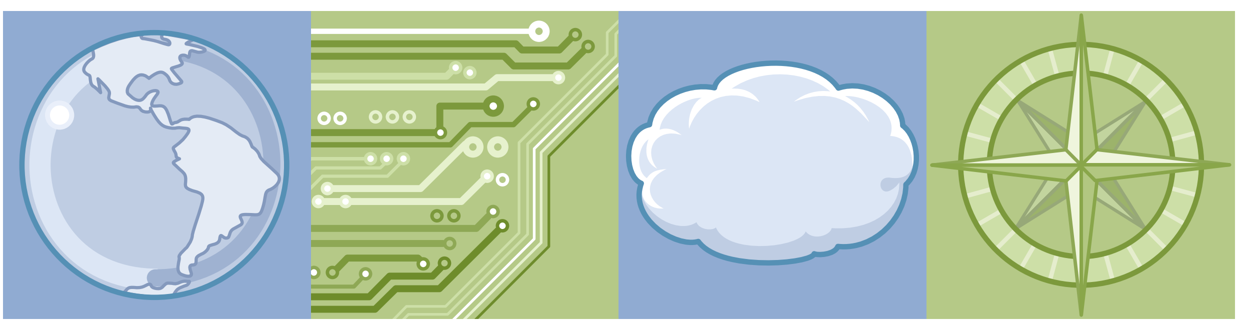
IBM Training

Student Notebook

IBM Business Process Manager V8.5 Performance and Tuning

Course code WB868 / ZB868 ERC 1.0



WebSphere Education

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

AIX®	CICS®	DB™
DB2®	developerWorks®	Express®
HACMP™	IMS™	Lombardi®
PartnerWorld®	pureScale®	Quantify®
Rational®	Redbooks®	Tivoli®
WebSphere®	z/OS®	

Intel and Intel Core are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Microsoft and Windows 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.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

VMware and the VMware “boxes” logo and design, Virtual SMP and VMotion are registered trademarks or trademarks (the “Marks”) of VMware, Inc. in the United States and/or other jurisdictions.

Other product and service names might be trademarks of IBM or other companies.

June 2014 edition

Contents

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

AIX®	CICS®	DB™
DB2®	developerWorks®	Express®
HACMP™	IMS™	Lombardi®
PartnerWorld®	pureScale®	Quantify®
Rational®	Redbooks®	Tivoli®
WebSphere®	z/OS®	

Intel and Intel Core are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Microsoft and Windows 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.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

VMware and the VMware “boxes” logo and design, Virtual SMP and VMotion are registered trademarks or trademarks (the “Marks”) of VMware, Inc. in the United States and/or other jurisdictions.

Other product and service names might be trademarks of IBM or other companies.

Course description

IBM Business Process Manager V8.5 Performance and Tuning

Duration: 3 days

Purpose

In this three-day course, you learn the advanced skills that are needed to monitor and tune the runtime of IBM Business Process Manager V8.5 for improved performance.

IBM Business Process Manager is a comprehensive BPM product that provides the visibility and insight to effectively manage organizational business processes. This course focuses on both the Standard and Advanced editions of IBM Business Process Manager. You learn how to configure the IBM Business Process Manager V8.5 system beyond the default settings to improve runtime performance. You also learn about performance tuning methods and monitoring tools that are available.

The course introduces you to a tuning checklist that lists the major components and their associated configuration properties. You learn about the configuration parameters available to system administrators, and how knowledge of these configuration settings is necessary for improving the runtime performance of IBM Business Process Manager V8.5. The course also provides hints and tips on monitoring and configuring IBM Business Process Manager V8.5 through in-depth lectures of various components.

Hands-on exercises are provided on Linux to reinforce lecture content. The exercises give you practical experience with IBM Business Process Manager V8.5 performance tuning and monitoring. Finally, you are introduced to the Tivoli Performance Viewer and IBM Support Assistant, and you learn how to use these tools to monitor the runtime performance of a server.

After completing this course, you have a solid foundation in best practices for tuning IBM Business Process Manager V8.5, and the skills to diagnose performance problems.

The lab environment for this course uses the RedHat Linux platform.

Audience

This course is designed for anyone who implements, deploys, or manages applications that run in IBM Business Process Manager V8.5. It is useful for system administrators, deployment engineers, process administrators, support engineers, and integration developers.

Prerequisites

Before taking this course, you should have:

- IBM Business Process Manager administration skills, which you can learn by successfully completing course WB867 or ZB867, *Administration of IBM Business Process Manager Advanced V8.5*
- Basic WebSphere performance tuning skills, which you can learn by successfully completing course WA815 or ZA815, *WebSphere Application Server V8.5.5 Performance Tuning*, or through practical experience with tuning a WebSphere Application Server environment
- Basic operating skills for the Linux operating system

Objectives

After completing this course, you should be able to:

- Explain the architecture and components for a typical IBM Business Process Manager deployment
- Explain basic performance concepts and methodologies
- Apply the Business Process Manager performance checklist and configure the server for better performance
- Identify key development best practices for IBM Process Designer and IBM Integration Designer
- Implement best practices for general WebSphere runtime performance
- Tune the target modules for various bindings
- Purge data that is no longer needed from the Business Process Manager environment
- Create efficient coaches and Coach Views
- Name the key best practices for the business flow manager (BFM)
- Identify key WebSphere monitoring facilities
- Evaluate various bottleneck patterns and determine a possible solution that is based on your observations
- Collect verbose GC trace logs from the runtime and analyze them to diagnose potential Java memory management issues
- Monitor application server performance by using WebSphere and the IBM Support Assistant
- Monitor and tune the JVM for optimum throughput and response time
- Monitor and tune connection pools for optimum performance
- Use the IBM Health Center tool to profile and tune Java EE applications
- Troubleshoot Business Process Manager performance problems

Agenda

Day 1

Course introduction
Unit 1: Overview of IBM Business Process Manager
Unit 2: Performance concepts and methodologies
Unit 3: Implementing for performance
Exercise 1: Performance testing
Unit 4: WebSphere monitoring and tuning concepts
Unit 5: IBM Business Process Manager performance concepts
Unit 6: Purging data in IBM Business Process Manager

Day 2

Exercise 2: Monitoring and purging data in the environment
Unit 7: Performance considerations for coaches
Unit 8: Threading
Unit 9: Business Process Choreography best practices
Unit 10: Performance tools
Exercise 3: Performance monitoring with Tivoli Performance Viewer
Exercise 4: Monitoring and tuning the environment

Day 3

Unit 11: Performance problem determination
Exercise 5: Hung thread issues
Unit 12: Database tuning
Unit 13: Java memory issues
Exercise 6: Analyzing Java memory
Unit 14: Course summary

Unit 1. Overview of IBM Business Process Manager

What this unit is about

This unit introduces the principles of business process management. You learn the capabilities of IBM Business Process Manager V8.5 for deploying BPM solutions. You also learn about the components and deployment topologies for IBM Business Process Manager.

What you should be able to do

After completing this unit, you should be able to:

- Describe the concepts of business process and business process management
- Describe the IBM products that support SOA application development
- Describe the capabilities of IBM Business Process Manager
- Describe the main components of IBM Business Process Manager
- Describe the deployment considerations for the databases
- Identify and explain the clustered topologies for IBM Business Process Manager

How you will check your progress

- Checkpoint

References

IBM Business Process Manager V8.5 IBM Knowledge Center

`http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/ditamaps/ic-homepage-bpm.html`

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Describe the concepts of business process and business process management
- Describe the IBM products that support SOA application development
- Describe the capabilities of IBM Business Process Manager
- Describe the main components of IBM Business Process Manager
- Describe the deployment considerations for the databases
- Identify and explain the clustered topologies for IBM Business Process Manager

Figure 1-1. Unit objectives WB868 / ZB8681.0

Notes:

WebSphere Education

Topics

- Business processes and business process management
- IBM Business Process Manager editions
- IBM Business Process Manager features and capabilities
- IBM Business Process Manager tools
- IBM Business Process Manager components
- IBM Business Process Manager deployment topologies

Figure 1-2. Topics WB868 / ZB8681.0

Notes:

1.1. Business processes and business process management

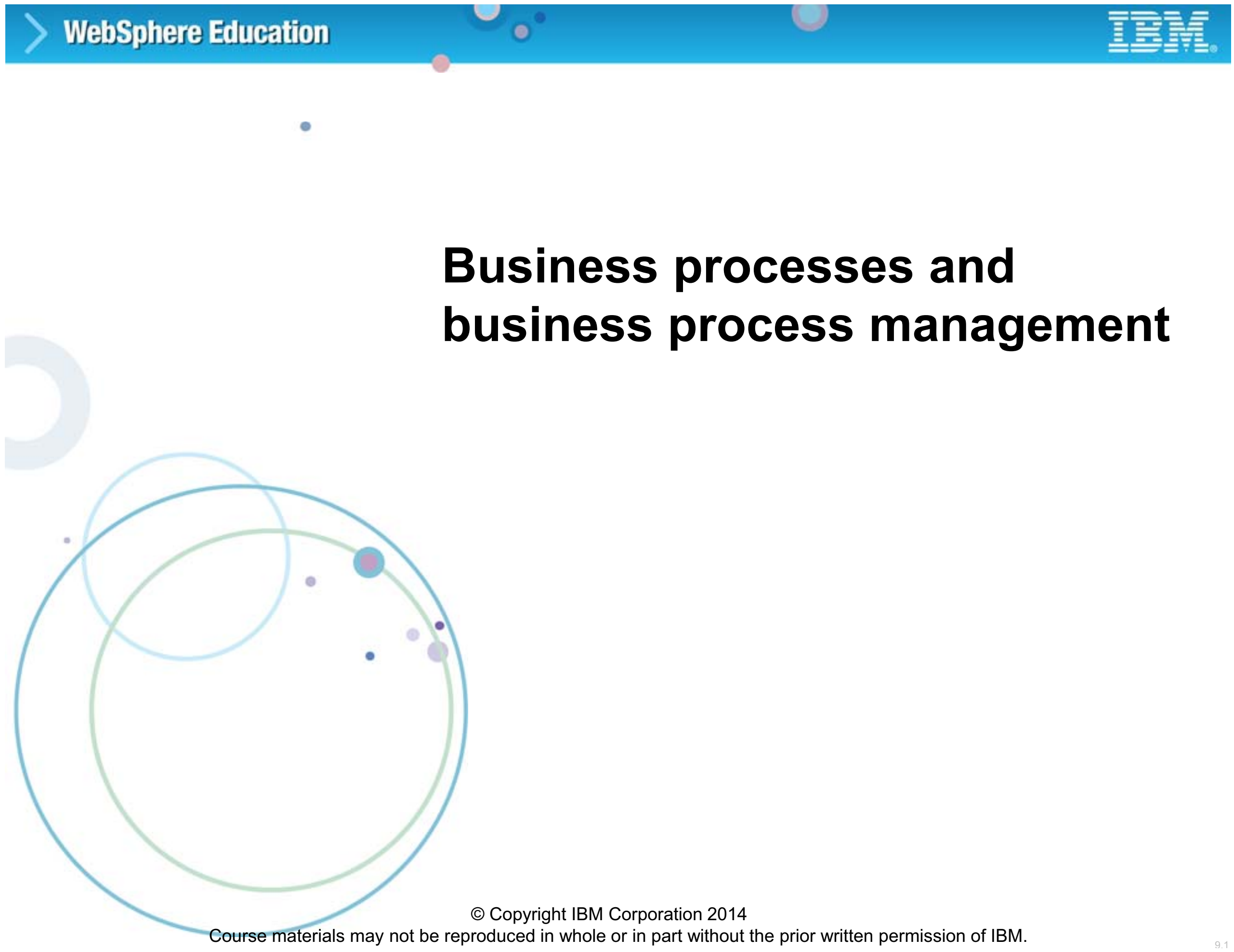


Figure 1-3. Business processes and business process management WB868 / ZB8681.0

Notes:

Business processes and business process management (BPM)

- A **business process** is a collection of service interactions and activities that are run to fulfill a business need
- A business process defines the potential execution order of services:
  - Defines how to coordinate interactions between a process instance and its partners
  - Specifies how to handle errors (faults)
  - Specifies other required technology patterns like compensation
- **Business process management (BPM)** is a systematic approach to improving business processes for an organization
  - BPM makes business processes more effective and efficient through a cycle of continuous improvement

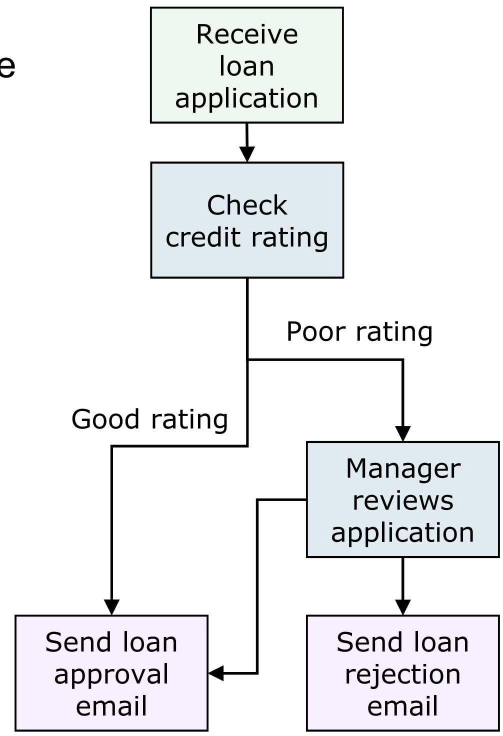


Figure 1-4. Business processes and business process management (BPM) WB868 / ZB8681.0

Notes:

Business process management (BPM) provides a way for business and IT to collaborate and optimize business processes across the process lifecycle that robust governance and compliance support. BPM is a discipline of software and expertise that is used to improve the performance, visibility, and agility of business processes. This expertise can be internal, such as an industry partner that is working with you. Expertise can be that of integrators and consultants such as IBM Global Services. Successful business outcomes come only with the involvement of the business expertise with IT expertise.

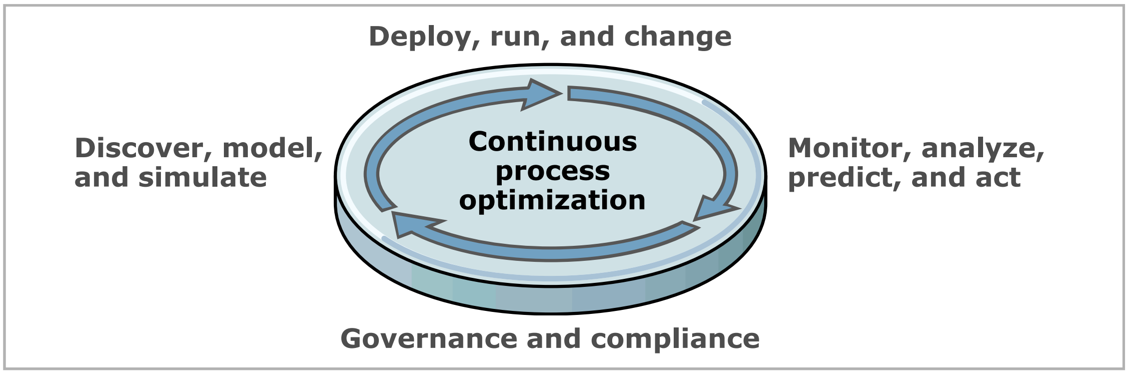
At its heart, business process management is about continuously optimizing business processes. Continuous optimization means working to improve business processes throughout the process lifecycle. The process lifecycle spans three steps. The first step is model and simulate, where business process improvements are documented and tested before deployment. The second step is to deploy, execute, and change, where new or improved processes are deployed in an automated, repeatable fashion with flexibility for rapid change. The last step is model, analyze, predict, and act where deployed processes are closely monitored and measured in real time to provide rapid response to emerging business situations in addition to identifying new process improvement opportunities.

Underpinning this cycle of continuous process improvement is the need for robust governance and compliance to ensure that business processes are operating consistently and are complying with internal policies and external regulations and controls. Processes are also most effective and efficient when they are provided with broad reuse of service-enabled IT assets. Optimizing end-to-end business processes across the lifecycle requires participation and collaboration between business and IT. Business and IT leaders must work together to develop the flexibility processes and underlying systems that allow the organization to embrace change and achieve a dynamic business network. Business process management provides the means and the tools to facilitate this collaboration.

WebSphere Education

IBM BPM lifecycle

- Implementing effective BPM is an ongoing journey



- Processes are designed, simulated, and tested based on a set of assumptions about many factors: costs, activity durations, resource availability, and probabilities around various process paths
- As BPM solutions are deployed and used in production environments, real-world analytics can challenge some key assumptions
- Continuous process improvement is taking production-time insights and refactoring them into original process models to evolve them for optimal performance

Figure 1-5. IBM BPM lifecycle WB868 / ZB8681.0

Notes:

Processes are initially designed, simulated, and tested based on a set of assumptions about many factors that include costs, activity durations, resource availability, and probabilities around various process paths. As BPM solutions are deployed to production environments, visibility into real-world analytics might challenge some of those key assumptions. Continuous process improvement is about taking those production-time insights and refactoring them into your original process models to evolve them for optimal performance.

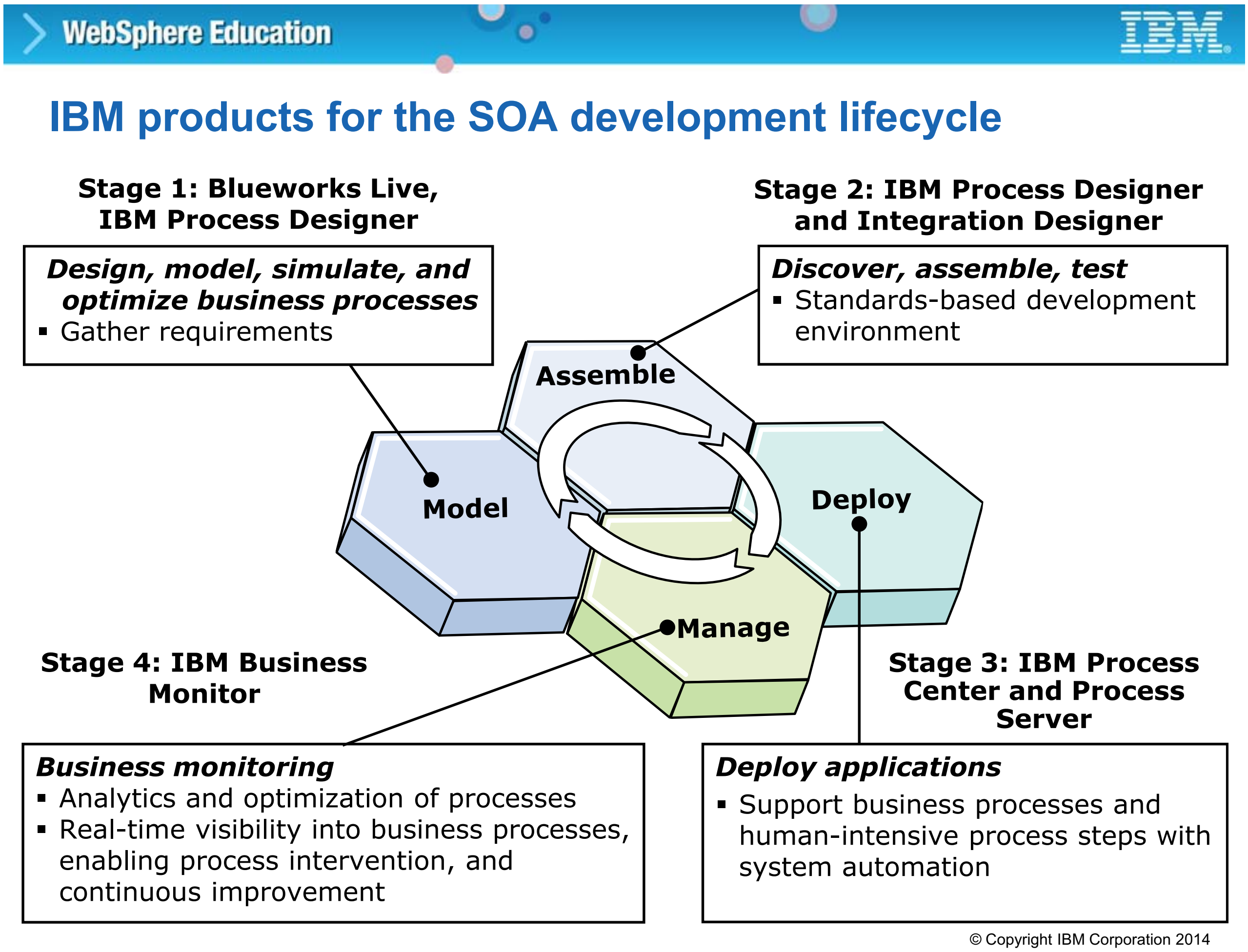


Figure 1-6. IBM products for the SOA development lifecycle WB868 / ZB8681.0

Notes:

Business process management (BPM) is a systematic approach to improving business processes for an organization. BPM makes business processes more effective and efficient through a cycle of continuous improvement.

BPM can include the steps of modeling, testing, deployment, running, and monitoring of the business processes.

1.2. IBM Business Process Manager editions

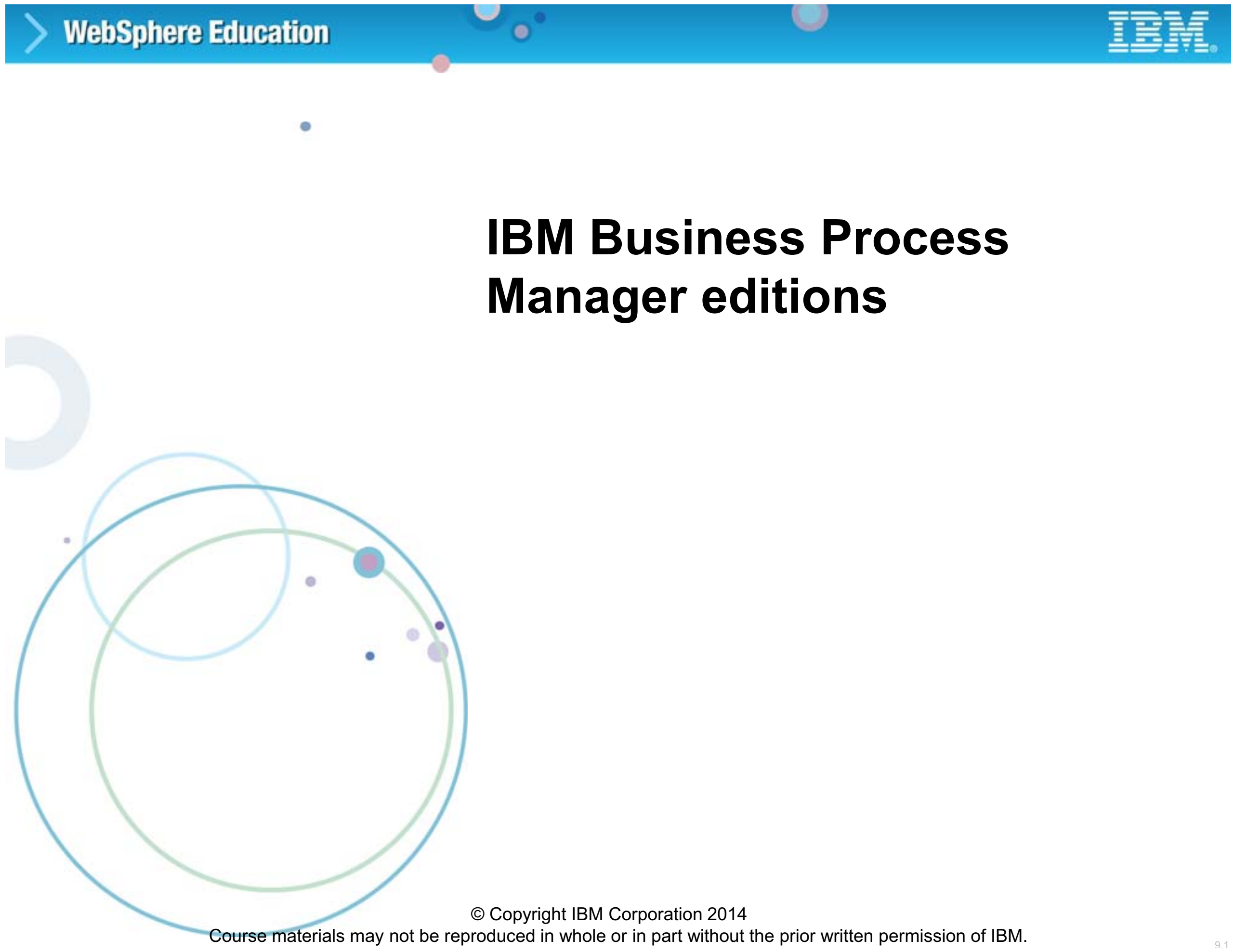


Figure 1-7. IBM Business Process Manager editions WB868 / ZB8681.0

Notes:

WebSphere Education

IBM Business Process Manager

- Tools for modeling, designing, implementing, and deploying business processes
- Includes:
  - **IBM Process Designer**: An authoring environment that is used for creating process models that contain automated and human tasks
  - **IBM Integration Designer**: An authoring environment that is used for creating process models and advanced implementations, including mediations, business rules, and human tasks
  - **IBM Process Center**: Includes a repository for all processes, services, and other assets that are created in the authoring environments
  - **IBM Process Server**: Provides a single runtime environment for supporting process models, service orchestration, and integration capabilities

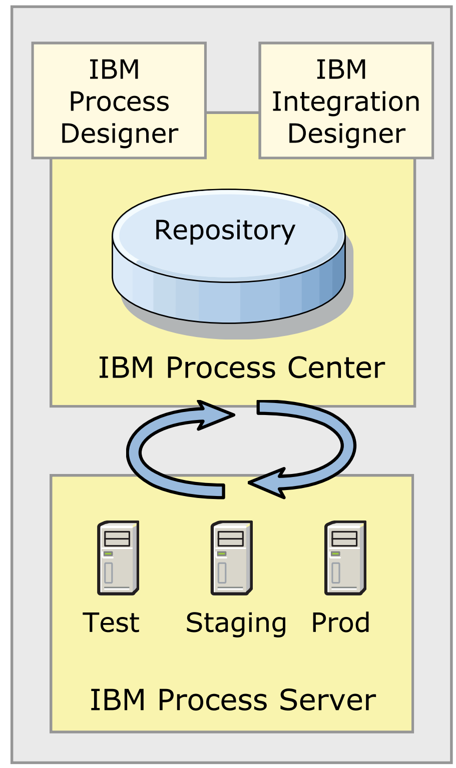


Figure 1-8. IBM Business Process Manager WB868 / ZB8681.0

Notes:

IBM Business Process Manager gives you the capability to model and run your business processes.

