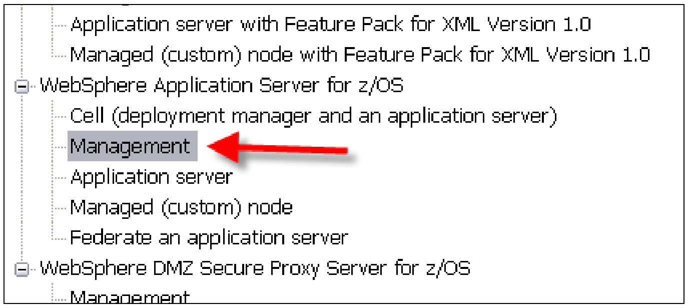

Application server with Feature Pack for XML Version 1.0
Managed (custom) node with Feature Pack for XML Version 1.0
WebSphere Application Server for z/OS
Cell (deployment manager and an application server)
Management
Application server
Managed (custom) node
Federate an application server
WebSphere DMZ Secure Proxy Server for z/OS
Management

Figure 3-2 Profile Management Tool V7, Environment Selection window showing WebSphere Application Server for z/OS: Management

3. Click **Next** and Browse for the location of your response file that you created from the spreadsheet. We provide copies of our response files.
4. As you step through the various windows, verify that the fields are accurately populated, as you specified them in the spreadsheet.
5. Click **Create** to save the customized jobs and property files.
6. Upload the customized jobs by clicking **Process**. Click **Allocate target z/OS data sets**, if you have not allocated the data sets at this time.
7. To add augmentation definitions for WebSphere Process Server for z/OS and the Feature Packs, click **Augment**:
   a. Select **Management**, and click **Next**.
   b. Select **Deployment manager with WebSphere Process Server**, and click **Next**.

c. Browse for the location of your response file that was copied from the spreadsheet, and click **Next**.

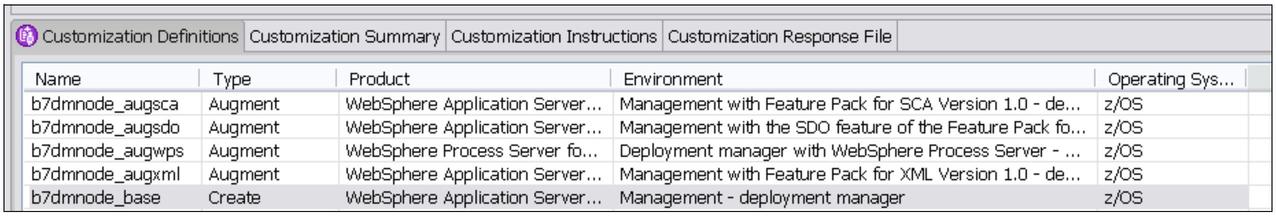
d. As you step through the various windows, verify that the fields are accurately populated, as you specified them in the spreadsheet.

e. If you have a dbDesign file, use it when applying the database variables, if not you must manually enter the correct variables.

f. Click **Augment** to save the customized jobs and property files. Upload the customized jobs by clicking **Process**.

Repeat steps 1 through 4, and step 6 for each of the three Feature Packs, selecting the appropriate Environment in step 2.

Eventually our location consisted of five new definitions, as shown in Figure 3-3.



Customization Definitions
Customization Summary
Customization Instructions
Customization Response File

Name	Type	Product	Environment	Operating Sys...
b7dmnode_augsca	Augment	WebSphere Application Server...	Management with Feature Pack for SCA Version 1.0 - de...	z/OS
b7dmnode_augsdo	Augment	WebSphere Application Server...	Management with the SDO feature of the Feature Pack fo...	z/OS
b7dmnode_augwps	Augment	WebSphere Process Server fo...	Deployment manager with WebSphere Process Server - ...	z/OS
b7dmnode_augxml	Augment	WebSphere Application Server...	Management with Feature Pack for XML Version 1.0 - de...	z/OS
b7dmnode_base	Create	WebSphere Application Server...	Management - deployment manager	z/OS

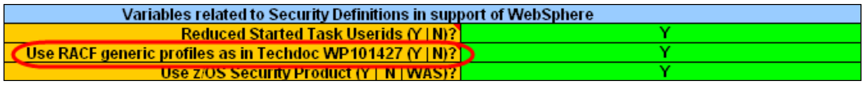
Figure 3-3 Profile Management Tool V7: View shows the five deployment manager definitions needed to augment the node with WebSphere Process Server for z/OS V7

3.2.2 Executing the installation jobs

After the customized jobs are uploaded to the z/OS partitioned data set, you can go to your TSO session and open the target .CNTL PDS, and follow the following instructions to find the order in which to execute the jobs.

The BBOSBRAK and BBODBRAK jobs create RACF user IDs, groups, and profiles, but the RACF profiles are specific to the deployment manager. We recommend that you do not run these jobs; instead, run jobs B7RAC1TM and B7RAC700, which are provided on the JCL_P worksheet of the spreadsheet. These jobs create *generic* RACF profiles, which will apply to any server or cluster in a cell. After running these jobs, you do not need to run any other jobs that are related to RACF and add new nodes or servers to the cell without having to create additional RACF profiles.

On the Variables worksheet, we recommend that you choose ‘Y’ for the option ‘Use RACF generic profiles as in Techdoc WP101427 (Y | N)?, as shown in Figure 3-4. This option customizes the Checklist worksheet of the spreadsheet so that it generates instructions to run B7RAC1TM and B7RAC700 instead of the standard BBOxBRAK jobs that the WebSphere customization generates.



Variables related to Security Definitions in support of WebSphere	
Reduced Started Task Userids (Y | N)?	Y
Use RACF generic profiles as in Techdoc WP101427 (Y | N)?	Y
Use z/OS Security Product (Y | N | WAS)?	Y

Figure 3-4 Choose to use RACF generic profiles as in Techdoc WP101427

The RACF jobs are:

B7RAC1TM Issues the RACF commands that are required to be executed one time (1TM) within the Sysplex, sharing the RACF database. As described in Techdoc WP101427, this is a recommended alternative to executing the standard RACF jobs, BBOSBRAK and BBODBRAK.

B7RAC700 Issues the RACF commands that create user IDs, groups, and generic RACF profiles, setting up RACF for the whole cell and avoiding the need to execute other BBOxBRAK jobs for the node and server/cluster. As described in Techdoc WP101427, this is a recommended alternative to executing the standard jobs, BBOSBRAK and BBODBRAK and BBOMBRAK and BBOWBRAK.

BBOSBRAK Skip this job if you ran B7RAC1TM and B7RAC700. It creates initial WebSphere groups and user IDs.

BBODBRAK Skip this job if you ran B7RAC1TM and B7RAC700. It creates various security definitions for the deployment manager.

> **BBOSBRAK and B7RAC700:** After running BBOSBRAK or B7RAC700, ask the RACF administrator to assign a non-expiring password to the WebSphere Administrator user ID. The RACF command to do this is:
>
> ```
> ALTUSER B7ADMIN PASSWORD(<b7adminpw>) NOEXPIRED
> ```
>
> After the cell configuration is complete, the password can be changed to be non-expiring but only if you are careful not use the WebSphere Administrator user ID as a J2C Authentication Alias.
>
> If you follow the choices that we made in this book, you can avoid that happening.

B7UNIXPR Sets up the UNIX privileges for the installer's user ID which are needed to run the various jobs when the installer's user ID is not UID(0). We recommend not to install WebSphere using a user ID that has UID(0) (superuser) privileges; instead, use the RACF UNIXPRIC controls to permit the necessary access to the file system.

> **Note:** Edit job B7UNIXPR and change xxxxxxxx to the installer's TSO user ID.

BBOSBRAM Creates initial WebSphere home directories in /var/WebSphere/home/ and changes the owning user and groups.

BBODCPY1 Copies customized members to the installation proclib.

BBODCFS Allocates the configuration ZFS and creates a mount point in UNIX System Services.

BBODHFSA Creates the node configuration skeleton.

BBOWWPFD Creates the default profile on the node.

IWODBRAK Creates an EJBROLE for SCA.

IWODAUGM Augments the profile with Feature Pack for SCA.

IWOJAUGM Augments the profile with Feature Pack for XML.

IWOKAUGM Augments the profile with Feature Pack for SDO.

BPZDOLNK Creates an intermediate symbolic link to the WebSphere Process Server product file system.

B7RACFG Creates WebSphere Process Server groups and user IDs when using GEJBROLE profiles or B7RACFE when using EJBROLE profiles.

B7HOMEGA Creates the wpswork directory when using **G**EJBROLE profiles or B7HOME**E**A when using **E**JBROLE profiles. Creates WebSphere Process Server home directories in /var/WebSphere/home/ and changes the owning user and groups.

B7GEJB Creates the GEJBROLE profiles needed for WebSphere Process Server when using GEJBROLE profiles or B7EJB when using EJBROLE profiles.

3.2.3 Verifying that the deployment manager starts up

Before starting up the deployment manager, you must allow the Controller region user ID to issue MVS START commands for servers, servants, and adjuncts. Usually there are RACF generic profiles that are already in place that restrict who can issue the MVS START, STOP, CANCEL, FORCE, and MODIFY commands. The specific PERMIT command that you must issue depends on how your RACF administrator defined the profiles in class OPERCMDS to protect these commands. The sample JCL shown in Example 3-1 is the kind of PERMIT command that is needed, assuming that the profile that controls access to MVS commands is a high-level generic profile called MVS.**, which must be run under a user with sufficient RACF privileges.

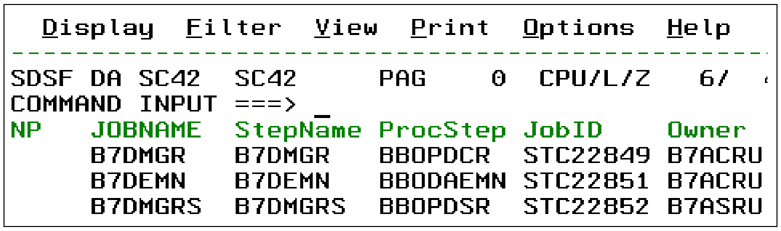
*Example 3-1 JCL to execute the RACF commands that allows B7ACRU to update MVS***

```
//B7PERMIT JOB 1,'IKJEFT01',CLASS=A,REGION=0M,MSGCLASS=H,
//  NOTIFY=&SYSUID
//*
/*JOBPARM SYSAFF=SC42
//TSOCMD EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
PERMIT MVS.** CLASS(OPERCMDS)ID (B7ACRU) ACCESS(UPDATE)
SETROPTS RACLIST(opercmds) REFRESH
/*
```

Probably a better solution is to create new profiles in class OPERCMDS that apply to the cell that you are configuring, and permit the necessary access. The sample job B7RAC700, on which is provided on the JCL_P worksheet of the planning spreadsheet, does this by defining new profiles, such as:

```
RDEFINE OPERCMDS MVS.START.STC.K2*.**   UACC(NONE) OWNER(K2CFG)
PERMIT           MVS.START.STC.K2*.**   CLASS(OPERCMDS) RESET
PERMIT           MVS.START.STC.K2*.**   CLASS(OPERCMDS) +
                                         ID(K2CFG) ACCESS(UPDATE)
```

After the controller region user is granted sufficient rights, start the deployment manager by issuing the command **/S B7DCR,JOBNAME=B7DMGR,ENV=B7CELL.B7DMNODE.B7DMGR**. The controller region and the servant region of the deployment manager will start up alongside the daemon task. After a few minutes, a SDSF Display Active window is displayed that resembles Figure 3-5 on page 77.



```
SDSF DA SC42   SC42      PAG    0  CPU/L/Z   6/
COMMAND INPUT ===> _
NP   JOBNAME   StepName ProcStep JobID    Owner
     B7DMGR    B7DMGR   BBOPDCR  STC22849 B7ACRU
     B7DEMN    B7DEMN   BBODAEMN STC22851 B7ACRU
     B7DMGRS   B7DMGRS  BBOPDSR  STC22852 B7ASRU
```

*Figure 3-5 SDSF, Display active, when looking for tasks starting with b7**

Using the SDSF Display Active window, select your deployment manager's started tasks with the **`pre b7*`** command, and look in the controller region (B7DMGR) and servant region (B7DMGRS).

Select each started task with the S action character, and look for any severe error messages with the **`f severe all`** command. Hopefully, there will be none. If there are, note the number displayed by SDSF, step through each one, and start your problem determination process.

For Java errors, select each started task with the S action character, and look for any java exception messages with the **`f exception all`** command. Hopefully, there will be none. If there are, note the number displayed by SDSF, step through each one, and start your problem determination process. There will be three occurrences of the string *exception* in each of the relevant joblogs, as shown in the Example 3-2.

Example 3-2 Three occurrences of the string ***exception,*** *that can be ignored*

```
BBOM0001I protocol_bboc_log_return_exception: NOT SET, DEFAULT=0.
...
BBOJ0077I: com.ibm.websphere.security.suppressExceptionStack = false
...
ExtendedMessage: BBOJ0077I: com.ibm.websphere.security.suppressExceptionStack =
false
```

Now that you verified that the deployment manager starts up without errors, the next step is to start up the Integrated Solutions Console. Sign on to the Administration Console with a URL of this form:

```
http://<host>:<port_number>/ibm/console
```

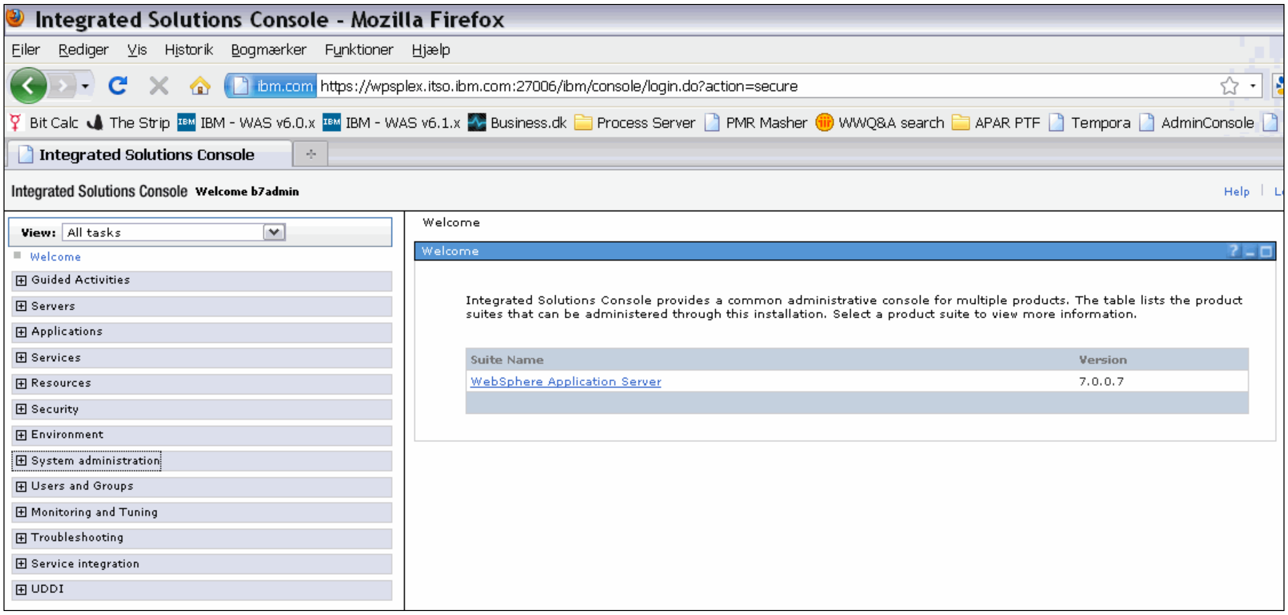
For our deployment manager, the URL was:

```
http://wpsplex.itso.ibm.com:27005/ibm/console
```

Because global security was enabled, we were re-directed to this URL:

```
https://wpsplex.itso.ibm.com:27006/ibm/console/
```

Logon using the WebSphere administrative user. In our example, the username is b7admin and the password is b7admin, as shown in Figure 3-6 on page 78.


Integrated Solutions Console - Mozilla Firefox
Filer Rediger Vis Historik Bogmærker Funktioner Hjælp
ibm.com https://wpsplex.itso.ibm.com:27006/ibm/console/login.do?action=secure
Bit Calc The Strip IBM - WAS v6.0.x IBM - WAS v6.1.x Business.dk Process Server PMR Masher WWQ&A search APAR PTF Tempora AdminConsole
Integrated Solutions Console
Integrated Solutions Console Welcome b7admin
Help
View: All tasks
Welcome
Guided Activities
Servers
Applications
Services
Resources
Security
Environment
System administration
Users and Groups
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Welcome
Welcome

Integrated Solutions Console provides a common administrative console for multiple products. The table lists the product suites that can be administered through this installation. Select a product suite to view more information.

Suite Name	Version
WebSphere Application Server	7.0.0.7

Figure 3-6 Integrated Solutions Console, welcome window

3.2.4 Backing up the deployment manager's configuration data set

Make a practice of backing up the configuration zFS file systems after any milestone and before you do anything that might corrupt it, such as augmenting it for an additional product. The Checklist worksheet in the spreadsheet suggests appropriate times to take a backup.

We used the JCL shown in Example 3-3 to perform the back up. You can find sample job B7BAKDMG on the JCL_P worksheet in the spreadsheet.

Example 3-3 JCL to back up the deployment manager's node

```
//B7BAKDMG   JOB (0),'WPS SETUP',CLASS=A,REGION=0M,
// MSGCLASS=H,NOTIFY=&SYSUID
/*JOBPARM    SYSAFF=SC42
//*-----------------------------------------------------*/
//* Take a backup of the DMGR configuration ZFS         */
//* Add VOL=SER= to the DUMP DD if not using DFSMS      */
//*-----------------------------------------------------*/
//*
//DFDSS    EXEC  PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//DUMP     DD DSN=WASCFG.B7CELL.B7DMNODE.DUMP.PREINS,
//            UNIT=SYSDA,
//            SPACE=(CYL,(500,100),RLSE),DISP=(,CATLG)
//SYSIN    DD  *
 DUMP OUTDDNAME(DUMP) -
      DATASET(INCLUDE( -
                      WASCFG.B7CELL.B7DMNODE.CONFIG.ZFS -
                     ) -
              ) -
 TOL(ENQF)  OPTIMIZE(4) COMPRESS WAIT(2,2) CANCELERROR
/*
```

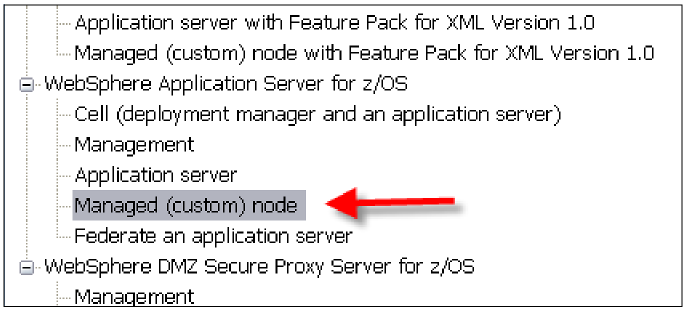
3.3 Creating the primary empty node

In this section, we describe the steps to create a primary empty node. The term *primary node* is a short-hand way to refer to the managed node that is configured first. After the cluster is configured there is no difference in the configuration between the primary node and any secondary nodes.

3.3.1 Generating and uploading installation jobs using WebSphere Customization Tool

To generate and upload the installation jobs using the WebSphere Customization Tool:

1. Using the planning spreadsheet from Chapter 2, “Planning for and preparing WebSphere Process Server” on page 25, verify that the variables and names worksheets are correct.
2. Extract the response file for the primary empty node, as described in the EmptyNode_P worksheet. This response file is used as input for the definition creation process.
3. Add a WebSphere Application Server for z/OS empty node definition to your location. Make sure that the WebSphere Application Server for z/OS: Managed (custom) node environment is selected, as shown in Figure 3-7.



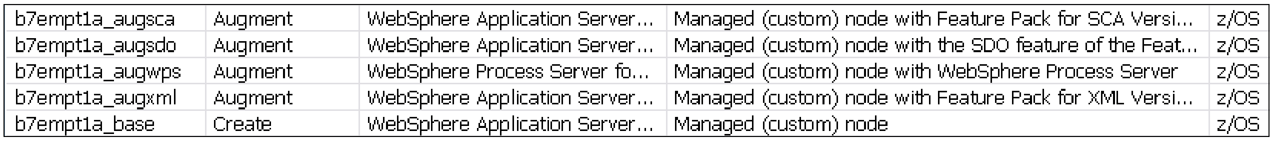
Application server with Feature Pack for XML Version 1.0
Managed (custom) node with Feature Pack for XML Version 1.0
WebSphere Application Server for z/OS
Cell (deployment manager and an application server)
Management
Application server
Managed (custom) node
Federate an application server
WebSphere DMZ Secure Proxy Server for z/OS
Management

Figure 3-7 Profile Management Tool V7, Environment Selection window showing WebSphere Application Server for z/OS: Managed (custom) node

4. Click **Next** and **Browse** for the location of your response file that you created from the spreadsheet. We provide copies of our response files.
5. As you step through the various windows, verify that the fields are accurately populated as you specified them in the spreadsheet.
6. Click **Create** to save the customized jobs and property files. Upload the customized jobs by clicking **Process**. Select **Allocate target z/OS data sets**, if you have not allocated the data sets at this time.
7. To add augmentation definitions for WebSphere Process Server for z/OS and the Feature Packs, click **Augment**:
   a. Select **Managed (custom) node**, and click **Next.**
   b. Select **Managed (custom) node with WebSphere Process Server**, and click **Next.**
   c. Browse for the location of your response file, which was copied from the spreadsheet, and click **Next.**
   d. As you step through the various windows, verify that the fields are accurately populated, as you specified them in the spreadsheet.
   e. Click **Augment** to save the customized jobs and property files. Upload the customized jobs by clicking **Process**.

8. Repeat steps 1 through 5 for the three Feature Packs and for WebSphere Process Server, selecting the appropriate Environment in step 2.

Eventually our location consisted of five new definitions, as per Figure 3-8.



b7empt1a_augsca	Augment	WebSphere Application Server...	Managed (custom) node with Feature Pack for SCA Versi...	z/OS
b7empt1a_augsdo	Augment	WebSphere Application Server...	Managed (custom) node with the SDO feature of the Feat...	z/OS
b7empt1a_augwps	Augment	WebSphere Process Server fo...	Managed (custom) node with WebSphere Process Server	z/OS
b7empt1a_augxml	Augment	WebSphere Application Server...	Managed (custom) node with Feature Pack for XML Versi...	z/OS
b7empt1a_base	Create	WebSphere Application Server...	Managed (custom) node	z/OS

Figure 3-8 Profile Management Tool V7, view showing the five primary empty node definitions needed to augment the node with WebSphere Process Server for z/OS V7

3.3.2 Executing the installation jobs

After the customized jobs are uploaded to the z/OS partitioned data set, to execute the installation jobs:

1. Go to your TSO session.
2. Open the .CNTL PDS, and follow the instructions in member BBOMNINS. These instructions are only for WebSphere Application Server.

The following task list includes the extra jobs that are required to augment the node for the Feature Packs and for WebSphere Process Server:

BBOMBRAK Creates security definitions for the managed node; however, you must *not* run job BBOMBRAK if you ran B7RAC1TM and B7RAC700 to create generic RACF profiles that apply to the whole cell.

BBOWCPY1 Copies customized members to the installation proclib.

BBOMCFS Allocates the configuration ZFS and creates a mount point in UNIX System Services.

BBOHFSA Creates the node configuration skeleton.

BBOWWPFM Creates the default profile on the node.

IWODAUGN Augments the profile with Feature Pack for SCA.

IWOJAUGN Augments the profile with Feature Pack for XML.

IWOKAUGN Augments the profile with Feature Pack for SDO.

BPZDOLNK Creates an intermediate symbolic link to the WebSphere Process Server product file system.

3.3.3 Backing up the primary empty node's configuration data set

We used the JCL in Example 3-4 to perform the back up.

Example 3-4 JCL to back up the primary empty node’s configuration

```
//B7BAKNA   JOB (0),'WPS SETUP',CLASS=A,REGION=0M,
// MSGCLASS=H,NOTIFY=&SYSUID
/*JOBPARM    SYSAFF=SC42
//*----------------------------------------------------*/
//* Take a backup of the primary node's config ZFS      */
//*                                                     */
//* Choose a dump dataset name that describes where you */
//* you are in the configuration process.               */
//* Add VOL=SER= to the DUMP DD if not using DFSMS.     */
//*----------------------------------------------------*/
//*
//DFDSS    EXEC  PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//DUMP     DD DSN=WASCFG.B7CELL.B7NODEA.DUMP.PREINS,
//           UNIT=SYSDA,
//           SPACE=(CYL,(500,100),RLSE),DISP=(,CATLG)
//SYSIN    DD  *
 DUMP OUTDDNAME(DUMP) -
      DATASET(INCLUDE( -
                      WASCFG.B7CELL.B7NODEA.CONFIG.ZFS -
                     ) -
              ) -
 TOL(ENQF)  OPTIMIZE(4) COMPRESS WAIT(2,2) CANCELERROR
/*
```

3.4 Creating the secondary empty node

In this section, we provide steps to create a secondary empty node.

3.4.1 Generating and uploading installation jobs using WebSphere Customization Tool

To generate and upload the installation jobs using the WebSphere Customization Tool:

1. Using the planning spreadsheet from Chapter 2, “Planning for and preparing WebSphere Process Server” on page 25, verify that the variables and names worksheets are correct.
2. Extract the response file for the secondary empty node, as described in the EmptyNode_S worksheet. This response file is used as input for the definition creation process.
3. Add a WebSphere Application Server for z/OS empty node definition to your location. Make sure that the WebSphere Application Server for z/OS: Managed (custom) node environment is selected, as shown in Figure 3-9.

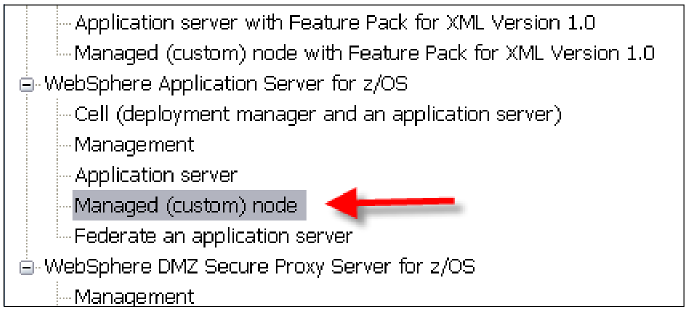

Application server with Feature Pack for XML Version 1.0
Managed (custom) node with Feature Pack for XML Version 1.0
WebSphere Application Server for z/OS
Cell (deployment manager and an application server)
Management
Application server
Managed (custom) node
Federate an application server
WebSphere DMZ Secure Proxy Server for z/OS
Management

Figure 3-9 Profile Management Tool V7, Environment Selection window showing WebSphere Application Server for z/OS: Managed (custom) node

4. Click **Next**, and then click **Browse** to search for the location of your response file that you created from the spreadsheet. We provide copies of our response files.
5. As you step through the various windows, verify that the fields are accurately populated as you specified them in the spreadsheet.

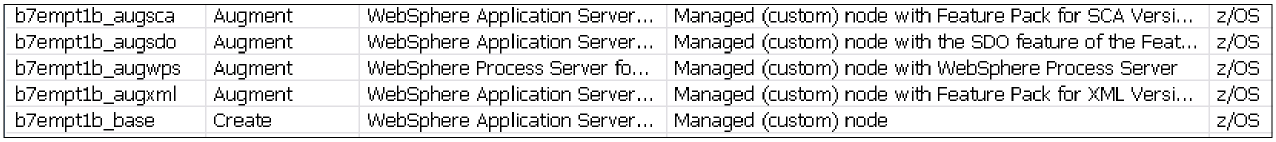
6. Click **Create** to save the customized jobs and property files. Upload the customized jobs by clicking **Process**. Click **Allocate target z/OS data sets**, if you have not allocated the data sets at this time.
7. Click **Next** and **Browse** for the location of your response file that you created from the spreadsheet. We provide copies of our response files.
8. As you step through the various windows, verify that the fields are accurately populated, as you specified them in the spreadsheet.
9. Click **Create** to save the customized jobs and property files. Upload the customized jobs by clicking **Process**. Click **Allocate target z/OS data sets**, if you have not allocated the data sets at this time.

To add augmentation definitions for WebSphere Process Server for z/OS and the Feature Packs, click **Augment**:

1. Select **Managed (custom) node**, and click **Next.**
2. Select **Managed (custom) node** with WebSphere Process Server, and click **Next.**
3. **Browse** for the location of your response file, which was copied from the spreadsheet, and click **Next.**
4. As you step through the various windows, verify that the fields are accurately populated, as you specified them in the spreadsheet.
5. Click **Augment** to save the customized jobs and property files. Upload the customized jobs by clicking **Process.**

Repeat steps 1 through 5, selecting the appropriate Environment in step 2.

Eventually our location consisted of five new definitions, as shown in Figure 3-10.



b7empt1b_augsca	Augment	WebSphere Application Server...	Managed (custom) node with Feature Pack for SCA Versi...	z/OS
b7empt1b_augsdo	Augment	WebSphere Application Server...	Managed (custom) node with the SDO feature of the Feat...	z/OS
b7empt1b_augwps	Augment	WebSphere Process Server fo...	Managed (custom) node with WebSphere Process Server	z/OS
b7empt1b_augxml	Augment	WebSphere Application Server...	Managed (custom) node with Feature Pack for XML Versi...	z/OS
b7empt1b_base	Create	WebSphere Application Server...	Managed (custom) node	z/OS

Figure 3-10 Profile Management Tool V7 view showing the five secondary empty node definitions needed to augment the node with WebSphere Process Server for z/OS V7

3.4.2 Executing the installation jobs

To execute the installation jobs:

1. After the customized jobs are uploaded to the z/OS partitioned data set, go to your TSO session and open the target .CNTL PDS.
2. Use the following list to find the order in which to execute the jobs:

BBOMBRAK Creates security definitions for the managed node; however, you must *not* run job BBOMBRAK if you ran B7RAC1TM and B7RAC700 to create generic RACF profiles that apply to the whole cell.

BBOSBRAM Creates initial WebSphere home directories in /var/WebSphere/home/, and changes the owning user and groups.

BBOWCPY1 Copies customized members to the installation proclib.

BBOMCFS Allocates the configuration ZFS and creates a mount point in UNIX System Services.

BBOHFSA Creates the node configuration skeleton.

BBOWWPFM Creates the default profile on the node.

IWODAUGN Augments the profile with Feature Pack for SCA.

IWOJAUGN Augments the profile with Feature Pack for XML.

IWOKAUGN Augments the profile with Feature Pack for SDO.

BPZDOLNK Creates an intermediate symbolic link to the WebSphere Process Server product file system.

B7HOMEGB Creates WebSphere Process Server home directories in /var/WebSphere/home/ and changes the owning user and groups when using **G**EJBROLE profiles or B7HOME**E**B when using **E**JBROLE profiles.

3.4.3 Backing up the secondary empty node's configuration data set

We used the JCL in Example 3-5 to perform the back up. You can find this job on the JCL_S worksheet of the spreadsheet.

Example 3-5 JCL to back up the secondary empty node's configuration

```
//B7BAKNB   JOB (0),'WPS SETUP',CLASS=A,REGION=0M,
// MSGCLASS=H,NOTIFY=&SYSUID
/*JOBPARM    SYSAFF=SC53
//*-----------------------------------------------------*/
//* Take a backup of the secondary node's config ZFS    */
//*                                                     */
//* Choose a dump dataset name that describes where you */
//* you are in the configuration process.               */
//* Add VOL=SER= to the DUMP DD if not using DFSMS.     */
//*-----------------------------------------------------*/
//*
//DFDSS    EXEC  PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//DUMP     DD DSN=WASCFG.B7CELL.B7NODEA.DUMP.PREINS,
//            UNIT=SYSDA,
//            SPACE=(CYL,(500,100),RLSE),DISP=(,CATLG)
//SYSIN    DD  *
 DUMP OUTDDNAME(DUMP) -
      DATASET(INCLUDE( -
                      WASCFG.B7CELL.B7NODEB.CONFIG.ZFS -
                     ) -
              ) -
 TOL(ENQF)  OPTIMIZE(4) COMPRESS WAIT(2,2) CANCELERROR
/*
```

3.5 Adding WebSphere Process Server to the deployment manager

After you define a deployment manager, you must install support for WebSphere Process Server for z/OS. This process involves running two shell scripts, zWPSInstall.sh and zWPSConfig.sh.

In WebSphere Process Server for z/OS V6.2, it was necessary to create a response file that can be input to the zWPSConfig.sh. The response file is provided as a sample in the <WPSV6_install_root>/zos.config/samples directory. The spreadsheet that was provided with previous versions of this book included a worksheet called WPS.rsp that showed you how to customize the response file that was passed as input to zWPSConfig.sh. Those samples are still provided in WebSphere Process Server for z/OS V7, but the zPMT within WebSphere Customization Tool (WCT) V7 now generates a response file when you augment a node for WebSphere Process Server for z/OS. The WCT places the response file in the .DATA data set for the node, and you can run zWPSConfig.sh using that response file as input. Consequently, the WPS.rsp worksheet is no longer needed in the spreadsheet that helps configure WebSphere Process Server for z/OS *V7* and is removed.

Also, when augmenting the deployment manager's node for WebSphere Process Server for z/OS, you can choose to import a dbDesign. Importing a dbDesign allows you to have more control over the nature of the databases and Data Definition Language (DDL) statements for WebSphere Process Server for z/OS, and we recommend that you use the dbDesign.

You can create a dbDesign using the `dbDesignGenerator.sh`, which we describe in 4.1.3, "Overview of the database configuration process" on page 102 and 4.2.1, "Creating the database design (dbDesign) document" on page 104. However the spreadsheet that accompanies this book now includes a worksheet called dbDesign that generates a dbDesign for you. The instruction at the top of the dbDesign worksheet tells you to copy the contents of column A into, for example, file `/wasv7config/b7cell/wpswork/b7.dbDesign`.

You can use that dbDesign during the execution of `zWPSConfig.sh`, which will generate DDL according to the names and bufferpool choices in the spreadsheet. Alternately, you can first use the dbDesign as input to `dbDesignGenerator.sh`, which will allow you to verify the syntax of the dbDesign and to review or make minor changes. However you are recommended to try and make such changes on the spreadsheet so that it contains all of the current settings that are being used in the cell or in DB2 for z/OS.

The response files that are generated on the DMGR, EmptyNode_P, and EmptyNode_S worksheets are *input* to the WCT. The response file that is generated on the DMGR worksheet includes the property `dbDesignEnabled=true`. When you import that response file into the WCT during augmentation of the deployment manager's node, `dbDesignEnabled=true` pre-populates a WCT panel with the choice to use the dbDesign. The property `wbidbDesign` sets the path to the dbDesign file.

> **Note:** The DMGR worksheet also includes several other properties with names that start with '**db**'. Mostly, these are *not* to be used by the WCT when you specify `dbDesignEnabled=true` because all of the details are in the dbDesign document. They are present on the DMGR worksheet in case you prefer not to use a dbDesign. In that case, the additional panels that you must complete in the WCT when augmenting the deployment manager's node are pre-populated with the values that you planned from the spreadsheet. Because the dbDesign is a much better solution, the spreadsheet uses `dbDesignEnabled=true`.

Property `dbOutputScriptDir` specifies the location where the DDL is generated. We recommend that you keep this location as the default because the createDB.sh that you will execute later expects the DDL to be located at the default location which is, for example, /wasv7config/b7cell/b7dmnode/DeploymentManager**/util/dbUtils**.

Figure 3-11 on page 85 illustrates the relationship between the spreadsheet worksheets, the response file and the dbDesign inputs to WCT, and the outputs of the WCT and zWPSConfig.sh.

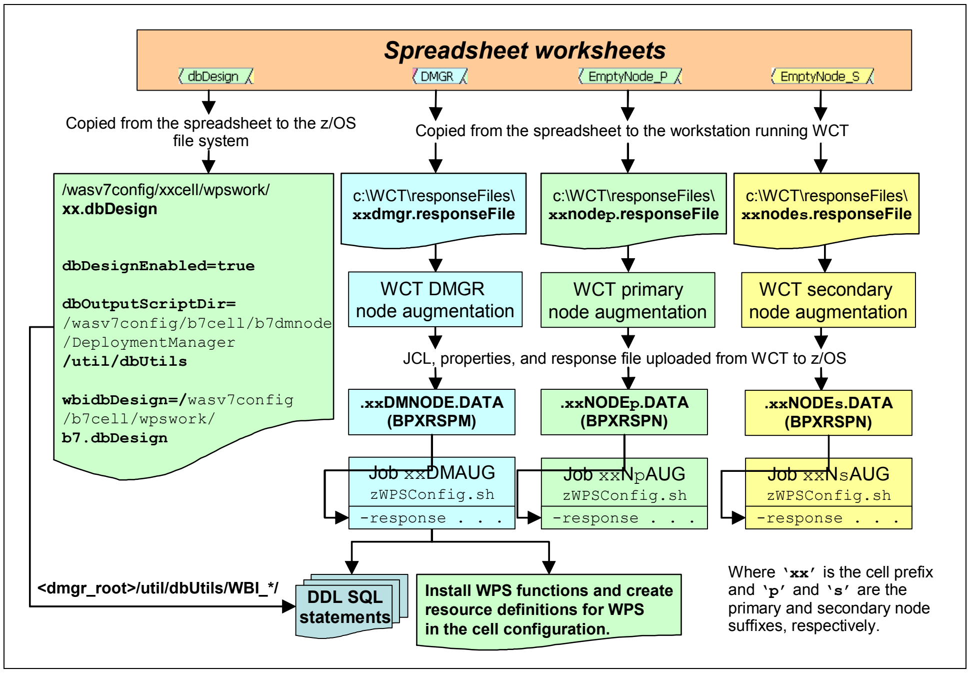


Figure 3-11 The spreadsheet inputs to WCT, and the outputs from WCT and zWPSConfig.sh

3.5.1 Running zSMPInstall.sh for the deployment manager

When invoked with the **-install** option, the zSMPInstall.sh script performs the following tasks:

- Validates the prerequisites.
- Creates the symbolic links from the WebSphere Process Server for z/OS installation file system to the deployment manager's /lib and /bin directories in the configuration file system.
- Updates the code base permissions.
- Updates the Integrated Solutions Console with the WebSphere Process Server for z/OS product definitions, including definitions for the following components:
  - Business Process Choreographer (BPC)
  - WebSphere Process Server for z/OS core resources
  - Common Event Infrastructure (CEI)

Before running the script, ensure that the deployment manager is stopped.

Copy the JCL xxDMINS in Example 3-6 from the JCL_P worksheet of the planning spreadsheet to the target data set .CNTL.

Example 3-6 B7DMINS JCL to run zSMPInstall.sh for the deployment manager node

```
//B7DMINS  JOB (0),'WPSV7 SETUP',CLASS=A,REGION=0M,
// MSGCLASS=H,NOTIFY=&SYSUID,
//*
/*JOBPARM SYSAFF=SC42
//*----------------------------------------------------*/
//* Run zWPSInstall.sh for DMGR's node                 */
//*----------------------------------------------------*/
//*      1         2         3         4        5        6         7
```

```
//ZSMPEIN  EXEC  PGM=IKJEFT01,TIME=1440
//SYSTSPRT DD  SYSOUT=*
//BPXOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//SYSTSIN  DD  *
 BPXBATCH  SH +
 cd  /wasv7config/b7cell/b7dmnode/DeploymentManager/bin; +
 export PATH=.:$PATH; +
 /wasv7config/b7cell/b7dmnode/wpssmpe/zos.config/bin/+
 zWPSInstall.sh +
 '-smproot /wasv7config/b7cell/b7dmnode/wpssmpe' +
 '-runtime /wasv7config/b7cell/b7dmnode/DeploymentManager' +
 '-install'
/*
```

The output from this job ends with the lines in Example 3-7, if successful.

Example 3-7 Output from B7DMINS

```
CWPIZ0253I: parsing command arguments...
CWPIZ0254I: parsing arguments complete
CWPIZ0255I: setting up configuration...
CWPIZ0256I: set up configuration complete
CWPIZ0257I: creating the symbolic links...
CWPIZ0259I: creation of symbolic links complete
CWPIZ0260I: doing post install file updates...
CWPIZ0262I: post install updates complete
CWPIZ0263I: running Configuration Manager update...
CWPIZ0264I: Configuration Manager update complete
OSGi cache successfully cleaned for
/wasv7config/b7cell/b7dmnode/DeploymentManager/profiles/default.
```

3.5.2 Copying the dbDesign from the spreadsheet to the UNIX file system

Parameters and their values are passed to the zWPSConfig.sh in a response file. That response file is generated by the WCT into the .DATA data set that was uploaded to z/OS. The response file generated on the DMGR worksheet includes the property `dbDesignEnabled=true`, which means that when you execute zWPSConfig.sh, it generates DDL statements and configures the WebSphere Process Server for z/OS database resources according to the dbDesign document.

You must copy the dbDesign document from column A of the dbDesign worksheet into the z/OS UNIX file system. We suggest that the best location is the /wpswork directory, and the instruction at the top of the dbDesign worksheet tells you exactly where to copy the dbDesign and what to call it, so that this location and file name matches what `zWPSConfig.sh` expects to receive as input in the next task.

The dbDesign must be in ASCII, so copy column A from the dbDesign worksheet of the planning spreadsheet into a file on your workstation, and then upload it in BINARY mode to the `/wpswork` directory on z/OS.

3.5.3 Running zWPSConfig.sh for the deployment manager

Before you run zWPSConfig.sh, it is a good idea to re-run the B7BAKDMG job to take another backup of the deployment manager's configuration file system.

The zWPSConfig.sh augments the node configuration with WebSphere Process Server for z/OS. Copy xxDMAUG, from the JCL_P worksheet of the planning spreadsheet, to the target data set .CNTL. We ran the script by submitting job B7DMAUG, as shown in Example 3-8.

Example 3-8 B7DMAUG JCL to run zWPSConfig.sh for the deployment manager node

```
//B7DMAUG  JOB (0),'WPS SETUP',CLASS=A,REGION=0M,
// MSGCLASS=H,NOTIFY=&SYSUID,
// USER=B7ADMIN
/*JOBPARM SYSAFF=SC42
//*-----------------------------------------------------*/
//*  Deployment Manager node.                           */
//*  Run zWPSConfig.sh using the response file created  */
//*  by WCT in WASCFG.B7DMNODE.DATA(BPZRSPM) as input.  */
//*-----------------------------------------------------*/
//*
//*-----------------------------------------------------*/
//*  Copy response file to the UNIX file system (/tmp)  */
//*-----------------------------------------------------*/
//COPY2USS  EXEC  PGM=IKJEFT1A
//SYSEXEC    DD DISP=SHR,
//           DSN=WASCFG.B7DMNODE.DATA
//SYSTSPRT  DD SYSOUT=*
//BPXOUT    DD SYSOUT=*
//SYSTSIN    DD *
 BBOWBOWN WASCFG.B7DMNODE.DATA(BPZRSPM) +
 /tmp/bpzrspm_40279  B7ACRU B7CFG
/*
//*-----------------------------------------------------*/
//*  Copy response file to joblog                       */
//*-----------------------------------------------------*/
//COPYRSP   EXEC PGM=IKJEFT01,REGION=0M,DYNAMNBR=300
//SYSTSPRT  DD SYSOUT=*
//BPXOUT    DD SYSOUT=*
//RSPOUT    DD PATH='/tmp/bpzrspm_40279'
//STDOUTL   DD SYSOUT=*,DCB=(RECFM=VB,LRECL=255,BLKSIZE=0)
//SYSPRINT  DD SYSOUT=*
//SYSTSIN    DD *
 OCOPY INDD(RSPOUT) OUTDD(STDOUTL)
/*
//*-----------------------------------------------------*/
//* Run zWPSConfig.sh for DMGR's node                   */
//*-----------------------------------------------------*/
//*
//ZWPSCFG  EXEC  PGM=IKJEFT01,TIME=1440
//SYSTSPRT DD  SYSOUT=*
//BPXOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//SYSTSIN  DD  *
 BPXBATCH SH +
 cd  /wasv7config/b7cell/b7dmnode/DeploymentManager/bin; +
 ./zWPSConfig.sh +
 -response /tmp/bpzrspm_40279 +
 -augment ;
//-------------------------------------------------------------------*/
//*  Delete response file from /tmp.                                 */
//-------------------------------------------------------------------*/
//DELETER1 EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD  SYSOUT=*
//BPXOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//SYSTSIN  DD  *
 BPXBATCH SH +
 rm /tmp/bpzrspm_40279 ;
/*
```

The output from this job ends with the lines shown in Example 3-9, if successful.

Example 3-9   Output from B7DMCFG

```
CWPIZ0253I: parsing command arguments...
CWPIZ0254I: parsing arguments complete
CWPIZ0255I: setting up configuration...
CWPIZ0256I: set up configuration complete
CWPIZ0265I: augmenting profile(s)...
INSTCONFSUCCESS: Profile augmentation succeeded.
CWPIZ0267I: augmenting profile(s) complete
```

3.5.4 Verifying that deployment manager starts up

Before starting the deployment manager, if the DB2 for z/OS SDSNEXIT, SDSNLOAD, and SDSNLOD2 are not in the system LNKLST, add a STEPLIB to the started task JCL of the deployment manager's controller and servant. For our B7 cell, we added the following STEPLIB to members B7DCR and B7DSR in SYS1.PROCLIB:

```
//STEPLIB    DD DISP=SHR,DSN=DB9F9.SDSNEXIT
//           DD DISP=SHR,DSN=DB9F9.SDSNLOAD
//           DD DISP=SHR,DSN=DB9F9.SDSNLOD2
```

Start the deployment manager to confirm that it initializes properly before you configure the empty nodes. We used the following command:

```
START B7DCR,JOBNAME=B7DMGR,ENV=B7CELL.B7DMNODE.B7DMGR
```

Database errors: At the moment, ignore any database errors because we have not configured the database yet.

You might want to backup the deployment manager node's configuration at this point using the B7BAKDMG job. However the next change to the deployment manager is when you federate the nodes into the cell, so you might prefer to take a backup of *all* the configuration datasets later when you are ready to federate the node. Use the sample job B7BAKZFS that is provided on the JCL_P worksheet to backup all of the configuration datasets.

3.6 Adding WebSphere Process Server to the primary empty node

After augmenting the deployment manager's node, the next step is to augment the primary empty node with WebSphere Process Server for z/OS. As before, you must execute `zWPSInstall.sh` and `zWPSConfig.sh`. The response file input for the `zWPSConfig.sh` was generated by the WCT into the .DATA target data set as member BPZRSPN.

3.6.1 Running zSMPInstall.sh for the primary node

Copy xxNAINS, from the JCL_P worksheet of the planning spreadsheet to the target data set .CNTL, as shown in Example 3-10.

Example 3-10 B7NAINS JCL to run zSMPInstall.sh for the primary empty node

```
//B7NAINS JOB (0),'WPSV7 SETUP',CLASS=A,REGION=0M,
// MSGCLASS=H,NOTIFY=&SYSUID,
//*
/*JOBPARM SYSAFF=SC42
//*-----------------------------------------------------*/
//* Run zWPSInstall.sh for primary node                 */
//*-----------------------------------------------------*/
//*      1         2         3         4        5        6         7
//ZSMPEIN  EXEC  PGM=IKJEFT01,TIME=1440
//SYSTSPRT DD  SYSOUT=*
//BPXOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//SYSTSIN  DD  *
 BPXBATCH  SH +
 cd  /wasv7config/b7cell/b7nodea/AppServer/bin; +
 export PATH=.:$PATH; +
 /wasv7config/b7cell/b7nodea/wpssmpe/zos.config/bin/+
 zWPSInstall.sh +
 '-smproot /wasv7config/b7cell/b7nodea/wpssmpe' +
 '-runtime /wasv7config/b7cell/b7nodea/AppServer' +
 '-install'
/*
```

The output from this job ends with the lines shown in Example 3-11, if successful.

Example 3-11 Output from B7NAINS

```
CWPIZ0253I: parsing command arguments...
CWPIZ0254I: parsing arguments complete
CWPIZ0255I: setting up configuration...
CWPIZ0256I: set up configuration complete
CWPIZ0257I: creating the symbolic links...
CWPIZ0259I: creation of symbolic links complete
CWPIZ0260I: doing post install file updates...
CWPIZ0262I: post install updates complete
CWPIZ0263I: running Configuration Manager update...
CWPIZ0264I: Configuration Manager update complete
```

```
OSGi cache successfully cleaned for
/wasv7config/b7cell/b7nodea/AppServer/profiles/default.
```

3.6.2 Running zWPSConfig.sh for the primary node

The zWPSConfig.sh augments the node configuration with WebSphere Process Server for z/OS. To run zWpsConfig.sh for the primary node:

1. Before running zWPSConfig.sh, back up the primary empty node's configuration, using the sample back up job B7BAKNA that is provided on the JCL_P worksheet.
2. Copy sample JCL xxNAAUG, from the JCL_P worksheet of the planning spreadsheet, to the target data set .CNTL. We ran the zWPSConfig.sh script by submitting job B7NAAUG, as in Example 3-12.

> **Note:** Make sure you copy the JCL to run zWPSConfig.sh against the primary node and not the JCL for the stand-alone server's node.

Example 3-12 B7NAAUG JCL to run zWPSConfig.sh for the primary empty node

```
//B7NAAUG JOB (0),'WPS SETUP',CLASS=A,REGION=0M,
// MSGCLASS=H,NOTIFY=&SYSUID,
// USER=B7ADMIN
//*
/*JOBPARM SYSAFF=SC42
//*-----------------------------------------------------*/
//*  Primary empty node.                                */
//*  Run zWPSConfig.sh using the response file created  */
//*  by WCT in WASCFG.B7EMPTA.DATA(BPZRSPN) as input.   */
//*-----------------------------------------------------*/
//*
//*-----------------------------------------------------*/
//*  Copy response file to the UNIX file system (/tmp)  */
//*-----------------------------------------------------*/
//COPY2USS  EXEC  PGM=IKJEFT1A
//SYSEXEC   DD DISP=SHR,
//          DSN=WASCFG.B7EMPTA.DATA
//SYSTSPRT  DD SYSOUT=*
//BPXOUT    DD SYSOUT=*
//SYSTSIN   DD *
 BBOWBOWN WASCFG.B7EMPTA.DATA(BPZRSPM) +
 /tmp/bpzrspn_b7nodea_40279  B7ACRU B7CFG
/*
//*-----------------------------------------------------*/
//*  Copy response file to joblog                       */
//*-----------------------------------------------------*/
//COPYRSP   EXEC PGM=IKJEFT01,REGION=0M,DYNAMNBR=300
//SYSTSPRT  DD SYSOUT=*
//BPXOUT    DD SYSOUT=*
//RSPOUT    DD PATH='/tmp/bpzrspn_b7nodea_40279'
//STDOUTL   DD SYSOUT=*,DCB=(RECFM=VB,LRECL=255,BLKSIZE=0)
//SYSPRINT  DD SYSOUT=*
//SYSTSIN   DD *
 OCOPY INDD(RSPOUT) OUTDD(STDOUTL)
/*
```

```
//*-----------------------------------------------------*/
//* Run zWPSConfig.sh for node B7NODEA                  */
//*-----------------------------------------------------*/
//*
//ZWPSCFG  EXEC  PGM=IKJEFT01,TIME=1440
//SYSTSPRT DD  SYSOUT=*
//BPXOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//SYSTSIN  DD  *
 BPXBATCH  SH +
 cd  /wasv7config/b7cell/b7nodea/AppServer/bin; +
 ./zWPSConfig.sh +
 -response /tmp/bpzrspn_b7nodea_40279 +
 -augment ;
/*
//*-----------------------------------------------------*/
//*  Delete response file from /tmp.                    */
//*-----------------------------------------------------*/
//DELETER1 EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD  SYSOUT=*
//BPXOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//SYSTSIN  DD  *
 BPXBATCH SH +
 rm /tmp/bpzrspn_b7nodea_40279 ;
/*
```

The output from this job ends with the lines shown in Example 3-13, if successful.

Example 3-13 Output from B7NAAUG

```
CWPIZ0253I: parsing command arguments...
CWPIZ0254I: parsing arguments complete
CWPIZ0255I: setting up configuration...
CWPIZ0256I: set up configuration complete
CWPIZ0265I: augmenting profile(s)...
INSTCONFSUCCESS: Profile augmentation succeeded.
CWPIZ0267I: augmenting profile(s) complete
```

Take a backup of all the cell's configuration datasets prior to federating the primary node into the cell. You can use the sample job B7BAKZFS that is provided on the JCL_P worksheet to backup all the configuration datasets.

3.7 Adding WebSphere Process Server to the secondary empty node

With the deployment manager augmented, you must augment the secondary empty node with WebSphere Process Server for z/OS.

3.7.1 Running zSMPInstall.sh for the secondary node

To run zSMPInstall.sh for the secondary node, copy xxNBINS from the JCL_S worksheet of the planning spreadsheet to the target data set .CNTL, as shown in Example 3-14.

Example 3-14 B7NBINS JCL to run zSMPInstall.sh for the secondary empty node

```
//B7NBINS JOB (0),'WPSV7 SETUP',CLASS=A,REGION=0M,
// MSGCLASS=H,NOTIFY=&SYSUID,
//*
/*JOBPARM SYSAFF=SC53
//*----------------------------------------------------*/
//* Run zWPSInstall.sh for secondary node                */
//*----------------------------------------------------*/
//*      1         2         3         4        5        6         7
//ZSMPEIN  EXEC  PGM=IKJEFT01,TIME=1440
//SYSTSPRT DD  SYSOUT=*
//BPXOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//SYSTSIN  DD  *
 BPXBATCH  SH +
 cd  /wasv7config/b7cell/b7nodeb/AppServer/bin; +
 export PATH=.:$PATH; +
 /wasv7config/b7cell/b7nodeb/wpssmpe/zos.config/bin/+
 zWPSInstall.sh +
 '-smproot /wasv7config/b7cell/b7nodeb/wpssmpe' +
 '-runtime /wasv7config/b7cell/b7nodeb/AppServer' +
 '-install'
/*
```

The output from this job ends with the lines shown in Example 3-15, if successful.

Example 3-15 Output from B7NBINS

```
CWPIZ0253I: parsing command arguments...
CWPIZ0254I: parsing arguments complete
CWPIZ0255I: setting up configuration...
CWPIZ0256I: set up configuration complete
CWPIZ0257I: creating the symbolic links...
CWPIZ0259I: creation of symbolic links complete
CWPIZ0260I: doing post install file updates...
CWPIZ0262I: post install updates complete
CWPIZ0263I: running Configuration Manager update...
CWPIZ0264I: Configuration Manager update complete
OSGi cache successfully cleaned for
/wasv7config/b7cell/b7nodeb/AppServer/profiles/default.
```

3.7.2 Running zWPSConfig.sh for the secondary node

The zWPSConfig.sh augments the node configuration with WebSphere Process Server for z/OS:

1. Before running zWPSConfig.sh, back up the secondary empty node’s configuration using the sample back up job B7BAKNB that is provided on the JCL_S worksheet.

2. Copy xxNBAUG, from the JCL_S worksheet of the planning spreadsheet to the target data set .CNTL. We ran the script by submitting job B7NBAUG, as shown in Example 3-16.

Example 3-16 B7NBAUG JCL to run zWPSConfig.sh for the secondary empty node

```
//B7NBAUG JOB (0),'WPS SETUP',CLASS=A,REGION=0M,
// MSGCLASS=H,NOTIFY=&SYSUID,
// USER=B7ADMIN
//*
/*JOBPARM SYSAFF=SC53
//*-----------------------------------------------------*/
//*  Primary empty node.                                */
//*  Run zWPSConfig.sh using the response file created  */
//*  by WCT in WASCFG.B7EMPTB.DATA(BPZRSPN) as input.   */
//*-----------------------------------------------------*/
//*
//*-----------------------------------------------------*/
//*  Copy response file to the UNIX file system (/tmp)  */
//*-----------------------------------------------------*/
//COPY2USS  EXEC  PGM=IKJEFT1A
//SYSEXEC   DD DISP=SHR,
//          DSN=WASCFG.B7EMPTB.DATA
//SYSTSPRT  DD SYSOUT=*
//BPXOUT    DD SYSOUT=*
//SYSTSIN   DD *
 BBOWBOWN WASCFG.B7EMPTB.DATA(BPZRSPM) +
 /tmp/bpzrspn_b7nodeb_40279  B7ACRU B7CFG
/*
//*-----------------------------------------------------*/
//*  Copy response file to joblog                       */
//*-----------------------------------------------------*/
//COPYRSP   EXEC PGM=IKJEFT01,REGION=0M,DYNAMNBR=300
//SYSTSPRT  DD SYSOUT=*
//BPXOUT    DD SYSOUT=*
//RSPOUT    DD PATH='/tmp/bpzrspn_b7nodeb_40279'
//STDOUTL   DD SYSOUT=*,DCB=(RECFM=VB,LRECL=255,BLKSIZE=0)
//SYSPRINT  DD SYSOUT=*
//SYSTSIN   DD *
 OCOPY INDD(RSPOUT) OUTDD(STDOUTL)
/*
//*-----------------------------------------------------*/
//* Run zWPSConfig.sh for node B7NODEB                  */
//*-----------------------------------------------------*/
//*
//ZWPSCFG  EXEC  PGM=IKJEFT01,TIME=1440
//SYSTSPRT DD  SYSOUT=*
//BPXOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//SYSTSIN  DD  *
 BPXBATCH  SH +
 cd  /wasv7config/b7cell/b7nodeb/AppServer/bin; +
 ./zWPSConfig.sh +
 -response /tmp/bpzrspn_b7nodeb_40279 +
 -augment ;
/*
```

```
//*-----------------------------------------------------*/
//*  Delete response file from /tmp.                    */
//*-----------------------------------------------------*/
//DELETER1 EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD  SYSOUT=*
//BPXOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//SYSTSIN  DD  *
 BPXBATCH SH +
 rm /tmp/bpzrspn_b7nodeb_40279 ;
/*
```

The output from this job ends with the lines shown in Example 3-17, if successful.

Example 3-17 Output from B7NBAUG

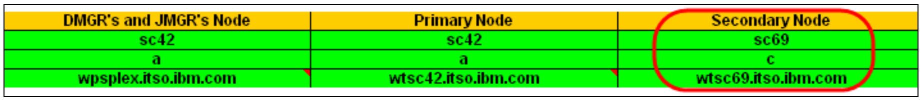
```
CWPIZ0253I: parsing command arguments...
CWPIZ0254I: parsing arguments complete
CWPIZ0255I: setting up configuration...
CWPIZ0256I: set up configuration complete
CWPIZ0265I: augmenting profile(s)...
INSTCONFSUCCESS: Profile augmentation succeeded.
CWPIZ0267I: augmenting profile(s) complete
```

Take a backup of all the cell's configuration datasets prior to federating the secondary node into the cell. You can use the sample job B7BAKZFS that is provided on the JCL_P worksheet to backup all of the configuration datasets.

3.7.3 Configuring any additional secondary nodes

In this book we describe a cluster comprised of two nodes that we refer to as the primary and secondary nodes. If you want to create a cluster of more than two nodes, you must repeat all of the steps that relate to the secondary node for the third LPAR.

The planning spreadsheet helps you in this because you simply return to the Variables worksheet of the spreadsheet, and set the variables for the secondary node so that they apply to the third LPAR, for example, our B7 cell is comprised of two nodes with the primary node on LPAR SC42 and a secondary node on LPAR SC53. If we wanted to configure a third node on LPAR SC69, we would have completed the configuration of the secondary node on SC53 and then set the Variables worksheet, as shown in Figure 3-12.



DMGR's and JMGR's Node	Primary Node	Secondary Node
sc42	sc42	sc69
a	a	c
wpsplex.itso.ibm.com	wtsc42.itso.ibm.com	wtsc69.itso.ibm.com

Figure 3-12 Setting variables for a second, secondary node

All worksheets with a name that ends in _S will now be customized for the third node. The instructions in the Checklist that relate to a secondary node will now be customized for the third node 'C'. Therefore you just follow the same instructions, and use the same worksheets from the spreadsheet that you used for the first secondary node.

3.8 Federating the nodes into the cell

Although the normal next step is to federate the augmented nodes into the cell, you might prefer to delay federating the nodes until after you have a chance to create the database(s) and execute the DDL for WebSphere Process Server for z/OS in DB2 for z/OS. This is because the deployment manager must access the WebSphere Process Server for z/OS tables when it starts up, and if those tables do not exist, it generates a large number of errors in the deployment manager's logs. The Integrated Solutions Console can still be accessed, but the volume of errors in the logs makes it very hard to know whether the deployment manager is configured correctly.

If you decide to delay federation of the nodes until after the tables are set up in DB2, jump forward to Chapter 4, "Configuring the database resources" on page 99, and then return to this point to federate the nodes after the tables are created.

3.8.1 Federating the primary empty node into the cell

To federate the primary empty node into the cell:

1. Start the deployment manager, and login to the Integrated Solutions Console.
2. If the cell is *not* using RACF to manage the cell's SSL keyrings (which happens if the cell was configured using WebSphere Family security or no security), run sample job B7SIGNA from the JCL_P worksheet. This job imports the deployment manager's CA certificate into the trust store that the un-federated node uses. When using a SAF/RACF security manager and RACF keyrings, it is not necessary to run B7SIGNA.
3. Return to the CNTL data set that contains the jobs for configuring the primary empty node. Locate the BBOWMNAN job, and edit it.
4. When RACF is managing the cell's SSL keyrings, add the user ID of the WebSphere administrator user to the job card. If you did not run the B7RAC700 job to set up RACF profiles, you probably do not have a RACF SURROGAT profile for the WebSphere administrator user ID. When there is no RACF SURROGAT profile for the WebSphere administrator user ID to which the installer's user ID is permitted, you must also include PASSWORD= on the JOB card. We ran the B7RAC700 job, which created a SURROGAT profile and permitted our installer user ID to it. Therefore the job card, when we federated our primary node into the B7CELL,specified on USER=, as shown in Example 3-18.

Example 3-18 Job card

```
//BBOWMNAN  JOB 1,'WPSv7 Config',CLASS=A,REGION=0M,MSGCLASS=H,
//  USER=B7ADMIN,NOTIFY=SENIOKJ
//*
/*JOBPARM SYSAFF=SC42
```

Note: If you do not run the job under the WebSphere administrator user ID, and security is enabled in the cell, the BBOWMNAN job will fail with an error that is related to SSL because the user ID that is running the job must have the deployment manager's CA certificate in its keyring for an SSL handshake with the deployment manager to succeed.

The WebSphere administrator user ID has such a keyring; however, a normal TSO user ID will not unless you manually create a keyring for your TSO user ID and connect the cell's CA certificate to the keyring. Running BBOWMNAN under the WebSphere administrator user ID avoids any problems.

Use a /*JOBPARM SYSAFF card to ensure that the BBOWMNAN job runs on the LPAR, which is the home of the primary node that you are federating. If you chose the option to start the node agent after federation, the BBOWMNAN job will issue a start command for the node agent. BBOWMNAN must run on the LPAR where the primary node's node agent is started.

5. Submit BBOWMNAN. If you chose to start the node agent during federation, the node agent is started by job BBOWMNAN just before it ends. If you did not choose to start the node agent during federation, start it up manually using the following command:

   ```
   START B7ACRA,JOBNAME=B7AGNTA,ENV=B7CELL.B7NODEA.B7AGNTA
   ```

After job BBOWMNAN ends normally, back up the configuration ZFS datasets of the deployment manager and the newly federated node's configuration using the sample job B7BAKZFS that is provided in the JCL_P worksheet and shown in Example 3-20 on page 97.

3.8.2 Federating the secondary empty node into the cell

To federate the secondary empty node into the cell:

1. Ensure that the deployment manager is started.
2. If the cell is not using RACF to manage the cell's SSL keyrings, (which will happen if the cell was configured using WebSphere Family security or no security), run the sample job B7SIGNB from the JCL_S worksheet, which imports the deployment manager's CA certificate into the trust store used by the unfederated node. When using a SAF/RACF security manager and RACF keyrings it is not necessary to run B7SIGNB.
3. Return to the CNTL data set that contains the jobs for configuring the secondary empty node. Locate the BBOWMNAN job, and edit it.
4. When RACF is managing the cell's SSL keyrings, add the user ID of the WebSphere administrator user to the job card. If you did not run the B7RAC700 job to set up RACF profiles, you probably do not have a RACF SURROGAT profile for the WebSphere administrator user ID. When there is no RACF SURROGAT profile for the WebSphere administrator user ID to which the installer's user ID is permitted, you must also include PASSWORD= on the JOB card. We ran the B7RAC700 job, which created a SURROGAT profile and permitted our installer user ID to the job card when we federated the secondary node into the B7CELL only specified USER=, as shown in Example 3-19.

Example 3-19 Job card

```
//BBOWMNAN  JOB 1,'WPSv7 Config',CLASS=A,REGION=0M,MSGCLASS=H,
//  USER=B7ADMIN,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC53
```

5. Submit BBOMNAN. If you chose to start the node agent during federation, the node agent is started by job BBOWMNAN just before it ends. If you did not choose to start the node agent during federation, start it up manually using the following command:

   ```
   START B7ACRB,JOBNAME=B7AGNTB,ENV=B7CELL.B7NODEB.B7AGNTB
   ```

3.8.3 Federating any additional secondary nodes into the cell

Repeat the instructions in 3.8.2, "Federating the secondary empty node into the cell" on page 96 for all additional secondary nodes.

3.8.4 Backing up the configuration

It is important to back up the entire cell's configuration at this point. You can use the sample JCL provided in job B7BAKZFS in the JCL_P worksheet, as shown in Example 3-20, to back up the cell's configuration zfses.

Example 3-20 Sample JCL to back up the entire cell's configuration

```
//B7BAKZFS   JOB (0),'WPS SETUP',CLASS=A,REGION=0M,
// MSGCLASS=H,NOTIFY=&SYSUID
/*JOBPARM    SYSAFF=SC42
//*------------------------------------------------------*/
//* Take a backup of the cell's config ZFS datasets      */
//*                                                      */
//* Choose a dump dataset name that describes where you */
//* you are in the configuration process.               */
//* Add VOL=SER= to the DUMP DD if not using DFSMS.     */
//*------------------------------------------------------*/
//*
//DFDSS    EXEC  PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//DUMP     DD DSN=WASCFG.B7CELL.DUMP.PREFED,
//            UNIT=SYSDA,
//            SPACE=(CYL,(2000,100),RLSE),DISP=(,CATLG)
//SYSIN    DD  *
 DUMP OUTDDNAME(DUMP) -
      DATASET(INCLUDE( -
                      WASCFG.B7CELL.B7DMNODE.CONFIG.ZFS -
                      WASCFG.B7CELL.B7NODEA.CONFIG.ZFS -
                      WASCFG.B7CELL.B7NODEB.CONFIG.ZFS -
                     ) -
              ) -
 TOL(ENQF)  OPTIMIZE(4) COMPRESS WAIT(2,2) CANCELERROR
/*
```

4

Configuring the database resources

In this chapter, we describe the steps to create the necessary DB2 objects that are required for WebSphere Process Server for z/OS. We also and discuss design considerations, helpful tools, and recommended procedures.

In this chapter, we discuss:

- 4.2, “Creating the SQL statements” on page 103
- 4.3, “Executing the SQL statements to create DB2 objects” on page 111
- 4.4, “Granting privileges to users in DB2” on page 117
- 4.5, “Verifying the DB2 table creation” on page 121
- 4.6, “Problem determination” on page 122
- 4.7, “DB2 terminology” on page 126

4.1 Introduction to configuring database resources

The BPM suite of components use database tables to store configuration and activity data. Configuring these data sources (resource definitions in WebSphere Application Server) and DB2 objects (tables) is automated by several scripts and samples, which we discuss in this chapter.

To configure the database objects for WebSphere Process Server for z/OS:

1. Generate the DB2 SQL statements with the `DbDesignGenerator.sh` and `createDB.sh` scripts, as described in 4.2, “Creating the SQL statements” on page 103.
2. Validate and customize these DB2 statements for your configuration, as described in 4.2.3, “Validating and customizing the generated SQL statements” on page 111.
3. Execute the SQL statements to create the required DB2 objects, as described in 4.3, “Executing the SQL statements to create DB2 objects” on page 111.
4. Issue the necessary GRANT statements to allow users and groups access to the DB2 tables, as described on 4.4, “Granting privileges to users in DB2” on page 117.
5. Validate the successful creation of tables, as described in 4.5, “Verifying the DB2 table creation” on page 121.
6. Use the trouble-shooting tips that we discuss in 4.6, “Problem determination” on page 122.

In this chapter, we describe these steps in detail. We also provide a brief section at the end of this chapter about DB2 terminology (4.7, “DB2 terminology” on page 126).

4.1.1 Reviewing the DB2 plans

Review the planning and preparation in 2.6, “Database planning and preparation” on page 54, where you decide:

- Single or multiple databases
- DB2 subsystem information:
  - WLM environment name to run User Defined Functions (UDFs)
  - DB2 location name
  - DB2 Universal JDBC driver path
- Names for DB2 objects:
  - Databases and Storage groups
  - VCATs and DASD Volume serial numbers
  - schema prefixes (also called schema qualifiers)
  - bufferpools
  - authentication alias IDs
  - BSpace table space name prefix
- Use of Schema qualifiers and SQLIDs
- Required GRANT privileges

4.1.2 Understanding the DB2 configuration tools

In prior releases of WebSphere Process Server for z/OS, such as Version 6, the createDB.sh script was introduced to simplify the tasks of creating DB2 objects (databases, table spaces, tables, and so on), which was an improvement over the DBUtility.sh script, but was still cumbersome.

Scripts provided for DB2 configuration

Several scripts are provided with WebSphere Process Server for z/OS to help automate the process of configuring your database resources. Each of them are described in the Information Center and in the following sections.

Changes in WebSphere Process Server for z/OS Version 7

If you are familiar with configuring DB2 objects in prior versions, several changes are introduced in Version 7:

- Database design documents (dbDesign), and the document definition tool (DDT): The **`DbDesignGenerator.sh`** script is introduced in Version 7. The dbDesign is required to create the SQL statements that the createDB.sh script uses.
- **`createDB.sh`** now uses the SQL statements generated by DbDesignGenerator.sh to generate and optionally execute the SQL statements. It can now be run in a batch job.
- **`Ddl2Pds.sh`** is an optional tool to convert the SQL statements to EBCDIC and copy them to MVS PDS members, so you can execute them with DSNTEP2 batch job or SPUFI.

zWPSInstall.sh

zWPSInstall.sh creates the symbolic links to the DbDesignGenerator.sh tool and the dbDesign document templates in the <server_root>/util/dbUtils/ directory. See 3.5.1, “Running zSMPInstall.sh for the deployment manager” on page 85.

zWPSConfig.sh

zWPSConfig.sh creates directory structures under <profile_root>/dbscripts/ directory, which are required by createDB.sh. See 3.5.3, “Running zWPSConfig.sh for the deployment manager” on page 87. For a standalone server, this can be used to generate and execute the SQL statements.

DbDesignGenerator.sh

This new tool creates a database design document, which is key to the new process in WebSphere Process Server for z/OS Version 7. This tool also creates SQL statements that the createDB.sh script uses for all components except the Common Even Infrastructure (CEI). The SQL for CEI is generated by the createDB.sh script.

See 4.2.1, “Creating the database design (dbDesign) document” on page 104 for more details.

createDB.sh

With Version 7, you must first run `DbDesignGenerator.sh` to create the SQL statements in the default directory. Next, you tailor the sample `createDB.sh` script to match your configuration. The tool generates customized SQL statements for your database configuration. It can also execute the SQL statements directly as an option to create the DB2 objects.

> **Note:** With WebSphere Process Server for z/OS Version 7.0.0.1, createDB.sh is required to generate the SQL statements for CEI. The DbDesignGenerator.sh script does not generate SQL for CEI tables.

See 4.2.2, “Generating the SQL with createDB.sh” on page 106.

Ddl2Pds.sh

Ddl2Pds.sh is a new tool that helps you copy the SQL statements to an MVS partitioned dataset (PDS) and converts to the EBCDIC code-page. This tool can be run by an individual component name (for instance, WPS or BPC), by directory or by a specific file name.

Using this tool you can easily run the SQL in a z/OS environment with SPUFI or DSNTEP2 in a batch job.

See 4.3.3, “Copying the SQL statements to partition datasets (PDSes)” on page 113 for more details.

4.1.3 Overview of the database configuration process

There are several steps and scenarios to consider when deciding how to configure your database objects for WebSphere Process Server for z/OS.

The phases are:

1. Create the sample SQL statements.
2. Tailor the SQL statements for your installation (optional).
3. Execute the SQL statements to create the DB2 objects.

When can you create these tables

You create the tables after the `zWPSInstall.sh` and `zWPSConfig.sh` scripts are run to augment the nodes for WebSphere Process Server for z/OS.

The `zWPSInstall.sh` script creates the directories and symbolic links that are used to run these database utilities in: `<server_root>/util/dbUtils/`

The `zWPSConfig.sh` script creates the symbolic links in the <profile_root>/dbscripts/ directory, pointing to the sql templates in `<smpe_root>/dbscripts/.` These are required by the `createDB.sh` script.

Planning your DB2 configuration activities

After creating and augmenting the WebSphere Application Server for z/OS deployment manager, you can start working work on the database objects required by WebSphere Process Server for z/OS:

Suggested database configuration steps

Figure 4-1 on page 103 shows the interaction of these tools:

- **`zWPSInstall.sh`** script creates the `<server_root>/util/dbUtils/` directory and link to the DbDesignGenerator.sh script.
- **`zWPSConfig.sh`** script creates the directory structures required by `createDB.sh`.
- **`DbDesignGenerator.sh`** reads a dbDesign document, and creates SQL statements in the default directory (currently `<server_root>/util/dbUtils/WBI_*`).
- At this point, you can review and tailor the SQL statements using your favorite editor.
- **`createDb.sh`** generates SQL statements, and optionally executes them to create the DB2 objects. createDB.sh is also required to generate the SQL for the CEI tables.

If you do not use createDB.sh to execute the SQL statements, there are two more steps:

- **`Ddl2Pds.sh`** copies the SQL statements to a MVS-partitioned dataset (PDS), which transforms them to the proper code-page (IBM-1047).
- **`DSNTEP2`** or **`SPUFI`** executes (run) the SQL statements to create the DB2 objects.

Figure 4-1 on page 103 shows an overview of the DB2 configuration process. The Information Center describes these programs, 4.2, “Creating the SQL statements” on page 103 has more guidance and recommendations.

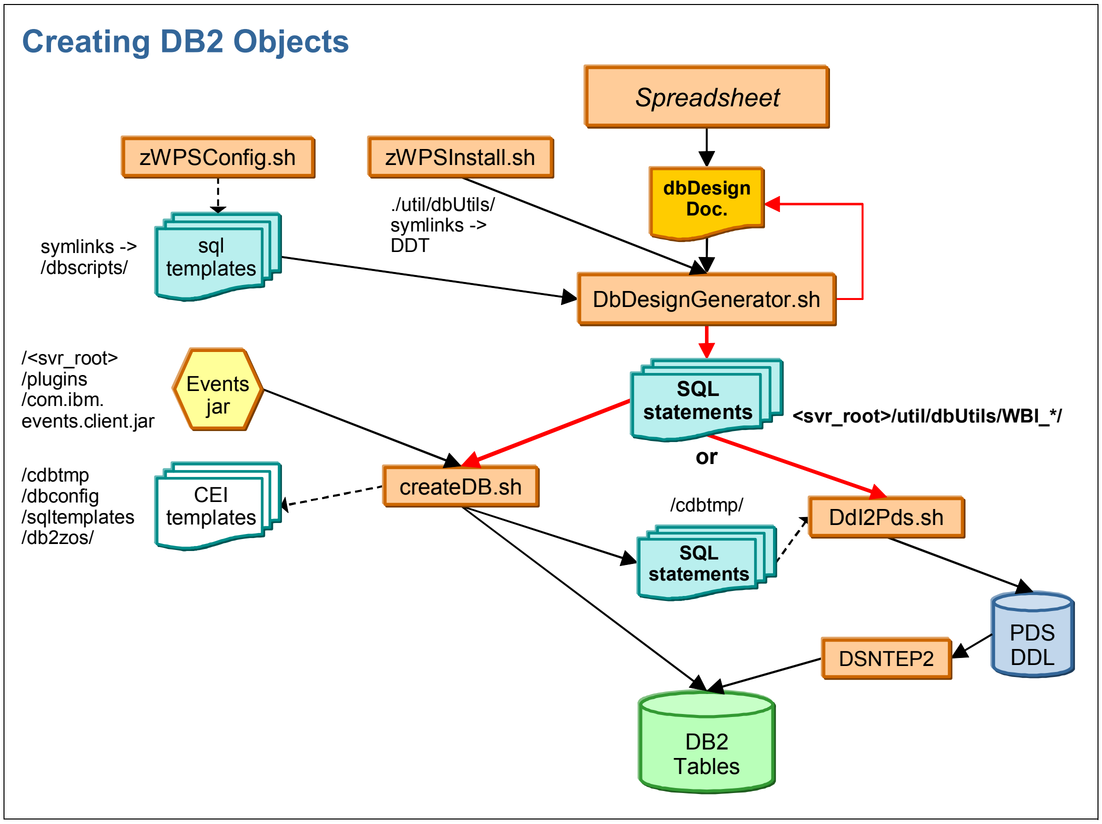


Figure 4-1 Using the database configuration tools

CEI is an exception. Use `createDB.sh` to generate the SQL instead of `DbDesignGenerator.sh`, and then use `createDB.sh` to execute it or copy it with `Ddl2Pds.sh`.

See 4.2.2, “Generating the SQL with createDB.sh” on page 106 for more details.

4.2 Creating the SQL statements

Here is a step-by-step description of the process we used to create the SQL statements:

1. Complete the spreadsheet with variable values for z/OS, DB2, WebSphere Application Server and WebSphere Process Server.
2. Copy the dbDesign document from the tab that is marked dbDesign.
3. Copy and paste the dbDesign file into a working directory in the UNIX file system, making certain that it is in ascii code.
4. Launch the database design Tool (DDT) by running the `DbDesignGenerator.sh` script in interactive mode with no parameters.
5. Complete the prompts, and save the dbDesign document and the generated SQL statements in the default directories: **`<WAS_HOME>/util/dbUtils/WBI_`**
6. Use the createDB.sh script to generate the SQL statements for the Common Event Infrastructure (CEI) component. The DDT does not currently support CEI.

After the SQL statements are generated, you can verify them, make any modifications that you want, and execute them to create the DB2 objects. See 4.3, “Executing the SQL statements to create DB2 objects” on page 111 for a description of those steps.

Looking at alternative methods

Alternative methods for creating the SQL statements are:

- For a standalone server, the `zWPSConfig.sh` script can generate and execute the SQL as it did in prior releases. We did not use it, and we do not recommend it in a production topology.
- The `DBUtility.sh` script is used by the createDB.sh script, but is more cumbersome to use directly.
- The `sibDDLGenerator.sh` script is available for generating the SQL statements for the messaging engines' databases, but we felt is was easier to use the `DbDesignGenerator.sh`, and `createDB.sh` scripts that use the `sibDDLGenerator.sh` script under the covers.
- The Deployment Environment wizard can also generate SQL (ddl) statements as part of creating a cluster. However, these database definitions are not complete or as easy to use. You must use one of the previously mentioned scripts for the messaging engine's databases.

4.2.1 Creating the database design (dbDesign) document

After your planning and DB2 preparation, the first step in creating the database objects for WebSphere Process Server for z/OS is to create a database design pattern that the various configuration scripts can use.

You start with the dbDesign document that the spreadsheet produced, or build it using the DbDesignGenerator.sh script in interactive mode:

- If you have a dbDesign document, proofread it for accuracy and applicability to your z/OS and DB2 environment, and the naming for databases, storage groups, and so forth.
- If not, use the DbDesignGenerator.sh script in interactive mode to create one for a deployment environment, and save the design document. See the discussion to step through the prompting in "Running the database design Tool (DDT): DbDesignGenerator.sh" on page 104.

After you are comfortable with the design document, you can chose the option to Generate database scripts from a database design, which you must do regardless of whether you plan to use the createDB.sh script, DSNTEP2 or some other tool to execute the SQL statements.

The **`DbDesignGenerator.sh`** script is created in the **`<WAS_HOME>/util/dbUtils/`** directory by the **`zWPSInstall.sh`** script. There are several optional parameters, such as help (-?), edit (-e), verify (-v), and generate (-g), but we usually run it without any parameters in interactive mode.

Running the database design Tool (DDT): DbDesignGenerator.sh

The **`DbDesignGenerator.sh`** script is accessed from the **`<WAS_HOME>/util/dbUtils/`** directory after the **`zWPSInstall.sh`** script is run. To run the script, change to the **`<WAS_HOME>/util/dbUtils/`** directory, and type the following in a command prompt in a UNIX System Services (telnet client) or OMVS session:

```
DbDesignGenerator.sh
```

Tips for running DbDesignGenerator.sh:

- Use this script for making updates to the dbDesign document instead of a text editor.
- You can start with the dbDesign file from the spreadsheet.
- The dbDesign file must be in ascii and must have .dbDesign as its file type. (Suffix)
- If you do edit the file, beware of “trailing blanks” at the end of each line.
- Watch out for the following parameter that might be in the response file causing the /dbscripts/ directory to be created in a location other than the default (<profile_root>/utils/dbUtils/).

  dbOutputScriptDir=/wasv7config/b7cell/wpswork/dbscripts
- If you use the second option in this list to create for an individual component, the SQL statements get generated into a directory that is different than the default (<profile_root>/utils/dbUtils/).
- If you plan to use separate DB2 databases for the components, you might need to edit the SQL statements that DbDesignGenerator.sh creates because some of them do not have unique database names specified.
- The CEI and SIB SQL statements might not have unique table space names. See the createDB.sh script which does generate unique names.
- You have a choice between Java and SQL functions for BPC Reporting configuration. We selected SQL functions and did not specify **“**`WLM environment name to run UDF.`**”** The warning messages “`WLMID @ line xx .in file createFunctionsJava_Observer.ddl`” are okay if you are not using the Java functions.
- You only need to run one of the ddl/sql files in the target directory (createDB.sh runs the .sql files).

Here is the suggested order to use:

1. Design the [master] component first so that its values can be used as defaults for other components.
2. Before configuring a component that has a parent, design the parent component. In this case, design WBI_SCA_SYS_ME before doing any of WBI_BPC_ME, WBI_CEI_ME, or WBI_SCA_APP_ME.
3. Save the dbDesign document.
4. Generate the SQL statements.

When stepping through the components in interactive mode, pay attention to the *master* and *parent*. Also, pay attention to the Status (complete or not complete).

- DB design for a component can inherit attribute values from a parent component to populate its default values.
- The hierarchy is indicated in the script: At the top of the tree is the Common DB, which is the [master] component. Configure the Common DB, and you will find that the other components inherited values that are set for the Common DB and marked complete.
- Currently the only other components that exhibit parenthood are the buses. If you set up the SYS bus, its values are used for the other buses.
- You might still want to go though the component configurations to (a) check that the default values inherited from Common DB are OK, and (b) change the schema qualifiers because they are not inherited from the Common DB and are hard-coded to various values.

A log file, dbDesignGenerator.log, is written to the <server_root>/util/dbUtils/ directory so that you can review the results of your interactive session.

Sample dbDesignGenerator session

Here is the beginning of a sample session running the ddl and database design generator.

Example 4-1 Sample dbDesignGenerator session

```
cd /wasv7config/b7cell/b7dmnode/DeploymentManager/util/dbUtils

DbDesignGenerator.sh

[info] running DbDesignGenerator in interactive mode...
[info] Enter 'q' to quit without saving; '-' for back to previous menu; '?' for help at any time.
[info] To accept the given default values, simply press the 'Enter' key.
[info] Please pick one of the following [design option(s)] :

(1)Create a database design for Standalone profile or Deployment Environment
(2)Create a database design for a single component (e.g. BPC, CEI etc)
(3)Edit an existing database design
(4)Generate database scripts from a database design
(5)exit [q]

Please enter the number for the design option :3
Please enter the database design file :/u/hutchjm/wps7/db2/B9_Feb4.dbDesign

[info] analyzing the database design ...
[status] wps.nd.topology is complete with 0 remaining item(s):
---------------------------------------------------------------
[info] Please edit any database component with status of 'not complete' for required properties.
[info] Completed database components can be edited to change existing or defaulted property values.
[info] Design the 'master' component first, and then any parent components, since other components
may inherit values from them.

[info] Please pick one of the following [database component(s)] :

(1)[CommonDB]   WBI_CommonDB : [master] [status = not complete]
(2)[BPCReporting]       WBI_BPCEventCollector :  [status = not complete]
(3)[BPC]        WBI_BPC :  [status = not complete]
(4)[BSpace]     WBI_BSPACE :  [status = not complete]
(5)[CEI]        WBI_CEI_EVENT :  [status = not complete]
(6)[SibME]      WBI_SCA_SYS_ME :  [status = not complete]
(7)[SibME]      WBI_BPC_ME : [parent = WBI_SCA_SYS_ME] [status = not complete]
(8)[SibME]      WBI_CEI_ME : [parent = WBI_SCA_SYS_ME] [status = not complete]
(9)[SibME]      WBI_SCA_APP_ME : [parent = WBI_SCA_SYS_ME] [status = not complete]
(10)[save and exit]

Please enter the number for the database component :
```

If you want to review the values in this interactive script, you can step through them by number (1-9) starting with the *master* item.

After the status is complete for all of the components, you can use option 10, to save the document to the same file or to a new location.

4.2.2 Generating the SQL with createDB.sh

The createDB.sh script is the recommended tool to generate the SQL statements, even after the DB2DesignGenerator.sh script generates them because it includes the SQL for CEI and tailors the SIB tablespaces so that they are unique.The createDB.sh script is also the most simple mechanism to execute the SQL statements to create the DB2 objects, but more on that later.

Note: The Deployment Environment wizard can also create DDL for CEI in the following location, but we recommend that you not use it because createDB.sh does a better job of customizing it correctly for DB2 z/OS:

`<profile_root>/databases/event/<cluster_name>/dbscripts/db2zos/ddl/`

Customizing the createDB.sh script

The createDB.sh script is a "sample" that you must customize for your installation in several areas. To customize the createDB.sh script:

1. Copy the sample script createDB.sh from the **`/<wps_smpe>/zos.config/samples/`** directory to your working directory. Also copy the `sibDropxx.sql` and `sibSchemaxx.sql` files from the samples directory into the working directory.

   The next step is to modify the createDB.sh script:

2. Update the copy of createDB.sh script in the "`Edit1`" section with the values that are appropriate for your environment as follows to export these environment variables:

   – Export WAS_HOME=/wasv7config/b9cell/b7dmnode/DeploymentManager
   – Export LIBPATH=/usr/lpp/db2/d9fg/db2910_jdbc/lib:$LIBPATH
   – Export STEPLIB=DB9F9.SDSNEXIT:DB9F9.SDSNLOAD:DB9F9.SDSNLOD2:$STEPLIB

3. Update the "`Edit2`" section with these variables to set the following variables.

   The following variables control what actions are performed by the createDB.sh script. Many of them can be overriden by parameters on the script when it is run:

   – DBRunSQL=true/false →

   – DelayConfig=true/false → The runtime parameter +RunSQL overrides DBDelayConfig and sets it to true.

   – DBCreate=true/false → We recommend setting this variable to false. It wipes out existing databases and storage groups.

   Caution: This is overriden if you specify +All when you run createDB.sh. DBCreate is set to true!

   – ConfirmRun=y/n → asks the interactive user to review the parameter settings before proceeding. Set to "n" if running in a batch job.

4. You can modify the remaining working variables in the script. Most of these can be overriden with parameters on the command that are used to run the script.

   These variables affect parameters on the generated SQL statements:

   DBPrefix Usually set to the cell prefix B7

   DBScope Set to the Cluster (C) or Server (S) number - usually C1

   Buffer Pools DBBP4K, DBBP8K, DBBP16K, and DBBPIndex

   The following variables can be overriden by runtime parameters when you invoke the script:

   DBName Name of the DB2 database: Default is {prefix}CELLDB.

   DBSto Storage Group name: Default is {prefix}DBSTO.

   DBSchema Schema prefix: Default is {prefix} CELL.

SCRIPTHOME Working directory for SQL statements, temporary files, and output: Default is the current directory where you launch createDB.sh.

ScriptTemp Directory of the temporary directory: Default is $SCRIPTHOME/cdbtmp.

DBSIBDrop Used when *DBSIB is specified, which means drop and recreate the SIB tables.

Leave this blank or specify the location of SIBDropxx.sql, which causes the SIB tables to be dropped.

The following parameters are not relevant if you use DB2 Type 2 connectors:

- DBHostName
- DBServerPort

The following parameters are only relevant if you use DBCreate=true:

- DBVCat
- DBVolumes

The following parameters are only relevant if you are executing the SQL statements:

- DBJDBCClasspath
- DBUser and DBPassword
- DBConnectionLocation
- DBJDBCProperties: Only the directory path and not the filename itself, which must be DB2JccConfiguration.properties

5. Save the updated copy of the script, and make certain that the permission bits are read and executed by all (755).

Running createDB.sh

After copying the sample createDB.sh script to your working directory and modifying for your environment, you can run it from there to generate, and optionally, execute the SQL.

The createDB.sh runtime parameters

Many of the parameters that you can change in the script can also be specified on the command line when you run the script.

To specify which components are generated, specify "+" or "-" in front of these to include or exclude:

- All for all components
- DBWPS for the Common WBI tables
- DBBPC for Business Process Choreography tables
- DBBPR for BPC Reporter tables
- DBCEI for CEI tables
- DBBS for Business Space tables
- DBSIB for Messaging Engine tables

Notes for DBSIB:

- DBSIB only applies to the SYS and APP ME tables.
- To get BPC ME tables, +DBBPC must also be specified.
- To get CEI ME tables, +DBCEI must also be specified.
- Specify "*" instead of "+" in front of DBSIB to drop the ME tables before you create them.

Results are written to ./cdbtmp/output.out and ./cdbtmp/error.out files. Refer to the Information Center and the source of the createDB.sh script for detailed information.

User ID requirements

The user ID requirements are:

- Execute access to your copy of the createDB.sh script.
- Read access to the SQL templates in `$WAS_HOME/util/dbUtils/WBI_*` directories.
- Write access to the working (cdbtmp) directory plus the authority to remove any existing cdbtmp directory.
- If executing the SQL statements (DBDelayConfig=false or RunSQL=true), you need DB2 authority to create the table spaces, tables, and so on.
- If you specify +DBCreate and -UseSqlid (both defaults), providing the user can create the database, they need no further authorizations.
- With other database owners or use of SET CURRENT SQLID, further authorizations are required.

Input

The inputs are:

- SQL templates are read from the $WAS_HOME/util/dbUtils/WBI_** directories where they were created by the DbDesigngenerator.sh script.
- If you need to drop the tables, the `sibDropxx.sql` file must also be copied into your <working_directory> from `/<wps_smpe>/zos.config/samples/`.

Output

The output is:

- The `output.out` file is written to ./cdbtmp/ directory (it is actually copied from `$WAS_HOME/logs/manageprofiles/default/commonDBUtility.ant.log`).
- The `error.out` file is also written to ./cdbtmp/ directory.
- The resultant SQL statements are written to the .sql files (depending on which components are selected) and are written to the same `./cdbtmp/` directory:
  - BSpace.sql
  - bpc.sql
  - bpcr.sql
  - ceidb.sql
  - common.sql
  - sibAPP.sql
  - sibBPC.sql
  - sibCEI.sql
  - sibSCA.sql
- The files ending in .ddl are temporary files:
  - ceidb.ddl
  - sibAPP.ddl
  - sibBPC.ddl
  - sibCEI.ddl
  - sibSCA.ddl

Sample createDB.sh commands

Here are a few of the common examples used to create tables with createDB.sh:

- `createDB.sh`: Uses the default values in the customized createDB.sh script.
- `createDB.sh +All`: Creates SQL for all components, which forces DBCreate=true unless you also specify **`-DBCreate`** on the command line.
- `createDB.sh *DBSIB`: Creates tables for SCA System and Application MEs.
- `createDB.sh -All +DBCEI`: Creates tables for CEI database.
- `createDB.sh -All +DBBPC +DBCEI *DBSIB`: Creates tables for all MEs.

Tips for running createDB.sh:

- SQL statements (except for the CEI database) are created by the DbDesignGenerator.sh script in **`<server_root>/util/dbUtils/WBI_*`** directories.
- For CEI, the sql templates are extracted from the **`<server_root>/plugins/com.ibm.events.client.jar`** file and tailored with your database name, schema, and other settings.

  (If you need to use an alternative source, modify the createDB.sh at the eight places where the change directory (cd) statements are issued to: **`$WAS_HOME/util/dbUtils/WBI_*`**
- For the CEI and SIB SQL, this script also tailors the statements to ensure that table space names are unique.
- The table space names are edited by combining a five-character schema qualifier, which is unique to each bus, with an abbreviated version of the table space name.
- RunSQL overrides DBDelayConfig.
- If you specify **DBCreate=false** inside the script, it is overridden by the **`+All`** option, which generates DROP and CREATE statements for the Storage Group and databases unless you also specify **`-DBCreate`** on the command line:
  - `+All` forces `DBCreate=true`
  - `-All` forces `DBCreate=false`

Running createDB.sh in a batch job

In WebSphere Process Server for z/OS Version 7, the createDB.sh script does not use the ‘vi’ editor as it did in Version 6; instead, it uses ‘sed’ to edit the SQL files, which means that it does not need to be run in a telnet client.

Specify the following parameter (around line number 50) in the createDB.sh script:

```
ConfirmRun=n
```

Run the batch job, as shown in Example 4-2, with a user ID that has the same authorities and permissions as when you run the script from the command line.

The output cdbtmp directory is created in the home directory of the user ID that is running the createDB.sh script command.

Example 4-2 Running createDB.sh in a batch job

```
/*JOBPARM S=SC42
//CREDB EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
```

```
//STDERR   DD  SYSOUT=*
//SYSTSIN  DD  *
 BPXBATCH sh  /u/hutchjm/wps7/db2/Batch_B7_createDB.sh +All +RunSQL
/*
```

For troubleshooting tips, see "Debugging createDB.sh" on page 124.

4.2.3 Validating and customizing the generated SQL statements

After running createDB.sh, the generated SQL statements are written to the ./cdbtmp/ directory. Review the SQL statements to verify that they are what you expect.

You might also want to change some of the parameters that are specified on the SQL statements, such as:

- PRIQTY and SECQTY
- FREEPAGE and PCTFREE
- BufferPools
- Database, storage group names, and schema prefix names

4.3 Executing the SQL statements to create DB2 objects

After the sample SQL statements are generated and tailored, you can execute or run them to create the DB2 objects: Table spaces, Tables, Indexes, and other objects defined in the SQL.

> **Note:** Skip this section if you already ran the createDB.sh script using the DBDelayConfig=false or RunSQL=true options.

There are several ways to do execute the SQL statements to create DB2 objects. Here are the two most common ways:

- Execute the SQL statements with the createDB.sh script. This is probably the easiest. See 4.3.2, "Executing the SQL statements with the createDB script" on page 112.
- Transfer the SQL statements to an MVS dataset and use a batch job with DSNTEP2 or with SPUFI. See 4.3.4, "Executing the SQL with DSNTEP2 (batch job)" on page 116.

Before you begin

The SQL statements must be generated and modified for your installation and saved in a location in order to be run (or "executed"), which can be in UNIX System Services files in your HFS or in MVS data sets, such as one or more PDSes.

Using a single DB2 database or multiple databases

You should plan whether you use a single or multiple DB2 databases at the beginning. We discussed the advantages and disadvantages in 2.6.1, "Designing the database configuration" on page 54.

Using a single database

Using a single database is the easiest scenario because most of the tools, such as createDB.sh, are designed to only work with one database (and storage group) at a time. If you want to use createDB.sh for multiple databases, you must make separated copies of the script, and modify each copy to use a different database.

Also, the database design tool that is driven by the DbDesignGenerator.sh script does not allow a unique database name for each of the components.

Using multiple databases

If you want to use separate databases for some components, you must make certain adjustments to the processes that we discussed in this chapter.

The createDB.sh script only has one DBName specification around line 107 in the script, which you change for each separate database. Make separate copies of createDB.sh to help manage this.

4.3.1 Creating the databases and storage groups

Usually, database and storage group objects are defined in a separate step by a person with DB2 SYSADM authority, such as the Database administrator (DBA). Example 4-3 uses the DSNTEP2 program in a batch job.

Example 4-3 Creating the DB2 database and storage group

```
//B7CR1DB JOB 1,'CR.DB2 TBLS',CLASS=A,MSGCLASS=H,REGION=0M USER=SYSADM0
/*JOBPARM S=SC42
//***********************************************************/
//JOBLIB  DD DISP=SHR,DSN=DB9F9.SDSNEXIT
//        DD DISP=SHR,DSN=DB9F9.SDSNLOAD
//CREATE   EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(D9F1)
 RUN PROGRAM(DSNTEP2) PLAN(DSNTEP91) LIB('DB9FU.RUNLIB.LOAD')
 END
//DBRMLIB  DD DISP=SHR,DSN=DB9F9.SDSNDBRM
//********************************************************************
//* DROP STATEMENTS WILL NOT BE RUN UNLESS //DROP CHANGED TO //SYSIN
//**************************************************************
//DROP DD *
  DROP   DATABASE B7CELLDB ;
  DROP   STOGROUP B7DBSTO;
  COMMIT;
//SYSIN  DD *
CREATE STOGROUP B7DBSTO VOLUMES ('TOTD97') VCAT B7DB2VC;
CREATE DATABASE B7CELLDB STOGROUP B7DBSTO BUFFERPOOL BP0  INDEXBP BP0;
COMMIT;
/*
```

What to do next

After the storage groups and databases are created in DB2, you can execute, or run the SQL statements to define the DB2 objects, such as table spaces, tables, and indexes.

4.3.2 Executing the SQL statements with the createDB script

You can use the createDB.sh script to create the DB2 objects directly or you can copy them to MVS datasets and use batch jobs (or SPUFI) to execute the SQL statements. (See 4.3.3, "Copying the SQL statements to partition datasets (PDSes)" on page 113 and 4.3.4, "Executing the SQL with DSNTEP2 (batch job)" on page 116 for this latter alternative.)

We found using createDB.sh to be the simplest way to execute the SQL if all the tables are going to the same DB2 database. You must run the script with a userID that has SYSADM or DBADM authority to create these DB2 objects.

See Example 4-4 for an example of running createDB.sh in a batch job to create the storage group, database, table spaces, tables, and all other DB2 objects.

Example 4-4 Running createDB.sh in a batch job to create the database and DB2 objects

```
//B7CREADB JOB 1,'CrDB2 Tbls',CLASS=A,MSGCLASS=H,REGION=0M USER=SYSADM
/*JOBPARM S=SC42
//CREDB EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//SYSTSIN  DD  *
 BPXBATCH sh  /u/hutchjm/wps7/db2/BatchB7createDB.sh +All +RunSQL
/*
```

If you set the DBDelayConfig parameter inside the script to "false", or specify the +RunSQL parameter on the command line, the SQL statements are executed to create the DB2 objects. See 4.2.2, "Generating the SQL with createDB.sh" on page 106 for more details about createDB.sh.

4.3.3 Copying the SQL statements to partition datasets (PDSes)

Copying the SQL statements to PDSes is an alternative to using createDB.sh to execute the SQL, which we discussed in 4.3.2, "Executing the SQL statements with the createDB script" on page 112.

The generated SQL statements can be run from their existing location with a script or transferred to MVS datasets so that they can be run by a batch job, or SPUFI.

Use the new Ddl2Pds.sh script, or your existing file transfer utility, to help automate this process. The Ddl2Pds.sh sample script is in the <wps_smpe_root>/zos.config/samples/ directory:

/usr/lpp/zWPS/V7R0/zos.config/samples/

Running the Ddl2Pds. shell script

You can run the Ddl2Pds. shell script after the SQL statements are created by **zWPSConfig.sh, DbDesignGenerator.sh,** or **createDB.sh**.

Use the following required parameters to run Ddl2Pds.sh:

-Source Directory containing the .sql source files.

-PDS The fully-qualified name of the partitioned dataset that you want the SQL statements transferred to. DO NOT SPECIFY MEMBER NAME!

-Component Specify a component ID from the following list, or specify the file name, which must end in .sql or .ddl:

WPS: Common database objects
BPC: Process choreography database objects

BPR: BPC Reporter database objects
CEI: Common Even Infrastructure database objects
BSP: Business space database objects
ALL: All of them
<path>/<.sql or .ddl file name>

The following optional parameters can also be specified:

-Prefix Two character prefix, can be used to separate DDL in the PDS:

Default = AA

-WorkDir Working directory where temporary files and logs get written.

Default = /tmp The files are cleaned up after use.

-ConvertCodePage Specify `y` to convert from ASCII to EBCDIC, `n` (do not translate).

Default is based upon filename suffix (.sql files are translated to EBCDIC).

User ID is required and must have the authority to:

- Read from the Source
- Write to the Working directory
- Write to the PDS, and create it if necessary. DSNAME profile class in RACF.

Tips for running Ddl2Pds.sh:

- If the target partitioned dataset does not exist, the script creates the PDS for you.
- You should run this script one component at a time. Pair the -Component parameter with the -PDS parameter so that the SQL statements for each component can be managed more easily. The ALL option is for special cases, such as grouping multiple components or for debugging.
- The member names in the target PDS have the names xxCCCn where:
  - xx is the cell prefix
  - CCC is the component (BPC, WPS, CEI, BSP or ALL)
  - n is a sequential number assigned starting at 1
- Specify the fully-qualified name of the partitioned dataset that you want the SQL statements transferred to. DO NOT SPECIFY MEMBER NAME!
- The xxREADME members contain a description of each of the members. If you re-run the Ddl2Pds.sh script again to the same PDS, it will be over-laid and might not reflect the contents of the other members.
- Not all of the members that are transferred to the PDSes need to be run. Some are for dropping, defining databases, storage groups, or migration. Pay attention to the xxREADME member to determine which members you want to use.
- Ddl2Pds.sh does no editing of the table space names for the buses; instead, it just copies DDL files over, so you need to manually edit the table space definitions.

For debugging suggestions, see “Debugging Ddl2Pds.sh” on page 124.

You also have the option of specifying a specific file name with the -Component parameter. In this case, the file identified is the only one to be transferred. The command might look something like:

```
Ddl2Pds.sh -Source /u/hutchjm/cdbtmp -Component ceidb.ddl -PDS 'HUTCHJM.DDL.CEI'
```

Running the Ddl2Pds.sh script in a batch job

You can run the Ddl2Pds.sh script in batch, as shown in Example 4-5.

Example 4-5 Batch job to run Ddl2Pds.sh to transfer BPC Reporting SQL files to a PDS

```
//DD22PDS EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//SYSTSIN  DD  *
 bpxbatch sh +
 cd /u/hutchjm/wps7/db2; +
 B9_Ddl2Pds.sh -WorkDir /tmp -Prefix B9  -Component BPR +
 -Source /wasv7config/b9cell/b9dmnode/DeploymentManager/util/dbUtils/+
 WBI_BPCEventCollector_DB2-zOS-9-BPCReporting
/*
```

Note for running Ddl2Pds.sh in a batch job:

Change the lines in the script that ask for operator prompts. Search for the “`read`” statements around line numbers 195, 219, and 300 and comment them out, including the following statements bracketed by:

```
if
....
fi
```

Example 4-6 is a sample job to copy the SQL statements for all components over to an MVS PDS. The last invocation is for the CEI tables, which points to a specific SQL file.

Example 4-6 Sample job to copy SQL statements for the BPC Reporter tables

```
//B7DL2BPR JOB 1,'Copy SQL to PDS',MSGCLASS=O,CLASS=A,REGION=0M
//********************************************************************/
//DD22PDS EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//SYSTSIN  DD  *
 BPXBATCH sh +
 /u/hutchjm/wps7/b7cell/B7_Ddl2Pds.sh  -Prefix SA -Component ALL +
 -Source /wasv7config/b7cell/b7dmnode/DeploymentManager/util/dbUtils/+
 WBI_SCA_APP_ME_DB2-zOS-9-SibME      -PDS HUTCH.B7DDL ;          +
/u/hutchjm/wps7/b7cell/B7_Ddl2Pds.sh   -Prefix SS -Component ALL +
 -Source /wasv7config/b7cell/b7dmnode/DeploymentManager/util/dbUtils/+
 WBI_SCA_SYS_ME_DB2-zOS-9-SibME      -PDS HUTCH.B7DDL ; +
/u/hutchjm/wps7/b7cell/B7_Ddl2Pds.sh   -Prefix SB -Component ALL +
 -Source /wasv7config/b7cell/b7dmnode/DeploymentManager/util/dbUtils/+
 WBI_BPC_ME_DB2-zOS-9-SibME          -PDS HUTCH.B7DDL ; +
/u/hutchjm/wps7/b7cell/B7_Ddl2Pds.sh   -Prefix SC -Component ALL +
 -Source /wasv7config/b7cell/b7dmnode/DeploymentManager/util/dbUtils/+
 WBI_BPC_ME_DB2-zOS-9-SibME          -PDS HUTCH.B7DDL ; +
 /u/hutchjm/wps7/b7cell/B7_Ddl2Pds.sh  -Prefix BS -Component BSP +
 -Source /wasv7config/b7cell/b7dmnode/DeploymentManager/util/dbUtils/+
 WBI_BSPACE_DB2-zOS-9-BSpace         -PDS HUTCH.B7DDL ; +
 /u/hutchjm/wps7/b7cell/B7_Ddl2Pds.sh  -Prefix BR -Component BPR +
 -Source /wasv7config/b7cell/b7dmnode/DeploymentManager/util/dbUtils/+
 WBI_BPCEventCollector_DB2-zOS-9-BPCReporting -PDS HUTCH.B7DDL ; +
 /u/hutchjm/wps7/b7cell/B7_Ddl2Pds.sh  -Prefix WP -Component WPS +
 -Source /wasv7config/b7cell/b7dmnode/DeploymentManager/util/dbUtils/+
 WBI_CommonDB_DB2-zOS-9-CommonDB     -PDS HUTCH.B7DDL ; +
 /u/hutchjm/wps7/b7cell/B7_Ddl2Pds.sh  -Prefix BP -Component BPC +
 -Source /wasv7config/b7cell/b7dmnode/DeploymentManager/util/dbUtils/+
 WBI_BPC_DB2-zOS-9-BPC               -PDS HUTCH.B7DDL ; +
 /u/hutchjm/wps7/b7cell/B7_Ddl2Pds.sh  -Prefix CE -Component ceidb.sql +
 -Source /u/hutch/cdbtmp             -PDS HUTCH.B7DDL
/*
```

4.3.4 Executing the SQL with DSNTEP2 (batch job)

Many DB2 administrators prefer to use batch jobs or SPUFI to run the DSQL statements instead of using a command-line script from the UNIX environment or from their workstation; however, before doing so, the files must be copied to MVS datasets (sequential files or partitioned datasets - PDSes) before running DSNTEP 2 or SPUFI.

> **Note:** Skip this step if you executed the SQL in Section 4.3.2, “Executing the SQL statements with the createDB script” on page 112.

Transferring the SQL files for DSNTEP2

If you prefer to execute the SQL files using the DB2 DSNTEP2 program in a batch job, use the Ddl2Pds.sh script to transfer them.

If you try to copy them from the UNIX file system into z/OS sequential files or as members of a partitioned dataset, be mindful that:

- Because the SQL is in ASCII, it must also be converted to EBCDIC before DSNTEP2 can execute it.
- We suggest that you pre-allocate the MVS files with a minimum of SPACE(TRK(1,2)) to avoid the risk that some records are not copied.
- If the sequential files are allocated as VB, after editing the SQL, you must copy it into a partitioned dataset with an LRECL of 80 so it can be executed using DSNTEP2.

Using the SQL processor using file input facility

You can execute SQL statements in a TSO session by using the SQL processor using file input facility (SPUFI) in ISPF. SPUFI is described in the z/OS ISPF User's Guide and at the DB2 for z/OS Information Center.

4.3.5 Using the DB2 command-line processor

You can use the command-line processor to run SQL statements in the UNIX Systems Services environment from the command prompt. See 2.6.8, “Using the DB2 command line processor” on page 66.

For more information, see the following topic in the DB2 V.9 Information Center at:
http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db29.doc.inst/db2z_useclp.htm

4.4 Granting privileges to users in DB2

There are several authorization levels that you must consider when authorizing users and groups to the various DB2 objects. See 2.6.4, “Using GRANT statements in DB2” on page 58 for background information about your options.

4.4.1 Looking at GRANT statements

In this section, we provide the GRANT statements that the database tools issue and the GRANT statements that are generated with the SQL statements to create the DB2 objects.

The createDB.sh script generated GRANTs

The createDB.sh script issues these GRANT statements whenever it drops and creates the storage group and database:

- GRANT USE OF STOGROUP $DBSto TO $DBSchema WITH GRANT OPTION;
- GRANT DBADM ON DATABASE $DBName TO $DBSchema;
- GRANT USE OF BUFFERPOOL $DBBP(sss) TO $DBSchema;

Because we GRANT DBADM on the database our DB2 was not using RACF for authorization checking. If DB2 is using RACF for authorization checking, an alternative to GRANTing DBADM to the alias user ID is to connect the alias user ID to the schema RACF group.

SIB ME generated GRANTs

SQL statements are generated by the `sibDDLGenerator.sh` script for the SIB ME tables. They contain GRANT commands that you can use as a basis for granting access to the SIB tables, as shown in Example 4-7.

Example 4-7 GRANT statements generated for the SIB ME tables

```
GRANT SELECT,INSERT,UPDATE ON B7C1x.SIBOWNER TO B7DBU;
GRANT SELECT,INSERT,UPDATE ON B7C1x.SIBOWNERO TO B7DBU;
GRANT SELECT,INSERT ON B7C1x.SIBCLASSMAP TO B7DBU;
GRANT SELECT,INSERT ON B7C1x.SIBLISTING TO B7DBU;
GRANT SELECT,INSERT,DELETE,UPDATE ON B7C1x.SIB000 TO B7DBU;
GRANT SELECT,INSERT,DELETE,UPDATE ON B7C1x.SIB001 TO B7DBU;
GRANT SELECT,INSERT,DELETE,UPDATE ON B7C1x.SIB002 TO B7DBU;
GRANT SELECT,INSERT,UPDATE,DELETE ON B7C1x.SIBXACTS TO B7DBU;
GRANT SELECT,INSERT,UPDATE ON B7C1x.SIBKEYS TO B7DBU;
```

BPC Reporter generated GRANTs for functions

These GRANT statements are generated as part of the SQL statements, but commented out. They allow execute permission to the schema name for functions by the authenticated user ID (B7DBU), if it is connected to the B7CELL schema group:

- GRANT EXECUTE ON SPECIFIC FUNCTION B7CELL.INTERVALIN_TS_VC TO B7CELL;
- GRANT EXECUTE ON SPECIFIC FUNCTION B7CELL.INTERVALIN_TS_TS TO B7CELL;
- GRANT EXECUTE ON SPECIFIC FUNCTION B7CELL.INTERVALIN_VC_VC TO B7CELL;
- GRANT EXECUTE ON SPECIFIC FUNCTION B7CELL.INTERVALIN_VC_TS TO B7CELL;

Alternatively, you can GRANT EXECUTE to the authenticated User ID directly - TO B7DBU;

WPS common database sequences

Eleven sequence objects are created for the `governancerepository` component in the WebSphere Process Server for z/OS Common database. USAGE permission must be GRANTED to the authenticated user, which can be generated by the GrantGenerator tool:

GRANT USAGE ON SEQUENCE B7CELL.SEQ_W_<Sequence Name> TO B7DBU;

Other components

Other components do not supply any GRANT statements.

To understand the GRANT statements that are required for WebSphere Process Server for z/OS, should look at them in two perspectives:

- GRANT privileges required to create the database(s)
- GRANT privileges required during the operation

Summarizing GRANT statements required to create DB2 objects

The database objects must be created by a user with DBADM authority. You can assign this authority to the WebSphere administrator or to a database administrator.

Your DB2 database administrator can choose to execute all of the DDL themselves using their DB2 SYSADM privileges, or they might prefer to delegate DBADM to a user ID or group so that someone else can execute the DDL, for example, DBADM might be delegated to the WebSphere administrator user ID during the configuration of WebSphere Process Server for z/OS.

Summarizing GRANT statements required for the runtime

WebSphere Process Server for z/OS servers need necessary privileges to access the DB2 objects.

The authorization ID that must be granted the privileges depends on how you choose to configure security on the WebSphere Process Server for z/OS data sources. The identity that is used to authenticate to DB2 might be the J2C authentication alias user ID, the servant started task user ID, the propagated SAF user ID of the end-user, or the identity set on the currentSQLID custom property of the data source.

By default, the data source security is configured to use a J2C authentication alias user ID, so in the rest of this section, we assume that this is the identity that authenticates to DB2 and which must be granted privileges in DB2.

Your DB2 database administrator might decide that it is simpler to grant DBADM to the JDBC J2C authentication alias user ID rather than issue GRANTs for all of the individual DB2 objects. The disadvantage of granting DBADM is that it includes the ability to issue DROP, and your DB2 system administrator might not be happy with that. When it is not possible to grant DBADM authority to the JDBC J2C authentication alias user ID, it will be necessary to issue GRANTs on all of the tables and views.

Regardless of which of the two approaches you use, you still need the following GRANT statements as a minimum for the authentication user ID during the runtime:

- SELECT on SYSIBM.SYSTABLES
- USAGE on SEQUENCE objects created for the WPS Common database.
- EXECUTE on FUNCTIONs in the “`governamcerepository`”

Here is a summary of the two approaches:

- Granting DBADM authority

  Grant DBADM on the databases to the J2C authentication alias user ID (B7DBU), which provides the privileges to SELECT, INSERT, DELETE, UPDATE, and ALTER any table in the database and CREATE and DROP on indexes for those tables. CREATEIN and DROPIN privileges are not part of DBADM.

  Granting DBADM to the JCA authentication alias user ID also gives it the ability to DROP tables.

- Granting specific privileges on all of the tables

  If you want the WebSphere Process Server to function and not allow the alias user ID to have DROP capability, create GRANT commands from the CREATE commands, for example:

  ```
  GRANT ALL PRIVILEGES ON TABLE <Schema>.<Table_name>;
  ```

  Use the Grant Generator, which we describe in 4.4.2, “Using the GRANT Generator” on page 119.

4.4.2 Using the GRANT Generator

Because WebSphere Process Server for z/OS does not provide GRANT statements for the tables, views, and functions, we created a script that goes through all of the generated SQL statements and generates a file with the necessary GRANT statements. We also added the grants for the SEQUENCES and command-line parameters for specifying the permissions to be granted for tables and views.

The gengrant.sh script processes the specified ddls and generates a grant statement for each CREATE TABLE, CREATE VIEW, CREATE FUNCTION, and CREATE SEQUENCE statement that it finds in the input files. Depending on the create statement, different permissions are granted to the specified user:

- CREATE TABLE: If parameter `tprivs` is specified, the privileges that are specified with this parameter are granted; otherwise, ALL PRIVILEGES is used as default.
- CREATE VIEW: If parameter `vprivs` is specified, the privileges specified with this parameter are granted; otherwise, ALL PRIVILEGES is used as default.
- CREATE FUNCTION: For functions GRANT EXECUTE ON SPECIFIC FUNCTION is generated.
- CREATE SEQUENCE: For sequences GRANT USAGE ON SEQUENCE is generated.

Note: If you are using privileges with blanks in the name ('ALL PRIVILEGES'), use quotes.

The script uses the schema name that is specified with the create statements in the input file for the grant statements. If a create statement has no schema qualifier defined, the schema name that is specified with parameter *schema* is used for the grant statement.

Parameter *mode* defines whether an already existing output file is overriden (N for mode) or not (A for mode). If not specified N is the default for mode.

Parameter *infile* defines the input file(s) to be processed, which can be a comma-separated list of files that includes wildcards. If wildcards are used, the script processes all matching files and subdirectories. With each filename, you can specify the codepage of the file by appending it to the filename separated by ';' (semi-colon). Here is an example where the ddl files were copied to the /u/cherr/ddls/ directory, which processes all files in directory /u/cherr/ddls/ that have a filename that ends with .ddl. All files are processed with codepage UTF-8:

```
/u/cherr/ddls/*.ddl;UTF-8
```

Parameter *outfile* defines the output file to store the generated grant statements. You can specify a codepage to be used for the output file by appending it to the filename separated by ';' (semicolon), for example:

```
grants.ddl;UTF-8
```

This syntax creates output file grants.ddl in codepage UTF-8.

Notes:

- If using wildcards, you must enclose the values in quotes.
- If no codepage is specified with a filename, the local codepage is used as default.
- Before running gengrant.sh make sure that you exported WAS_HOME with the WAS root directory, for example, export `WAS_HOME=/wasv7config/b7cell/b7dmnode/DeploymentManager`.

Sample use of gengrant.sh

Here is a sample for calling the Grant Generator script. (All on one line with out the back-slashes (\):

```
gengrant.sh \
-infile "/u/cherr/ddls/*.sql;UTF-8,\
/u/cherr/ddls/CEI_DDL_FromWSC1/cr*.ddl;UTF-8" \
-outfile "grant_statements.sql;UTF-8" \
-schema B7CELL
```

This sample processes first all files having .sql as a file extension in the directory ddls and its subdirectories in codepage UTF-8. Next, it processes all files that have .ddl as a file extension in the directory ddls/CEI_DDL_FromWSC1 and its subdirectories, also in codepage UTF-8, and writes all generated grant statements in file grant_statements.sql, which is also in codepage UTF-8.

4.4.3 Verifying GRANT statements

You can use DB2 commands in the command-line processor, SPUFI, or the DB2 administration utility to verify that the GRANT statements are correct. Alternately, you can start up the servers and watch for DB2 violations with SQLCODEs -551, -553, and so on.

To launch the DB2 ISPF Administration application:

1. Login to TSO with a DB2 system administrator's ID.
2. Type the command (such as, ADM72 in our case).
3. Select the following options:
   – DB2 subsystem (if there are multiple subsystems)
   – 1: DB2 system catalog
   – AO: Authorization options:

GA: Storage group auths	Checks for the Administrators and Schema
DA: Database authorizations	Checks on DBADM

TA: Table authorizations	Checks on SYSIBM.SYSTABLES authorization
VA: View authorizations	Similar to Tables
UA: User authorizations	A Summary,; drill down to databases
RA: Resource authorizations	Checks BufferPools
FA: Function authorizations	
QA: Sequence authorizations	

The options can be used to view privileges on various DB2 objects and to GRANT or REVOKE privileges if you have the proper authority.

4.5 Verifying the DB2 table creation

If you encounter any problems with the DB configuration, here are some helpful resources:

- DB2 Information Center
- WebSphere Process Server for z/OS Information Center

Use the DB2 administrator’s tool or SPUFI to verify that all of the tables got created:

1. From the primary panel, select **(1)** for DB2 Catalog and **(D)** for Databases.
2. Specify a filter of ‘B7*’ to see your database, as shown in Example 4-8.

Example 4-8 D9F1 databases

```
DB2 Admin ------------------- D9F1 Databases ---------- Row 21 from 29
Command ===>  Scroll ===> CSR

Storage  Buffer         Created      Index
Select Name     Owner    Group    Pool      DBID By       T E BPool
       B7*      *        *        *            * *        * * *
------ -------- -------- -------- -------- ------ -------- - - -------
       B7WPSDB  SENIOKJ  B7WPSSG  BP1        275 SENIOKJ    E BP2
```

3. Display the tables in each of the databases by typing `T` in the Select column and pressing the Enter key, as shown in Example 4-9.

Example 4-9 DB2 table display

```
DB2 Admin ------ D9F1 Tables, Views, ---- Row 1 to 25 of 456
Command ===>

Sel   Name               Schema   T DB Name  TS Name    Cols
----- -----------------> -------- - -------- -------- ------
       STAFF_QUERY_INSTAN B7WPS    T B7WPSDB  STAF801      16
      STAFF_LOCK_T       B7WPS    T B7WPSDB  STAFF02       3
      PROCESS_TEMPLATE_B B7WPS    T B7WPSDB  TEMP802      40
      ACTIVITY_TEMPLATE_ B7WPS    T B7WPSDB  TEMP808      49
      TASK_TEMPLATE_T    B7WPS    T B7WPSDB  TEMP836      45
      SCHEMA_VERSION     B7WPS    T B7WPSDB  TEMPL01       2
   . . .
```

If you use one database, you should have 456 tables and 412 table spaces in your database.

Separate databases

If you use separate databases for WPS, BPC, CEI, and the SIBs, as we discussed in “Using multiple databases” on page 55, you will see something like this for the number of tables that we created for our B7CELLDB:

- B7WPSDB: 57 Tables, 45 TableSpaces
- B7BPCDB: 267 Tables, 264 TableSpaces
- B7BPRDB: 8 Tables, 12 TableSpaces, 10 Functions
- B7CEIDB: 38Tables, 36 TableSpaces (table names all start with cei)
- B7SIBDB: 48 Tables, 48 TableSpaces for 4 MEs (table names all start with “SIB”)
- B7BSPDB: 34Tables, 7 TableSpaces (table space names all start with “BS”)

Total: 456 Tables, 412 TableSpaces

There are also several other objects that the components create:

- BPR: 5 Functions
- WPS Common: 11 Sequences

In addition, there are 104 Tables and 106 TableSpaces for WebSphere Business Monitor.

If you are using LDAP, there might be 11 Tables for LDAP in the B7GLDDB database. The table names start with DIR_.

What to do next

After the tables and other objects are created in DB2, validated, and so on, you can now start the deployment manager and federate the empty node into the deployment manager’s cell.

4.6 Problem determination

In this section, we provide some helpful information about handling problems that are associated with database configuration. First, be aware of the Information Centers:

- DB2 Information Center

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

- WebSphere Application Server for z/OS Information Center

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

- WebSphere Process Server for z/OS Information Center

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

4.6.1 Managing DB2 objects

In this section, we list some observations about specifying, creating, and deleting various DB2 objects, which you might need to perform when initially setting up the DB2 databases for WebSphere Process Server for z/OS components:

- Verifying the jdbc driver

  See“Verifying Java and the DB2 Universal JDBC Driver” on page 68.

- Messaging engine tables

Typical tables that must be rebuilt are the ones for the message engines. You can do this using createDB.sh with the `*DBSIB` option.

- Production tables

> **Important:** After WebSphere Process Server for z/OS is operational you cannot simply DROP and CREATE the tables at any moment because they contain references to process instances and templates that are needed by SCA and the BPEL application. For the applications to work after dropping the tables, you must uninstall and re-install all of the BPEL the applications; however, this might not work correctly.

Also consider your backup strategy for WebSphere Process Server for z/OS. When applications are installed, changes occur in the configuration files in the UNIX file system and in DB2 for z/OS. Design a backup strategy that allows you to restore both the ZFS and the DB2 tablespaces so that they are consistent with each other. This process means taking backups at approximately the same time and when no application deployments are taking place.

Deleting the WPSDB common tables can actually cause the Migration Utilities to run on server startup, which does not work well on z/OS.

- Functions and sequences

There are several components that create these special DB2 objects:

– BPR: 5 Functions
– WPS CommonDB: 11 Sequences

These objects exist outside of the database. They will not be dropped by DROP DATABASE commands. You need explicit DROPs for the functions or sequences. The createDB.sh script generates DROP statements for these if DBCreate=true is specified.

> **Note:** If you use the DBCreate=true option in createDB.sh script, the DROP statements for functions and sequences will be issued automatically.

4.6.2 Password problems

There are several symptoms of problems that are associated with not having the correct password on J2C authentication definitions.

Symptom

In SYSLOG, you might see these messages:

```
ICH408I USER(B7DBU   ) GROUP(B7DBG   ) NAME(B7 WPS JDBC ALIAS   )
  LOGON/JOB INITIATION - INVALID PASSWORD
IRR013I  VERIFICATION FAILED. INVALID PASSWORD GIVEN.
```

Solution

Check that:

- The password on the J2C alias is correct.

Verify this by using the ISC to update the J2C aliases in the data source properties. Check the J2C Alias definitions.

Update all of the password fields, and restart the cell.

- The password in RACF (or your security subsystem) is correct.

  In this case, try issuing these commands to update the password and restart the cell:

  ```
  PW USER(B9DBU)     NOINTERVAL
  ALU B9DBU          PASSWORD(B9DBU) NOEXPIRED
  ```

4.6.3 Debugging database scripts

In this section, we provide some suggested troubleshooting techniques.

Debugging DbDesignGenerator.sh

Run the DbDesignGenerator.sh script before you run the createDB.sh script to create the SQL statements tailored to your environment:

- The dbDesign file must be in ascii and must have dbDesign as its file type. (Suffix)
- If you do edit the file, make sure that you leave no trailing blanks at the end of each line.
- Watch out for the following parameter that might be specified in the WCT-provided response file. It causes the /dbscripts/ directory to be created in a location other than the default (<profile_root>/utils/dbUtils/).

  ```
  dbOutputScriptDir=/wasv7config/b7cell/wpswork/dbscripts
  ```

- You can ignore the warning messages “`WLMID @ line xx .in file createFunctionsJava_Observer.ddl`” if you are using the SQL and not the Java functions.

Debugging Ddl2Pds.sh

Ddl2Pds.sh copies the SQL statements from a UNIX directory to MVS PDS members:

- If you specify a component that is not recognized, such as SCA, you will get this kind of error message:

  `Source file /wasv7config/b7cell/b7dmnode/DeploymentManager/util/dbUtils/WBI_SCA_APP_ME_DB2-zOS-9-SibME/SCA` not found

- If the PDS members are being accessed by a TSO ISPF user, even in browse mode, the transfer will fail with an error message, such as:

  ```
  cp: FSUM6259 target file "//'WASCFG.B9CELL.BPRDDL(B9BPR1)'":
  EDC5061I An error occurred when attempting to define a file to the system.
  ```

- You can ignore these error messages:
  - `FSUM6764 logname: cannot get login name`
  - `FSUM6807 expression syntax error`
- You can use the ALL option for debugging.

Debugging createDB.sh

The createDB.sh script generates and optionally executes SQL statements to create DB2 objects for WebSphere Process Server for z/OS:

- As an aid to debugging, you can insert the following statement after line number 1 to trace the substitution of parameters and variables:

  ```
  set -x
  ```

  This statement generates messages in the STDERR file for trouble-shooting.

- When we fixed these errors, they work with DBDelayConfig=true but not with DBDelayConfig=false

The DBDelayConfig setting gets fed directly into DBUtility.sh, which runs the SQL. It is normally set to false to allow the tables to be created. So with true it does not try to run anything, which gives you a good chance of not getting any errors

- Problems with the JDBC driver: `Failure in loading native library db2jcct2zos, java.lang.UnsatisfiedLinkError: db2jcct2zos` (`Not found in java.library.path`). Make sure that your two exports are set up correctly:

  ```
  export LIBPATH=/usr/lpp/db2/d9fg/db2910_jdbc/lib:$LIBPATH
  export STEPLIB=DB9F9.SDSNEXIT:DB9F9.SDSNLOAD:$STEPLIB
  ```

Debugging DSNTEP2

In this section, we provide some troubleshooting suggestions for running DSNTEP2 program in a batch job.

-454, -601 errors

If you drop a database and re-create the tables with SQL that contain CREATE FUNCTION and CREATE SEQUENCE statements, you will get SQLCODE = -454 or -601 errors, and the remaining SQL statements might be flushed after 10 such errors.

Dropping a database does not drop functions or sequences, as shown in Example 4-10.

Example 4-10 SQLCODE = -454 or -601

```
***INPUT STATEMENT:
 CREATE FUNCTION B7CELL.INTERVALIN (INTERVAL INT,
. . .;
SQLERROR ON  CREATE  COMMAND, EXECUTE  FUNCTION
 RESULT OF SQL STATEMENT:
 DSNT408I SQLCODE = -454, ERROR:  THE SIGNATURE PROVIDED IN THE CREATE FUNCTION
STATEMENT FOR B7CELL.INTERVALIN MATCHES THE SIGNATURE OF SOME OTHER FUNCTION
ALREADY EXISTING IN THE SCHEMA
 DSNT418I SQLSTATE   = 42723 SQLSTATE RETURN CODE

***INPUT STATEMENT:
 CREATE SEQUENCE B7CELL.SEQ_W_OBJ_LIT_ANY_ID
 as integer START WITH 1000 increment by 1
 no  MAXVALUE  MINVALUE 1001
 NO CYCLE CACHE 20 ORDER ;
SQLERROR ON  CREATE  COMMAND, EXECUTE  FUNCTION
 RESULT OF SQL STATEMENT:
 DSNT408I SQLCODE = -601, ERROR:  THE NAME (VERSION OR VOLUME SERIAL NUMBER) OF
THE OBJECT TO BE DEFINED . . . RENAME STATEMENT IS IDENTICAL TO THE EXISTING NAME
(VERSION OR VOLUME SERIAL NUMBER)         B7CELL.SEQ_W_OBJ_LIT_-
      ANY_ID OF THE OBJECT TYPE SEQUENCE
 DSNT418I SQLSTATE   = 42710 SQLSTATE RETURN CODE
```

Solution: You can allow these and other exceptions to occur by setting the following option at the beginning of the SYSIN DD statements:

-- SET MAXERRORS -1;

-616 SQLCODEs

If you try to drop a storage group that still contains some DB2 objects, such as an indexes, you will get the error in Example 4-11.

Example 4-11 SQLCODE

```
***INPUT STATEMENT:
  DROP  STOGROUP B7DBSTO;
SQLERROR ON  DROP   COMMAND, EXECUTE  FUNCTION
 RESULT OF SQL STATEMENT:
 DSNT408I SQLCODE = -616, ERROR:  STOGROUP B7DBSTO CANNOT BE DROPPED BECAUSE IT IS
REFERENCED BY INDEX B7LDAP.DIR_ENTRYXO
 DSNT418I SQLSTATE   = 42893 SQLSTATE RETURN CODE
```

Solution: Drop the Index and Tables of the database before dropping the storage group.

-647 SQLCODEs

Example 4-12 shows Error -647.

Example 4-12 SQLCODE -647

```
***INPUT STATEMENT:
 CREATE TABLESPACE INST805
  IN B9BPCDB
  USING STOGROUP B9DBSTO
  BUFFERPOOL BP8K1
. . .
SQLERROR ON  CREATE  COMMAND, EXECUTE  FUNCTION
 RESULT OF SQL STATEMENT:
 DSNT408I SQLCODE = -647, ERROR:  BUFFERPOOL BP8K1 FOR IMPLICIT OR EXPLICIT
TABLESPACE OR INDEXSPACE B7BPCDB.INST805 HAS NOT BEEN ACTIVATED
DSNT418I SQLSTATE   = 57003 SQLSTATE RETURN CODE
```

Solution: Specify a different buffer pool or add it with the following SQL statement in your SYSTSIN DD data:

```
-ALTER BUFFERPOOL(BP8K1) VPSIZE(2000);
```

4.7 DB2 terminology

There are many DB2 constructs that you will need to set up your databases for WebSphere Process Server for z/OS. Sample names in this paper help clarify these terms. As always, work with your DBA to assign the names and perform the tasks:

VCAT High-level dataset name qualifier (or alias) for the VSAM backed datasets for WebSphere Process Server for z/OS databases.

Volumes Contain VSAM backing datasets for these databases or an SMS-managed storage class.

DB2 Storage groups (STOGROUPs) for databases that help them to be managed and refreshed as a group.

Database Collection of tables that reference the storage group.

Tablespaces VSAM datasets that contain tables with records in them.

Tablespace names must be unique within storage groups.

Tables Data objects consisting of rows and specified columns.

Indexes VSAM-backed files that contain pointers to rows in a table.

Schema name Qualifier (typically CREATOR) to uniquely define SQL definitions (Tables, Views, Procs, Indices) and avoid name collisions across databases and between tables for multiple SIBs in a server.

Table Names must be unique within a subsystem (ssid) on z/OS where DB2 subsystems can access multiple Databases.

User IDs Connects to the databases from the WPS runtime (authorization IDs and SQL IDs). These are usually IDs with DBADM authority.

DB2 stored procedure Builder must be enabled in your DB2 subsystems (built with the DSNTPSMP utility) if you are using Relationships. The Stored Procedure is used because multiple relationships will exist for each Key. When the relationship is updated, there are typically several rows that are being updated/deleted. The storedProcedure manages these updates locally to the database.

STEPLIBs If you do not have the DB2 libraries in your linklist concatenation, you must add them to your Procs and shell scripts (SDSNEXIT, SDSNLOAD, SDSNLOD2).

Buffer Pools See “Planning DB2 buffer pools” on page 56.

SSID DB2 subsystem identifier.

JCC Java Common Connectivity is a DB2 feature that supports the IBM DB2 Driver for JDBC and SQLJ. Additional DB2 tailoring may be required to support this feature. (See DSNTIJSG).

For a good paper about DB2 concepts, see:
`http://www.ibm.com/developerworks/data/library/techarticle/0212wieser/0212wieser.html`

5

Configuring the WebSphere Process Server cluster

In this chapter, we describe how to configure a WebSphere Process Server for z/OS cluster and all of the components that WebSphere Process Server for z/OS uses.

In this chapter, we discuss:

5.1 Overview of the WebSphere Process Server cluster

After the deployment manager and node profiles are created and augmented with the WebSphere Process Server for z/OS code and the node is federated into the cell you can start creating the network deployment (ND) topology.

As we already discussed in Chapter 2, “Planning for and preparing WebSphere Process Server” on page 25, the preferred ND topology for WebSphere Process Server for z/OS is the single cluster topology. There are several methods for configuring such a topology:

- Manual configuration: This is the way a ND topology was configured in the past, which implies creating everything by hand, starting with the creation of the cluster and cluster members, and then installing and configuring all of the components and applications step-by-step and finally adjusting resource properties.
- Using the pattern-based deployment environment wizard: This wizard performs all of the installation and configuration of the topology under the covers. You only perform some final adjustments to some resource properties, for example, the short names of cluster and cluster members. The main disadvantage of this wizard is that it also generates the long names for the cluster and cluster members, which cannot be changed anymore.
- Using the custom deployment environment wizard: This wizard is a mixture of the manual configuration and the pattern-based deployment environment wizard. You can use your cluster and cluster members, and the wizard installs and configures the components and applications. The disadvantage of this wizard is that it does not install Business Space powered by WebSphere, and it also does not support the usage of the output file generated by the dbDesign tool.

Our preferred and recommended method is the second method, the pattern based deployment environment wizard. If you need your own cluster and cluster member long names, use the last method.

> **Note**: The illustrations and the values that we use in these chapters are from cell b8cell, which is very similar to cell b7cell.

5.2 Creating a cluster using the pattern-based deployment environment wizard

In this section, we use the pattern-based deployment environment wizard for creating the network deployment topology.

5.2.1 Task overview

In summary, the steps to create a deployment environment are:

1. Check prerequisites.
2. Disable auto-synchronization for the nodes.
3. Extract the xml on sheet DE from the configuration spreadsheet.
4. Import the extracted xml file in the deployment environment wizard, and verify the settings. Alternatively you can go through the pattern-based deployment environment wizard without importing an existing topology definition xml file.

5. Save and generate the deployment environment.
6. Adjust resource definitions.
7. Create a log structure and log streams for the XA transaction partner logs.
8. Configure IBM HTTP Server(s) if you want to use web server(s) for load balancing.

5.2.2 Checking prerequisites

Before you can configure your deployment environment, make sure that:

- The deployment manager and managed nodes profiles are augmented with WebSphere Process Server for z/OS.
- The deployment manager and, optionally, the node agents are started and running.
- The database and database artifacts for your WebSphere Process Server for z/OS cell are created.

5.2.3 Disabling auto-synchronization for the nodes

Make sure that auto-synchronization for the nodes is disabled. If auto-synchronization is enabled you will get strange error messages at the end of the deployment environment generation when you save the changes to the master repository:

1. Login to the Integrated Solutions Console, and navigate to **System administration** → **Node agents**.
2. Click the first Node Agent listed, and then click **File synchronization service**.
3. Clear the Automatic synchronization option, and then click **OK**.

5.2.4 Extracting the deployment environment configuration from the spreadsheet

If you want to make your life easy, use the deployment environment configuration that is generated in the configuration spreadsheet, which ensures that you have the correct values specified for all of your components and artifacts created during the deployment environment generated.

To extract the deployment environment configuration from the spreadsheet:

1. Go to the sheet named DE in the configuration spreadsheet, as shown in Figure 5-1.

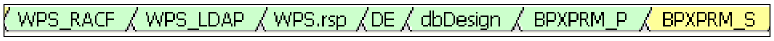
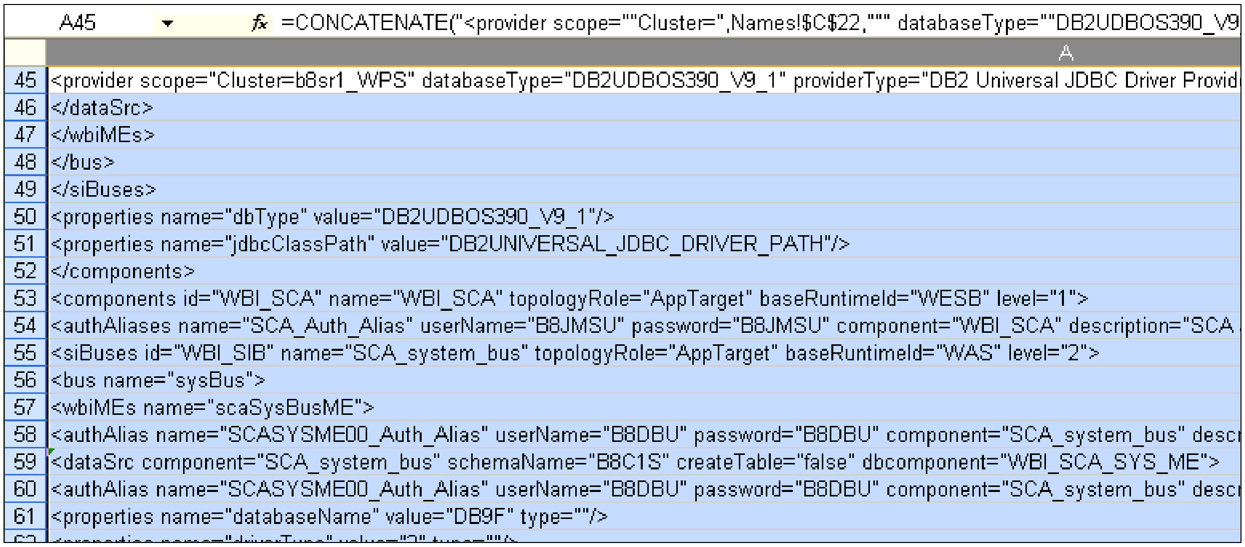


Figure 5-1 DE sheet

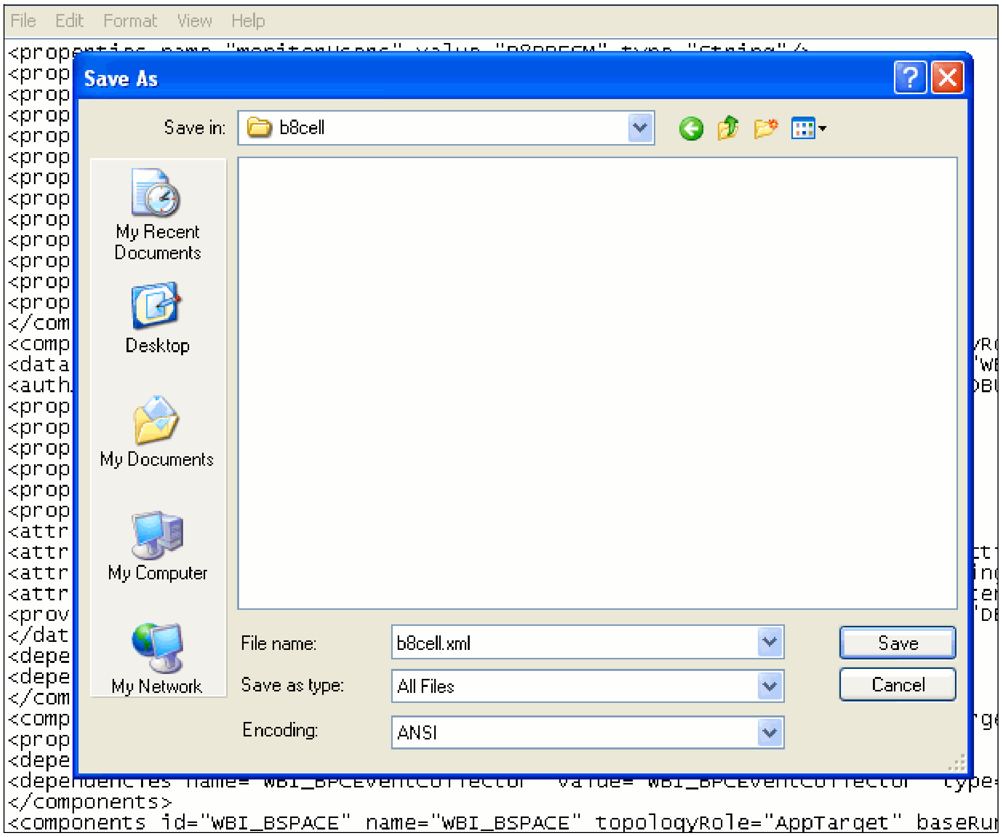
2. Move the mouse pointer on the heading of column A, and click one time. This action selects all of the rows in column A, as shown in Figure 5-2 on page 132.



```
A45    fx =CONCATENATE("<provider scope=""Cluster=",Names!$C$22,""" databaseType=""DB2UDBOS390_V9
45 <provider scope="Cluster=b8sr1_WPS" databaseType="DB2UDBOS390_V9_1" providerType="DB2 Universal JDBC Driver Provid
46 </dataSrc>
47 </wbiMEs>
48 </bus>
49 </siBuses>
50 <properties name="dbType" value="DB2UDBOS390_V9_1"/>
51 <properties name="jdbcClassPath" value="DB2UNIVERSAL_JDBC_DRIVER_PATH"/>
52 </components>
53 <components id="WBI_SCA" name="WBI_SCA" topologyRole="AppTarget" baseRuntimeId="WESB" level="1">
54 <authAliases name="SCA_Auth_Alias" userName="B8JMSU" password="B8JMSU" component="WBI_SCA" description="SCA
55 <siBuses id="WBI_SIB" name="SCA_system_bus" topologyRole="AppTarget" baseRuntimeId="WAS" level="2">
56 <bus name="sysBus">
57 <wbiMEs name="scaSysBusME">
58 <authAlias name="SCASYSME00_Auth_Alias" userName="B8DBU" password="B8DBU" component="SCA_system_bus" descr
59 <dataSrc component="SCA_system_bus" schemaName="B8C1S" createTable="false" dbcomponent="WBI_SCA_SYS_ME">
60 <authAlias name="SCASYSME00_Auth_Alias" userName="B8DBU" password="B8DBU" component="SCA_system_bus" descr
61 <properties name="databaseName" value="DB9F" type=""/>
```

Figure 5-2 Select deployment environment configuration definitions

3. Copy and paste the selection into an ASCII text editor, such as notepad, and save it in a file, as shown in Figure 5-3. Make sure that you use the file extension .xml for the filename.



Save in: b8cell

File name: b8cell.xml

Save as type: All Files

Encoding: ANSI

```
</components>
<components id="WBI_BSPACE" name="WBI_BSPACE" topologyRole="AppTarget" baseRu
```

Figure 5-3 Save deployment environment configuration definitions to an XML file

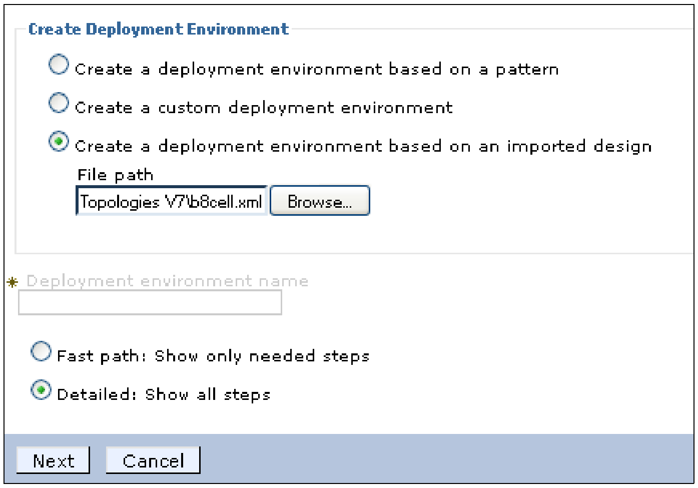
5.2.5 Importing the deployment environment configuration XML file

In this section, you will import an existing deployment environment configuration file into the deployment environment wizard. You then go through the panels of the wizard and verify and modify, where necessary, the specified values.

Important: Before you import the previously created deployment environment configuration file make sure that you configured and federated all of the nodes in your cell that you are using in your deployment configuration file; otherwise, the import will fail.

To import the deployment environment configuration XML file:

1. Login to the Integrated Solutions Console (ISC).
2. Open the Servers folder, and click **Deployment Environments**.
3. In the Deployment Environments view, click **Import**.
4. Browse to the Deployment Environment XML file, and click **Detailed: Show all steps**, as shown in Figure 5-4.



Create Deployment Environment

○ Create a deployment environment based on a pattern

○ Create a custom deployment environment

● Create a deployment environment based on an imported design

File path: Topologies V7\b8cell.xml

* Deployment environment name

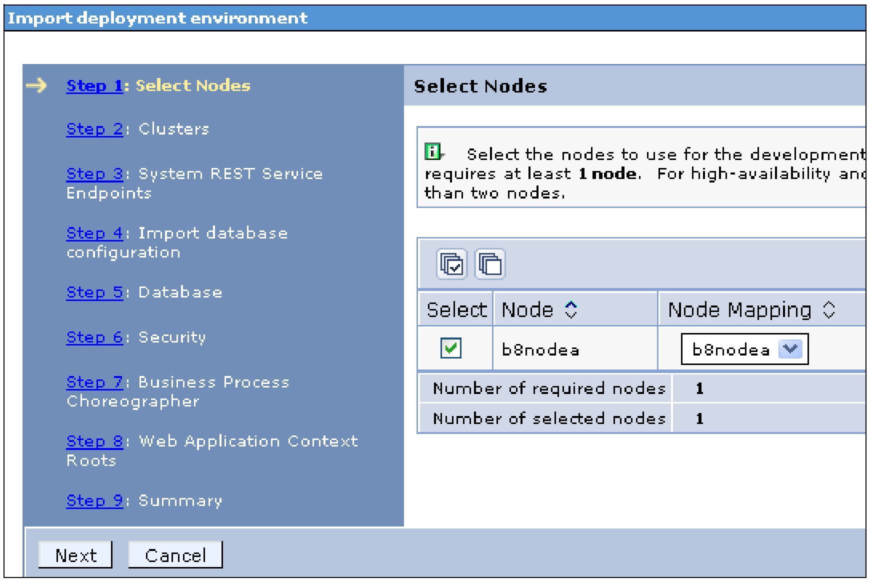
○ Fast path: Show only needed steps

● Detailed: Show all steps

Figure 5-4 Choose deployment environment configuration file

5. After the file is successfully imported you can go through the panels by pressing the **Next**. Verify the values in the fields, and make modifications where necessary.

   Figure 5-5 on page 134 shows the steps for the deployment environment wizard.


Import deployment environment
Step 1: Select Nodes
Step 2: Clusters
Step 3: System REST Service Endpoints
Step 4: Import database configuration
Step 5: Database
Step 6: Security
Step 7: Business Process Choreographer
Step 8: Web Application Context Roots
Step 9: Summary
Select Nodes
Next
Cancel

Select the nodes to use for the development
requires at least **1 node**. For high-availability and
than two nodes.

Select	Node	Node Mapping
☑	b8nodea	b8nodea

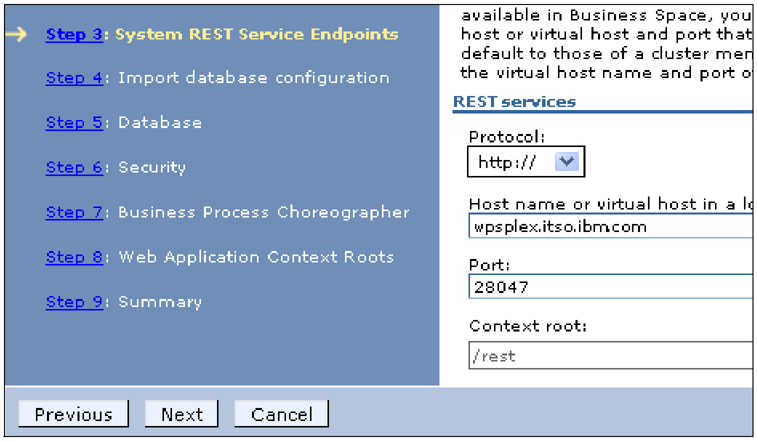
Number of required nodes	1
Number of selected nodes	1

Figure 5-5 Steps of deployment environment wizard

On Step 1: Select nodes, shown in Figure 5-5, the Node column lists the nodes that are federated into the cell, and the Node Mapping column shows the Nodes that are currently part of the deployment environment configuration that was imported from the DE worksheet of the planning spreadsheet. The <nodes name> tag in the deployment environment XML file that you imported into the wizard determines the number of nodes in the DE configuration.