With IBM Business Process Manager, you get to choose the types of models and services you want to create, and the type of development environment you want to use.

For example, you decide whether you want to model the business processes by using BPMN (Business Process Model and Notation) or BPEL (Business Process Execution Language).

The integrated runtime supports both BPMN and BPEL for business processes, and also provides support for services and enterprise applications.

The components of IBM Business Process Manager include:

- IBM Process Server: The runtime platform
- IBM Process Center: A unified BPM asset repository
- IBM Integration Designer: An authoring environment for developing services and self-contained enterprise applications
- IBM Process Designer: An authoring environment for developing process models

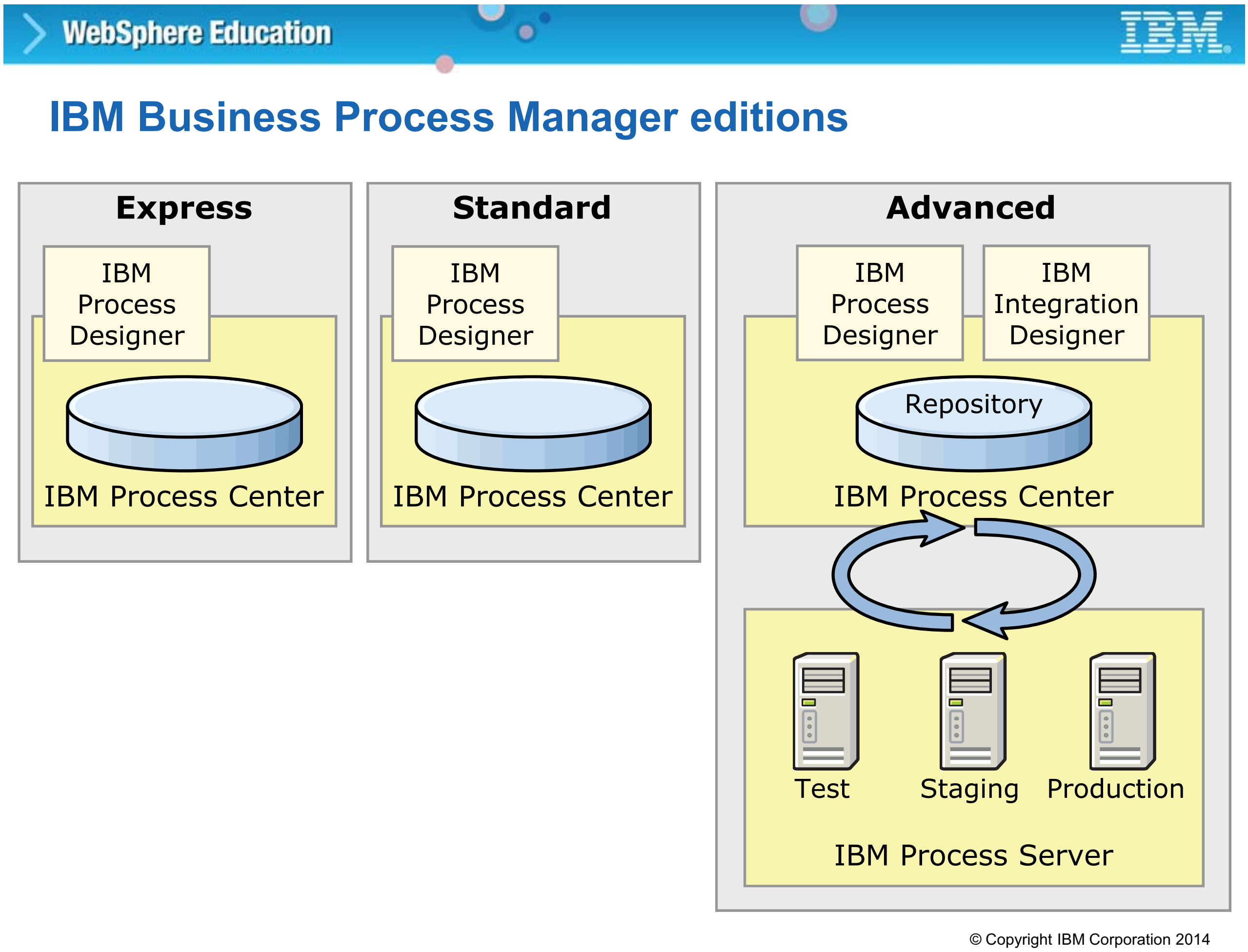


Figure 1-9. IBM Business Process Manager editions WB868 / ZB8681.0

Notes:

There are three different configurations of IBM Business Process Manager that correlate with typical entry points or stages in a business process management program for a company. Each configuration matches increasing levels of functional complexity.

IBM Business Process Manager Express is designed for a small business process management (BPM) project. It is configured to operate with a few users or a single server, with no clustering.

IBM Business Process Manager Standard is used for typical BPM projects. It is designed for multi-project improvement programs that have a high business involvement. The standard configuration offers an improved user productivity and basic system integration support.

IBM Business Process Manager Advanced offers the complete set of advanced BPM capabilities. It extends the support for high-volume process automation, with high quality of service. The advanced configuration offers built-in service-oriented architecture (SOA) components and all of the capabilities of WebSphere Enterprise Service Bus.

WebSphere Education

IBM Business Process Manager Express

- IBM Process Designer limited to three authors
- IBM Process Center
  - Two development cores
  - No high availability
- Process Server
  - Run BPMN processes
  - Rules and monitoring support
  - No clustering support
  - No BPEL, SCA, or ESB support
- Small number of users
- Single server and no clustering
- Simple installation
- Low pricing

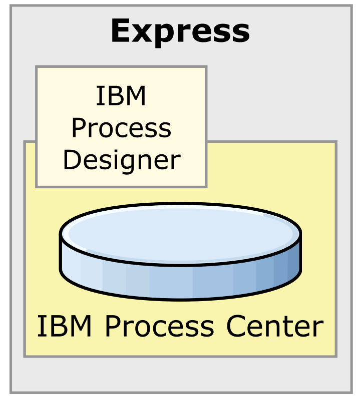


Figure 1-10. IBM Business Process Manager Express WB868 / ZB8681.0

Notes:

The entry level product is IBM Business Process Manager Express and is good for a group just starting out with business process management. The IBM Business Process Manager Express contains functions that come from the WebSphere Lombardi Edition heritage. IBM Business Process Manager Express can be installed only in a stand-alone server with no clustering and has a simple installation process. The pricing is low so that you can get started with a business process management project without a major investment.

WebSphere Education

IBM Business Process Manager Standard (1 of 2)

- IBM Process Designer
- IBM Process Center
  - Able to version BPMN processes
  - Clustering supported
- Process Server
  - Run BPMN processes
  - Run monitoring support
  - No BPEL, SCA, or ESB support
- Includes basic system integration support
- Focus on improved workflow and productivity
- Larger number of users than Express configuration

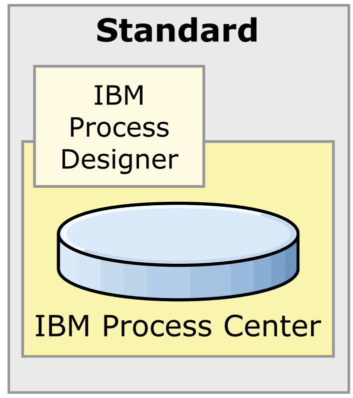


Figure 1-11. IBM Business Process Manager Standard (1 of 2) WB868 / ZB8681.0

Notes:

IBM Business Process Manager Standard is used for typical business process management projects that require a deeper business user engagement and IT collaboration through the process improvement lifecycle.

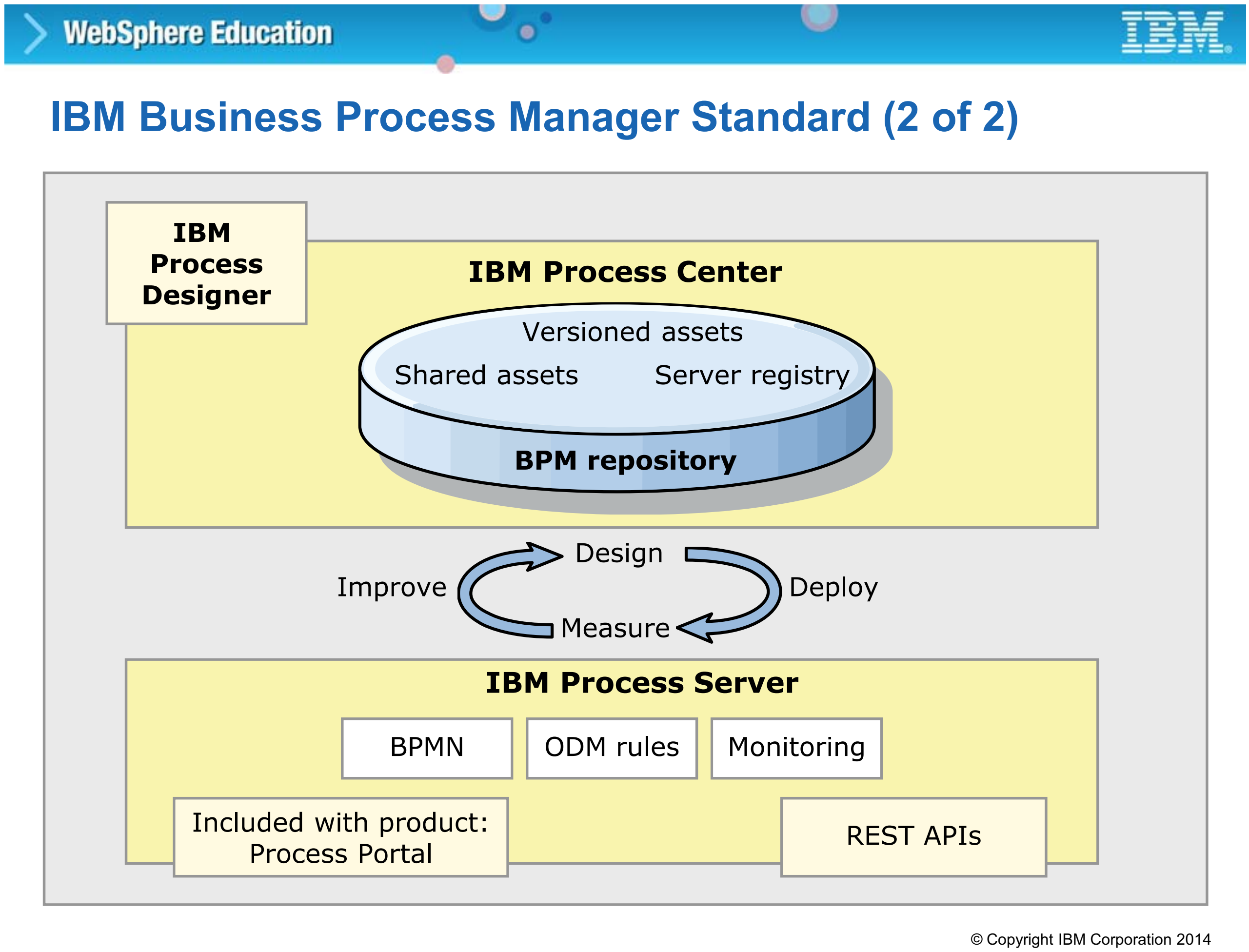


Figure 1-12. IBM Business Process Manager Standard (2 of 2) WB868 / ZB8681.0

Notes:

IBM Process Designer is an authoring environment that is used to create process models that contain automated and human tasks that are developed with the BPMN and BPD (business process definition) formats.

WebSphere Education

IBM Business Process Manager Advanced (1 of 2)

- Tools for modeling, designing, implementing, and deploying business processes and services
- Includes:
  - **IBM Process Designer**: An authoring environment that is used for creating process models in BPMN and BPD formats
  - **IBM Integration Designer**: An authoring environment that is used for creating processes and services
  - **IBM Process Center**: Includes a repository for all processes, services, and other assets
  - **IBM Process Server**: Includes a runtime environment for supporting process models and services

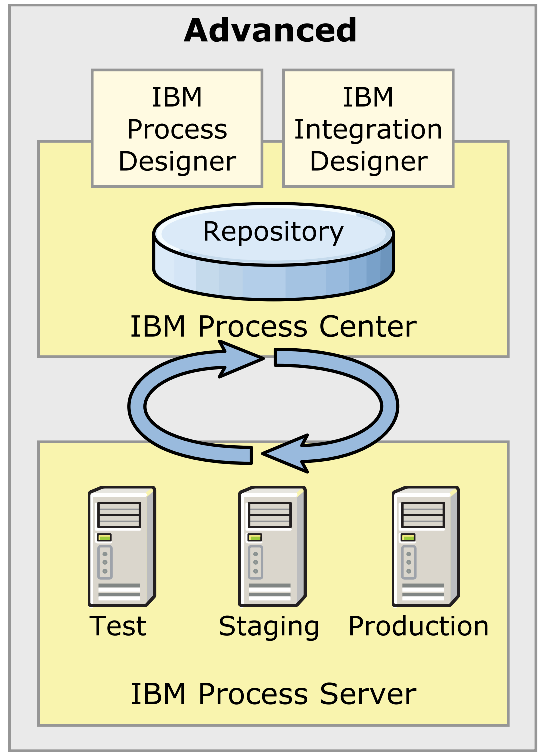


Figure 1-13. IBM Business Process Manager Advanced (1 of 2) WB868 / ZB8681.0

Notes:

IBM Integration Designer is an authoring environment that is used to create process models and advanced service implementations, including mediations, SCA (Service Component Architecture) modules, business rules, and BPEL (Business Process Execution Language) with human tasks.

IBM Process Center includes a repository for all processes, services, and other assets that are created in the authoring environments. IBM Process Server provides a single runtime environment for supporting process models, service orchestration, and integration capabilities.

WebSphere Education

IBM Business Process Manager Advanced (2 of 2)

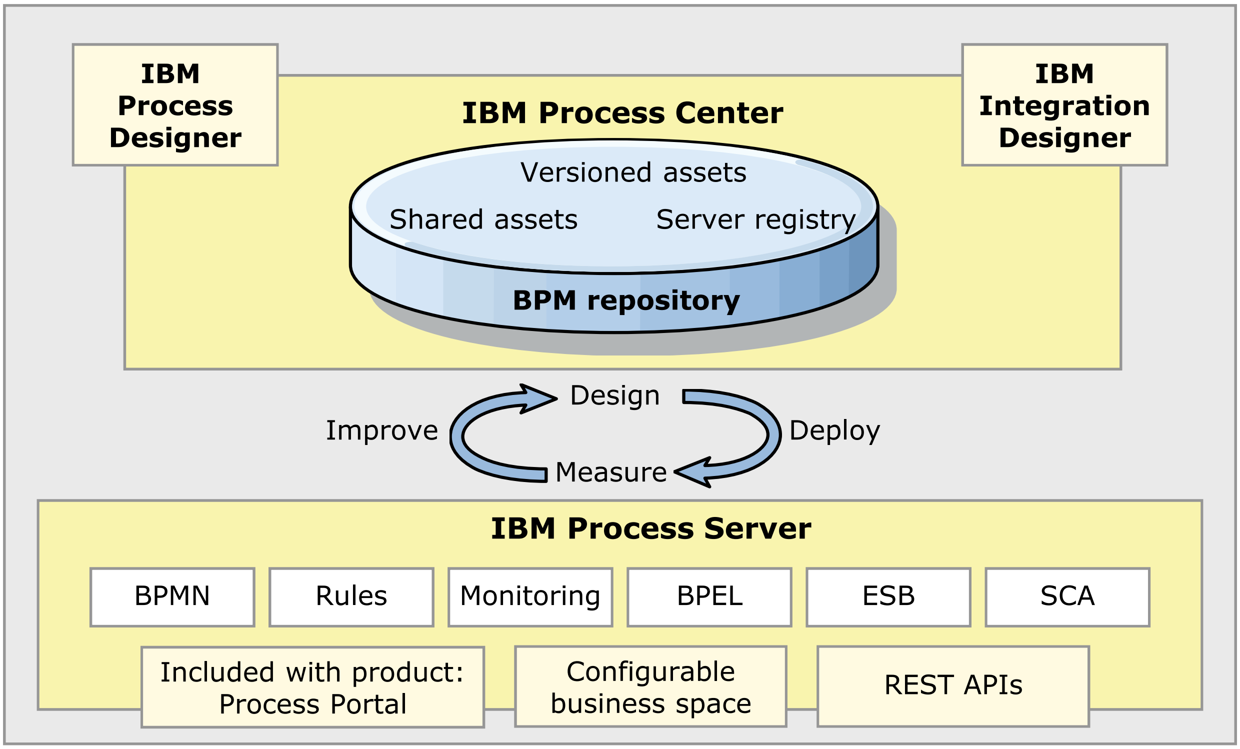


Figure 1-14. IBM Business Process Manager Advanced (2 of 2) WB868 / ZB8681.0

Notes:

IBM Business Process Manager uses a concept that is called the shared model. In simple terms, it means that no matter what is being done within the overall solution, there is only one common repository and a single representation of that solution. Because of the repository, it is impossible to get two phases of the same solution out of sync with each other.

The BPM component that is called the IBM Process Center realizes this shared model. IBM Process Center is a key component within IBM Business Process Manager. Part of the IBM Process Center is a data repository that is called the repository. Within the repository, there exists the representation of the solution. The IBM Business Process Manager tool connects as a client to the IBM Process Center to obtain copies of the solution for working upon. When a change is made and saved, the results are written back to the repository.

The IBM Process Center repository is implemented as tables within a database (commonly DB2).

1.3. IBM Business Process Manager features and capabilities

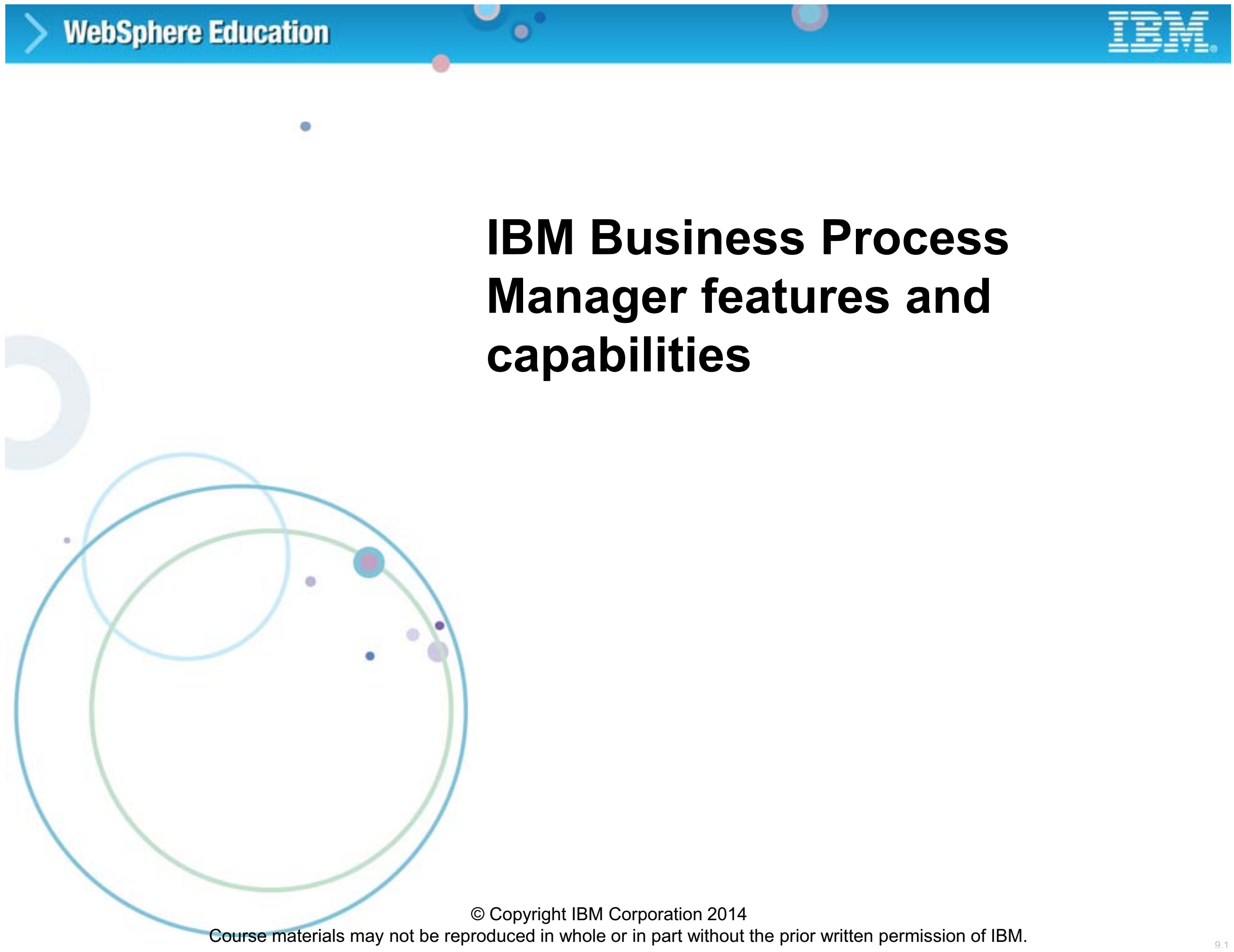


Figure 1-15. IBM Business Process Manager features and capabilities WB868 / ZB8681.0

Notes:

WebSphere Education

IBM Process Designer

- Tool to develop and manage business processes
- Model, simulate, and inspect business processes
- Artifacts: BPMN (Business Process Model and Notation) and BPD (business process definitions)

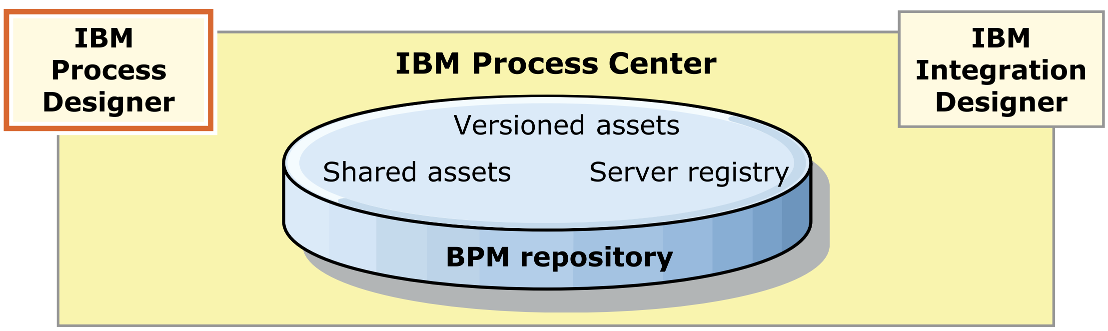


Figure 1-16. IBM Process Designer WB868 / ZB8681.0

Notes:

IBM Process Designer is an Eclipse-based tool that business process authors use. It offers capabilities to model and implement business processes as process applications. IBM Process Designer includes tools, the Process Inspector, and the Process Optimizer, for interacting with processes on the Process Center Server (playback server) or a Process Server deployment target.