> **Note:** The spreadsheet assumes that the cluster will comprise two cluster members across two nodes; therefore, you must manually add one or more <nodes name tags to the DE configuration XML file if your cluster will span more than two nodes.

6. Specify the HTTP server's distributed VIPA and port, as shown in Figure 5-6.


Step 3: System REST Service Endpoints
Step 4: Import database configuration
Step 5: Database
Step 6: Security
Step 7: Business Process Choreographer
Step 8: Web Application Context Roots
Step 9: Summary
REST services
Previous
Next
Cancel

available in Business Space, you
host or virtual host and port that
default to those of a cluster men
the virtual host name and port o

Protocol: http://

Host name or virtual host in a lo
wpsplex.itso.ibm.com

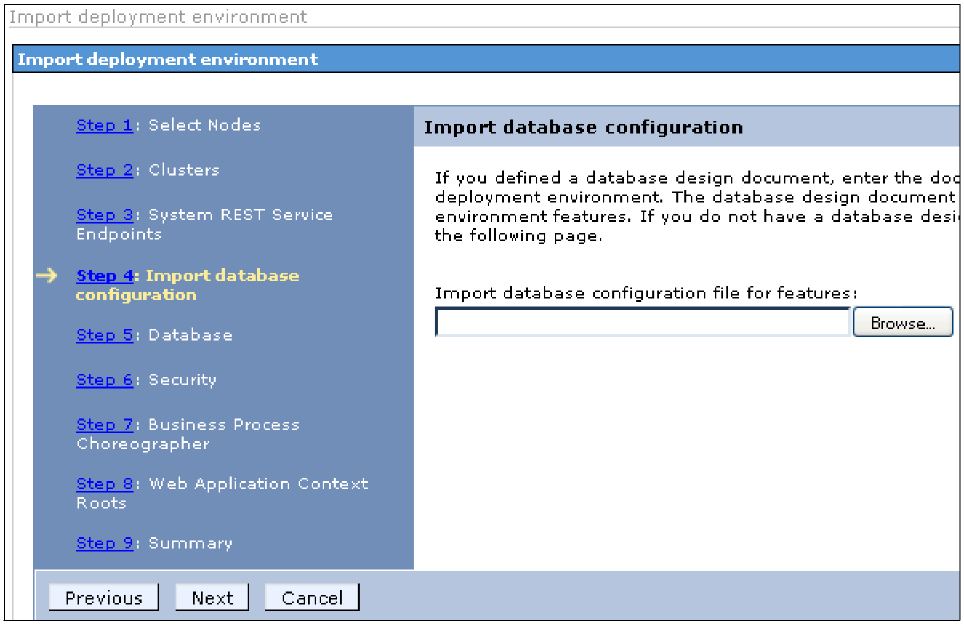
Port:
28047

Context root:
/rest

Figure 5-6 REST service endpoints

7. On the Import database configuration panel, Figure 5-7 on page 135, if you made chances outside of the used configuration spreadsheet, you can import the output file of the dbDesign tool; otherwise, you can move to the next panel. For more information about how to use the dbDesign tool, see the WebSphere Process Server for z/OS InfoCenter at

http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/topic/com.ibm.websphere.wesb.doc/doc/tcfg_ddl_generator.html



Import deployment environment

Import deployment environment

Step 1: Select Nodes

Step 2: Clusters

Step 3: System REST Service Endpoints

→ Step 4: Import database configuration

Step 5: Database

Step 6: Security

Step 7: Business Process Choreographer

Step 8: Web Application Context Roots

Step 9: Summary

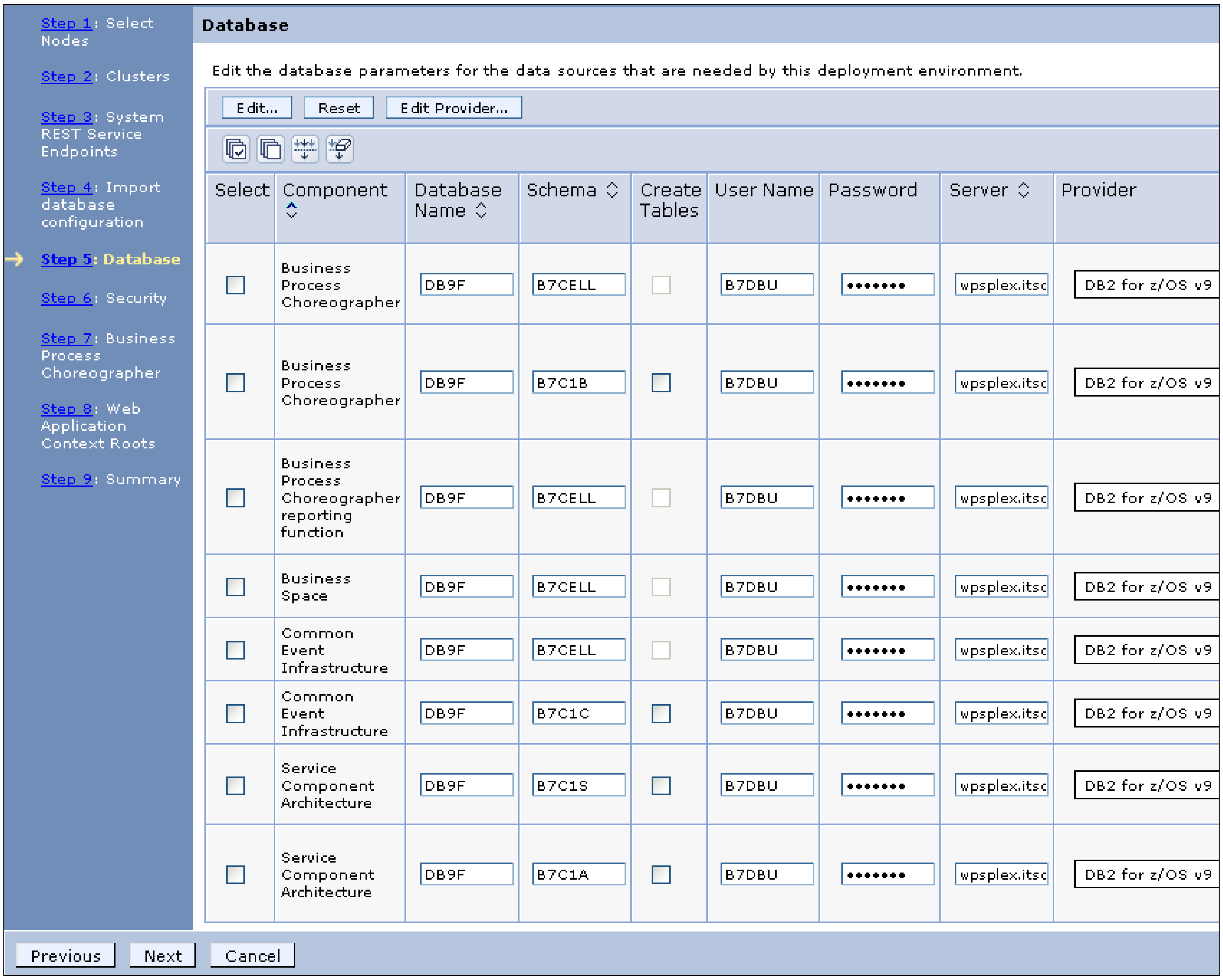
Import database configuration

If you defined a database design document, enter the doc
deployment environment. The database design document
environment features. If you do not have a database desi
the following page.

Import database configuration file for features:

Figure 5-7 Optionally import database configuration file

8. On the Database panel, Figure 5-8 on page 136, make sure that the:
   - Fields in the Database Name column contain the DB2 location name
   - Schema names are set correctly, especially for the datasources of the messaging engines
   - User that is specified in the User Name column is the user that is used with the grant statements of the database setup
   - Checkboxes in the Create Tables column are all unchecked


Step 1: Select Nodes
Step 2: Clusters
Step 3: System REST Service Endpoints
Step 4: Import database configuration
Step 5: Database
Step 6: Security
Step 7: Business Process Choreographer
Step 8: Web Application Context Roots
Step 9: Summary
Edit...
Reset
Edit Provider...
Previous
Next
Cancel

Database

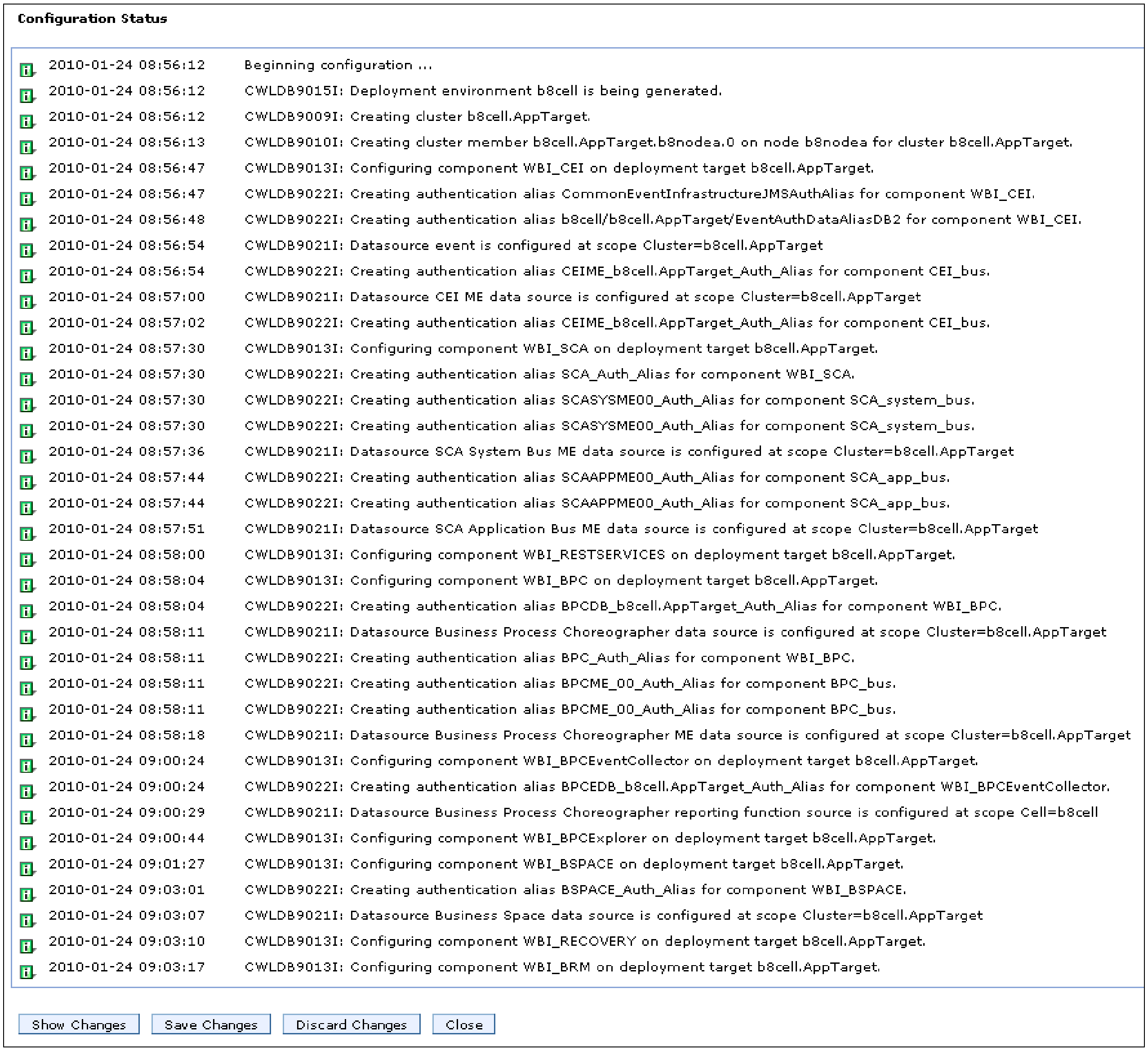
Edit the database parameters for the data sources that are needed by this deployment environment.

Select	Component	Database Name	Schema	Create Tables	User Name	Password	Server	Provider
□	Business Process Choreographer	DB9F	B7CELL	□	B7DBU	•••••••	wpsplex.itsc	DB2 for z/OS v9
□	Business Process Choreographer	DB9F	B7C1B	□	B7DBU	•••••••	wpsplex.itsc	DB2 for z/OS v9
□	Business Process Choreographer reporting function	DB9F	B7CELL	□	B7DBU	•••••••	wpsplex.itsc	DB2 for z/OS v9
□	Business Space	DB9F	B7CELL	□	B7DBU	•••••••	wpsplex.itsc	DB2 for z/OS v9
□	Common Event Infrastructure	DB9F	B7CELL	□	B7DBU	•••••••	wpsplex.itsc	DB2 for z/OS v9
□	Common Event Infrastructure	DB9F	B7C1C	□	B7DBU	•••••••	wpsplex.itsc	DB2 for z/OS v9
□	Service Component Architecture	DB9F	B7C1S	□	B7DBU	•••••••	wpsplex.itsc	DB2 for z/OS v9
□	Service Component Architecture	DB9F	B7C1A	□	B7DBU	•••••••	wpsplex.itsc	DB2 for z/OS v9

Figure 5-8 Datasources configuration overview

9. On the summary panel, you can review some of the settings. Click **Finish**. Click **Generate Environment** to create the deployment environment.

10. In the Figure 5-9 on page 137, you can track the processing of the deployment environment generation.



```
2010-01-24 08:56:12    Beginning configuration ...
2010-01-24 08:56:12    CWLDB9015I: Deployment environment b8cell is being generated.
2010-01-24 08:56:12    CWLDB9009I: Creating cluster b8cell.AppTarget.
2010-01-24 08:56:13    CWLDB9010I: Creating cluster member b8cell.AppTarget.b8nodea.0 on node b8nodea for cluster b8cell.AppTarget.
2010-01-24 08:56:47    CWLDB9013I: Configuring component WBI_CEI on deployment target b8cell.AppTarget.
2010-01-24 08:56:47    CWLDB9022I: Creating authentication alias CommonEventInfrastructureJMSAuthAlias for component WBI_CEI.
2010-01-24 08:56:48    CWLDB9022I: Creating authentication alias b8cell/b8cell.AppTarget/EventAuthDataAliasDB2 for component WBI_CEI.
2010-01-24 08:56:54    CWLDB9021I: Datasource event is configured at scope Cluster=b8cell.AppTarget
2010-01-24 08:56:54    CWLDB9022I: Creating authentication alias CEIME_b8cell.AppTarget_Auth_Alias for component CEI_bus.
2010-01-24 08:57:00    CWLDB9021I: Datasource CEI ME data source is configured at scope Cluster=b8cell.AppTarget
2010-01-24 08:57:02    CWLDB9022I: Creating authentication alias CEIME_b8cell.AppTarget_Auth_Alias for component CEI_bus.
2010-01-24 08:57:30    CWLDB9013I: Configuring component WBI_SCA on deployment target b8cell.AppTarget.
2010-01-24 08:57:30    CWLDB9022I: Creating authentication alias SCA_Auth_Alias for component WBI_SCA.
2010-01-24 08:57:30    CWLDB9022I: Creating authentication alias SCASYSME00_Auth_Alias for component SCA_system_bus.
2010-01-24 08:57:30    CWLDB9022I: Creating authentication alias SCASYSME00_Auth_Alias for component SCA_system_bus.
2010-01-24 08:57:36    CWLDB9021I: Datasource SCA System Bus ME data source is configured at scope Cluster=b8cell.AppTarget
2010-01-24 08:57:44    CWLDB9022I: Creating authentication alias SCAAPPME00_Auth_Alias for component SCA_app_bus.
2010-01-24 08:57:44    CWLDB9022I: Creating authentication alias SCAAPPME00_Auth_Alias for component SCA_app_bus.
2010-01-24 08:57:51    CWLDB9021I: Datasource SCA Application Bus ME data source is configured at scope Cluster=b8cell.AppTarget
2010-01-24 08:58:00    CWLDB9013I: Configuring component WBI_RESTSERVICES on deployment target b8cell.AppTarget.
2010-01-24 08:58:04    CWLDB9013I: Configuring component WBI_BPC on deployment target b8cell.AppTarget.
2010-01-24 08:58:04    CWLDB9022I: Creating authentication alias BPCDB_b8cell.AppTarget_Auth_Alias for component WBI_BPC.
2010-01-24 08:58:11    CWLDB9021I: Datasource Business Process Choreographer data source is configured at scope Cluster=b8cell.AppTarget
2010-01-24 08:58:11    CWLDB9022I: Creating authentication alias BPC_Auth_Alias for component WBI_BPC.
2010-01-24 08:58:11    CWLDB9022I: Creating authentication alias BPCME_00_Auth_Alias for component BPC_bus.
2010-01-24 08:58:11    CWLDB9022I: Creating authentication alias BPCME_00_Auth_Alias for component BPC_bus.
2010-01-24 08:58:18    CWLDB9021I: Datasource Business Process Choreographer ME data source is configured at scope Cluster=b8cell.AppTarget
2010-01-24 09:00:24    CWLDB9013I: Configuring component WBI_BPCEventCollector on deployment target b8cell.AppTarget.
2010-01-24 09:00:24    CWLDB9022I: Creating authentication alias BPCEDB_b8cell.AppTarget_Auth_Alias for component WBI_BPCEventCollector.
2010-01-24 09:00:29    CWLDB9021I: Datasource Business Process Choreographer reporting function source is configured at scope Cell=b8cell
2010-01-24 09:00:44    CWLDB9013I: Configuring component WBI_BPCExplorer on deployment target b8cell.AppTarget.
2010-01-24 09:01:27    CWLDB9013I: Configuring component WBI_BSPACE on deployment target b8cell.AppTarget.
2010-01-24 09:03:01    CWLDB9022I: Creating authentication alias BSPACE_Auth_Alias for component WBI_BSPACE.
2010-01-24 09:03:07    CWLDB9021I: Datasource Business Space data source is configured at scope Cluster=b8cell.AppTarget
2010-01-24 09:03:10    CWLDB9013I: Configuring component WBI_RECOVERY on deployment target b8cell.AppTarget.
2010-01-24 09:03:17    CWLDB9013I: Configuring component WBI_BRM on deployment target b8cell.AppTarget.
```

Figure 5-9 Deployment environment generation completed

11. When the generation of the deployment environment finishes, click **Save Changes** to save the changes to the master repository.

12. The Deployment Environment configuration is now listed but with a red cross showing that it stopped. Click **Deployment Environment configuration**, which you can view in Figure 5-10 on page 138.

13. At the top of Figure 5-10 on page 138, there is a message informing you to perform deferred configuration steps. There are deferred configuration steps because the wizard knows that the WebSphere Process Server for z/OS tables are created by another process (Chapter 4, “Configuring the database resources” on page 99), and you must confirm that the tables were created before continuing.

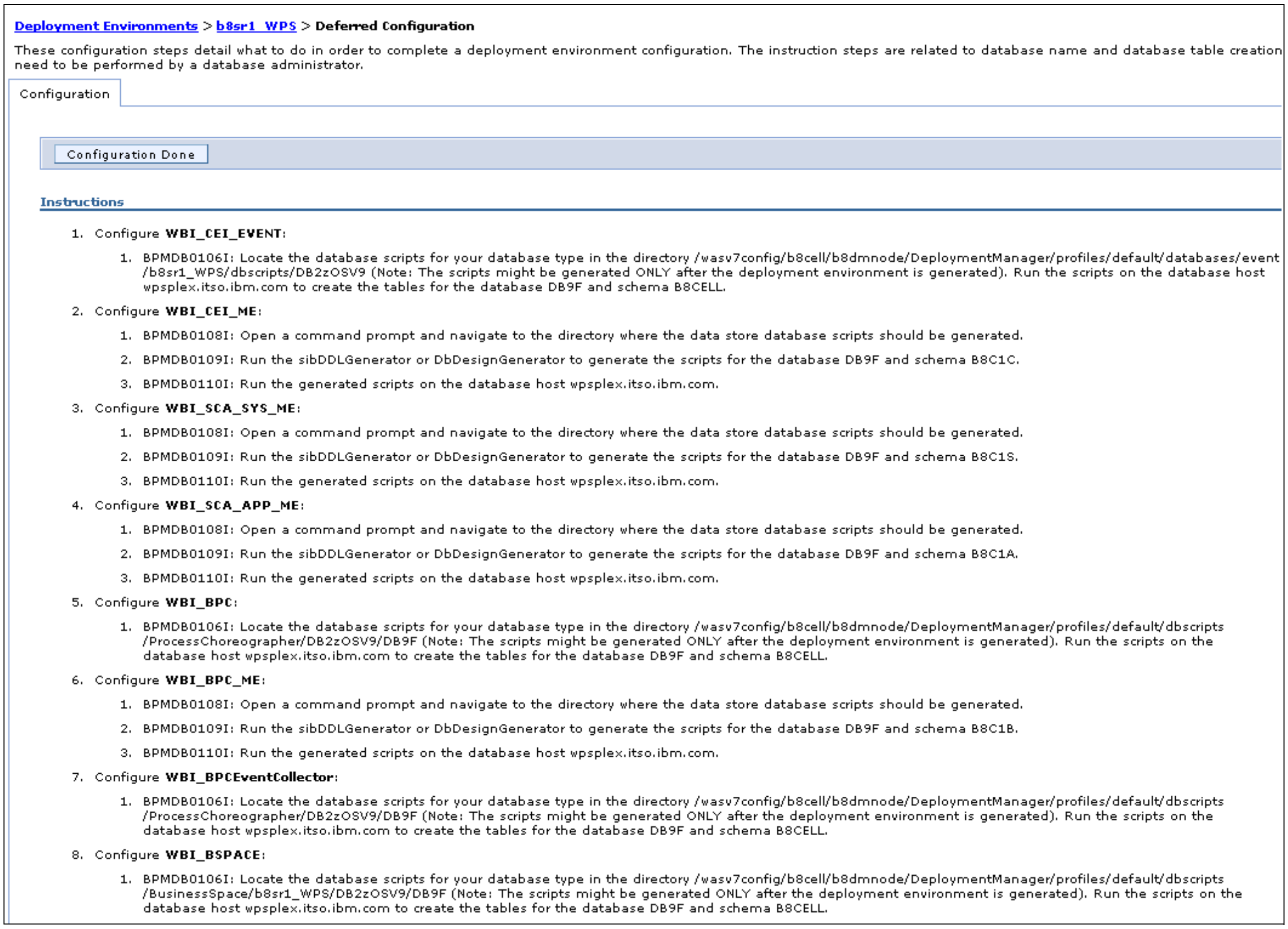
    Under Additional properties, click **Deferred Configuration**, click **Configuration Done**, save the configuration change, and synchronize to the nodes.

Required ddl files: The deployment environment wizard also generates the required ddl files in the profile directory of the deployment manager:

- <profile_root>/databases/event/k1sr1_WPS/dbscripts/db2zos/ddl for the CEI database artifacts
- <profile_root>/dbscripts/BusinessSpace/b8sr1_WPS/DB2zOSV9/DB9F for the Business Space database artifacts
- <profile_root>/dbscripts/CommonDB/DB2zOSV9/B8CELL for the database artifacts of the process server common database
- <profile_root>/dbscripts/ProcessChoreographer/DB2zOSV9/DB9F/B8CELL for the BPC/HTC database artifacts
- for the SIBus database artifacts you need to create the ddl files by using the sibDDLGenerator.sh script located in the bin directory of the deployment manager profile directory

Figure 5-10 shows the location of the DDL that the Deployment Environment wizard generates.

We recommend that you use the DbDesignGenerator and createDB.sh script to generate and customize the DDL because at this time those scripts do a better job at creating DDL that requires minimal additional editing.



Deployment Environments > b8sr1_WPS > Deferred Configuration

These configuration steps detail what to do in order to complete a deployment environment configuration. The instruction steps are related to database name and database table creation need to be performed by a database administrator.

Instructions

1. Configure **WBI_CEI_EVENT**:
   1. BPMDB0106I: Locate the database scripts for your database type in the directory /wasv7config/b8cell/b8dmnode/DeploymentManager/profiles/default/databases/event/b8sr1_WPS/dbscripts/DB2zOSV9 (Note: The scripts might be generated ONLY after the deployment environment is generated). Run the scripts on the database host wpsplex.itso.ibm.com to create the tables for the database DB9F and schema B8CELL.
2. Configure **WBI_CEI_ME**:
   1. BPMDB0108I: Open a command prompt and navigate to the directory where the data store database scripts should be generated.
   2. BPMDB0109I: Run the sibDDLGenerator or DbDesignGenerator to generate the scripts for the database DB9F and schema B8C1C.
   3. BPMDB0110I: Run the generated scripts on the database host wpsplex.itso.ibm.com.
3. Configure **WBI_SCA_SYS_ME**:
   1. BPMDB0108I: Open a command prompt and navigate to the directory where the data store database scripts should be generated.
   2. BPMDB0109I: Run the sibDDLGenerator or DbDesignGenerator to generate the scripts for the database DB9F and schema B8C1S.
   3. BPMDB0110I: Run the generated scripts on the database host wpsplex.itso.ibm.com.
4. Configure **WBI_SCA_APP_ME**:
   1. BPMDB0108I: Open a command prompt and navigate to the directory where the data store database scripts should be generated.
   2. BPMDB0109I: Run the sibDDLGenerator or DbDesignGenerator to generate the scripts for the database DB9F and schema B8C1A.
   3. BPMDB0110I: Run the generated scripts on the database host wpsplex.itso.ibm.com.
5. Configure **WBI_BPC**:
   1. BPMDB0106I: Locate the database scripts for your database type in the directory /wasv7config/b8cell/b8dmnode/DeploymentManager/profiles/default/dbscripts/ProcessChoreographer/DB2zOSV9/DB9F (Note: The scripts might be generated ONLY after the deployment environment is generated). Run the scripts on the database host wpsplex.itso.ibm.com to create the tables for the database DB9F and schema B8CELL.
6. Configure **WBI_BPC_ME**:
   1. BPMDB0108I: Open a command prompt and navigate to the directory where the data store database scripts should be generated.
   2. BPMDB0109I: Run the sibDDLGenerator or DbDesignGenerator to generate the scripts for the database DB9F and schema B8C1B.
   3. BPMDB0110I: Run the generated scripts on the database host wpsplex.itso.ibm.com.
7. Configure **WBI_BPCEventCollector**:
   1. BPMDB0106I: Locate the database scripts for your database type in the directory /wasv7config/b8cell/b8dmnode/DeploymentManager/profiles/default/dbscripts/ProcessChoreographer/DB2zOSV9/DB9F (Note: The scripts might be generated ONLY after the deployment environment is generated). Run the scripts on the database host wpsplex.itso.ibm.com to create the tables for the database DB9F and schema B8CELL.
8. Configure **WBI_BSPACE**:
   1. BPMDB0106I: Locate the database scripts for your database type in the directory /wasv7config/b8cell/b8dmnode/DeploymentManager/profiles/default/dbscripts/BusinessSpace/b8sr1_WPS/DB2zOSV9/DB9F (Note: The scripts might be generated ONLY after the deployment environment is generated). Run the scripts on the database host wpsplex.itso.ibm.com to create the tables for the database DB9F and schema B8CELL.

Figure 5-10 Deferred configuration steps

5.3 Making adjustments to the configured resources

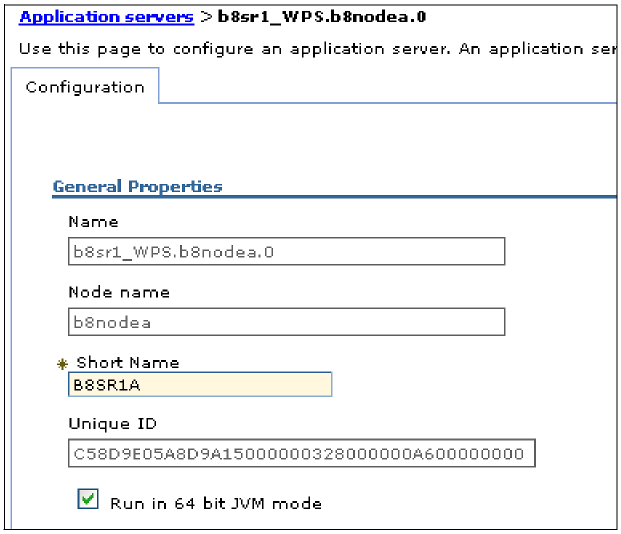
After successfully generating the deployment environment, various resources need further modifications, such as:

- Server and cluster short names
- Ports of servers
- REST services endpoints
- Virtual hosts
- JVM custom properties of server processes
- Set currentSchema and remove currentSQLID from the custom properties of datasources
- Peer recovery and transaction logs
- Setting auto-synchronization and synchronize the nodes

5.3.1 Adjusting the short names of the servers and the cluster

Up to now the pattern-based deployment environment wizard does not support the configuration of custom cluster and server long names or custom cluster and server short names. Whereas, the long names cannot be changed, the short names can be modified manually:

1. Open the Servers folder, and then open the Server Types folder. Click **WebSphere application servers**.
2. In the application server list, shown in Figure 5-11, click the first application server, and enter the appropriate short name in the Short Name field, as defined in the configuration spreadsheet. Click **OK**.



Application servers > b8sr1_WPS.b8nodea.0

Use this page to configure an application server. An application ser

Configuration

General Properties

Name

b8sr1_WPS.b8nodea.0

Node name

b8nodea

* Short Name

B8SR1A

Unique ID

C58D9E05A8D9A15000000328000000A600000000

☑ Run in 64 bit JVM mode

Figure 5-11 Enter short name of application server

3. Repeat the previous step for all of your application servers.
4. Change the short name of your cluster. Open the **Servers** → **Clusters** folders, and click **WebSphere application server clusters**, as shown in Figure 5-12 on page 140.

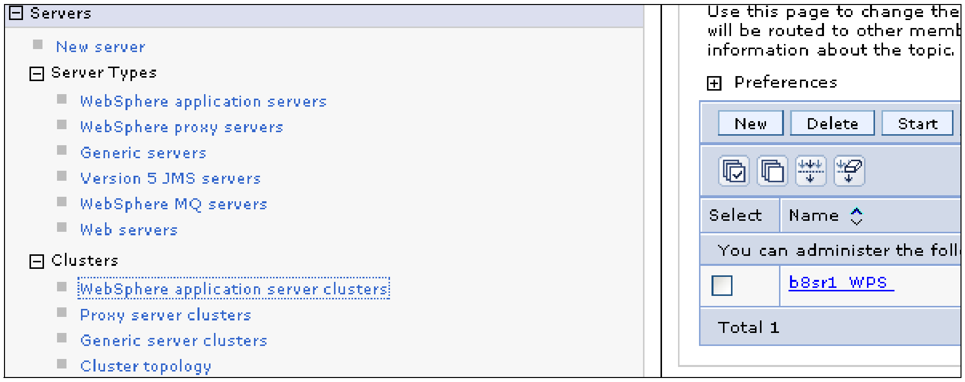
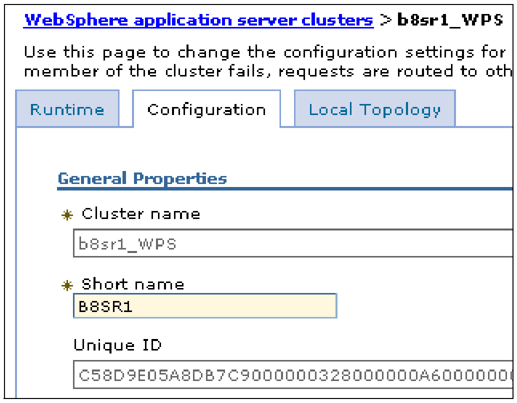


Figure 5-12 Open application server cluster list

5. Click the cluster name, and enter the appropriate cluster short name, as defined in your configuration spreadsheet in the Short Name field, as shown in Figure 5-13. Click **OK**.


WebSphere application server clusters > b8sr1_WPS
Use this page to change the configuration settings for
member of the cluster fails, requests are routed to oth
Runtime
Configuration
Local Topology
General Properties

* Cluster name

b8sr1_WPS

* Short name

B8SR1

Unique ID

C58D9E05A8DB7C9000000328000000A6000000

Figure 5-13 Cluster short name

6. Save the changes to the master repository.

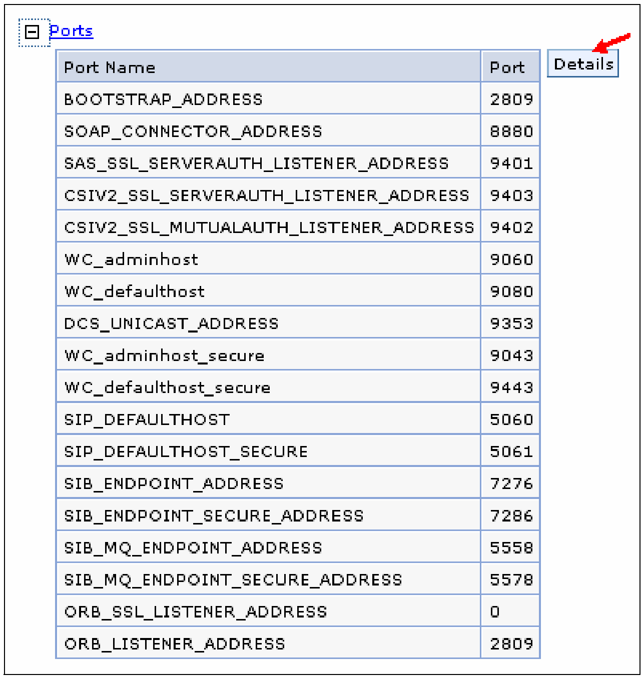
5.3.2 Adjusting the ports of the servers

Because the ports of the servers were generated by the deployment environment wizard they do not match with the values that are defined in the configuration spreadsheet. Therefore the ports have to be adjusted now, which you can do manually, as described in this section or using the Jython script updNewServerv7.py that we provided with the additional material of this book.

Adjusting the ports of the servers manually

To adjust the ports of the servers manually:

1. Open the **Servers** → **Server Types**, and click **WebSphere application servers**.
2. In the application server list, click the first application server, and click **Ports**. If you only open the Ports folder, you must click **Details** to modify the ports, as shown in Figure 5-15 on page 144.



Port Name	Port
BOOTSTRAP_ADDRESS	2809
SOAP_CONNECTOR_ADDRESS	8880
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	9401
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	9403
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	9402
WC_adminhost	9060
WC_defaulthost	9080
DCS_UNICAST_ADDRESS	9353
WC_adminhost_secure	9043
WC_defaulthost_secure	9443
SIP_DEFAULTHOST	5060
SIP_DEFAULTHOST_SECURE	5061
SIB_ENDPOINT_ADDRESS	7276
SIB_ENDPOINT_SECURE_ADDRESS	7286
SIB_MQ_ENDPOINT_ADDRESS	5558
SIB_MQ_ENDPOINT_SECURE_ADDRESS	5578
ORB_SSL_LISTENER_ADDRESS	0
ORB_LISTENER_ADDRESS	2809

Figure 5-14 Application server default ports list

3. Go through the ports list, and enter the values as defined in the configuration spreadsheet for the appropriate server. Click **OK**.
4. Repeat steps 2 through 3 for all of your application servers.
5. Save the changes to the master repository.

Adjusting the ports of the servers using a Jython script

If you want to facilitate the port adjustment, you use the Jython script that we provided with the additional material of this book. We copied the Jython scripts directory, jscripts, to our working directory /u/cherr.

To adjust the ports of the servers using a Jython script:

1. Open an OMVS or telnet session.
2. Change to the working directory where your jscripts directory is located.

   ```
   cd /u/cherr
   ```

3. Execute the Jython script for the first application server with the appropriate values. For our cell it is:

   ```
   /wasv7config/b8cell/b8dmnode/DeploymentManager/profiles/default/bin/wsadmin.sh
   -lang jython -user b8admin -password b8admin -conntype RMI -port 28003 -f
   jscripts/updNewServerv7.py b8sr1_WPS.b8nodea.0 b8nodea 28060
   ```

4. Check the output of the script. In case of a misspelled server or node name, no ports are changed; however, the script shows a list of valid servers and nodes, as shown in Example 5-1.

Example 5-1 Sample output of Jython script "updNewServerv7.py"

```
B8ADMIN @
wtsc42:/wasv7config/b8cell/b8dmnode/DeploymentManager/profiles/default/bin>wsadmin
```

```
.sh -lang jython -conntype RMI -port 28003 -f /u/cherr/b8cell/updNewServerv7.py
b8sr1_WPS.b8nodea.0 b8nodea 28060
WASX7209I: Connected to process "dmgr" on node b8dmnode using RMI connector;  The
type of process is: DeploymentManager
WASX7303I: The following options are passed to the scripting environment and are
available as arguments that are stored in the argv variable:
"[b8sr1_WPS.b8nodea.0, b8nodea, 28060]"
Server name is  b8sr1_WPS.b8nodea.0
Node name  is  b8nodea
origin port number is  28060
BOOTSTRAP_ADDRESS  values before modification: host =  wtsc42.itso.ibm.com port =
2809
BOOTSTRAP_ADDRESS  values after modification: host =  wtsc42.itso.ibm.com port =
28063

SOAP_CONNECTOR_ADDRESS  values before modification: host =  wtsc42.itso.ibm.com
port =  8880
SOAP_CONNECTOR_ADDRESS  values after modification: host =  wtsc42.itso.ibm.com
port =  28062

WC_adminhost  values before modification: host =  * port =  9060
WC_adminhost  values after modification: host =  * port =  28065

WC_defaulthost  values before modification: host =  * port =  9080
WC_defaulthost  values after modification: host =  * port =  28067

DCS_UNICAST_ADDRESS  values before modification: host =  wtsc42.itso.ibm.com port
=  9353
DCS_UNICAST_ADDRESS  values after modification: host =  wtsc42.itso.ibm.com port =
28070

WC_adminhost_secure  values before modification: host =  * port =  9043
WC_adminhost_secure  values after modification: host =  * port =  28066

WC_defaulthost_secure  values before modification: host =  * port =  9443
WC_defaulthost_secure  values after modification: host =  * port =  28068

SIP_DEFAULTHOST  values before modification: host =  * port =  5060
SIP_DEFAULTHOST  values after modification: host =  * port =  28075

SIP_DEFAULTHOST_SECURE  values before modification: host =  * port =  5061
SIP_DEFAULTHOST_SECURE  values after modification: host =  * port =  28076

SIB_ENDPOINT_ADDRESS  values before modification: host =  * port =  7276
SIB_ENDPOINT_ADDRESS  values after modification: host =  * port =  28071

SIB_ENDPOINT_SECURE_ADDRESS  values before modification: host =  * port =  7286
SIB_ENDPOINT_SECURE_ADDRESS  values after modification: host =  * port =  28072

SIB_MQ_ENDPOINT_ADDRESS  values before modification: host =  * port =  5558
SIB_MQ_ENDPOINT_ADDRESS  values after modification: host =  * port =  28073

SIB_MQ_ENDPOINT_SECURE_ADDRESS  values before modification: host =  * port =  5578
SIB_MQ_ENDPOINT_SECURE_ADDRESS  values after modification: host =  * port =  28074
```

```
ORB_SSL_LISTENER_ADDRESS  values before modification: host =  * port =  0
ORB_SSL_LISTENER_ADDRESS  values after modification: host =  * port =  28064

ORB_LISTENER_ADDRESS  values before modification: host =  * port =  2809
ORB_LISTENER_ADDRESS  values after modification: host =  * port =  28063

Adding Host Alias entry for port number  28065
Adding Host Alias entry for port number  28066
Adding Host Alias entry for port number  28067
Adding Host Alias entry for port number  28068
Adding Host Alias entry for port number  28075
Adding Host Alias entry for port number  28076
Skipping node b8dmnode
Done with node b8nodea
All Done!
```

5. Repeat step 2 for all of your application servers with the appropriate values, as defined in the configuration spreadsheet.

The Jython script, updNewServerv7.py, also updates the ports of the host aliases of the virtual host default_host except on the web server ports, which must be adjusted manually in the ISC, as described in 5.3.4, “Adjusting the virtual hosts” on page 146.

There is also a sample JCL available in the additional material of this book for executing updNewServerv7.py, as shown in Example 5-2.

Example 5-2   Sample JCL for adjusting server ports

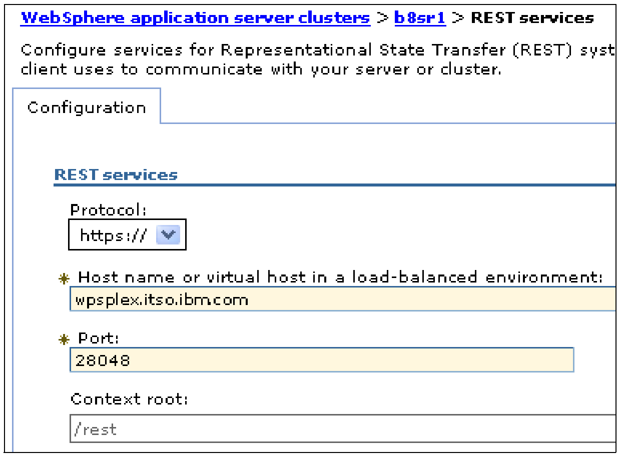
```
//B8ADJPOR JOB (ACCTNO,ROOM),'ADJ PORTS',CLASS=A,MSGCLASS=X,REGION=0M,
//         MSGLEVEL=(1,1),NOTIFY=&SYSUID
/*JOBPARM S=SC42
//*********************************************************/
//*  Make sure you edit this file with CAPS OFF.          */
//*  Run this job using an administrator userid.          */
//*********************************************************/
//*  Adjust server ports                                  */
//*********************************************************/
//* */
//ADJPORT EXEC PGM=IKJEFT01,REGION=0M,TIME=1440
//SYSTSPRT DD SYSOUT=*
//BPXOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//SYSTSIN  DD *
 BPXBATCH SH +
  export USER_INSTALL_ROOT=/wasv7config/b8cell/b8dmnode/+
  DeploymentManager/profiles/default/bin; +
  cd /u/cherr; +
  $USER_INSTALL_ROOT/wsadmin.sh +
   -lang jython +
   -user b8admin +
   -password b8admin +
   -conntype RMI +
   -port 28003 +
   -f jscripts/updNewServerv7.py +
   b8sr1_WPS.b8nodea.0 +
   b8nodea +
   28060;
//* */
//
```

5.3.3 Adjusting the REST services endpoints

Like the servers, the REST services endpoints are also assigned default port values. You must adjust these settings manually because they are not changed automatically when you change the server ports.

To adjust the REST services endpoints:

1. In the ISC, open the folders **Servers** → **Clusters**, and click **WebSphere application server clusters**.
2. Click your cluster. In the cluster detail view, click **REST services**.
3. In the REST services detail view, shown in Figure 5-15, change the host name to the dynamic VIPA and the port to the HTTPS port of the web servers.



WebSphere application server clusters > b8sr1 > REST services

Configure services for Representational State Transfer (REST) syst
client uses to communicate with your server or cluster.

REST services

Protocol: https://

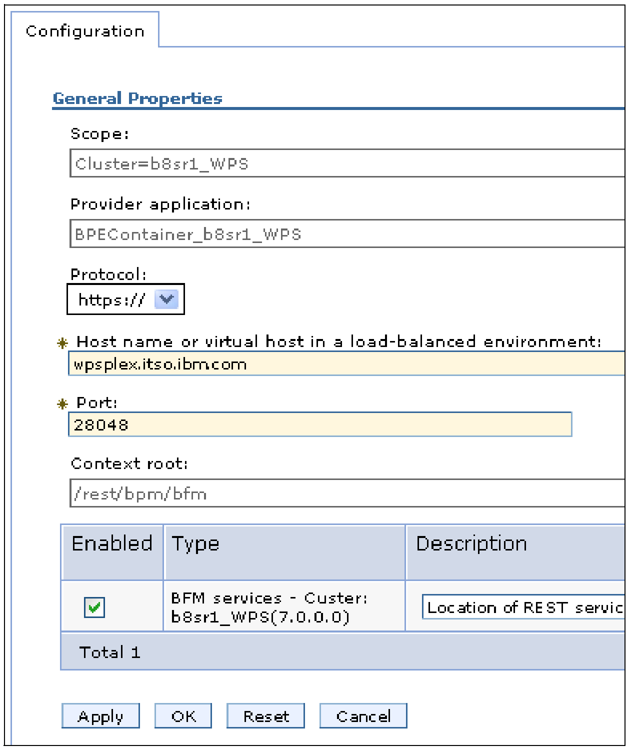
* Host name or virtual host in a load-balanced environment: wpsplex.itso.ibm.com

* Port: 28048

Context root: /rest

Figure 5-15 Adjust REST services endpoints

4. Click **OK**, and save the changes to the master repository.
5. Adjust the REST service endpoints for the business flow manager (BFM) and the human task manager (HTM). Open the folders **Services** → **REST services**, and click **REST service providers**.
6. Click **BPEContainer_b8sr1_WPS**.
7. In the detail view of the BPM REST service provider, shown in Figure 5-16 on page 145, enter the HTTPS port and distributed VIPA of your HTTP server.



Scope:
Cluster=b8sr1_WPS

Provider application:
BPEContainer_b8sr1_WPS

Protocol:
https://

* Host name or virtual host in a load-balanced environment:
wpsplex.itso.ibm.com

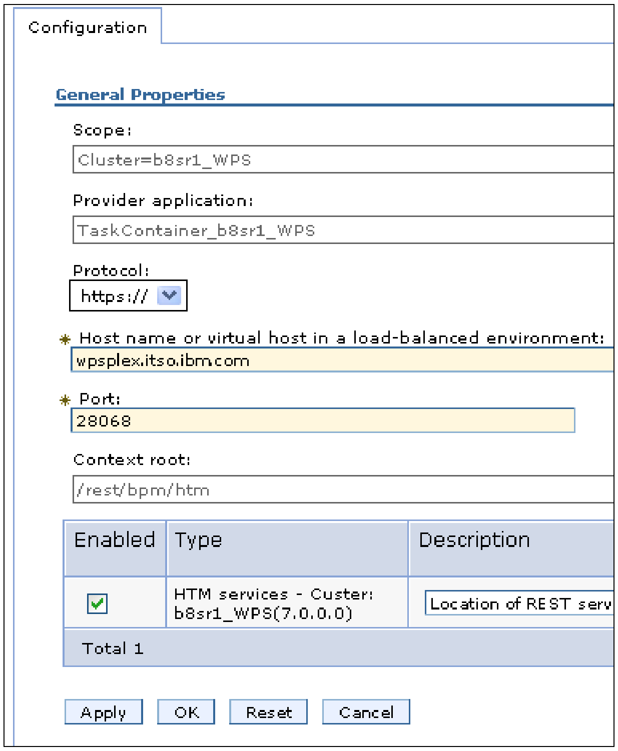
* Port:
28048

Context root:
/rest/bpm/bfm

Enabled	Type	Description
☑	BFM services - Custer: b8sr1_WPS(7.0.0.0)	Location of REST servic
Total 1		

Figure 5-16 BPM REST service provider details

8. Click **OK**, and then click **TaskContainer_b8sr1_WPS**.
9. In the detail view of the BPM REST service provider, Figure 5-17 on page 146, enter the HTTPS port and distributed VIPA of your HTTP server.



Scope:
Cluster=b8sr1_WPS

Provider application:
TaskContainer_b8sr1_WPS

Protocol:
https://

* Host name or virtual host in a load-balanced environment:
wpsplex.itso.ibm.com

* Port:
28068

Context root:
/rest/bpm/htm

Enabled	Type	Description
☑	HTM services - Custer: b8sr1_WPS(7.0.0.0)	Location of REST serv
Total 1		

Figure 5-17 HTM REST service provider details

10. Click **OK**, and save the changes to the master repository.

5.3.4 Adjusting the virtual hosts

The deployment environment wizards generates default ports for all of the servers that do not match to the ports that are defined in the configuration spreadsheet. It also adds ports to the virtual host, default_host. If you adjusted the port settings of your servers manually in the ISC, you must adjust the ports of the virtual host, default_host, now too.

If you used the Jython script, updNewServerv7.py, from the additional material of this book, the script already adjusted the virtual host, default_host; therefore, you only have to add the ports of the web servers to the default_host.

To adjust the virtual hosts:

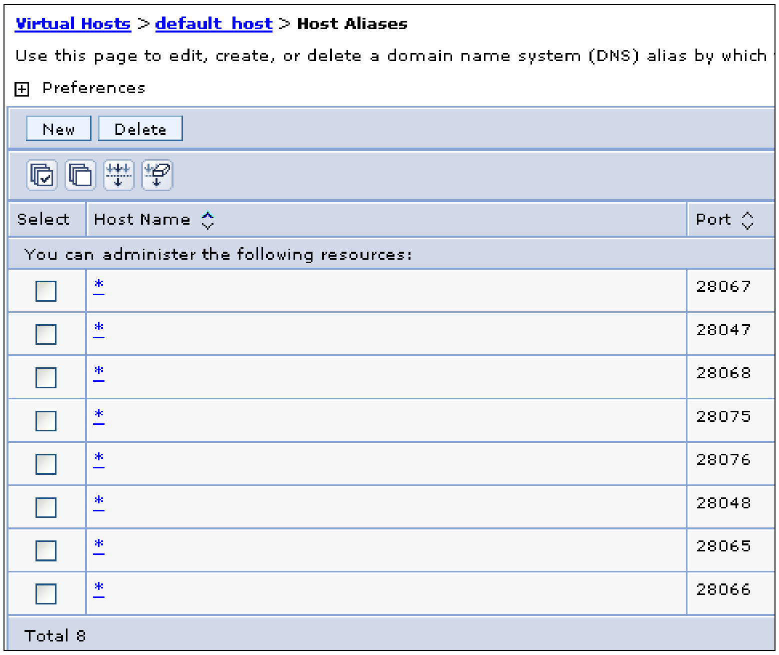
1. In the ISC, open the **Environment** folder, and click **Virtual hosts**.
2. Click **default_host** → **Host Aliases**. A list of host names and ports that are defined for the default_host are displayed. It will look similar to Figure 5-18 on page 147.
3. Replace the generated ports of the host aliases in the virtual host, default_host, by the appropriate ports as defined in your configuration spreadsheet. Click each host name, and change the port value. For our sample cell with one server configured, the adjustment looked like:
   - 9080 → 28067
   - 9443 → 28068
   - 9060 → 28065
   - 9043 → 28066

- 5060 → 28075
- 5061 → 28076

There are two more ports left now, if you have only one server configured in your cell and these two ports relate to the default web server ports. We will configure the web server in the cell in 5.4.2, “Configuring the web server” on page 160, but we can adjust the web server ports here according to the values in the configuration spreadsheet.

4. Replace the two remaining web server ports, for example:
   - 80 → 28047
   - 443 → 28048

Figure 5-18 shows the list of host aliases of the virtual host, default_host.



Virtual Hosts > default_host > Host Aliases

Use this page to edit, create, or delete a domain name system (DNS) alias by which

⊞ Preferences

Select	Host Name	Port
You can administer the following resources:		
□	*	28067
□	*	28047
□	*	28068
□	*	28075
□	*	28076
□	*	28048
□	*	28065
□	*	28066

Total 8

Figure 5-18 Adjusted host aliases of virtual host “default_host”

   If you have more than one application server or more than one web server, adjust the remaining ports accordingly.

5. When you are done, press **OK**, and save your changes to the master repository.

5.3.5 Adjusting the JVM custom properties of the servant processes

To get your cell working properly you must configure several JVM custom properties on the different process types of your application servers. You can adjust these custom properties manually in the ISC or you can use the Jython script, modjvmprops.py, which we provided with the additional material of this book.

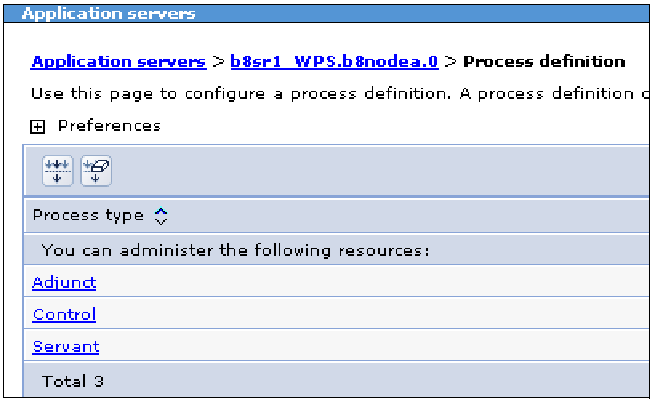
The following JVM custom properties must be configured for all process types of all application servers:

- com.ibm.ws.webservices.searchForAppServer: This custom property must be set to true on all process types to suppress the message `MetaDataLoade E loadWebContainerPorts could not find any http or https ports` in the server logs.

- db2.jcc.propertiesFile: This JVM custom property defines the DB2 location file to be used by the server and must be configured for all three process types. In our cell, we set it to /etc/d9fg/B7_DB2JccConfiguration.properties.
- com.ibm.ws390.management.proxy.ApplicationMgrMBeanProxy.zos.preFilterCheck: This JVM custom property must be configured to true on all process types of all application servers to allow the stopping and restarting of applications in a WebSphere Process Server for z/OS cell. Alternately, you can also define this property with the WebSphere environment variable IBM_JAVA_OPTIONS, such as `IBM_JAVA_OPTIONS = -Dcom.ibm.ws390.management.proxy.ApplicationMgrMBeanProxy.zos.preFilterCheck=true` on cell level, but we did not test it in our cell.

To configure the JVM custom properties manually:

1. In the ISC, open the **Servers** → **ServerTypes** folders, and click **WebSphere application servers**.
2. In the list of application servers, click the first WebSphere application server.
3. Open the **Java and Process Management** folder, and click **Process definition**. You will see now all three process types, Adjunct, Control and Servant, as shown in Figure 5-19, “Application server process types” on page 148. Click **Adjunct**.



Application servers

Application servers > b8sr1_WPS.b8nodea.0 > Process definition

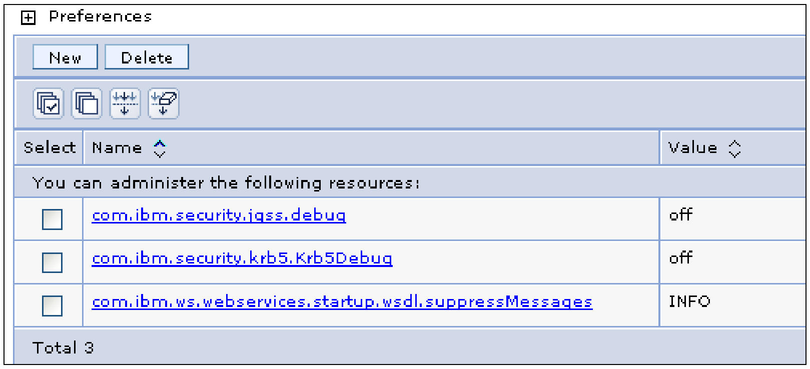
Use this page to configure a process definition. A process definition d

⊞ Preferences

Process type ⇕
You can administer the following resources:
Adjunct
Control
Servant
Total 3

Figure 5-19 Application server process types

4. Click **Java Virtual Machine** → **Custom properties**, and you will see a window with initial custom properties, as shown in Figure 5-20.

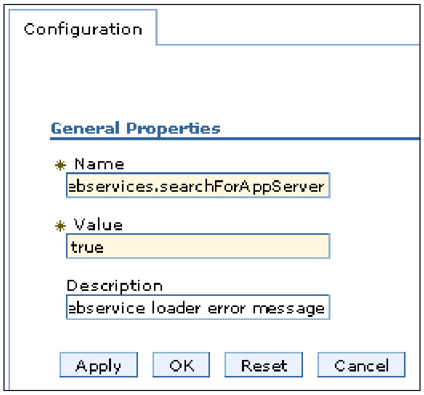


⊞ Preferences

Select	Name ⇕	Value ⇕
You can administer the following resources:		
□	com.ibm.security.jgss.debug	off
□	com.ibm.security.krb5.Krb5Debug	off
□	com.ibm.ws.webservices.startup.wsdl.suppressMessages	INFO
Total 3		

Figure 5-20 Initial JVM custom properties of adjunct

5. Click **New**, and enter the data for the first JVM custom property, as shown in Figure 5-21 on page 149.



* Name
ebservices.searchForAppServer

* Value
true

Description
ebservice loader error message

Figure 5-21 Create new JVM custom property

6. Press **OK**, and repeat step 4 for all three custom properties.
7. Return to the process type panel, and click the next process type. Repeat steps 3 through 5 for the other two process types.
8. When you are done with the first application server repeat steps 1 through 6 for the other application servers.
9. Save your changes to the master repository.

Adjusting the JVM custom properties using the Jython script

Using the provided Jython script facilitates the modification of JVM custom properties, especially if you must modify several properties for several servers for several process types. We copied the Jython scripts directory, jscripts, to our working directory /u/cherr.

To adjust the JVM custom properties using the Jython script:

1. Open an OMVS or telnet session.
2. Change to the working directory where your jscripts directory is located:

   ```
   cd /u/cherr
   ```

3. Execute the Jython script to configure all three JVM custom properties, as described at the beginning of this chapter, for all of the application servers in your cluster with the appropriate values. For our cell it is:

   ```
   /wasv7config/b8cell/b8dmnode/DeploymentManager/profiles/default/bin/wsadmin.sh
   -lang jython -user b8admin -password b8admin -conntype RMI -port 28003 -f
   jscripts/modjvmprops.py -server “b8sr1_WPS*” -props
   com.ibm.ws390.management.proxy.ApplicationMgrMBeanProxy.zos.preFilterCheck,true
   ;com.ibm.ws.webservices.searchForAppServer,true;db2.jcc.propertiesFile,/etc/d9f
   g/B7_DB2JccConfiguration.properties
   ```

4. The script displays all modifications that occurred during the processing, as shown in Example 5-3, and saves the changes if no error occurs during the processing.

Example 5-3 Sample output of Jython script modjvmprops.py

```
****  Processing JVM custom properties  ****

==> Processing JVM custom properties
com.ibm.ws390.management.proxy.ApplicationMgrMBeanProxy.zos.preFilterCheck,true;co
m.ibm.ws.webservices.searchForAppServer,t
```

```
  ==> Creating JVM custom property
com.ibm.ws390.management.proxy.ApplicationMgrMBeanProxy.zos.preFilterCheck of
Control process...
  <== JVM custom property created successfully.
  ==> Creating JVM custom property com.ibm.ws.webservices.searchForAppServer of
Control process...
  <== JVM custom property created successfully.
  ==> Creating JVM custom property db2.jcc.propertiesFile of Control process...
  <== JVM custom property created successfully.
  ==> Creating JVM custom property
com.ibm.ws390.management.proxy.ApplicationMgrMBeanProxy.zos.preFilterCheck of
Servant process...
  <== JVM custom property created successfully.
  ==> Creating JVM custom property com.ibm.ws.webservices.searchForAppServer of
Servant process...
  <== JVM custom property created successfully.
  ==> Creating JVM custom property db2.jcc.propertiesFile of Servant process...
  <== JVM custom property created successfully.
  ==> Creating JVM custom property
com.ibm.ws390.management.proxy.ApplicationMgrMBeanProxy.zos.preFilterCheck of
Adjunct process...
  <== JVM custom property created successfully.
  ==> Creating JVM custom property com.ibm.ws.webservices.searchForAppServer of
Adjunct process...
  <== JVM custom property created successfully.
  ==> Creating JVM custom property db2.jcc.propertiesFile of Adjunct process...
  <== JVM custom property created successfully.
<== Processing of JVM custom property in this scope completed.
==> Saving changes.
```

There is also a sample JCL for modifying JVM custom properties, as shown in Example 5-4.

Example 5-4 Sample JCL for modifying JVM custom properties

```
//B8JVMCP  JOB (ACCTNO,ROOM),'GEN GRANTS',CLASS=A,MSGCLASS=H,REGION=0M,
//         MSGLEVEL=(1,1),NOTIFY=&SYSUID
/*JOBPARM S=SC42
//***********************************************************/
//*  Make sure you edit this file with CAPS OFF.            */
//*  Run this job using an administrator userid.            */
//***********************************************************/
//*  Config JVM custom properties                           */
//***********************************************************/
//* */
//JVMCP EXEC PGM=IKJEFT01,REGION=0M,TIME=1440
//SYSTSPRT DD SYSOUT=*
//BPXOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//SYSTSIN  DD *
 BPXBATCH SH +
  export USER_INSTALL_ROOT=/wasv7config/b8cell/b8dmnode/+
  DeploymentManager/profiles/default/bin; +
  cd /u/cherr; +
  $USER_INSTALL_ROOT/wsadmin.sh +
   -lang jython +
   -user b8admin +
   -password b8admin +
   -conntype RMI +
   -port 28003 +
   -f jscripts/modjvmprops.py +
   -server 'b8sr1_WPS*' +
   -node '*' +
   -region '*' +
   -props 'com.ibm.ws390.management.proxy.ApplicationMgrMBeanProxy+
   .zos.preFilterCheck,true;com.ibm.ws.webservices.searchForApp+
   Server,true;db2.jcc.propertiesFile,/etc/d9fg/B8_DB2JccConfig+
   uration.properties' +
   -mode S;
//* */
//
```

5.3.6 Adjusting the custom properties of the datasources

For most of the configured datasources various custom properties, namely the currentSchema and currentSQLID properties, require modifications. Best practice for the datasources is to use currentSchema to address the correct database artifacts. When using currentSQLID, the user of the authentication alias must have special permissions to switch to the currentSQLID, which usually is not wanted to be granted to a user. Some of the datasources have the currentSQLID configured instead of the currentSchema.

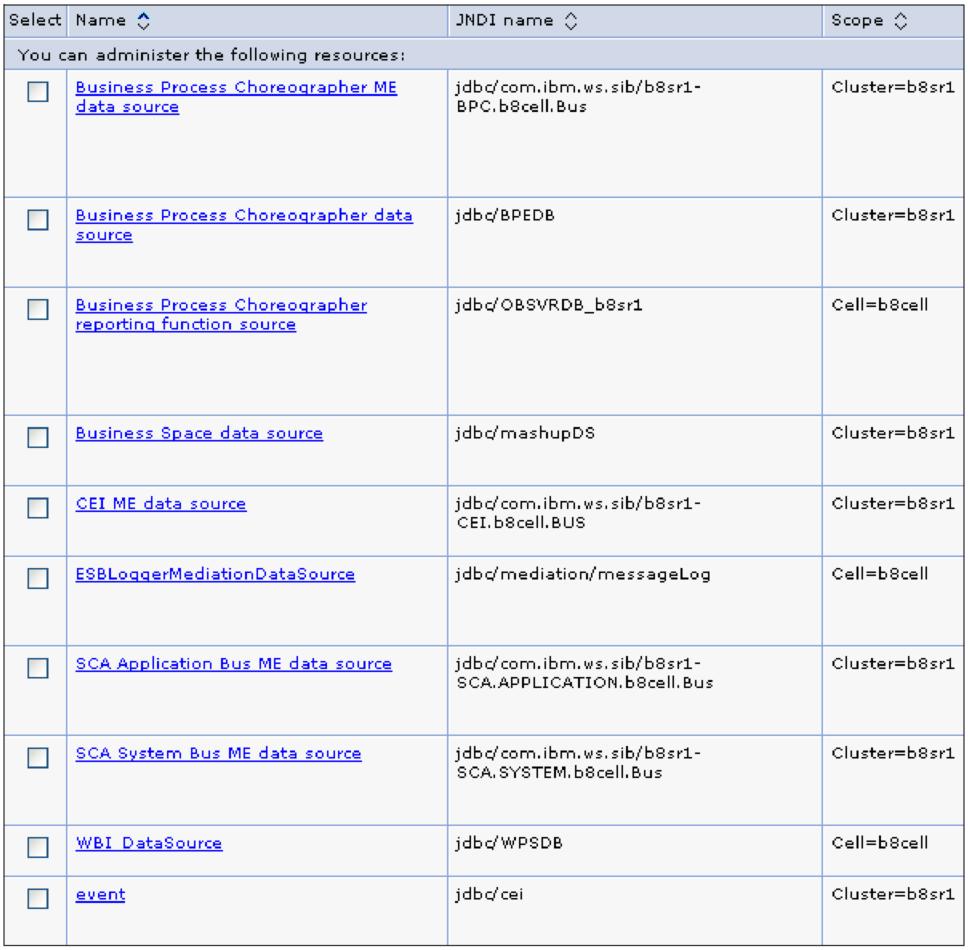
For the datasources of the messaging engines there is a special rule. Because they have the schema name defined with the data store of the messaging engine, both currentSQLID and currentSchema should be empty.

In this section, we describe how to change the currentSQLID and currentSchema custom properties of the datasources manually in the ISC. You can also use a Jython script, adjustDS.py, which is provided with the additional material of this book, to adjust these custom properties automatically.

Adjusting the custom properties of the datasources manually

To adjust the custom properties of the data source manually:

1. In the ISC, click **Resources** → **JDBC** → **Data sources**. A list of datasources is displayed, as shown in Figure 5-22 on page 152.



Select	Name	JNDI name	Scope
You can administer the following resources:			
□	Business Process Choreographer ME data source	jdbc/com.ibm.ws.sib/b8sr1-BPC.b8cell.Bus	Cluster=b8sr1
□	Business Process Choreographer data source	jdbc/BPEDB	Cluster=b8sr1
□	Business Process Choreographer reporting function source	jdbc/OBSVRDB_b8sr1	Cell=b8cell
□	Business Space data source	jdbc/mashupDS	Cluster=b8sr1
□	CEI ME data source	jdbc/com.ibm.ws.sib/b8sr1-CEI.b8cell.BUS	Cluster=b8sr1
□	ESBLoggerMediationDataSource	jdbc/mediation/messageLog	Cell=b8cell
□	SCA Application Bus ME data source	jdbc/com.ibm.ws.sib/b8sr1-SCA.APPLICATION.b8cell.Bus	Cluster=b8sr1
□	SCA System Bus ME data source	jdbc/com.ibm.ws.sib/b8sr1-SCA.SYSTEM.b8cell.Bus	Cluster=b8sr1
□	WBI_DataSource	jdbc/WPSDB	Cell=b8cell
□	event	jdbc/cei	Cluster=b8sr1

Figure 5-22 List of data sources

2. Click the first data source.
3. In the detail view of the data source, click **Custom properties**.
4. Locate and click the property **currentSchema**.
5. If it is a data source used by a messaging engine, remove a possibly configured value; otherwise, enter the schema value as defined in the configuration spreadsheet. A data source for a messaging engine can be identified by having either SCA or messaging in the description of the data source. Click **OK**.
6. Locate and click the property **currentSQLID**. Remove a possibly configured value, and click **OK**.
7. Return to the data source list, and repeat steps 3 through 8 for the next data source in the list until all data sources are processed.
8. Save the changes to the master repository.

Adjusting the custom properties of the datasources with a Jython script

To facilitate the modification of the custom properties of the datasources you can use the provided Jython script, adjustDS.py. We copied the Jython script to our working directory /u/cherr/b8cell.

To adjust the custom properties of the data source using a Jython script:

1. Open an OMVS or telnet session.
2. Change to the working directory where your jscripts directory is located:

   ```
   cd /u/cherr
   ```

3. Set the WebSphere environment:

   ```
   ./wasv7config/b8cell/b8dmnode/DeploymentManager/profiles/default/bin/setupCmdLi
   ne.sh
   ```

   Make sure that you include the period (.) in front of the command.

4. Execute the Jython script to configure the custom properties currentSchema and currentSQLID of all datasources except the datasources of the messaging engines. For our cell it is:

   ```
   /wasv7config/b8cell/b8dmnode/DeploymentManager/profiles/default/bin/wsadmin.sh
   -lang jython -user b8admin -password b8admin -conntype RMI -port 28003 -f
   jscripts/adjustDS.py -schema B8CELL -exclme
   ```

5. The script displays all modifications that occurred during the processing, as shown in Example 5-5, and saves the changes if no error occurs during the processing.

Example 5-5 Sample output of Jython script adjustDS.py

```
!! Datasources of messaging engines are not modified.
o  Datasource <Business Process Choreographer ME data source>
  --> currentSchema is not set.
  --> currentSQLID is not set.
o  Datasource <Business Process Choreographer data source>
  --> currentSchema is not set.
  ==> will now be changed to B8CELL...
  <== currentSchema changed successfully
  --> currentSQLID is not set.
o  Datasource <Business Process Choreographer reporting function source>
  --> currentSchema is not set.
  ==> will now be changed to B8CELL...
  <== currentSchema changed successfully
  --> currentSQLID is not set.
o  Datasource <Business Space data source>
  --> currentSchema is not set.
  ==> will now be changed to B8CELL...
  <== currentSchema changed successfully
  --> currentSQLID is set to B8CELL
  ==> will now be cleared...
  <== currentSQLID cleared successfully
o  Datasource <CEI ME data source>
  --> currentSchema is not set.
  --> currentSQLID is not set.
o  Datasource <SCA Application Bus ME data source>
  --> currentSchema is not set.
  --> currentSQLID is not set.
o  Datasource <SCA System Bus ME data source>
```

```
  --> currentSchema is not set.
  --> currentSQLID is not set.
o  Datasource <ESBLoggerMediationDataSource>
  --> currentSchema is not set.
  ==> will now be changed to B8CELL...
  <== currentSchema changed successfully
  --> currentSQLID is set to B8CELL
  ==> will now be cleared...
  <== currentSQLID cleared successfully
o  Datasource <WBI_DataSource>
  --> currentSchema is not set.
  ==> will now be changed to B8CELL...
  <== currentSchema changed successfully
  --> currentSQLID is set to B8CELL
  ==> will now be cleared...
  <== currentSQLID cleared successfully
o  Datasource <event>
  --> currentSchema is not set.
  ==> will now be changed to B8CELL...
  <== currentSchema changed successfully
  --> currentSQLID is set to B8CELL
  ==> will now be cleared...
  <== currentSQLID cleared successfully
<==  All datasources processed.
==> Saving changes now...
<==  Changes saved successfully.
```

There is also a sample JCL for modifying data source custom properties, as shown in Example 5-4 on page 150.

Example 5-6 Sample JCL

```
//B8ADJDS JOB (ACCTNO,ROOM),'ADJUST DS',CLASS=A,MSGCLASS=X,REGION=0M,
//        MSGLEVEL=(1,1),NOTIFY=&SYSUID
/*JOBPARM S=SC42
//*******************************************************/
//*  Make sure you edit this file with CAPS OFF.          */
//*  Run this job using an administrator userid.          */
//*  Adjust resource properties of WPS datasources        */
//*******************************************************/
//* */
//ADJDS EXEC PGM=IKJEFT01,REGION=0M,TIME=1440
//SYSTSPRT DD SYSOUT=*
//BPXOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//SYSTSIN  DD *
 BPXBATCH SH +
  export USER_INSTALL_ROOT=/wasv7config/b8cell/b8dmnode/+
  DeploymentManager/profiles/default/bin; +
  cd /u/cherr; +
  $USER_INSTALL_ROOT/wsadmin.sh +
   -lang Jython +
   -user b8admin +
   -password b8admin +
   -conntype RMI +
   -port 28003 +
   -f jscripts/adjustDS.py +
   -schema B8CELL +
   -exclme;
//* */
//
```

5.3.7 Configuring peer recovery and transaction XA partner logs

WebSphere provides a component called the High Availability Manager (HA manager) for monitoring services, such as transaction managers, messaging engines, and workload managers. This component is also responsible for the automatic peer recovery of both indoubt and inflight transactions for any server that fails in the defined HA cluster. An *indoubt transaction* is any transaction that becomes stuck in the indoubt state indefinitely because of an exceptional circumstance, such as the removal of a node causing messaging engines to be destroyed. An indoubt transaction state results after the database finishes its phase 1 commit processing and before it starts phase 2. An *inflight transaction* is a transaction that has not yet completed the prepare phase of the commit process and where the transaction or message is persisted somewhere that it can be recovered. The automatic recovery functionality that the HA Manager performs enables the cluster to rebalance itself if one or more cluster members fail.

To enable automatic peer recovery the following tasks must be done:

1. Configure transaction XA partner logs of the servers in a way that they can be accessed by any other server in the cluster.
2. Enable the high availability function for the cluster.
3. Configure security in RACF to allow the servers to call the ATRSRV macro.
4. Enable automatic peer recovery mode.

Configuring transaction XA partner logs

For peer recovery, you need a reliable and accessible transaction log for each server, which you can achieve either using files in a shared HFS or using z/OS log streams. The latter is the preferred solution for WebSphere on z/OS.

Before you configure WebSphere using log streams as transaction XA partner logs, you must create these log streams. Each server needs a pair of log streams that have fixed names of the following naming schema:

- <<HLQ>>.<server>.M
- <<HLQ>>.<server>.D

The only variable part in this naming schema is <<HLQ>>, which is also used later on for configuring the log stream usage in WebSphere. In our cell, we used the cell name as HLQ so that the log stream names are:

- B8CELL.B8SR1A.M
- B8CELL.B8SR1A.D

The JCL sample in Example 5-7 on page 156 shows how to create the log streams for two servers. Notice that the log structure is sized to hold four log streams, two for each of 2 cluster members. Adjust the LOGSNUM statement if your cluster comprises more than two cluster members. Notice that the sample JCL creates a model log stream called B8CELL.B8SR1A.X which is then used as a template for creating the log streams for the servers.

Important: To avoid trouble when using log streams, make sure that AUTODELETE(NO) and RETPD(0) are set for the log streams.

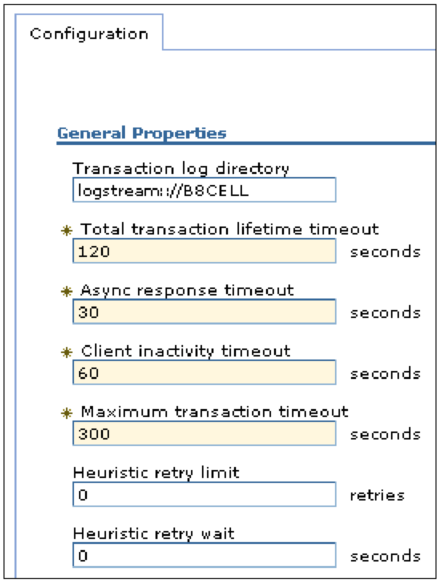
Example 5-7 Sample JCL for creating log streams

```
//B7LOGSTR JOB (0)'WSADMIN',CLASS=S,MSGCLASS=R
//*
//STEP1    EXEC PGM=IXCMIAPU
//SYSPRINT DD   SYSOUT=*
//SYSABEND DD   SYSOUT=*
//SYSIN    DD   *
DATA TYPE(LOGR) REPORT(YES)
     DEFINE STRUCTURE NAME(B8_WPS_XALOGS)
                      LOGSNUM(4)
                      MAXBUFSIZE(65532)
                      AVGBUFSIZE(14000)
     DEFINE LOGSTREAM NAME(B8CELL.B8SR1.X)
                      STRUCTNAME(B8_WPS_XALOGS)
                      MODEL(YES)
                      STG_DUPLEX(YES)
                      DUPLEXMODE(COND)
                      LOGGERDUPLEX(COND)
                      LS_SIZE(5000)
                      STG_DATACLAS(IXSTAGE)
                      STG_SIZE(6400)
                      AUTODELETE(NO)
                      RETPD(0)
                      HLQ(IXGLOGR)
                      LOWOFFLOAD(0)
                      HIGHOFFLOAD(80)
     DEFINE LOGSTREAM NAME(B8CELL.B8SR1A.D)
                      LIKE(B8CELL.B8SR1.X)
     DEFINE LOGSTREAM NAME(B8CELL.B8SR1A.M)
                      LIKE(B8CELL.B8SR1.X)
     DEFINE LOGSTREAM NAME(B8CELL.B8SR1B.D)
                      LIKE(B8CELL.B8SR1.X)
     DEFINE LOGSTREAM NAME(B8CELL.B8SR1B.M)
                      LIKE(B8CELL.B8SR1.X)
```

If the log streams are created, you can configure your server to use the created log streams as transaction XA partner logs.

To configure transaction XA partner logs:

1. In the ISC, click **Servers** → **Server types** → **WebSphere application servers**.
2. Click the first application server in the list.
3. In the server detail view, go to the Container Settings section, and click **Container Services** → **Transaction Service**.
4. In the Transaction log directory field, enter the HLQ of your z/OS log stream that the server uses, for example, `logstream://B8CELL`, as shown in Figure 5-23 on page 157.



General Properties

Transaction log directory
logstream://B8CELL

* Total transaction lifetime timeout
120 seconds

* Async response timeout
30 seconds

* Client inactivity timeout
60 seconds

* Maximum transaction timeout
300 seconds

Heuristic retry limit
0 retries

Heuristic retry wait
0 seconds

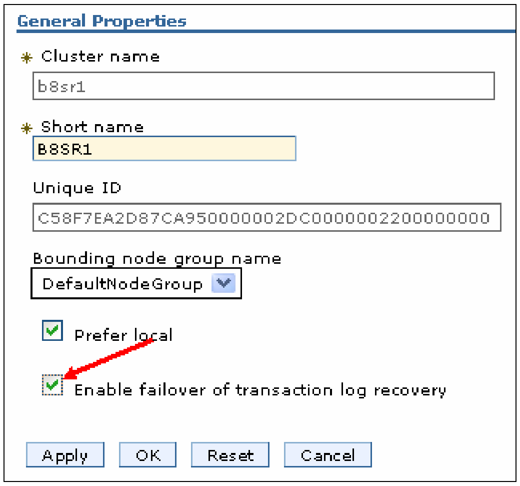
Figure 5-23 Setting the log stream name

5. Click **OK**, and return to the list of application servers.
6. Repeat step 2 through 6 for each of your cluster members.
7. Save the changes to the master repository.

Enabling the high-availability function

Per default, the high-availability function is disabled for all clusters. To enable this function you must select it in the clusters detail view:

1. In the ISC, click **Servers** → **Clusters** → **WebSphere application server clusters**.
2. Click the cluster in the cluster list, as shown in Figure 5-23.


General Properties

* Cluster name
b8sr1

* Short name
B8SR1

Unique ID
C58F7EA2D87CA950000002DC0000002200000000

Bounding node group name
DefaultNodeGroup

☑ Prefer local

☑ Enable failover of transaction log recovery

Figure 5-24 Enabling high availability function for a cluster

3. Click **OK**, and save the changes to the master repository.

Configuring security for calling macro ATRSRV

To get peer recovery working you must configure RACF to allow the application servers to call the ATRSRV macro.

The ATRSRV macro allows a server to commit and back out transactions for other servers. This process differs from peer restart and recovery support, where the other server is started on another system. The ATRSRV macro is provided by MVS Resource Recovery Services (RRS).

The user ID that the application server controller region runs under must have ALTER access to the MVSADMIN.RRS.COMMANDS.gname.sysname resource in the FACILITY class, where gname is the RRS logging group (usually the SYSPLEX name) and sysname is the system name. To allow access to all logging groups and systems, use wildcards in the resource name, for example MVSADMIN.RRS.COMMANDS.*.

Note: Because the controller region runs as an authorized address space, it implicitly has ALTER access to this resource class, unless the RACF configuration explicitly restricts access. By explicitly allowing access to this resource, you are not relying on the authorized state of the controller region.

Enabling automatic peer recovery mode

Peer recovery can be done either automatically or manually. In automatic mode, the HA manager automatically assigns the recovery of a failed server to another member in the cluster, and this member is recovering the outstanding transaction of the failed server. If this is not desired, you must configure peer recovery in manual mode. Then you must specify and trigger in the ISC which server will perform recover for the failed server. Per default, the recovery mode is configured to automatic, so you do not have to do extra configuration work here. More information about configuring the mode of the peer recovery is in the InfoCenter for WebSphere Application Server for z/OS.

5.3.8 Synchronizing the nodes and resuming auto-synchronization

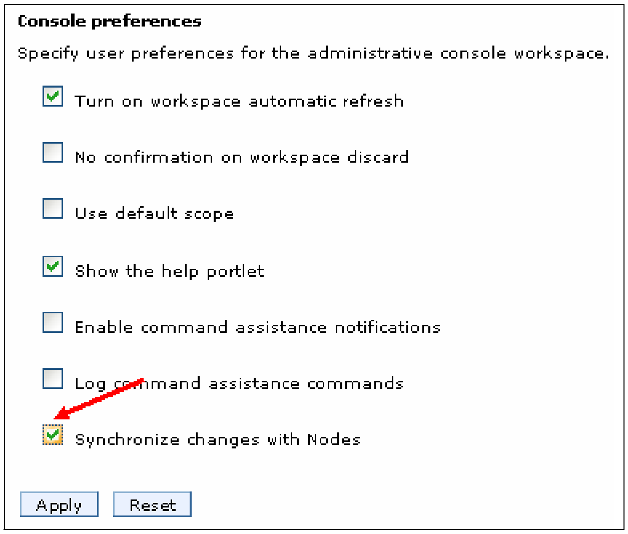
So far we have not synchronized the configuration in master repository to the nodes. To synchronize the nodes:

1. In the ISC, click folder **System administration** → **Nodes**.
2. Select all nodes by clicking ☑.
3. Click Full Resynchronize.

The synchronization process is started and might take a while to complete. When the nodes are synchronized you should see the icon ⇔ in the Status column.

If you want to have your configuration changes automatically synchronized with the nodes when you save them to the master repository, you must enable auto-synchronization.

1. In the System administration folder, click **Console Preferences**.
2. You will get a list of console options, as shown in Figure 5-25 on page 159. Click **Synchronize changes with nodes**, and click **Apply**.



Console preferences

Specify user preferences for the administrative console workspace.

☑ Turn on workspace automatic refresh

□ No confirmation on workspace discard

□ Use default scope

☑ Show the help portlet

□ Enable command assistance notifications

□ Log command assistance commands

☑ Synchronize changes with Nodes

Figure 5-25 Console preferences

Now the master repository will always be automatically synchronized with the nodes when you save changes to it.

5.4 Configuring HTTP servers or proxies

When you have a cluster, it is usually necessary to define one or more HTTP Servers or some kind of proxy in front of the cluster. The HTTP Servers run the WebSphere plug-in, which is responsible for distributing work and maintaining affinity between a client and a cluster member. Alternatively, you can use the on-demand router (ODR) that is part of WebSphere Extended Deployment for z/OS Virtual Enterprise, or perhaps use some other third-party proxy. The IBM White Paper PRS2663, *WebSphere z/OS - Comparing Front End HTTP Options*, can help you to understand the differences between these solutions. You can obtain that paper from the following web site:

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

When you run HTTP servers on z/OS you have a choice between the older IBM HTTP Server for z/OS (based on Lotus® Domino® Go technology) and the newer IBM HTTP Server (based on Apache). You can also choose to run the HTTP Server on some other platform. We configured the IBM HTTP Server for z/OS based on Apache. Hereafter, we refer to this HTTP Server as IBM HTTP Server.

HTTP Servers were configured on the LPARs hosting the nodes of our b7cell cell. We do not include step-by-step instructions about configuring an HTTP Server in this publication. Instead, we describe the two z/OS HTTP Server options and provide references to existing step-by-step guides for each of these options. We then describe how to configure the web server in your cell.

5.4.1 Overview of IBM HTTP servers for z/OS

In this section, we introduce IBM HTTP Servers.

IBM HTTP Server for z/OS powered by Apache

The newer HTTP Server is shipped as one component of the WebSphere Application Server for z/OS optional materials and is usually found in the file system at /usr/lpp/zWebSphere_OM/V6R1/HTTP/Server. The IBM HTTP Server for z/OS Powered by Apache is FMID JIWO610, Component IDs 5655I3510 and 5655I3511. IBM White Paper WP101170, *The IBM HTTP Server for z/OS Powered by Apache*, describes how to configure. You can download that White Paper from:

http://www.ibm.com/support/techdocs

The IHS worksheet on the planning spreadsheet provides a step-by-step guide for configuring an IBM HTTP Server for each node. This worksheet refers to some sample jobs that are provided on the JCL_P and JCL_S worksheets, which do everything for you.

The service level of the HTTP Server powered by Apache must be kept in step with the underlying WebSphere Application Server for z/OS. You can find details of the corresponding service levels at the following web page:

http://www.ibm.com/support/docview.wss?&uid=swg27009131

IBM HTTP Server for z/OS (LGW)

The IBM HTTP Server for z/OS (LGW) has been shipped as part the z/OS base operating system for many years. There are three FMIDs:

- HIMW530
- JIMW53B
- JIMW531

The LGW is typically found in the file system at /usr/lpp/internet.

You can find a step-by-step guide to configuring the IBM HTTP Server for z/OS (LGW) in the IBM Redpaper *WebSphere Process Server for z/OS: Configuring a Network Deployment Environment*, REDP-4388. Although that paper discusses WebSphere Process Server for z/OS V6.0.2, the instructions are largely the same for V7.0.

5.4.2 Configuring the web server

The IHS worksheet in the planning spreadsheet provides step-by-step instructions for configuring an IBM HTTP Server for each node. The IHS worksheet guides you through the execution of several sample jobs, provided on the JCL_P worksheet, which configure an IBM HTTP Server to be managed by the primary node. These jobs are listed in this section. The same jobs are provided on the JCL_S worksheet in order to configure an IBM HTTP Server managed by a secondary node.

The sample jobs are:

@IHSDSA Creates an intermediate symbolic link to the IBM HTTP Server product.

@IHSINA Creates the configuration for the web server B8IHSA under directory /wasv7config/b8cell/b8ihsa.

@IHSP1A Creates an installation directory for the WebSphere HTTP plug-in.

@IHSP2A Installs the plug-in into B8IHSA.

B8IHSA Starts task JCL for the web server.

@IHSSTC Defines a RACF STARTED profile that associates the controller user ID, B8ACRU, with the web server started task.

@IHSWSA Adds the web server to managed node b8nodea.

If you already configured one or more IBM HTTP Servers using other documentation, the only step you must perform is to add the web server to the WebSphere Application Server for z/OS configuration so it can be managed from the deployment manager. You can do this by editing job @IHSWSA to match the names and paths used by your existing IBM HTTP Server(s), but you might find it easier to add the web server into WebSphere Application Server for z/OS using the Integrated Solutions Console.

The following steps describe how to use the Integrated Solutions Console to add an IBM HTTP Server that is already configured to the cell. The HTTP server can then be managed from your Integrated Solutions Console using the node agents:

1. Click **Server Types** → **Web servers**, and then click **New**.

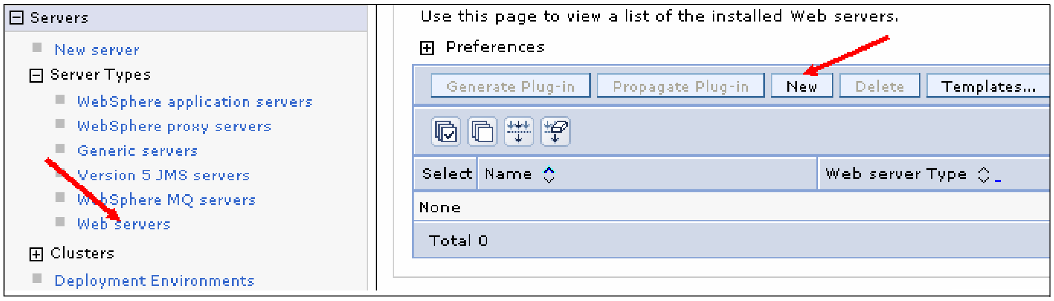
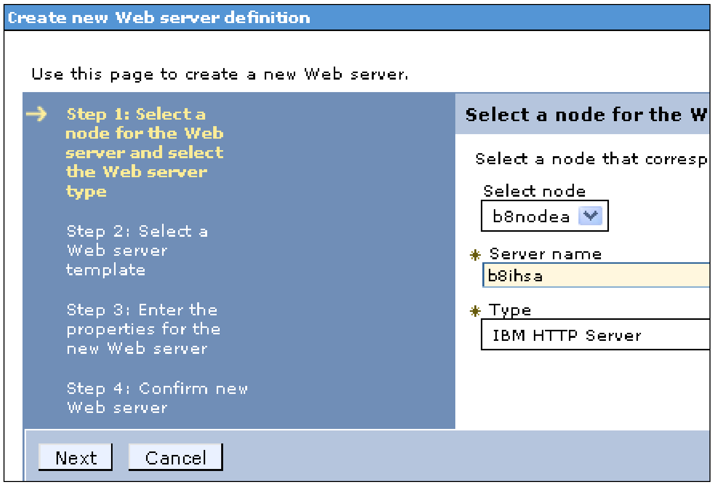


Figure 5-26 Configure new Web server

2. Select the appropriate node where the web server is running on, enter the name of the web server, and select **IBM HTTP Server** from the pull-down list, as shown in Figure 5-27. Click **Next**.


Create new Web server definition
Use this page to create a new Web server.
Step 1: Select a node for the Web server and select the Web server type
Step 2: Select a Web server template
Step 3: Enter the properties for the new Web server
Step 4: Confirm new Web server
Select a node for the W
Select a node that corresp
Next
Cancel

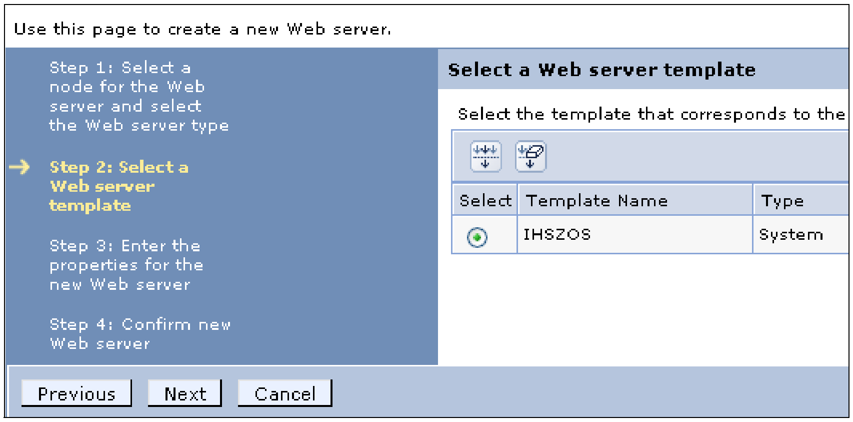
Select node: b8nodea

* Server name: b8ihsa

* Type: IBM HTTP Server

Figure 5-27 Web server name

3. Figure 5-28 on page 162 shows the template that is used for the web server configuration. Click **Next** again.


Use this page to create a new Web server.

Step 1: Select a node for the Web server and select the Web server type
Step 2: Select a Web server template
Step 3: Enter the properties for the new Web server
Step 4: Confirm new Web server
Previous
Next
Cancel

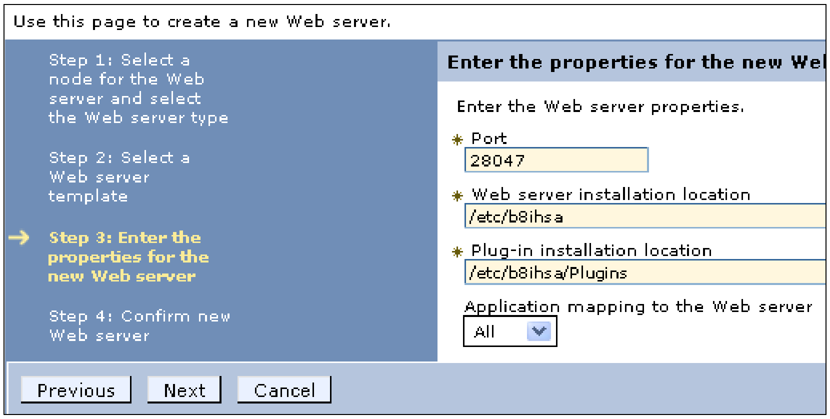
Select a Web server template

Select the template that corresponds to the

Select	Template Name	Type
●	IHSZOS	System

Figure 5-28 Web server template

4. In Figure 5-29, enter the port, the web server, and the WebSphere plug-in installation directories. Click **Next**.


Use this page to create a new Web server.

Step 1: Select a node for the Web server and select the Web server type
Step 2: Select a Web server template
Step 3: Enter the properties for the new Web server
Step 4: Confirm new Web server
Previous
Next
Cancel

Enter the properties for the new Wel

Enter the Web server properties.

* Port
28047

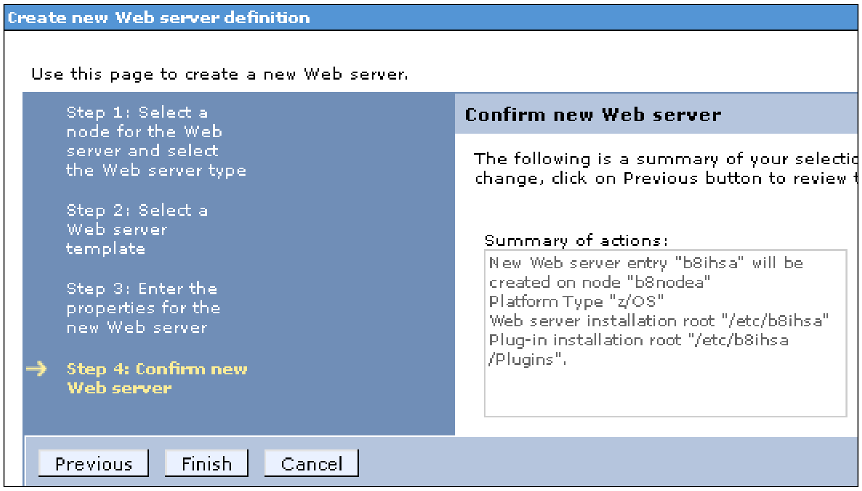
* Web server installation location
/etc/b8ihsa

* Plug-in installation location
/etc/b8ihsa/Plugins

Application mapping to the Web server
All

Figure 5-29 Web server port and installation directory settings

5. Verify the settings on the summary page again, as shown in Figure 5-30, and click **Finish**.


Create new Web server definition

Use this page to create a new Web server.

Step 1: Select a node for the Web server and select the Web server type
Step 2: Select a Web server template
Step 3: Enter the properties for the new Web server
Step 4: Confirm new Web server
Previous
Finish
Cancel

Confirm new Web server

The following is a summary of your selectio
change, click on Previous button to review t

Summary of actions:
New Web server entry "b8ihsa" will be created on node "b8nodea"
Platform Type "z/OS"
Web server installation root "/etc/b8ihsa"
Plug-in installation root "/etc/b8ihsa/Plugins".

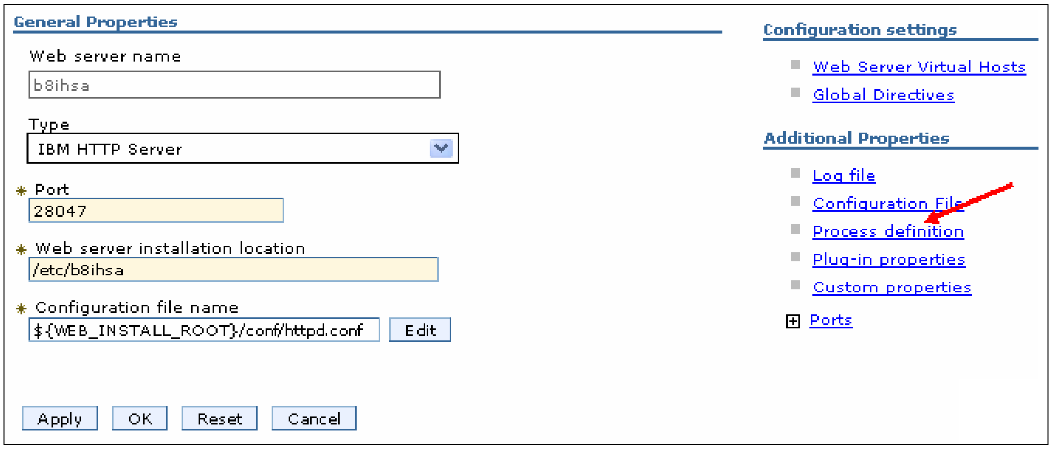
Figure 5-30 Web server configuration summary

6. After the web server configuration process completes successfully, click **Save** to save the configuration changes to the master repository.

5.4.3 Modifying the web server configuration

After the successful configuration of the web server, you must modify the web server configuration because some of the settings do not work on z/OS. To modify the web server configuration:

1. Select your web server, and click **Process definition**, as shown in Figure 5-31.



General Properties

Web server name: b8ihsa

Type: IBM HTTP Server

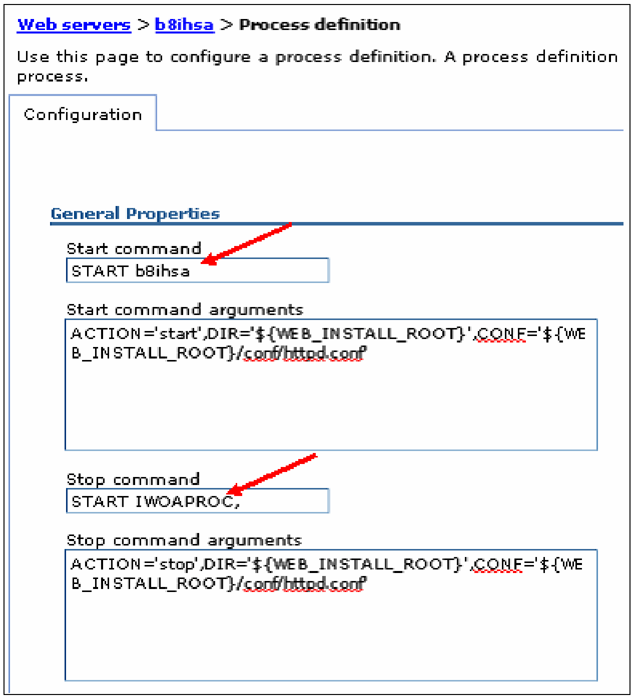
* Port: 28047

* Web server installation location: /etc/b8ihsa

* Configuration file name: ${WEB_INSTALL_ROOT}/conf/httpd.conf

Figure 5-31 Web server process definition

2. Replace IWOAPROC in the Start and Stop command fields by your web server name, and remove the comma (,) at the end, as shown in Figure 5-32.



Web servers > b8ihsa > Process definition

Use this page to configure a process definition. A process definition process.

General Properties

Start command: START b8ihsa

Start command arguments:

```
ACTION='start',DIR='${WEB_INSTALL_ROOT}',CONF='${WE
B_INSTALL_ROOT}/conf/httpd.conf'
```

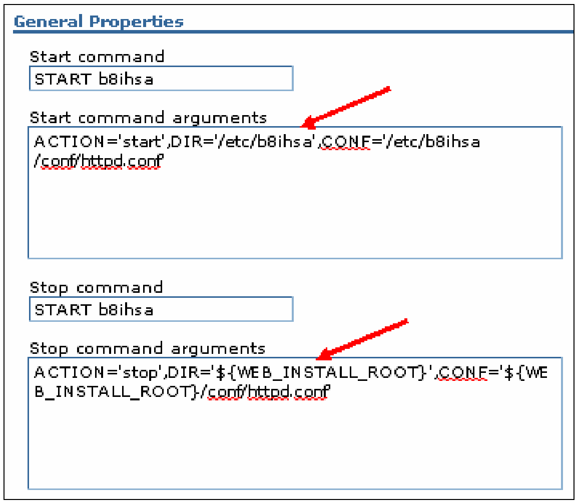
Stop command: START IWOAPROC,

Stop command arguments:

```
ACTION='stop',DIR='${WEB_INSTALL_ROOT}',CONF='${WE
B_INSTALL_ROOT}/conf/httpd.conf'
```

Figure 5-32 Invalid Web server process definitions

3. In the Start and Stop command arguments fields, replace the WebSphere variable ${WEB_INSTALL_ROOT} with the real path name that you used with the web server configuration, as shown in Figure 5-33.



General Properties

Start command
START b8ihsa

Start command arguments
ACTION='start',DIR='/etc/b8ihsa',CONF='/etc/b8ihsa/conf/httpd.conf'

Stop command
START b8ihsa

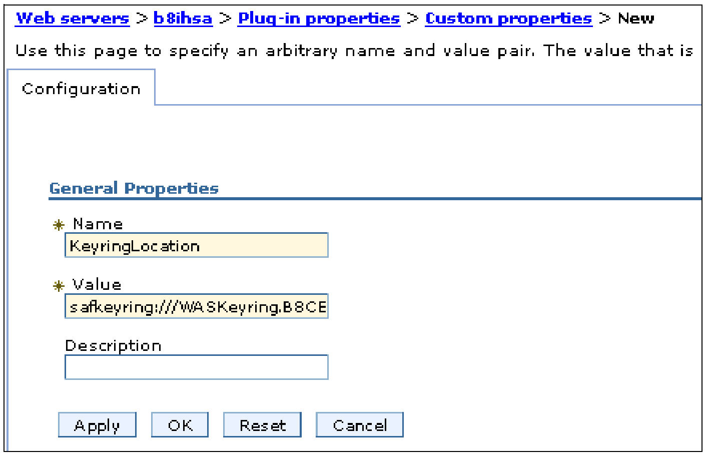
Stop command arguments
ACTION='stop',DIR='${WEB_INSTALL_ROOT}',CONF='${WEB_INSTALL_ROOT}/conf/httpd.conf'

Figure 5-33 Adjusting command arguments

4. Define that SAF is used instead of files in the filesystem as the key repository. In the web server detail view, click **Custom properties** → **Plug-in properties**.

5. Here you must define two new custom properties. Click **New**.

6. In the Name field, enter `KeyringLocation`, and in the Value field, type `safkeyring:///WASKeyring.B8CELL`, as shown in Figure 5-34.



Web servers > b8ihsa > Plug-in properties > Custom properties > New

Use this page to specify an arbitrary name and value pair. The value that is

General Properties

* Name
KeyringLocation

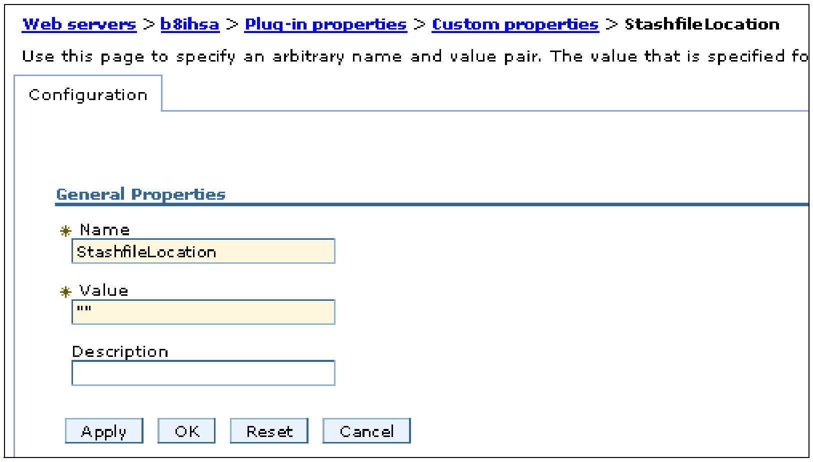
* Value
safkeyring:///WASKeyring.B8CE

Description

Figure 5-34 New custom property KeyringLocation

7. Click **OK** → **New**.

8. In the Name and Value fields, type `StashfileLocation`, as shown in Figure 5-35 on page 165.



Web servers > b8ihsa > Plug-in properties > Custom properties > StashfileLocation

Use this page to specify an arbitrary name and value pair. The value that is specified fo

General Properties

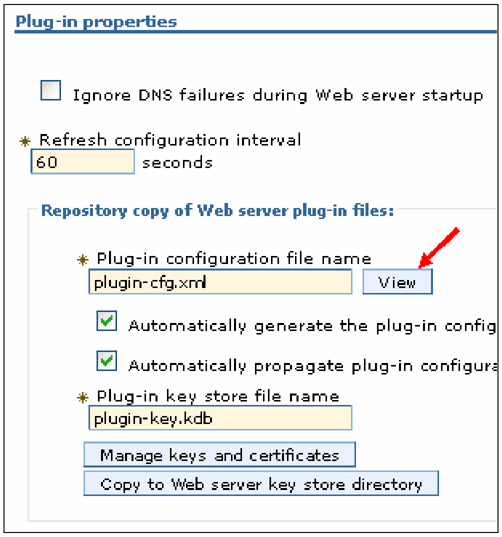
* Name
StashfileLocation

* Value
""

Description

Figure 5-35 Custom property StashfileLocation

9. Click **OK**, save your changes to the master repository.
10. Return to the web server list, select the web server, and click **Generate plug-in**.
11. After successful generation of the plug-in file, click the web server, and then click **plug-in properties**.
12. In Figure 5-36, click **View** to check the plug-in configuration file.



Plug-in properties

☐ Ignore DNS failures during Web server startup

* Refresh configuration interval
60 seconds

Repository copy of Web server plug-in files:

* Plug-in configuration file name
plugin-cfg.xml

☑ Automatically generate the plug-in config

☑ Automatically propagate plug-in configura

* Plug-in key store file name
plugin-key.kdb

Manage keys and certificates
Copy to Web server key store directory

Figure 5-36 View the generated plug-in file

13. Make sure the keyring and stashfile properties are set, as shown in Example 5-8.

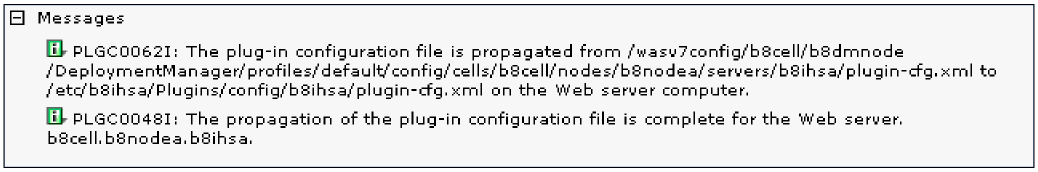
Example 5-8 Excerpt of "plug-in.xml" file

```
<ServerCluster CloneSeparatorChange="false" GetDWLMTable="false"
IgnoreAffinityRequests="true" LoadBalance="Round Robin" Name="b8sr1_WPS"
PostBufferSize="64" PostSizeLimit="-1" RemoveSpecialHeaders="true"
RetryInterval="60">
      <Server CloneID="C5929AF78CE6CF90000003E80000000200000000"
ConnectTimeout="5" ExtendedHandshake="false" LoadBalanceWeight="2"
MaxConnections="-1" Name="b8nodea_b8sr1_WPS.b8nodea.0" ServerIOTimeout="60"
WaitForContinue="false">
```

```
      <Transport Hostname="wtsc42.itso.ibm.com" Port="28067" Protocol="http"/>
      <Transport Hostname="wtsc42.itso.ibm.com" Port="28068" Protocol="https">
         <Property Name="keyring" Value="safkeyring:///WASKeyring.B8CELL"/>
         <Property Name="stashfile" Value="&quot;&quot;"/>
      </Transport>
   </Server>
   <PrimaryServers>
      <Server Name="b8nodea_b8sr1_WPS.b8nodea.0"/>
   </PrimaryServers>
</ServerCluster>
```

14. Return to the web server list, and select the web server.

15. Click **Propagate plug-in**. After a while you will see a completion message, as shown in Figure 5-37.



⊟ Messages

PLGC0062I: The plug-in configuration file is propagated from /wasv7config/b8cell/b8dmnode/DeploymentManager/profiles/default/config/cells/b8cell/nodes/b8nodea/servers/b8ihsa/plugin-cfg.xml to /etc/b8ihsa/Plugins/config/b8ihsa/plugin-cfg.xml on the Web server computer.

PLGC0048I: The propagation of the plug-in configuration file is complete for the Web server. b8cell.b8nodea.b8ihsa.

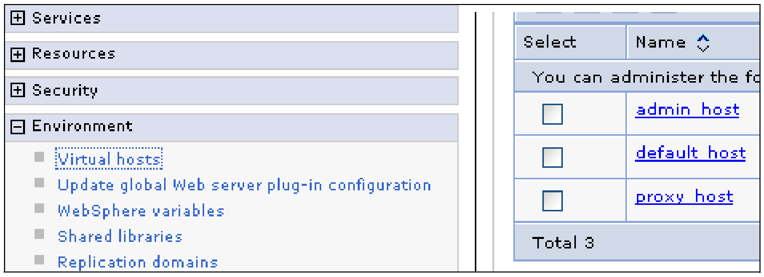
Figure 5-37 Completion message of plug-in propagation

16. If you want to disable round robin workload balancing in the WebSphere plug-in, modify the propagated plug-in.xml file, as described in 5.4.5, “Optional: Disabling workload balancing in the WebSphere plug-in” on page 168.

5.4.4 Adjusting virtual hosts

If you did not adjust virtual hosts in 5.3.4, “Adjusting the virtual hosts” on page 146, you must add the web server ports to the virtual host, default_host.

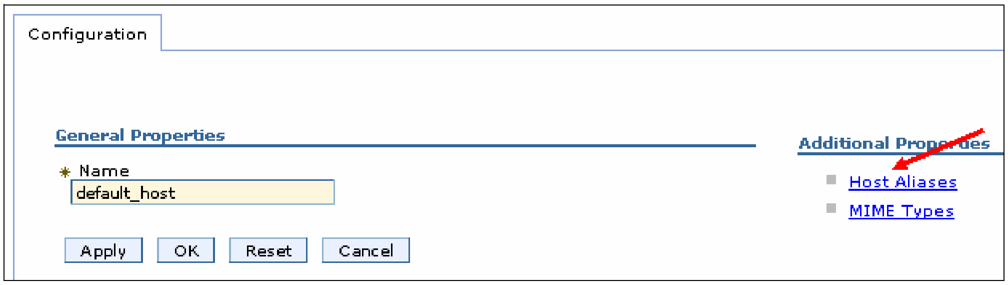
1. Click **Environment** → **Virtual hosts**, as shown in Figure 5-38.



Select	Name
□	admin_host
□	default_host
□	proxy_host
Total 3	

Figure 5-38 Virtual hosts list

2. Click **default_host** → **Host Aliases**, as shown in Figure 5-39 on page 167.



Configuration

General Properties

* Name

default_host

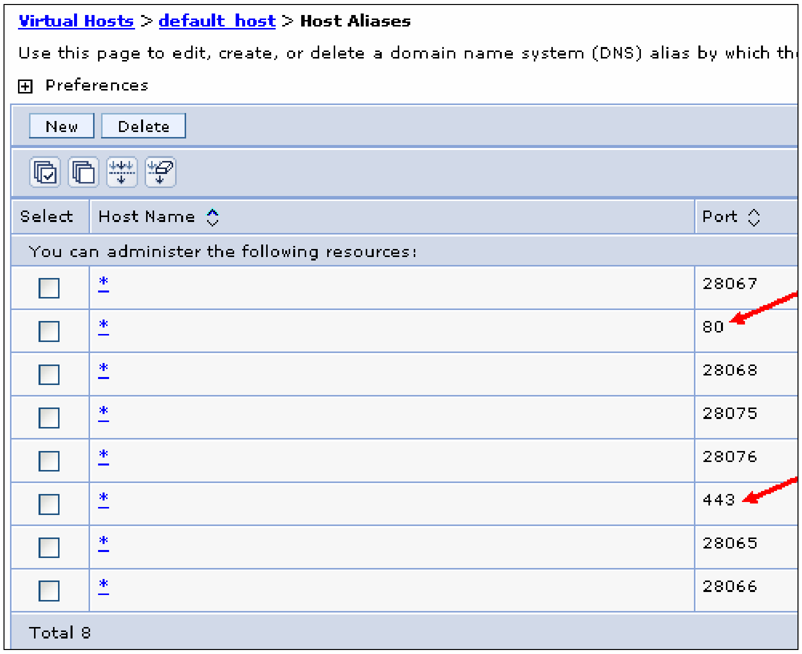
Apply OK Reset Cancel

Additional Properties

- Host Aliases
- MIME Types

Figure 5-39   Host aliases

3. Verify whether the generated web server ports 80 and 443, shown in Figure 5-40, are still in your host aliases list. If so, click he appropriate host name, and change the port according to the definitions in your configuration spreadsheet.



Virtual Hosts > default_host > Host Aliases

Use this page to edit, create, or delete a domain name system (DNS) alias by which th

⊞ Preferences

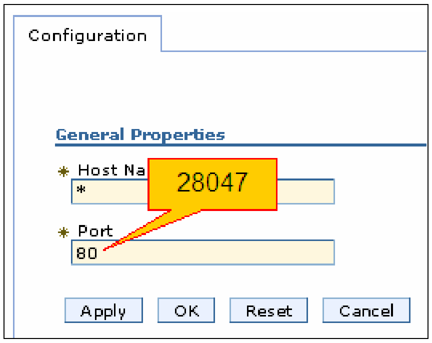
New Delete

Select	Host Name	Port
You can administer the following resources:		
□	*	28067
□	*	80
□	*	28068
□	*	28075
□	*	28076
□	*	443
□	*	28065
□	*	28066

Total 8

Figure 5-40   Generated web server ports

4. If there is no generated port to change, create new host aliases for your web server ports. Press New, leave ”*” for the host name, and change the default port value 80 according to the definitions in your configuration spreadsheet. For the first web server port enter 28047, as shown in Figure 5-41 on page 168, and click **OK**.

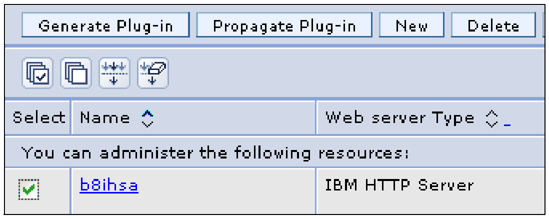


* Host Na[me]: *

* Port: 80

Figure 5-41 Define new host alias

5. Repeat step 4 for the HTTPS port of your web server, and save the changes to the master repository.
6. Now you generate and propagate the plug-in configuration. Return to the web server list view, select the web server, and click **Generate Plug-in**.



Select	Name	Web server Type
You can administer the following resources:		
☑	b8ihsa	IBM HTTP Server

Figure 5-42 Select Web server

7. After successful completion of the plug-in generation, select the web server again, and click **Propagate plug-in**.

5.4.5 Optional: Disabling workload balancing in the WebSphere plug-in

If you are using sysplex distributor for the web servers and you have a web server in each WebSphere Process Server for z/OS LPAR, you can disable the workload balancing in the WebSphere plug-in. For this you must edit the propagated plug-in.xml files in the web server Plug-ins directories. But note, these changes are lost after you propagate the plug-in configuration again.

To disable workload balancing in the WebSphere plug-in:

1. Open a OMVS or telnet session.
2. Go to the plug-in configuration directory /etc/b8ihsa/Plugins/config/b8ihsa, and edit the file plug-in.xml with an ASCII editor.
3. Look for the XML tag *ServerCluster*. In this section, locate the attributes LoadBalance and LoadBalanceWeight, as shown in Example 5-9, and remove both attributes from the tag.

Example 5-9 plug-in.xml

```
<ServerCluster CloneSeparatorChange="false" GetDWLMTable="false"
IgnoreAffinityRequests="true" LoadBalance="Round Robin" Name="b8sr1_WPS"
PostBufferSize="64" PostSizeLimit="-1" RemoveSpecialHeaders="true"
RetryInterval="60">
      <Server CloneID="C5929AF78CE6CF90000003E80000000200000000"
ConnectTimeout="5" ExtendedHandshake="false" LoadBalanceWeight="2"
```

```
MaxConnections="-1" Name="b8nodea_b8sr1_WPS.b8nodea.0" ServerIOTimeout="60"
WaitForContinue="false">
```

4. Repeat step 3 for all ServerCluster tags in the file.
5. Save the file, and restart your web server.

> **Note:** You must modify these values after each new propagation of the plug-in.xml file.

5.4.6 Disabling the trust association interceptors

By default, the trust association interceptor, com.ibm.portal.auth.tai.HTTPBasicAuthTAI, is enabled, which causes errors in the server logs during server startup. To suppress these error messages the trust association interceptor must be disabled unless it is really used.

To disable the trust association interceptors:

1. In the ISC, click **Security** → **Global security**.
2. In the security panel, click **Web and SIP security** → **Trust association**, as shown in Figure 5-43.

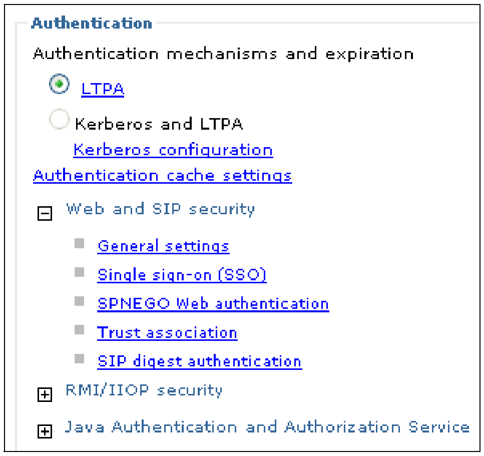
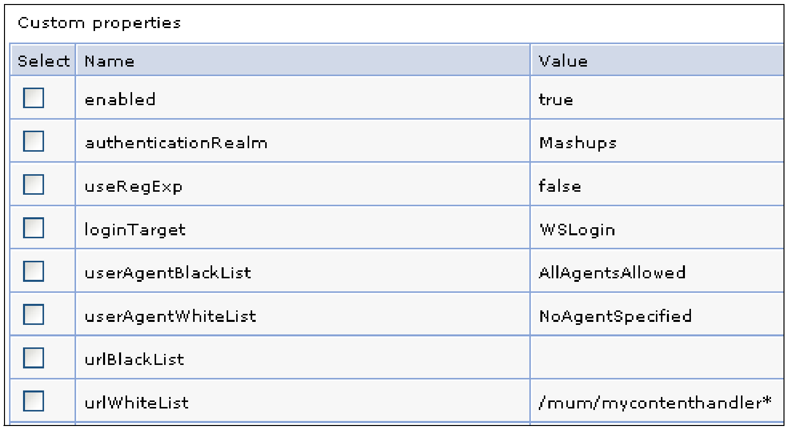

Authentication
Authentication mechanisms and expiration
● LTPA
○ Kerberos and LTPA
Kerberos configuration
Authentication cache settings
⊟ Web and SIP security
General settings
Single sign-on (SSO)
SPNEGO Web authentication
Trust association
SIP digest authentication
⊞ RMI/IIOP security
⊞ Java Authentication and Authorization Service

Figure 5-43 Global security: Trust association

3. Click **Interceptors** → **com.ibm.portal.auth.tai.HTTPBasicAuthTAI**.
4. Figure 5-44 on page 170 shows a list of properties. Usually, only the enable property has to be modified, but we found property urlBlackList configured with no value, which prevented us from saving the changes. Therefore we removed that property. Select **urlBlackList**, and click **Delete**.

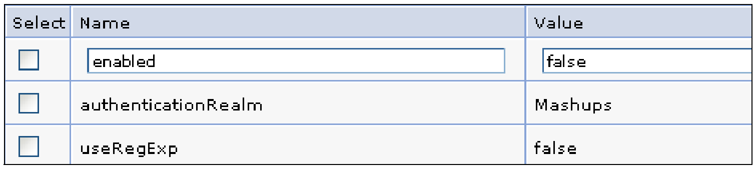


Custom properties

Select	Name	Value
□	enabled	true
□	authenticationRealm	Mashups
□	useRegExp	false
□	loginTarget	WSLogin
□	userAgentBlackList	AllAgentsAllowed
□	userAgentWhiteList	NoAgentSpecified
□	urlBlackList	
□	urlWhiteList	/mum/mycontenthandler*

Figure 5-44 Properties of interceptor com.ibm.portal.auth.tai.HTTPBasicAuthTAI

5. Select **enabled**, as shown in Figure 5-45, and click **Edit**.



Select	Name	Value
□	enabled	false
□	authenticationRealm	Mashups
□	useRegExp	false

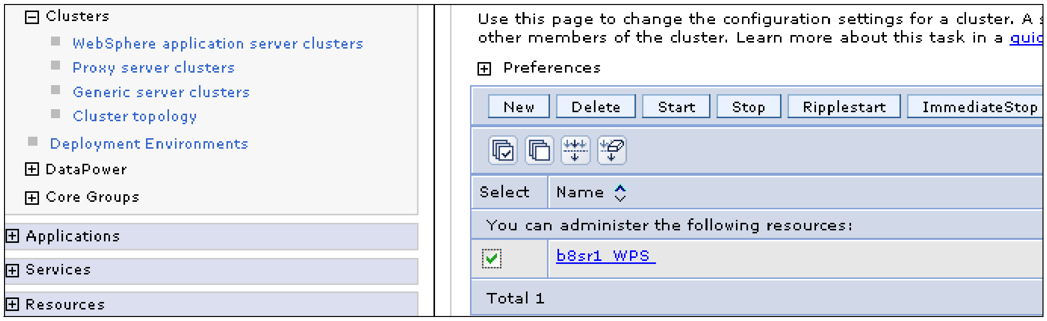
Figure 5-45 Change value for property enabled

6. Change the value to **false**, and click **OK**.
7. Save the changes to the master repository.

5.5 Starting the cluster

Now you completed all of the configuration and adjustments of the cluster, so you can start it for the first time. To start the cluster:

1. Click **Clusters** → **Application server clusters**. Select your cluster in the cluster list, and click **Start**, as shown in Figure 5-46.



Use this page to change the configuration settings for a cluster. A s
other members of the cluster. Learn more about this task in a guid

Select	Name
You can administer the following resources:	
☑	b8sr1_WPS
Total 1	

Figure 5-46 Start cluster

2. It will take some time until the cluster, respectively the servers, are started and ready for e-business. Open a 3270 session, and observe the job logs of the servers controller, adjunct, and servant regions. Look for error messages.

3. If the arrows in the application server view are green, the servers are ready for e-business. Also, you will see the following message in the server region logs:

   ```
   WSVR0001I: Server SERVANT PROCESS b8sr1_WPS.b8nodea.0 open for e-business
   ```

4. Now you can start with the verification of your installation and configuration, which we describe in Chapter 6, “Verifying the configuration” on page 195.

5.6 Optional: Creating a cluster using the custom deployment environment wizard

If you have your own specific naming conventions for resources in your WebSphere environment, you can use the custom deployment environment wizard. This wizard supports defining cluster and servers with names according to your naming conventions. But the price for this advantage is that you must install most of the applications manually after the deployment generation.

In summary, the steps to create a custom deployment environment are:

1. Check prerequisites.
2. Configure prerequisite resources.
3. Go through the custom deployment environment wizard.
4. Save and generate the deployment environment.

5.6.1 Checking prerequisites

Before you can configure your deployment environment make sure that:

- Your deployment manager and managed nodes profiles are augmented with WebSphere Process Server for z/OS.
- The deployment manager and, optionally the node agents, are started and running.
- The database and database artifacts for your WebSphere Process Server for z/OS cell are created.
- The web server(s) are installed and configured, if you want to use web server(s) for load balancing.
- Log streams are configured for transaction logging.
- Auto-synchronization for the nodes is disabled. If auto-synchronization is enabled, you will get an error messages at the end of the deployment environment generation when saving the changes to the master repository.

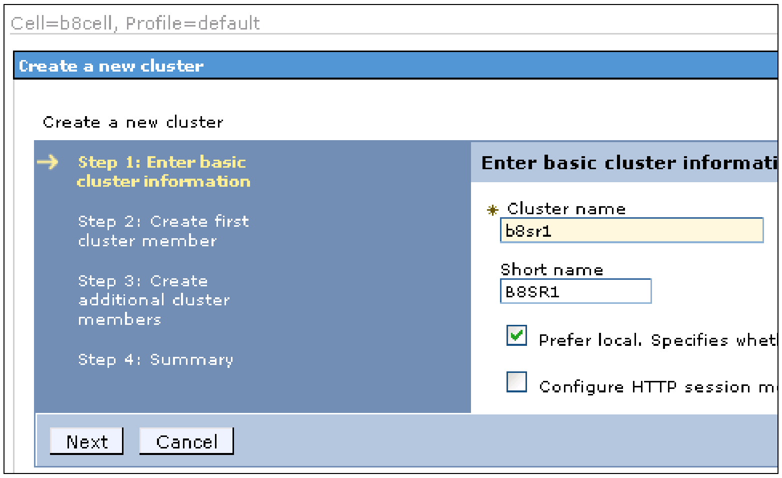
5.6.2 Configuring prerequisite resources

For the custom deployment environment wizard, the following resources must be configured in advance:

- The cluster
- The cluster members
- The JDBC provider on cluster scope

Creating the cluster and cluster members

1. Login to the ISC, and click **Servers** → **Clusters** → **New**.
2. Enter the cluster long and short name, as shown in Figure 5-47. In our sample, we used the cluster transition name b8sr1 and the cluster short name B8SR1.



Cell=b8cell, Profile=default

Create a new cluster

Create a new cluster

→ **Step 1: Enter basic cluster information**

Step 2: Create first cluster member

Step 3: Create additional cluster members

Step 4: Summary

Enter basic cluster informati

* Cluster name: b8sr1

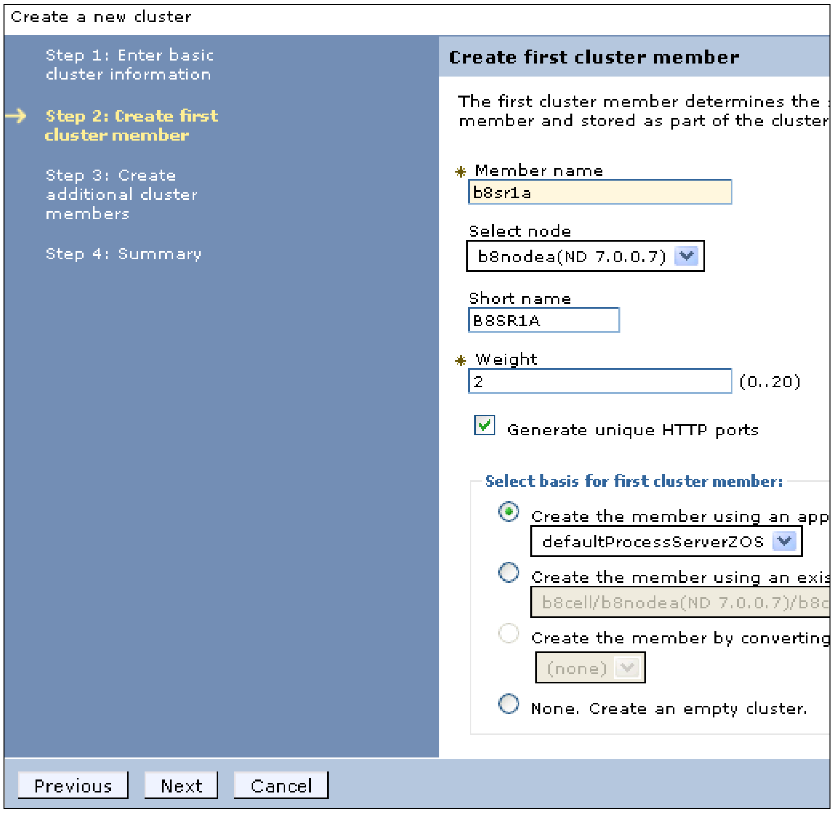
Short name: B8SR1

☑ Prefer local. Specifies whet

☐ Configure HTTP session m

Figure 5-47 Custom cluster definition

3. Click **Next**.
4. In Figure 5-48, define the first cluster member. In our sample, we used b8sr1a for the server long name and B8SR1A for the short name.



Create a new cluster

Step 1: Enter basic cluster information

→ **Step 2: Create first cluster member**

Step 3: Create additional cluster members

Step 4: Summary

Create first cluster member

The first cluster member determines the
member and stored as part of the cluster

* Member name: b8sr1a

Select node: b8nodea(ND 7.0.0.7)

Short name: B8SR1A

* Weight: 2 (0..20)

☑ Generate unique HTTP ports

Select basis for first cluster member:

● Create the member using an app
defaultProcessServerZOS

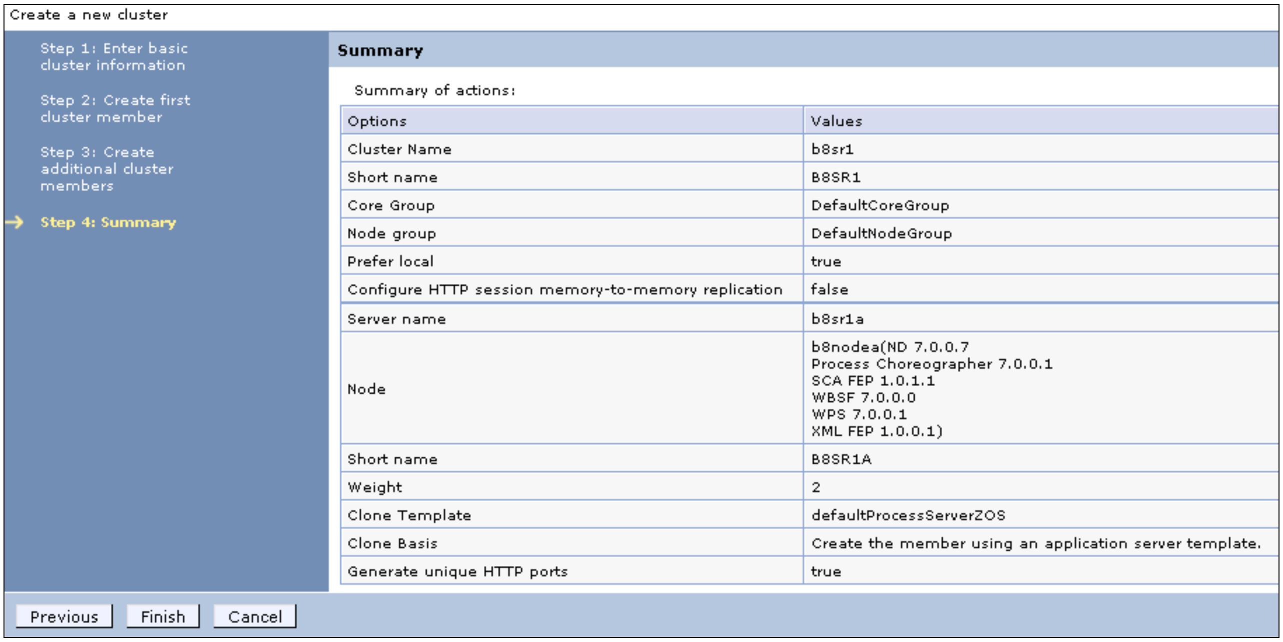
○ Create the member using an exis
b8cell/b8nodea(ND 7.0.0.7)/b8c

○ Create the member by converting
(none)

○ None. Create an empty cluster.

Figure 5-48 Custom cluster member definition

5. Click **Next**.
6. Repeat steps 4 and 5 until you define all of your cluster members.
7. Click **Next**. A summary panel is displayed, as shown in Figure 5-49.



Create a new cluster

Step 1: Enter basic cluster information
Step 2: Create first cluster member
Step 3: Create additional cluster members
Step 4: Summary

Summary

Summary of actions:

Options	Values
Cluster Name	b8sr1
Short name	B8SR1
Core Group	DefaultCoreGroup
Node group	DefaultNodeGroup
Prefer local	true
Configure HTTP session memory-to-memory replication	false
Server name	b8sr1a
Node	b8nodea(ND 7.0.0.7 Process Choreographer 7.0.0.1 SCA FEP 1.0.1.1 WBSF 7.0.0.0 WPS 7.0.0.1 XML FEP 1.0.0.1)
Short name	B8SR1A
Weight	2
Clone Template	defaultProcessServerZOS
Clone Basis	Create the member using an application server template.
Generate unique HTTP ports	true

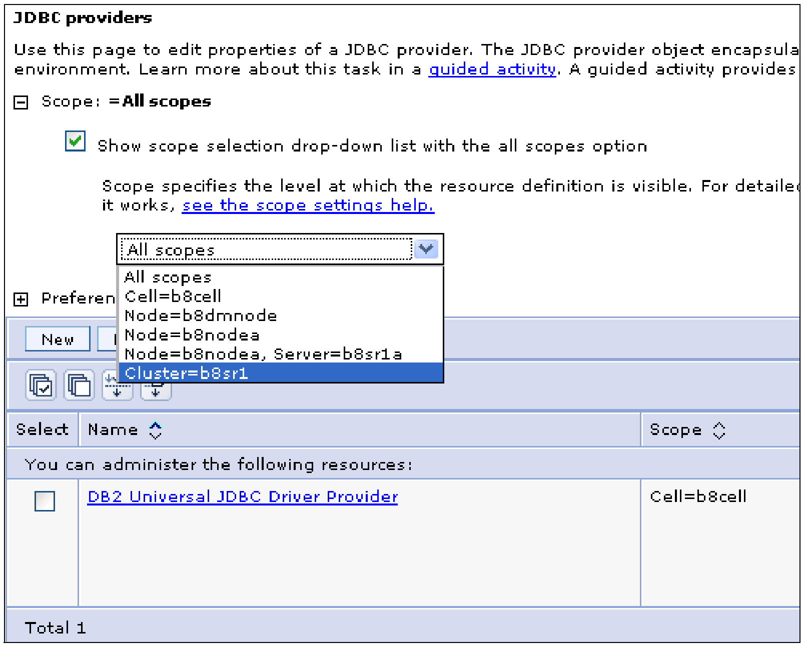
Figure 5-49 Cluster definition summary

8. Click **Finish**, and your cluster and cluster members are created.
9. After successful creation of the cluster and cluster members, save the changes to the master repository.

Configuring a DB2 JDBC provider on the cluster scope

Before you can create the custom deployment environment you must configure a DB2 JDBC provider on the scope of the previously created cluster:

1. Click **Resources** → **JDBC provider**.
2. Switch the scope for the JDBC provider list to the new cluster scope, as shown in Figure 5-50 on page 174.



JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsula
environment. Learn more about this task in a guided activity. A guided activity provides

⊟ Scope: =**All scopes**

☑ Show scope selection drop-down list with the all scopes option

Scope specifies the level at which the resource definition is visible. For detaile
it works, see the scope settings help.

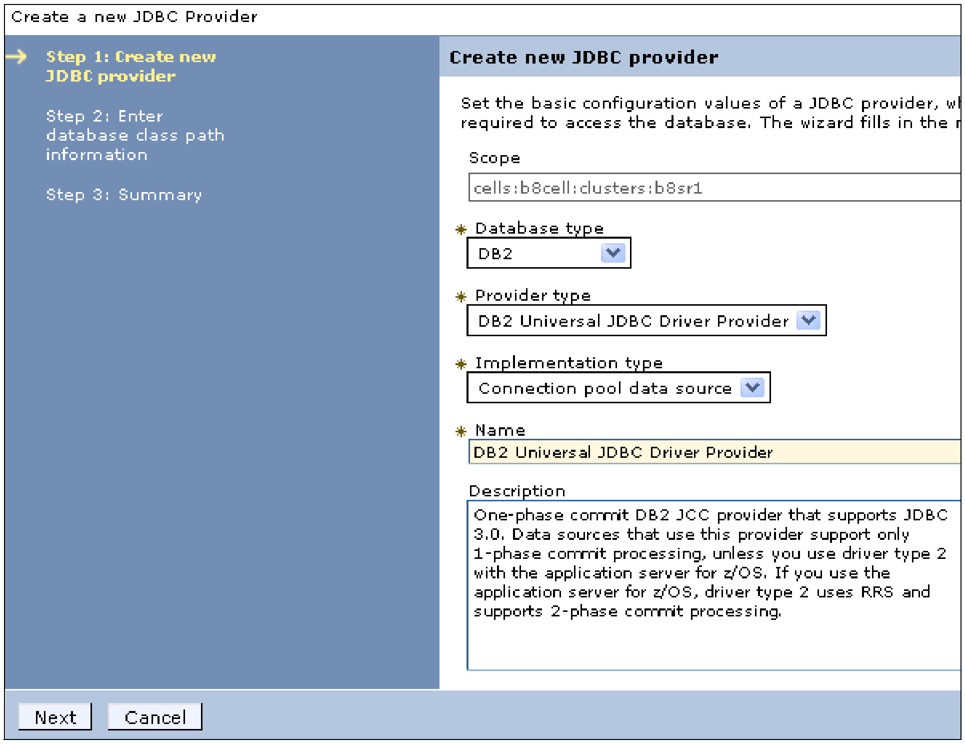
All scopes
- All scopes
- Cell=b8cell
- Node=b8dmnode
- Node=b8nodea
- Node=b8nodea, Server=b8sr1a
- Cluster=b8sr1

⊞ Preferen

Select	Name	Scope
You can administer the following resources:		
□	DB2 Universal JDBC Driver Provider	Cell=b8cell
Total 1		

Figure 5-50 Switch scope for JDBC provider list

3. Click **New**.
4. Make the following selections in Figure 5-51:
   a. In the Database type field, select **DB2**.
   b. In the Provider type field, select **DB2 Universal JDBC Driver Provider**.
   c. In the Implementation type field, select **Connection pool data source**.
   d. In the Name field, type `DB2 Universal JDBC Driver Provider`.



Create a new JDBC Provider

→ **Step 1: Create new JDBC provider**
Step 2: Enter database class path information
Step 3: Summary

Create new JDBC provider

Set the basic configuration values of a JDBC provider, wl
required to access the database. The wizard fills in the r

Scope
cells:b8cell:clusters:b8sr1

* Database type
DB2

* Provider type
DB2 Universal JDBC Driver Provider

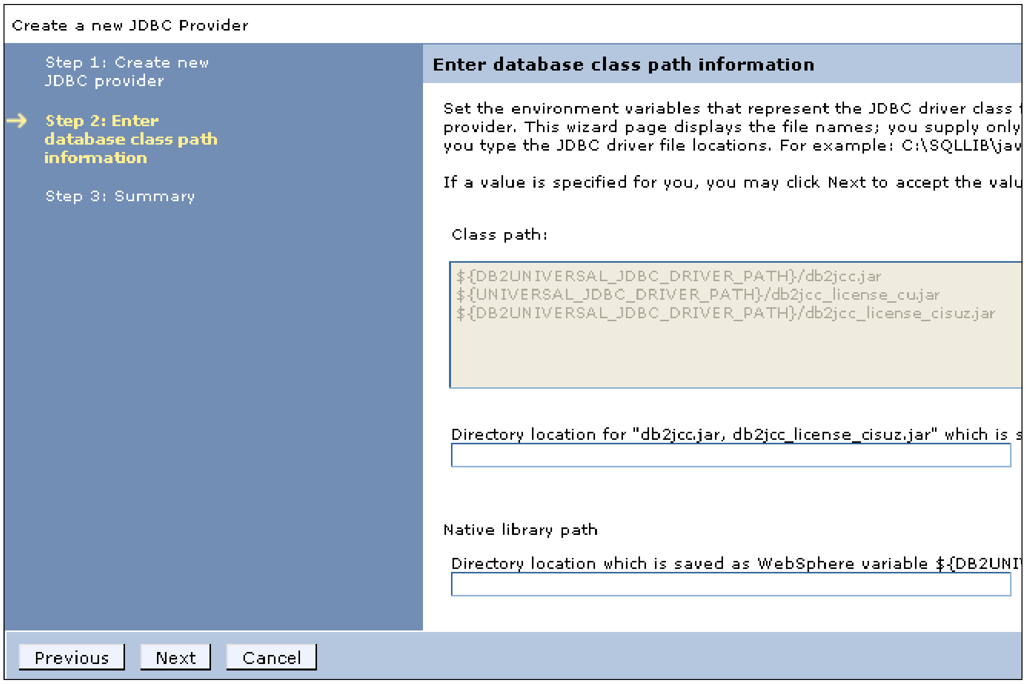
* Implementation type
Connection pool data source

* Name
DB2 Universal JDBC Driver Provider

Description
One-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support only 1-phase commit processing, unless you use driver type 2 with the application server for z/OS. If you use the application server for z/OS, driver type 2 uses RRS and supports 2-phase commit processing.

Figure 5-51 Define a new DB2 JDBC provider

5. Click **Next**. Figure 5-52 is displayed.



Create a new JDBC Provider

Step 1: Create new JDBC provider
→ Step 2: Enter database class path information
Step 3: Summary

Enter database class path information

Set the environment variables that represent the JDBC driver class f
provider. This wizard page displays the file names; you supply only
you type the JDBC driver file locations. For example: C:\SQLLIB\jav

If a value is specified for you, you may click Next to accept the valu

Class path:

```
${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar
${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar
${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar
```

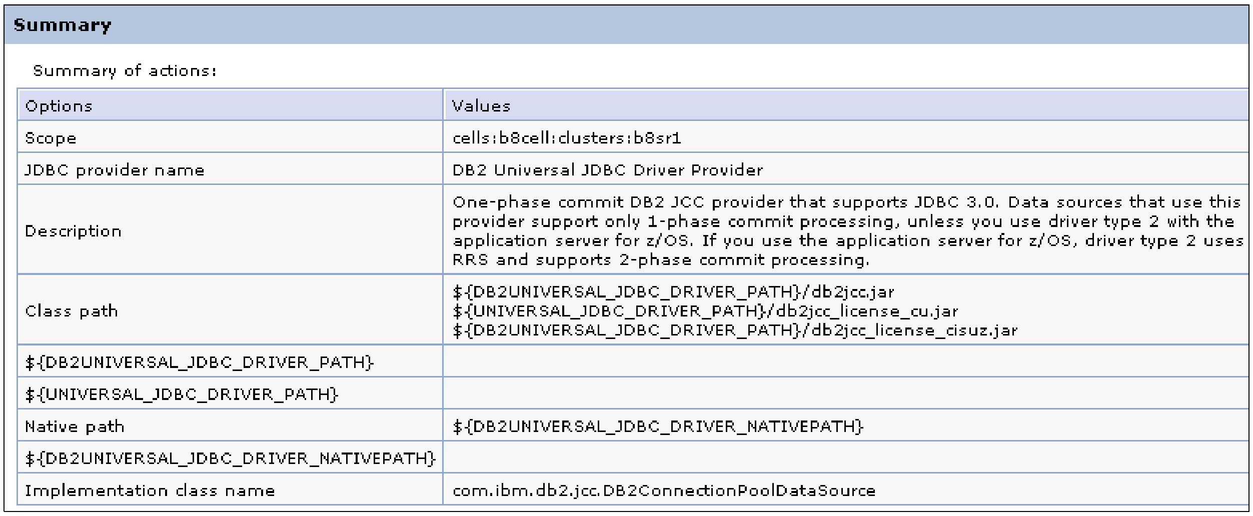
Directory location for "db2jcc.jar, db2jcc_license_cisuz.jar" which is s

Native library path

Directory location which is saved as WebSphere variable ${DB2UNI

Figure 5-52 JDBC provider path summary

6. Click **Next**. A summary panel is displayed, as shown in Figure 5-53.



Summary

Summary of actions:

Options	Values
Scope	cells:b8cell:clusters:b8sr1
JDBC provider name	DB2 Universal JDBC Driver Provider
Description	One-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support only 1-phase commit processing, unless you use driver type 2 with the application server for z/OS. If you use the application server for z/OS, driver type 2 uses RRS and supports 2-phase commit processing.
Class path	${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar ${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar ${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar
${DB2UNIVERSAL_JDBC_DRIVER_PATH}	
${UNIVERSAL_JDBC_DRIVER_PATH}	
Native path	${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}
${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}	
Implementation class name	com.ibm.db2.jcc.DB2ConnectionPoolDataSource

Figure 5-53 JDBC provider summary

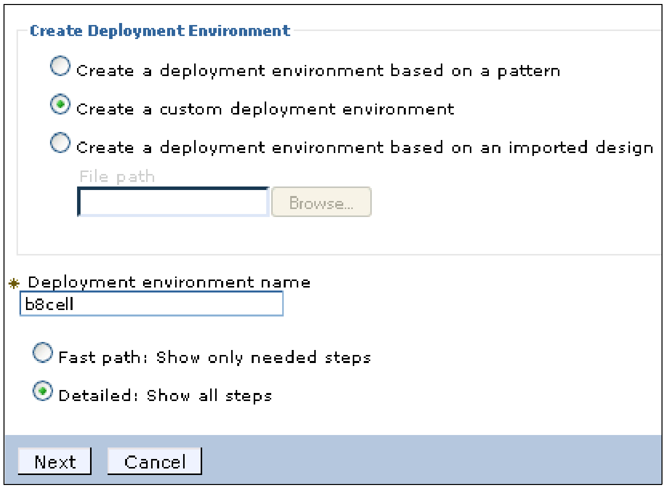
7. Click **Finish** to create the JDBC provider.
8. After successful creation of the new JDBC provider save the changes to the master repository.

5.6.3 Using the custom deployment environment wizard

Now that you created all of the prerequisite resources you can create the deployment environment.

To create the deployment environment:

1. Click **Servers** → **Deployment environments** → **New**.
2. Select the **Create a custom deployment environment** option, and enter a name for your deployment environment, as shown in Figure 5-54. In our sample, we used b8cell.



Create Deployment Environment

○ Create a deployment environment based on a pattern

● Create a custom deployment environment

○ Create a deployment environment based on an imported design

File path

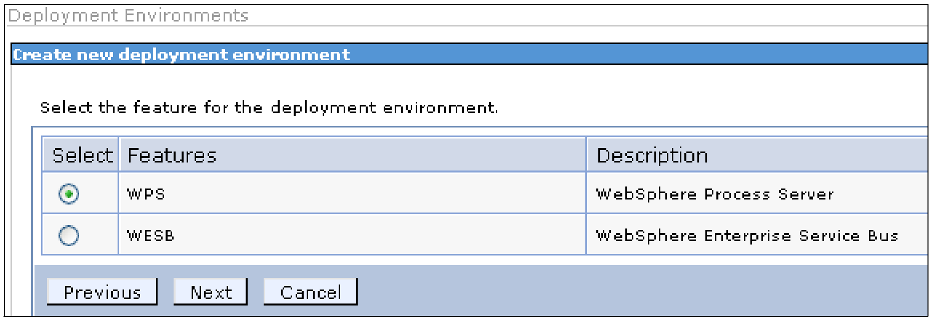
* Deployment environment name: b8cell

○ Fast path: Show only needed steps

● Detailed: Show all steps

Figure 5-54 Select custom deployment environment

3. Click **Next**.
4. Select the **WPS** option, as shown in Figure 5-55, and click **Next**.



Deployment Environments

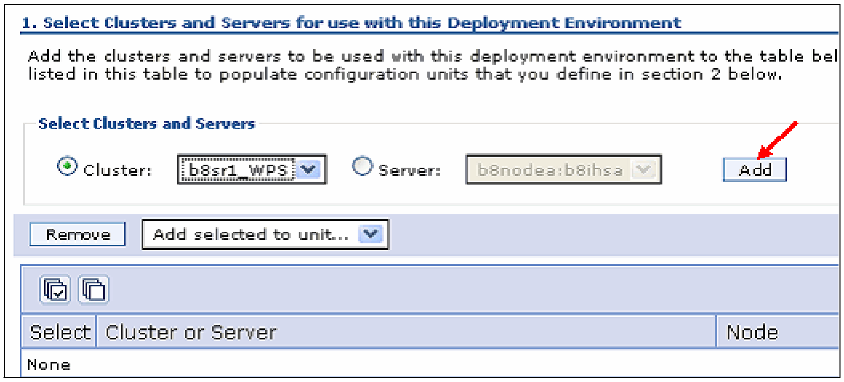
Create new deployment environment

Select the feature for the deployment environment.

Select	Features	Description
●	WPS	WebSphere Process Server
○	WESB	WebSphere Enterprise Service Bus

Figure 5-55 Select WPS feature

5. Select the **b8sr1** cluster for the deployment environment, as shown in Figure 5-56, and click **Add**.



1. Select Clusters and Servers for use with this Deployment Environment

Add the clusters and servers to be used with this deployment environment to the table bel
listed in this table to populate configuration units that you define in section 2 below.

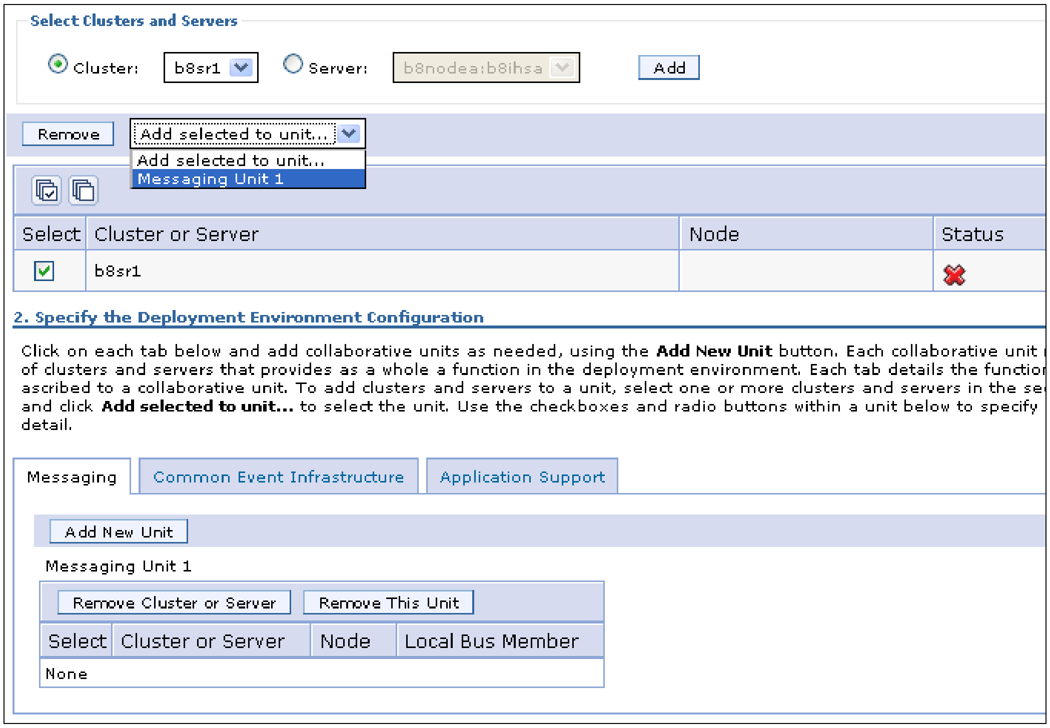
Select Clusters and Servers

● Cluster: b8sr1_WPS ○ Server: b8nodea:b8ihsa

Select	Cluster or Server	Node
None		

Figure 5-56 Select the cluster to be used

6. Select the cluster and add it to the messaging infrastructure by selecting **Messaging Unit 1** from the drop-down list, as shown in Figure 5-57.