A process is the major unit of logic in IBM Business Process Manager. It is the container for all components of a process definition, including services, activities, and gateways; timer, message, and exception events; sequence lines, rules, and variables. When you model a process, you create a reusable business process definition (BPD).

All Process Designer projects are contained in process applications. You store those process applications and associated artifacts in the Process Center repository.

Toolkits are containers that store library items (for example, BPDs) for reuse by process applications or other toolkits. Process applications can share library items from one or more toolkits, and toolkits can share library items from other toolkits.

Process applications that are developed in Process Designer can run any time on the Process Center server or can be saved to a snapshot and deployed on the Process Server. The same is true of services that are developed in Integration Designer and associated with process applications.

IBM Integration Designer

- **Available with the Advanced edition of the product only**
- Development tool for building SCA-based integration applications
- Provides a visual development environment for developing, assembling, testing, deploying, and managing integration modules and mediation modules
- Artifacts: Service Component Architecture (SCA) modules and libraries, and Advanced Integration services (AIS)

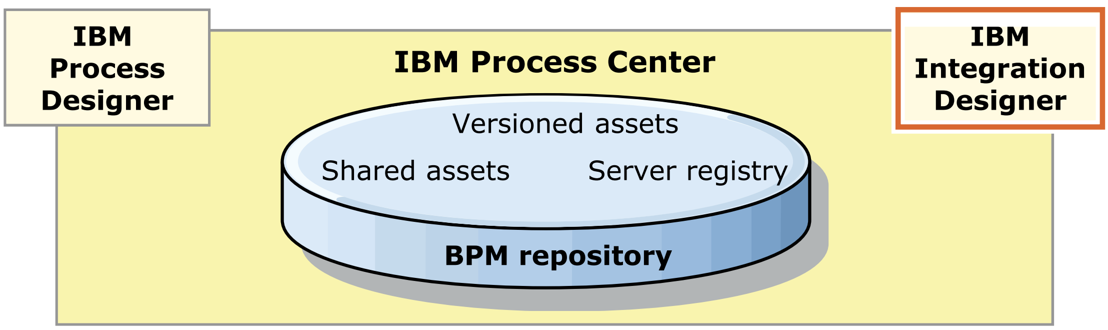


Figure 1-17. IBM Integration Designer WB868 / ZB8681.0

Notes:

IBM Integration Designer is an Eclipse-based tool that IT developers use. IBM Integration Designer is used to author complex integrations and fully automated processes that support process applications that are designed in the Process Designer.

It incorporates a fully integrated testing environment with test cases and test suites. Using IBM Integration Designer, IT developers build reusable SOA services, orchestrate those services, and access traditional systems.

The artifacts that are produced in IBM Integration Designer include:

- SCA components that contain one or more modules and libraries and are deployable as EAR files
- Advanced Integration services that are traditional BPEL processes.

The SCA modules and libraries that are created with IBM Integration Designer can be associated with a process application by using the Process Center.

The artifacts that are produced in Integration Designer can be used as services by processes that are created in Process Designer. In such cases, they are deployed with the process application.

WebSphere Education

IBM Process Center

- Repository for all Business Process Manager assets
- Lifecycle management and deployment of all applications
- Includes execution environment for development and testing
- Accessible from IBM Process Designer and from IBM Integration Designer
- Web interface by using IBM Process Center Console
- Includes Process Center server and the Performance Data Warehouse server

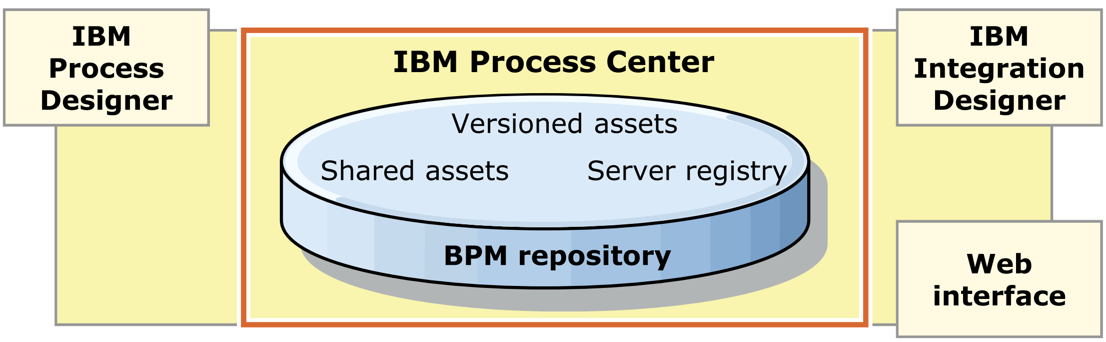


Figure 1-18. IBM Process Center WB868 / ZB8681.0

Notes:

You can use the IBM Process Center repository to share business process management assets with other users who are developing process applications and toolkits. These assets include process applications, reusable toolkits, monitor models, and more. It also can manage dependencies, versions, and deployment to servers.

The repository also maintains a registry of the process servers in the environment. The Process Center is involved in the modeling and development of process applications, running the applications during initial testing, and deploying the applications to test and production servers.

The IBM Process Center includes two servers, the Process Center server and the Performance Data Warehouse server. These servers allow developers who are working in Process Designer to run their process applications and store performance data for testing and playback during development efforts. Performance Data Warehouse retrieves tracked data from the Process Server or Process Center server at regular intervals.

WebSphere Education

IBM Process Server: Foundation

- WebSphere Application Server Network Deployment provides high availability, workload management, and qualities of service
- WebSphere ESB integration provides a communication infrastructure for integrating services, applications, and data
- IBM Process Server adds business process management functions to the operating system

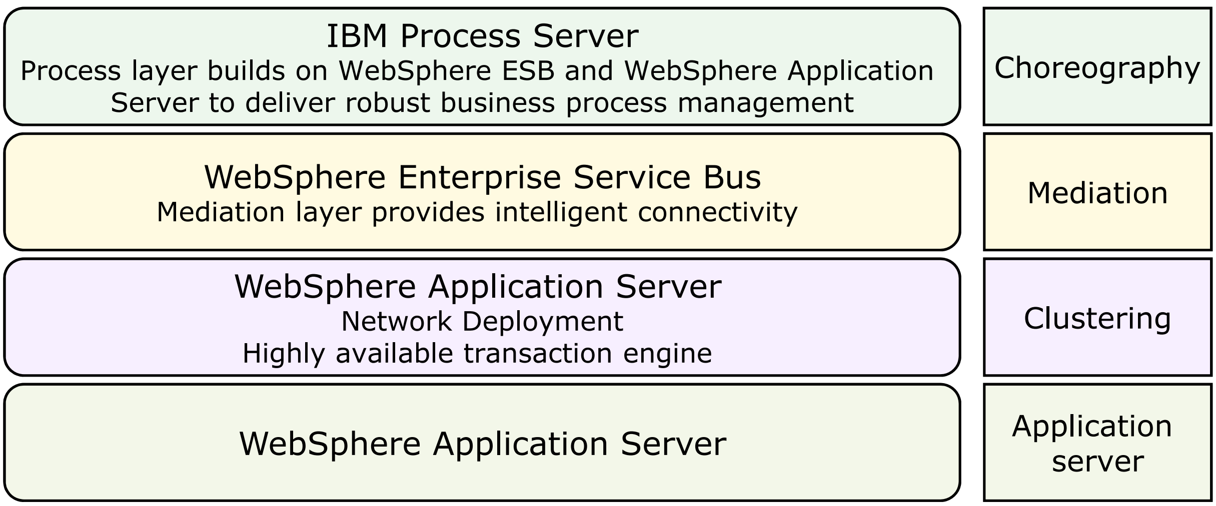


Figure 1-19. IBM Process Server: foundation WB868 / ZB8681.0

Notes:

At the lowest level, IBM Process Server is based on the WebSphere Application Server product. WebSphere Application Server implements the Java EE standard and allows applications that are written on top of it to be portable and efficient. To IBM, Java EE is a platform-neutral operating system that provides all the richness of functions that are needed to build and run applications. Users of Java EE can divest themselves from concerns of security, transaction support, resource management, and much more, and leave those functions to the Java EE environment. In this way, programmers can focus on the intended business functions.

Although IBM Process Server is itself implemented on top of WebSphere Application Server, Java EE skills are not required to design and implement SOA solutions in IBM Process Server. IBM Process Server provides a higher level of abstraction, hiding its own implementation details.

Each layer encapsulates and builds on the lower layer. Everything begins with the application server. The higher abstraction layers indirectly use the application server or WebSphere Application Server Network Deployment for security, user registry, transactions, scalability, clustering, high availability, failover, platform messaging, and automated deployment.

WebSphere Enterprise Service Bus adds support for ESB service integration points, message mediation flows, and central management of integration logic and integration resources. IBM Process Server adds the capabilities of business process development and choreography.

WebSphere Education

Standard versus Advanced edition key capabilities (1 of 2)

Capability	Advanced edition	Standard edition
WebSphere Lombardi Edition compatible execution	X	X
Process Designer – BPMN	X	X
Coach user interface	X	X
Process Portal	X	X
Reporting	X	X
Performance Data Warehouse	X	X
Process Center shared asset repository	X	X
Unlimited authors and users	X	X

Figure 1-20. Standard versus Advanced edition key capabilities (1 of 2) WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Standard versus Advanced edition key capabilities (2 of 2)

Capability	Advanced edition	Standard edition
WebSphere Process Server compatible execution	X	
Integration Designer – BPEL / SOA	X	
Transaction	X	
Adapters	X	
Business Space	X	
ESB	X	
Cluster	X	X
Advanced operating system support – Linux on System z, IBM AIX, Solaris	X	X

Figure 1-21. Standard versus Advanced edition key capabilities (2 of 2) WB868 / ZB8681.0

Notes:

WebSphere Education

Business processes in IBM Business Process Manager

- Business processes can be captured in IBM Process Designer or in IBM Integration Designer
- IBM Process Designer:
  - Captured as business process
  - Represented as business process diagram (BPD)
  - Implementations are only captured in one process application or toolkit
  - Limited implementation options
  - One human task client
- IBM Integration Designer
  - Captured as BPEL process
  - Represented with Business Process Execution Language (BPEL)
  - Implementation can span several modules
  - Can be shared in libraries
  - Several implementations options
  - Several human task clients

Figure 1-22. Business processes in IBM Business Process Manager WB868 / ZB8681.0

Notes:

Business processes might be captured in either IBM Process Designer, IBM Integration Designer, or both. There are differences between the two tools, however:

- IBM Process Designer
  - Business processes are captured as business process artifacts based on BPMN (Business Process Model and Notation). They must be built in either a process application or a toolkit.
  - Business processes do not use a standard language, such as BPEL.
  - The only way to share a process is to place it in a toolkit.
  - Implementation options are limited (human tasks, JavaScript services, simple business rules, and others).
- IBM Integration Designer
  - Business processes are captured as BPEL artifacts. They can be built in either a module or a library.
  - They use the standard BPEL with IBM extensions (WS-BPEL).

- They are loosely based on BPMN.
- Business processes might rely on services in other modules through imports and exports, or through libraries.
- BPEL processes in modules might be exposed as other types of services through their bindings (such as web services, WebSphere MQ bindings, and other services).
- There are several types of implementations for a BPEL process, including the full power of Java, integration with JRules, and integration with external services.
- There are several types of human task clients available, including HTML-Dojo in Business Space, JavaServer Faces, and the Business Process Choreographer Explorer.

WebSphere Education

Comparison: IBM Integration Designer and IBM Process Designer

	IBM Integration Designer	IBM Process Designer
Container for integration artifacts	**Module**, which includes: • Integration logic (BPEL processes, human tasks, business rules) • Data and interfaces • Transformations	**Process App**, includes: • Processes (BPD, human tasks, rules) • Data and services
Container for shareable artifacts	**Library**, which includes: • Integration logic • Data and interfaces • Transformations • Web service ports	**Toolkit**, includes: • Processes • Data and services
Container for mediation services	**Mediation module**, which includes: • Mediation flows	N/A

Figure 1-23. Comparison: IBM Integration Designer and IBM Process Designer WB868 / ZB8681.0

Notes:

Modules and libraries contain multiple SCA artifacts that are grouped by type. Libraries are projects that are used to store shared resources and are accessed by adding them to module dependencies. Integration modules provide the business services, and mediation modules provide connectivity logic. Mediation flows and business services are modeled as SCA components. SCA components are wired together in the assembly diagram to form applications.

Process applications and toolkits in IBM Process Designer are analogous to modules and libraries in IBM Integration Designer. Some similarities include:

- Process applications (like modules) are deployed to the server.
- Toolkits (like libraries) are not deployed to the server.
- Process applications contain business process modeling artifacts.
- Process applications have dependencies on any number of toolkits.
- Toolkits have dependencies on other toolkits.

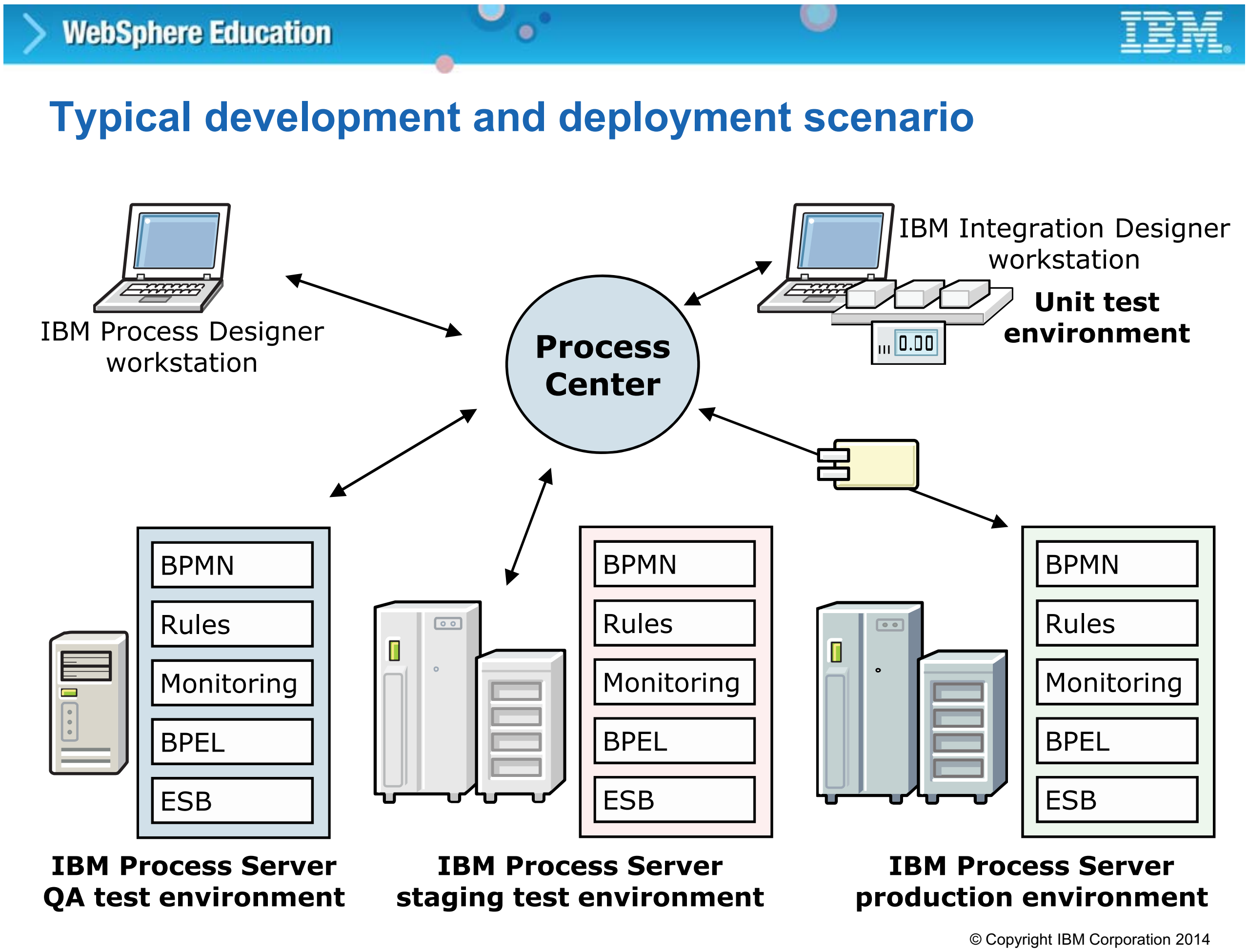


Figure 1-24. Typical development and deployment scenario WB868 / ZB8681.0

Notes:

The diagram shows a typical development and deployment topology.

There is a unit test environment with IBM Integration Designer installed in a UTE environment. In this mode, the unit tester can test SCA modules locally in the IBM Process Server running inside the UTE or test them directly on the Process Center. Either way, it is fine depending on the requirements.

There is also an IBM Process Designer workstation, which communicates directly with the Process Center Console in the center.

There are independent QA, staging, and production environments, each with its own full stand-alone IBM Process Server connected to the central IBM Process Center. Artifacts are being published and synchronized back and forth between all the environments while the IBM Process Center is managing the central repository.

This scenario is just a sample, and the topology can be modified to suit the organization requirements.

1.4. IBM Business Process Manager tools

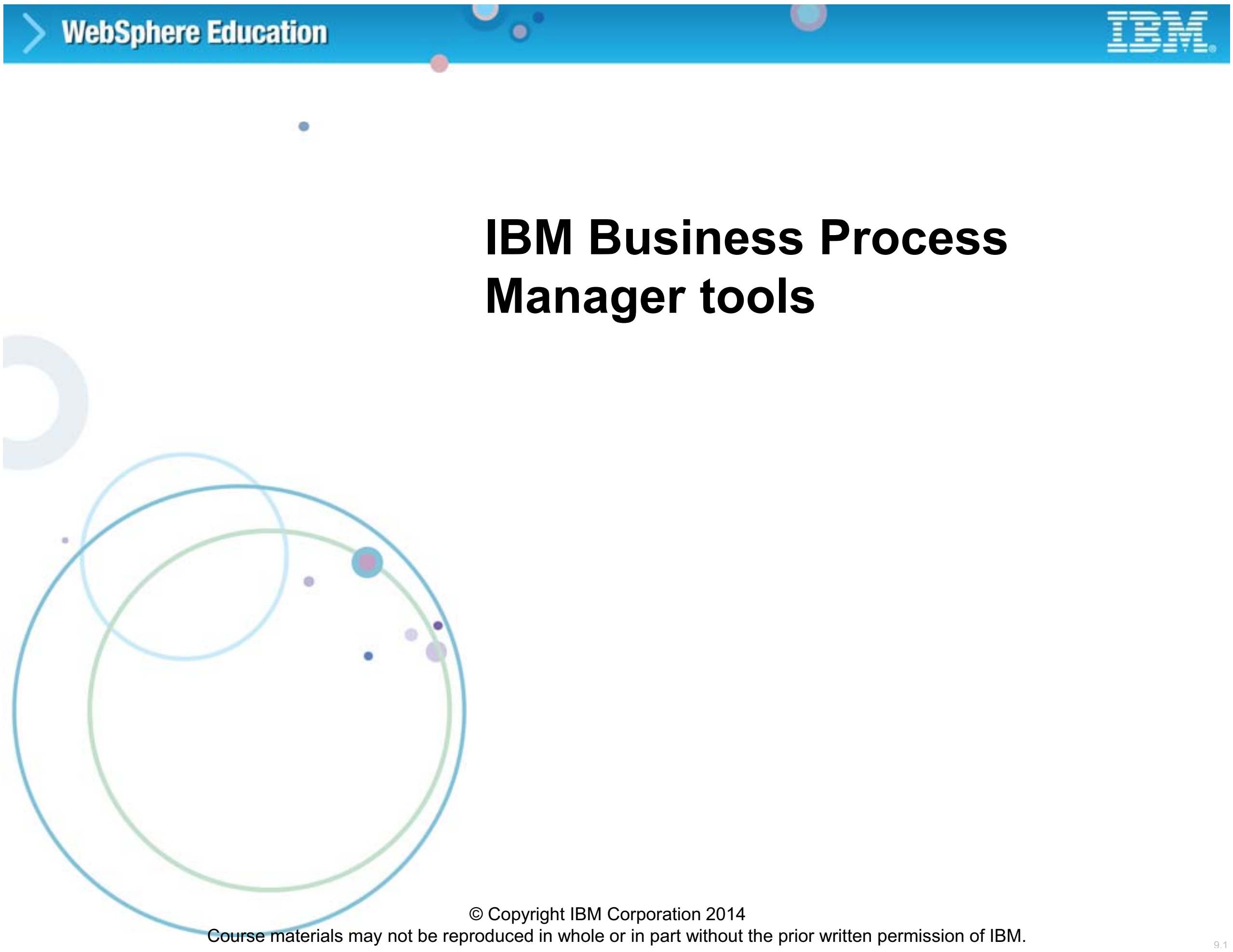


Figure 1-25. IBM Business Process Manager tools WB868 / ZB8681.0

Notes:

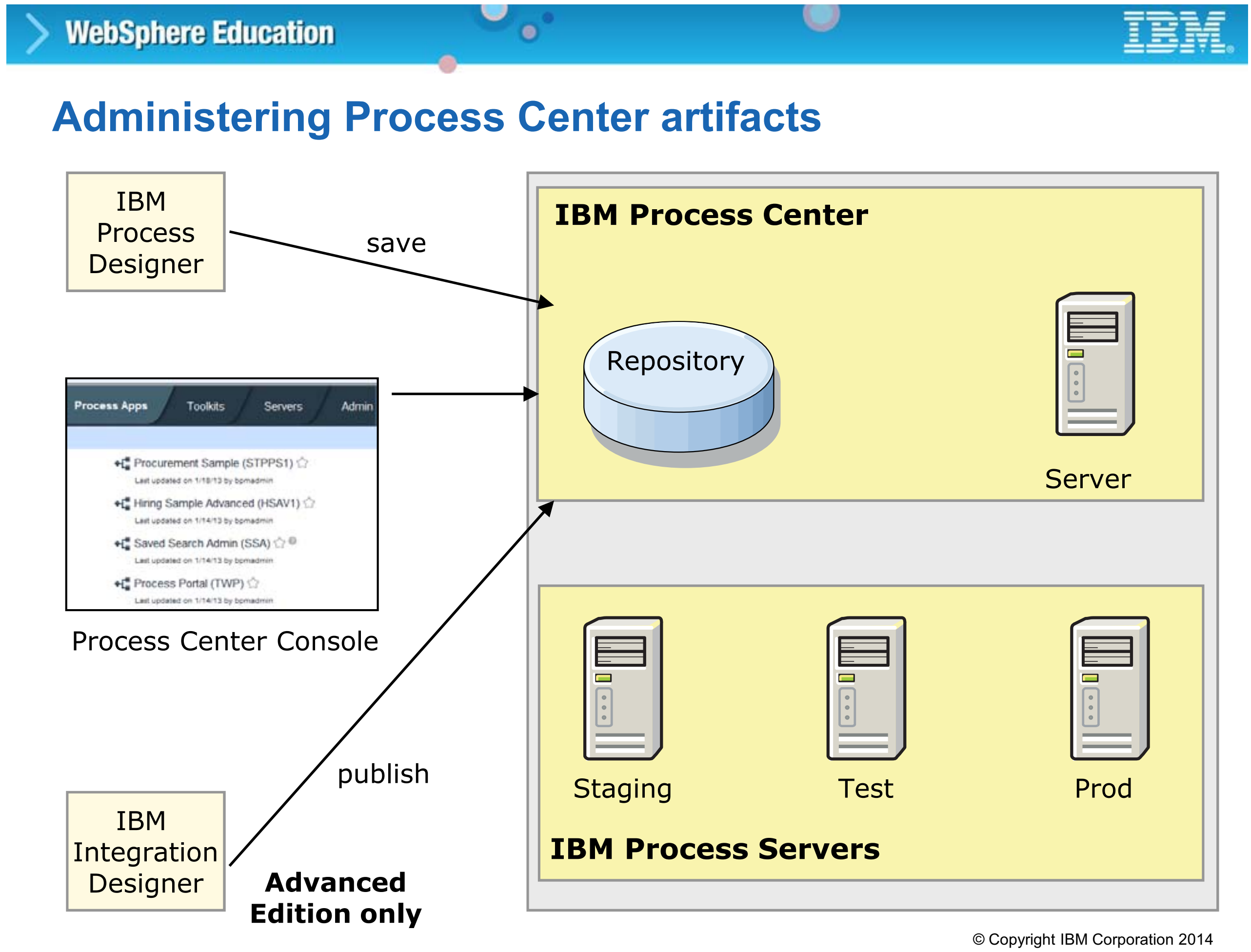


Figure 1-26. Administering Process Center artifacts WB868 / ZB8681.0

Notes:

The Process Center Console is a tool that is intended for administrators and developers who must manage the lifecycle of application components.