Select Clusters and Servers

● Cluster: b8sr1 ○ Server: b8nodea:b8ihsa Add

Remove | Add selected to unit...

Add selected to unit...
Messaging Unit 1

Select	Cluster or Server	Node	Status
☑	b8sr1		✖

2. Specify the Deployment Environment Configuration

Click on each tab below and add collaborative units as needed, using the **Add New Unit** button. Each collaborative unit of clusters and servers that provides as a whole a function in the deployment environment. Each tab details the functio ascribed to a collaborative unit. To add clusters and servers to a unit, select one or more clusters and servers in the se and click **Add selected to unit...** to select the unit. Use the checkboxes and radio buttons within a unit below to specify detail.

Messaging | Common Event Infrastructure | Application Support

Add New Unit

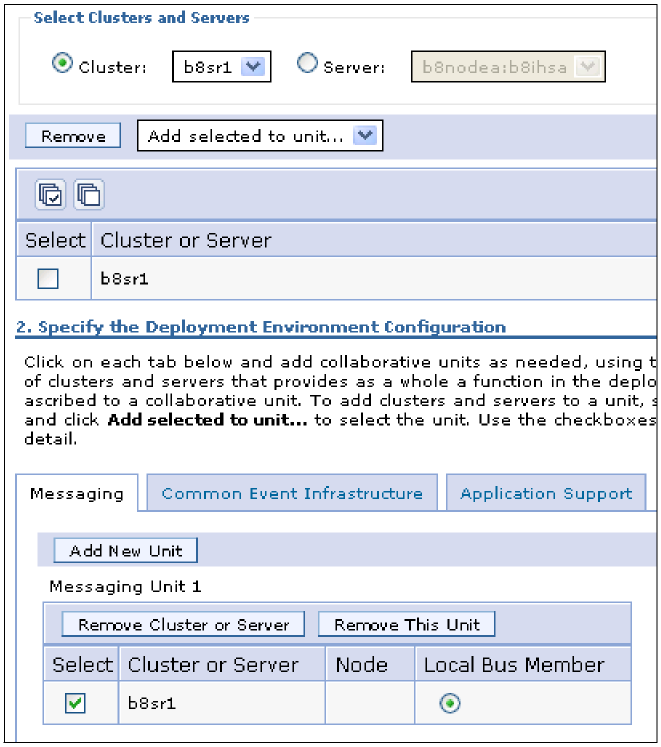
Messaging Unit 1

Remove Cluster or Server | Remove This Unit

Select	Cluster or Server	Node	Local Bus Member
None			

Figure 5-57 Add cluster to the messaging infrastructure

7. Select the cluster as a bus member, as shown in Figure 5-58.



Select Clusters and Servers

● Cluster: b8sr1 ○ Server: b8nodea:b8ihsa

Remove | Add selected to unit...

Select	Cluster or Server
□	b8sr1

2. Specify the Deployment Environment Configuration

Click on each tab below and add collaborative units as needed, using t of clusters and servers that provides as a whole a function in the deplo ascribed to a collaborative unit. To add clusters and servers to a unit, s and click **Add selected to unit...** to select the unit. Use the checkboxes detail.

Messaging | Common Event Infrastructure | Application Support

Add New Unit

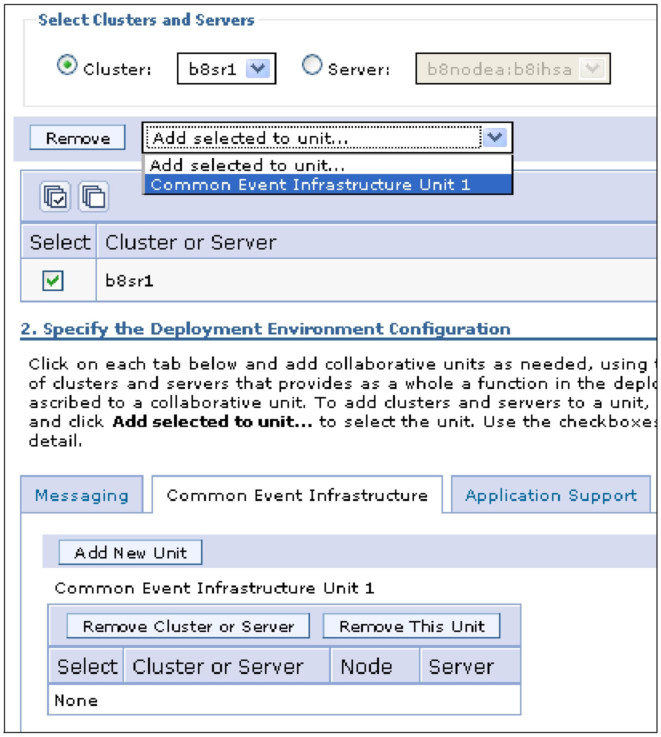
Messaging Unit 1

Remove Cluster or Server | Remove This Unit

Select	Cluster or Server	Node	Local Bus Member
☑	b8sr1		●

Figure 5-58 Select cluster as bus member

8. Click the **Common Event Infrastructure** tab. Select the cluster, and select **Common Event Infrastructure** from the pull-down list, as shown in Figure 5-59.



● Cluster: b8sr1 ○ Server: b8nodea:b8ihsa

Select	Cluster or Server
☑	b8sr1

2. Specify the Deployment Environment Configuration

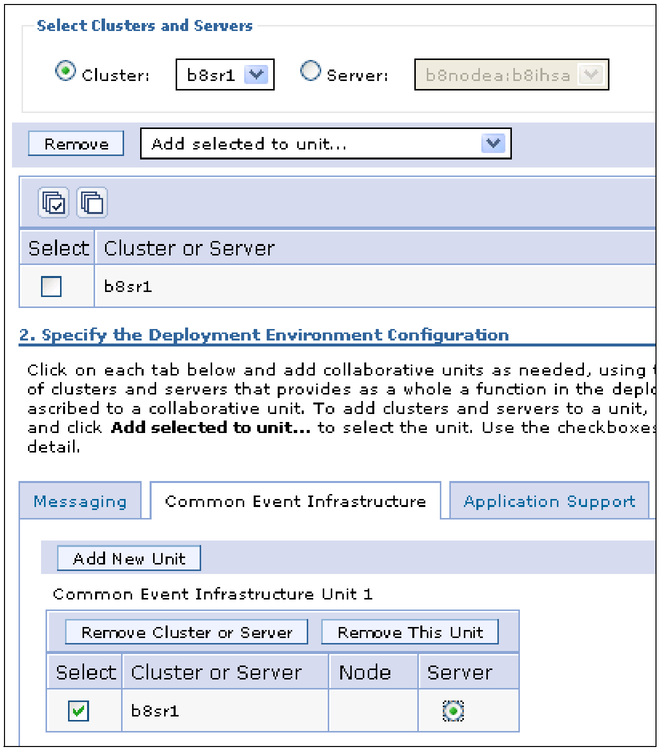
Click on each tab below and add collaborative units as needed, using t of clusters and servers that provides as a whole a function in the deplo ascribed to a collaborative unit. To add clusters and servers to a unit, and click **Add selected to unit...** to select the unit. Use the checkboxe detail.

Common Event Infrastructure Unit 1

Select	Cluster or Server	Node	Server
None			

Figure 5-59 Add cluster to Common Event Infrastructure

9. In the Select Clusters and Servers section, select the cluster. In the Common Event Infrastructure Unit 1 section, select the cluster and select the **Server** option, as shown in Figure 5-60 on page 179.



Select	Cluster or Server
□	b8sr1

2. Specify the Deployment Environment Configuration

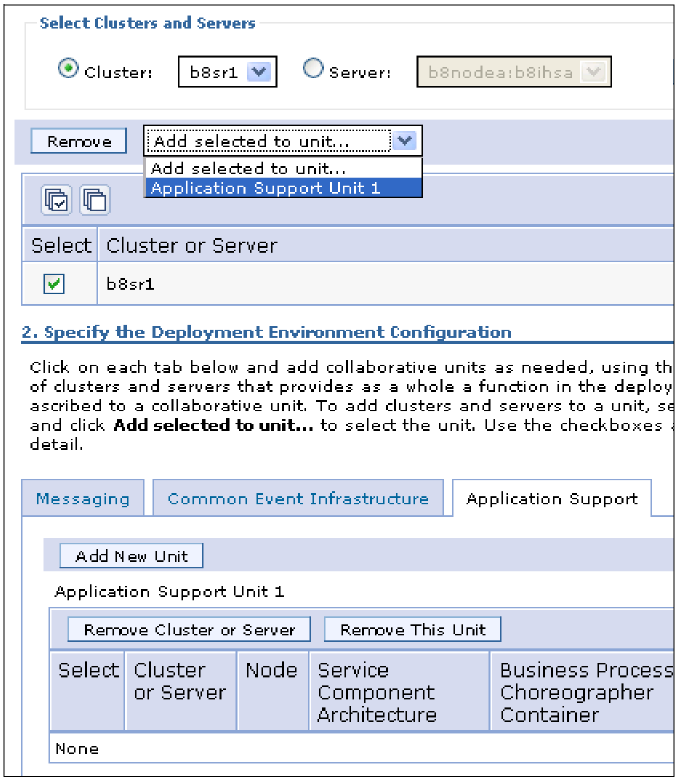
Click on each tab below and add collaborative units as needed, using t of clusters and servers that provides as a whole a function in the deplo ascribed to a collaborative unit. To add clusters and servers to a unit, and click **Add selected to unit...** to select the unit. Use the checkboxe detail.

Common Event Infrastructure Unit 1

Select	Cluster or Server	Node	Server
☑	b8sr1		●

Figure 5-60 Select the event server target

10.Click the Application Support tab. Select the cluster, and select **Application Support Unit** from the pull-down list, as shown in Figure 5-61.



Select	Cluster or Server
☑	b8sr1

2. Specify the Deployment Environment Configuration

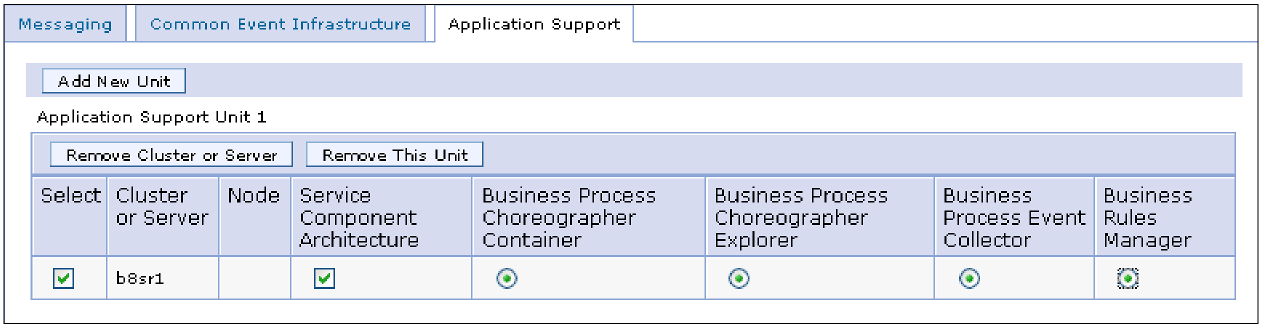
Click on each tab below and add collaborative units as needed, using th of clusters and servers that provides as a whole a function in the deploy ascribed to a collaborative unit. To add clusters and servers to a unit, se and click **Add selected to unit...** to select the unit. Use the checkboxes detail.

Application Support Unit 1

Select	Cluster or Server	Node	Service Component Architecture	Business Process Choreographer Container
None				

Figure 5-61 Add cluster to application support unit

11. In the Application Support Unit 1 panel, select all of the options, as shown in Figure 5-62. Click **Next**.

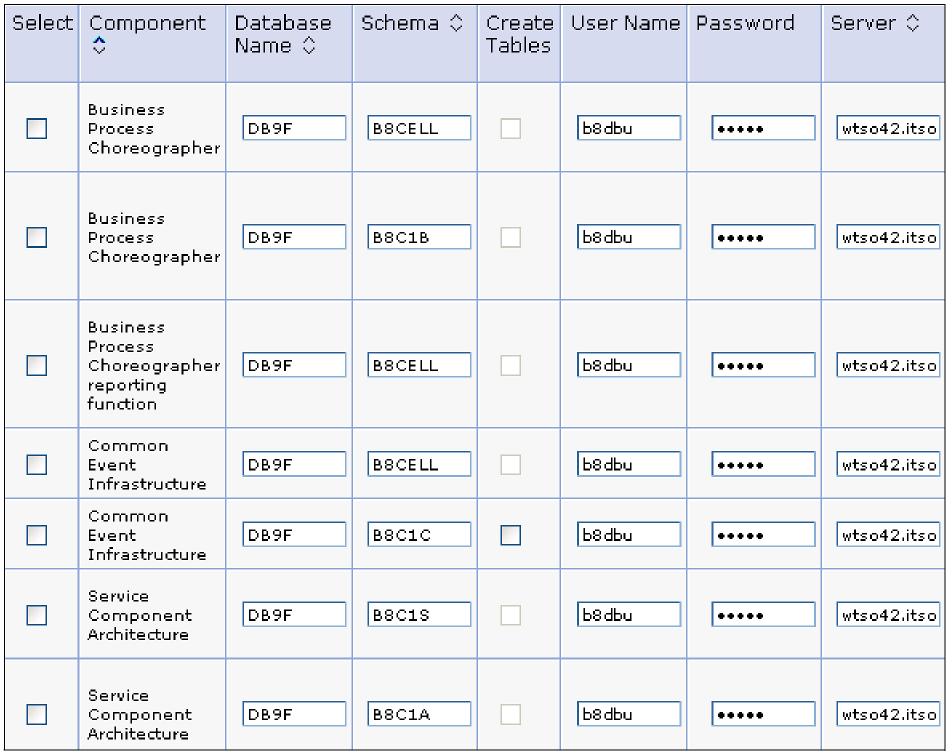


Application Support Unit 1

Select	Cluster or Server	Node	Service Component Architecture	Business Process Choreographer Container	Business Process Choreographer Explorer	Business Process Event Collector	Business Rules Manager
☑	b8sr1		☑	◉	◉	◉	◉

Figure 5-62 Add cluster for all application targets

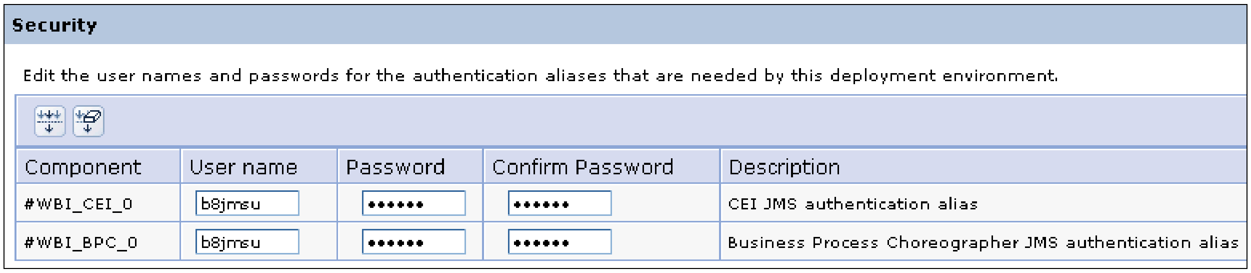
12. In Figure 5-63, you must configure all data sources. Make sure to specify the correct schema, user names, and passwords. Click **Next**.



Select	Component	Database Name	Schema	Create Tables	User Name	Password	Server
□	Business Process Choreographer	DB9F	B8CELL	□	b8dbu	•••••	wtso42.itso
□	Business Process Choreographer	DB9F	B8C1B	□	b8dbu	•••••	wtso42.itso
□	Business Process Choreographer reporting function	DB9F	B8CELL	□	b8dbu	•••••	wtso42.itso
□	Common Event Infrastructure	DB9F	B8CELL	□	b8dbu	•••••	wtso42.itso
□	Common Event Infrastructure	DB9F	B8C1C	□	b8dbu	•••••	wtso42.itso
□	Service Component Architecture	DB9F	B8C1S	□	b8dbu	•••••	wtso42.itso
□	Service Component Architecture	DB9F	B8C1A	□	b8dbu	•••••	wtso42.itso

Figure 5-63 Data source definition table

13. Enter the JMS user IDs and passwords, as shown in Figure 5-64 on page 181, and click **Next**.



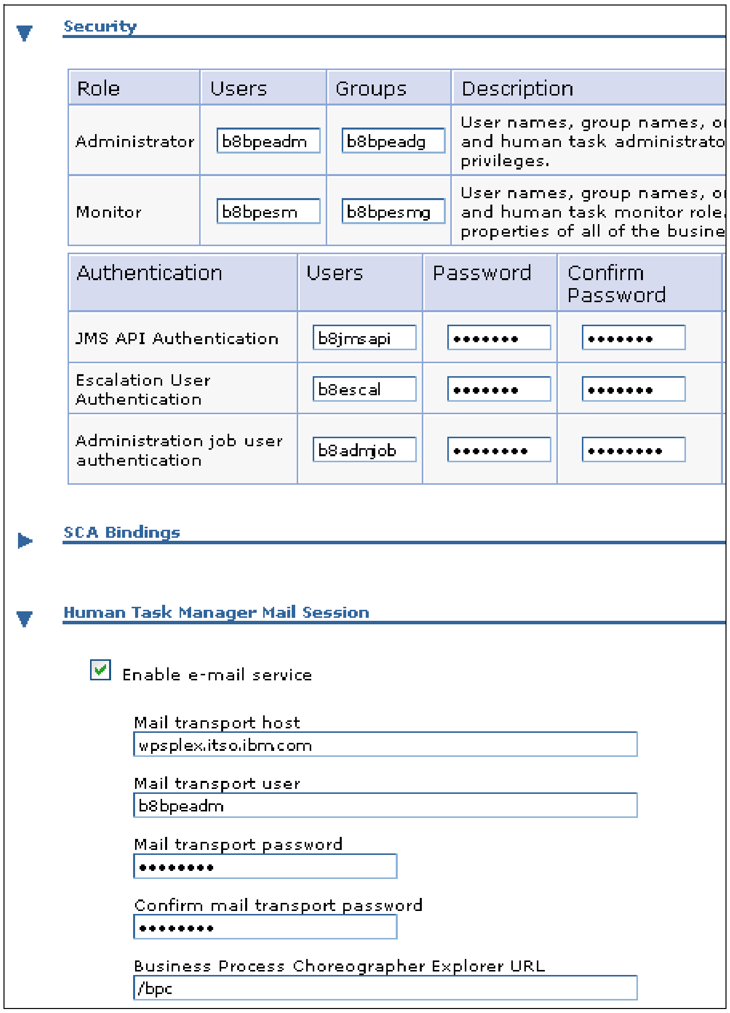
Security

Edit the user names and passwords for the authentication aliases that are needed by this deployment environment.

Component	User name	Password	Confirm Password	Description
#WBI_CEI_0	b8jmsu	••••••	••••••	CEI JMS authentication alias
#WBI_BPC_0	b8jmsu	••••••	••••••	Business Process Choreographer JMS authentication alias

Figure 5-64 JMS user ids and passwords

14. As shown in Figure 5-65, enter the user IDs, passwords, and groups for the business process choreographer configuration and for the mail service. Click **Next**.



Security

Role	Users	Groups	Description
Administrator	b8bpeadm	b8bpeadg	User names, group names, o and human task administrato privileges.
Monitor	b8bpesm	b8bpesmg	User names, group names, o and human task monitor role. properties of all of the busine

Authentication	Users	Password	Confirm Password
JMS API Authentication	b8jmsapi	•••••••	•••••••
Escalation User Authentication	b8escal	•••••••	•••••••
Administration job user authentication	b8admjob	••••••••	••••••••

SCA Bindings

Human Task Manager Mail Session

☑ Enable e-mail service

Mail transport host
wpsplex.itso.ibm.com

Mail transport user
b8bpeadm

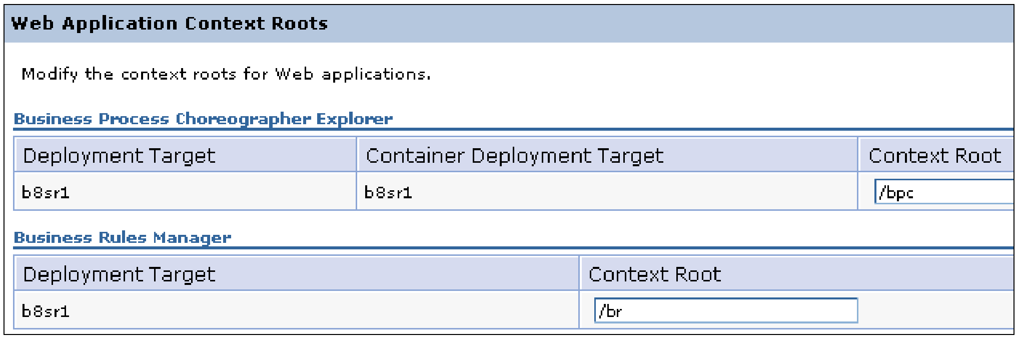
Mail transport password
••••••••

Confirm mail transport password
••••••••

Business Process Choreographer Explorer URL
/bpc

Figure 5-65 Business process choreographer configuration

15. Figure 5-66 on page 182 defines the web application context roots. Verify the values, and click **Next**.



Web Application Context Roots

Modify the context roots for Web applications.

Business Process Choreographer Explorer

Deployment Target	Container Deployment Target	Context Root
b8sr1	b8sr1	/bpc

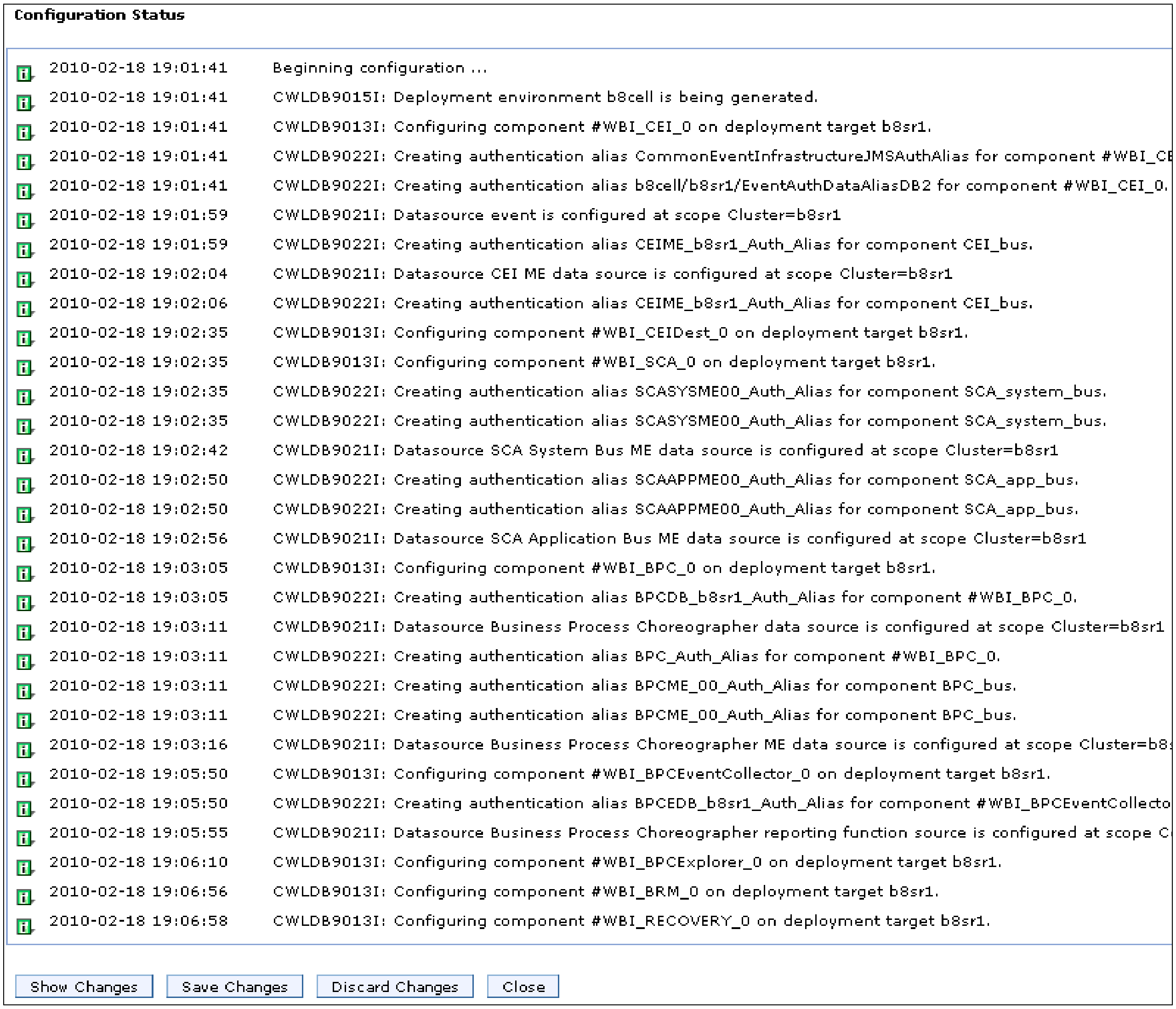
Business Rules Manager

Deployment Target	Context Root
b8sr1	/br

Figure 5-66 Web application context roots

16. On the summary panel, Figure 5-67, verify again the values, and click **Finish** → **Generate Environment**. The deployment environment is generated.

17. After successful completion of the deployment environment generation, save the changes to the master repository.



Configuration Status

```
2010-02-18 19:01:41    Beginning configuration ...
2010-02-18 19:01:41    CWLDB9015I: Deployment environment b8cell is being generated.
2010-02-18 19:01:41    CWLDB9013I: Configuring component #WBI_CEI_0 on deployment target b8sr1.
2010-02-18 19:01:41    CWLDB9022I: Creating authentication alias CommonEventInfrastructureJMSAuthAlias for component #WBI_CE
2010-02-18 19:01:41    CWLDB9022I: Creating authentication alias b8cell/b8sr1/EventAuthDataAliasDB2 for component #WBI_CEI_0.
2010-02-18 19:01:59    CWLDB9021I: Datasource event is configured at scope Cluster=b8sr1
2010-02-18 19:01:59    CWLDB9022I: Creating authentication alias CEIME_b8sr1_Auth_Alias for component CEI_bus.
2010-02-18 19:02:04    CWLDB9021I: Datasource CEI ME data source is configured at scope Cluster=b8sr1
2010-02-18 19:02:06    CWLDB9022I: Creating authentication alias CEIME_b8sr1_Auth_Alias for component CEI_bus.
2010-02-18 19:02:35    CWLDB9013I: Configuring component #WBI_CEIDest_0 on deployment target b8sr1.
2010-02-18 19:02:35    CWLDB9013I: Configuring component #WBI_SCA_0 on deployment target b8sr1.
2010-02-18 19:02:35    CWLDB9022I: Creating authentication alias SCASYSME00_Auth_Alias for component SCA_system_bus.
2010-02-18 19:02:35    CWLDB9022I: Creating authentication alias SCASYSME00_Auth_Alias for component SCA_system_bus.
2010-02-18 19:02:42    CWLDB9021I: Datasource SCA System Bus ME data source is configured at scope Cluster=b8sr1
2010-02-18 19:02:50    CWLDB9022I: Creating authentication alias SCAAPPME00_Auth_Alias for component SCA_app_bus.
2010-02-18 19:02:50    CWLDB9022I: Creating authentication alias SCAAPPME00_Auth_Alias for component SCA_app_bus.
2010-02-18 19:02:56    CWLDB9021I: Datasource SCA Application Bus ME data source is configured at scope Cluster=b8sr1
2010-02-18 19:03:05    CWLDB9013I: Configuring component #WBI_BPC_0 on deployment target b8sr1.
2010-02-18 19:03:05    CWLDB9022I: Creating authentication alias BPCDB_b8sr1_Auth_Alias for component #WBI_BPC_0.
2010-02-18 19:03:11    CWLDB9021I: Datasource Business Process Choreographer data source is configured at scope Cluster=b8sr1
2010-02-18 19:03:11    CWLDB9022I: Creating authentication alias BPC_Auth_Alias for component #WBI_BPC_0.
2010-02-18 19:03:11    CWLDB9022I: Creating authentication alias BPCME_00_Auth_Alias for component BPC_bus.
2010-02-18 19:03:11    CWLDB9022I: Creating authentication alias BPCME_00_Auth_Alias for component BPC_bus.
2010-02-18 19:03:16    CWLDB9021I: Datasource Business Process Choreographer ME data source is configured at scope Cluster=b8
2010-02-18 19:05:50    CWLDB9013I: Configuring component #WBI_BPCEventCollector_0 on deployment target b8sr1.
2010-02-18 19:05:50    CWLDB9022I: Creating authentication alias BPCEDB_b8sr1_Auth_Alias for component #WBI_BPCEventCollecto
2010-02-18 19:05:55    CWLDB9021I: Datasource Business Process Choreographer reporting function source is configured at scope C
2010-02-18 19:06:10    CWLDB9013I: Configuring component #WBI_BPCExplorer_0 on deployment target b8sr1.
2010-02-18 19:06:56    CWLDB9013I: Configuring component #WBI_BRM_0 on deployment target b8sr1.
2010-02-18 19:06:58    CWLDB9013I: Configuring component #WBI_RECOVERY_0 on deployment target b8sr1.
```

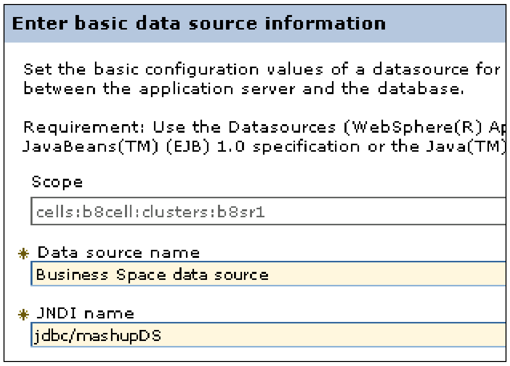
Figure 5-67 Deployment environment generation result

5.6.4 Posting deployment environment generation steps

After successfully generating the deployment environment you must install Business Space powered by WebSphere manually. Before installing Business Space powered by WebSphere, you must create a separate data source for Business Space powered by WebSphere; otherwise, the WBI_Datasource is used.

Creating a separate Business Space powered by WebSphere data source

1. In the ISC, click Resources → JDBC → Data sources. In the Scope field, select **Cluster=b8sr1**, and click **New**, as shown in Figure 5-68.
2. In the Data source name field, type `Business Space data source`. In the JNDI name field, type `jdbc/mashupDS`, as shown in Figure 5-68. Click **Next**.



Enter basic data source information

Set the basic configuration values of a datasource for
between the application server and the database.

Requirement: Use the Datasources (WebSphere(R) Ap
JavaBeans(TM) (EJB) 1.0 specification or the Java(TM)

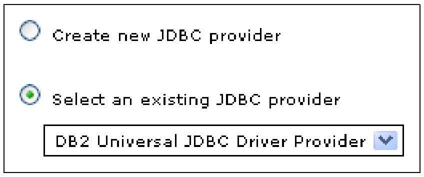
Scope
cells:b8cell:clusters:b8sr1

* Data source name
Business Space data source

* JNDI name
jdbc/mashupDS

Figure 5-68 Basic Business Space data source data

3. Select **Select an existing JDBC provider**. From the pull-down list, select **DB2 Universal JDBC Driver Provider**. Click **Next**, as shown in Figure 5-69.



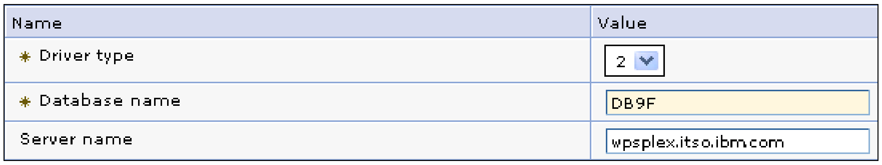
○ Create new JDBC provider

● Select an existing JDBC provider

DB2 Universal JDBC Driver Provider

Figure 5-69 Select DB2 JDBC provider

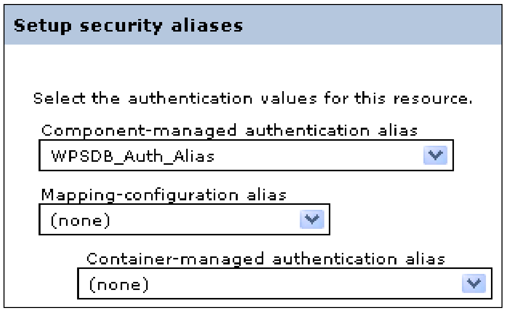
4. Select **driver type 2**. In the Database name field, type the DB2 location name for the database name. In the Server name field, enter the dynamic VIPA of your DB2 data sharing group, as shown in Figure 5-70. Click **Next**.



Name	Value
* Driver type	2
* Database name	DB9F
Server name	wpsplex.itso.ibm.com

Figure 5-70 Select driver data

5. From the Component-managed authentication alias field, select **WPSDB_Auth_Alias**, as shown in Figure 5-71. Click **Next**.



Setup security aliases

Select the authentication values for this resource.

Component-managed authentication alias: WPSDB_Auth_Alias

Mapping-configuration alias: (none)

Container-managed authentication alias: (none)

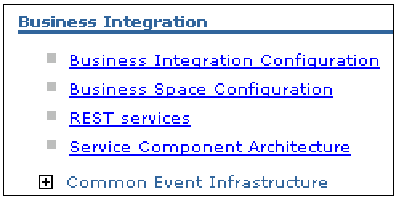
Figure 5-71 Setup security alias

6. Check the summary, and click **Finish** to create the data source. After the successful creation, save the changes to the master repository.

Installing Business Space powered by WebSphere

After you create the Business Space powered by WebSphere specific data source you can install Business Space powered by WebSphere:

1. Click **Servers** → **Clusters** → **WebSphere application server clusters**.
2. Click your cluster. In the cluster detail view, go to the Business Integration section, and click **Business Space Configuration**, as shown in Figure 5-72.



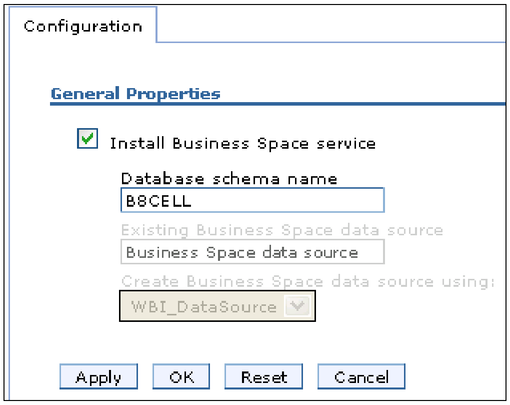
Business Integration

- Business Integration Configuration
- Business Space Configuration
- REST services
- Service Component Architecture

⊞ Common Event Infrastructure

Figure 5-72 Business Integration section

3. In Figure 5-73, select **Install Business Space service**. In the Database schema name field, enter the schema name as defined in your configuration spreadsheet. In our sample, it is B8CELL. The data source is your previously created data source. Click **OK**.



Configuration

General Properties

☑ Install Business Space service

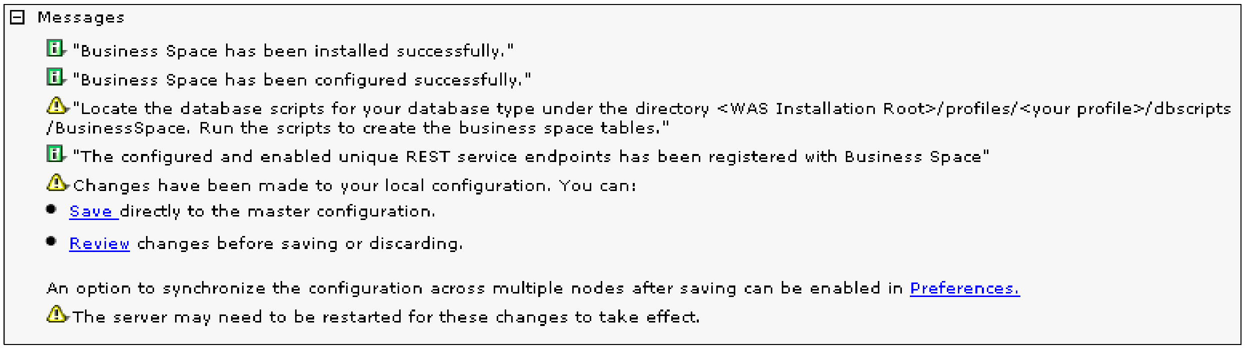
Database schema name: B8CELL

Existing Business Space data source: Business Space data source

Create Business Space data source using: WBI_DataSource

Figure 5-73 Business Space powered by WebSphere installation wizard

4. After a successful configuration, as shown in Figure 5-74, save the changes to the master repository.



⊟ Messages

"Business Space has been installed successfully."

"Business Space has been configured successfully."

"Locate the database scripts for your database type under the directory <WAS Installation Root>/profiles/<your profile>/dbscripts /BusinessSpace. Run the scripts to create the business space tables."

"The configured and enabled unique REST service endpoints has been registered with Business Space"

Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

An option to synchronize the configuration across multiple nodes after saving can be enabled in Preferences.

The server may need to be restarted for these changes to take effect.

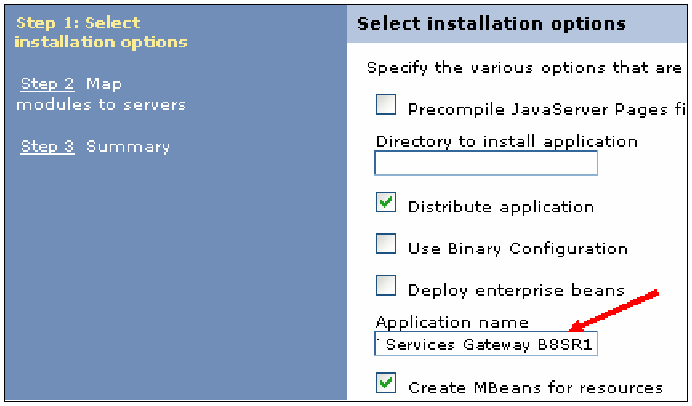
Figure 5-74 Installation completed

Installing REST Services Gateway

You must configure the REST Services Gateway in the cluster. Although there is already a REST Services Gateway Dmgr that is installed, you must install this application again because you cannot map the existing application to an additional target without loosing the mapping to the deployment manager. To install REST Services Gateway:

1. In the ISC, click **Applications** → **New application**.
2. In the New Application view, click **New Enterprise Application**.
3. Select the **Remote file system** option, and click **Browse**.
4. Click the **deployment manager node**. On the root directory "/", navigate the directory path until you locate the installableApps directory:

   /wasv7config/b8cell/b8dmnode/DeplyomentManager/installableApps
5. In the installableApps directory, select **RESTServiceGateway.ear** file, and click **OK**.
6. Click **Next**, select the **Fast Path** option, and click **Next** again.
7. In Figure 5-75, you can rename the application name if you want to have this application installed for each cluster separately, for example, REST Services Gateway B8SR1.



Step 1: Select installation options
Step 2 Map modules to servers
Step 3 Summary

Select installation options

Specify the various options that are

☐ Precompile JavaServer Pages fi

Directory to install application

☑ Distribute application

☐ Use Binary Configuration

☐ Deploy enterprise beans

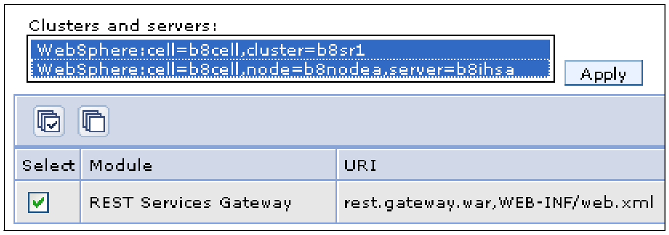
Application name

Services Gateway B8SR1

☑ Create MBeans for resources

Figure 5-75 Rename application

8. Click **Next**, and select the cluster and web servers as target for this application, as shown in Figure 5-76 on page 186.



Clusters and servers:

WebSphere:cell=b8cell,cluster=b8sr1
WebSphere:cell=b8cell,node=b8nodea,server=b8ihsa

Select	Module	URI
☑	REST Services Gateway	rest.gateway.war,WEB-INF/web.xml

Figure 5-76 Clusters and servers

9. Click **Apply**, and click **Next**.
10. In the summary panel, verify the values, and click **Finish**. The application is installed.
11. After the application is installed successfully, save the changes to the master repository.

Now you must make the same adjustments to the configured resources as though using the pattern-based deployment environment wizard. For this go to 5.3, "Making adjustments to the configured resources" on page 139.

5.7 Optional: Configuring business process choreographer with WebSphere MQ for z/OS as messaging provider

If you can use WebSphere MQ as a message provider, the potential bottleneck of running one message engine in an adjunct disappears. For this reason, if your applications make heavy use of messaging, it is more advantageous to use WebSphere MQ as a message provider than to split the message engines into a separate cluster or consider partitioning.

There are four busses in our environment and many queues that are defined in support of them. We chose to focus our efforts on the bpc.b8cell.bus.

There are 10 queues that are defined for the BPC bus. The resources were created by the BPC installation process and use the WebSphere Default Messaging Provider. We suggest that you navigate through these resource definitions using the Integrated Solutions Console, and save the values for reference. We are creating new queues using MQ shared queues.

Topic spaces: Topic spaces cannot be configured on WebSphere MQ queue managers.

In the steps in this section, we detail how to use the installed WebSphere MQ Messaging Provider to set up the BPC and HTC queues to use our shared MQ infrastructure. This set up involves defining multiple resources and changing our BPC bus to use them. There is a fair amount of information about this topic available. Your best resources are the WebSphere Application Server for z/OS Information Center and this publication.

To set up:

1. Ensure that the MQ libraries are specified in your JCL started PROCs for the servers. See Example 5-10. You must update the JCL for the controllers, adjuncts, and servants.

Example 5-10 Server B8ASRS JCL from SYS1.PROCLIB

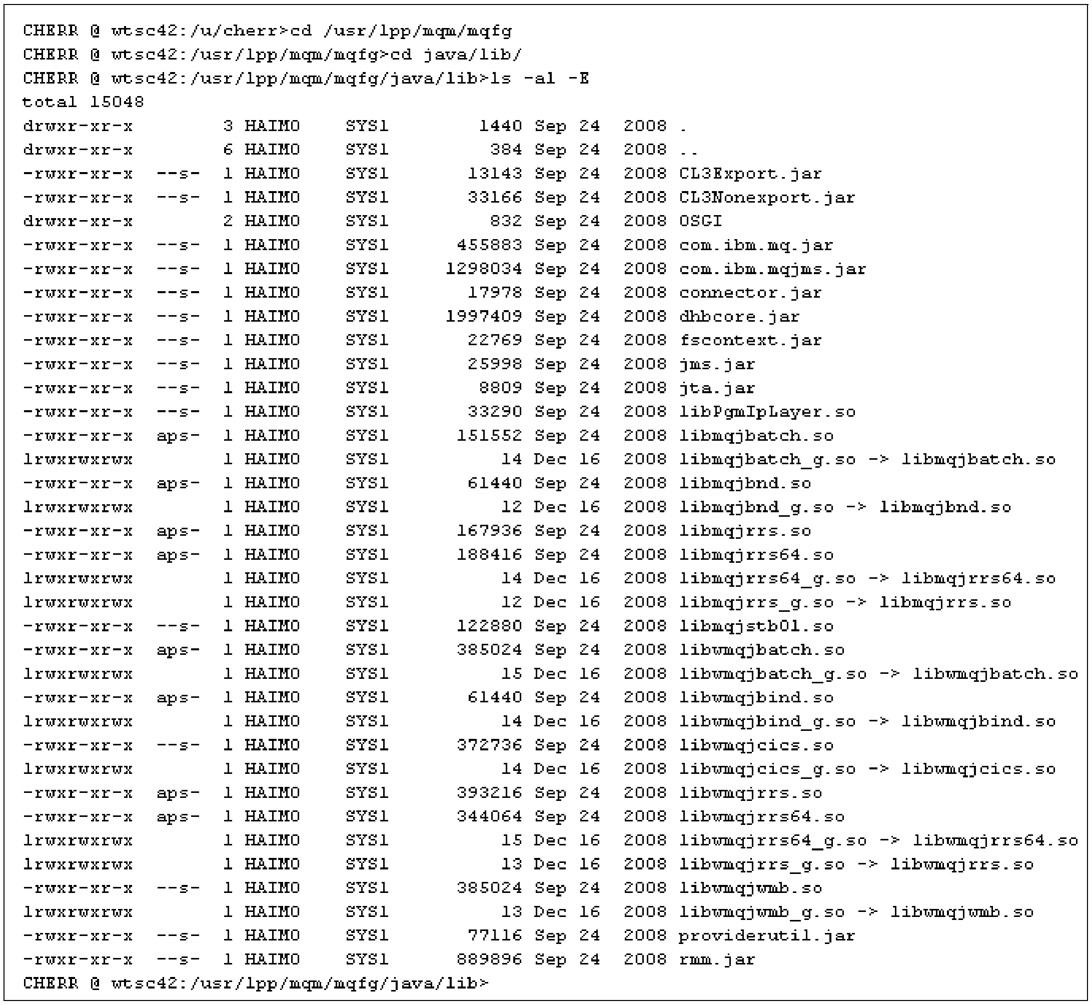
```
//STEPLIB  DD DISP=SHR,DSN=BBO6153.MDNODEB.SBBOLD2
//         DD DISP=SHR,DSN=BBO6153.MDNODEB.SBBOLOAD
//         DD DISP=SHR,DSN=BBO6153.MDNODEB.SBBGLOAD
//         DD DISP=SHR,DSN=DB9F9.SDSNEXIT
//         DD DISP=SHR,DSN=DB9F9.SDSNLOAD
//         DD DISP=SHR,DSN=DB9F9.SDSNLOD2
//         DD DISP=SHR,DSN=MQ600.SCSQANLE
//         DD DISP=SHR,DSN=MQ600.SCSQAUTH
```

2. Ensure that the program attribute bit is set for the JMS runtime libraries.

   Our JMS libraries are mounted at /usr/lpp/mqm/mqfg:

   a. Navigate to the /usr/lpp/mqm/mqfg/java/lib directory.

   b. Ensure that the +a and +p bits are turned on for all jms files that end in .so. See Figure 5-77 for an example of how to do this.



```
CHERR @ wtsc42:/u/cherr>cd /usr/lpp/mqm/mqfg
CHERR @ wtsc42:/usr/lpp/mqm/mqfg>cd java/lib/
CHERR @ wtsc42:/usr/lpp/mqm/mqfg/java/lib>ls -al -E
total 15048
drwxr-xr-x         3 HAIMO    SYS1        1440 Sep 24  2008 .
drwxr-xr-x         6 HAIMO    SYS1         384 Sep 24  2008 ..
-rwxr-xr-x  --s-   1 HAIMO    SYS1       13143 Sep 24  2008 CL3Export.jar
-rwxr-xr-x  --s-   1 HAIMO    SYS1       33166 Sep 24  2008 CL3Nonexport.jar
drwxr-xr-x         2 HAIMO    SYS1         832 Sep 24  2008 OSGI
-rwxr-xr-x  --s-   1 HAIMO    SYS1      455883 Sep 24  2008 com.ibm.mq.jar
-rwxr-xr-x  --s-   1 HAIMO    SYS1     1298034 Sep 24  2008 com.ibm.mqjms.jar
-rwxr-xr-x  --s-   1 HAIMO    SYS1       17978 Sep 24  2008 connector.jar
-rwxr-xr-x  --s-   1 HAIMO    SYS1     1997409 Sep 24  2008 dhbcore.jar
-rwxr-xr-x  --s-   1 HAIMO    SYS1       22769 Sep 24  2008 fscontext.jar
-rwxr-xr-x  --s-   1 HAIMO    SYS1       25998 Sep 24  2008 jms.jar
-rwxr-xr-x  --s-   1 HAIMO    SYS1        8809 Sep 24  2008 jta.jar
-rwxr-xr-x  --s-   1 HAIMO    SYS1       33290 Sep 24  2008 libPgmIpLayer.so
-rwxr-xr-x  aps-   1 HAIMO    SYS1      151552 Sep 24  2008 libmqjbatch.so
lrwxrwxrwx         1 HAIMO    SYS1          14 Dec 16  2008 libmqjbatch_g.so -> libmqjbatch.so
-rwxr-xr-x  aps-   1 HAIMO    SYS1       61440 Sep 24  2008 libmqjbnd.so
lrwxrwxrwx         1 HAIMO    SYS1          12 Dec 16  2008 libmqjbnd_g.so -> libmqjbnd.so
-rwxr-xr-x  aps-   1 HAIMO    SYS1      167936 Sep 24  2008 libmqjrrs.so
-rwxr-xr-x  aps-   1 HAIMO    SYS1      188416 Sep 24  2008 libmqjrrs64.so
lrwxrwxrwx         1 HAIMO    SYS1          14 Dec 16  2008 libmqjrrs64_g.so -> libmqjrrs64.so
lrwxrwxrwx         1 HAIMO    SYS1          12 Dec 16  2008 libmqjrrs_g.so -> libmqjrrs.so
-rwxr-xr-x  --s-   1 HAIMO    SYS1      122880 Sep 24  2008 libmqjstb01.so
-rwxr-xr-x  aps-   1 HAIMO    SYS1      385024 Sep 24  2008 libwmqjbatch.so
lrwxrwxrwx         1 HAIMO    SYS1          15 Dec 16  2008 libwmqjbatch_g.so -> libwmqjbatch.so
-rwxr-xr-x  aps-   1 HAIMO    SYS1       61440 Sep 24  2008 libwmqjbind.so
lrwxrwxrwx         1 HAIMO    SYS1          14 Dec 16  2008 libwmqjbind_g.so -> libwmqjbind.so
-rwxr-xr-x  --s-   1 HAIMO    SYS1      372736 Sep 24  2008 libwmqjcics.so
lrwxrwxrwx         1 HAIMO    SYS1          14 Dec 16  2008 libwmqjcics_g.so -> libwmqjcics.so
-rwxr-xr-x  aps-   1 HAIMO    SYS1      393216 Sep 24  2008 libwmqjrrs.so
-rwxr-xr-x  aps-   1 HAIMO    SYS1      344064 Sep 24  2008 libwmqjrrs64.so
lrwxrwxrwx         1 HAIMO    SYS1          15 Dec 16  2008 libwmqjrrs64_g.so -> libwmqjrrs64.so
lrwxrwxrwx         1 HAIMO    SYS1          13 Dec 16  2008 libwmqjrrs_g.so -> libwmqjrrs.so
-rwxr-xr-x  --s-   1 HAIMO    SYS1      385024 Sep 24  2008 libwmqjwmb.so
lrwxrwxrwx         1 HAIMO    SYS1          13 Dec 16  2008 libwmqjwmb_g.so -> libwmqjwmb.so
-rwxr-xr-x  --s-   1 HAIMO    SYS1       77116 Sep 24  2008 providerutil.jar
-rwxr-xr-x  --s-   1 HAIMO    SYS1      889896 Sep 24  2008 rmm.jar
CHERR @ wtsc42:/usr/lpp/mqm/mqfg/java/lib>
```

Figure 5-77 Verify WebSphere MQ runtime libraries

3. Create a new MQ server for our shared queue implementation.

4. In the Integrated Solutions Console, perform the following steps:

   a. Navigate to **Servers WebSphere MQ Servers**.

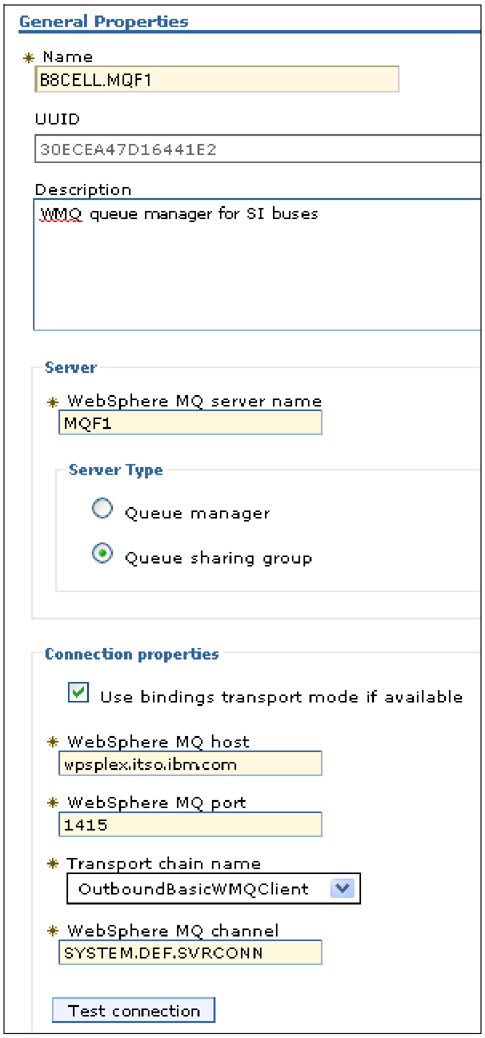
   b. Click **New**.

   c. Specify the following parameters:

      - Name: **B8CELL.MQF1**

- Description: **MQ shared queue**
- WebSphere MQ server name: **MQF1**
- Server Type: **Queue sharing group**
- Select **Use bindings transport mode if available**
- WebSphere MQ host: **wpsplex.itso.ibm.com**
- WebSphere MQ port: **1415**

Accept default values for the remainder of the fields, as shown in Figure 5-78.



General Properties

* Name
B8CELL.MQF1

UUID
30ECEA47D16441E2

Description
WMQ queue manager for SI buses

Server

* WebSphere MQ server name
MQF1

Server Type

○ Queue manager

● Queue sharing group

Connection properties

☑ Use bindings transport mode if available

* WebSphere MQ host
wpsplex.itso.ibm.com

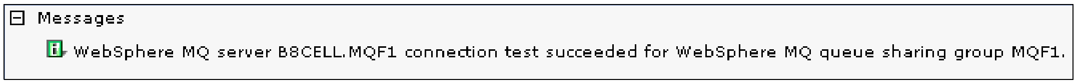
* WebSphere MQ port
1415

* Transport chain name
OutboundBasicWMQClient

* WebSphere MQ channel
SYSTEM.DEF.SVRCONN

Figure 5-78 WMQ queue manager definition

d. Click **Test Connection**. The result is a good connection to the MQ shared queue specified. See Figure 5-79.

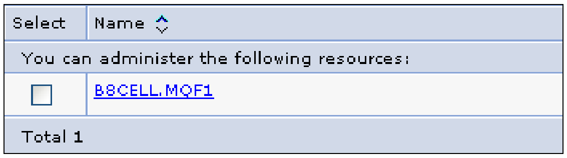


Messages

WebSphere MQ server B8CELL.MQF1 connection test succeeded for WebSphere MQ queue sharing group MQF1.

Figure 5-79 Test WMQ connection

e. Click **OK**, and save the changes to the master repository. See Figure 5-80.

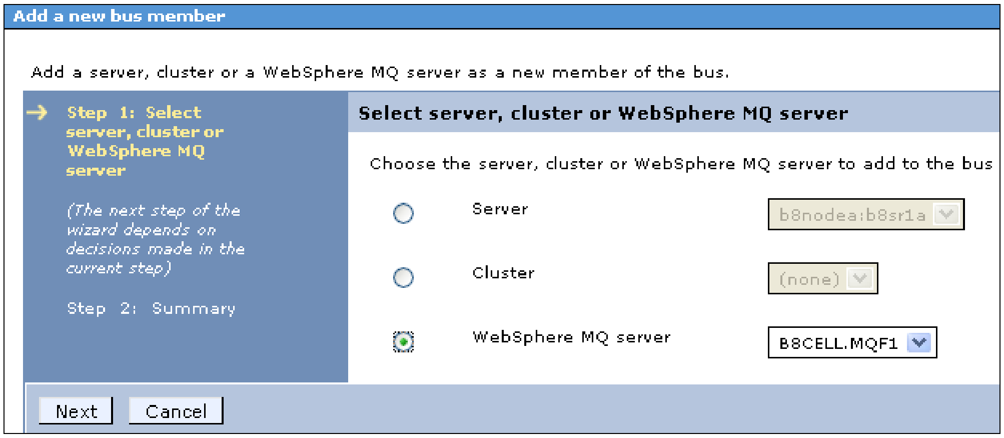


Select	Name ⇕
You can administer the following resources:	
□	B8CELL.MQF1
Total 1	

Figure 5-80 Configured WMQ queue manager

5. Add a new bus member to the BPC.b8cell.bus.
6. In the Integrated Solutions Console, perform the following steps:

a. Navigate to **Service Integration Buses**.
b. Click **BPC.b8cell.bus.**
c. Click **Bus Members**.
d. Click **Add**.
e. Select **WebSphere MQ server** and **B8CELL.MQF1** from the pull-down menu, as shown in Figure 5-81.



Add a new bus member

Add a server, cluster or a WebSphere MQ server as a new member of the bus.

→ **Step 1: Select server, cluster or WebSphere MQ server**

(The next step of the wizard depends on decisions made in the current step)

Step 2: Summary

Select server, cluster or WebSphere MQ server

Choose the server, cluster or WebSphere MQ server to add to the bus

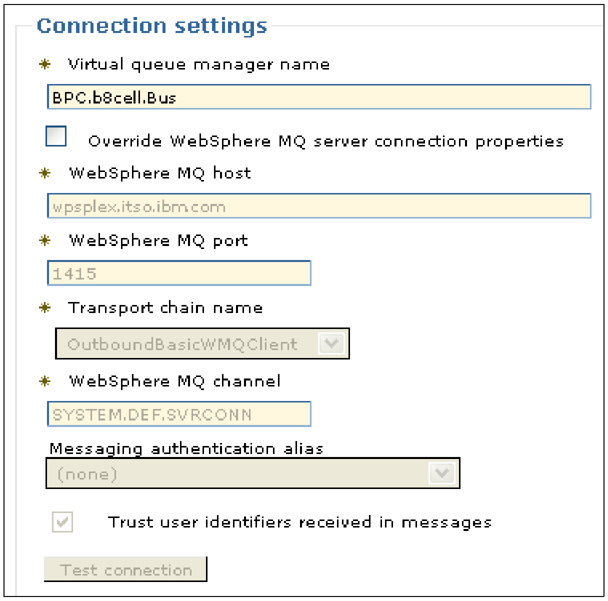
○ Server b8nodea:b8sr1a

○ Cluster (none)

● WebSphere MQ server B8CELL.MQF1

Figure 5-81 Add WMQ queue manager to BPC bus

f. Click **Next**, and the Connection settings panel is displayed, as shown in Figure 5-82.



Connection settings

* Virtual queue manager name

BPC.b8cell.Bus

□ Override WebSphere MQ server connection properties

* WebSphere MQ host

wpsplex.itso.ibm.com

* WebSphere MQ port

1415

* Transport chain name

OutboundBasicWMQClient

* WebSphere MQ channel

SYSTEM.DEF.SVRCONN

Messaging authentication alias

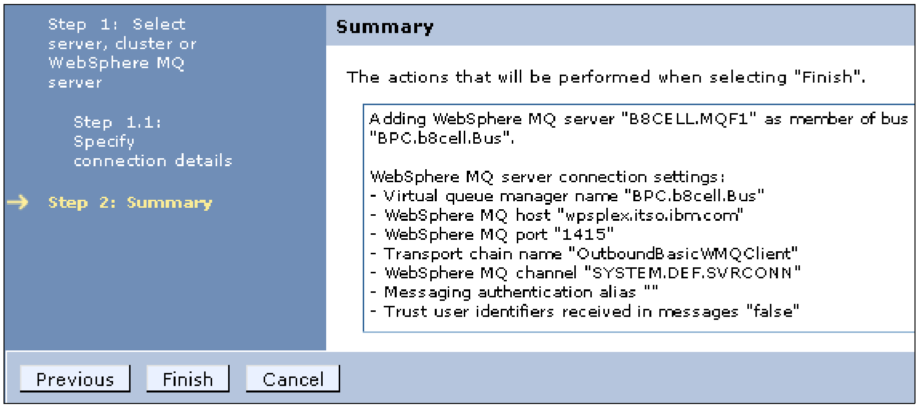
(none)

☑ Trust user identifiers received in messages

Test connection

Figure 5-82 Specify bus member parameters

g. Click **Next**. See Figure 5-83.



Step 1: Select server, cluster or WebSphere MQ server

Step 1.1: Specify connection details

→ Step 2: Summary

Summary

The actions that will be performed when selecting "Finish".

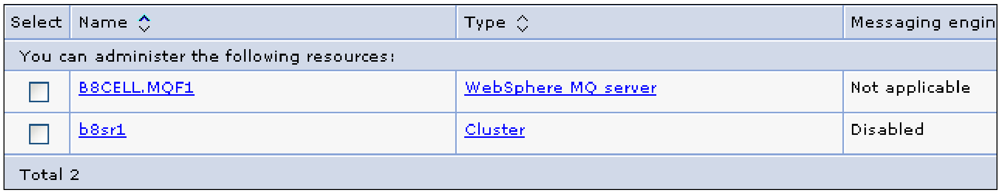
Adding WebSphere MQ server "B8CELL.MQF1" as member of bus "BPC.b8cell.Bus".

WebSphere MQ server connection settings:
- Virtual queue manager name "BPC.b8cell.Bus"
- WebSphere MQ host "wpsplex.itso.ibm.com"
- WebSphere MQ port "1415"
- Transport chain name "OutboundBasicWMQClient"
- WebSphere MQ channel "SYSTEM.DEF.SVRCONN"
- Messaging authentication alias ""
- Trust user identifiers received in messages "false"

Previous | Finish | Cancel

Figure 5-83 Summary of WMQ queue manager bus member

h. Click **Finish**, and save the changes to the master repository. See Figure 5-84.



Select	Name ⇕	Type ⇕	Messaging engin
You can administer the following resources:			
□	B8CELL.MQF1	WebSphere MQ server	Not applicable
□	b8sr1	Cluster	Disabled
Total 2			

Figure 5-84 BPC.b8cell.bus bus member list

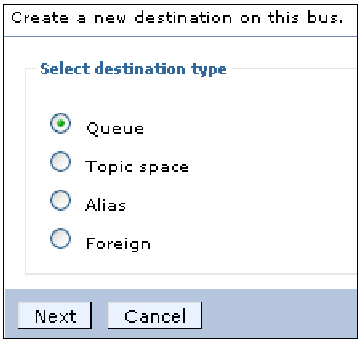
7. Add the queues of the BPC bus to the to WMQ bus member.

8. In the Integrated Solutions Console, perform the following steps:

   a. Navigate to **Service Integration Buses**.

   b. Click **BPC.b8cell.bus**.

c. Click **Destinations**.
d. Click **New**.
e. Select **Queue**, as shown in Figure 5-85.



Create a new destination on this bus.

Select destination type

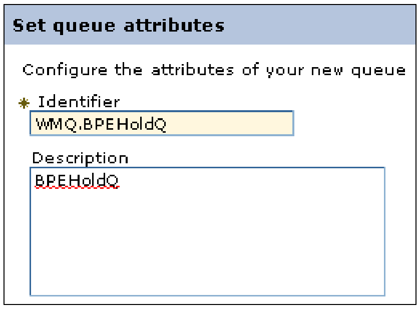
● Queue
○ Topic space
○ Alias
○ Foreign

Figure 5-85 Create a new destination

f. Click **Next**.

g. Specify the following values in the Set queue attributes panel, which we show in Figure 5-86:

- Identifier: **WMQ.BPEHoldQ**.
- Description: **BPEHoldQ**.



Set queue attributes

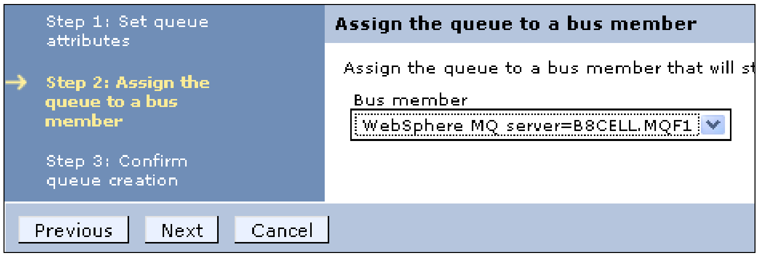
Configure the attributes of your new queue

* Identifier: WMQ.BPEHoldQ

Description: BPEHoldQ

Figure 5-86 Queue destination name

h. Click **Next**.
i. Specify Bus Member: **WebSphere MQ server = B8CELL.MQF1**, as shown in Figure 5-87.



Step 1: Set queue attributes

→ Step 2: Assign the queue to a bus member

Step 3: Confirm queue creation

Assign the queue to a bus member

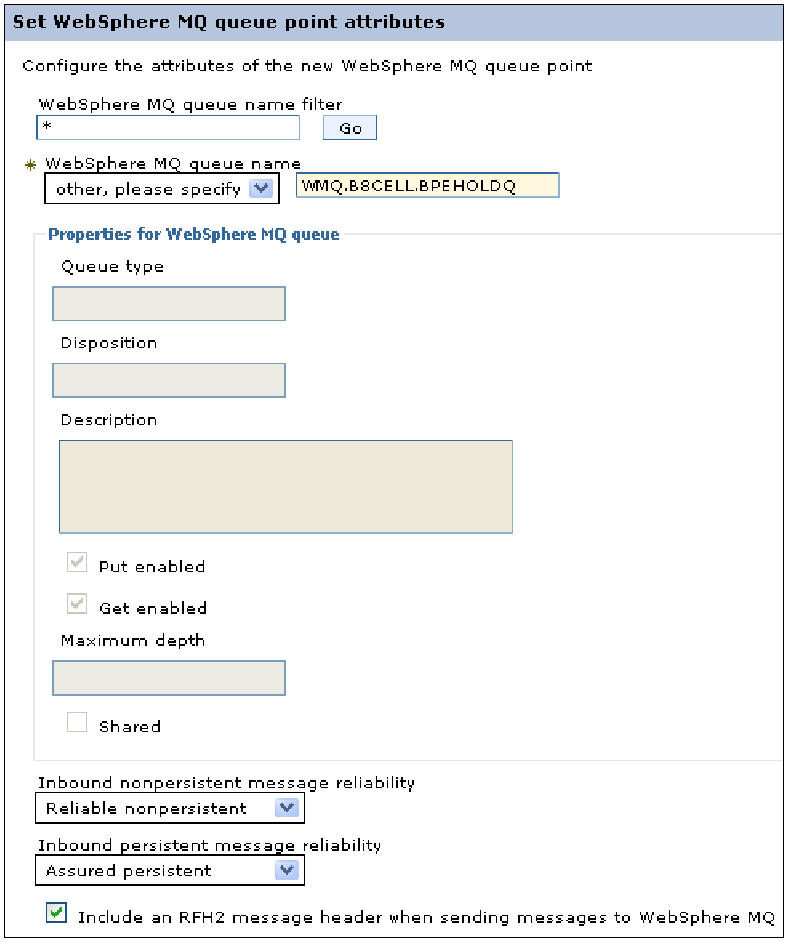
Assign the queue to a bus member that will st

Bus member: WebSphere MQ server=B8CELL.MQF1

Figure 5-87 Assign destination to bus member

j. Click **Next**.

k. In the WebSphere MQ queue name field, select **other, please specify** from the pull-down menu. Type `WMQ.B8CELL.BPEHOLDQ` in the field, as shown in Figure 5-88.



Set WebSphere MQ queue point attributes

Configure the attributes of the new WebSphere MQ queue point

WebSphere MQ queue name filter: *

* WebSphere MQ queue name: other, please specify — WMQ.B8CELL.BPEHOLDQ

Properties for WebSphere MQ queue

Queue type

Disposition

Description

☑ Put enabled

☑ Get enabled

Maximum depth

□ Shared

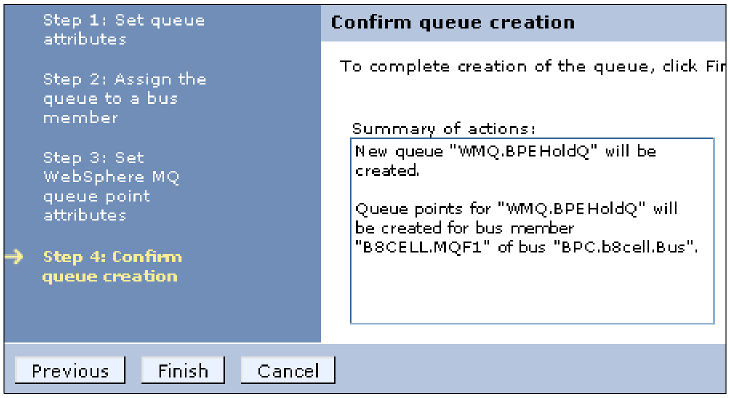
Inbound nonpersistent message reliability: Reliable nonpersistent

Inbound persistent message reliability: Assured persistent

☑ Include an RFH2 message header when sending messages to WebSphere MQ

Figure 5-88 Define corresponding WMQ queue

l. Click **Next**. The Confirm queue creation window is displayed, as shown in Figure 5-89. Review the Summary of actions.



Step 1: Set queue attributes
Step 2: Assign the queue to a bus member
Step 3: Set WebSphere MQ queue point attributes
→ Step 4: Confirm queue creation
Previous
Finish
Cancel

Confirm queue creation

To complete creation of the queue, click Fi

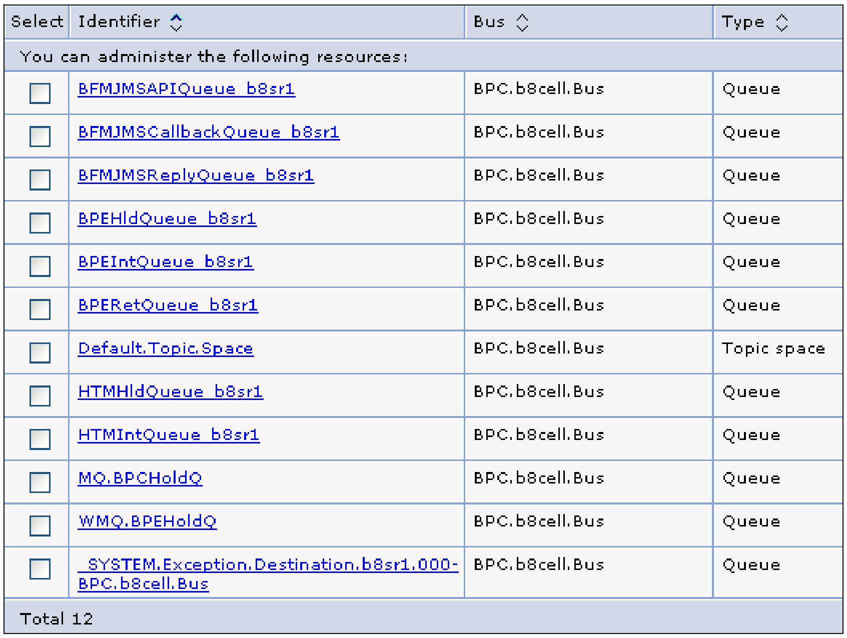
Summary of actions:

New queue "WMQ.BPEHoldQ" will be created.

Queue points for "WMQ.BPEHoldQ" will be created for bus member "B8CELL.MQF1" of bus "BPC.b8cell.Bus".

Figure 5-89 Review parameters

m. Click **Finish** → **Save**. See Figure 5-90 on page 193.

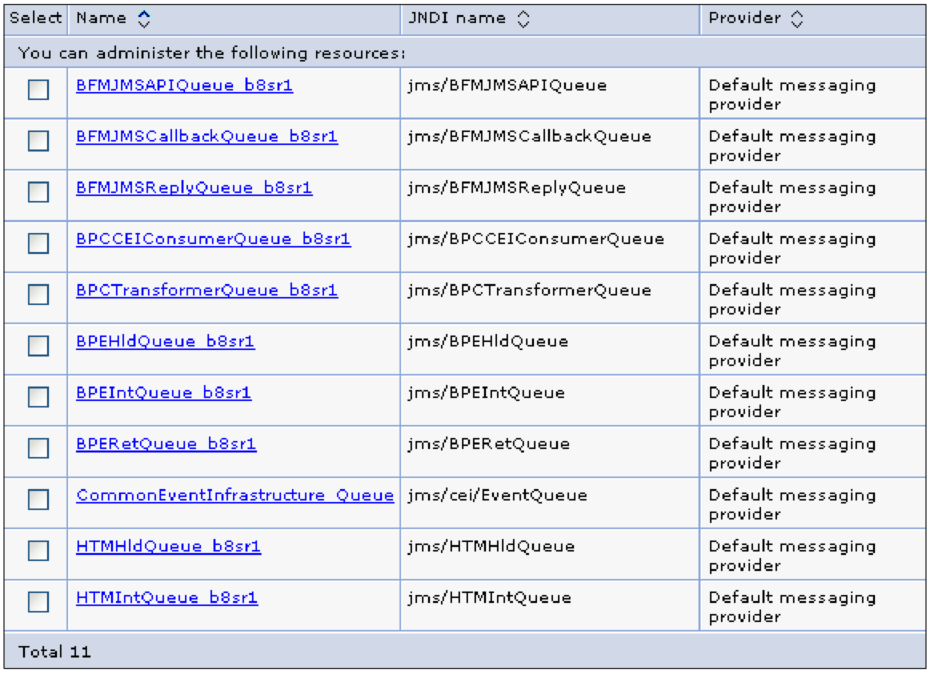


Select	Identifier	Bus	Type
You can administer the following resources:			
□	BFMJMSAPIQueue_b8sr1	BPC.b8cell.Bus	Queue
□	BFMJMSCallbackQueue_b8sr1	BPC.b8cell.Bus	Queue
□	BFMJMSReplyQueue_b8sr1	BPC.b8cell.Bus	Queue
□	BPEHldQueue_b8sr1	BPC.b8cell.Bus	Queue
□	BPEIntQueue_b8sr1	BPC.b8cell.Bus	Queue
□	BPERetQueue_b8sr1	BPC.b8cell.Bus	Queue
□	Default.Topic.Space	BPC.b8cell.Bus	Topic space
□	HTMHldQueue_b8sr1	BPC.b8cell.Bus	Queue
□	HTMIntQueue_b8sr1	BPC.b8cell.Bus	Queue
□	MQ.BPCHoldQ	BPC.b8cell.Bus	Queue
□	WMQ.BPEHoldQ	BPC.b8cell.Bus	Queue
□	_SYSTEM.Exception.Destination.b8sr1.000-BPC.b8cell.Bus	BPC.b8cell.Bus	Queue
Total 12			

Figure 5-90 New destination created

You should now have a WMQ shared queue destination defined for the BPC.b8cell.bus message bus.

9. Change the existing JMS resource definition for the corresponding BPE queue to use our new WMQ queue.
10. In the Integrated Solutions Console, perform the following steps:
    a. Navigate to **Resources JMS**.
    b. Click **Queues**.

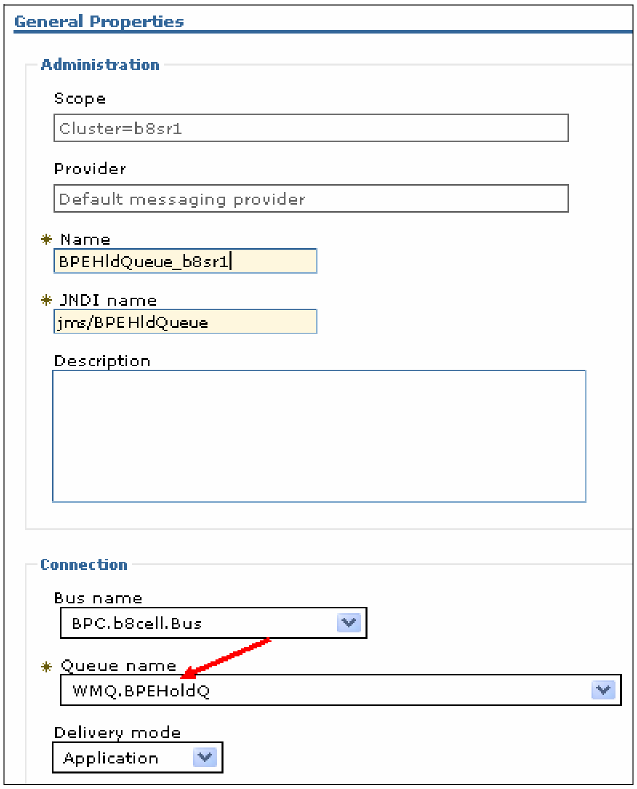


Select	Name	JNDI name	Provider
You can administer the following resources:			
□	BFMJMSAPIQueue_b8sr1	jms/BFMJMSAPIQueue	Default messaging provider
□	BFMJMSCallbackQueue_b8sr1	jms/BFMJMSCallbackQueue	Default messaging provider
□	BFMJMSReplyQueue_b8sr1	jms/BFMJMSReplyQueue	Default messaging provider
□	BPCCEIConsumerQueue_b8sr1	jms/BPCCEIConsumerQueue	Default messaging provider
□	BPCTransformerQueue_b8sr1	jms/BPCTransformerQueue	Default messaging provider
□	BPEHldQueue_b8sr1	jms/BPEHldQueue	Default messaging provider
□	BPEIntQueue_b8sr1	jms/BPEIntQueue	Default messaging provider
□	BPERetQueue_b8sr1	jms/BPERetQueue	Default messaging provider
□	CommonEventInfrastructure_Queue	jms/cei/EventQueue	Default messaging provider
□	HTMHldQueue_b8sr1	jms/HTMHldQueue	Default messaging provider
□	HTMIntQueue_b8sr1	jms/HTMIntQueue	Default messaging provider
Total 11			

Figure 5-91 JMS queue list

c. Select **BPEHldQueue_b8sr1**.

d. In the Queue name field, change the value from **BPEHldQueue_b8sr1_WPS**, shown in Figure 5-92 to **WMQ.BPEHoldQ**.



General Properties

Administration

Scope
Cluster=b8sr1

Provider
Default messaging provider

* Name
BPEHldQueue_b8sr1

* JNDI name
jms/BPEHldQueue

Description

Connection

Bus name
BPC.b8cell.Bus

* Queue name
WMQ.BPEHoldQ

Delivery mode
Application

Figure 5-92 Select WMQ queue destination

11. Click **OK** → **Save**. You completed configuration and changing of the first BPC bus queues. Repeat steps 8 to 11 for all remaining BPC JMS queues to use WMQ-shared queues instead of the WebSphere Default messaging provider.

12. Create the WMQ queues as *shared queues* in your WMQ queue sharing group. They are not automatically created when saving the changes to the master repository.

6

Verifying the configuration

In this chapter, we describe how to verify your WebSphere Process Server for z/OS topology. Specifically, we discuss:

6.1 Starting the cluster and web server

Installation verification consists of two parts:

1. Verifying resource configurations.
2. Using test applications for testing WebSphere functionality.

Before verifying your environment, stop and restart the complete cell to make sure every configuration change becomes active in the cell. Some resource modifications are ignored by the servers until they are restarted.

> **Note:** The verification description in this chapter was done in cell b8cell, which is equal to cell b7cell.

6.2 Verifying configured resources and applications

In this section, we provide verification steps for resources and applications.

6.2.1 Reviewing error messages and repair

When the server finishes initialization, look in the server's log files for any error messages, and examine the Integrated Solutions Console for the settings of various components. Looking for the messages is quite easy if you know what you are looking for.

Examining the server logs on JES spool

To examine the server logs on JES spool:

1. Using the SDSF Display Active window, select your application server's started tasks with the **`PREfix`** or **`Select B8*`** commands, and look in the control region (B8SR1A), control region adjunct (B8SR1AA), and servant regions (B8SR1S).
2. Select each with the S action character, and look for any sever error messages with the **`FIND SEVERE ALL`** command. Hopefully, there will be none. If there are, note the number displayed by SDSF, and step through each one and start your problem determination process.

> **Note:** There are one or two severe messages relating to TCP/IP channels at startup. These messages are usually safe to ignore after all of the address spaces fully initialized.
>
> ```
> CHFW0030E: Error starting chain _InboundTCPProxyBridgeService
> ```

Exploring with the Integrated Solutions Console

To explore with the ISC:

1. Sign on to the Integrated Solutions Console, and check the configured resources and applications:

   ```
   http://<host>:<port_number>/ibm/console
   ```

2. Perform a visual configuration check of the following resource categories:

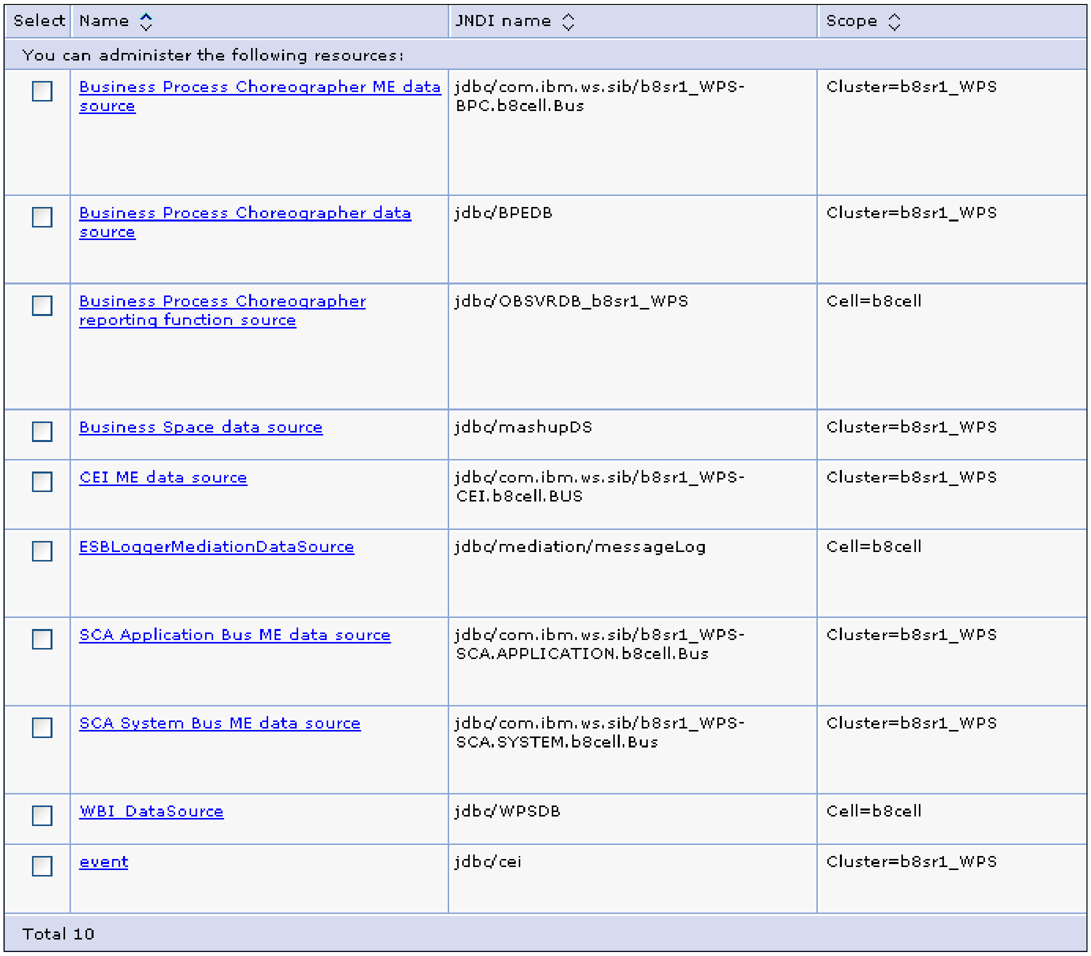
- Datasources
- Applications

- Buses
- Authentication aliases

Ensuring that the data sources are properly configured

To ensure that the data sources are properly configured:

1. Click **Resources** → **JDBC** → **Data sources**. A list similar to Figure 6-1 is displayed.



Select	Name	JNDI name	Scope
You can administer the following resources:			
□	Business Process Choreographer ME data source	jdbc/com.ibm.ws.sib/b8sr1_WPS-BPC.b8cell.Bus	Cluster=b8sr1_WPS
□	Business Process Choreographer data source	jdbc/BPEDB	Cluster=b8sr1_WPS
□	Business Process Choreographer reporting function source	jdbc/OBSVRDB_b8sr1_WPS	Cell=b8cell
□	Business Space data source	jdbc/mashupDS	Cluster=b8sr1_WPS
□	CEI ME data source	jdbc/com.ibm.ws.sib/b8sr1_WPS-CEI.b8cell.BUS	Cluster=b8sr1_WPS
□	ESBLoggerMediationDataSource	jdbc/mediation/messageLog	Cell=b8cell
□	SCA Application Bus ME data source	jdbc/com.ibm.ws.sib/b8sr1_WPS-SCA.APPLICATION.b8cell.Bus	Cluster=b8sr1_WPS
□	SCA System Bus ME data source	jdbc/com.ibm.ws.sib/b8sr1_WPS-SCA.SYSTEM.b8cell.Bus	Cluster=b8sr1_WPS
□	WBI_DataSource	jdbc/WPSDB	Cell=b8cell
□	event	jdbc/cei	Cluster=b8sr1_WPS
Total 10			

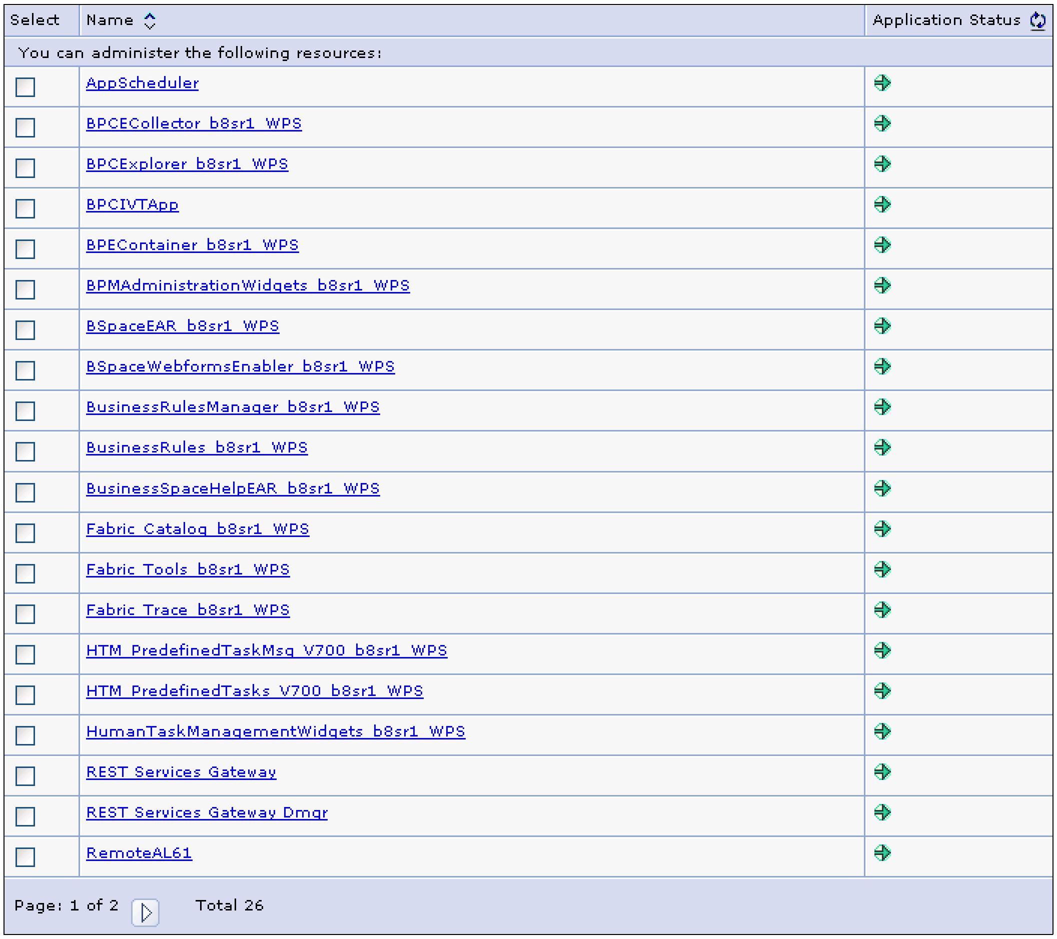
Figure 6-1 List of configured datasources

2. Click the data sources, and verify that the custom properties currentSchema and currentSQLID, are set for the messaging engine data sources. In the first instance, check the data sources WBI_Datasource, event, and CEI ME data source.

Verifying installed applications

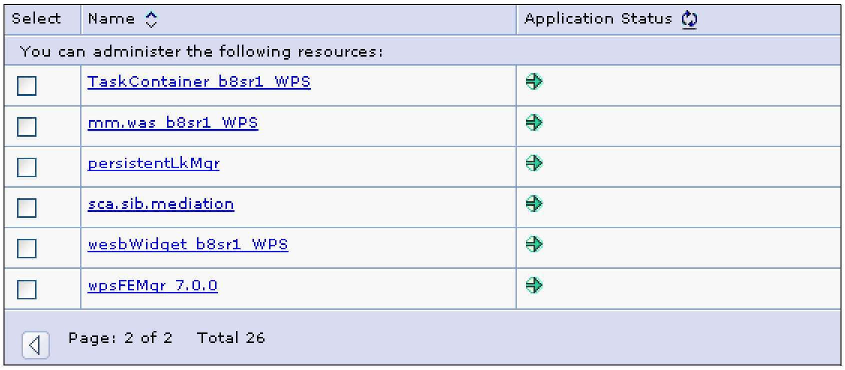
To verify the installed applications:

1. Click **Applications** → **Application Types** → **WebSphere enterprise applications**. You get a list of applications, all in the running state, as shown in Figure 6-2 on page 198 and Figure 6-3 on page 198.



Select	Name ⇕	Application Status ⟳
You can administer the following resources:		
□	AppScheduler	➔
□	BPCECollector b8sr1 WPS	➔
□	BPCExplorer b8sr1 WPS	➔
□	BPCIVTApp	➔
□	BPEContainer b8sr1 WPS	➔
□	BPMAdministrationWidgets b8sr1 WPS	➔
□	BSpaceEAR b8sr1 WPS	➔
□	BSpaceWebformsEnabler b8sr1 WPS	➔
□	BusinessRulesManager b8sr1 WPS	➔
□	BusinessRules b8sr1 WPS	➔
□	BusinessSpaceHelpEAR b8sr1 WPS	➔
□	Fabric Catalog b8sr1 WPS	➔
□	Fabric Tools b8sr1 WPS	➔
□	Fabric Trace b8sr1 WPS	➔
□	HTM PredefinedTaskMsg V700 b8sr1 WPS	➔
□	HTM PredefinedTasks V700 b8sr1 WPS	➔
□	HumanTaskManagementWidgets b8sr1 WPS	➔
□	REST Services Gateway	➔
□	REST Services Gateway Dmgr	➔
□	RemoteAL61	➔
Page: 1 of 2 ▷ Total 26		

Figure 6-2 Application list page 1

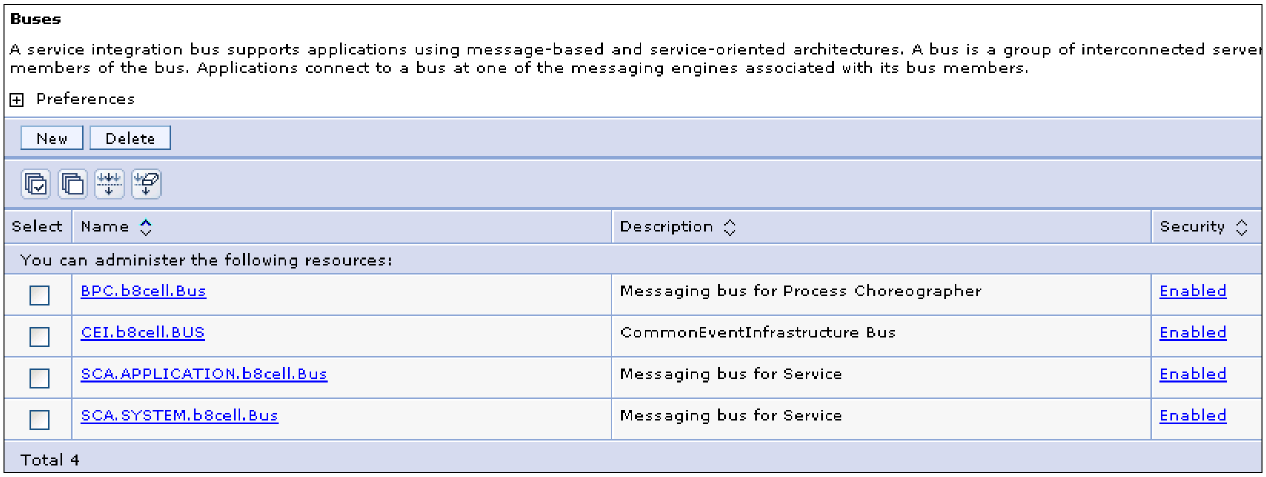


Select	Name ⇕	Application Status ⟳
You can administer the following resources:		
□	TaskContainer b8sr1 WPS	➔
□	mm.was b8sr1 WPS	➔
□	persistentLkMgr	➔
□	sca.sib.mediation	➔
□	wesbWidget b8sr1 WPS	➔
□	wpsFEMgr 7.0.0	➔
◁ Page: 2 of 2 Total 26		

Figure 6-3 Application list page 2

Verifying SI bus configuration

To check the SI bus configuration, click **Service integration** → **Buses**. A list of buses similar to Figure 6-4 on page 199 are displayed.



Buses

A service integration bus supports applications using message-based and service-oriented architectures. A bus is a group of interconnected server members of the bus. Applications connect to a bus at one of the messaging engines associated with its bus members.

⊞ Preferences

Select	Name	Description	Security
You can administer the following resources:			
□	BPC.b8cell.Bus	Messaging bus for Process Choreographer	Enabled
□	CEI.b8cell.BUS	CommonEventInfrastructure Bus	Enabled
□	SCA.APPLICATION.b8cell.Bus	Messaging bus for Service	Enabled
□	SCA.SYSTEM.b8cell.Bus	Messaging bus for Service	Enabled
Total 4			

Figure 6-4 Configured SI buses

Checking the messaging engines

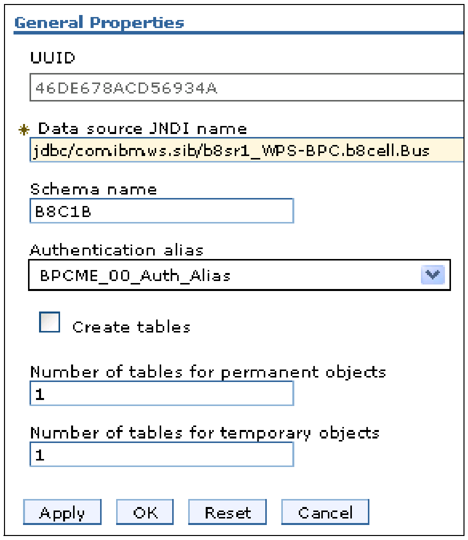
To check the status and configuration of the messaging engines:

1. Click a bus in the list.
2. In the details view of the selected bus, click **Messaging engines** to look for their runtime status. If a green arrow shows under the status column, the status is good.

Verifying the ME data store settings

To verify the ME data store settings:

1. Click the **messaging engine**. In the detail view of the messaging engine, click **Message store**.
2. Review the configuration of the message store. Especially pay attention to the schema name and the authentication alias, as shown in Figure 6-5.



General Properties

UUID
46DE678ACD56934A

* Data source JNDI name
jdbc/com.ibm.ws.sib/b8sr1_WPS-BPC.b8cell.Bus

Schema name
B8C1B

Authentication alias
BPCME_00_Auth_Alias

□ Create tables

Number of tables for permanent objects
1

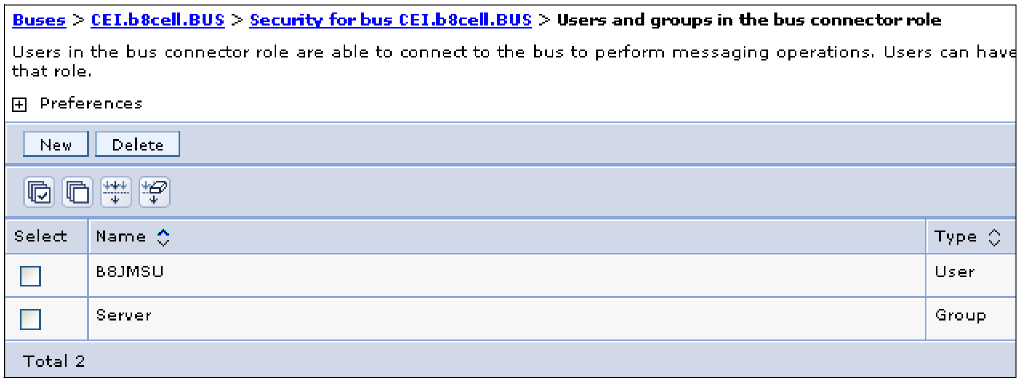
Number of tables for temporary objects
1

Figure 6-5 Messaging engine message store details

Verifying messaging engines connector roles

To verify messaging engines connector roles:

1. Return to the bus detail view. If security is enabled for the bus, in the Additional Properties section, click **Security** → **Users and groups in the bus connector role**.
2. Verify that the correct JMS user ID is used for type User, as shown in Figure 6-6.



Buses > CEI.b8cell.BUS > Security for bus CEI.b8cell.BUS > **Users and groups in the bus connector role**

Users in the bus connector role are able to connect to the bus to perform messaging operations. Users can have that role.

⊞ Preferences

Select	Name	Type
□	B8JMSU	User
□	Server	Group
Total 2		

Figure 6-6 Users and groups in bus connector role for CEI bus

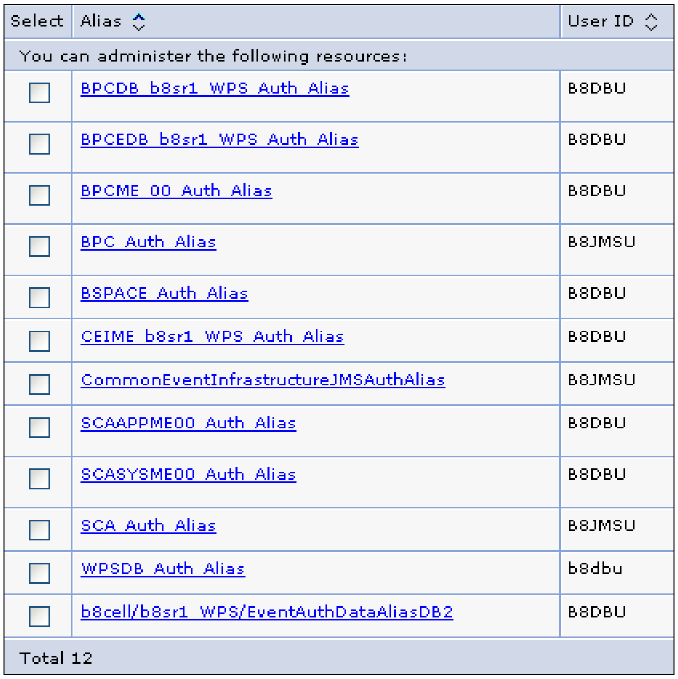
3. Repeat these steps for each bus.

Verifying authentication aliases

To verify the configured J2C authentication aliases:

1. Click **Security** → **Global security**.
2. In the Global security detail view, click **Java Authentication and Authorization Service** → **J2C Authentication data**.

   A list similar to Figure 6-7 is displayed. Make sure that you have the correct user IDs set for all authentication aliases.



Select	Alias	User ID
You can administer the following resources:		
□	BPCDB_b8sr1_WPS_Auth_Alias	B8DBU
□	BPCEDB_b8sr1_WPS_Auth_Alias	B8DBU
□	BPCME_00_Auth_Alias	B8DBU
□	BPC_Auth_Alias	B8JMSU
□	BSPACE_Auth_Alias	B8DBU
□	CEIME_b8sr1_WPS_Auth_Alias	B8DBU
□	CommonEventInfrastructureJMSAuthAlias	B8JMSU
□	SCAAPPME00_Auth_Alias	B8DBU
□	SCASYSME00_Auth_Alias	B8DBU
□	SCA_Auth_Alias	B8JMSU
□	WPSDB_Auth_Alias	b8dbu
□	b8cell/b8sr1_WPS/EventAuthDataAliasDB2	B8DBU
Total 12		

Figure 6-7 List of J2C authentication aliases

When you complete all of the verification steps successfully, you can continue with the sample applications.

Other errors

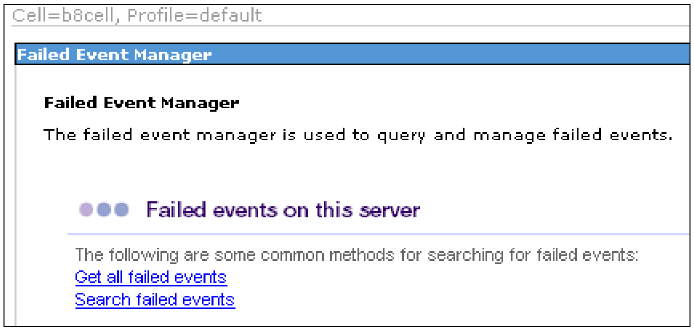
If you encounter any errors, you can probably fix them using the Integrated Solutions Console, depending on the particular error message. Here are helpful resources:

- WebSphere Process Server for z/OS Information Center
- Chapter 8, “Problem determination and prevention” on page 243 in this Redbooks publication

6.3 Verifying integrated applications

Now that you successfully finished the visual verification of various resource categories and the application runtime states, verify the integrated applications of WebSphere Process Server for z/OS. These applications are installed only on the deployment manager.

1. Click **Integrated Applications**. Here you will find three applications:
   – Failed Event Manager
   – Relationship Manager
   – Common Base Event Browser
2. Click **Failed Event Manager**, which opens the failed event manager application view, as shown in Figure 6-8.



Cell=b8cell, Profile=default

Failed Event Manager

Failed Event Manager

The failed event manager is used to query and manage failed events.

Failed events on this server

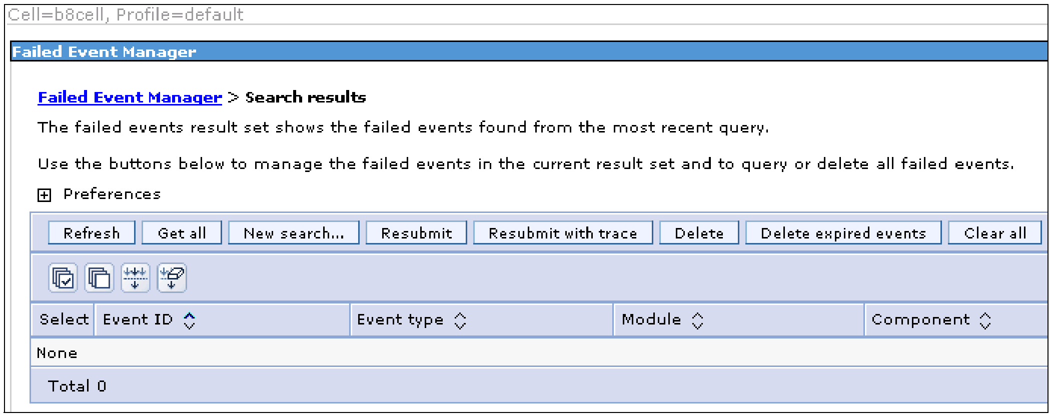
The following are some common methods for searching for failed events:

Get all failed events

Search failed events

Figure 6-8 Failed event manager application view

3. Click **Get all failed events**. A Search results window is displayed, as shown in Figure 6-9 on page 202.



Cell=b8cell, Profile=default

Failed Event Manager

Failed Event Manager > Search results

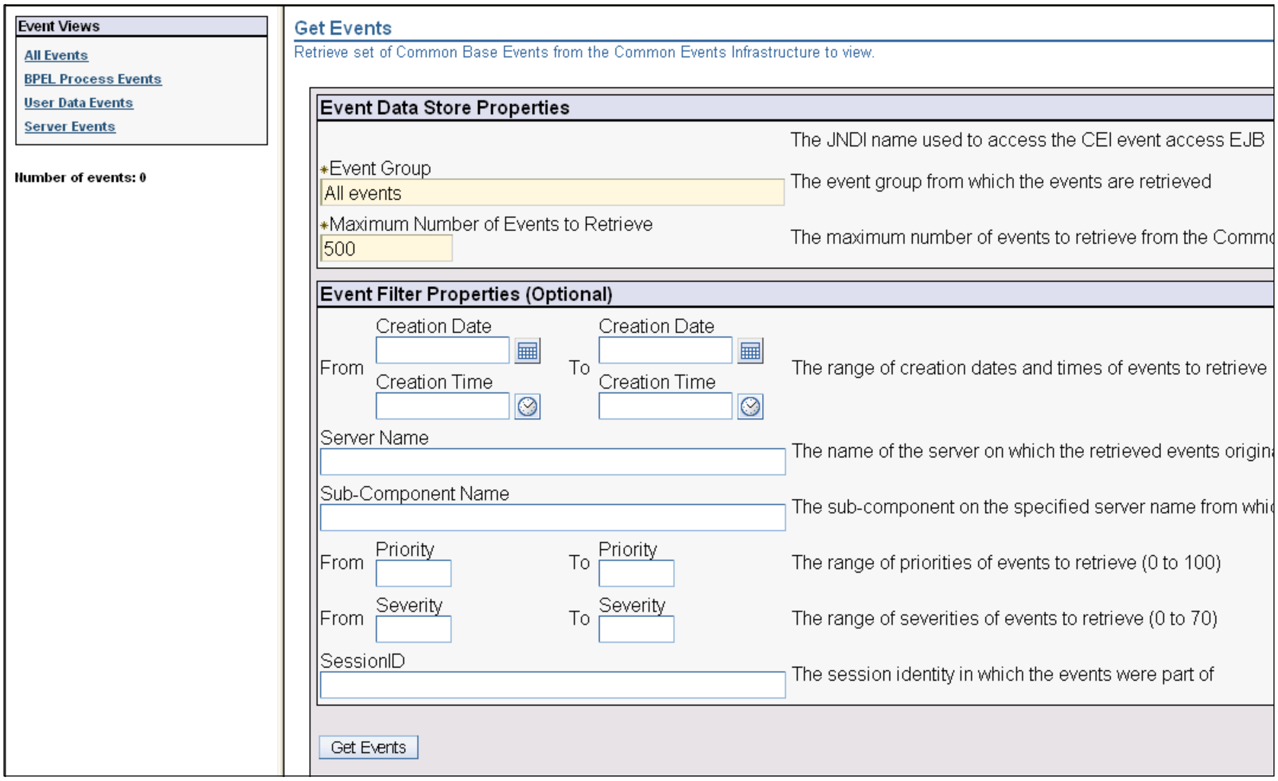
The failed events result set shows the failed events found from the most recent query.

Use the buttons below to manage the failed events in the current result set and to query or delete all failed events.

Select	Event ID	Event type	Module	Component
None				
Total 0				

Figure 6-9 No failed events in the failed event manager

4. Verify the Common Base Event browser (CBE browser). Click **Integrated applications** → **Common Base Event Browser**. After a short time, a page is displayed, as shown in Figure 6-10, where you must verify the text under the heading `Get Events`. If something is wrong with the CBE browser configuration, you will see an error message there.



Number of events: 0

Get Events

Retrieve set of Common Base Events from the Common Events Infrastructure to view.

Event Data Store Properties

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 Comm

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 origin

Sub-Component Name — The sub-component on the specified server name from whi

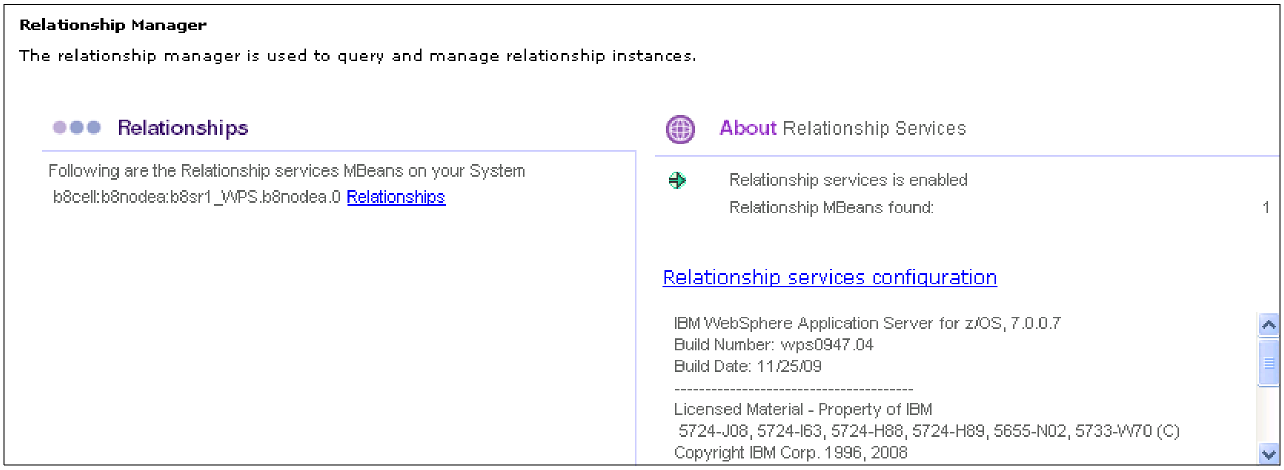
From Priority To Priority — The range of priorities of events to retrieve (0 to 100)

From Severity To Severity — The range of severities of events to retrieve (0 to 70)

SessionID — The session identity in which the events were part of

Figure 6-10 Common Base Event browser

5. In the Integration applications section, click **Relationship Manager**, and a view similar to Figure 6-11 on page 203 is displayed.



Relationship Manager

The relationship manager is used to query and manage relationship instances.

Relationships

Following are the Relationship services MBeans on your System

b8cell:b8nodea:b8sr1_WPS.b8nodea.0 Relationships

About Relationship Services

Relationship services is enabled

Relationship MBeans found: 1

Relationship services configuration

IBM WebSphere Application Server for z/OS, 7.0.0.7
Build Number: wps0947.04
Build Date: 11/25/09

Licensed Material - Property of IBM
5724-J08, 5724-I63, 5724-H88, 5724-H89, 5655-N02, 5733-W70 (C)
Copyright IBM Corp. 1996, 2008

Figure 6-11 Relationship Manager

Note the runtime state for the relationship service indicated by the green arrow.

6.4 Installing sample applications for installation verification

If everything looks good in the Integrated Solutions Console, try installing the sample applications. There are two listed here that are easy to test.

These applications do not have a security constraint and will not prompt for login. You can permit the B8GUEST user ID READ access to the required roles.

We assume that you extracted these applications to your working directory on the host.

6.4.1 Installing a simple BPC application: WPSEcho.ear

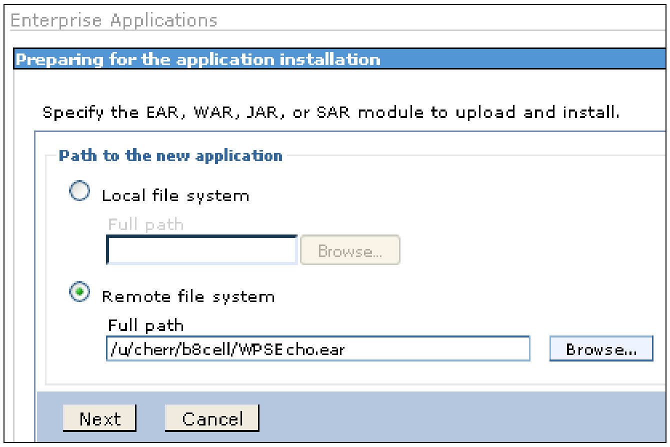
There are two sample BPEL applications that are available in the additional materials for this publication (and in Techdoc PRS2520).

The WPSEcho application has common event logging enabled, which was done in WebSphere Integration Developer:

- This application does not have a security constraint and therefore will not prompt for login. Permit the B8GUEST user ID READ access to the CEI eventCreator role (B8CELL.eventCreator).
- If you want to test a copy of this without CEI logging, select the wpsecho-nocei-20080909.ear file.

To install:

1. Using the Integrated Solutions Console, click **Applications** → **New Application**.
2. In the New Application view, click **New Enterprise Applications**.
3. Select **Remote file system**, click **Browse**, and navigate to the WPSEcho.EAR EAR file, as shown in Figure 6-12 on page 204.


Enterprise Applications
Preparing for the application installation
Path to the new application
Browse...
Next
Cancel

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

○ Local file system

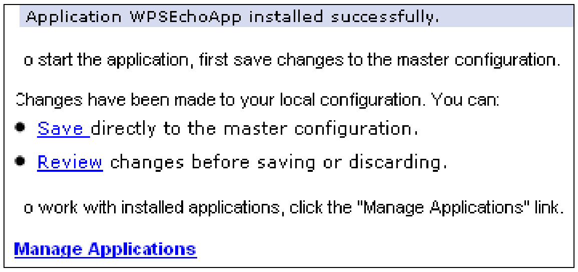
Full path

● Remote file system

Full path: /u/cherr/b8cell/WPSEcho.ear

Figure 6-12 Install application WPSEcho

4. Click **Next**. In the next panel, leave the Fast Path option selected, and click **Next**.
5. When you get to the window that shows steps 1 through 3:
   a. Click **STEP 2 – Map modules to servers**.
   b. Map the web module of the application to the cluster and the web server(s).
6. Click **Next**. Review the summary page, and click **Finish**.
7. When you see the message `Application WPSEchoApp installed successfully`, as shown in Figure 6-13, click the **Save directly to Master Configuration** link.



Application WPSEchoApp installed successfully.

o start the application, first save changes to the master configuration.

Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

o work with installed applications, click the "Manage Applications" link.

Manage Applications

Figure 6-13 Successful installation of WPSEcho application

8. Start the WPSEchoApp application in the Integrated Solutions Console.
9. Click **Applications** → **Enterprise Applications**.
10. Select **WPSEchoApp**.
11. Click **Start**.
12. Test it by entering the following URL, where we used the distributed VIPA and port of the web server:

    `http://<host>:<port>/WPSEchoWeb/WPSEchoInput.html`

    The reply from the Echo Application echoes back to your window.

6.4.2 Installing a long-running BPEL application

Perform the following steps to install a long-running BPEL application:

1. Install the WscHtTest.ear (Human Task Application), which is in the SampleWpsHtmApps directory. See Appendix A, “Additional material” on page 275. Accept all of the defaults.

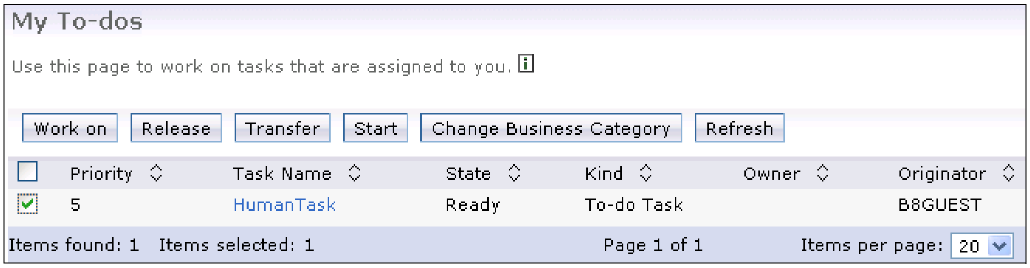
2. In the administrative console, start the WscHtTest application.
3. Test it with the following URL:

   `http://<host>:<port>/WscHtTestWeb/WscHtTestInput.html`

4. Enter a character string, and click **Submit**. The application will respond with a reply Request sent for processing.
5. Go to the BPC Explorer worklist at:

   `http://wpsplex.itso.ibm.com:28047/bpc`

6. Select **My Tasks**, select the work item, and click **Work on**, as shown in Figure 6-14.



My To-dos

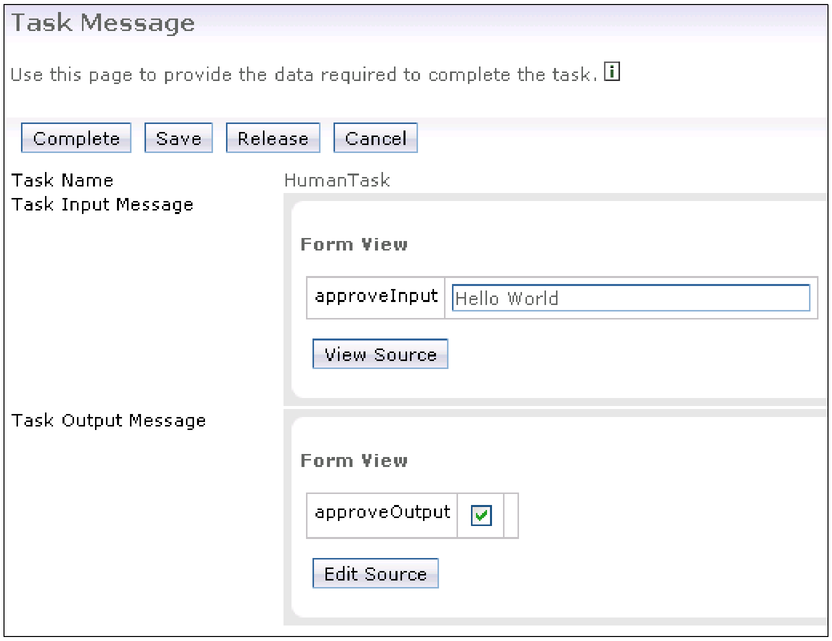
Use this page to work on tasks that are assigned to you.

□	Priority	Task Name	State	Kind	Owner	Originator
☑	5	HumanTask	Ready	To-do Task		B8GUEST

Items found: 1 Items selected: 1 Page 1 of 1 Items per page: 20

Figure 6-14 Human task to work on

7. Click **Complete**, as shown in Figure 6-15.



Task Message

Use this page to provide the data required to complete the task.

Task Name: HumanTask

Task Input Message: Form View — approveInput: Hello World

Task Output Message: Form View — approveOutput: ☑

Figure 6-15 Human task details

8. Look in the SYSPRINT output of the servant region and see messages starting with `WscHtTest`. These messages show that the Human Task was invoked, a message was received, and the process completed, as shown in Example 6-1.

Example 6-1 SYSPRINT output

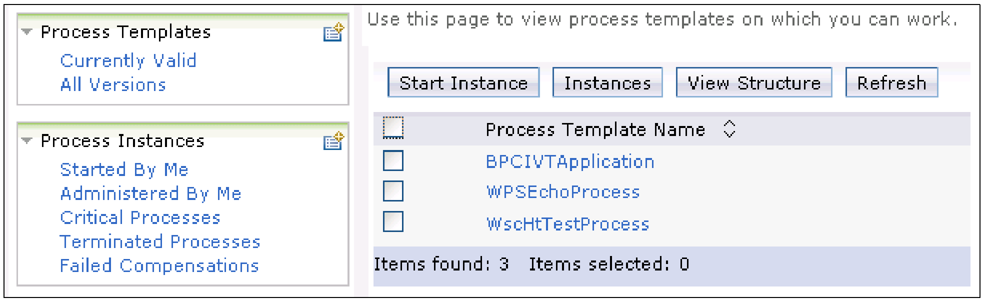
```
WscHtTest: Request received.
WscHtTest: Input data = Input to HumTask
WscHtTest: Invoking Human Task.
```

```
WscHtTest: Reply recieved from Human Task.
WscHtTest: Human Task reply = true
WscHtTest: Process complete.
```

6.4.3 Installing the bpcivt sample application

WebSphere Process Server for z/OS also provides sample applications for verifying the installation of WebSphere Process Server for z/OS. You will install BPCIVTApp, for the verification of the installation:

1. Install BPCIVTApp.ear (BPC installation verification test application), which is in the installableApps directory in the deployment manager profiles home directory. In our installation, this is /wasv7config/b8cell/b8dmnode/DeploymentManager/installableApps.
2. In the administrative console, start the BPCIVTApp application.
3. Login to BPC Explorer, and in the Process Template section, click **Currently Valid**, as shown in Figure 6-16.



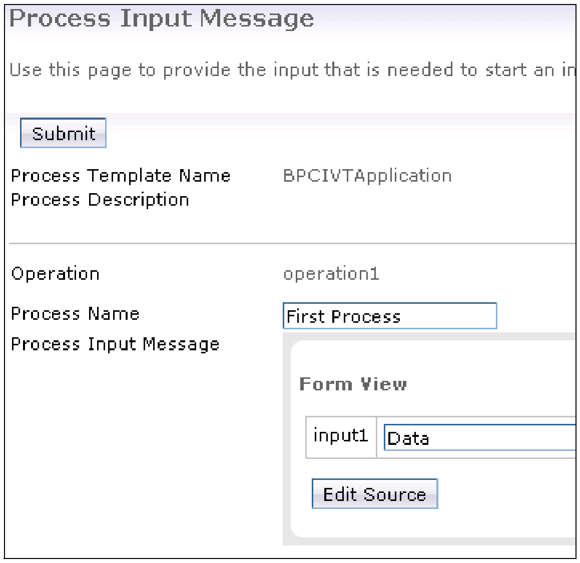
Use this page to view process templates on which you can work.

□	Process Template Name
□	BPCIVTApplication
□	WPSEchoProcess
□	WscHtTestProcess

Items found: 3 Items selected: 0

Figure 6-16 Process templates

4. In the process templates list, select **BPCIVTApplication**, and click **Start Instance**.
5. Enter values of your choice for process name and input value, as shown in Figure 6-17, and click **Submit**. The new instance starts, which creates a new human task.



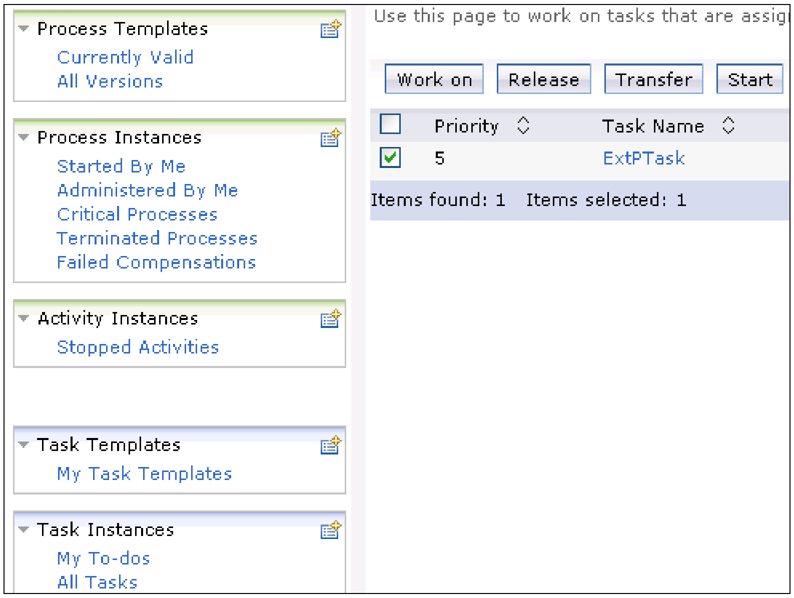
Process Input Message

Use this page to provide the input that is needed to start an in

Process Template Name	BPCIVTApplication
Process Description	
Operation	operation1
Process Name	First Process
Process Input Message	
input1	Data

Figure 6-17 Start a new process instance

6. Click **My To-dos**. In the list of tasks, select **ExtPTask**, and click **Work on**.



Use this page to work on tasks that are assig

☐	Priority	Task Name
☑	5	ExtPTask

Items found: 1   Items selected: 1

Figure 6-18 Human task of BPCIVT application

7. Enter a value of your choice for the output value, and click Complete. The process instance finishes successfully and is automatically deleted afterwards.

6.4.4 Installing the vehicle loan process application

In this publication, we provide a sample vehicle loan process that was created for the fictitious company ITSOBank. You can use this vehicle loan process to test the topology you built in this chapter. The sample application described in this section is supplied in Appendix A, “Additional material” on page 275.

For more information about the vehicle loan process, refer to *WebSphere Business Process Management V6.2 Production Topologies*, SG24-7732.

6.4.5 Installing the sample application

After you obtain the additional materials, copy the ITSOApp.ear and ITSO_implApp.ear files to a working directory on your workstation. Installation follows the normal process.

To install the sample application:

1. Using the Integrated Solutions Console, click **Applications** → **New Application**.
2. In the New Application view, click **New Enterprise Applications**.
3. Select **Remote file system**, and click **Browse**. Navigate to the ITSO_implApp.ear EAR file.
4. Click Next. In the next panel, leave the **Fast Path** option selected, and click **Next**.
5. When you get to the window that shows steps 1 through 3:
   a. Click **STEP 2 – Map modules to servers**.
   b. Map the web module of the application to the cluster and the web server(s).
   c. Click **Next**.

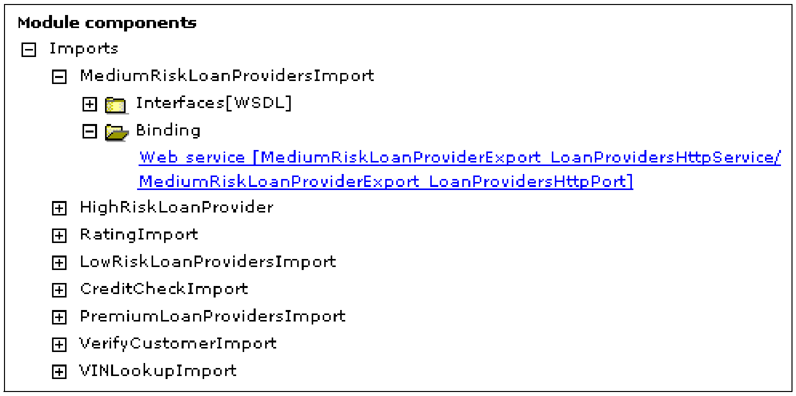
6. The summary page is displayed. After reviewing it, click **Finish**. When you see the message `Application ITSO_implApp installed successfully`, click **Save directly to Master Configuration**.
7. Repeat this process for the ITSOApp.ear file.

6.4.6 Updating the SCA import URLs

In the sample application, all of the Web service SCA imports are bound to URLs that point to itsodmgr. There is no port explicitly defined in the URLs, so the default HTTP port of 9080 is used. Change these to match your host name:port.

> **Note:** Ensure that at least one WebSphere Process Server for z/OS cluster member is started and that the three ITSO sample applications are started.

1. In the Integrated Solutions Console, click **Applications** → **SCA Modules**. In the SCA modules list, click **ITSO**.
2. In the SCA modules details view, locate the Modules components section, and click **Imports**. A list of imports is displayed.
3. Click **Import** → **Bindings**, as shown in Figure 6-19.



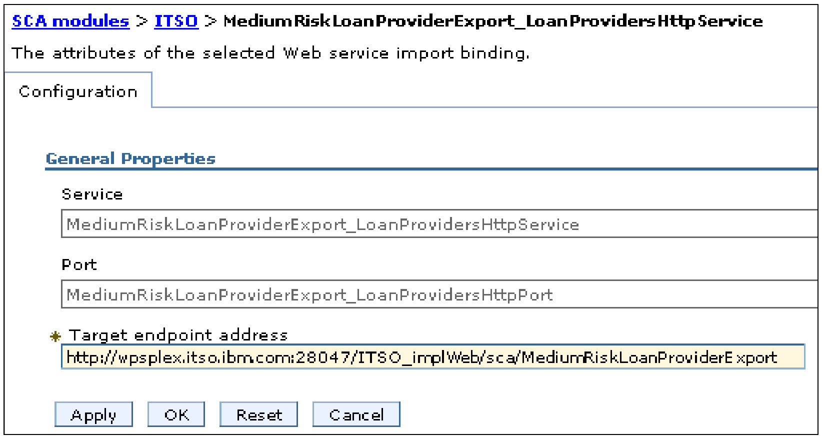
Module components

⊟ Imports
- ⊟ MediumRiskLoanProvidersImport
  - ⊞ Interfaces[WSDL]
  - ⊟ Binding
    - Web service [MediumRiskLoanProviderExport_LoanProvidersHttpService/MediumRiskLoanProviderExport_LoanProvidersHttpPort]
- ⊞ HighRiskLoanProvider
- ⊞ RatingImport
- ⊞ LowRiskLoanProvidersImport
- ⊞ CreditCheckImport
- ⊞ PremiumLoanProvidersImport
- ⊞ VerifyCustomerImport
- ⊞ VINLookupImport

Figure 6-19 Expand the MediumRiskLoanProvidersImport bindings

4. Click **Web Service [MediumRiskLoanProviderExport...]**.
5. In the Endpoint field, Figure 6-20 on page 209, change the host name and port number. Click **OK**.

   When you have a cluster with Web servers that are running the WebSphere plug-in, you normally specify the Distributed VIPA and port of the web server. We specified the distributed VIPA of the web server.



SCA modules > ITSO > MediumRiskLoanProviderExport_LoanProvidersHttpService

The attributes of the selected Web service import binding.

Configuration

General Properties

Service

MediumRiskLoanProviderExport_LoanProvidersHttpService

Port

MediumRiskLoanProviderExport_LoanProvidersHttpPort

* Target endpoint address

http://wpsplex.itso.ibm.com:28047/ITSO_implWeb/sca/MediumRiskLoanProviderExport

Figure 6-20 Adjust import URL

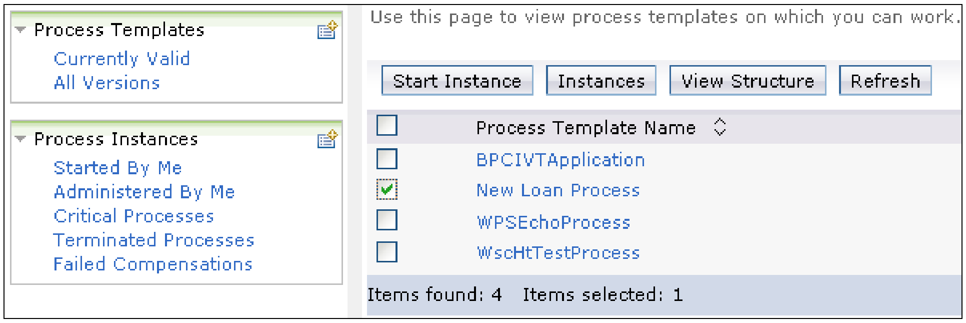
6. Repeat these steps for the remaining import bindings in the SCA module, such as HighRiskLoadProvider through VerifyCustomerImport.
7. Save the configuration changes to the master repository.

6.4.7 Testing the sample application

To test the sample application:

1. Navigate to **Applications** → **Application Types** → **WebSphere enterprise applications**. Select the **ITSOApp** and **ITSO_implApp** options, and click **Start**.
2. Login to the Business Process Choreographer explorer. In our environment, we used the following URL:

   `http://wpsplex.itso.ibm.com:28047/bpc`
3. In the Process Templates, section click **Currently Valid**. Figure 6-21 is displayed.



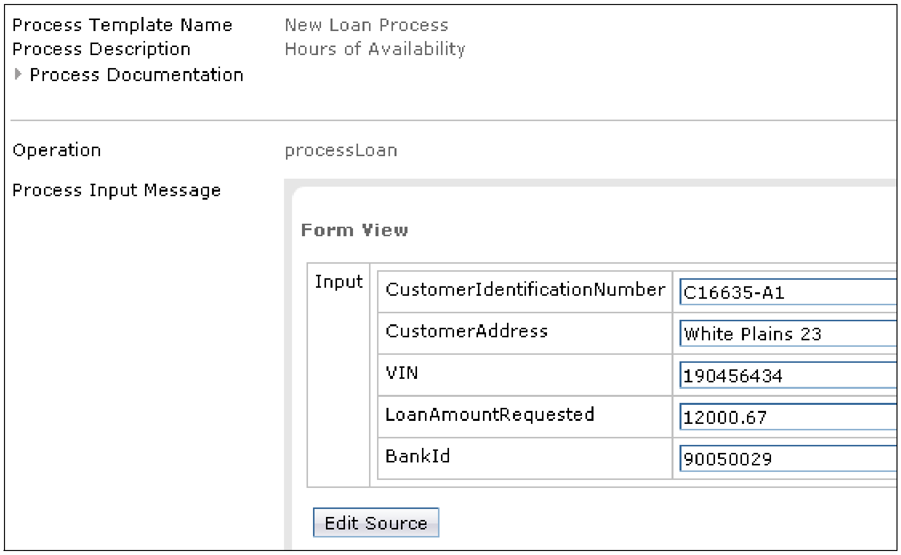
Use this page to view process templates on which you can work.

□	Process Template Name
□	BPCIVTApplication
☑	New Loan Process
□	WPSEchoProcess
□	WscHtTestProcess

Items found: 4 Items selected: 1

Figure 6-21 Process Templates, including New Loan Process

4. Select **New Loan Process** → **Start Instance**. Provide test input data, and click **Submit**, as shown in Figure 6-22 on page 210.



Process Template Name New Loan Process
Process Description Hours of Availability
Process Documentation

Operation processLoan

Process Input Message

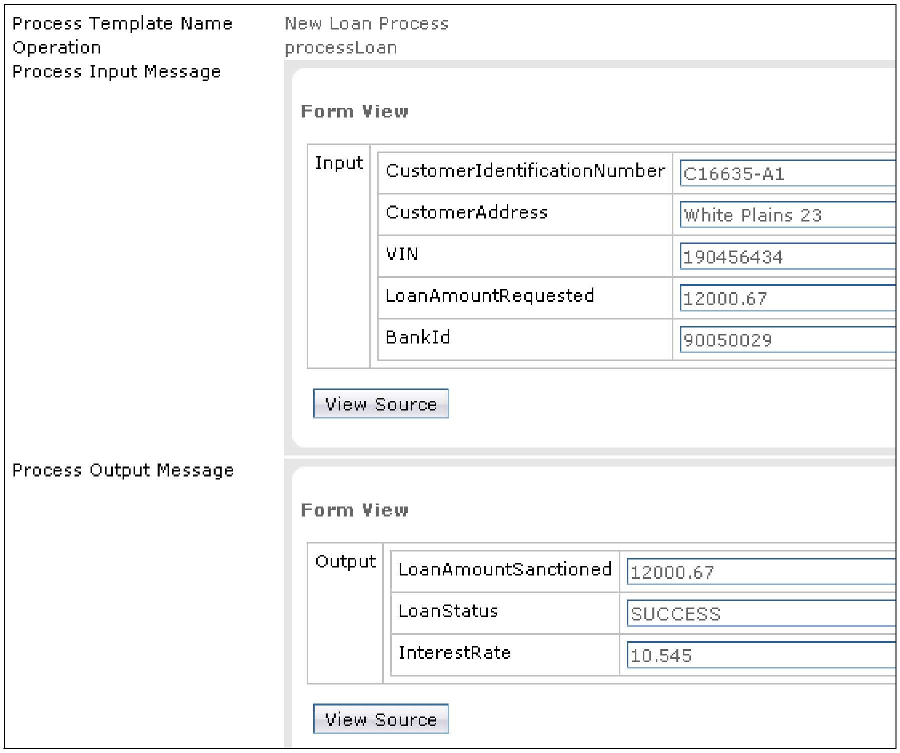
Form View

Input	CustomerIdentificationNumber	C16635-A1
	CustomerAddress	White Plains 23
	VIN	190456434
	LoanAmountRequested	12000.67
	BankId	90050029

Figure 6-22 Process input message

The business process is launched.

If the environment is working correctly, it returns a response in the Process Output Message window shown in Figure 6-23.



Process Template Name New Loan Process
Operation processLoan
Process Input Message

Form View

Input	CustomerIdentificationNumber	C16635-A1
	CustomerAddress	White Plains 23
	VIN	190456434
	LoanAmountRequested	12000.67
	BankId	90050029

Process Output Message

Form View

Output	LoanAmountSanctioned	12000.67
	LoanStatus	SUCCESS
	InterestRate	10.545

Figure 6-23 Process output message

Note: If you get the following error message in your browser, instead of the process output message, you might not have reset all of the SCA service Endpoints:

```
WSWS3713E: Connection to the remote host localhost failed.Received the
following error: EDC8128I Connection refused.
```

6.5 Verifying Business Space powered by WebSphere

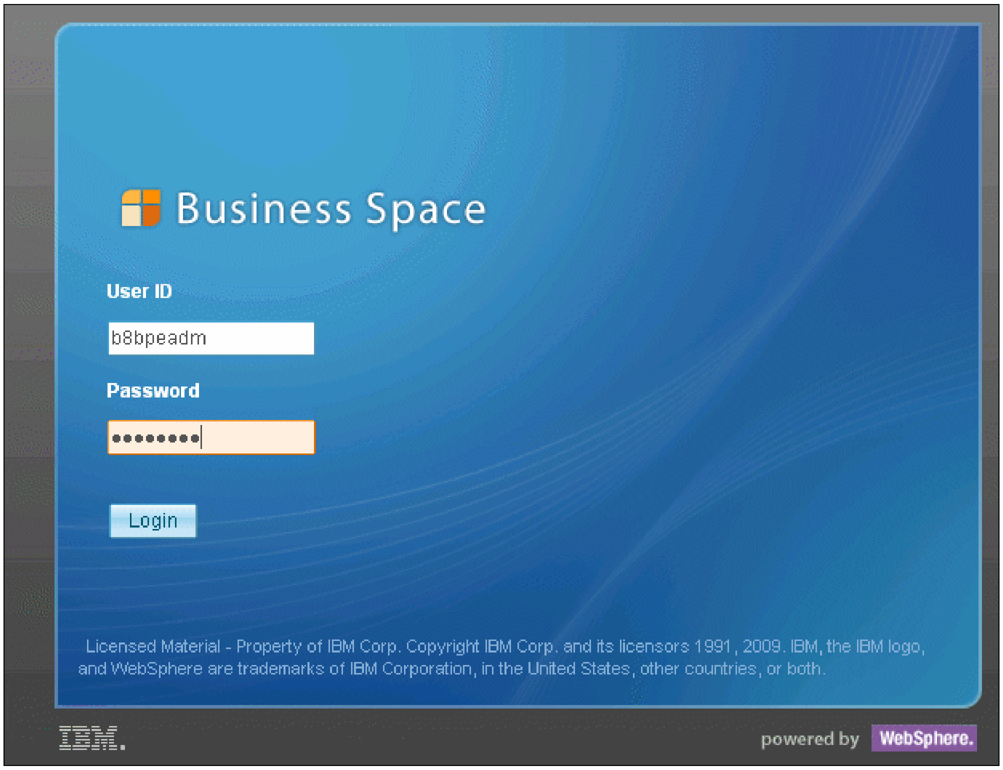
To verify Business Space powered by WebSphere:

1. Open the login panel. In your browser, enter your web servers' HTTP address followed by context URL /BusinessSpace. In our cell this is:

   ```
   http://wpsplex.itso.ibm.com:28047/BusinessSpace
   ```

   The login panel of Business Space powered by WebSphere is displayed.

2. Enter the user ID and password. In our sample, our user ID and password are b8bpeadm and b8bpeadm, as shown in Figure 6-24.



Business Space

User ID

b8bpeadm

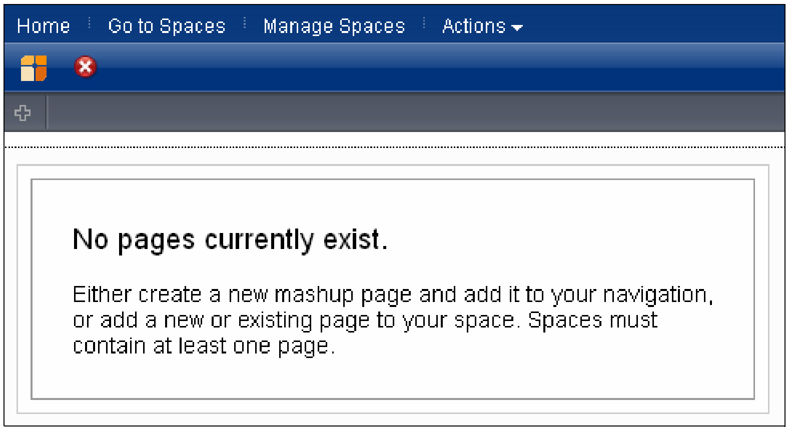
Password

••••••••

Licensed Material - Property of IBM Corp. Copyright IBM Corp. and its licensors 1991, 2009. IBM, the IBM logo, and WebSphere are trademarks of IBM Corporation, in the United States, other countries, or both.

Figure 6-24 Business Space powered by WebSphere login panel

After the successful login, you get a window similar to Figure 6-25 on page 212, if no spaces are installed.

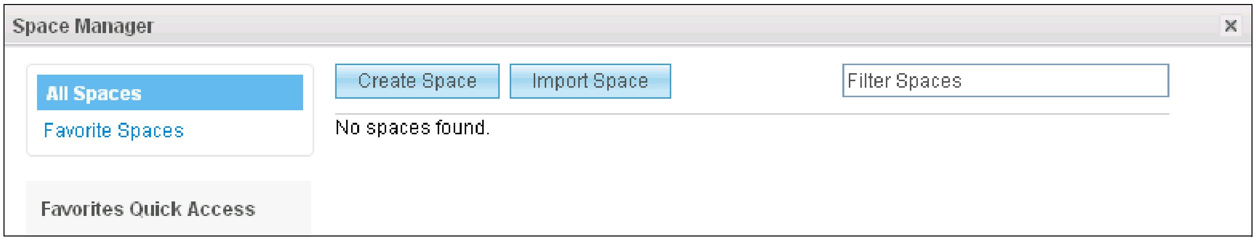


No pages currently exist.

Either create a new mashup page and add it to your navigation, or add a new or existing page to your space. Spaces must contain at least one page.

Figure 6-25 Business Space after login

3. Click **Manage Spaces** to get the Space Manager window, as shown in Figure 6-26.



No spaces found.

Figure 6-26 Space Manager panel

6.6 Uninstalling the BPEL applications

To uninstall an enterprise application from the server, issue the commands shown in Example 6-2.

Make sure that at least one cluster member is running when you execute the command. The -force option stops and deletes all running process instances of the application before uninstalling it. You can omit the -force option if you already stopped and deleted all process instances and stopped all process templates related to that application.

Example 6-2 Uninstalling bpel applications

```
cd <DMGR_root>/ProcessChoreographer/admin

../../bin/wsadmin.sh -lang jacl -f bpcTemplates.jacl -uninstall
"<AppName>" -force -user b8admin -password b8admin -port 28002
```

7

Adding WebSphere Business Services Fabric for z/OS to an existing z/OS cell

In this chapter, we provide detailed instructions about how to incorporate WebSphere Business Services Fabric into a production WebSphere Process Server for z/OS topology. The WebSphere Business Services Fabric for z/OS product consists of two distinct sets of functions:

- IBM Business Services Foundation Pack

  IBM Business Services Foundation Pack provides the runtime components for installing and managing WebSphere Business Services Fabric for z/OS components and is the primary subject of this chapter.

- IBM Business Services Tool Pack

  IBM Business Services Tool Pack provides tooling extensions for creating business services. These tooling extensions run only on workstation platforms.

In this chapter, we discuss:

7.1 Planning and preparing WebSphere Business Services Fabric for z/OS installation

WebSphere Business Services Fabric for z/OS is a comprehensive service-oriented architecture (SOA)-based offering to deliver dynamic SOA that uses existing IT assets and delivers business value incrementally.

In this chapter, we focus on incorporating WebSphere Business Services Fabric for z/OS into an existing WebSphere Process Server for z/OS production topology, which was created by using the pattern-based deployment environment wizard.

We tested in a single-topology with WebSphere Process Server for z/OS and WebSphere Business Services Fabric for z/OS in the same cluster. This is the BPM reference topology for z/OS, as explained in 1.5, “BPM reference topology for z/OS and how we use it here” on page 21.

In this topology, a single cluster runs the WebSphere Process Server for z/OS support and messaging infrastructure along with components supporting Business Process Choreographer (BPC), WebSphere Business Services Fabric for z/OS, and the applications.

Figure 7-1 shows the BPM reference topology for z/OS. It also shows where WebSphere Business Services Fabric for z/OS components are added to it.

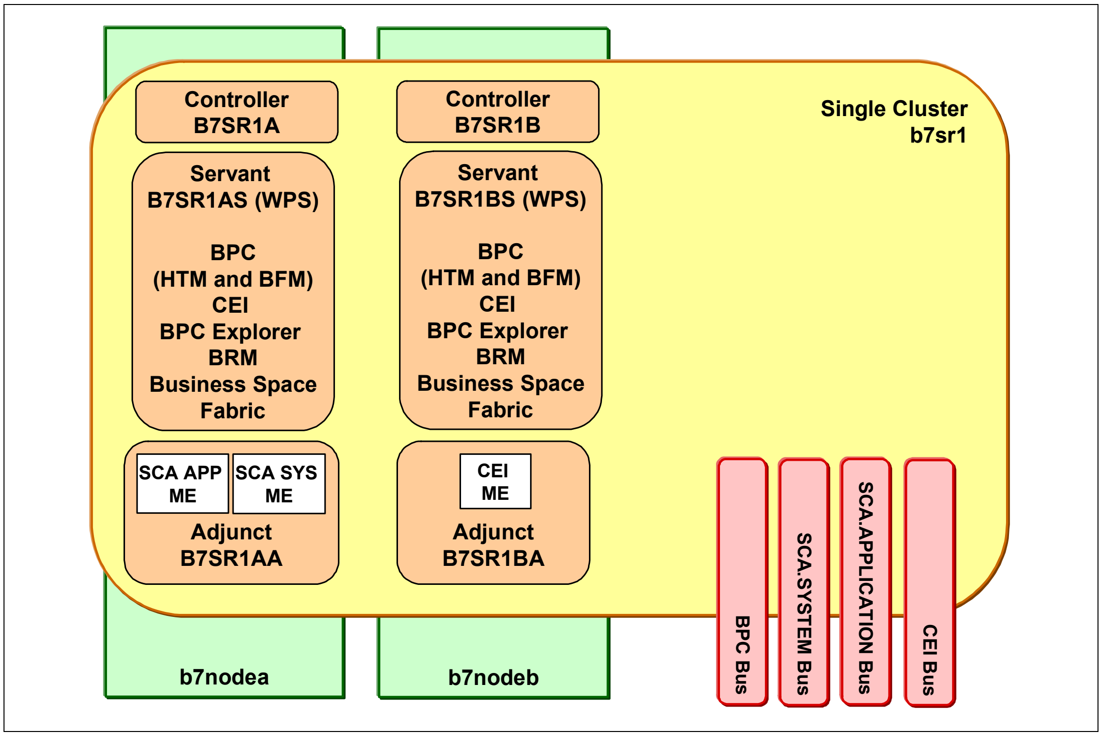


Figure 7-1 WebSphere Business Services Fabric for z/OS in a WebSphere network deployment cell

When you add WebSphere Business Services Fabric for z/OS to the topology, the following components are added:

- The WebSphere Business Services Fabric for z/OS core application EAR files are added to the cluster.

  In the BPM reference topology for z/OS the Fabric EAR files Fabric tools, Fabric trace, and Fabric catalog are deployed to the cluster running the other BPM components (b7sr1 in Figure 7-1 on page 214).
- An object cache instance Fabric Context Cache is created.
- Two Name Space Bindings are created on cluster scope.
- WebSphere Business Services Fabric for z/OS events are emitted to the JMS destinations that are present in the cluster.

7.1.1 Planning and preparing the Fabric installation

In this section, we describe the software pre-requisites and the installation of the product using SMP/E. We introduce the additional material that we provide with this publication. We also provide comments about creating the deployment directory.

The WebSphere Business Services Fabric for z/OS installation and configuration is tightly integrated with, and dependent on, the installation and configuration of WebSphere Application Server for z/OS. The task is, therefore, a multi-phase process that can span multiple roles.

7.1.2 Software pre-requisites

The pre-requisites for WebSphere Business Services Fabric for z/OS are in the IBM support article, *System requirements for WebSphere Business Services Fabric V7.0 on z/OS*, which is available at the following web page:

http://www-01.ibm.com/support/docview.wss?rs=36&uid=swg27016904

The key requirement is WebSphere Process Server for z/OS V7R0M0, which includes Business Space powered by WebSphere. The requirement for LDAP depends on whether you decide to use LDAP or a local SAF user registry.

7.1.3 Installing WebSphere Business Services Fabric for z/OS Foundation Pack into SMP/E

The instructions for installing WebSphere Business Services Fabric for z/OS into SMP/E are in *WebSphere Business Service Fabric 7.0 Program Directory*, GI11-4301-00, which is shipped with the product. The program directory can also be downloaded from IBM Publications at the following web page:

http://www.ibm.com/shop/publications/order

The product ZFS occupies around 2800 TRACKS of disk space.

When SMP/E work is finished, the WebSphere Business Services Fabric for z/OS product ZFS is typically mounted read-only at /usr/lpp/zWBSF/V7R0. That copy is shared across the sysplex. Configuration of WebSphere Business Services Fabric for z/OS starts after the product is made available in the UNIX System Services file system.