Administrators and developers can create, export, clone, activate, or archive snapshots of process applications or toolkits and grant access to these applications.

There are a number of ways to access the Process Center Console.

If you are primarily an administrator and do not actively work on the application development, you can view the Process Center Console by using the web-based Process Center Console.

The Process Center Console is started with the web address `http://hostname:port/ProcessCenter`.

If you are a business analyst and you work on the creation of business process definitions and associated assets, you can view the Process Center Console from inside IBM Process Designer.

If you work as a developer on the Advanced Integration service assets of the process application, you can also view the Process Center Console in a separate perspective in IBM Integration Designer.

WebSphere Education

IBM Process Center console capabilities

- Intended for administrators and integration developers
- Used for managing the lifecycle of application components from development to testing to production
  - Snapshots of process applications and toolkits
- Create, export, clone, activate, or archive snapshots or toolkits
- Grant access to integration developers for authoring in Process Designer or Integration Designer

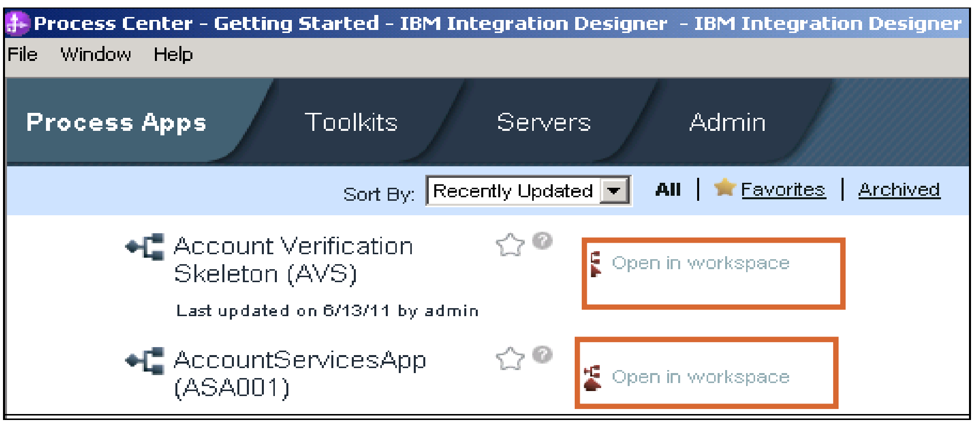


Figure 1-27. IBM Process Center console capabilities WB868 / ZB8681.0

Notes:

The Process Center Console provides a web-based interface for managing the Process Center maintained projects. The default URL for Process Center Console is `http://localhost:9080/ProcessCenter`.

The Process Center includes a repository for all processes, services, and other assets. The Process Center Console provides the tools that you need for maintaining the repository.

From the Process Center Console:

- You can create process applications and toolkits and grant other users access to those process applications and toolkits.
- Administrators can install process applications that are ready for testing or production on the Process Servers in those environments.
- Administrators manage running instances of process applications in configured environments.

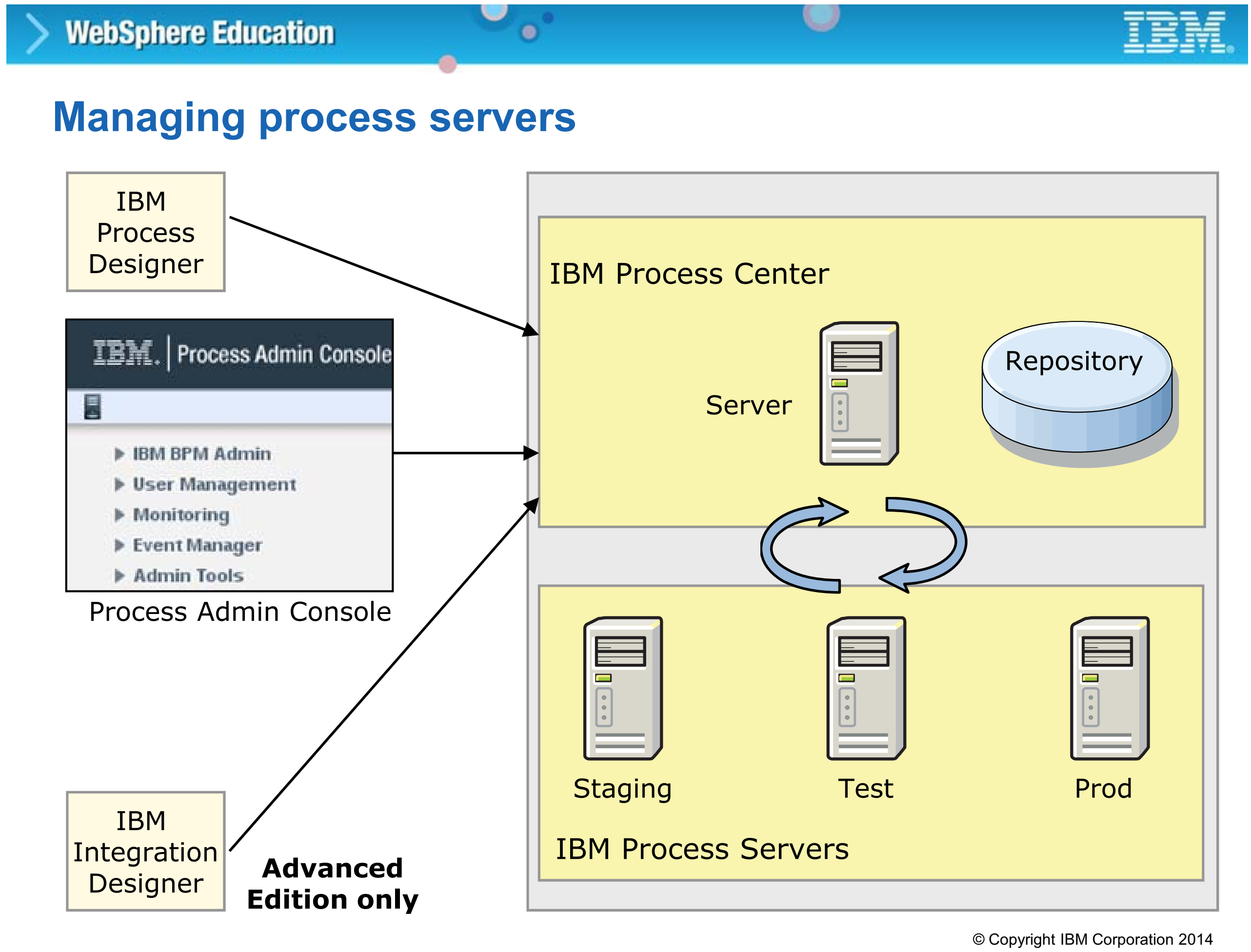


Figure 1-28. Managing process servers WB868 / ZB8681.0

Notes:

You use the Process Admin Console to manage the process servers in your runtime environments, and also the Process Center server that is part of the Process Center.

To access the Process Admin Console, type: `http://hostname:port/ProcessAdmin`

An IBM Process Server on IBM Process Center is used to run process applications on a Process Server that is connected to a Process Center. The server that is created can be a staging server, test server, or production server.

The capability of the Process Center server and any associated process (production) servers must match, both for server registration purposes and for process application snapshot testing and deployment purposes.

To import or test a snapshot on the Process Center server or deploy it on a production process server, the target server must support all of the functionality in the process application. For example, you cannot import a process application with Service Component Architecture (SCA) modules unless the Process Center server supports IBM Business Process Manager Advanced.

WebSphere Education

Process Admin Console capabilities

- Intended for administrators
- Manage the various Process Server runtime environments
- Manage snapshots that are deployed to a Process Server
- Includes Process Inspector for viewing and managing process instances
- Manage user accounts

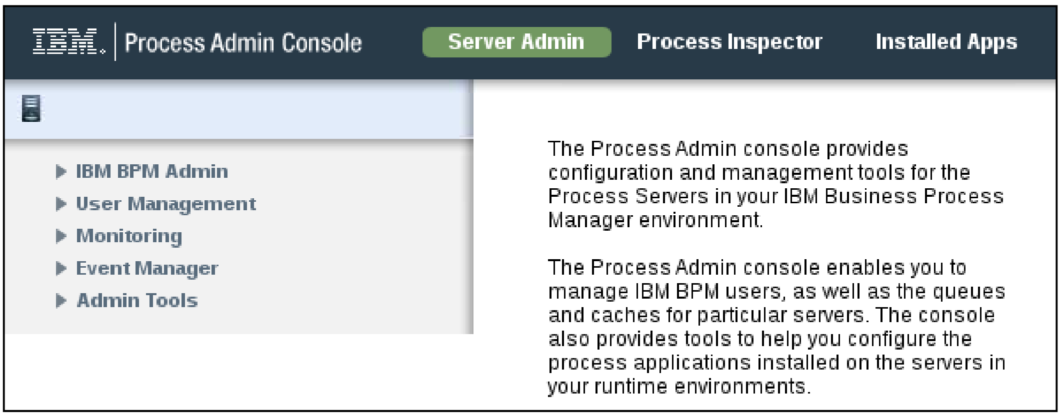


Figure 1-29. Process Admin Console capabilities WB868 / ZB8681.0

Notes:

The Process Admin Console allows administrators to manage the Process Servers in the runtime environments (staging, test, production). It is also available to manage the Process Center server that is part of the Process Center.

The most important management tasks are managing user accounts (creating and managing applications and participant groups) and managing installed applications (activating and deactivating applications, migrating in-flight instances).

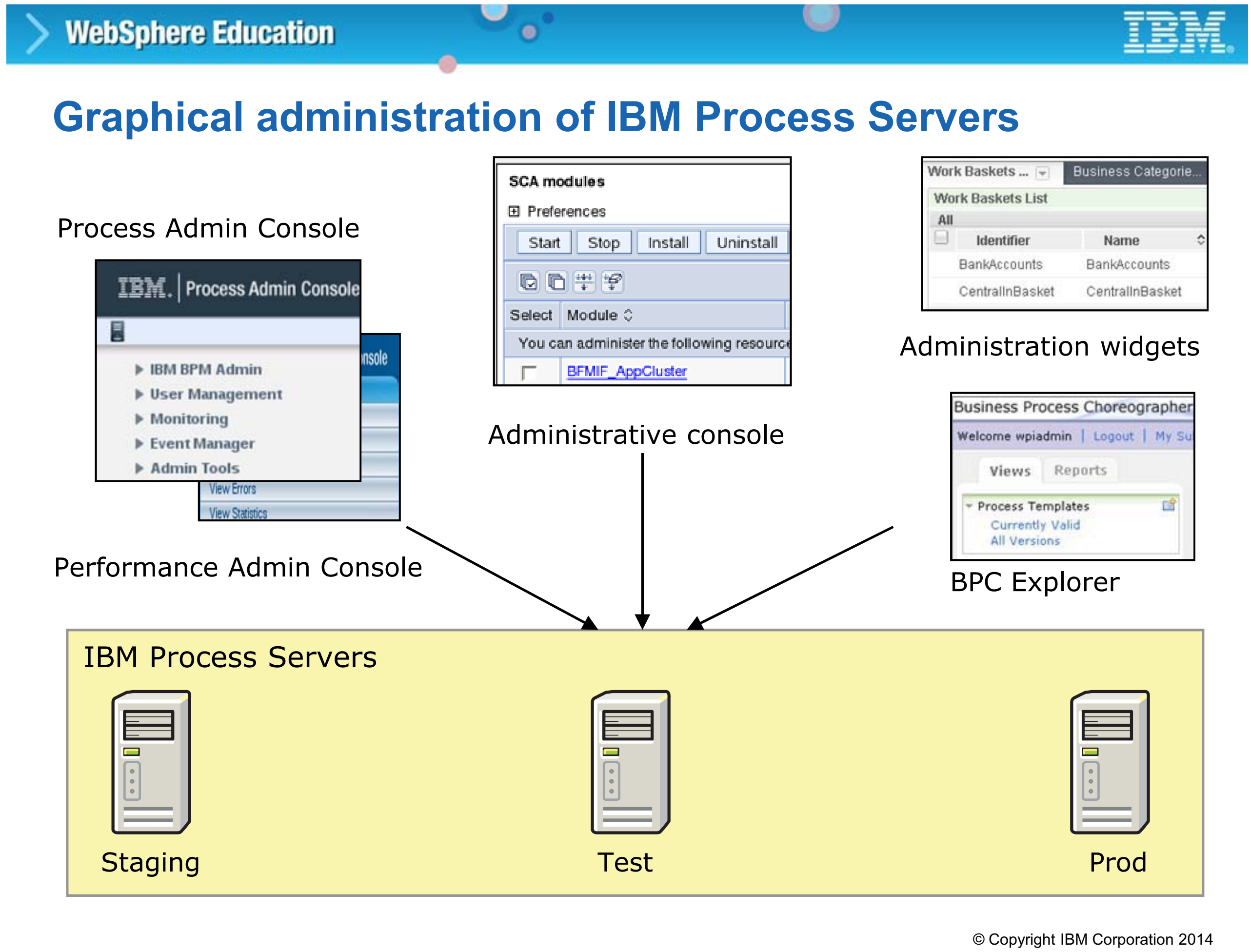


Figure 1-30. Graphical administration of IBM Process Servers WB868 / ZB8681.0

Notes:

This page shows some of the graphical tools that are used to administer IBM Process Servers.

These graphical administration tools are covered in greater detail in later units.

WebSphere Education

Performance Admin Console capabilities

- Collects performance data that represents key business events and metrics as processes are run
- Uses the process model to correlate the business events in real time and aggregate raw performance data into a single database view for reporting and auditing

Figure 1-31. Performance Admin Console capabilities WB868 / ZB8681.0

Notes:

The Performance Data Warehouses in your IBM Business Process Manager configuration retrieve and store tracked performance data, which allows users in IBM Process Designer to create reports and also analyze processes by using the Optimizer.

As part of system maintenance, you might want to view the Performance Data Warehouse load queue to determine which records have yet to be loaded to the database. You might also want to view the error queue to determine whether any errors occurred while data was being loaded from the Process Server to the Performance Data Warehouse.

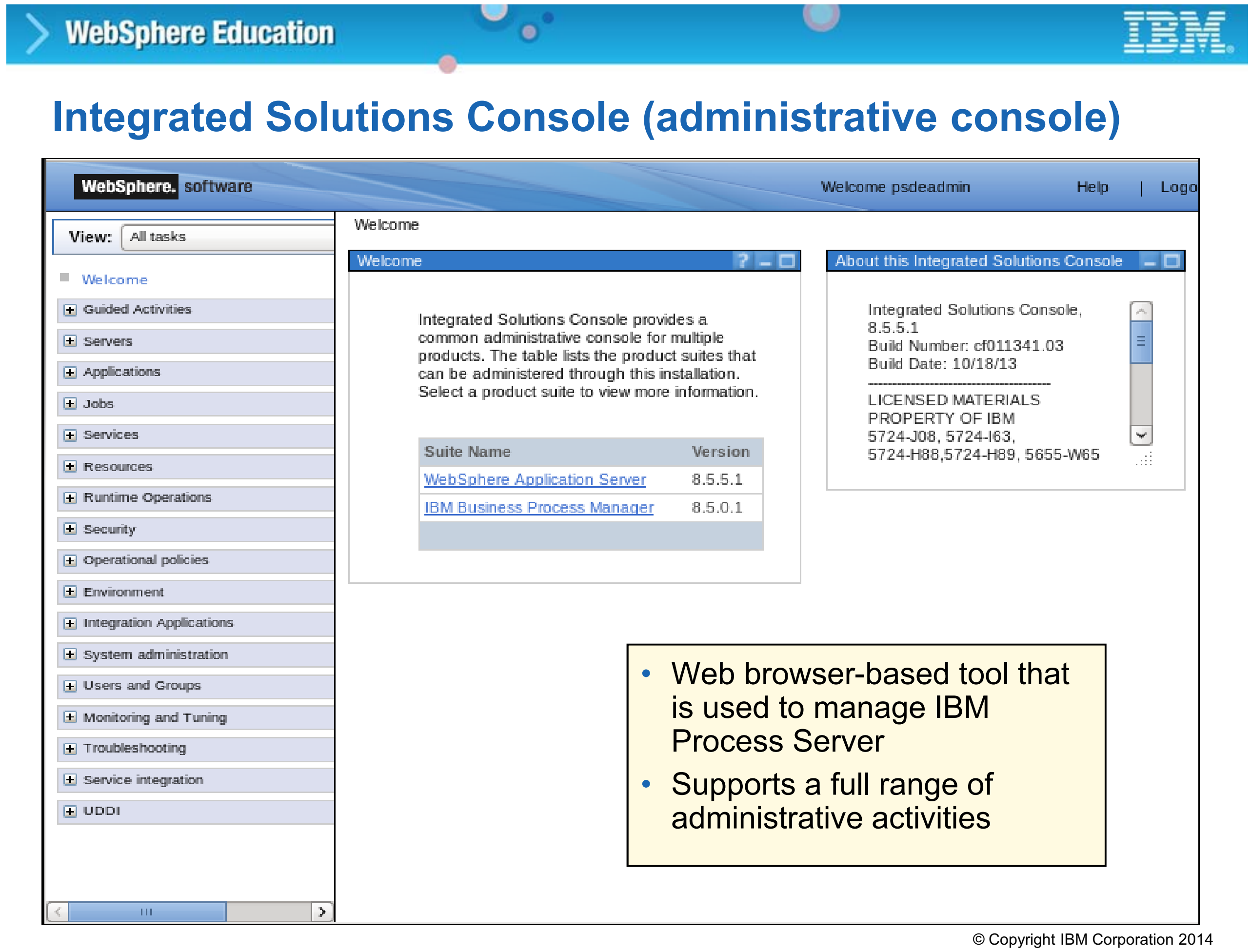


Figure 1-32. Integrated Solutions Console (administrative console) WB868 / ZB8681.0

Notes:

WebSphere Education

IBM

Process Portal

- Provides the primary graphical user interface for users to work with assigned tasks and other to complete work efficiently
- Process Portal is redesigned to provide a highly collaborative work experience with increased social capabilities
- Dashboards to help you visualize status data for one or more business processes

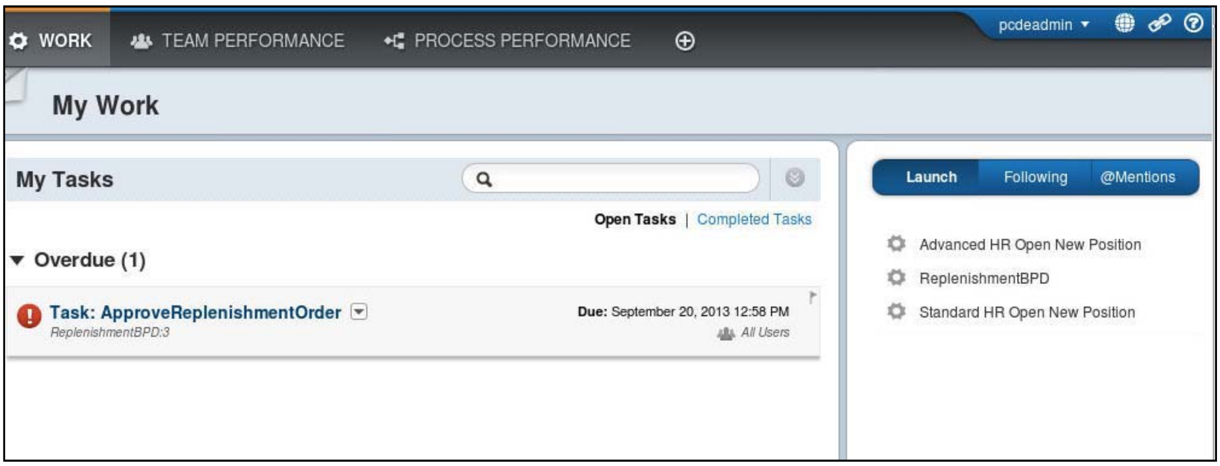


Figure 1-33. Process Portal WB868 / ZB8681.0

Notes:

The Process Portal was new in Business Process Manager Advanced V7.5. It is a highly collaborative work experience with features for managing an online conversation with experts and for subscribing to specific process instances. There is a built-in notification system, and a record of the conversations is stored in the business process “stream.” The stream is also a place where you and the experts can review all the steps that are previously completed and are currently active. You can also include supporting documentation as attachments to the business process streams.

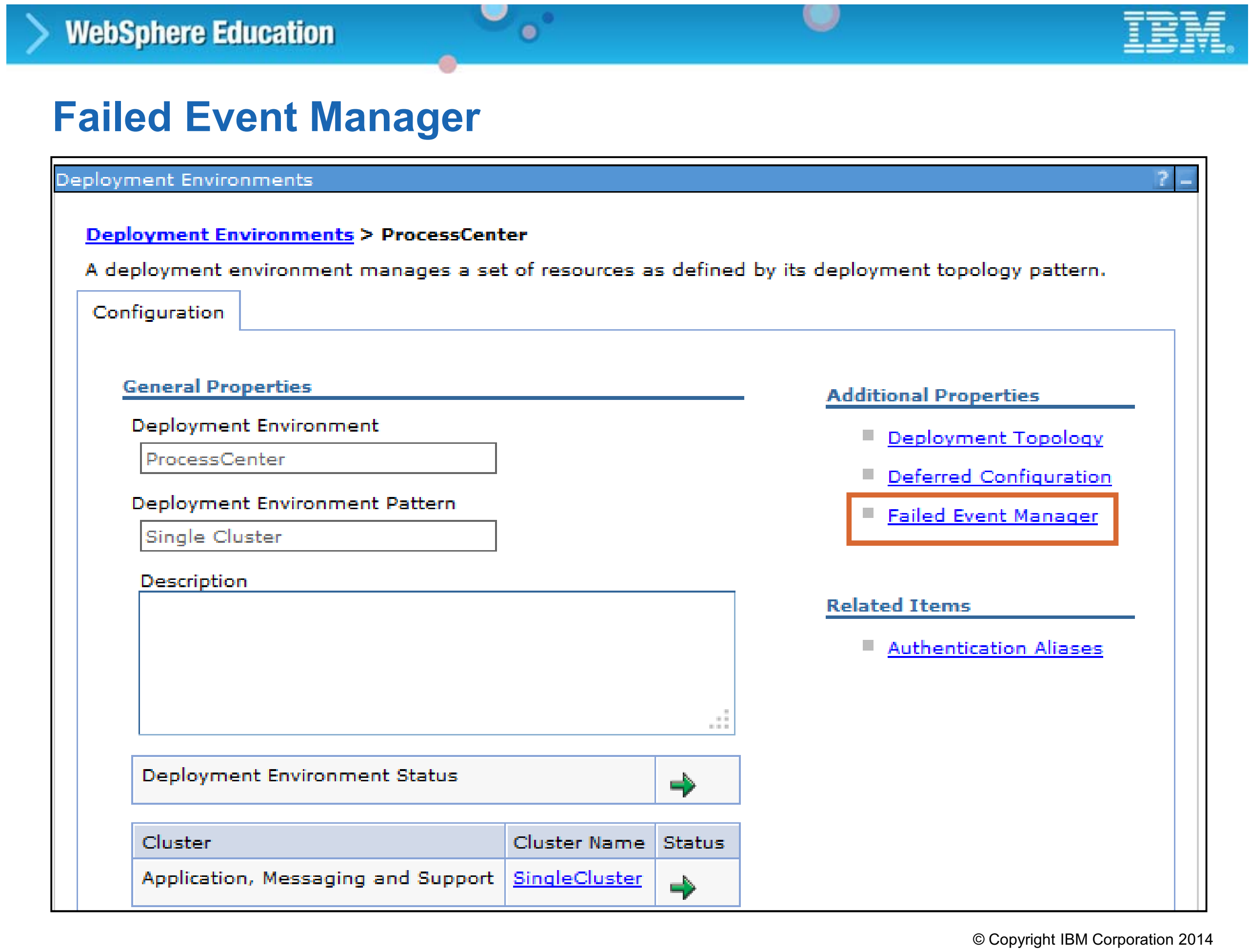


Figure 1-34. Failed Event Manager WB868 / ZB8681.0

Notes:

The Failed Event Manager is available from the administrative console under your deployment environment.

WebSphere Education

Business Process Choreographer Explorer client

- Web application that implements a user interface for interacting with business processes and human tasks
  – Built with reusable, customizable JavaServer Faces (JSF) components
- As administrator, view information about process and task templates, process instances, task instances, and associated objects
- As a user, view and act only on tasks that are assigned to you

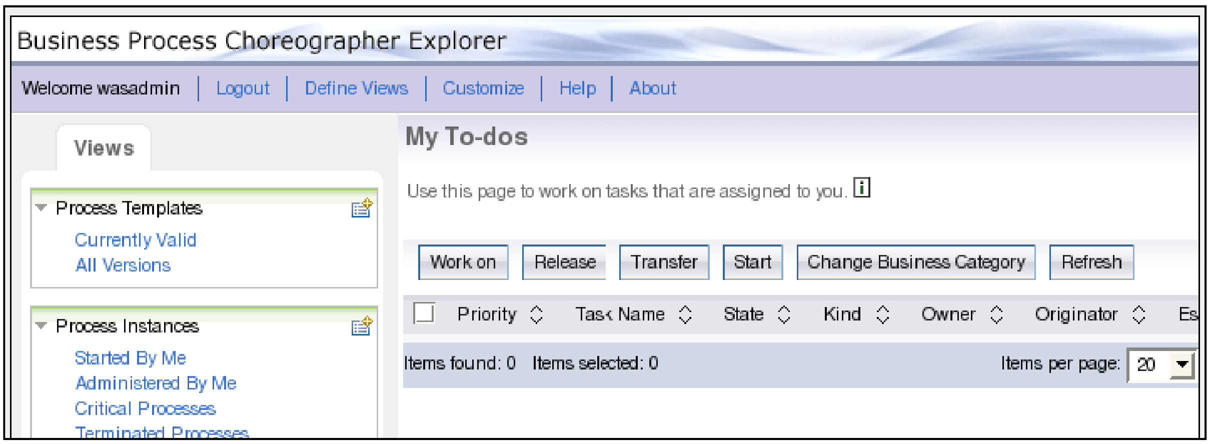


Figure 1-35. Business Process Choreographer Explorer client WB868 / ZB8681.0

Notes:

Business Process Choreographer Explorer is a web application that can be installed as part of the configuration of the business process container. Before you can use Business Process Choreographer Explorer from a web browser, you must install the business process container, human task container, and the Business Process Choreographer Explorer application. The event collector application must be installed and running before you can use the reporting function.

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

1.5. IBM Business Process Manager components

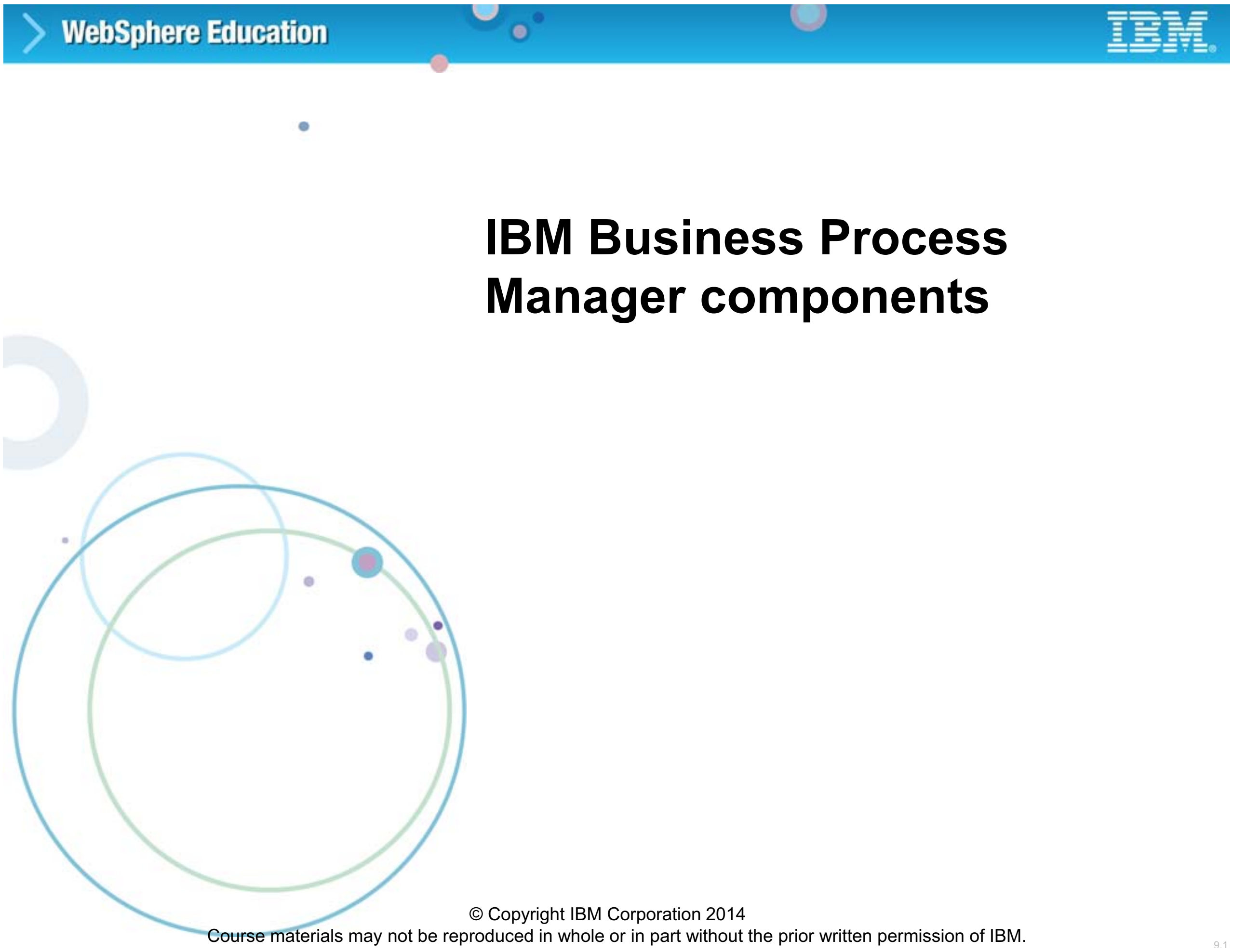


Figure 1-36. IBM Business Process Manager components WB868 / ZB8681.0

Notes:

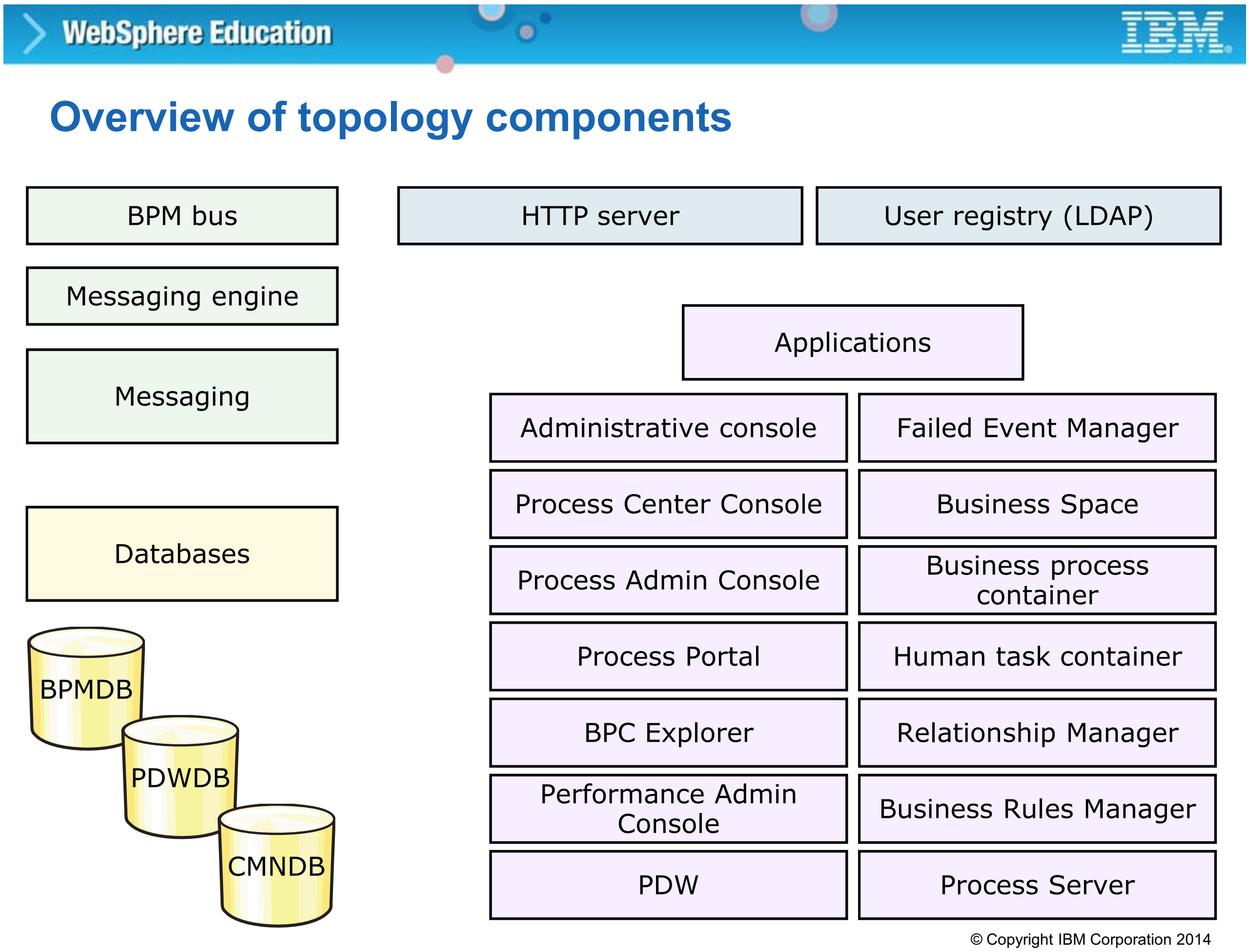


Figure 1-37. Overview of topology components WB868 / ZB8681.0

Notes:

Before examining the key cluster topologies, a review of the components is useful. Every IBM BPM environment has three fundamental layers:

- Application: The application layer comprises the business process, the Process Server components to support them, and the administration functions (administrative console, Business Process Choreographer Explorer, and applications like the Failed Event Manager).
- Database: A production-ready, relational database system is required (Derby does not support production environments). Process Server requires access to certain application configuration and runtime information that is stored in relational database tables. IBM Process Server requires the database layer, but it is managed outside IBM Process Server. In general, you might configure five or more databases, depending on how you intend to provide adequate performance and scalability.
- Messaging: IBM Process Server also requires a messaging infrastructure that consists of a messaging engine, buses, and queues. Clustering the messaging infrastructure is the most complex aspect of the high availability environment. In general, since you use WebSphere Platform Messaging (service integration bus), the messaging infrastructure is also clustered by

using WebSphere clustering techniques. However, you must understand a number of considerations when you select a topology.

Depicted in the upper right corner of the graphic are more components that run outside Process Server but might be needed in most Process Server environments. Load balancers and HTTP servers are used to process web requests and direct them to cluster members, which are able to do the work (workload balancing). For a clustered environment, the user registry must be LDAP or a custom registry because distributed environments cannot use a local operating system as a user registry.

WebSphere Education IBM

Databases support in a topology

- Databases and schemas are different
- Database schemas share a database
  - Messaging, Business Space, and Process Server (BPC) are assigned to database schemas
  - Process Server and Performance Data Warehouse do not have schema support; therefore they cannot share a database
- The three databases that are required to support your topology:
  - Common database: **CMNDB**
  - Process Server database: **BPMDB**
  - Performance Data Warehouse: **PDWDB**
- The common database is split into two pieces:
  - Cell scoped tables per the entire cell
  - Deployment environment scoped tables per the deployment environment

Figure 1-38. Databases support in a topology WB868 / ZB8681.0

Notes:

To plan your database configuration, you must know which databases must be in place and configured to use the software. You must also know which components of IBM Business Process Manager you use and their associated databases. Also, you must know about the tasks that are required to administer the databases, and the security privileges of the database system that you are using.

This slide lists the databases and their purpose within an IBM BPM infrastructure.

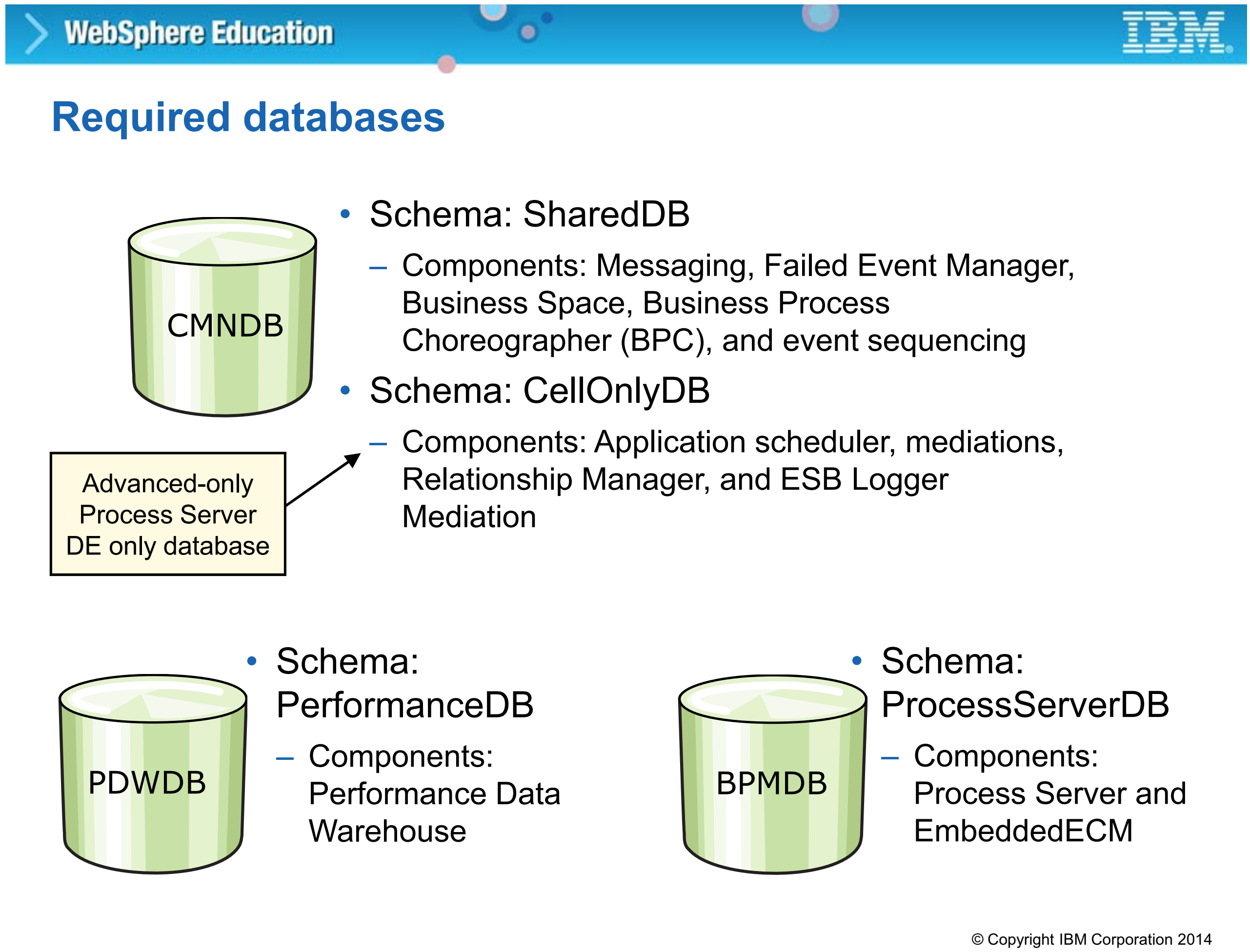


Figure 1-39. Required databases WB868 / ZB8681.0

Notes:

The Process Server and Performance Data Warehouse components do not support case-sensitive databases. These databases must *not* be case-sensitive.

For Microsoft SQL Server and Oracle databases, the following restrictions apply:

- For Microsoft SQL Server databases, components other than Process Server or Performance Data Warehouse require that their databases be case-sensitive.
- For Oracle databases, the Process Server, Performance Data Warehouse, and Common database components must use a separate schema or user. They can use the same instance.

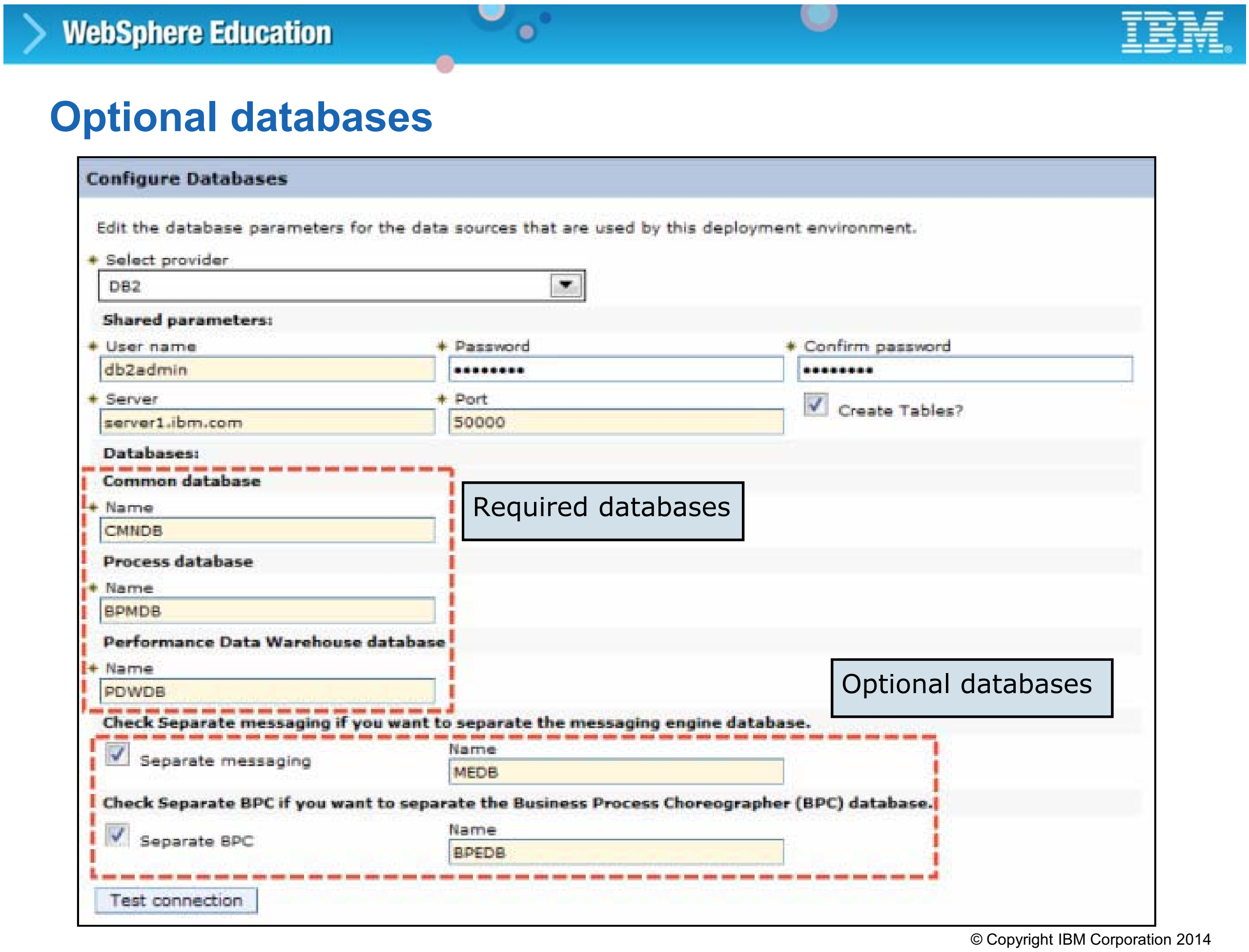


Figure 1-40. Optional databases WB868 / ZB8681.0

Notes:

WebSphere Education

IBM BPM Bus

- A service integration bus is a managed communication mechanism that supports service integration through synchronous and asynchronous messaging
- A single service integration bus and single messaging engine that use the same database schema as the product database by default
- Each deployment environment has its own bus
- The single bus is called `BPM.`*`deployment_environment_name`*`.Bus`

Figure 1-41. IBM BPM Bus WB868 / ZB8681.0

Notes:

The SIBus provides the following capabilities:

- Any application can exchange messages with any other application by using a destination to which one application sends, and from which the other application receives.
- A message-producing application, that is, a producer, can produce messages for a destination regardless of which messaging engine the producer uses to connect to the bus.
- A message-consuming application, that is, a consumer, can consume messages from a destination (whenever that destination is available) regardless of which messaging engine the consumer uses to connect to the bus.

The bus supports the following types of messaging:

- Sending messages synchronously requires the consuming application to be running and reachable.
- Sending messages asynchronously (possible whether the consuming application is running or not and if the destination is reachable). Both point-to-point and publish/subscribe messaging are supported.

- Publishing events or other notifications. The bus can also generate notification messages.

WebSphere Education

Components

- Process Server runtime
  - Where process applications are deployed and made available to run and manage process tasks
- Process Center server
  - Includes a repository for all process applications, business process diagrams, service components, and other assets that are created in the BPM authoring environments

Figure 1-42. Components WB868 / ZB8681.0

Notes:

Each IBM Business Process Manager Process Server contains resources for running processes that are *installed* from Process Center. Every Process Server deployment environment contains a Process Server runtime, which runs the process applications that are installed from the Process Center repository.