7.1.4 Sample JCL and shell scripts used in this chapter

The sample JCL and shell scripts that we describe in this chapter are on the /zos_wbsf directory in the additional materials disk.

The JCL is in the WBSF.CNTL library and the shell scripts in the /wpswork directory.

Creating an intermediate symbolic link

If you followed our recommendation to configure all nodes so that they refer to the product mount points through an intermediate symbolic link, create such a link when you install WebSphere Business Services Fabric into the nodes in the cell. Intermediate symbolic links were created for WebSphere Application Server, the Feature Packs, and for WebSphere Process Server for z/OS by the jobs generated by the zPMT tool that is running in WCT; however, you must manually create an intermediate symbolic link for WebSphere Business Services Fabric for z/OS.

The planning spreadsheet generates sample job xxSYMLDM on the JCL_P wprksheet and sample job xxSYMLNB on the JCL_S worksheet, where xx is the cell prefix that you choose. The xxSYMDM job generates an intermediate symbolic link from the deployment manager's node and the primary managed node to the mount point of the WebSphere Business Services Fabric product. The job xxSYMLNB generates the symbolic link from the secondary node. These jobs are also in the WBSF.CNTL directory within the Additional Material that we provided in Appendix A, “Additional material” on page 275. Example 7-1 shows the sample job B7SYMLDM.

Example 7-1 Sample job B7SYMLDM to define intermediate sumbolic link sfor WBSF

```
//B7SYMLDM  JOB (0),'WPS SETUP',CLASS=A,REGION=0M,
// MSGCLASS=H,NOTIFY=&SYSUID,
// USER=B7ADMIN
//*
/*JOBPARM SYSAFF=SC42
//*---------------------------------------------------*/
//* Create intermediate symbolic links for WBSF        */
//* Set correct ownership:group                        */
//*---------------------------------------------------*/
//DEFSYML  EXEC  PGM=IKJEFT01
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *
 BPXBATCH  SH +
 ln -s /usr/lpp/zWBSF/V7R0 +
      /wasv7config/b7cell/b7dmnode/wbsfsmpe ; +
 ln -s /usr/lpp/zWBSF/V7R0 +
      /wasv7config/b7cell/b7nodea/wbsfsmpe ;
/*
//*---------------------------------------------------*/
//* Set the correct owner:group                        */
//*---------------------------------------------------*/
//CHMOD     EXEC  PGM=IKJEFT01
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
 BPXBATCH   SH  +
  chown -Rh B7ADMIN:B7CFG  /wasv7config/b7cell/b7dmnode/wbsfsmpe ; +
  chown -Rh B7ADMIN:B7CFG  /wasv7config/b7cell/b7nodea/wbsfsmpe ;
/*
```

Configuring WebSphere Business Services Fabric for z/OS security

The WebSphere Business Services Fabric for z/OS Information Center describes how to use four types of user registries:

- Standalone LDAP
- Local operating system
- Federated repositories
- Federated repositories with LDAP

For our Fabric installation, we focus on using the local operating system option. On z/OS, this option means RACF. When using RACF as the user registry for the cell, consult the Information Center article *Configuring WebSphere Process Server security to use local operating system,* which is available at the following web page:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/topic/com.ibm.ws.fabric.admenv.doc/task/t_confwpssecuwinlocalosrepository_zos.html

Some RACF groups and user IDs are required for J2C authentication aliases. Plan groups and user IDs to permit to various roles. We planned the groups and user IDs that we wanted to use with a spreadsheet. The spreadsheet is included in Appendix A, “Additional material” on page 275. If you decide to use RACF GEJBROLE profiles, run jobs B7RACFG, B7HOMEG, and B7GEJB. If you decide to use RACF EJBROLE profiles, run jobs B7RACFE, B7HOMEE, and B7EJB.

Adding RACF users and groups

We created sample job B7RACF, shown in Example 7-2 to define all of the groups and user IDs that we required for use with Fabric and B7HOME to define their home directories.

Example 7-2 Sample job B7RACF to define user IDs and groups for Fabric

```
//B7RACFE JOB (0),'B7ADMIN',CLASS=A,MSGCLASS=H,REGION=0M,
//    NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC42
//***********************************************************/
//*  Make sure you edit this file with CAPS OFF.            */
//*  Run this job using a userid with RACF SPECIAL.         */
//*  Change B7 to your two character cell prefix.           */
//*         For example, B7. (Use upper case.)              */
//*  Set the OWNER as you want.                             */
//*  Set the UIDs and GIDs so they are unique for this cell.*/
//*  Run job B7HOME after this to create home directories.  */
//***********************************************************/
//*  Add groups and userids for roles related to Fabric
//RACF     EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN  DD *
  /*   Group for FAB administrators        */
  AG  B7FADMG        SUPGROUP(B7CFG) OMVS(AUTOGID) +
                     DATA('B7 Fabric Admins')
  /*   Fabric admin userid                     */
  AU  B7FABADM       DFLTGRP(B7FADMG) OMVS(AUTOUID  +
                     HOME(/var/WebSphere/home/B7FADMG) +
                     PROGRAM(/bin/sh))                  +
                     NAME('B7 Fabric Admin')            +
                     OWNER(B7CFG)
  PW  USER(B7FABADM) NOINTERVAL
  ALU B7FABADM       PASSWORD(B7FABADM) NOEXPIRED
  /*   Group for Fabric Governance admininstrators        */
  AG  B7FGADMG       SUPGROUP(B7CFG) OMVS(AUTOGID) +
                     DATA('B7 Fabric Governance Admins')
  /*   Fabric governance admin userid          */
  AU  B7FGADM        DFLTGRP(B7FGADMG)  OMVS(AUTOUID  +
                     HOME(/var/WebSphere/home/B7FGADMG) +
                     PROGRAM(/bin/sh))                  +
                     NAME('B7 Fab Gov Adm')             +
                     OWNER(B7CFG)
  PW  USER(B7FGADM)  NOINTERVAL
  ALU B7FGADM        PASSWORD(B7FGADM)  NOEXPIRED
  /*   Group for Fabric Performance Users   */
  AG  B7FPUG         SUPGROUP(B7CFG) OMVS(AUTOGID) +
                     DATA('B7 FAB PERF USER')
  /*   Fabric Performance User userid       */
  AU  B7FPU          DFLTGRP(B7FPUG) OMVS(AUTOUID  +
                     HOME(/var/WebSphere/home/B7FPUG) +
                     PROGRAM(/bin/sh))                  +
                     NAME('B7 Fabric Performance Users') +
                     OWNER(B7CFG)
  PW  USER(B7FPU)    NOINTERVAL
  ALU B7FPU          PASSWORD(B7FPU) NOEXPIRED
  /*   Group for Fabric Subscriber Managers */
  AG  B7FSMG         SUPGROUP(B7CFG) OMVS(AUTOGID) +
                     DATA('B7 Fabric Subscriber Managers')
  /*   Fabric Subscriber Manager userid     */
  AU  B7FSMU         DFLTGRP(B7FSMG) OMVS(AUTOUID  +
                     HOME(/var/WebSphere/home/B7FSMG) +
                     PROGRAM(/bin/sh))                  +
                     NAME('B7 FAB SUB MGR') +
                     OWNER(B7CFG)
  PW  USER(B7FSMU)   NOINTERVAL
  ALU B7FSMU PASSWORD(B7FSMU) NOEXPIRED
  /*   Group for Fabric Basic Users         */
  AG  B7FBUG         SUPGROUP(B7CFG) OMVS(AUTOGID) +
                     DATA('B7 Fabic Basic Users')
  /*   Fabric Basic User userid             */
  AU  B7FBU          DFLTGRP(B7FBUG)   OMVS(AUTOUID  +
                     HOME(/var/WebSphere/home/B7FBUG) +
                     PROGRAM(/bin/sh))                  +
                     NAME('B7 FAB Base User') +
                     OWNER(B7CFG)
  PW  USER(B7FBU)    NOINTERVAL
  ALU B7FBU PASSWORD(B7FBU) NOEXPIRED
  /*   Group for Fabric Studio Users        */
  AG  B7FSUG         SUPGROUP(B7CFG) OMVS(AUTOGID) +
                     DATA('B7 Fabric Studio Users')
  /*   Fabric Studio User userid            */
  AU  B7FSU          DFLTGRP(B7FSUG)   OMVS(AUTOUID  +
                     HOME(/var/WebSphere/home/B7FSUG) +
                     PROGRAM(/bin/sh))                  +
                     NAME('B7 FAB Studio User') +
                     OWNER(B7CFG)
  PW  USER(B7FSU)    NOINTERVAL
  ALU B7FSU          PASSWORD(B7FSU) NOEXPIRED
  /* Permit the role groups to the  APPL profile */
  PE  B7 CLASS(APPL) ID(B7FADMG, B7FGADM, B7FPUG,  B7FSMG)
  PE  B7 CLASS(APPL) ID(B7FBUG,  B7FSUG)
  SETR RACLIST(APPL) REFRESH
/*
```

Creating home directories for the users

When using EJBROLE profiles, sample job B7HOMEE makes home directories for the users that job B7RACFE created. When using GEJBROLE profiles, sample job B7HOMEG makes home directories for the users that job B7RACFG created.

Adding RACF EJBROLEs

We created sample job B7EJB, shown in Example 7-3, to define RACF EJBROLE profiles for the roles that are related to WebSphere Business Services Fabric for z/OS. This job also permitted the necessary groups to use the roles. We used EJBROLE instead of GEJBROLE because in contrast to WebSphere Process Server for z/OS security configuration there is no advantage using GEJBROLE for WebSphere Business Services Fabric for z/OS, and there are no roles that can be combined in an overall role such as in WebSphere Process Server for z/OS.

Example 7-3 Sample job B7EJB to define EJBROLE profiles for Fabric

```
//B7EJB JOB (0),'WBSF SEC',CLASS=A,MSGCLASS=A,REGION=0M,
//   NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC42
//***********************************************************/
//*  Make sure you edit this file with CAPS OFF.            */
//*  Run this job using a userid with RACF SPECIAL.         */
//*  The RACF userids and groups used here must exist.      */
//*          Run job XXRACF first.                          */
//*  Change B7 to your two character cell prefix.           */
//*        For example, B7. (Use upper case.)               */
//*  Set the OWNER as you want.                             */
//***********************************************************/
//*
//*  Create EJB role profiles for Fabric
//*
//RACFDBA  EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN  DD *
/*   Define FabricAdministrator role           */
  RDEFINE EJBROLE  B7CELL.FabricAdministrator                  +
                   UACC(NONE)                                  +
                   OWNER(B7CFG)                                +
                   DATA('B7CELL FabricAdministrator role')
  PERMIT           B7CELL.FabricAdministrator                  +
                   CLASS(EJBROLE)                              +
                   RESET
  PERMIT           B7CELL.FabricAdministrator                  +
                   CLASS(EJBROLE)                              +
                   ID(B7FADMG,B7FABADM) ACCESS(READ)
 /*   Define FabricGovernanceAdministrator role */
  RDEFINE EJBROLE  B7CELL.FabricGovernanceAdministrator        +
                   UACC(NONE)                                  +
                   OWNER(B7CFG)                                +
                   DATA('B7CELL FabricGovernanceAdministrator role')
  PERMIT           B7CELL.FabricGovernanceAdministrator        +
                   CLASS(EJBROLE)                              +
                   RESET
  PERMIT           B7CELL.FabricGovernanceAdministrator        +
                   CLASS(EJBROLE)                              +
                   ID(B7FADMG,B7FGADMG)                        +
                   ACCESS(READ)
 /*   Define FabricPerformanceUser role         */
  RDEFINE EJBROLE  B7CELL.FabricPerformanceUser                +
                   UACC(NONE)                                   +
                   OWNER(B7CFG)                                 +
                   DATA('B7CELL FabricPerformanceUser role')
  PERMIT           B7CELL.FabricPerformanceUser                 +
                   CLASS(EJBROLE)                               +
                   RESET
  PERMIT           B7CELL.FabricPerformanceUser                 +
                   CLASS(EJBROLE)                               +
                   ID(B7FPUG,B7FADMG)                           +
                   ACCESS(READ)
   /*   Define FabricSubscriberManager role       */
  RDEFINE EJBROLE  B7CELL.FabricSubscriberManager               +
                   UACC(NONE)                                   +
                   OWNER(B7CFG)                                 +
                   DATA('B7CELL FabricSubscriberManager role')
  PERMIT           B7CELL.FabricSubscriberManager               +
                   CLASS(EJBROLE)                               +
                   RESET
  PERMIT           B7CELL.FabricSubscriberManager               +
                   CLASS(EJBROLE)                               +
                   ID(B7FSMG,B7FADMG)                           +
                   ACCESS(READ)
 /*   Define FabricBasicUser role               */
  RDEFINE EJBROLE  B7CELL.FabricBasicUser                       +
                   UACC(NONE)                                   +
                   OWNER(B7CFG)                                 +
                   DATA('B7CELL FabricBasicUser role')
  PERMIT           B7CELL.FabricBasicUser                       +
                   CLASS(EJBROLE)                               +
                   RESET
  PERMIT           B7CELL.FabricBasicUser                       +
                   CLASS(EJBROLE)                               +
                   ID(B7FBUG,B7FADMG)                           +
                   ACCESS(READ)
   /*   Define FabricStudioUser role              */
  RDEFINE EJBROLE  B7CELL.FabricStudioUser                      +
                   UACC(NONE)                                   +
                   OWNER(B7CFG)                                 +
                   DATA('B7CELL FabricStudioUser role')
  PERMIT           B7CELL.FabricStudioUser                      +
                   CLASS(EJBROLE)                               +
                   RESET
  PERMIT           B7CELL.FabricStudioUser                      +
                   CLASS(EJBROLE)                               +
                   ID(B7FSUG,B7FADMG)                           +
                   ACCESS(READ)
 SETR RACLIST(EJBROLE) REFRESH
 SEARCH MASK(B7) CLASS(EJBROLE)
/*
```

7.2 Installing WebSphere Business Services Fabric for z/OS in the deployment manager

In this section, we describe the steps to install WebSphere Business Services Fabric for z/OS into a clustered environment.

To install WebSphere Business Services Fabric for z/OS into a clustered environment:

1. Take a backup of your deployment manager and managed nodes file systems.
2. Install WebSphere Business Services Fabric for z/OS in deployment manager.
3. Install WebSphere Business Services Fabric for z/OS in a managed node.

The majority of the actual configuration for WebSphere Business Services Fabric for z/OS occurs during the profile augmentation process. This process is unique for each node and the deployment manager. We go through a series of verification steps to ensure that the augmentation worked in the manner expected. We then start the deployment manager and the node agents for the next steps.

Previous versions of WebSphere Business Services Fabric for z/OS required many manual configuration steps to properly set up and configure a WebSphere Process Server for z/OS server for Fabric. Improvements for V7.0 removed most of these requirements. Most configuration for WebSphere Business Services Fabric for z/OS V7.0 is accomplished through profile augmentation.

Next, we describe how to install and augment an existing WebSphere Process Server for z/OS deployment manager profile to contain WebSphere Business Services Fabric for z/OS functions. With V7.0 of WebSphere Business Services Fabric for z/OS, the installation and augmentation is aligned to the installation and augmentation of WebSphere Process Server for z/OS, which means that it is the same two-step approach:

- Running a zXXXInstall script to link the product file system to the configuration file system
- Preparing a response file and running a zXXXConfig script to augment the existing profile with the new product code

In zXXXInstall and zXXXConfig, XXX represents WPS for the WebSphere Process Server for z/OS installation and augmentation and WBSF for WebSphere Business Services Fabric for z/OS installation and augmentation.

7.2.1 Installing WebSphere Business Services Fabric for z/OS into the deployment manager profile

In this section, we describe how to install WebSphere Business Services Fabric for z/OS.

Prerequisites for Fabric installation

Before installing WebSphere Business Services Fabric for z/OS into the deployment manager profile, make sure you complete the following tasks:

1. Ensure that you fully configured your WebSphere Process Server for z/OS cluster environment and verified its basic operation. WebSphere Business Services Fabric for z/OS no longer provides its own database SQL scripts; instead, the database artifacts that WebSphere Business Services Fabric for z/OS uses are already configured with WebSphere Process Server for z/OS in the common database.
2. Ensure that you complete the security requirements for your Fabric installation. See “Configuring WebSphere Business Services Fabric for z/OS security” on page 217.

3. Ensure that SMP/E work for Fabric is complete.
4. Ensure that you shut down the cell.
5. Ensure that you have enough space in the target HFS and in /tmp to complete the installation process. The target HFS needs a minimum of 100 MB of free space for the WebSphere Business Services Fabric for z/OS installation.

Installing WebSphere Business Services Fabric for z/OS

After you augment your deployment manager profile with WebSphere Process Server for z/OS, you can install support for WebSphere Business Services Fabric for z/OS. The WebSphere Business Services Fabric for z/OS InfoCenter describes how to use zWBSFInstall.sh from a UNIX shell to install WebSphere Business Services Fabric for z/OS into your deployment manager profile. In the following steps, we describe how to use the shell script for the augmentation, and we provide a sample JCL that allows you to execute the augmentation script from TSO.

To install WebSphere Business Services Fabric for z/OS:

1. Open an OMVS or telnet session.
2. Switch to your administrator user ID:

   ```
   su b7admin (pwd='xxxxxx')
   ```

3. Change to directory /usr/lpp/zWBFKS/V7R0/zos.config/bin:

   ```
   cd /usr/lpp/zWBFKS/V7R0/zos.config/bin
   ```

4. Run the install script:

   ```
   zWBSFInstall.sh -smproot /usr/lpp/zWBFKS/V7R0 -runtime
   /wasv7config/b7cell/b7dmnode/DeploymentManager -install
   ```

It is important to run the zWBSFInstall.sh script from the WBSF zos.config/bin directory of the WebSphere Business Services Fabric for z/OS installation file system. Example 7-4 shows the output of the script execution.

Example 7-4 Sample output of zWBSFInstall.sh execution

```
B7ADMIN@ wtsc42:/usr/lpp/zWBFKS/V7R0/zos.config/bin>zWBSFInstall.sh -smproot /us
r/lpp/zWBFKS/V7R0 -runtime /wasv7config/b8cell/b8nodea/AppServer -install
CWPIZ0253I: parsing command arguments...
CWPIZ0254I: parsing arguments complete
CWPIZ0255I: setting up configuration...
Feb 1, 2010 3:39:50 PM null null
INFO: MSG_INITIAL_TRACE_STATE
Feb 1, 2010 3:39:51 PM null null
INFO: BBZWI044
Feb 1, 2010 3:39:51 PM null null
INFO: BBZWI275
Feb 1, 2010 3:39:51 PM null null
INFO: BBZWI117
Feb 1, 2010 3:39:51 PM null null
INFO: BBZWI024
Feb 1, 2010 3:39:51 PM null null
INFO: BBZWI014
Feb 1, 2010 3:39:51 PM null null
INFO: BBZWI052
Feb 1, 2010 3:39:51 PM null null
INFO: BBZWI045
```

```
Feb 1, 2010 3:39:51 PM null null
INFO: BBZWI248
CWPIZ0256I: set up configuration complete
CWPIZ0257I: creating the symbolic links...
CWPIZ0259I: creation of symbolic links complete
CWPIZ0260I: doing post install file updates...
CWPIZ0262I: post install updates complete
CWPIZ0263I: running Configuration Manager update...
Feb 1, 2010 3:39:54 PM null null
INFO: BBZWI218
Feb 1, 2010 3:39:54 PM null null
INFO: BBZWI219
CWPIZ0264I: Configuration Manager update complete
Feb 1, 2010 3:39:54 PM null null
INFO: BBZWI016
OSGi cache successfully cleaned for /wasv7config/b8cell/b8dmnode/DeploymentManag
er/profiles/default.
```

Null strings: In the response messages of the zWBSFInstall.sh script, we found several null strings instead of the real message values. But we concentrated on the CWPIZnnnnI messages and on the successful completion messages.

We created sample job B7IFABDM to install WebSphere Business Services Fabric for z/OS into the deployment manager profile, as shown in Example 7-5.

Example 7-5 Sample JCL

```
//B7IFABDM JOB (ACCTNO,ROOM),'Fabric DM install',CLASS=A,MSGCLASS=X,
//          MSGLEVEL=(1,1),NOTIFY=&SYSUID,REGION=0M
/*JOBPARM S=SC42
//***********************************************************/
//*  Install WBSF in deployment manager profile             */
//***********************************************************/
//* */
//INSFABDM EXEC PGM=IKJEFT01,REGION=0M,TIME=1440
//SYSTSPRT DD SYSOUT=*
//BPXOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//SYSTSIN  DD *
 BPXBATCH SH +
  cd /usr/lpp/zWBFKS/V7R0/zos.config/bin; +
  export PATH=.:$PATH; +
    zWBSFInstall.sh +
   -smproot /usr/lpp/zWBFKS/V7R0 +
   -runtime /wasv7config/b8cell/b8dmnode/DeploymentManager +
   -install;
//* */
//
```

Prerequisites for profile augmentation

Before augmenting a profile, ensure that you complete the following tasks:

1. Ensure that you successfully installed WebSphere Business Services Fabric for z/OS by executing zWBSFInstall.sh.
2. Ensure that the cell is shut down.
3. Ensure that you have enough space in the target HFS and in /tmp to complete the augmentation process.

> **Available free space:** The Fabric applications require a lot more space in your ZFS file systems, so they might need to extend during the deployment process. After installing Fabric, our deployment manager's node ZFS occupied 1800 CYLs and each node occupied more than 2000 CYLs.
>
> To ensure a clean deployment, ensure that the volumes that host your cell's ZFS datasets have enough free space to allow them to obtain secondary extents and grow to the sizes mentioned here.

Augmenting a profile with WebSphere Business Services Fabric for z/OS is very similar to augmenting a profile with WebSphere Process Server for z/OS. It comprises the following steps:

1. Prepare a response file.
2. Execute zWBSFConfig.sh with the prepared response file.

Usually these steps are the only two steps for augmenting a profile with the WebSphere Business Services Fabric for z/OS feature, but in our installation we run into a problem when we execute the augmentation script. After a short time, the script terminated with an error, as shown in Example 7-6.

Example 7-6 Output of failing augmentation

```
B7ADMIN @ wtsc42:/wasv7config/b7cell/b7dmnode/DeploymentManager/bin>zWBSFConfig.sh
-response /u/cherr/b7cell/fabricDmgrProfiles.rsp -augment
CWPIZ0253I: parsing command arguments...
CWPIZ0254I: parsing arguments complete
CWPIZ0255I: setting up configuration...
Feb 1, 2010 3:50:10 PM null null
INFO: MSG_INITIAL_TRACE_STATE
Feb 1, 2010 3:50:11 PM null null
INFO: BBZWI044
Feb 1, 2010 3:50:11 PM null null
INFO: BBZWI275
Feb 1, 2010 3:50:11 PM null null
INFO: BBZWI117
Feb 1, 2010 3:50:11 PM null null
INFO: BBZWI024
Feb 1, 2010 3:50:11 PM null null
INFO: BBZWI014
Feb 1, 2010 3:50:11 PM null null
INFO: BBZWI251
Feb 1, 2010 3:50:11 PM null null
INFO: BBZWI250
Feb 1, 2010 3:50:11 PM null null
INFO: BBZWI247
CWPIZ0256I: set up configuration complete
```

```
CWPIZO265I: augmenting profile(s)...
Feb 1, 2010 3:50:11 PM null null
INFO: BBZWI223
INSTCONFFAILED: Profile augmentation failed. For more information, consult
/wasv7config/b7cell/b7dmnode/DeploymentManager/logs/manageprofiles/default_augment
.log.
Feb 1, 2010 3:50:26 PM null null
SEVERE: BBZWI225
```

Looking into the log file default_augment.log, shown in Example 7-7, in the deployment manager's logs/manageprofiles directory, we found a permission problem with the shell script collectNodeMetadata.sh in the dmgr.wbsfabric template.

Example 7-7 Excerpt of log file "default_augment.log"

```
<record>
    <date>Feb 01, 2010 3:50:11 PM</date>
    <millis>1264622110327</millis>
    <sequence>1395</sequence>

<logger>com.ibm.ws.install.configmanager.actionengine.ant.utils.ANTLogToCmtLogAdap
ter</logger>
    <level>INFO</level>

<class>com.ibm.ws.install.configmanager.actionengine.ant.utils.ANTLogToCmtLogAdapt
er</class>
    <method>targetFinished</method>
    <thread>0</thread>
    <message>Target stopped for: runCollectNodeMetadata - FAILURE</message>
</record>
<record>
    <date>Feb 01, 2010 3:50:11 PM</date>
    <millis>1264622110327</millis>
    <sequence>1396</sequence>

<logger>com.ibm.ws.install.configmanager.actionengine.ant.utils.ANTLogToCmtLogAdap
ter</logger>
    <level>WARNING</level>
    <class>com.ibm.ws.install.configmanager.logging.LogUtils</class>
    <method>logException</method>
    <thread>0</thread>
    <message>Exception was thrown, type of exception is: class
org.apache.tools.ant.BuildException</message>
</record>
<record>
    <date>Feb 01, 2010 3:50:11 PM</date>
    <millis>1264622110327</millis>
    <sequence>1397</sequence>

<logger>com.ibm.ws.install.configmanager.actionengine.ant.utils.ANTLogToCmtLogAdap
ter</logger>
    <level>WARNING</level>
    <class>com.ibm.ws.install.configmanager.logging.LogUtils</class>
    <method>logException</method>
    <thread>0</thread>
```

```
   <message>The exception message is: Execute failed: java.io.IOException: Cannot
run program
"/wasv7config/b7cell/b7dmnode/DeploymentManager/profileTemplates/dmgr.wbsfabric/ac
tions/scripts/collectNodeMetadata.sh" (in directory
"/wasv7config/b7cell/b7dmnode/DeploymentManager/profileTemplates/dmgr.wbsfabric/ac
tions"):
/wasv7config/b7cell/b7dmnode/DeploymentManager/profileTemplates/dmgr.wbsfabric/act
ions/scripts/collectNodeMetadata.sh
```

The reason for this problem is that the script in the read-only installation file system has no execute permission set, as shown by the directory listing.

Directory listing:

```
B7ADMIN @ wtsc42:/u/cherr/b7cell>ls -al
/usr/lpp/zWBFKS/V7R0/profileTemplates/dmgr.wbsfabric/actions/scripts/
total 24
drwxr-xr-x   5 HAIMO    SYS1         480 Dec 14 21:44 .
drwxr-xr-x   3 HAIMO    SYS1         928 Dec 14 21:44 ..
drwxr-xr-x   2 HAIMO    SYS1         320 Dec 14 21:44 cluster
-rw-r--r--   2 HAIMO    SYS1        4091 Dec 14 21:44 collectNodeMetadata.bat
-rw-r--r--   2 HAIMO    SYS1        2476 Dec 14 21:44 collectNodeMetadata.sh
drwxr-xr-x   2 HAIMO    SYS1         544 Dec 14 21:44 configuration
drwxr-xr-x   2 HAIMO    SYS1         896 Dec 14 21:44 uninstallation
```

To get the augmentation working, we renamed the link to the shell script in the installation file system, created a local copy of that script, and set the necessary file permissions:

1. Open an OMVS or telnet session.
2. Switch to your WebSphere administrator user ID:

   ```
   su b7admin (pwd='xxxxxx')
   ```

3. Change to the collectNodeMetadata.sh script location in the deployment manager profileTemplates directory:

   ```
   cd
   /wasv7config/b7cell/b7dmnode/DeploymentManager/profileTemplates/dmgr.wbsfabric/
   actions/scripts
   ```

4. Rename the link to collectNodeMetadata.sh to collectNodeMetadata.sh.org:

   ```
   mv collectNodeMetadata.sh collectNodeMetadata.sh.org
   ```

5. Create a local copy of the original file:

   ```
   cp collectNodeMetadata.sh.org collectNodeMetadata.sh
   ```

6. Set the necessary file permissions:

   ```
   chmod 555 mv collectNodeMetadata.sh
   ```

Now the augmentation can run without problems.

Preparing the response file

To prepare the response file:

1. WebSphere Business Services Fabric for z/OS does not provide a plug-in for WCT so far for configuring the response file; therefore, we must do it manually. Copy the sample

response file, fabricDmgrProfiles.rsp, for the deployment manager from /usr/lpp/zWBFKS/V7R0/zos.config to your working directory.

2. Open the file with an editor, and modify the properties in Example 7-9, which are spread over the response file.

Example 7-8 Sample properties of dmgr response file

```
augment
profileName=default
profilePath=/wasv7config/b7cell/b7dmnode/DeploymentManager/profiles/default
templatePath=/wasv7config/b7cell/b7dmnode/DeploymentManager/profileTemplates/dm
gr.wbsfabric
cellName=b7cell
nodeName=b7dmnode
```

Note: There are several invalid comments and sample settings in the response file. The sample setting for templatePath in the file fabricDmgrProfiles.rsp must contain DeploymentManager instead of AppServer.

```
templatePath=/fabric1/v1cell/v1nodea/AppServer/profileTemplates/dmgr.wbsfabri
c
```

3. Make sure the response file has read permission for the WebSphere administrator user b7admin.
4. Save the response file.
5. Perform the augmentation of the deployment manager profile using the modified response file.

Augmenting the deployment manager’s profile Fabric

To augment the deployment manager’s profile Fabric:

1. Open an OMVS or telnet session.
2. Switch to your WebSphere administrator user ID:

   ```
   su b7admin (pwd=’xxxxxx’)
   ```

3. Change to directory deployment manager bin directory /wasv7config/b7cell/b7dmnode/DeploymentManager/bin:

   ```
   cd /wasv7config/b7cell/b7dmnode/DeploymentManager/bin
   ```

4. Run the augmentation script:

   ```
   zWBSFConfig.sh -response /u/cherr/b7cell/fabricDmgrProfiles.rsp -augment
   ```

   Example 7-9 shows the output of the script.

Example 7-9 Output of successful augmentation

```
B7ADMIN @ wtsc42:/wasv7config/b7cell/b7dmnode/DeploymentManager/bin>zWBSFConfig.s
h -response /u/cherr/b7cell/fabricDmgrProfiles.rsp -augment
CWPIZ0253I: parsing command arguments...
CWPIZ0254I: parsing arguments complete
CWPIZ0255I: setting up configuration...
Feb 1, 2010 3:45:57 PM null null
INFO: MSG_INITIAL_TRACE_STATE
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI044
```

```
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI275
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI117
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI024
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI014
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI251
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI250
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI247
CWPIZ0256I: set up configuration complete
CWPIZ0265I: augmenting profile(s)...
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI223
INSTCONFSUCCESS: Profile augmentation succeeded.
Feb 1, 2010 3:48:00 PM null null
INFO: BBZWI224
CWPIZ0267I: augmenting profile(s) complete
Feb 1, 2010 3:48:00 PM null null
INFO: BBZWI016
```

Sample JCL: We created a sample JCL that performs all of the correction steps and the augmentation. The JCL is in the additional material of this book.

Example 7-10 Sample JCL B7AFABDM for augmentation of deployment manager profile

```
//B7AFABDM  JOB (ACCTNO,ROOM),'Fabric DM augment',CLASS=A,MSGCLASS=X,
//          MSGLEVEL=(1,1),NOTIFY=&SYSUID,REGION=0M
/*JOBPARM S=SC42
//***********************************************************/
//*  Augment deployment manager profile with WBSF           */
//***********************************************************/
//* */
//FIXSCR   EXEC PGM=IKJEFT01,REGION=0M,TIME=1440
//SYSTSPRT DD SYSOUT=*
//BPXOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//SYSTSIN  DD *
  BPXBATCH SH +
   cd /wasv7config/b7cell/b7dmnode/DeploymentManager/+
   profileTemplates/dmgr.wbsfabric/actions/scripts; +
   mv collectNodeMetadata.sh collectNodeMetadata.org; +
   cp collectNodeMetadata.org collectNodeMetadata.sh; +
   chmod 555 collectNodeMetadata.sh;
//* */
//AUGFABDM EXEC PGM=IKJEFT01,REGION=0M,TIME=1440
//SYSTSPRT DD SYSOUT=*
//BPXOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
```

```
//SYSTSIN  DD *
 BPXBATCH SH +
  cd /wasv7config/b7cell/b7dmnode/DeploymentManager/bin; +
  zWBSFConfig.sh +
   -response /u/cherr/b7cell/fabricDmgrProfiles.rsp +
   -augment;
//* */
//
```

You successfully augmented the deployment manager profile with the WebSphere Business Services Fabric for z/OS feature. Next, you install WebSphere Business Services Fabric for z/OS into each node in your cell, and augment the node profiles.

7.3 Installing WebSphere Business Services Fabric for z/OS in a managed node

In this section, we describe how to install WebSphere Business Services Fabric for z/OS into a managed node.

7.3.1 Prerequisites for Fabric installation

Before installing WebSphere Business Services Fabric for z/OS into the nodes, the same prerequisites apply as for the WebSphere Business Services Fabric for z/OS installation in the deployment manager, as explained in “Prerequisites for Fabric installation” on page 221.

7.3.2 Installing WebSphere Business Services Fabric for z/OS

Install WebSphere Business Services Fabric for z/OS in node b7nodea:

1. Open an OMVS or telnet session.
2. Switch to your administrator user ID:

   ```
   su b7admin (pwd=’xxxxxx’)
   ```

3. Change to directory /usr/lpp/zWBFKS/V7R0/zos.config/bin:

   ```
   cd /usr/lpp/zWBFKS/V7R0/zos.config/bin
   ```

4. Run the install script:

   ```
   zWBSFInstall.sh -smproot /usr/lpp/zWBFKS/V7R0 -runtime
   /wasv7config/b7cell/b7nodea/AppServer -install
   ```

Example 7-4 on page 222 shows the output of the script execution.

Example 7-11 Sample output of zWBSFInstall.sh execution for node b7nodea

```
B7ADMIN@ wtsc42:/usr/lpp/zWBFKS/V7R0/zos.config/bin>zWBSFInstall.sh -smproot /us
r/lpp/zWBFKS/V7R0 -runtime /wasv7config/b7cell/b7nodea/AppServer -install
CWPIZ0253I: parsing command arguments...
CWPIZ0254I: parsing arguments complete
CWPIZ0255I: setting up configuration...
Feb 1, 2010 3:39:50 PM null null
INFO: MSG_INITIAL_TRACE_STATE
Feb 1, 2010 3:39:51 PM null null
```

```
INFO: BBZWI044
Feb 1, 2010 3:39:51 PM null null
INFO: BBZWI275
Feb 1, 2010 3:39:51 PM null null
INFO: BBZWI117
Feb 1, 2010 3:39:51 PM null null
INFO: BBZWI024
Feb 1, 2010 3:39:51 PM null null
INFO: BBZWI014
Feb 1, 2010 3:39:51 PM null null
INFO: BBZWI052
Feb 1, 2010 3:39:51 PM null null
INFO: BBZWI045
Feb 1, 2010 3:39:51 PM null null
INFO: BBZWI248
CWPIZ0256I: set up configuration complete
CWPIZ0257I: creating the symbolic links...
CWPIZ0259I: creation of symbolic links complete
CWPIZ0260I: doing post install file updates...
CWPIZ0262I: post install updates complete
CWPIZ0263I: running Configuration Manager update...
Feb 1, 2010 3:39:54 PM null null
INFO: BBZWI218
Feb 1, 2010 3:39:54 PM null null
INFO: BBZWI219
CWPIZ0264I: Configuration Manager update complete
Feb 1, 2010 3:39:54 PM null null
INFO: BBZWI016
OSGi cache successfully cleaned for
/wasv7config/b7cell/b7nodea/AppServer/profiles/default.
```

We created the sample job B7IFABA to install WebSphere Business Services Fabric for z/OS into the managed node profile, as shown in Example 7-12.

Example 7-12 Sample JCL for installing fabric in a managed node

```
//B7IFABA JOB (ACCTNO,ROOM),'Fabric DM install',CLASS=A,MSGCLASS=X,
//          MSGLEVEL=(1,1),NOTIFY=&SYSUID,REGION=0M
/*JOBPARM S=SC42
//*******************************************************/
//*  Install WBSF in a managed node profile              */
//*******************************************************/
//* */
//INSFABMN EXEC PGM=IKJEFT01,REGION=0M,TIME=1440
//SYSTSPRT DD SYSOUT=*
//BPXOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//SYSTSIN  DD *
 BPXBATCH SH +
  cd /usr/lpp/zWBFKS/V7R0/zos.config/bin; +
  export PATH=.:$PATH; +
    zWBSFInstall.sh +
   -smproot /usr/lpp/zWBFKS/V7R0 +
   -runtime /wasv7config/b7cell/b7nodea/AppServer +
   -install;
//* */
//
```

Preparing the response file

Now, you can prepare the response file for the augmentation of the node profile of node b7nodea with the WebSphere Business Services Fabric for z/OS feature:

1. Copy the sample response file, fabricManagedProfiles.rsp, for the managed node from /usr/lpp/zWBFKS/V7R0/zos.config to your working directory, for example, /u/cherr/b7cell.
2. Open the file with an editor, and modify the properties shown in Example 7-13, which are spread over the response file.

   Example 7-13 Sample properties of managed node response file

   ```
   augment
   profileName=default
   profilePath=/wasv7config/b7cell/b7nodea/AppServer/profiles/default
   templatePath=/wasv7config/b7cell/b7nodea/Appserver/profileTemplates/managed.wbs
   fabric
   cellName=b7cell
   nodeName=b7nodea
   ```

   > **Note:** There are several invalid comments and sample settings in the response file, for example, the sample setting for templatePath in the file fabricDmgrProfiles.rsp must contain DeploymentManager instead of AppServer:
   >
   > templatePath=/fabric1/v1cell/v1nodea/**AppServer**/profileTemplates/dmgr.wbsfabr
   > ic

3. Save the response file.

Now perform the augmentation of the managed node profile using the modified response file. There is an issue with executing a shell script in the WebSphere Business Services Fabric for z/OS templates, which also applies to the WebSphere Business Services Fabric for z/OS template for the managed node. Therefore you must fix this problem before running the augmentation script:

1. Open an OMVS or telnet session.
2. Switch to your WebSphere administrator user ID:

   ```
   su b7admin (pwd=’xxxxxx’)
   ```

3. Change to the collectNodeMetadata.sh script location in the managed node profileTemplates directory:

   ```
   cd
   /wasv7config/b7cell/b7nodea/AppServer/profileTemplates/managed.wbsfabric/action
   s/scripts
   ```

4. Rename the link to collectNodeMetadata.sh to collectNodeMetadata.sh.org:

   ```
   mv collectNodeMetadata.sh collectNodeMetadata.sh.org
   ```

5. Create a local copy of the original file:

   ```
   cp collectNodeMetadata.sh.org collectNodeMetadata.sh
   ```

6. Set the necessary file permissions:

   ```
   chmod 555 mv collectNodeMetadata.sh
   ```

7. The augmentation can now run without problems. Change to the managed node's bin directory, /wasv7config/b7cell/b7nodea/AppServer/bin:

   ```
   cd /wasv7config/b7cell/b7nodea/AppServer/bin
   ```

8. Run the augmentation script:

   ```
   zWBSFConfig.sh -response /u/cherr/b7cell/fabricManagedProfiles.rsp -augment
   ```

   Example 7-14 shows the output of the script.

Example 7-14 Output of augmentation script for managed node b7nodea

```
B7ADMIN @ wtsc42:/wasv7config/b7cell/b7nodea/AppServer/bin>zWBSFConfig.s
h -response /u/cherr/b7cell/fabricManagedProfiles.rsp -augment
CWPIZ0253I: parsing command arguments...
CWPIZ0254I: parsing arguments complete
CWPIZ0255I: setting up configuration...
Feb 1, 2010 3:45:57 PM null null
INFO: MSG_INITIAL_TRACE_STATE
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI044
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI275
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI117
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI024
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI014
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI251
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI250
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI247
CWPIZ0256I: set up configuration complete
CWPIZ0265I: augmenting profile(s)...
Feb 1, 2010 3:45:59 PM null null
INFO: BBZWI223
INSTCONFSUCCESS: Profile augmentation succeeded.
Feb 1, 2010 3:48:00 PM null null
INFO: BBZWI224
CWPIZ0267I: augmenting profile(s) complete
Feb 1, 2010 3:48:00 PM null null
INFO: BBZWI016
```

There is also a sample JCL for augmenting a managed node with WebSphere Business Services Fabric for z/OS, as shown in Example 7-15.

Example 7-15 Sample JCL for managed node augmentation

```
//B7AFABA JOB (ACCTNO,ROOM),'Fabric MN augment',CLASS=A,MSGCLASS=X,
//          MSGLEVEL=(1,1),NOTIFY=&SYSUID,REGION=0M
/*JOBPARM S=SC42
```

```
//*********************************************************/
//*  Augment managed node profile with WBSF              */
//*********************************************************/
//* */
//FIXSCR   EXEC PGM=IKJEFT01,REGION=0M,TIME=1440
//SYSTSPRT DD SYSOUT=*
//BPXOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//SYSTSIN  DD *
  BPXBATCH SH +
   cd /wasv7config/b7cell/b7nodea/AppServer/+
   profileTemplates/managed.wbsfabric/actions/scripts; +
   mv collectNodeMetadata.sh collectNodeMetadata.org; +
   cp collectNodeMetadata.org collectNodeMetadata.sh; +
   chmod 555 collectNodeMetadata.sh;
//* */
//AUGFABDM EXEC PGM=IKJEFT01,REGION=0M,TIME=1440
//SYSTSPRT DD SYSOUT=*
//BPXOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//SYSTSIN  DD *
 BPXBATCH SH +
  cd /wasv7config/b7cell/b7nodea/AppServer/bin; +
  zWBSFConfig.sh +
   -response /u/cherr/b7cell/fabricManagedProfiles.rsp +
   -augment;
//* */
//
```

You successfully augmented the b7nodea node profile with WebSphere Business Services Fabric for z/OS. If you have further nodes in your cell, repeat the instructions in this chapter for every node in your cell.

7.4 Completing cluster configuration for WebSphere Business Services Fabric for z/OS

After successful augmentation of the deployment manager and all of the node profiles with the WebSphere Business Services Fabric for z/OS feature you can now install the WebSphere Business Services Fabric for z/OS applications. The installation of the WebSphere Business Services Fabric for z/OS applications is achieved by executing a Jython script in wsadmin because there is no WebSphere Business Services Fabric for z/OS installation wizard in the ISC that is available yet:

1. Open an OMVS or telnet session.
2. Switch to your administrator user ID:

   ```
   su b7admin (pwd=’xxxxxx’)
   ```

3. Change to the bin directory of the deployment manager profile:

   ```
   cd /wasv7config/b7cell/b7dmnode/DeploymentManager/profiles/default/bin
   ```

4. Execute the WebSphere Business Services Fabric for z/OS installation script fabricAppDeploy.py. This script is located in the WebSphere Business Services Fabric for z/OS template of the deployment manager. The script can be either run in interactive or in non-interactive mode. If you are not sure which parameters and values to specify with the script, run it in interactive mode. It only shows valid values for the parameters. We used the non-interactive mode for our installation:

   ```
   wsadmin.sh -port 28003 -conntype RMI -user b7admin -password b7admin -f
   /wasv7config/b7cell/b7dmnode/DeploymentManager/profileTemplates/dmgr.wbsfabric/
   actions/scripts/cluster/fabricAppDeploy.py -cluster b7sr1_WPS -toolsUserid
   b7fabadm
   ```

   While executing the script, you will see an output similar to Example 7-16.

Example 7-16 Output of WebSphere Business Services Fabric for z/OS installation script

```
B7ADMIN @
wtsc42:/wasv7config/b7cell/b7dmnode/DeploymentManager/profiles/default/bin>wsadmin
.sh -user b7admin -password <xxxxx> -port 27003 -conntype RMI -f
/wasv7config/b7cell/b7dmnode/DeploymentManager/profileTemplates/dmgr.wbsfabric/act
ions/scripts/cluster/fabricAppDeploy.py -cluster b7sr1_WPS -toolsUserid b7fabadm
Trace: 2010/02/21 10:44:53.160 01 t=7E5E00 c=UNK key=P8 (03020004)
  Description: ANR not supported
WASX7209I: Connected to process "dmgr" on node b7dmnode using RMI connector;  The
type of process is: DeploymentManager
WASX7303I: The following options are passed to the scripting environment and
areavailable as arguments that are stored in the argv variable: "[-cluster,
b7sr1_WPS, -toolsUserid, b7fabadm]"
Deploying Fabric enterprise applications.
WASX7327I: Contents of was.policy file:
 grant codeBase "file:${application}" {
        permission java.security.AllPermission;
};

ADMA5016I: Installation of Fabric_Catalog_b7sr1_WPS started.
ADMA5058I: Application and module versions are validated with versions of
deployment targets.
ADMA5005I: The application Fabric_Catalog_b7sr1_WPS is configured in the WebSphere
Application Server repository.
ADMA5053I: The library references for the installed optional package are created.
ADMA5005I: The application Fabric_Catalog_b7sr1_WPS is configured in the WebSphere
Application Server repository.
ADMA5001I: The application binaries are saved in
/wasv7config/b7cell/b7dmnode/DeploymentManager/profiles/default/wstemp/Script126f0
21362c/workspace/cells/b7cell/applications/Fabric_Catalog_b7sr1_WPS.ear/Fabric_Cat
alog_b7sr1_WPS.ear
ADMA5005I: The application Fabric_Catalog_b7sr1_WPS is configured in the WebSphere
Application Server repository.
SECJ0400I: Successfully updated the application Fabric_Catalog_b7sr1_WPS with the
appContextIDForSecurity information.
ADMA5005I: The application Fabric_Catalog_b7sr1_WPS is configured in the WebSphere
Application Server repository.
ADMA5113I: Activation plan created successfully.
ADMA5011I: The cleanup of the temp directory for application
Fabric_Catalog_b7sr1_WPS is complete.
ADMA5013I: Application Fabric_Catalog_b7sr1_WPS installed successfully.
WASX7327I: Contents of was.policy file:
```

```
 grant codeBase "file:${application}" {
        permission java.security.AllPermission;
};

ADMA5016I: Installation of Fabric_Trace_b7sr1_WPS started.
ADMA5058I: Application and module versions are validated with versions of
deployment targets.
ADMA5005I: The application Fabric_Trace_b7sr1_WPS is configured in the WebSphere
Application Server repository.
ADMA5053I: The library references for the installed optional package are created.
ADMA5005I: The application Fabric_Trace_b7sr1_WPS is configured in the WebSphere
Application Server repository.
ADMA5001I: The application binaries are saved in
/wasv7config/b7cell/b7dmnode/DeploymentManager/profiles/default/wstemp/Script126f0
21362c/workspace/cells/b7cell/applications/Fabric_Trace_b7sr1_WPS.ear/Fabric_Trace
_b7sr1_WPS.ear
ADMA5005I: The application Fabric_Trace_b7sr1_WPS is configured in the WebSphere
Application Server repository.
SECJ0400I: Successfully updated the application Fabric_Trace_b7sr1_WPS with the
appContextIDForSecurity information.
ADMA5005I: The application Fabric_Trace_b7sr1_WPS is configured in the WebSphere
Application Server repository.
ADMA5113I: Activation plan created successfully.
ADMA5011I: The cleanup of the temp directory for application
Fabric_Trace_b7sr1_WPS is complete.
ADMA5013I: Application Fabric_Trace_b7sr1_WPS installed successfully.
WASX7327I: Contents of was.policy file:
 grant codeBase "file:${application}" {
        permission java.security.AllPermission;
};

ADMA5016I: Installation of Fabric_Tools_b7sr1_WPS started.
ADMA5058I: Application and module versions are validated with versions of
deployment targets.
ADMA5005I: The application Fabric_Tools_b7sr1_WPS is configured in the WebSphere
Application Server repository.
ADMA5053I: The library references for the installed optional package are created.
ADMA5005I: The application Fabric_Tools_b7sr1_WPS is configured in the WebSphere
Application Server repository.
ADMA5001I: The application binaries are saved in
/wasv7config/b7cell/b7dmnode/DeploymentManager/profiles/default/wstemp/Script126f0
21362c/workspace/cells/b7cell/applications/Fabric_Tools_b7sr1_WPS.ear/Fabric_Tools
_b7sr1_WPS.ear
ADMA5005I: The application Fabric_Tools_b7sr1_WPS is configured in the WebSphere
Application Server repository.
SECJ0400I: Successfully updated the application Fabric_Tools_b7sr1_WPS with the
appContextIDForSecurity information.
ADMA5005I: The application Fabric_Tools_b7sr1_WPS is configured in the WebSphere
Application Server repository.
ADMA5113I: Activation plan created successfully.
ADMA5011I: The cleanup of the temp directory for application
Fabric_Tools_b7sr1_WPS is complete.
ADMA5013I: Application Fabric_Tools_b7sr1_WPS installed successfully.
Deployed Fabric enterprise applications.
Updating rest service: {com.ibm.bpm}BSLM
Updating rest service: {com.ibm.bpm}Glossary
Updating rest service: {com.ibm.bpm}Governance
Updating rest service: {com.ibm.bpm}BVARS
Successfully updated fabric rest services.
Done with synchronization for node b7nodea
```

There is also a sample JCL that is available for deploying the WebSphere Business Services Fabric for z/OS applications, as shown in Example 7-17.

Example 7-17 Sample JCL for deploying WebSphere Business Services Fabric for z/OS applications

```
//B7IFABAP JOB (ACCTNO,ROOM),'INS FAB AP',CLASS=A,MSGCLASS=X,REGION=0M,
//        MSGLEVEL=(1,1),NOTIFY=&SYSUID
/*JOBPARM S=SC42
//***********************************************************/
//*  Make sure you edit this file with CAPS OFF.            */
//*  Run this job using an administrator userid.            */
//***********************************************************/
//*  Install Fabric applications                            */
//***********************************************************/
//* */
//IFABAP EXEC PGM=IKJEFT01,REGION=0M,TIME=1440
//SYSTSPRT DD SYSOUT=*
//BPXOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//SYSTSIN  DD *
 BPXBATCH SH +
  cd /wasv7config/b7cell/b7dmnode/DeploymentManager/profiles+
    /default/bin'; +
  wsadmin.sh +
   -lang jython
   -user b7admin +
   -password b7admin +
   -conntype RMI +
   -port 27003 +
   -f /wasv7config/b7cell/b7dmnode/DeploymentManager/profile+
   Templates/dmgr.wbsfabric/actions/scripts/cluster/fabricApp+
   Deploy.py +
   -cluster b7sr1_WPS +
   -toolsUserid b7fabadm;
//* */
//
```

You installed WebSphere Business Services Fabric for z/OS successfully in your cell.

7.5 Verifying the WebSphere Business Services Fabric for z/OS configuration

After you install WebSphere Business Services Fabric for z/OS in your cell, check the list of installed applications in the ISC. There are three new applications in the list:

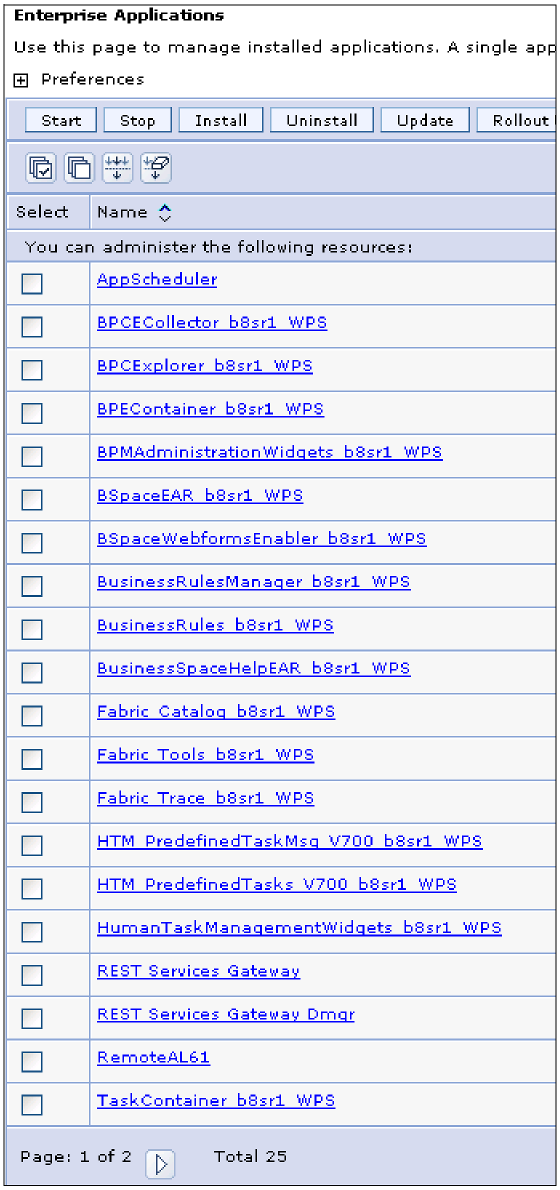
- Fabric_Catalog_b7sr1_WPS
- Fabric_Tools_b7sr1_WPS
- Fabric_Trace_b7sr1_WPS

The following verification steps are from the similar cell b8cell:

1. Login to the Integrated Solutions Console.
2. Click **Applications** → **Application Types** → **WebSphere enterprise applications**.

   A list of applications similar to the two Figure 7-2 on page 238 and Figure 7-3 on page 239 are displayed. This list contains the three WebSphere Business Services Fabric for z/OS applications:

   – Fabric_Tools_b8sr1_WPS
   – Fabric_Catalog_b8sr1_WPS
   – Fabric_Trace_b8sr1_WPS



Enterprise Applications

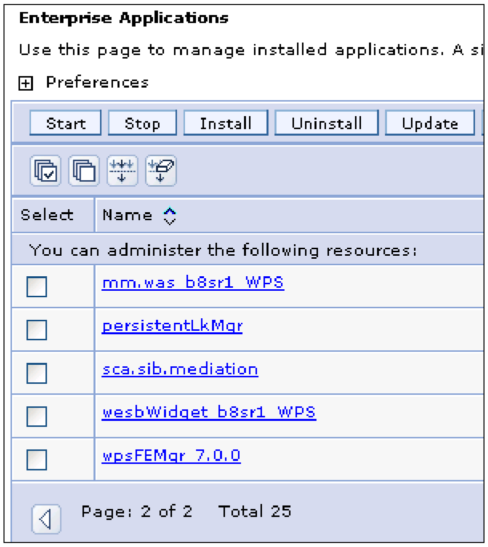
Use this page to manage installed applications. A single app

⊞ Preferences

Select	Name
You can administer the following resources:	
□	AppScheduler
□	BPCECollector b8sr1 WPS
□	BPCExplorer b8sr1 WPS
□	BPEContainer b8sr1 WPS
□	BPMAdministrationWidgets b8sr1 WPS
□	BSpaceEAR b8sr1 WPS
□	BSpaceWebformsEnabler b8sr1 WPS
□	BusinessRulesManager b8sr1 WPS
□	BusinessRules b8sr1 WPS
□	BusinessSpaceHelpEAR b8sr1 WPS
□	Fabric Catalog b8sr1 WPS
□	Fabric Tools b8sr1 WPS
□	Fabric Trace b8sr1 WPS
□	HTM PredefinedTaskMsg V700 b8sr1 WPS
□	HTM PredefinedTasks V700 b8sr1 WPS
□	HumanTaskManagementWidgets b8sr1 WPS
□	REST Services Gateway
□	REST Services Gateway Dmgr
□	RemoteAL61
□	TaskContainer b8sr1 WPS

Page: 1 of 2 ▷ Total 25

Figure 7-2 Installed applications after WebSphere Business Services Fabric installation



Enterprise Applications

Use this page to manage installed applications. A si

⊞ Preferences

Select	Name
You can administer the following resources:	
□	mm.was_b8sr1_WPS
□	persistentLkMgr
□	sca.sib.mediation
□	wesbWidget_b8sr1_WPS
□	wpsFEMgr_7.0.0

Page: 2 of 2   Total 25

Figure 7-3 Installed applications after WebSphere Business Services Fabric installation

Next step of WebSphere Business Services Fabric for z/OS installation verification is to login to the WebSphere Business Services Fabric console.

Logging into WebSphere Business Services Fabric console

Before logging in to WebSphere Business Services Fabric console, make sure the cluster, servers, and the applications are started and running.

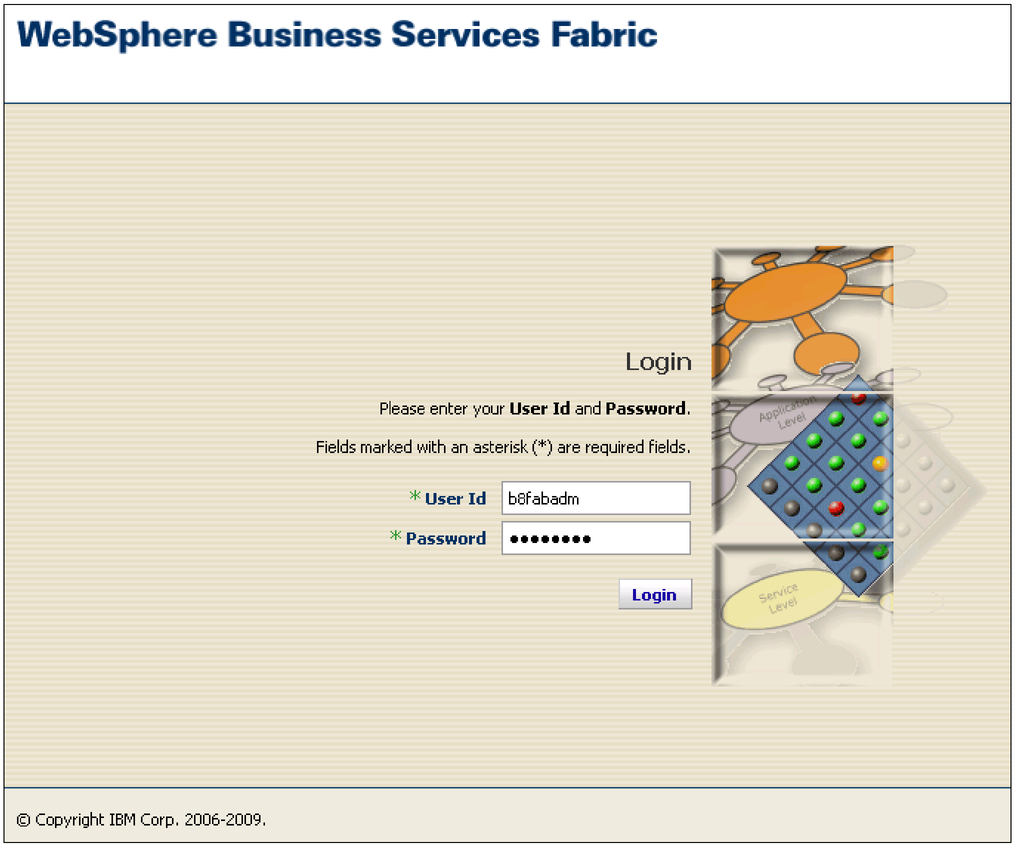
Figure 7-4 on page 240 through Figure 7-7 on page 242are from the similar cell b8cell.

To login to the WebSphere Business Services Fabric console:

1. Type the following URL in the browser:

   `http://wpsplex.itso.ibm.com:28047/fabric/app.`

2. Figure 7-4 on page 240 is displayed. Login with B8FABADM, which is the user ID that you added to RACF as the Fabric administrator user ID.



WebSphere Business Services Fabric

Login

Please enter your **User Id** and **Password**.

Fields marked with an asterisk (*) are required fields.

* **User Id** b8fabadm

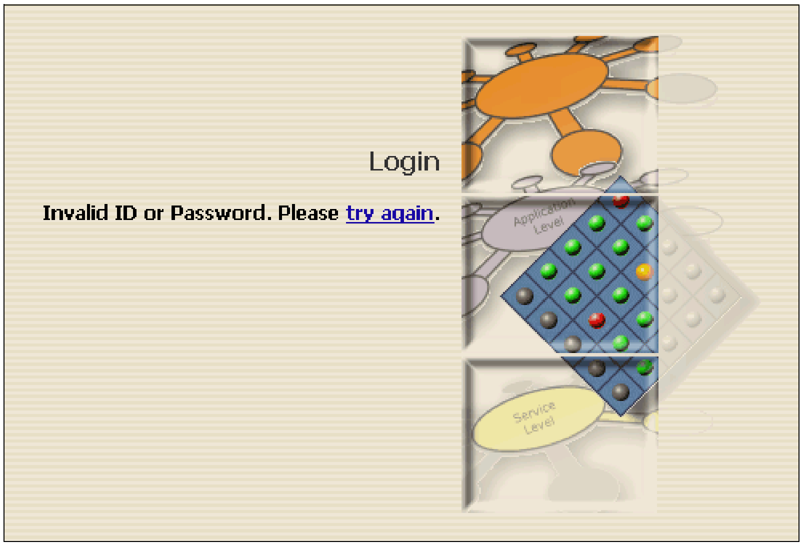
* **Password** ••••••••

Login

© Copyright IBM Corp. 2006-2009.

Figure 7-4 Log in to WebSphere Business Services Fabric

If the user ID does not have access to the required role, you will receive the error shown in Figure 7-5.



Login

Invalid ID or Password. Please try again.

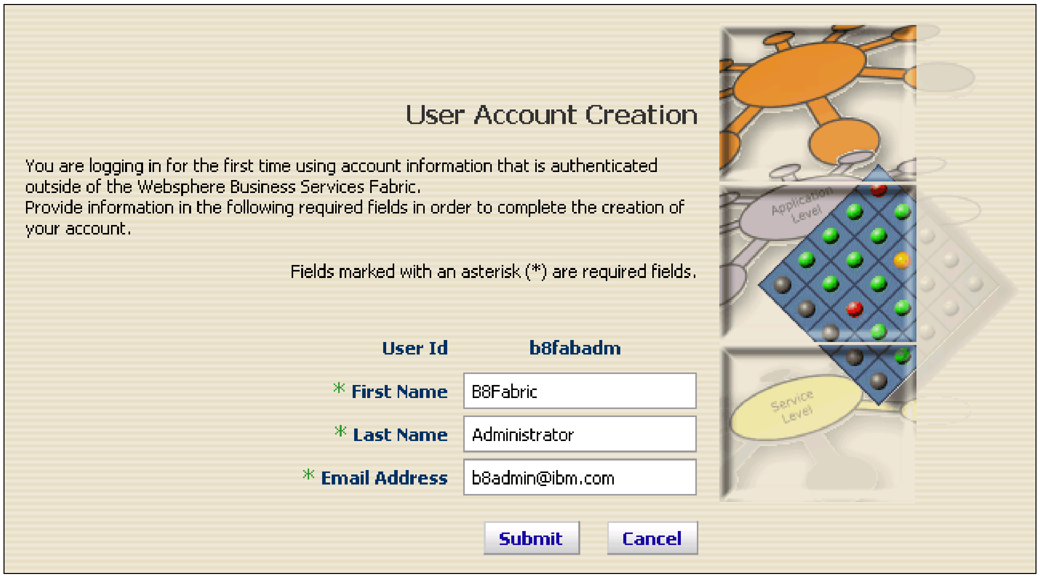
Figure 7-5 Log in failed

3. When the WebSphere Process Server for z/OS cell is using a RACF user registry, go to the z/OS SYSLOG, where you will see a related ICH408I error, as shown in Example 7-18 on page 241.

Example 7-18 RACF ICH408I error when unauthorized to Fabric role(s)

```
ICH408I USER(B8FABADM) GROUP(B8FADMG ) NAME(B8CELL FABRIC ADMIN      )
  B8CELL.FabricStudioUser CL(EJBROLE )
  INSUFFICIENT ACCESS AUTHORITY
  ACCESS INTENT(READ    )  ACCESS ALLOWED(NONE    )
+BBOO0222I: SECJ0129E: Authorization failed for B8FABADM while invoking GET on
default_host:fabric/app, Authorization failed, Not granted any of the required
roles: FabricAdministrator
 FabricGovernanceAdministrator FabricPerformanceUser
 FabricSubscriberManager FabricBasicUser FabricStudioUser
```

4. To resolve this error, permit the user ID to one of the EJBROLE profiles. In this case the user ID B8FABADM, which we wanted to be the Fabric administrator user ID, has not been granted to any of the roles.
5. The first time you log in, you are prompted to register your name and email address, as shown in Figure 7-6.



User Account Creation

You are logging in for the first time using account information that is authenticated outside of the Websphere Business Services Fabric.
Provide information in the following required fields in order to complete the creation of your account.

Fields marked with an asterisk (*) are required fields.

User Id b8fabadm

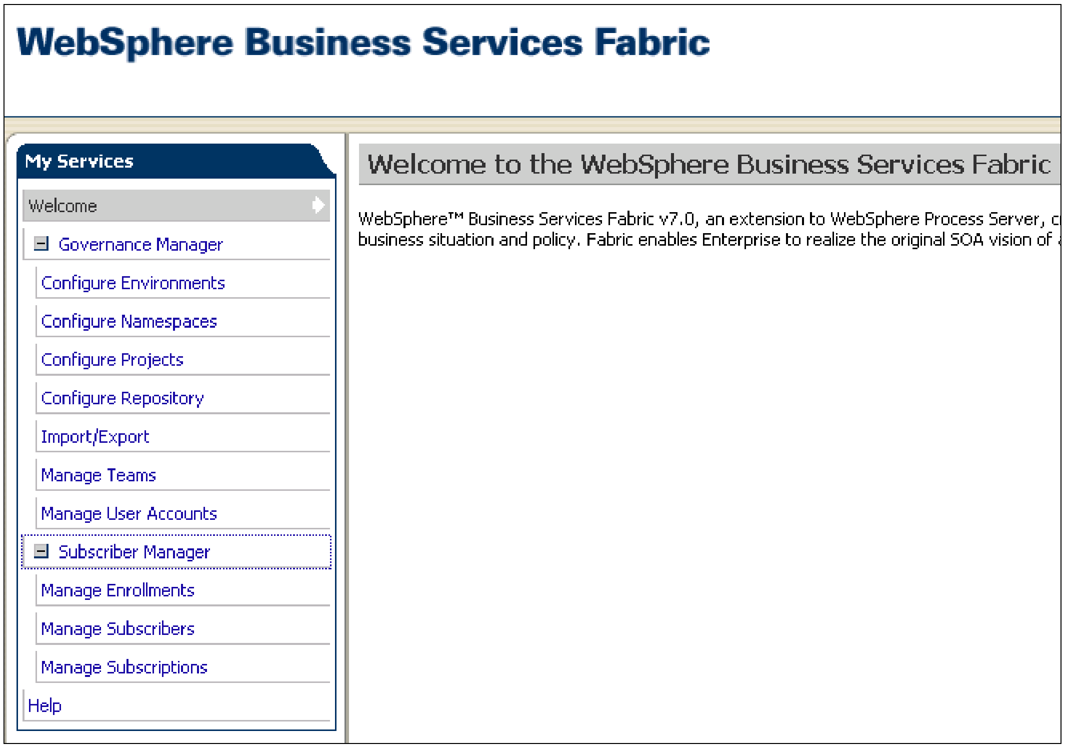
* First Name: B8Fabric

* Last Name: Administrator

* Email Address: b8admin@ibm.com

Figure 7-6 First time login user account creation

6. After registering, the Welcome to the WebSphere Business Services Fabric for z/OS window, Figure 7-7 on page 242, is displayed.



Welcome to the WebSphere Business Services Fabric

WebSphere™ Business Services Fabric v7.0, an extension to WebSphere Process Server, c
business situation and policy. Fabric enables Enterprise to realize the original SOA vision of

Figure 7-7 Successful login to WebSphere Business Services Fabric console

7. Verify if the following tasks are completed for the Fabric interactive script, fabricAppDeploy.py:

   The following Name Space Bindings are created at the Application Cluster level:

   – wbsf.da.executionTrace.maxcount
   – wbsf.da.executionTrace.viewer.enable

8. A Fabric DA Replication domain is created for the context replication. The number of replicas is targeted to an entire domain.

9. An Object Cache instance named Fabric Context Cache is created at the Application Cluster level.

   The Application cluster name is provided when running the script.

10. The Fabric Context Cache instance has the Enable the cache replication option turned on. The Full group replication domain is targeted to Fabric DA Replication.

Your WebSphere Business Services Fabric for z/OS installation verification is successfully.

8

Problem determination and prevention

In this chapter, we discuss problems that we encountered and describe how we discovered and fixed them. We also provide techniques and tools that we use in our troubleshooting experiences.

Effective problem diagnosis is not an exact science. It is based on a thorough understanding of WebSphere products and z/OS, experience (usually unsuccessful ones), pattern recognition, contacts, and the ability to use many tools. Your best tools are your brain, memory, ability to search many information sources, and communication with other subject-matter experts. Here are a few important points:

- 8.1, “Problem prevention” on page 244:
  - Good naming conventions, organization, and management practices
  - Hardware and software resources
  - Required skills
- 8.2, “Tools and techniques” on page 245:
  - Adequate testing environments
  - TSO
  - UNIX
  - DB2
  - Security
  - Tracing
- 8.3, “Configuration problems” on page 255 and 8.4, “Runtime problems” on page 266:
  - Configuration and Runtime
  - Security
  - DB2
  - Other

Problems can be categorized into Configuration problems, Deployment problems, Runtime problems (and No problems), but let us start with prevention.

8.1 Problem prevention

Avoiding problems depends on the skills of the people, the preparation of the material and resources, and the organization and documentation of the implementation plan:

- The importance of a well-organized and documented naming standard cannot be over-emphasized.
- A good naming standard, such as the one documented by the Washington Systems Center in Techdoc WP100653, does a good job of naming the components of a WebSphere Application Server Network Deployment for z/OS cell.

The Techdocs mentioned in this section are at the following Web page:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/

8.1.1 Good management practices

A smoothly running set of servers helps you to prevent and recover from problems. In this section, we provide good practices to help. Some particular recommendations worth highlighting are:

- Plan for sufficient hardware (Real memory, CPUs, DASD)
- Use symbolic links for mount points of product HFS libraries
- Controlling use of UNIX super-user privileges
- Controlling over-use of DB2 privileges (SYSADM, DBADM, GRANTs)

Additional good management practice resources are:

Managing servers For good advice about managing your servers in WebSphere Application Server for z/OS, see Techdoc article WP101138, *Hidden Gems: Little known features of WebSphere Application Server for z/OS*.

Managing messages Too many messages from WebSphere Application Server for z/OS servers fills up SYSLOG and makes it difficult to see the important messages there. To help with the routing of messages away from SYSLOG, see Techdoc TD10369.

Managing timeouts Your application servers must cope with a variety of work requests, and can occasionally experience time-out errors. See Techdoc article WP101233 *Configuration Options for Handling Application Dispatch Timeouts*.

8.1.2 Assembling required resources

Plan for adequate resources to ensure that your production topology can perform up to your expectations of performance and availability goals. See techdoc WP101620 for a discussion about *Basic Sizing for WebSphere Application Server z/OS V7*. This article does not include anything specific to WebSphere Process Server for z/OS or other BPM products.

System z processor requirements

Multiple LPARs with multiple CPUs, including general-purpose and zAAP processors, are required for most production topologies. Contact your IBM representative to have a capacity planning or sizing study run for your environment.

Real memory requirements

Do not try running WebSphere Process Server for z/OS on any system with less than 2 GB of real storage. Memory requirements depend on the number of servers, the size of the JVM heaps, and whether you are running with 64-bit heaps. Also, consider DB2 storage requirements.

Roughly speaking, allow 1 GB for system tasks and 1 GB for each server and deployment manager. Adjust upward for multiple servant regions and 64-bit heaps. Having a good auxiliary storage (paging) subsystem might also be required if you are pushing the limits of real storage.

Virtual storage requirements

Do not be constrained in your private region size by REGION size specifications or other limits, such as the IEALIMIT or IEFUSI exits.

DASD space requirements

Do not under-estimate the room that is required to hold ZFS and HFS file systems, backups of the same, DB2 database tables, auxiliary (paging) data sets, JES SPOOL, and so forth. In our testing, we were fortunate enough to have a substantial farm of 3390 (preferably mode 9 or larger) volumes.

After configuring WebSphere Process Server for z/OS Version 7 and WebSphere Business Services Fabric for z/OS, the configuration zFS file systems grew to about 1300 cylinders for each node.

8.1.3 Acquiring appropriate skills

Many skills are required to configure and troubleshoot these products:

- MVS systems programming
- UNIX, telnet, vi, scripting
- TSO, ISPF, SDSF
- Security (RACF, LDAP)
- Communications (TCP/IP)
- Java programming

Ensure that you have these skills on your team, or access to them when necessary.

8.2 Tools and techniques

In this section, we provide valuable techniques that we learned along with useful tools that you might find helpful.

8.2.1 Parallel testing cells

We configured multiple cells for our use, including the B7CELL, B8CELL, B9CELL, and even a K1CELL. The B8CELL and B9CELL were almost exact copies of the B7CELL, except for the cell prefix, which proved valuable in our implementation, testing, problem determination, and repair work.

Additionally, we also configured WebSphere Process Server for z/OS on a system outside of the ITSO to perform isolated problem determination.

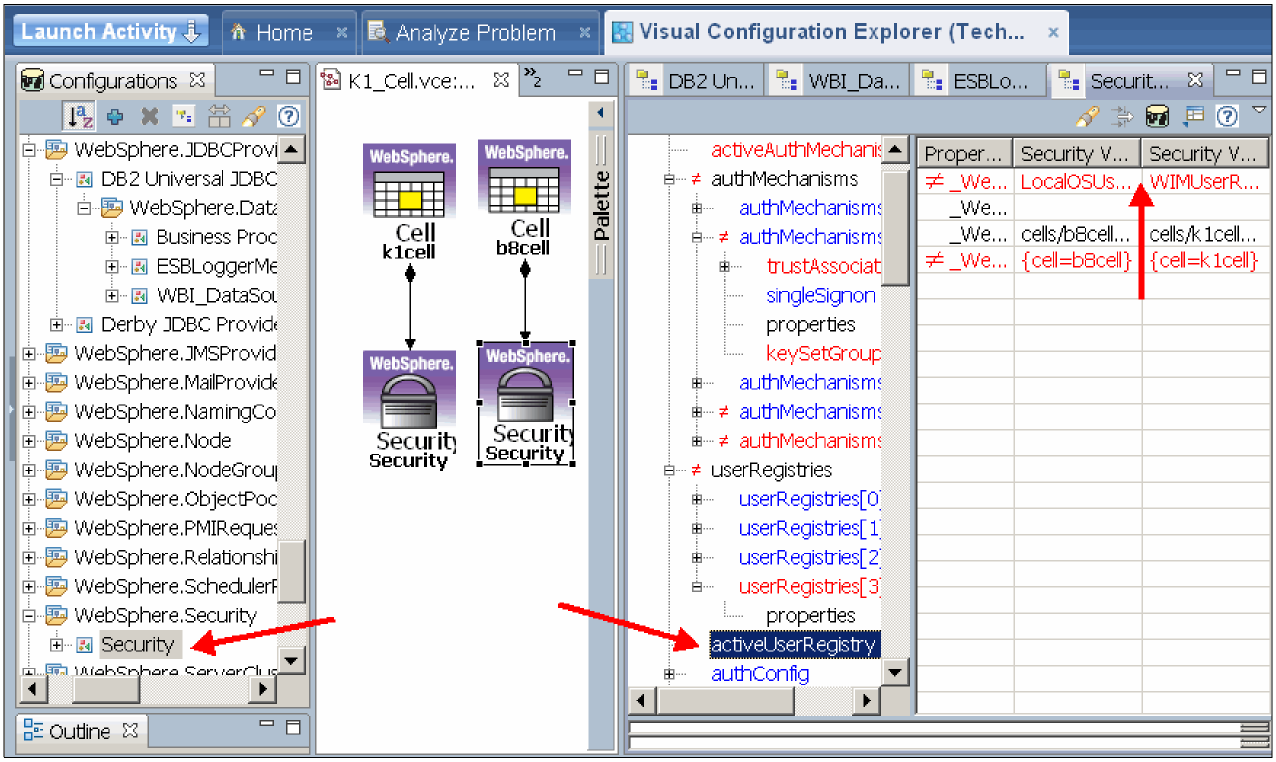
As team members configured WebSphere Application Server for z/OS, other team members used parallel cells for testing WebSphere Process Server for z/OS and WebSphere Business Services Fabric for z/OS, for example, if there were configuration errors when creating clusters, we verified various resource settings on data sources and messaging engines by comparing the cells.

Using the Virtual Configuration Explorer

The IBM Support Assistant (ISA) Visual Configuration Explorer (VCE) is an excellent tool to compare the configuration settings in our cell. For more information about the ISA and VCE, see Techdoc white paper WP101575.

To compare two configurations:

1. Select one of the objects, and right-click **Show Properties Node**.
2. In the left outline panel, select a comparable object, and drag it into the Comparison panel. Figure 8-1 shows the VCE visualization panel with Object Comparison of the Security panels.



Proper...	Security V...	Security V...
≠ _We...	LocalOSUs...	WIMUserR...
_We...		
_We...	cells/b8cell...	cells/k1cell...
≠ _We...	{cell=b8cell}	{cell=k1cell}

Figure 8-1 Comparing Security settings with the Visual Configuration Explorer

8.2.2 TSO techniques

As MVS systems programmers, this is the most common interface you can do your work with. Make sure that you master the efficiencies of TSO, ISPF, SDSF, and other products, such as RACF panels, OMVS, IShell, WLM, RMF™, and DB2ADM to assist you.

Using wide 3270 panels

When using TSO and SDSF, it is difficult to browse logs with an 80-character wide 3270 panel because you must scroll right and left to see complete messages. See Techdoc TD102151 for directions to set up your PCOM sessions to display a 132-column by 50 or 60-row format.

Displaying square brackets in your JES logs

Do you ever see strings in your logs that look similar to the following sample?

```
Ýjcc¨Ý50053¨Ý12311¨Ý3.53.81¨ T2zOS exception: Ýjcc¨ÝT2zos¨
```

Consequently, you logs are to look similar to this sample:

```
[jcc][50053][12311][3.53.81] T2zOS exception: [jcc][T2zos]
```

The display profile for your PCOM 3270 session can be specified to translate square brackets with the xlt file in Example 8-1.

Example 8-1 display.xlt file for translating square brackets

```
[Profile]
id=XLT
Description=Translation table to correctly display C square brackets []

[Option]
Replace=Y
ErrorMessage=Y

[SB Xlate]
AD=5B
BD=5D
```

Save this display.xlt file in the following location:

```
c:\Program Files\Personal Communications\private\display.xlt
```

Add the stanza in Example 8-2 to your PCOM .WS profile.

Example 8-2 Translation directive in your PCOM WS file

```
[Translation]
IBMDefaultView=0
DefaultView=c:\Program Files\Personal Communications\private\display.xlt
```

SDSF tricks

SDSF is the primary interface for z/OS operations in most installations. There are many panels and commands in SDSF that can make you more productive in operating and troubleshooting these products. See Technote TD100589, *Using SDSF for WebSphere for z/OS Operations*.

Generalizing RACF profiles

If you build a production topology with WebSphere Application Server Network Deployment for z/OS and BPM products, you can set up a generic set of RACF profiles to cover multiple servers and nodes in your cell from the beginning. See Techdoc WP101427, *Using Generic RACF Profiles with WebSphere on z/OS V7*.

8.2.3 Scripting

There are many useful scripts that can help diagnose and correct problems in the WebSphere Application Server for z/OS environment. We discuss a few of those scripts in this section.

Fixing permissions and ownership of files in configuring HFS

We ran the following scripts in the BBOWHFSB job when we configured the standalone server:

- bbowhfsb: Sets the permission bits
- bbowhfsc: Sets ownership attributes

There might be times when the permission and ownership attributes must be corrected after activities or mistakes. Example 8-3 is an example of running these scripts in a telnet session.

Example 8-3 Running the bbowhfsc and bbowhfsc scripts for deployment manager and nodeb

```
cd /usr/lpp/zWebSphere/V7R0/bin
./bbowhfsb.sh /wasv7config/b7cell/b7dmnode/DeploymentManager
./bbowhfsc.sh /wasv7config/b7cell/b7dmnode/DeploymentManager b7admin b7cfg
./bbowhfsb.sh /wasv7config/b7cell/b7nodea/AppServer
./bbowhfsc.sh /wasv7config/b7cell/b7nodea/AppServer b7admin b7cfg
```

The wsadmin scripting

Many scripts use the wsadmin.sh tool to automate configuration tasks. See the following techdocs articles for good examples:

- *WSADMIN Primer (with Jython)* - WP101014:
  - Includes several sample hands-on exercises
  - Replaces WSADMIN Primer (with JACL) - WP100421
- *Using Jython Scripting Language with WSADMIN* - WP100963:
  - Includes several sample scripts to do common administrative tasks

Additionally, many Information Center articles have good examples.

Updating the ports, virtual host aliases, and short names

Techdoc article TD104066 can help you to update or create new application servers in WebSphere Application Server for z/OS. This Techdoc article also includes sample jython scripts to automatically correct the port assignments to the WSC standard numbering scheme.

Running scripts in batch jobs

You have many choices of how to run your scripts besides telnet or OMVS. Example 8-4 is a sample job to run the zWPSConfig.sh script.

Example 8-4 Running a shell script in a batch job

```
//MYSCRJOB JOB (0),CLASS=A,MSGCLASS=A,REGION=0M,...
//*******************************************************/
//*  Run zxy.sh                                         */
//*******************************************************/
//INSTO EXEC PGM=IKJEFT01,REGION=0M,TIME=1440
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
//STDOUT DD *
//STDERR DD *
 BPXBATCH SH +
  cd /wasv7config/b7cell/b7dmnode/DeploymentManager/bin/; +
./zWPSConfig.sh
```

Running scripts without interruption in telnet sessions

You can use the nohup utility to ignore the SIGHUP signal, which might occur if you lose your connection with the host. Whether you use Windows® telnet client, TeraTerm, or PuTTy to execute scripts in the background without being interrupted or disconnected, see Example 8-5.

Note the position of quotes around the command being executed, and the ampersand (&) at the end to run the command in another thread so that you can disconnect or continue with other commands in your telnet session.

Example 8-5 Running shell scripts in the background to disconnect

```
nohup sh –c ’./wasv7config/b7cell/b7nodea/AppServer/bin/zWPSConfig.sh \
   -augment -response /u/user1/wpswork/b7Profile.rsp' \
   >/tmp/b7WPSConfig.out 2>&1
```

Tracing shell scripts

You can insert the following command in the beginning of shell scripts:

```
set -x
```

If you want to see the detailed results of a shell script that you are running from the command line (UNIX prompt), you can precede it with the sh -x string, as shown in Example 8-6.

Example 8-6 Invoking a shell script with tracing

```
sh –x ’./wasv7config/b7cell/b7nodea/AppServer/bin/zWPSConfig.sh \
   -augment -response /u/user1/wpswork/b7Profile.rsp' \
   >/tmp/b7WPSConfig.out 2>&1
```

8.2.4 UNIX System Services: Tools and techniques

You have many choices of tools to explore, such as logs, property, and other files in the HFS and ZFS file systems besides OMVS and IShell.

8.2.5 Using telnet, vi, and ISPF 3.17

The following profile and set up files in your home directory will make your life easier when operating in a telnet session.

Tailoring your .profile and .setup files

In your home directory, you can customize your UNIX environment to provide short-cuts and settings that will make you more productive when you are exploring and working in the UNIX file system environment.

Creating a .setup file

To help with typing in telnet sessions, we created alias names in the .setup files in the same home directories as the .profile files.

Example 8-7 on page 250 is a sample setup file that is run automatically when you log in using telnet. You might want to tailor it according to your preferences.

Example 8-7 Sample setup file

```
alias ll="ls -lat"
alias dirs="ls -alLD"
set -o vi
```

In Example 8-7, `ll` allows you to list the files' details with the newest ones at the top.

The syntax `set -o vi` helps you to retrieve commands from the access history. On your keyboard, press the escape key one time (esc), release it, and press `k`. The last command that you typed will appear on the window. Keys j and k help you to scroll back and forth in the history buffer.

8.2.6 UNIX System Services tools on z/OS

There are many tools available. Here are a few to help you.

Dealing with ASCII files

z/OS (and MVS before it) uses an EBCDIC code page, but the WebSphere Application Server for z/OS runtime uses an ASCII code page for a lot of its .xmp files, property files, and logs. Most TSO file editors, such as, ISPF do not deal well with ASCII files (until recently); however, in this section, we provide a few tools that do.

iconv

Convert a file from ASCII (`ISO8859-1`) to EBCDIC (`IBM-1047`) with the iconv command:

```
iconv -f ISO8859-1 -t IBM-1047 x.asc > x.ebc
```

viascii

viascii is a useful tool for editing ascii files with the 'vi' editor.

http://www.ibm.com/servers/eserver/zseries/zos/unix/bpxa1ty2.html

Other z/OS UNIX tools

For other useful tools, see the following web sites:

ASCII to EBCDIC conversion:

http://www.ibm.com/servers/eserver/zseries/zos/unix/bpxa1p03.html

Tools and Toys:

http://www.ibm.com/servers/eserver/zseries/zos/unix/bpxa1ty2.html

Write your own

There are samples in Techdoc PRS2520 of how to write your own UNIX System Services tools.

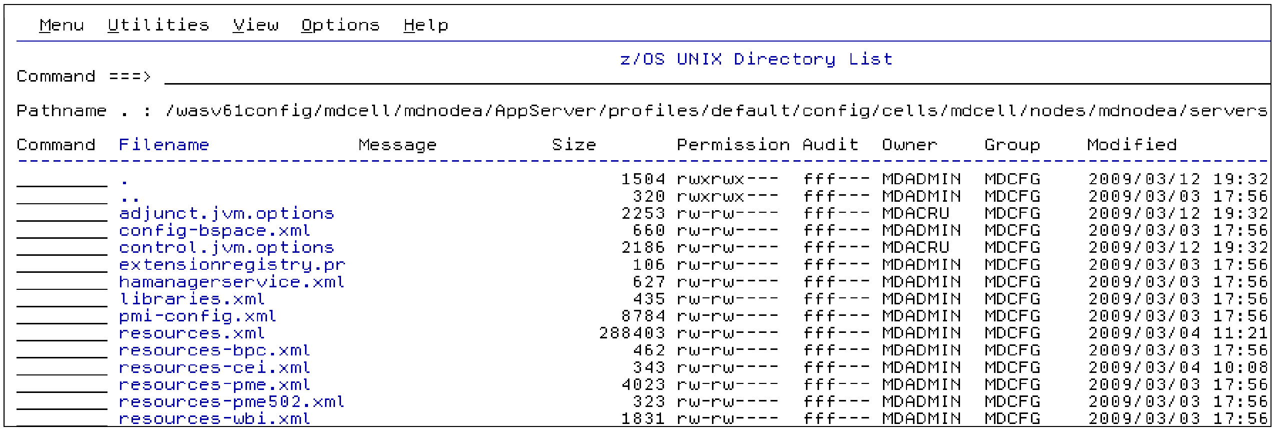
TSO/E tools that deal with ASCII files

You do not have to use telnet and viascii to browse or edit ASCII files. There are tools now available for TSO users.

ISPF 3.17: The z/OS UNIX Directory List Utility

This was first provided in z/OS 1.8 and gives you a full-panel column-arranged panel where you can browse and edit files in EBCDIC and in ASCII.

Use the Menu, Utilities, View, Options, and Help pull-down menus to learn about this useful panel on Figure 8-2.



```
  Menu  Utilities  View  Options  Help
                                                     z/OS UNIX Directory List
Command ===>
Pathname . : /wasv61config/mdcell/mdnodea/AppServer/profiles/default/config/cells/mdcell/nodes/mdnodea/servers

Command  Filename             Message          Size       Permission Audit  Owner    Group    Modified
--------------------------------------------------------------------------------------------------------------
________ .                                           1504 rwxrwx---  fff--- MDADMIN  MDCFG    2009/03/12 19:32
________ ..                                           320 rwxrwx---  fff--- MDADMIN  MDCFG    2009/03/03 17:56
________ adjunct.jvm.options                         2253 rw-rw----  fff--- MDACRU   MDCFG    2009/03/12 19:32
________ config-bspace.xml                            660 rw-rw----  fff--- MDADMIN  MDCFG    2009/03/03 17:56
________ control.jvm.options                         2186 rw-rw----  fff--- MDACRU   MDCFG    2009/03/12 19:32
________ extensionregistry.pr                         106 rw-rw----  fff--- MDADMIN  MDCFG    2009/03/03 17:56
________ hamanagerservice.xml                         627 rw-rw----  fff--- MDADMIN  MDCFG    2009/03/03 17:56
________ libraries.xml                                435 rw-rw----  fff--- MDADMIN  MDCFG    2009/03/03 17:56
________ pmi-config.xml                              8784 rw-rw----  fff--- MDADMIN  MDCFG    2009/03/03 17:56
________ resources.xml                             288403 rw-rw----  fff--- MDADMIN  MDCFG    2009/03/04 11:21
________ resources-bpc.xml                            462 rw-rw----  fff--- MDADMIN  MDCFG    2009/03/03 17:56
________ resources-cei.xml                            343 rw-rw----  fff--- MDADMIN  MDCFG    2009/03/04 10:08
________ resources-pme.xml                           4023 rw-rw----  fff--- MDADMIN  MDCFG    2009/03/03 17:56
________ resources-pme502.xml                         323 rw-rw----  fff--- MDADMIN  MDCFG    2009/03/03 17:56
________ resources-wbi.xml                           1831 rw-rw----  fff--- MDADMIN  MDCFG    2009/03/03 17:56
```

Figure 8-2 ISPF 3.17 z/OS UNIX Directory List panel

You can tailor these panels to adjust the column arrangement and width.

8.2.7 IBM Support Assistant (ISA)

The ISA workbench is a Launch Pad for many diagnostic tools, some of which were previously available as separate downloads from alphaWorks® or developerWorks®. Here is a list of the tools that are helpful when debugging problems in the WebSphere Process Server for z/OS environment:

Real-time analysis: Health Center (IBM Monitoring and Diagnostic Tools for Java)

- Garbage Collection analysis:
  - GC and Memory Visualizer (IBM Monitoring and Diagnostic Tools for Java - GCMV)
  - Pattern Modeling Tool for Java GC (PMAT)
- Dump analysis:
  - Memory Analyzer (IBM Monitoring and Diagnostic Tools for Java)
  - Dump Analyzer (IBM Monitoring and Diagnostic Tools for Java)
  - Memory Dump Diagnostic for Java (MDD4J)
  - HeapAnalyzer (New!)
  - Thread & Monitor Dump Analyzer for Java (TMDA)
- Trace analysis:
  - Trace and Request Analyzer for WAS
  - Log Analyzer
- Configuration analysis: Visual Configuration Explorer (VCE)

The installation of the ISA and its analysis tools is well documented in several web sites, such as the download site at:

http://www.ibm.com/software/support/isa/

The IBM Education Assistant (IEA) has several tutorials about installing and using the ISA at:

http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.selfassist/selfassist/ISAv41_Task.html

After you have the ISA installed on your workstation, it is easy to download specific diagnostic tools as add-ons. There are over 100 product add-ons and a growing list of tool add-ons.

For more information about the ISA, see Techdoc white paper WP101575.

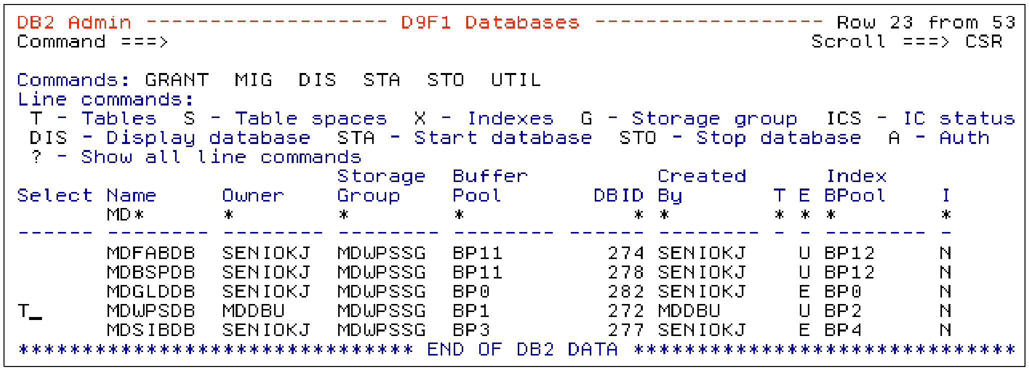
8.2.8 DB2 tools

If you do not have a DB2 administrator or expert on your team, it is good to know how to use the helpful tools that are available.

Using ADBL: The DB2 administrator's friend

DB2 Admin is an interface between ISPF and DB2. It can issue SQL calls and display any resulting rows in an ISPF table panel.

It is a panel-driven ISPF application that allows you to drill down into the DB2 catalog, databases, tables, table spaces, and other DB2 objects, for example, to see what tables are in a particular database, enter `1` (DB2 catalog) and `D` (databases), then enter the database name prefix at the top of the Name column, as shown in Figure 8-3.



```
DB2 Admin ------------------- D9F1 Databases ------------------ Row 23 from 53
Command ===>                                                  Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                         Storage  Buffer          Created      Index
Select Name     Owner    Group    Pool       DBID By       T E BPool    I
       MD*      *        *        *             * *        * * *        *
------ -------- -------- -------- -------- ------ -------- - - -------- -
       MDFABDB  SENIOKJ  MDWPSSG  BP11          274 SENIOKJ    U BP12     N
       MDBSPDB  SENIOKJ  MDWPSSG  BP11          278 SENIOKJ    U BP12     N
       MDGLDDB  SENIOKJ  MDWPSSG  BP0           282 SENIOKJ    E BP0      N
T_     MDWPSDB  MDDBU    MDWPSSG  BP1           272 MDDBU      U BP2      N
       MDSIBDB  SENIOKJ  MDWPSSG  BP3           277 SENIOKJ    E BP4      N
******************************* END OF DB2 DATA *******************************
```

Figure 8-3 DB2 Administrative tool showing databases for the B7 cell

You can then enter `T` next to one of the database names to display the tables in the database. You can use the sort command to sort various columns, and drill down deeper to see the columns and comments of the tables.

DB2 tricks

The following techniques can help you manage the DB2 administration tasks.

DASD space used by your table spaces

DB2 Table spaces are backed by VSAM datasets whose names have the following format:

```
Vcatname.DSNDBD.dbname.tsname.y0001.znnn
```

In the data set:

- Vcatname: The integrated catalog name (high level index), or VCAT
- DSNDBD: The data component of the VSAM cluster
- dbname: The database name
- tsname: The table space name or index name

Example 8-8 on page 253 displays the DB2 table spaces, sorted with the largest ones on the top.

Example 8-8 Displaying the DB2 table spaces sorted with the largest ones on top

```
Command ===> sort tracks                          TRACKS
   -------------------------------------------------
   B7DB2VC.DSNDBD.B7ESBDB.ESBCLOB.I0001.A001       150
   B7DB2VC.DSNDBD.B7WPSDB.BBLOB2TS.I0001.A001      150
   B7DB2VC.DSNDBD.B7WPSDB.TICKETTS.I0001.A001      150
   B7DB2VC.DSNDBD.B7BPEDB.SCHE1Y3Y.I0001.A001       15
   B7DB2VC.DSNDBD.B7ESBDB.ESBTS.I0001.A001          15
   B7DB2VC.DSNDBD.B7EVT.BLOB00RV.I0001.A001          6
```

Using the DbDesignGenerator

This new script is useful for creating and updating the database design document that is used to created sample DDL statements for creating the DB2 objects for WebSphere Process Server for z/OS runtime. (Also called the database design tool - DDT.) For more particulars, see “Debugging DbDesignGenerator.sh” on page 124.

Using the Ddl2Pds.sh script

This message usually occurs because there is an ISPF session browsing/editing the PDS. The copy command needs exclusive access to the PDS to work:

```
cp: FSUM6258 cannot open file "//'WASCFG.B8CELL.ALLDDL(B8BPR1)'": EDC5061I An
error occurred when attempting to define a file to the system."
```

You must run Ddl2Pds.sh one time against each directory to get them all. Specifying ALL is not a problem because currently each directory only has the one file in it.

For more particulars, see “Debugging DbDesignGenerator.sh” on page 124.

Using the createDB.sh script

The createDB.sh script is provided as a sample script to help automate the generation and execution of the sample .sql statements that create the DB2 objects (Tables, etc.).

See the sections 4.2.2, “Generating the SQL with createDB.sh” on page 106 and “Debugging DbDesignGenerator.sh” on page 124 for helpful hints.

Using DSNTEP2: Executing DDL without errors

Many times during the initial attempts at building your DB2 databases and creating the tables, you might be forced to drop the databases, storage groups, table spaces, or tables. When you re-create them using a large generated set of SQL or DDL statements, there might be duplicate objects, such as functions or sequences that were not dropped. Usually when DSNTEP2 encounters ten SQLCODE=-601 errors, it flushes the remaining SQL statements.

If you use DSNTEP2, you can add the statement at the beginning of your DDL statements, as in the following example, where `n` is the number of errors allowed before the program can continue. Use ‘`-1`’ to indicate that the program must tolerate an unlimited number of errors. Severe SQL errors cause program termination whenever encountered:

```
--#SET MAXERRORS n
```

8.2.9 Tracing

When you cannot determine the cause of a problem, turn on tracing to get more information. However, we recommend that you do not enable tracing until you need to. You get plenty of information in the logs by default.

zWPSInstall tracing

Add the following string to your zWPSInstall.sh parameters:

```
    -trace *=all=enabled
```

The trace logs go to the `{app_server_root}/logs/` directory.

zWPSConfig.sh tracing

Add the same string to the zWPSConfig.sh command:

```
   -trace *=all=enabled
```

These trace logs go to the `{profile_root}/logs/` directory.

We recommend that you do not enable tracing until you need to. You get plenty of information in the logs by default.

WebSphere Java runtime tracing

The MVS modify command ('f' for short) is extremely powerful and dynamic:

```
   F <server_name>,tracejava='com.ibm.xxx.*= all=enabled'
```

Here is an example to start BPE tracing dynamically:

```
   F server-name,TRACEJAVA='com.ibm.bpe.*=all=enabled'
```

To turn off tracing, issue:

```
   F server-name,TRACEINIT
```

wsadmin tracing

Edit the {profile_root}/properties/wsadmin.properties file, by uncommenting the line, as shown in Example 8-9.

```
   #com.ibm.ws.scripting.traceString=com.ibm.*=all=enabled
```

Example 8-9   Portion of the {profile_root}/properties/wsadmin.properties file

```
#-------------------------------------------------------------------
# The traceFile property determines where trace and logging
# output are directed.  If more than one user will be using
# wsadmin simultaneously, different traceFile properties should
# be set in user properties files.
# The default is that all tracing and logging go to the console;
# it is recommended that a value be specified here.
# If the file name contains DBCS characters, use unicode format such as
#-----------------------------------------------------------------
com.ibm.ws.scripting.traceFile=/wasv7config/b7cell/b7nodea/AppServer/profiles/defa
ult/logs/wsadmin.traceout
#------------------------------------------------------------------
# The traceString property governs the trace in effect for
# the scripting client process.
# The default is no tracing.
#------------------------------------------------------------------
com.ibm.ws.scripting.traceString=com.ibm.*=all=enabled
```

Run the script, and look in the location specified by the property:

```
   scripting.traceFile={profile_root}/logs/wsadmin.traceout
```

JDBC tracing

Enable JDBC tracing by specifying a traceFile in the DB2JccConfiguration properties file, which is pointed to from the servant Java options.

In the Integrated Solutions Console, navigate to **Application servers** → **b7sr01a_WPS Server Instance** → **Process Definition** → **Servant Java Virtual Machine** → **Custom Properties**, and enter a Name of `db2.jcc.propertiesFile` with a value pointing to the location of your properties files, such as:

```
/wasv7config/b7cell/b7dmnode/DB2JccConfiguration.properties
```

Example 8-10 is a typical file that shows the DB2 subsystem ID and the parameter to turn on JDBC tracing.

Example 8-10   DB2JccConfiguration.properties file

```
db2.jcc.ssid=D9FG
db2.jcc.override.traceFile=/tmp/B7_tracejdbc.txt
```

For detailed guidance about reading the traces, see *DB2 Universal Database for z/OS Application Programming Guide and Reference FOR JAVA Version 8* SC18-7414, which is available at the following Web page:

http://publibfi.boulder.ibm.com/epubs/pdf/dsnjvj11.pdf

8.3 Configuration problems

Most of the problems that we encountered were during the configuration of the various products:

- WebSphere Application Server for z/OS
- WebSphere Process Server for z/OS
- WebSphere Business Services Fabric for z/OS

The following sections are summarized approaches for troubleshooting what we use.

8.3.1 Methodologies for determining configuration problems

The general approach to determining configuration problems depends on the phase of configuration:

- Running configuration set up jobs or scripts
- Creating DB2 databases and tables
- Starting up servers
- Testing the runtime with sample applications

Running scripts

Before you run the scripts, make certain that you prepared all of the prerequisite material and that you know how to run it:

1. Verify the correct syntax.
2. Verify proper user identities (and passwords).
3. Verify any environmental settings, such as STEPLIBs, $WAS_HOME variables.

After you run the scripts perform the following tasks:

1. Check the return codes.

2. Browse the output.
3. Examine the logs.

If you encounter errors, perform the following tasks:

1. Desk-check the previously mentioned preparatory steps.
2. Read the output messages and logs.
3. Research any error messages using the Integrated Solutions Console, IBMLink, and Google.
4. Turn on tracing.
5. Call for help.

Creating DB2 databases and tables

Before you start, make certain that you have good database naming conventions for your DB2 objects.

Verifying DB2 parameter settings

Here is a check-list for DB2 environmental settings, and the values that we used as examples:

STEPLIBs	DB9F9.SDSNEXIT, DB9F9.SDSNLOAD, DB9F9.SDSNLOD2
CLASSPATH	/usr/lpp/db2/d9fg/db2910_jdbc/classes/
LIBPATH	/usr/lpp/db2/d9fg/db2910_jdbc/lib/
DB2 Location Name	DB9F
DB2 JCC Properties	/etc/d9fg/DB2JccConfiguration.properties
AuthIDs:	B7DBU (verify password correct in data source and in RACF)
	Servant and Adjunct UserID - xxASRU
GRANT DBADM to	Database, StoGroup, Schema, BufferPools
GRANT SELECT to	SYSIBM.SYSTABLES
currentSchema	In data source custom properties for non-ME data sources
Data store Schema	In messaging engine data stores

Before you generate the SQL or DDL statements, understand the different ways that you can create them:

- Using createDB.sh scripts, DSNTEP2, or others, such as sibDDLGenerator.sh.
- Using single or multiple databases, schema prefixes, or SQLIDs.
- Using GRANT privileges at the database (DBADM) or individual Table level.

Before you execute the SQL or DDL statements, understand the different ways that you can run them:

- Use createDB.sh scripts, or convert them to EBCDIC, and copy them to z/OS files and run with DSNTEP2.
- Verify proper user identities and GRANTs.
- Verify any environmental settings, such as STEPLIBs, $WAS_HOME variables.

After you run the scripts perform the following tasks:

1. Check the return code: Use the following command to look for negative SQLCodes:

```
find ‘code = -’
```

2. Use the DB2 Administrative panels to check the number of tables and table spaces. See "Using ADBL: The DB2 administrator's friend" on page 252.

If you encounter errors, perform the following tasks:

1. Desk-check the previously mentioned preparatory steps.
2. Read the output messages and logs, including SYSLOG.
3. Check to see if you have enough room on your DASD volumes for the table spaces. See "DASD space used by your table spaces" on page 252.
4. See 8.4.1, "Solving DB2 runtime problems" on page 268.
5. If necessary, research any error messages using the Integrated Solutions Console, IBMLink, and Google.
6. If still necessary, turn on tracing.
7. Compare the results with any other testing cell that you might have.
8. Call your DBA.

Starting up servers

Before you start the servers, make certain that you prepare all of the executables:

1. Know the exact syntax of the `Start` command.
2. Verify proper user identities (and passwords).
3. Verify any environmental settings, such as STEPLIBs.

After the servers are started:

1. Check the return codes and make sure that they stay up.
2. Browse the JES logs for the controller, adjunct, and servant started tasks:
   - Issue the FIND SEVERE ALL command in SDSF
   - Look for warnings with `find warning all`
   - Look for Java exceptions: `find exception: all`
3. If there are ffdc logs, you can examine them, but our experience is that they seldom hold more useful information than the JES logs (SYSOUT, SYSPRINT, and the JES JOBLOG.).
4. Take a look at SYSLOG. Some error messages are easier to find there.

If you encounter errors, perform the following tasks:

1. Desk-check the previously mentioned preparatory steps.
2. Study the output messages and logs.
3. If necessary, research any error messages using the Integrated Solutions Console, IBMLink, and Google.
4. If still necessary, turn on tracing.
5. Call for help.

Testing the runtime with sample applications

You must deploy the application in the runtime, and start it. If you have problems, follow these steps:

1. Before you run the test, make sure the servers are started, and check the logs for any SEVERE error messages.
2. Run your tests and perform the following tasks:
   a. Check the return messages in your browser.

b. Browse the JES logs for the servant started task:

   - See what messages occurred since you ran the test.
   - Issue the FIND SEVERE ALL command in SDSF.
   - Look for warnings with `find warning all`.
   - Look for Java exceptions: `find exception: all`.

c. Take a quick look at SYSLOG because error messages are easier to find there.

3. If you encounter errors, perform the following tasks:

   a. Desk-check the previously mentioned preparatory steps.

   b. Study the output messages and logs.

   c. If there are ffdc logs, you can examine them, but our experience is that they seldom hold more useful information than the JES logs (SYSOUT, SYSPRINT, and JES JOBLOG).

   d. If necessary, research any error messages using the Integrated Solutions Console, IBMLink, and Google.

   e. If still necessary, turn on tracing.

   f. Call for help.

8.3.2 Common problems

There are a few problems that we encountered, besides the usual typing, spelling, and not following directions. We list the problems in this section.

Configuration

Several tasks during configuration seem to be frequent stumbling blocks:

- Reading, following directions, spelling, typing, and clicking
- STEPLIBs for WebSphere Application Server for z/OS and DB2
- GRANTs and DB2 privileges
- EJBROLEs
- Symbolic Links
- UNIX file permissions and ownership
- SQLID versus Schema
- DASD space
- Virtual and real storage
- Known APARs

The next sections contain detailed problems that we encountered along with the solutions that we used.

Spreadsheet and zPMT errors

We used the Excel spreadsheet to create response files for the zPMT tool when configuring the servers. Here are manual corrections we had to make in the zPMT:

- Administrator’s user ID: For the job to federate an empty node, this showed up as WSADMIN in the zPMT and the generated job stream instead of the correct administrator’s user ID (B7ADMIN in our case).

- Symbolic Link for WebSphere Application Server for z/OS smpe root: The zPMT in WCT has three variables that have to do with the definition of intermediate symbolic link and the path to the WebSphere install directory:

  – zEnableIntermediateSymlink: True or false.

- zIntermediateSymlink: The intermediate symbolic link name to be used.
- zSmpePath: The real path to SBBOHFS no matter the setting of zEnableIntermediateSymlink.

The default in the WCT zPMT is to define an intermediate symbolic link.

Configuring WebSphere Application Server for z/OS

The following sections detail errors that we encountered while building the servers.

BBODBRAK

This job, BBODBRAK, runs the RACF commands for the deployment manager's certificates, as shown in Example 8-11.

Example 8-11 Errors in the BBODBRAK job

```
Generating certificates for WebSphere Deployment Manager
   228 *-* "RACDCERT ID (B7ACRU) GENCERT SUBJECTSDN(CN('wpsplex.itso.ibm.com') O"
,"('IBM') OU('B7CELL'))
  ...  NOTAFTER(DATE(2010/12/31))"
  The certificate that you are creating has an incorrect date range.
  The certificate is added with NOTRUST status.
```

We fixed the problem by changing the RACDCERT ID (B7ACRU) GENCERT command to use an earlier date: NOTAFTER(DATE(2010/11/29))

Be careful when you generate new certificates and sign them with CA certificates that were created in the past. To avoid problems, do not use the same expiration date for your new certificate as the one specified in your CA certificate. Subtract a day from the expiration date of your CA certificate, and use that date for the expiration date of your new certificate. Otherwise, the new certificate's expiration date and time might be later than the CA certificate that signed it, which leads to your new certificate having the NOTRUST status in RACF, which leads to this message in WebSphere:

```
ExtendedMessage: BBOO0220E: SSLC0008E: Unable to initialize SSL connection.
Unauthorized access was denied or security settings have expired.  Exception is
javax.net.ssl.SSLHandshakeException: Client requested protocol Unknown 0.2 not
enabled or not supported.
```

See Techdoc paper PRS3584 *Renewing expiring RACF certificates for WebSphere on z/OS* for more information and alternate solutions.

BBOWWPFA job fails

The BBOWWPFA job sets up the profiles in the runtime file system. We received the following errors:

```
INSTCONFFAILED: The profile could not be created. For more information, consult the
/wasv7config/b7cell/b7nodea/AppServer/logs/manageprofiles.
```

Missing STEPLIBs

STEPLIBs are required in the servant's procedures for the application servers and for the deployment manager's servant if you do not have these libraries in link list or LPA.

They are also required in the setupCmdLine.sh script in the {profile_root}/bin/ directory for each node and for the deployment manager.

We added STEPLIBs for the following libraries:

- WebSphere runtime libraries
- DB2 for z/OS runtime libraries
- MQ for z/OS runtime libraries

Federation errors

Here are a few errors that we encountered when we tried to federate an empty node:

- NodeAgent does not start

  This error might happen if the BBOWMNAN job was run earlier (unsuccessfully) with the incorrect user ID. When re-run with the correct (administrator's) ID, the federation succeeded, but the node agent failed to start because of bad permissions and ownerships in the configuration HFS.

  To fix this error, run the commands shown in Example 8-12.

Example 8-12 Running the bbowhfsb.sh and bbowhfsc.sh scripts

```
cd /usr/lpp/zWebSphere/V6R1/bin
./bbowhfsb.sh /wasv7config/b7cell/b7dmnode/DeploymentManager
./bbowhfsc.sh /wasv7config/b7cell/b7dmnode/DeploymentManager b7admin b7cfg
./bbowhfsb.sh /wasv7config/b7cell/b7nodea/AppServer
./bbowhfsc.sh /wasv7config/b7cell/b7nodea/AppServer b7admin b7cfg
```

- Nodes do not appear the in Integrated Solutions Console

  After federation, sometimes the nodes will not appear in the Environment Nodes pane of the Integrated Solutions Console.

  A common solution is to log out of the Integrated Solutions Console, and log back in. Sometimes recycling the deployment manager is required (Stop and restart it).

Clustering servers

When an unclustered server is added to a cluster, several resources are re-defined from the server scope to the cluster scope. In many cases, such as the messaging engines for the CEI bus, the data stores are re-defined with default values.

Running the Deployment Environment wizard

Make sure to back up the ZFS file systems before you try these steps because you might not be able to recover.

Recovering from the DE wizard

The ISC might not totally discard all of the changes when you get the choice. When we tried to re-run the DE wizard, it threw the errors in Example 8-13.

Example 8-13 Deployment Environment Wizard problems

```
CWLDB9012E: Creating cluster member b7sr1_WPS.b7nodea.0 failed.
Reason: Error: The server b7sr1_WPS.b7nodea.0 on node b7nodea already exists
before topology role AppTarget is even configured..
CWLDB9016E: The generation for deployment environment b7sr1_WPS failed. Reason:
CWLDB9012E: Creating cluster member b7sr1_WPS.b7nodea.0 failed. Reason: Error: The
server b7sr1_WPS.b7nodea.0 on node b7nodea already exists before topology role
AppTarget is even configured... If in doubt, discard the changes and do not save
the failed configuration to the master repository.
```

The solution was to restore the B7Cell from a backup of the zFS files at the state right after Federation, and then re-try the DE wizard.

Cleaning up the new cluster

To clean up the new cluster:

1. Examine the data sources in the custom properties. For us, this meant:
   - Removing the currentSQLID property
   - Adding currentSchema for the BPC and BPR_Observer data sources
2. Change the port numbers to conform with your numbering scheme. We used the updNewServerv7.py script.

Running updNewServerv7.py

If you misspell your server's long name, the jython script does not give you a good error message; instead, it just exits with a return code of 0, and none of the ports are changed.

To fix this, the updated copy of updNewServerv7.py was enhanced to display an error if the server does not exist.

8.3.3 Configuring WebSphere Process Server for z/OS

Most of the problems that we encountered while augmenting the various nodes (standalone server, empty nodes, deployment manager) occurred because of spelling and typing errors in the response file parameters. It is important to double-check your response files because it is hard to diagnose errors that are due to the parameter values in the response files not matching the values used in WebSphere Application Server for z/OS and DB2. It is much easier to desk-check your parameters than to figure out the mistake from the symptoms in many cases.

zWPSInstall.sh problem

When running the zSMPEInstall.sh for the second node, it failed with a RC=12 and the messages shown in Example 8-14.

Example 8-14 zSMPEInstall.sh failed RC=12 error message

```
BPXBATCH SH cd /wasv7config/b7cell/b7nodeb/AppServer/bin; export PATH=.:$PATH;
/wasv7config/b7cell/b7nodeb_wps_smpe/zos.config/bin/zWPSIzWPSInstall.sh
'-smproot /wasv7config/b7cell/b7nodeb_wps_smpe' '-runtime
/wasv7config/b7cell/b7nodeb/AppServer'
'-install'
. . .
ERROR: Unable to read file PATH. Error: 81 5620062
READY
```

The problem was caused by having an invalid UNIX environment for the user ID executing the job. The user ID (B7ADMIN) did not have a home directory on the secondary LPAR because the BBOSBRAM job, which creates the home directory for B7ADMIN, was only run on the primary LPAR.

Ensure that the user ID that is running zSMPEInstall.sh has a valid UNIX environment by logging on to a telnet client using the user ID.

Alternatively, use an OMVS session, and try to **`su`** to that user ID. Echo $PATH, $CLASSPATH, and so forth, and try commands, such as **`java -fullversion`**, to ensure that the user ID is working normally in UNIX on the LPAR where you are running the job.

zWPSInstall.sh log files

If any errors occur while running the script, start by looking in: `<server_root>/logs/wbi/install/installconfig.log`

When you find the error, if you issue `FIND ‘buildfile’ PREVIOUS`, it will tell you the ant file that failed, which directs you to one of the following logs to troubleshoot the problem. It is helpful to sort the logs by date to look at the most recent log file first by using the **`ls -t`** UNIX command. In our configuration, <server_root>, is as shown in Example 8-15.

Example 8-15 Log files created by the zWPSInstall.sh script

```
/wasv7config/b7cell/b7nodea/AppServer/logs/wbi > ls -1t

zWPSInstall.log
zWPSInstall.trace
100SCleanOSGICache.ant.log
94SDeployCoreAdminConsolePlugins.ant.log
93SDeployServerAdminConsolePlugins.ant.log
93SDeployWBICommonAdminConsolePlugins.ant.log
93SDeployBPCAdminConsolePlugins.ant.log
85SConfigNoProfileFirstStepsWBI.ant.log
90SCleanDeployTool.ant.log
90SConfigureWSProfileForAugmentation.ant.log
90SConfigureWSProfileForWBI.ant.log
91SConfigNoProfileFirstStepsCharset.ant.log
80SCopyInstallValidatorLog.ant.log
```

zWPSConfig.sh problems

The zWPSConfig.sh problems are:

- BBOWWPFD BBOWWPFM RC=512
- INSTCONFPARTIALSUCCESS

zWPSConfig RC=2048 with OutOfMemoryError in first.ant.log

In our sysplex, the BBOWWPFM job ran good in one LPAR but failed with RC=512 and INSTCONFPARTIALSUCCESS in the other. The joblog contained a message indicating that the location of the log file to consult:

```
INSTCONFPARTIALSUCCESS: The profile now exists, but errors occurred.
For more information, consult
/wasv7config/k1cell/k1node1a/AppServer/logs/manageprofiles/default_create.log.
```

In the `default_create.log` the error in Example 8-16 was displayed.

Example 8-16 Default_create.log error

```
<record>
    <date>Feb 6, 2010 4:14:31 AM</date>
    <millis>1265447671529</millis>
    <sequence>2714</sequence>
<logger>com.ibm.ws.install.configmanager.actionengine.ant.utils.ANTLogToCmtLogAdap
ter</logger>
    <level>WARNING</level>
<class>com.ibm.ws.install.configmanager.actionengine.ant.utils.ANTLogToCmtLogAdapt
er</class>
    <method>messageLogged</method>
    <thread>0</thread>
    <message>JVMJ9VM015W Initialization error for library j9jit24(11): cannot
initialize JIT
Could not create the Java virtual machine.</message>
. .
<message>Target stopped for: zCreateUUIDs - FAILURE</message>
. .
<message>Returning with return code:
INSTCONFPARTIALSUCCESS</message>
```

The zWPSConfig.sh failed with a RC=2048. In the log file at /wasv7config/k1cell/k1node1a/AppServer/logs/manageprofiles/default_create.log there was no clear cause, but in the ../manageprofiles/default directory the first.ant.log showed a java.langOutOfMemoryError. There was no problem with REGION size, MAXASSIZE, or region size limitations in IEFUSI.

Both of these problems were caused because the .profile of the user submitting the job contained _BPX_SHAREAS=YES. See the following Technote for similar problems with WebSphere migration jobs:

http://www.ibm.com/support/docview.wss&uid=swg21377885

Make sure that the .profile of the user running the job has _BPX_SHAREAS=NO, or comment out _BPX_SHAREAS=YES.

The JIT j9jit24 is not initializing and the suffix 24 suggests that the job might be picking up an old level of the Java SDK. In fact, the /etc/profile on this system was setting JAVA_HOME to the J1.4 Java SDK and the PATH contained the Java 1.4 /bin.

Upgrade the level of Java in the /etc/profile to J1.6 or create a .profile for the user ID that is submitting the BBO* installation jobs, which establishes Java 1.6 as the default for that user ID.

zWPSConfig.sh log files

If any errors occur while running zWPSConfig.sh script:

1. Start by looking in the <server_root>/logs/wbi/zWPSConfig.trace file for a high-level view.
2. Look in the <server_root>/logs/manageprofiles/default_augment.log file. When you find the error, if you issue **FIND ‘buildfile’ PREVIOUS**, it will tell you the ant file that failed, which will direct you to one of the following logs in the <server_root>/logs/manageprofiles/default/ directory, as shown in Example 8-17.

Example 8-17 Log files created by zWPSConfig.sh (partial view)

```
/wasv7config/b8cell/b8nodea/AppServer/logs/manageprofiles/default: >ls -1t

aboutThisProfileManaged_wbiserver.ant.log
addNodeToTopology.ant.log
configFirstStepsWBIManaged.ant.log
federateLaterProcServer_executeManagedProfileSetup.ant.log
wsadminListener.log
createProfileShortCut2StartMenuManagedWBIServer.ant.log
updateNodeMetaData.ant.log
configCopyESB.ant.log
loadDBProperties.ant.log
configCopy.ant.log
```

```
first.ant.log
saveParamsWbiServer.ant.log
bpcaugment.log
bpcaugment.wsadmin.log
aboutThisProfileManaged_wbicore.ant.log
. . .
```

8.3.4 DB2 problems configuring WebSphere Process Server for z/OS

In this section, we provide problems with DB2.

Failure in loading T2 native library db2jcct2zos

Failure in loading T2 native library db2jcct2zos when initializing the servant or adjunct. This error can have several causes:

- Cause 1: The SDSNLOAD and SDSNLOD2 libraries are not present on the STEPLIB of the address space reporting the error. To fix, add the SDSNLOAD and SDSNLOD2 to the STEPLIB.
- Cause 2: The LIBPATH was not accurately defined to the JDBC libraries. To fix, export the correct LIBPATH variable setting.
- Cause 3: You are trying to use the Test Connection function on a Type 2 data source using the Integrated Solutions Console and the data source is defined at cluster level. Typically the error also includes the text shown in Example 8-18.

Example 8-18 Error text for second cause of failure in loading T2 native library db2jcct2zos

```
reason: java.lang.UnsatisfiedLinkError: db2jcct2zos (Not found in
java.library.path)DSRA0010E: SQL State = null, Error Code = -99,999
```

There is a limitation in WebSphere Application Server for z/OS that Test Connection will not work for data sources defined at node or cluster level. You will find that you can test the connection of the jdbc/WPSDB data source that is defined at cell level but not the other data sources defined at cluster level.

- Cause 4: DB2 subsystem is not started. To fix, start DB2.

8.3.5 Solving DB2 configuration problems

Review the description of tools in Chapter 2, “Planning for and preparing WebSphere Process Server” on page 25, depending on what you are using.

B37 ABEND codes

B37 ABENDs in MVS almost always mean Out of Space, either on the volume (not likely with our generous DASD volumes) or out of extents - see the specific reasons for IEC030I B37-04.

Debugging common SQLCODES

In this section, we provide the more common SQL error codes and how we fixed them.

-454: CREATE FUNCTIONs

This error occurs if you drop the BPC database and try re-creating the tables because the five functions that are being defined in the BPC.sql file did not get dropped when the database was dropped.

You can ignore these messages if the functions that are being defined are identical, or you can explicitly drop the functions when you drop the database.

-454: CREATE SEQUENCEs

This error occurs if you drop the WPS common database and try re-creating the tables because the 14 sequences that are being defined in the WPS.sql file did not get dropped when the database was dropped.

You can ignore these messages if the sequences that are being defined are identical, or you can explicitly drop the sequences when you drop the database.

-922: Connecting to DB2

The problem is that the SQLException in Example 8-19 is issued when connecting to DB2.

Example 8-19 SQLException issued when connecting to DB2

```
java.sql.SQLException:
[IBM/DB2][T2zos/3.6.67]T2zosReusableConnection.flowConnect:execConnect:
DB2 engine SQL error, SQLCODE = -922, SQLSTATE = 42505, error tokens =
CONNECT;00D31024DSRA0010E: SQL State = 42505, Error Code = -922
```

The most common reason for this error is specifying an incorrect Location name in the Database name field of a Type 2 data source definition. This error is likely to happen because several of the wizards will set the database name in the Database name field. For a Type 2 data source, you must set the DB2 Location name in the Database name field.

To fix this error, make sure that the Database name property on the data source is set to the DB2 z/OS Location name rather than the name of the DB2 database.

Dropping functions, sequences, and table spaces

Here are sample drop statements for functions and sequences:

```
DROP FUNCTION B7WPS.OBSVR_JAR_ACTIVENT ();
DROP SEQUENCE SEQ_W_OBJ_LIT_DATE_ID;
```

The createDB.sh script will drop the functions and sequences when the databases are dropped and created with the -DBCreatre=true option. Example 8-20 provides an example of the statements.

Example 8-20 The createDB.sh drop statements

```
   DROP FUNCTION B7CELL.INTERVALIN (INTERVAL INT,TS_START TIMESTAMP,TS_END
   TIMESTAMP);
   DROP FUNCTION B7CELL.INTERVALIN (INTERVAL INT,TS_START TIMESTAMP,VC_END
   VARCHAR(26));
   DROP FUNCTION B7CELL.INTERVALIN (INTERVAL INT,VC_START VARCHAR(26),TS_END
   TIMESTAMP);
   DROP FUNCTION B7CELL.INTERVALIN (INTERVAL INT,VC_START VARCHAR(26),VC_END
   VARCHAR(26));
   DROP FUNCTION B7CELL.OBSVR_JAR_ACTIVE ();

   DROP SEQUENCE B7CELL.SEQ_W_LIT_DOUBLE_ID;
   DROP SEQUENCE B7CELL.SEQ_W_LIT_FLOAT_ID;
   DROP SEQUENCE B7CELL.SEQ_W_LIT_LONG_ID;
   DROP SEQUENCE B7CELL.SEQ_W_NAMESPACE_ID;
   DROP SEQUENCE B7CELL.SEQ_W_OBJ_LIT_ANY_ID;
   DROP SEQUENCE B7CELL.SEQ_W_OBJ_LIT_DATE_ID;
```

```
DROP SEQUENCE B7CELL.SEQ_W_OBJ_LIT_DATETIME_ID;
DROP SEQUENCE B7CELL.SEQ_W_OBJ_LIT_STRING_ID;
DROP SEQUENCE B7CELL.SEQ_W_STATEMENT_ID;
DROP SEQUENCE B7CELL.SEQ_W_URI_ID;
DROP SEQUENCE B7CELL.SEQ_W_VERSION_ID;
```

The createDB.sh script problems

We encountered several errors while running the createDB.sh script, and they were almost always caused by the sequence of running the script or user errors. See “Debugging createDB.sh” on page 124.

Specifying the DB2JccConfiguration.properties file location

The following parameter must specify the directory where the DB2JccConfiguration.properties file is located, but not the file itself. If you specify the complete file name, such as DBJDBCProperties=/u/user1/wpswork/DB2JccConfiguration.properties, you will get the error in Example 8-21.

Example 8-21 Error opening DB2JccConfiguration.properties file

```
[sql] java.util.zip.ZipException: Error opening zip file
/u/user1/wpswork/DB2JccConfiguration.properties
```

If you code DBJDBCProperties=/u/user1/wpswork, you will get no errors.

Saving output from createDB.sh

When diagnosing problems with createDB.sh, you can pipe the output to a file so that all of the messages that get written to the window are not lost. Use the following format:

```
createDB.sh -All +SIBDB > traceDb.out 2> traceDb.err
```

Also, the output.out file is overlaid (re-used) if you are creating SQL statements for multiple components. You can run the createDB.sh script one time for each component and review the output after each run. Another way is to modify the createDB.sh script to append to the $Output file (Change all occurances of ‘`> $Output`’ to ‘`>> $Output`’).

8.4 Runtime problems

We found that most of the problems that we encountered occurred when we were configuring the topology or when the servers were first started after configuration. We did however encounter the problems detailed in the sections that follow.

CWWBZ0052W: BPC Explorer REST API Verify this URL is correct

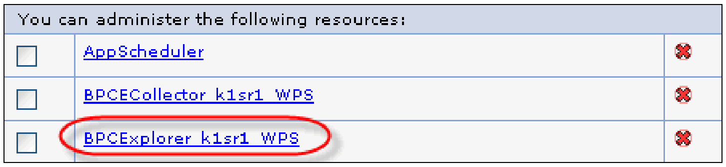
The following error is displayed in the Deployment Manager logs when running the Deployment Environment:

```
ExtendedMessage: CWWBZ0052W: Business Process Choreographer Explorer
BPCExplorer_k1sr1_WPS was configured to use the
https://wtsc42.itso.ibm.com:9443/rest/bpm/bfm URL to access the Representational
State Transfer (REST) API of the Business Flow Manager component on the k1sr1_WPS
deployment target. Verify that this URL is correct.
```

This error occurs because the Deployment Environment wizard sets up the URL for the System REST endpoints according to the details that you enter on a panel in the wizard, but the BPC Explorer itself references REST URLs, which you must manually update.

To fix this issue:

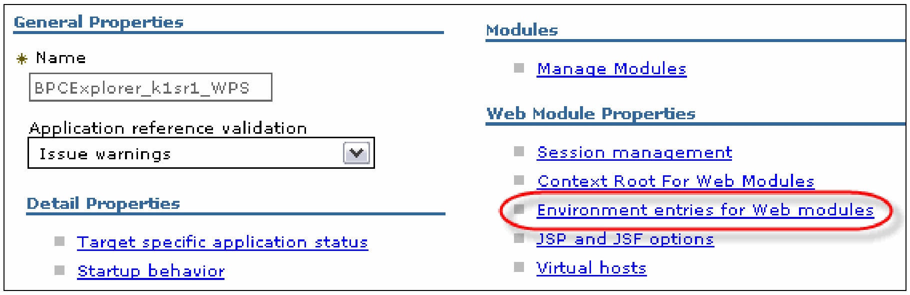
1. Using the Integrated Solutions Console, navigate to **Applications** → **Enterprise applications**, and click the BPC Explorer application that is mentioned in the message in Figure 8-4.



You can administer the following resources:		
□	AppScheduler	✖
□	BPCECollector_k1sr1_WPS	✖
□	BPCExplorer_k1sr1_WPS	✖

Figure 8-4 Click BPC Explorer

2. Click **Environment entries for Web modules**, as shown in Figure 8-5.



General Properties

* Name
BPCExplorer_k1sr1_WPS

Application reference validation
Issue warnings

Detail Properties

- Target specific application status
- Startup behavior

Modules

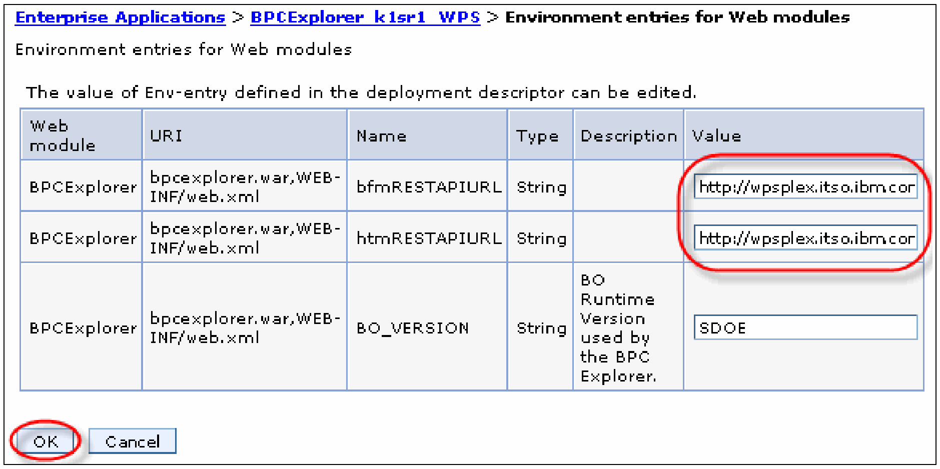
- Manage Modules

Web Module Properties

- Session management
- Context Root For Web Modules
- Environment entries for Web modules
- JSP and JSF options
- Virtual hosts

Figure 8-5 Click Environment Entries for Web Modules in order to change the REST URL:port

3. Set the correct REST URL and port for the Business Flow Manager (bfm) and the Human Task Manager (htm), as shown in Figure 8-6, and then click **OK**. Because this is a cluster, these REST URLs must reference the distributed VIPA host name and port of an HTTP server, so that REST requests are balanced across the cluster.



Enterprise Applications > BPCExplorer_k1sr1_WPS > Environment entries for Web modules

Environment entries for Web modules

The value of Env-entry defined in the deployment descriptor can be edited.

Web module	URI	Name	Type	Description	Value
BPCExplorer	bpcexplorer.war,WEB-INF/web.xml	bfmRESTAPIURL	String		http://wpsplex.itso.ibm.cor
BPCExplorer	bpcexplorer.war,WEB-INF/web.xml	htmRESTAPIURL	String		http://wpsplex.itso.ibm.cor
BPCExplorer	bpcexplorer.war,WEB-INF/web.xml	BO_VERSION	String	BO Runtime Version used by the BPC Explorer.	SDOE

Figure 8-6 Set the URL:port for the REST APIs to the HTTP Server host name:port

4. Save the configuration changes.

CannotLoadUserException

After configuring WebSphere Process Server in a node, there are errors in the controller and servant logs during initialization.

The controller log showed this error:

```
BBOO0220E: SECJ0348E: Could not get the display name of the user
server:b8cell_b8nodea_b8cell.AppTarget.b8nodea.0
```

In the servant log, there was this error:

```
Category: SEVERE
  ExtendedMessage: com.ibm.mm.server.model.user.exception.CannotLoadUserException:
   com.ibm.mm.server.model.user.exception.CannotLoadUserException:
   com.ibm.websphere.security.EntryNotFoundException: Display name not found in
registry
```

Notice that the message in the controller says the 'user server' is b8cell_b8nodea_b8cell.AppTarget.b8nodea.0. This is an automatically generated identity of the server. Security traces were enabled: `com.ibm.ws.security.*=all:ORBRas=all:SASRas=all` and `TRACEDETAIL=(E)`.

From the trace in the servant log we saw that the java Subject was the server identity, and it appears that the display name of the server cannot be resolved in the user registry. The display name is probably the name that is obtained from mapping a certificate to a user ID. The user registry, in this case, was LocalOS (RACF), so it is expected that an identity with a name, such as `b8cell_b8nodea_b8cell.AppTarget.b8nodea.0`, is not found in the user registry.

To test this idea, we changed the cell security so that the server identity was set to the started task user ID:

1. Click **Security** → **Global security** → **Configure**. The Available realm definition should indicate Local Operating System when you configure the cell to use SAF/RACF security.
2. On the next panel, clear the `Automatically generated server identity` option. Instead click **User identity for the z/OS started task**, and click **OK**.
3. Save the configuration change, synchronize to the nodes, and restart the whole cell.
4. After restarting the cell the error messages no longer appeared. Traces showed that this part of the startup was now running under the started task user ID, B8ACRU, and because that user existed in the LocalOS user registry, the code was happy to continue without producing an error.

This process is probably just a circumvention because it should be possible to initialize WebSphere Application Server for z/OS using the Automatically generated server identity option. It is possible that the underlying cause was APAR PK97688, which is part of WebSphere Application Server service level 7.0.0.8. However we were unable to test that service level.

8.4.1 Solving DB2 runtime problems

Many of the problems that we discuss in this section occurred when we tried starting the servers or the messaging engines.

Tracing DB2 SQL statements

Enable trace com.ibm.ws.db2.*=all, which causes SQL statements to be written to the trace. Use the following MVS modify command to turn this on dynamically:

```
F <server_name>,tracejava='com.ibm.ws.db2.*=all'
```

Start BPE tracing dynamically with the following command:

```
F server-name,TRACEJAVA='com.ibm.bpe.*=all=enabled'
```

Schema problems (general)

Set the currentSchema property to the schema b8cell, not the JAAS Alias user ID b8dbu.

Business Space schema

The Business Space data source was hard-coded to have a Schema name of IBMBUSSP in various cases.

Messaging Engine Schema

For the message engines, do not set currentSchema in the ME data sources. Leave both currentSchema and currentSQLID blank in the data source custom properties. Set the schema for the bus in the Message Store definition, which you get to by going through Buses > <bus> Message Engines > <message engine id> > Message Store.

On that same page, set the J2C alias that are associated with the message store, so that J2C alias can map to b7dbu.

Missing jdbc properties file

```
CWSIS1501E: The data source has produced an unexpected exception: java.sql.SQLException:
[jcc][1031][11001][3.57.91] Global properties file "B9_DB2JccConfiguration.properties"
cannot be accessed. ERRORCODE=-4475, SQLSTATE=nullDSRA0010E: SQL State = null, Error Code =
-4,475
```

```
Problem/FIX: Application servers > b9sr01.AppTarget.b9nodea.0 > Process definition > Adjunct
> Java Virtual Machine > Custom properties > db2.jcc.propertiesFile  >>
/etc/d9fg/B9_DB2JccConfiguration.properties
```

In our case, it was missing the full path, so we entered the proper name, which corrected the problem.

-443 errors after applying DB2 maintenance

```
com.ibm.db2.jcc.am.SqlException: [jcc][50053][12311][3.57.91] T2zOS exception:
[jcc][T2zos]ab.readExecuteCallInternal:nativeExecuteCall:9016: DB2 engine SQL error, SQLCODE = -443,
SQLSTATE = 38112, error tokens = SQLCOLUMNS;SQLCOLUMNS;-805 DB9F.DSNASPCC.DSNACOL8.1887F0F9062B73D4;DSN
ERRORCODE=-443, SQLSTATE=38112
```

To fix this error, rebind of the DSNASPCC package.

-551: Connecting to DB2

When trying to start the BPC Event Collector and Observer applications, they failed to start and gave -551 errors in the servant logs.

```
com.ibm.db2.jcc.am.jo: [jcc][50053][12311][3.57.91] T2zOS exception:
[jcc][T2zos]T2zosPreparedStatement.readPrepareDescribeOutput_:nativePrepareInto:1560: DB2
engine SQL error, SQLCODE = -551, SQLSTATE = 42501, error tokens =
B8DBU;SELECT;SYSIBM.SYSTABLES ERRORCODE=-551, SQLSTATE=42501
```

When WebSphere Process Server for z/OS components start they query SYSTABLES to see if the required tables exist. If the tables do not exist, they handle the error and they assume that the tables do not exist. The query attempts to create the tables, which means that b8dbu tries to issue CREATE TABLE, and so on, which it should not be authorized to do.

Sometimes, you do not see the fact that you cannot select from SYSTABLES. So be careful here because an error in the log that suggests b8dbu is not authorized to create tables does not mean that you have to grant create to b8dbu. It usually means that b9dbu failed to select from systables and therefore thinks that the tables do not exist.

Other DB2 connect problems

The following errors occurred, which we fixed by entering the correct values for:

- STEPLIBs
- LIBPATHs
- CLASSPATHs
- DB2 Location Name
- JCC Properties
- Authentication alias UserID and password
- Correct GRANT privileges

```
WSVR0194E: Composition unit WebSphere:cuname=BPCExplorer_b9sr01.AppTarget in BLA
WebSphere:blaname=BPCExplorer_b9sr01.AppTarget failed to start.
```

```
J2CA0045E: Connection not available while invoking method
createOrWaitForConnection for resource jdbc/BPEDB
```

```
CWWBB0638E: Failed to connect to the database. The error reported by the
database is: 'com.ibm.websphere.ce.cm.ConnectionWaitTimeoutException: Connection
not available, Timed out waiting for 180000'.
```

```
com.ibm.db2.jcc.am.jo: [jcc][50053][12311][3.57.91] T2zOS exception:
[jcc][T2zos]T2zosStatement.readExecuteImmediate:nativeExecuteImmediate:953: DB2
engine SQL error, SQLCODE = -553, SQLSTATE = 42503, error tokens = B8C1C
ERRORCODE=-553, SQLSTATE=42503
```

```
CWMFN4004E: Unable to connect to the Business Space database. Possible causes
are the database was down, database connectivity was lost, network error, incorrect
or insufficient credentials to connect to the database.java.sql.SQLException:
[jcc][50053][12311][3.57.91] T2zOS exception:
[jcc][T2zos]T2zosReusableConnection.flowConnect:execConnect:1347: DB2 engine SQL
error, SQLCODE = -922, SQLSTATE = 42505, error tokens = CONNECT;00D31024
ERRORCODE=-922, SQLSTATE=42505DSR A0010E: SQL State = 42505, Error Code = -922
```

-553 errors: GRANTs

To figure out the root of these problems, you can trace the SQL and see what user ID is being used in the set current sql ID and what ID the SQL is trying to use to access DB2.

After you know what ID is being used and what objects are being accessed, you can then look in the DB2 catalog to see if the ID has the authority to access that ID.

-553 errors

The -553 error occurs because it is trying to switch user IDs on the connection to DB2 that is defined by the CEI Message Engine's data source and cannot. The reason it is trying to switch user IDs from the servant to B7C1C is because you have currentSQLID set on the CEI ME data source, and that should not be there. Simply remove currentSQLID from the data source custom properties, and you will not have the -553.
Also check that the CEI Bus' Message store correctly sets the schema to B7C1C.

-805 errors

A return code of -805 on DSNASPCC indicates that those packages must be rebound. When DB2 maintenance is changed, the DB2 packages are modified by PTFs.

Check with the HOLDDATA actions. In one case, part of the job in DSNTIJSG (install member in SDSNSAMP) must be run to rebind the packages. A good practice is to always rebind DB2 packages with any maintenance.

Rebind all packages in STEP DSNTIRU in job DSNTIJSG.

-904 errors: Resource unavailable

The following error message occurs in the BPC explorer when starting a process instance. The tablespace INSTA60 of the BPC database is not available.

```
"com.ibm.db2.jcc.am.SqlException - T2zOS exception: DB2 engine SQL error, SQLCODE = -904,
SQLSTATE = 57011, error tokens = 00D70025;00000220;B8DB2VC.DSNDBC.B8CELL.INSTA60.I0001.A001
```

This error is caused by out of space on the volumes. To fix this issue, add more volumes to the storage group.

Other -922 errors

Apparent logon failures of the J2C authentication user ID due to invalid passwords can in fact be caused because the user ID has not been permitted to the RACF APPL profile that is protecting the cell. When RACF class APPL is active and a profile exists for the cell's security domain prefix, all RACF user ID that authenticate in the cell, including all J2C authentication aliases, must be permitted to the APPL profile.

If the RACF APPL class is active and a RACF profile for the cell was created in the APPL class, then additional permissions might be needed. By default the BBOxBRAK job permits xxCFG to the APPL profile, but it does not know about the other RACF users/groups created for WPS.

There are commands in the spreadsheet or in the xxRACFG job that are commented out and permit various groups to the APPL. If you have APPL active, run these RACF permit statements.

Fixing security errors

In this section, we discuss problems that are related to security settings.

Password issues

In SYSLOG there is:

```
ICH408I USER(B9DBU   ) GROUP(B9DBG   ) NAME(B9 WPS JDBC ALIAS   )
  LOGON/JOB INITIATION - INVALID PASSWORD
IRR013I  VERIFICATION FAILED. INVALID PASSWORD GIVEN.
```

Double check that:

- The password on the J2C alias is correct.
- The password in RACF is correct.

Fix it in RACF by issuing these commands:

```
PW USER(B9DBU)     NOINTERVAL

 ALU B9DBU          PASSWORD(B9DBU) NOEXPIRED
```

Then go into the JDBC J2C aliases, update all the password fields to be B9DBU, and restart the cell.

EJBROLE authorizations

Note: These are really configuration errors, but do not show up until you try to run an application or open the BPC explorer.

Many EJBROLE profiles are used in WebSphere Process Server for z/OS, Business Space powered by WebSphere and WebSphere Business Services Fabric for z/OS. When troubleshooting an application or administrative function in your browser, you might not see the error message in your browser, but it might be in the system log. Using SDSF, browse the log, and look for ICH408I messages, shown in Example 8-22, around the time of your problem.

Example 8-22 RACF ICH408I message on console (syslog)

```
ICH408I USER(B7FABADM) GROUP(B7FADMG ) NAME(B7 FABRIC ADMIN )
B7.FabricStudioUser CL(EJBROLE )
INSUFFICIENT ACCESS AUTHORITY
ACCESS INTENT(READ ) ACCESS ALLOWED(NONE )
+BBOO0222I: SECJ0129E: Authorization failed for B7FABADM while
invoking GET on default_host:fabric/app, Authorization failed, Not
granted any of the required roles: FabricAdministrator
FabricGovernanceAdministrator FabricPerformanceUser
FabricSubscriberManager FabricBasicUser FabricStudioUser
```

To resolve this error, permit the user ID to one of the EJBROLE profiles. In this case, the user ID B7FABADM, which we wanted to be the Fabric Administrator user ID, had not been granted to any of the roles.

8.4.2 Errors you can ignore

You can ignore the messages in this section, although they make it hard to find the Real SEVERE Errors.

Adjuncts

CHFW0030E: `Error starting chain _InboundTCPProxyBridgeService`

Servants and Controllers

SECJ0384E: `Trust Association Init Error. The Trust Association interceptor implementation ... failed. The error status/exception is 1. If you receive this error message in association with a trust association interceptor that you are not using, you can ignore this message.`

CWMU2003E: `Unable to activate [MonitorAdminTaskForwarder] MBean.`

ADMN0005E: `The service is unable to activate MBean.`

CWSPN0009E: `SPNEGO Trust Association Interceptor configuration is not valid. Failure condition: com.ibm.ws.security.spnego.isEnabled JVM property is false or not set, no further processing will be done.. If you are not using the SPNEGO TAI, you can ignore this message.`

CHFW0030E: `Error starting chain _InboundTCPProxyBridgeService because of exception com.ibm.wsspi.channel.framework.exception.RetryableChannelException: An exception was thrown when attempting to start the TCPProxyChannel.`

Errors you can fix

You can prevent this error by installing the favicon.ear file (See Techdoc TD105415):

```
SRVE0255E: A WebGroup/Virtual Host to handle /favicon.ico has not been defined.
```

A

Additional material

In this chapter, we refer to additional material that can be downloaded from the Internet.

Locating the Web material

The Web material that is associated with this book is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser at:

`ftp://www.redbooks.ibm.com/redbooks/SG247831`

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, SG247831.

Using the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material .zip file into this folder.

The contents of the subdirectories are:

/doc Some useful notes and an example .profile.

/samples The sample application's ITSOApp.ear and ISTO_ImplApp.ear files.

/wpswork Sample jython scripts together with documentation. Copy the contents of this directory into your wpswork directory on z/OS.

/spreadsheet The planning spreadsheet.

IHS.CNTL Sample JCL to createIBM HTTP Servers.

WPS.CNTL Sample JCL for WebSphere Process Server for z/OS.

WBSF.CNTL Sample JCL for WebSphere Business Services Fabric for z/OS.

The sample JCL in the .CNTL directories was generated by the spreadsheet for our B7 cell. You can use the JCL that we provided in the additional material, but you face quite a lot of editing to make the JCL suitable for your environment. It is probably better to plan your cell using the spreadsheet, let it do the work of customizing JCL for your environment, and then copy the JCL that it generates from the JCL_P and JCL_S worksheets.

If you do decide to use the JCL in the Additional Material .zip file, upload the IHS.CNTL, WPS.CNTL, and WBSF.CNTL directories in text/ASCII mode into z/OS partitioned data sets that are allocated as FB LRECL 80 with about two tracks of space.

We did not provide DDLs in the Additional Material .zip file because it is safer to generate the DDL by following one of the methods described in the book.

The /wpswork directory contains a mixture of WORD documents, shell scripts, and jython scripts. Upload all files with a .sh extension to your z/OS system in text/ASCII mode so that they are stored in a wpswork directory on z/OS in EBCDIC. Upload the files with a .py extension and the whole of the jscripts directory to the wpswork directory in BINARY mode.

Abbreviations and acronyms

AST	Application Server Toolkit
BFM	Business Flow Manager
BPC	Business Process Choreographer
BPM	Business Process Management
BRM	Business Rules Manager
CEI	Common Event Infrastructure
DBA	Database Administrator
DDL	Data Definition Language
DDT	Database Design Tool
DMZ	De-Militarized Zone
HFS	Hierarchical File System
HTM	Human Task Manager
IBM	International Business Machines Corporation
IEA	IBM Education Assistant
ISA	IBM Support Assistant
ISC	Integrated Solutions Console
ISS	Integrated Security Services
ITD	Integrated Technology Delivery
ITSO	International Technical Support Organization
JCC	Java Common Connectivity
JVM	Java Virtual Machine
LNA	LDAP Native Authentication
LPAR	Logical Partition
MDD4J	Memory Dump Diagnostic for Java
ND	Network Deployment
ODR	On-Demand Router
PDS	Partitioned Dataset
PSP	Preventative Service Planning
REST	Representational State Transfer
RRS	Resource Recovery Services
SOA	Service-Oriented Architecture
UDF	User Defined Function
USS	UNIX Systems Services
VCE	Visual Configuration Explorer
W3C	Worldwide Web Consortium
WCT	WebSphere Configuration Tool
WSC	Washington Systems Center
zAAPs	z Application Assist Processors
zIIP	z Integrated Information Processor
zPMT	zProfile Management Tool

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see "How to get Redbooks" on page 280. Some of the documents referenced here might be available in softcopy only.

- *Best Practices for SOA Management,* REDP-4233
- *IBM WebSphere Application Server V6.1 Security Handbook,* SG24-6316
- *IBM WebSphere Business Process Management V6.1 Performance Tuning,* REDP-4431
- *Patterns: SOA Foundation Service Creation Scenario,* SG24-7240
- *Production Topologies for WebSphere Process Server and WebSphere ESB V6,* SG24-7413
- *WebSphere Application Server Network Deployment V6: High Availability Solutions,* SG24-6688
- *WebSphere Application Server V6.1: JMS Problem Determination,* REDP-4330
- *WebSphere Business Process Management V6.1.2 Production Topologies,* SG24-7665
- *WebSphere Business Process Management V6.2 Production Topologies,* SG24-7732
- *WebSphere Business Process Management V7 Production Topologies,* SG24-7854
- *WebSphere Process Server for z/OS: Configuring a Network Deployment Environment,* REDP-4388
- *z/OS: WebSphere Business Process Management V6.2 Production Topologies,* SG24-7733

Other publications

These publications are also relevant as further information sources:

- *DB2 Universal Database for z/OS Application Programming Guide and Reference for Java,* SC18-7414
- *Program Directory for WebSphere Process Server for z/OS V7,* GI13-0553
- *The Mixed Platform Stack Project: Deploying a secure SOA solution into z/OS,* WP101300
- *WebSphere Business Service Fabric 7.0 Program Directory,* GI11-4301-00

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks publications, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

z/OS: WebSphere Business Process Management V7 Production Topologies

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

z/OS: WebSphere Business Process Management V7 Production Topologies

Create production topologies for WebSphere Process Server

Incorporate WebSphere Business Services Fabric

Learn by example with practical scenarios

In this IBM Redbooks publication, we address the configuration, administration, and security of the key runtime environments in business process management: WebSphere Process Server V7.0 and WebSphere Business Services Fabric V7.0 for z/OS. This book provides detailed guidance to z/OS system and database administrators who want to configure WebSphere Business Process Management production topologies.

We introduce production topology concepts and terminology and explore the differences between production topologies on distributed platforms and z/OS. Through a series of step-by-step instructions, you will learn how to create and verify a production topology environment for WebSphere Process Server V7 for z/OS.

We extend the production topology concept for WebSphere Process Server by describing step-by-step how to add WebSphere Business Services Fabric V7 for z/OS into the topology. You also get problem diagnosis and prevention guidance to use when you create your own production topologies.

A separate publication that covers distributed platforms is also available: *WebSphere Business Process Management V7 Production Topologies*, SG24-7854.

SG24-7831-00 ISBN 0738434353