WebSphere Education

Business Process Choreographer

- Business process container (Business Flow Manager)
  - To run applications that contain business processes on a Process Server, you must configure the necessary resources and install the business process container application
  - The business process container is installed in the Application cluster
- Human task container (Human Task Manager)
  - To run applications that contain human tasks on a Process Server, you must install the human task container
  - The human task container is installed in the Application cluster
- Deployed as four enterprise applications
- Advanced edition

Figure 1-43. Business Process Choreographer WB868 / ZB8681.0

Notes:

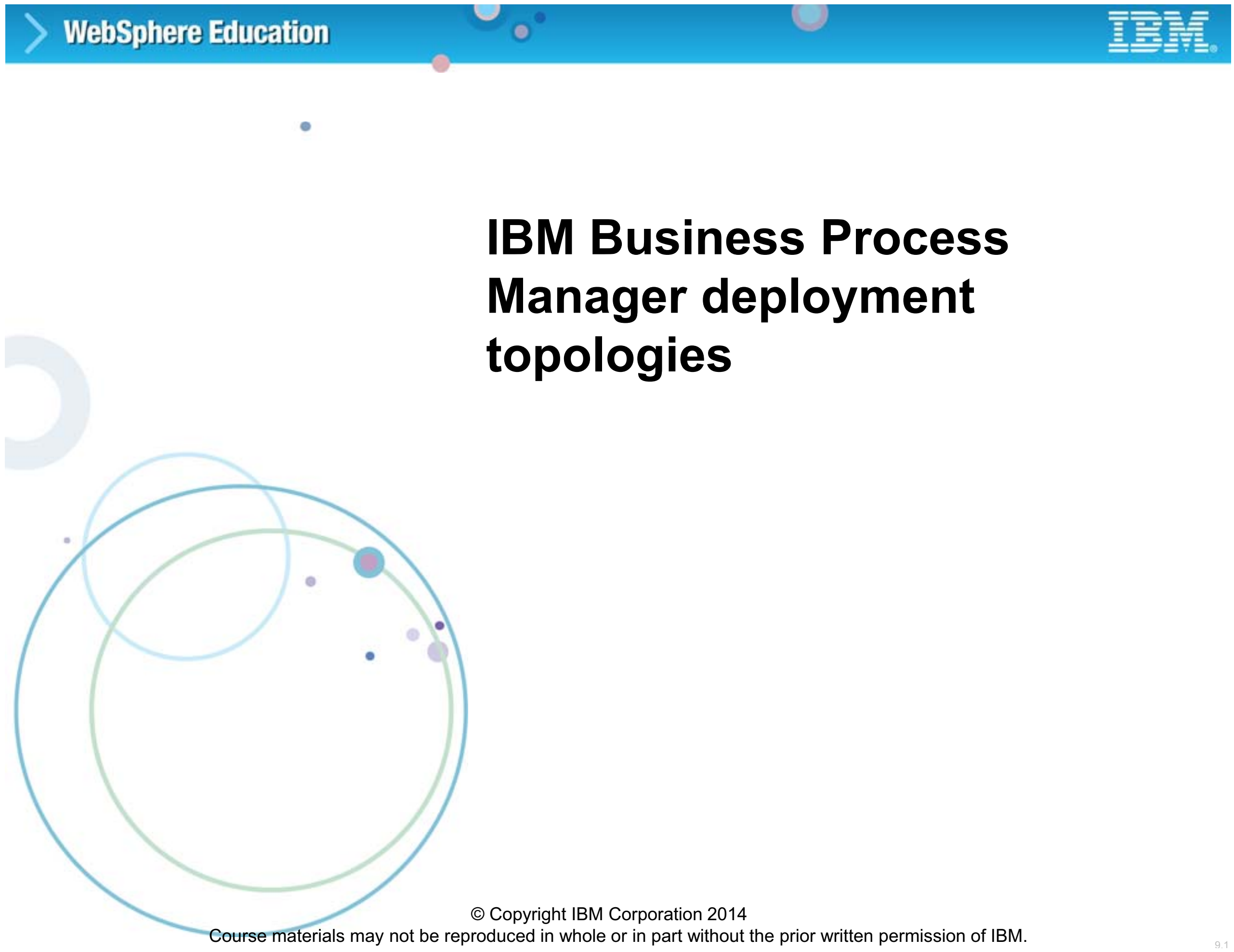


Figure 1-44. IBM Process Server deployment topologies WB868 / ZB8681.0

Notes:

WebSphere Education

Selecting a topology

- Selecting a topology depends on a number of factors, including:
  - Available hardware resources and operating system
  - Types of business processes that you plan to implement
  - Resource requirements and constraints
  - Scalability requirements
  - Administrative effort that is involved
  - Proof-of-concept (POC), testing, demonstration, or a fully functional production environment
- Consider the advantages and disadvantages of each topology pattern
  - Application, Remote Messaging, and Remote Support is the preferred pattern for IBM Business Process Manager Advanced
- The design characteristics of each topology are captured as *topology patterns* that are supplied as configuration templates with the product

Figure 1-45. Selecting a topology WB868 / ZB8681.0

Notes:

A topology is the physical layout of the deployment environment. You can create the topology that best addresses your business needs by choosing an IBM-provided pattern or by creating your own customized pattern.

Selecting an appropriate topology for your deployment environment depends upon several factors. When you select a topology pattern, consider these factors:

- Available hardware resources
- Application invocation patterns
- Types of business processes that you plan to implement (interruptible versus non-interruptible)
- Individual scalability requirements
- Administrative effort that is involved

The IBM-supplied topologies can be applied to both Process Server and Process Center topologies. Therefore, Process Center and Process Server network deployment environments can be organized in a similar way.

The procedures for creating environments for Process Server and Process Center that are based on IBM-supplied topologies are also similar. The only difference that is related to IBM-supplied patterns consists of the suggested patterns for a production environment, and the components that are configured on the clusters for those patterns.

WebSphere Education

Deployment environment types (1 of 2)

- After product installation, you can create a network deployment configuration that is based on the topology pattern templates that are packaged with the software
- The following template types are included:
  - Advanced Process Center deployment environment
  - Standard Process Center deployment environment
  - Advanced Process Server deployment environment
  - Standard Process Server deployment environment
  - Advanced-only Process Server deployment environment
- There are various methods to create deployment environments
  - Multiple tools to create profiles and deployment environment
  - `BPMConfig` command

Figure 1-46. Deployment environment types (1 of 2) WB868 / ZB8681.0

Notes:

Setting up a network deployment environment involves many decisions, such as the number of physical workstations and the type of pattern you choose. Each decision affects how you set up your deployment environment. When you plan the layout of interconnected servers, you must make decisions. These decisions influence trade-offs that you make between the available hardware and physical connections, the complexity of the management and configuration, and requirements such as performance, availability, scalability, isolation, security, and stability.

You can configure a standardized network deployment environment that is based on a topology pattern template included with the software, and you can implement it using the BPMConfig command or the Deployment Environment wizard.

WebSphere Education IBM

Deployment environment types (2 of 2)

- Advanced Process Center deployment environment
  - Run and administer process applications and toolkits that are developed in Process Designer and Integration Designer
- Standard Process Center deployment environment
  - Run and administer process applications and toolkits that are developed in Process Designer
- Advanced Process Server deployment environment
  - Run processes, services, and modules that are deployed from Process Center
- Standard Process Server deployment environment
  - Run processes that are deployed from Process Center
- Advanced-only Process Server deployment environment
  - Run SCA modules that are created in Integration Designer
- Multiple deployment environments per cell are supported
  - For Process Server deployment environments only

Figure 1-47. Deployment environment types (2 of 2) WB868 / ZB8681.0

Notes:

The following options are available based on your installation:

For IBM Business Process Manager Advanced:

- **Advanced Process Center**: For storing, testing, and administering process applications and toolkits that are authored in Process Designer and Integration Designer.
- **Advanced Process Server**: For running processes, services, and modules that are contained in process applications that are deployed from the Advanced Process Center, or for running SCA modules that are deployed directly.
- **Standard Process Center**: For storing, testing, and administering process applications and toolkits that are authored in Process Designer.
- **Standard Process Server**: For running processes and services that are contained in process applications that are deployed from the Standard Process Center.
- **Advanced-only Process Server**: For running SCA modules only. You deploy these modules from the command line or the administrative console. This server is the IBM Business Process Manager equivalent of IBM WebSphere Process Server.

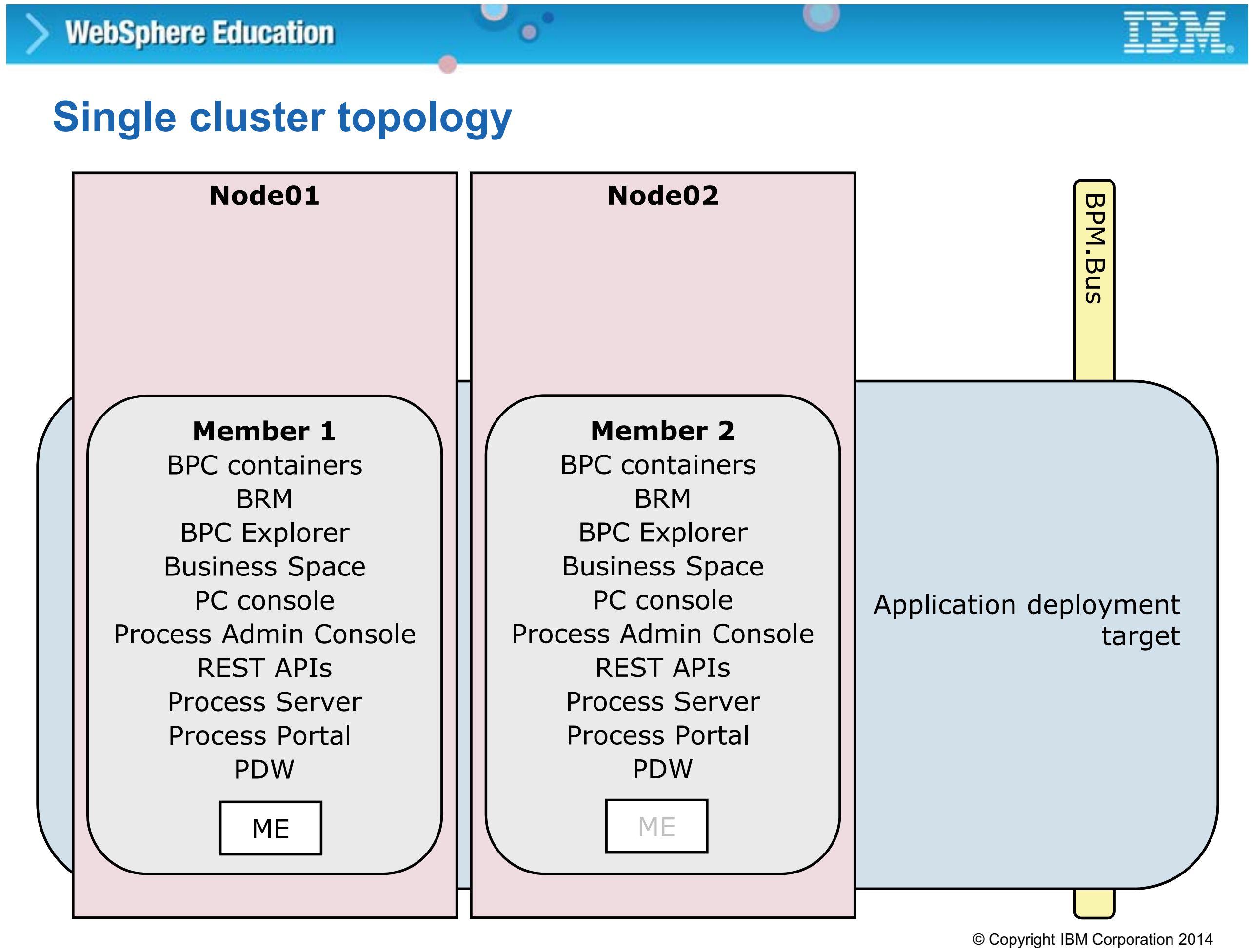


Figure 1-48. Single cluster topology WB868 / ZB8681.0

Notes:

Use this legend to identify components in each topology:

- BPC: Business Process Choreographer
- PC: Process Center
- BRM: Business Rules Manager
- PDW: Process Data Warehouse

Typically, this topology is used for testing, proofs of concept, and demonstration environments. Note the following aspects of this example:

- All of the components are configured in a single cluster.
- The Application deployment target cluster is a member of the BPM bus. The Business Process Choreographer is configured in the cluster, so each cluster member has a business process container and a human task container. All of the supporting infrastructure applications are configured in the cluster: the Business Process Choreographer Explorer, the Business Process Choreographer reporting function, the Business Rules Manager, the Common Event Infrastructure, and Business Space. Each cluster member is an application deployment target.

Cluster member 1 has the active messaging engine. Cluster member 2 has a joined messaging engine.

The behavior of the messaging engines in a single cluster topology is different from the behavior that occurs when the messaging engines are in a remote cluster. When the messaging engines and the applications are collocated, the default behavior is for message producers and consumers to always use a local active messaging engine if one is available. For example, there might be a situation in which two applications communicate asynchronously and are deployed to each cluster member. After each message producer places messages in the queues, the message consumer on the computer where the engine is local always consumes all of the messages that are produced. Thus, the consuming application processes messages on only one server, which is 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, and the active messaging engines of each server contain a portion of the queues that are assigned to that engine. Thus, you can attain more throughput if there are active messaging engines on each server. However, this configuration can create issues for your applications.

If you partition destinations when the applications and messaging engines are in the same cluster, you no longer can maintain message order. In addition, partitioned destinations can create unpredictable behavior if one or more messaging engines fail in a single cluster topology. If you are prepared to endure possible unpredictable behavior and the loss of message order, partitioning the destinations in a single cluster topology might be acceptable. However, this configuration is discouraged.

WebSphere Education

Decision criteria

- Advantages:
  - Can run on limited hardware (fewer computers)
  - Easier to set up and administer
  - Scalability is easy: add new cluster members or new nodes
  - Requires more hardware resources, but less than a full remote messaging and remote support topology

- Disadvantages:
  - Memory requirements are much greater
  - Performance tuning is much more critical
  - Extensive use of the messaging infrastructure interferes with application processing as they are in the same cluster
  - Infrastructure components cannot be scaled independently; components are scaled at the same rate
  - Adding a cluster member adds capability to every component

Figure 1-49. Decision criteria WB868 / ZB8681.0

Notes:

From an administrative and scalability perspective, the Single Cluster topology pattern has advantages and disadvantages. A single cluster, where each member runs all the IBM Business Process Manager components, is simpler to administer. Instead of several server instances in multiple clusters, you have a single cluster with fewer members. If the needs of your environment grow, scaling the infrastructure is a simple matter of adding more nodes and cluster members. Thus, the process of adding capability is simple, but all components are scaled at the same rate. For example, each additional cluster member adds component processing whether you need it or not. If the messaging engines spread across server members use policies, there might be some additional administrative effort in creating and maintaining the policies.

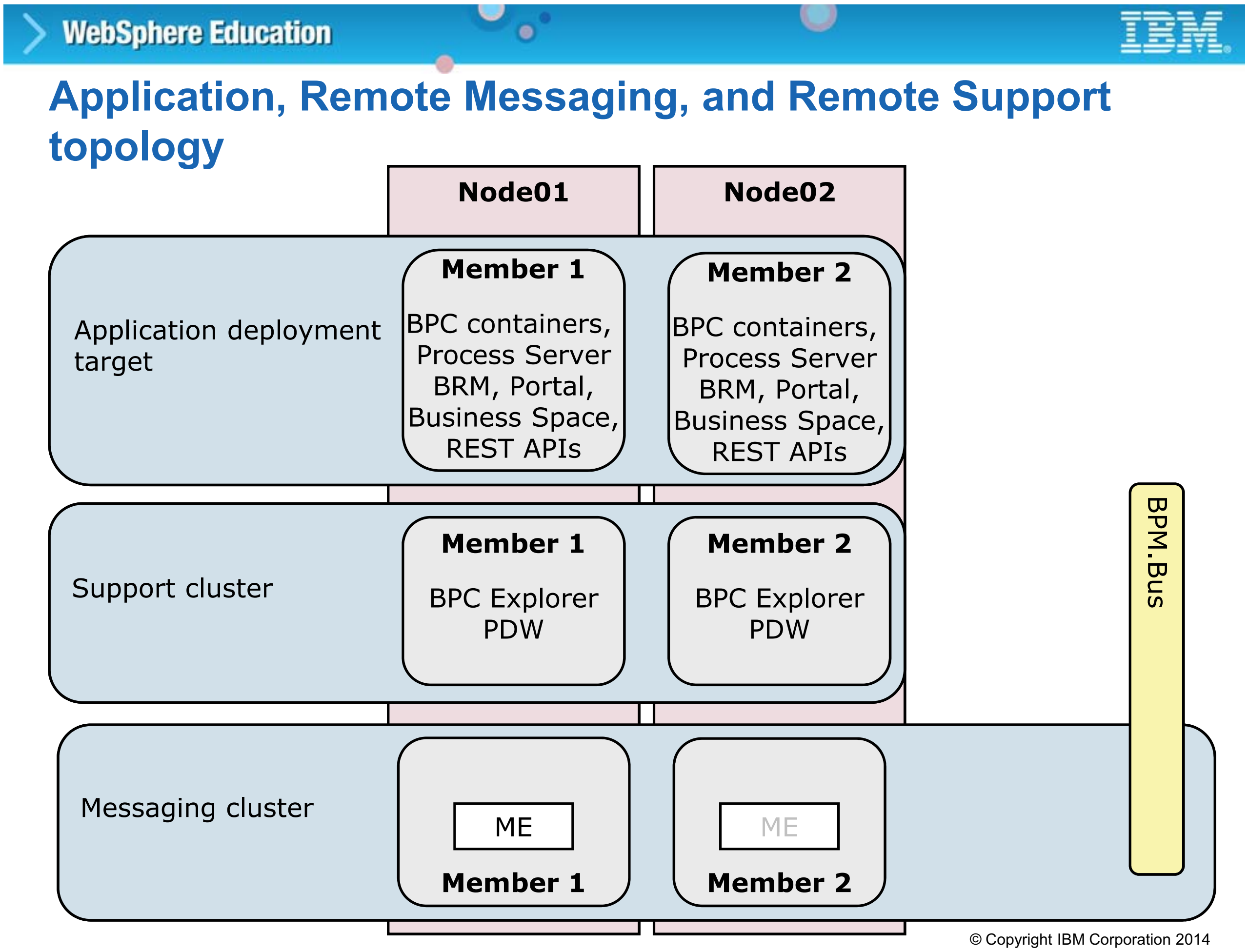


Figure 1-50. Application, Remote Messaging, and Remote Support topology WB868 / ZB8681.0

Notes:

The Application, Remote Messaging, and Remote Support topology pattern is a topology for production environments. This topology provides three separate clusters:

- A remote messaging cluster
- A remote support cluster
- An application deployment target cluster

Note the following aspects of this example:

- All of the applications are deployed to the Application deployment target cluster.
- The Business Process Choreographer is configured in the Application deployment target cluster so each cluster member has a business process container and a human task container.
- The messaging cluster is a member of the required BPM bus.
- Supporting infrastructure applications are configured in the Support cluster: the Business Process Choreographer Explorer and Performance Data Warehouse.

The behavior of the messaging engines in an Application, Remote Messaging, and Remote Support topology is different from the behavior that occurs when the messaging engines are collocated with the applications. Because the messaging engines are in a remote cluster, there is no preference for the message producers and consumers as to using a local messaging engine. Each member of the Application deployment target cluster connects to the appropriate bus and uses the remote messaging engine for that bus.

This behavior creates issues if you attempt to partition the destinations in the remote messaging cluster. When you create more than one active set of messaging engines, partitioning results. The active messaging engines of each server contain a portion of the queues that are assigned to that engine. Thus, you can attain more throughput if there are active messaging engines on each member of the messaging cluster. However, this configuration can create issues for your applications. If you partition destinations when the application and messaging engines are in separate clusters, you no longer can maintain message order. Any time you partition destinations, you lose message order.

In addition, partitioned destinations can create other issues when the messaging engines are remote. By default, you have no control over which active messaging engine your applications use at run time. This behavior can create situations in which two applications on the same server attach to two different messaging engines. If one application produces messages for one engine, and the the message consumer is using a different engine, then stranded messages can result. Thus, partitioned destinations are discouraged in a remote messaging and remote support scenario.

WebSphere Education IBM.

Decision criteria

- Advantages
  - Ideal from a performance perspective as each component can be tuned independently
  - Each component can be scaled independently
  - Good topology for long-running processes, human tasks
  - Independent messaging cluster allows extensive use of asynchronous communication
  - Independent support cluster
- Disadvantages
  - Three clusters, many nodes, many servers
  - Must performance tune all clusters
  - Scalability of messaging is still limited
  - Multiple active messaging engines are not supported
- You must also configure a routing server to ensure that requests that are intended for Process Portal are directed to the correct cluster

Figure 1-51. Decision criteria WB868 / ZB8681.0

Notes:

For large computing infrastructures, the Application, Remote Messaging, and Remote Support topology pattern is the preferred environment. The hardware requirements for distributed platforms are more intensive. However, you have greater flexibility in adjusting and tuning memory usage for the Java virtual machines (JVMs) when you have three or more clusters with multiple members that do specific functions.

When you create three clusters, each with specific functions and applications, you add another administrative burden. As you add clusters and cluster members, your performance tuning plan and the troubleshooting burden can expand greatly. Spreading messaging engines across the members of the messaging cluster also adds to the administrative burden associated with creating and maintaining policies.

From a scalability standpoint, the Application, Remote Messaging, and Remote Support topology pattern provides the most flexibility. Because each of the distinct functions within IBM Business Process Manager is divided among the three clusters, you can pinpoint performance bottlenecks and adjust the cluster size fairly easily. If you need more processing capability for your business processes or human tasks, you can add more nodes and members to the application target cluster. Because expanding the messaging infrastructure beyond three cluster members has no effect on processing capability, the scalability limitations of the Application, Remote Messaging, and Remote Support topology pattern apply.

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Describe the concepts of business process and business process management
- Describe the IBM products that support SOA application development
- Describe the capabilities of IBM Business Process Manager
- Describe the main components of IBM Business Process Manager
- Describe the deployment considerations for the databases
- Identify and explain the clustered topologies for IBM Business Process Manager

Figure 1-52. Unit summary WB868 / ZB8681.0

Notes:

WebSphere Education

Checkpoint questions

1. A graphical console that is intended for administrators and developers who must manage the lifecycle of application components is:
   A. Administrative console
   B. Process Center Console
   C. Process Admin Console
   D. Performance Admin Console

2. A business process is deployed and runs on which of the following products?
   A. IBM Process Server
   B. IBM Process Center

3. Which topology is ideal from a performance perspective and the best topology for long-running processes and human tasks?
   A. Single cluster
   B. Application, Remote Messaging, and Remote Support

Figure 1-53. Checkpoint questions WB868 / ZB8681.0

Notes:

Write your answers here:

1.

2.

3.

WebSphere Education IBM

Checkpoint answers

1. B. Process Center Console

2. A. IBM Process Server

3. B. Application, Remote Messaging, and Remote Support

Figure 1-54. Checkpoint answers WB868 / ZB8681.0

Notes:

Unit 2. Performance concepts and methodologies

What this unit is about

This unit describes the need for performance monitoring and tuning. The unit provides an introduction to performance methodologies, performance terms, and describes how to do measurements.

What you should be able to do

After completing this unit, you should be able to:

- Describe performance monitoring and tuning methodologies
- Describe performance terminology
- Describe performance planning and design tasks that are performed during development
- Describe the goals of performance monitoring
- Identify key steps for Business Process Manager performance

How you will check your progress

- Checkpoint

References

IBM Business Process Manager V8.5 IBM Knowledge Center

`http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/ditamaps/ic-homepage-bpm.html`

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Describe performance monitoring and tuning methodologies
- Describe performance terminology
- Describe performance planning and design tasks that are performed during development
- Describe the goals of performance monitoring
- Identify key steps for Business Process Manager performance

Figure 2-1. Unit objectives WB868 / ZB8681.0

Notes:

WebSphere Education

Topics

- Performance tuning and monitoring overview
- Performance terminology
- Test planning and design
- Production monitoring and tuning
- Key steps to Business Process Manager performance tuning

Figure 2-2. Topics WB868 / ZB8681.0

Notes:

2.1. Performance tuning and monitoring overview

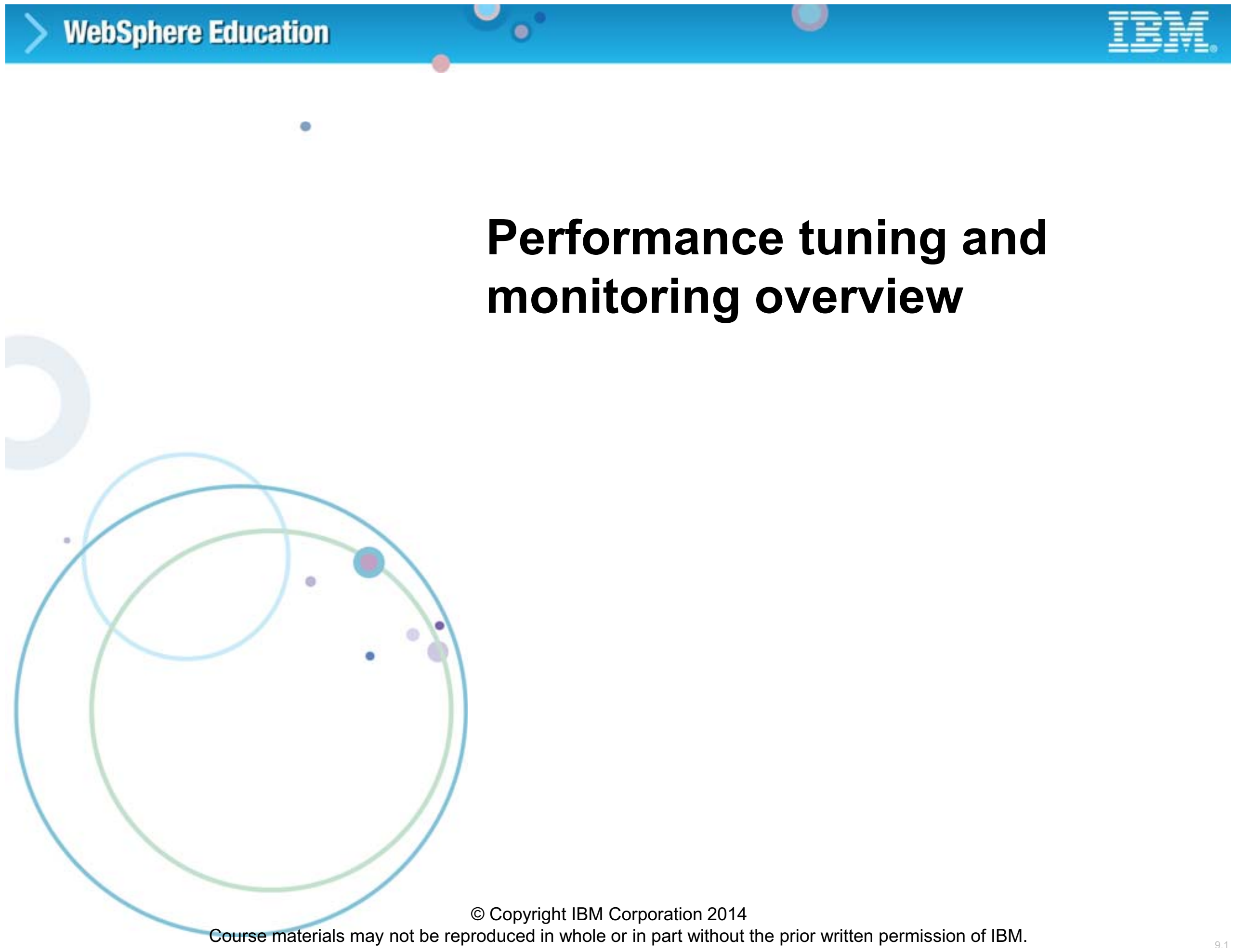


Figure 2-3. Performance tuning and monitoring overview WB868 / ZB8681.0

Notes:

WebSphere Education

Website performance concerns

- When a user clicks the submit button to send the request to the web application, the web application returns an HTML page to the browser on the user's computer
- The website performance can be measured by using high-level performance metrics:
  - Response time
  - Number of users
  - Error rate

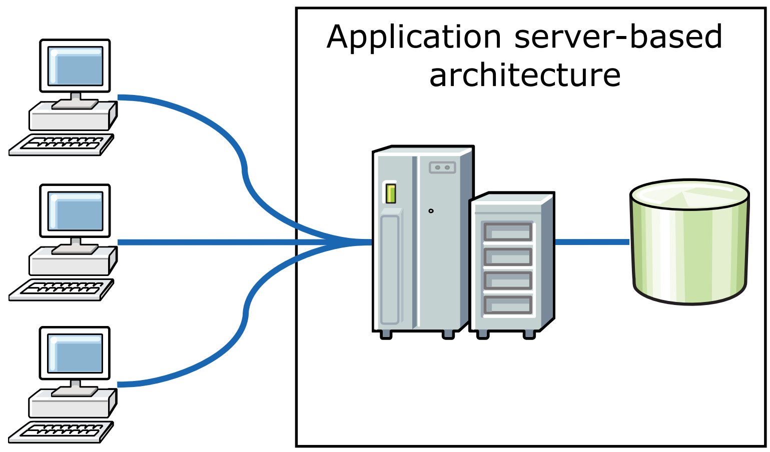


Figure 2-4. Website performance concerns WB868 / ZB8681.0

Notes:

WebSphere Education IBM

The need for performance monitoring and tuning

- How well a website performs while receiving heavy user traffic is an essential factor in the overall success of an organization
- Poor performance results in:
  – Escalated support costs
  – Loss of customer confidence
  – Loss of revenue
  – Loss of credibility
- Performance problems can be anywhere in the server environment
  – Monitoring ensures that applications are running as expected and, if not, determines why and where the problem lies
- Business Process Manager can function with default settings
  – Improving throughput and reducing server response times requires more tuning

Figure 2-5. The need for performance monitoring and tuning WB868 / ZB8681.0

Notes:

WebSphere Education

Tuning performance best practices

- Plan for performance
- Take advantage of performance functions
- Obtain performance advice from the advisors
- Tune the environment
- Troubleshoot performance problems

Figure 2-6. Tuning performance best practices WB868 / ZB8681.0

Notes:

2.2. Performance terminology

WebSphere Education

Performance terminology

9.1

Figure 2-7. Performance terminology

WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Performance terms

- Fundamental vocabulary for performance specialists includes these key terms:
  - Response time
  - Load
  - Throughput
  - Bottleneck
  - Scalability
  - Capacity

Figure 2-8. Performance terms WB868 / ZB8681.0

Notes:

Fundamental vocabulary that performance specialists use includes key terms such as response time, load, throughput, bottleneck, scalability, and capacity.

WebSphere Education

Response time

- Response time measures an **individual** user's wait for a request
  - Usually expressed as average
- Major components of response time:
  - Processing time
  - Transit time (usually part of processing time)
  - Any wait time in queues
- What is acceptable response time?
  - Set by current industry standards

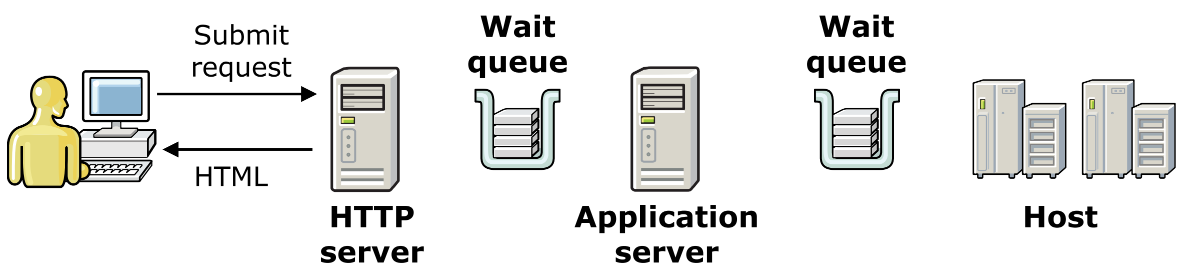
Figure 2-9. Response time WB868 / ZB8681.0

Notes:

Response time measures an individual user's wait for a request and is usually expressed as average. Major components of response time include processing time, transit time (usually part of processing time), and any wait time in queues. What is an acceptable response time? Current industry standards typically set acceptable response time.

WebSphere Education

Response time: Website example



- Measured from when request is made to when HTML returns
- Website response time is a function of:
  – Raw processing time
  – Plus wait time at any number of queues
  – Plus transfer time between multiple components

Figure 2-10. Response time: Website example WB868 / ZB8681.0

Notes:

Response time is measured from when a request is made to when the HTML returns.

Website response time is a function of raw processing time, the wait time at any number of queues, and the transfer time between multiple components.

WebSphere Education

Response time critical measurement

- Response time is a critical measurement
- Poor response time results in dissatisfied customers
- Many websites fail because of response time issues

- Consider response time:
  – Under peak loading
  – Under pathological loading (extreme market days)
  – Over modest dial-up or slow connections

Figure 2-11. Response time critical measurement WB868 / ZB8681.0

Notes:

Response time is a critical measurement. Poor response time results in dissatisfied customers. Many websites fail because of response time issues. You should consider response time under peak loading, under pathological loading (extreme market days), and modest dial-up or slow connections.

WebSphere Education

Load and throughput

- *Load* is the pressure against the application. Expressed as:
  - **User activity**: Users arriving, logging in, and sending requests
  - **Request activity**: Requests per second, pages per hour
- *Throughput* measures things that are completed in a unit of time; usually measured as requests per second
  - Example: Website pages that are served per second
  - Example: If a restaurant has only one server, and takes 1 minute to serve a customer, then the maximum throughput is one customer per minute
- *Maximum throughput* is a capacity measurement
  - Maximum obtainable system output in a unit of time
  - Not a measurement of requests; only how many are fulfilled
  - Excess requests might queue, leave, or be discarded

Figure 2-12. Load and throughput WB868 / ZB8681.0

Notes:

Load is the pressure against the website. It is expressed as user activities such as arriving, logging in, and sending requests. Request activity can be measured as requests per second, pages per hour, and so forth.

Throughput measures activities that are completed in a unit of time; for example, website pages that are served per second.

Throughput saturation

- At maximum throughput, more load does not yield more throughput
- Maximum throughput is a saturation point
  - 100% CPU utilization in the ideal case

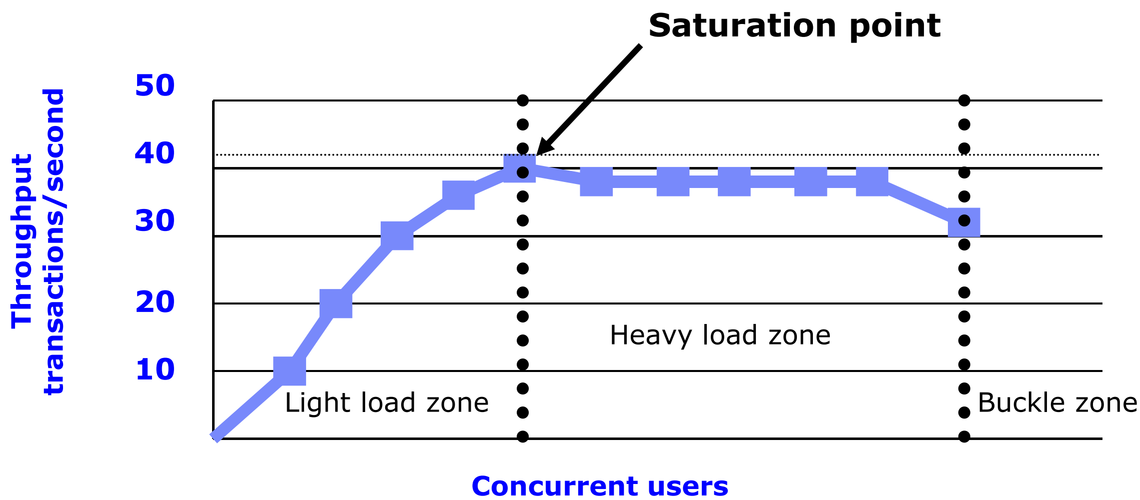


Figure 2-13. Throughput saturation WB868 / ZB8681.0

Notes:

Maximum throughput is a capacity measurement. It is the maximum obtainable system output in a unit of time.

The chart does not take into account response time. Ideally response time does not increase until the saturation point is reached.

WebSphere Education

Response time and throughput relationship

- Response time is closely tied to maximum throughput
- Beyond maximum throughput:
  - New arrivals begin to queue
  - Time in queue is added to overall response time
  - Wait is linear beyond maximum throughput

Figure 2-14. Response time and throughput relationship WB868 / ZB8681.0

Notes:

Response time is closely tied to maximum throughput. When maximum throughput is exceeded, new arrivals begin to queue up. The time that is spent in the queue is added to overall response time. Wait time is linear beyond maximum throughput.

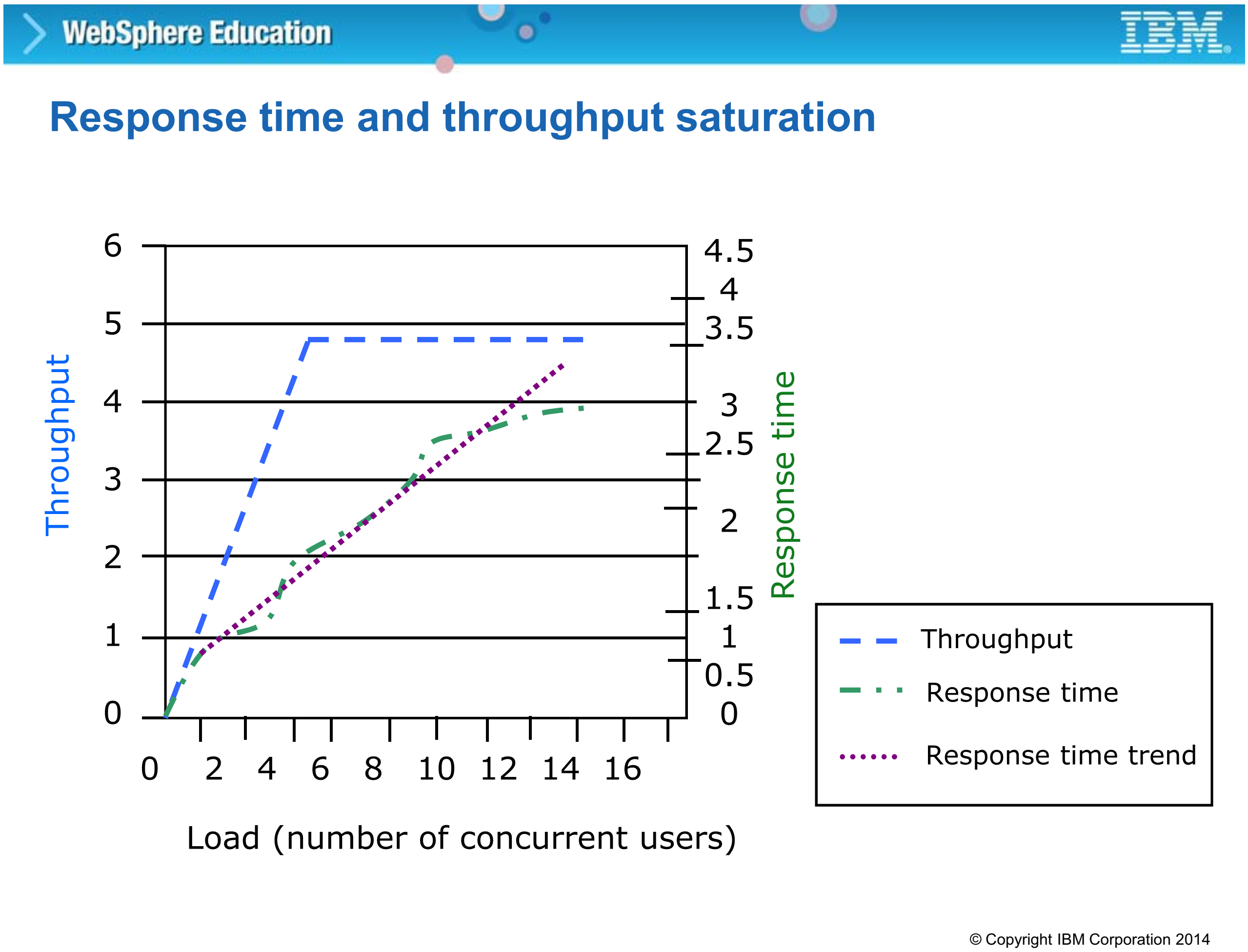


Figure 2-15. Response time and throughput saturation WB868 / ZB8681.0

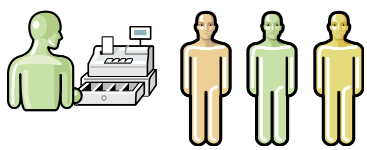
Notes:

This graph shows throughput and response time as a function of concurrent users. Throughput increases linearly as the number of concurrent users increases up to maximum capacity or throughput saturation. When maximum capacity is exceeded, throughput remains constant. Response time generally shows a linearly increasing trend beyond maximum capacity as the number of concurrent users increases.

WebSphere Education

Saturated cafeteria example

- A system can support more load beyond maximum throughput
- Taking a lesson from the cafeteria:



- If customers arrive slightly faster than the server serves:
  - A line begins (queuing)
  - Response time remains good
    Response time = (Time in queue + service time)
  - Throughput is unchanged (service time is constant)

- If traffic arrives significantly faster than the service rate:
  - Response time increases as wait time increases
  - Throughput eventually becomes compromised

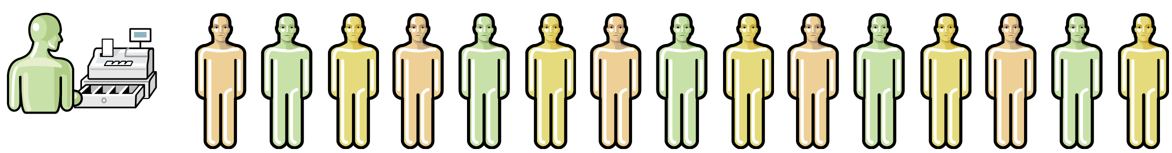


Figure 2-16. Saturated cafeteria example WB868 / ZB8681.0

Notes:

A system might support more load beyond maximum throughput. Using the example of customers who are waiting to be served in a cafeteria, two scenarios involve maximum throughput.

In the first case, customers arrive only slightly faster than the server serves. A small waiting line begins, but response time is still good. Because the service time is constant, the wait in line slightly increases the response time. This case works only for bursts of activity. Inactive periods are needed to keep the waiting line under control.

In the second case, customers arrive significantly faster that the service rate. A long line forms, and response time increases as the wait time increases.

WebSphere Education

What is a bottleneck?

- Defines a point of congestion or blockage in the system
- Appears only in multi-threaded or multi-user programming
- Users queued waiting for a shared resource
  - CPU
  - Pooled resource (threads, database connections, JMS connections)
  - Disk I/O
- Threads waiting for some task to complete
- Resolve bottlenecks in order of their severity

- Conclusion:
  - Your system is as fast as your slowest component

Figure 2-17. What is a bottleneck? WB868 / ZB8681.0

Notes:

Bottleneck defines a point of congestion or blockage in the system. It appears only in multi-threaded or multi-user programming. Users are queued waiting for a shared resource such as CPU, a pooled resource, or disk I/O. Threads might be waiting for some task to complete. Resolve bottlenecks in order of their severity. Your system is as fast as your slowest component.

WebSphere Education

Bottleneck example 1

- A slow component, such as a database server

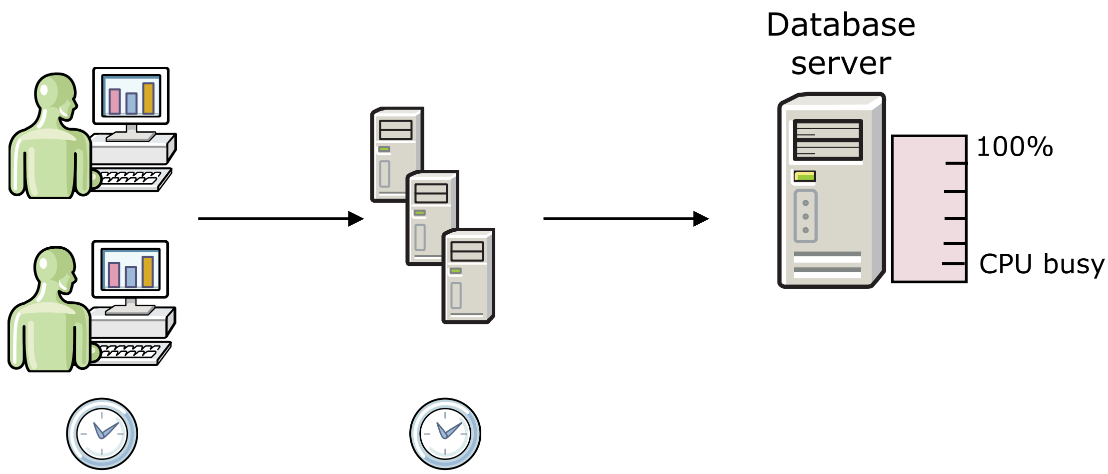


Figure 2-18. Bottleneck example 1 WB868 / ZB8681.0

Notes:

Any slow component in the request flow such as a web server or a remote database server can cause website bottlenecks.

Bottleneck example 2

- Insufficient resources, such as a small connection pool

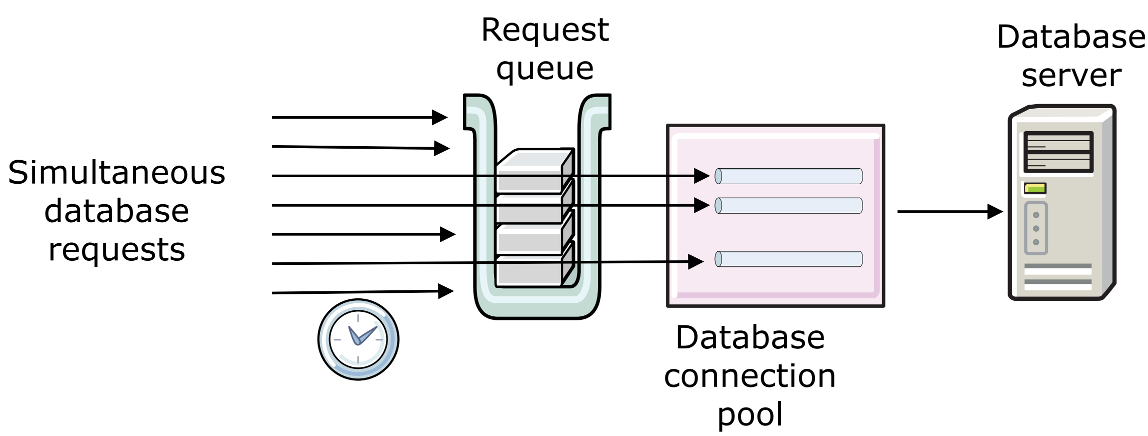


Figure 2-19. Bottleneck example 2 WB868 / ZB8681.0

Notes:

Insufficient resources, such as connection pools that are too small, can cause website bottlenecks.

WebSphere Education

Prioritizing bottlenecks

- Almost every system has bottlenecks
  - Eliminating every bottleneck is not feasible
  - Not every bottleneck has the same performance impact
- Eliminate bottlenecks according to severity
  - The primary activity of performance analysis

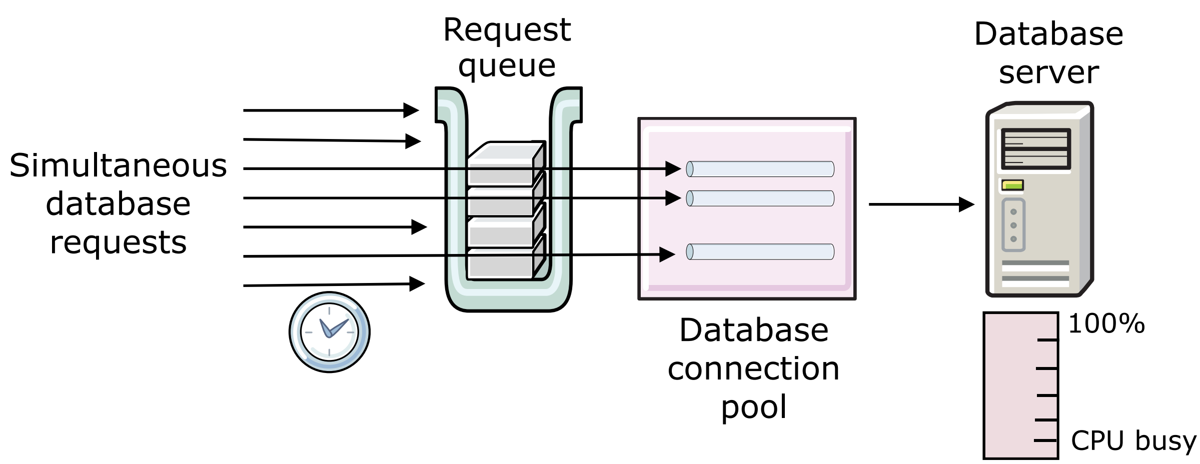


Figure 2-20. Prioritizing bottlenecks WB868 / ZB8681.0

Notes:

Almost every system has bottlenecks. Eliminating every bottleneck is usually not feasible. Not every bottleneck has the same performance impact. The process of identifying and eliminating bottlenecks according to severity is one of the primary activities of performance analysis.

What is scalability?

- Defines how easily a site expands
- Sites must expand, sometimes with little warning
- Grow to support increased load
- Load can come from many sources
  - New markets
  - Normal growth
  - Extreme peaks
- Good scalability makes site growth possible and easy

Figure 2-21. What is scalability? WB868 / ZB8681.0

Notes:

Scalability defines how easily a site expands. Sites must expand, sometimes with little warning, to support increased load. Load can come from many sources: new markets, normal growth, and extreme peaks in activity. Good scalability makes site growth possible and easy.

WebSphere Education

Scalability example

- How can you grow a website to handle more load?
- Vertical scaling
  - Add CPUs to existing machines
  - Can your JVM use these CPUs?
  - Can you fully use the CPUs you already have?
  - Do you need more JVMs?
- Horizontal scaling
  - Add machines to the site cluster
  - How many do you need?
  - Each machine hosts one or more new JVMs
- Remember the infrastructure
  - Network, databases, HTTP servers

Figure 2-22. Scalability example WB868 / ZB8681.0

Notes:

For website scalability, the following must be considered. How does a website grow to handle more load?

There are two approaches to scaling:

- Vertical
- Horizontal

For vertical scaling, you can add CPUs to existing machines, if your JVM can use these CPUs. Perhaps you can more fully use the CPUs you already have. Consider whether you need more JVMs.

For horizontal scaling, you can add machines to the site cluster. Consider how many more machines you need. Each machine might host one or more new JVMs.

Finally, you might combine both vertical and horizontal scaling if necessary.

WebSphere Education

What is capacity?

- Describes how much load the site supports
- The result of performance and load testing
- Start with simplest elements and work up
  – How much load will one server support?
  – How much load will 2, 8, or 10 servers support?
- Capacity
  – Determines hardware and infrastructure needed today
  – Leaves enough space for emergency capacity
  – Considers growth plans for future load expansion

Figure 2-23. What is capacity? WB868 / ZB8681.0

Notes:

Capacity describes how much load the site supports. Discovering the site’s capacity is the result of performance and load testing.

Start with simplest elements and work up. How much load will one server support? How much load can 2, 8, or 10 servers support?

Capacity dictates current hardware and infrastructure requirements. However, you should leave enough space for emergency capacity and consider growth plans for future load expansion.

Here are some capacity considerations for a website:

- How many servers are needed to handle 10,000 visitors per day?
- Do you need extra capacity for high-volume days?
- When do you exceed this capacity?
- How can you increase the capacity of the site?
- Do you need to add servers or CPUs?
- At what point do you upgrade the infrastructure?

WebSphere Education IBM

Performance terminology summary

- Each concept is probably suitable for a scholarly dissertation
  - More theoretical and mathematical information is available
- Keep terminology accessible for external groups
  - What are they trying to find in their testing?
  - What goals are important to them?
- Keep terminology consistent within the team
  - Many shops misuse some of these terms such as throughput, capacity, and scalability
  - Make sure that you understand what your customer means

Figure 2-24. Performance terminology summary WB868 / ZB8681.0

Notes:

In summary, keep in mind the following about performance terminology.

Keep terminology accessible for external groups. What are they trying to find in their testing? What goals are important to them?

Keep terminology consistent within the team. Many people misuse some of these terms. Make sure that you understand what your customer means.

2.3. Test planning and design

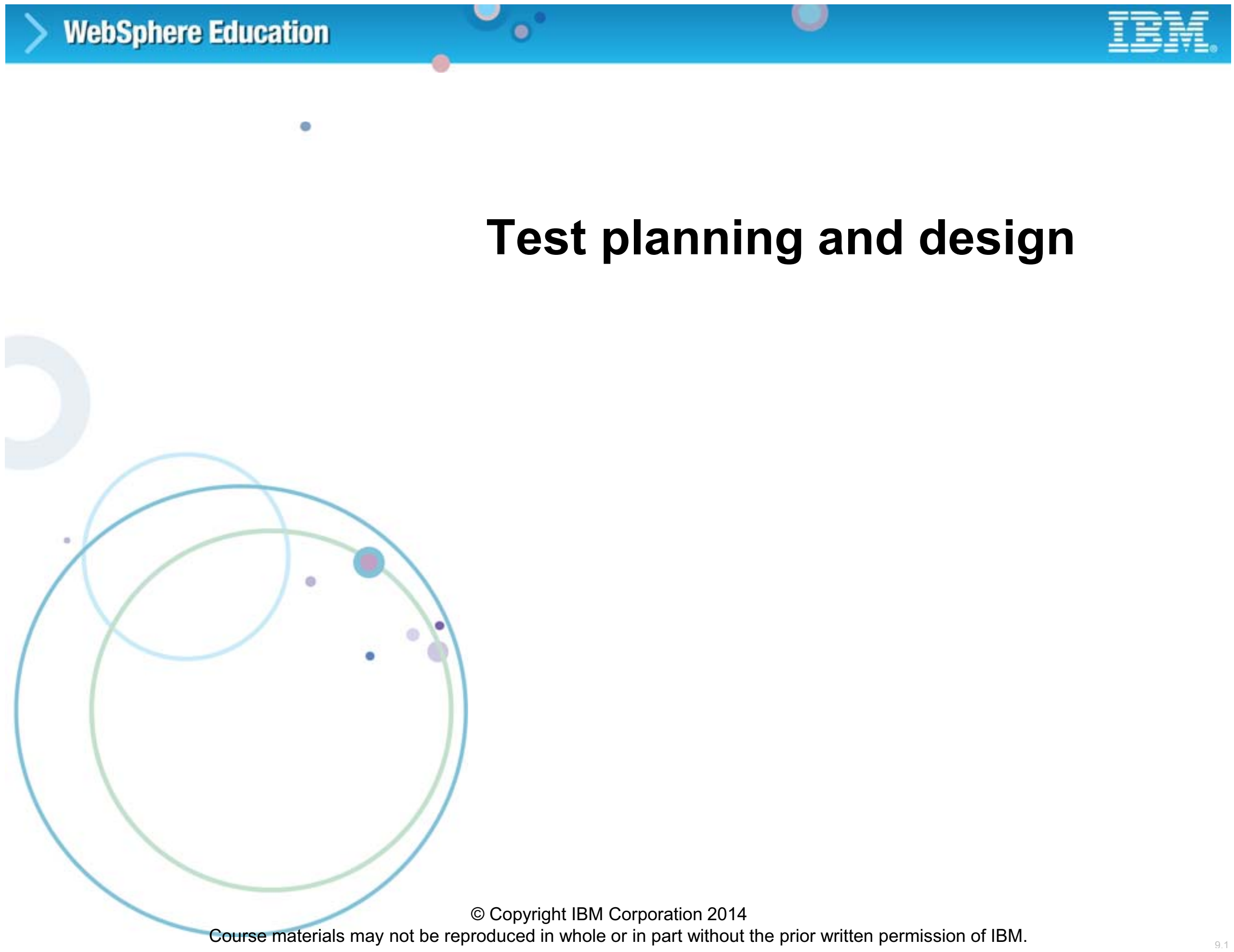


Figure 2-25. Test planning and design WB868 / ZB8681.0

Notes:

WebSphere Education

What is performance testing?

- The process of exercising an application by emulating actual users with a load generation tool for finding system bottlenecks

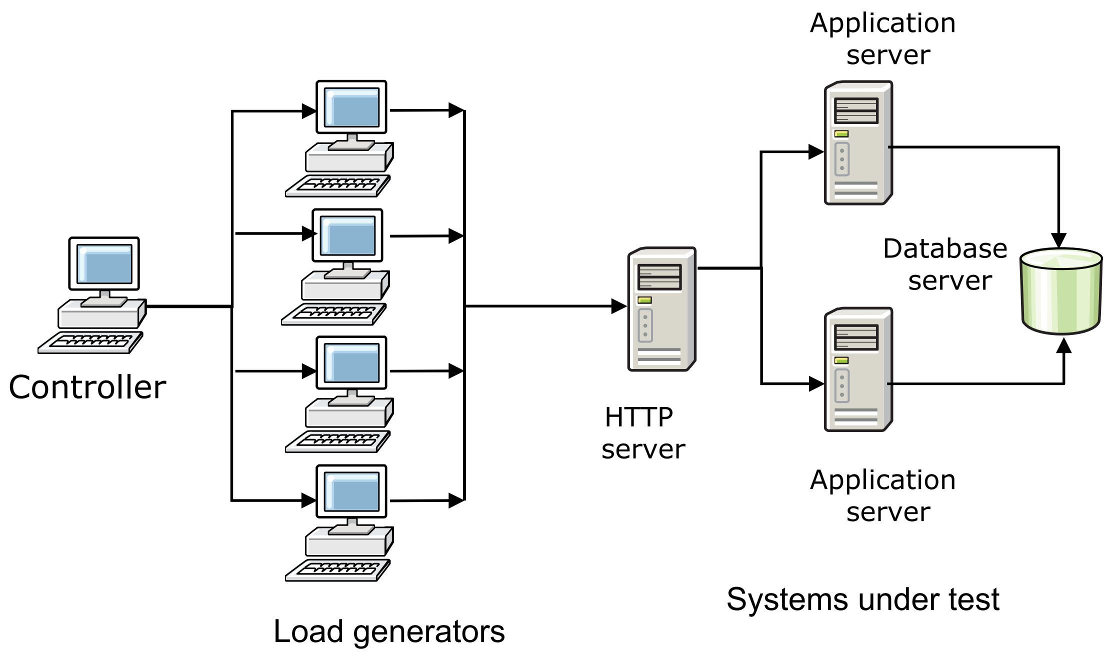


Figure 2-26. What is performance testing? WB868 / ZB8681.0

Notes:

A load generation tool creates the load that simulates actual users. Typically, a master computer controls other computers, and “virtual testers”, to emulate this production load. When the test run is complete, the data is returned to the master where the user analyzes the data to find the performance bottlenecks.

Performance testing includes not only testing the application and application server but also all of the other pieces that help create the entire environment. Typical activities to emulate include:

- User transactions and scenarios
- Peak activity periods
- Batch jobs and other internal system processes
- Administrator tasks (for example, system backups)

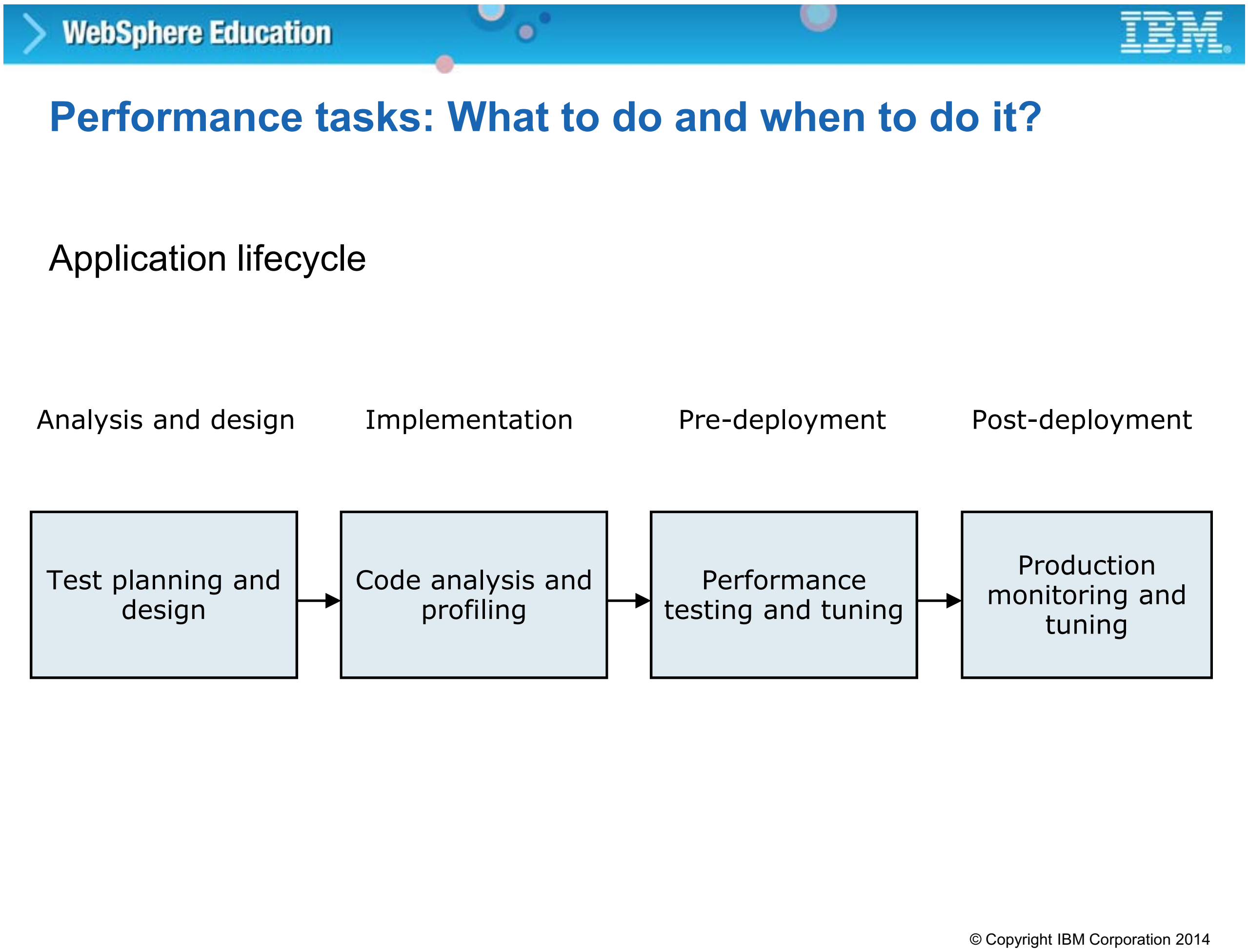


Figure 2-27. Performance tasks: What to do and when to do it? WB868 / ZB8681.0

Notes:

Test planning and design should be done during the analysis and design phase of the application lifecycle.

WebSphere Education

Test planning and design objectives

- Develop a performance test plan
  - Define performance test scope
  - Describe performance goals and success criteria
  - Identify required resources
  - Generate a test plan, select a set of reasonable initial parameter settings
- Perform workload analysis
  - Identify critical business functions to be measured
  - Specify performance measurement metrics
  - Define scenarios and user types
  - Generate a workload analysis document
- Establish a benchmark:
  - A **benchmark** is a standard set of application functions to run
  - Use the benchmark to test the application under expected loads
  - Record throughput and response time under normal load and peak load

Figure 2-28. Test planning and design objectives WB868 / ZB8681.0

Notes:

The goal of test planning is to create a test plan document that describes what is tested, how it is tested and measured, and who and what are needed to successfully implement it.

Workload analysis (as defined in Rational Unified Process) is a design time task that includes the following activities:

- Clarify the objectives of performance testing and the use cases
- Identify the use cases to be implemented in the model
- Identify the actors and actor characteristics to be simulated or emulated in the performance tests
- Identify the workload to be simulated or emulated in the performance tests (in terms of number of actors, actor classes, and actor profiles)
- Determine the performance measures and criteria
- Review the use cases to be implemented and identify the execution frequency
- Select the most frequently called use cases and those cases that generate the greatest load on the system

- Generate test cases for each of the use cases that are identified in the previous step
- Identify the critical measurement points for each test case

The main objective of workload analysis is to define a realistic representative workload that allows performance risks to be accurately assessed. The generated artifact, a workload analysis document, communicates in greater detail than the test plan what is tested, how it is run, and how the tests are verified.

WebSphere Education IBM

Test planning and design considerations

- Involve all of the necessary people
  - Business analysts
  - Architects
  - Developers
  - Testers and QA team
  - Administrators
- Think real world or production level testing
  - Ensure that tests represent real user activity
  - Ensure that the test environment is representative of production

Figure 2-29. Test planning and design considerations WB868 / ZB8681.0

Notes:

All five groups are stakeholders and need to be included in the planning phase to make sure that the performance test plan is going to test the system properly:

- **Developers** understand what pieces of the code do the work and provide the major functions.
- **Architects** know how the application was designed and the layout of the code.
- **Business analysts** know what they are marketing and how the public plans to use the application.
- **Testers and QA team members** know the test environment and do most of the work to prepare for the performance test.
- **Administrators** are called to set up the performance test environment.

This group facilitates and constructs the performance test plan document that is used throughout the testing efforts.

Test plan creation steps

- Identify test requirements
  - Define what is tested
  - Establish test priorities
- Define test strategy
  - Describe testing approach
  - Define what makes the test a success
  - Specify what level of performance is required
- Identify resources that are needed for testing
  - Time
  - Money
  - Hardware and software
- Create a schedule
  - Identify who does what
  - Define order of tests
- Document the test plan

Figure 2-30. Test plan creation steps WB868 / ZB8681.0

Notes:

Other important resources that are needed for a successful performance test should also be planned:

- Time for testing should be scheduled following development. This schedule includes both time to develop test content and time to run the test.
- Money to buy the necessary performance test tools should be allocated. Test tools can be expensive, but this resource is money that is spent well.
- Hardware that is comparable to the deployment environment should be used. If it is not available, you must extrapolate data from smaller systems to the deployment systems.

Make sure to document the test plan and adjust it as needed as testing progresses.

WebSphere Education

Workload analysis steps

- Identify business processes to be tested
  - Include major business processes
  - Cover at least 70% to 80% of your code
  - Define test cases and data
- Define workloads
  - Identify user types and characteristics
  - Define the percentage of users that run each business process
  - Approximate data access patterns
  - Define scenarios
- Select measurement criteria
  - Response times
  - Throughput
  - Validity of transactions
- Generate workload analysis document
  - Used to create actual tests and scenarios

Figure 2-31. Workload analysis steps WB868 / ZB8681.0

Notes:

A workload is a complete test scenario (target load) to be placed on the system under tests that consist of:

- User attributes (types and characteristics)
- Transaction mix and frequency
- Data set
- Test environment configuration
- Measurement criteria

It should be a realistic representation of actual users and transactions. It is important to define what is monitored and what is acceptable for the application testing. Well-chosen measurement points make performance analysis easier.

The first step in workload definition is to clarify and document the goals of the performance-testing project. Testing goals are defined with user or customer involvement and are documented to ensure that everyone agrees with the goals. The major activity of workload definition is identifying and documenting the critical business functions that make up most of the user activity. Each business function or task is first identified and then described in terms of user input data and data that is accessed. The workload definition must describe the details about how the user works through the scenario, including average typing rates and user think-time intervals. The definition should also describe an acceptable approximation for data access frequencies and patterns for data that is retrieved from the computer database as part of the user scenario.

Document the workload analysis and make sure that it includes a user percentage breakdown by business process that reflects how the application is used in production. By documenting all of the user scenarios, the project members have complete user documentation for later creating test scripts and schedules.

Select measurement criteria that includes:

- Response times
- Throughput
- Validity of transactions

During workload analysis, a workload analysis document should be generated which is used to create actual tests and scenarios.

WebSphere Education IBM

Test environment setup considerations

- Start test environment setup after test plan and workload analysis are documented
- Servers should:
  - Use production-level code
  - Duplicate production configuration
  - Not be a production system
- Client systems should:
  - Be able to drive the planned maximum number of virtual users
  - Assign a different IP address for each virtual user
- Network should be set up with:
  - A dedicated LAN between client and application
  - A separate LAN for testing tool and client communication
- Environment should be:
  - Representative of deployment
  - Controlled
  - Dedicated

Figure 2-32. Test environment setup considerations WB868 / ZB8681.0

Notes:

It is a good idea to start setting up the test environment as soon as the test plan and workload analysis documents are produced. This plan allows more time to procure, install, and configure all of the hardware, system software, and testing tools required.

The following needs to be set up in the test environment:

- Lab hardware
- System software
- Data set
- Test and monitoring tools

Ideally, the servers are similar to servers that are used in production.

The client systems are used to drive the simulated requests.

The network should be dedicated. In the best circumstances, the user traffic and application communication are on separate LANs.

Most importantly, the environment needs to be controlled and dedicated. Only certain people are allowed to implement modifications so that everything can be documented.

WebSphere Education

Typical goals of performance testing

- Identify system response times
  - Validation of requirements and performance goals
  - Benchmarking
  - Service level agreements (SLAs)
- Determine the maximum number of users for a system
  - Capacity planning
  - Scalability
- Discover optimum or minimum configurations

Figure 2-33. Typical goals of performance testing WB868 / ZB8681.0

Notes:

Performance testing allows you to:

- Determine the response time of an application
- Determine how many users the system can support
- Determine the optimum system configuration
- Verify that different hardware and software works satisfactorily
- Find out what happens when a system is put under heavy load

Beyond observation and gathering, engineers use the data and metrics to tweak the target application to improve its performance. This discovery and resolution of bottlenecks (instances of slow or unacceptable performance) are the real return on investment (ROI) of a performance testing tool.

Benchmarking is the process of comparing a system’s performance to an established level to match or improve upon it.

Performance testing can help to reduce system costs by determining what system resources, such as memory and disk resources, are needed to deliver acceptable performance. Understanding the minimum requirements enables you to make better decisions about what hardware and software needs to be purchased.

WebSphere Education

Types of performance tests

- Load testing
  - Simulates anticipated user load and actual business operations
  - Identifies load level with acceptable response times
  - Provides metrics on performance bottlenecks
- Stress testing
  - Evaluates performance at a level well beyond estimated loads
  - Performs sustained tests at high loads or short-term burst loading
  - Determines the load under which a system fails and how it fails
- Longevity and endurance testing
  - Simulates normal loads for a long time
  - Identifies performance problems, such as slow memory leaks, that might appear after a system is running for an extended time

Figure 2-34. Types of performance tests WB868 / ZB8681.0

Notes:

Load testing refers to more realistic loads for an application. It is based on business patterns and should simulate actual business operations. It is suggested that you do load testing at 10 to 20 percent above the projected user load to accommodate growth.

Stress testing refers to long runs at saturation levels. It also refers to short-term burst loading, such as a large numbers of logins at one time. In general during stress testing, you ignore poor response times.

Other types of performance tests include:

- **Failover testing**: Simulates a hardware or software failure to test failover mechanisms.
- **Configuration testing**: Tests for compatibility issues, determines minimal and optimal configuration of hardware and software, and determines the effect of adding or modifying resources such as memory, disk drives, and processor.

WebSphere Education

Results analysis and problem resolution

- Analyze results to find:
  - Slow pages
  - Resource bottlenecks

Identify performance problems

- Resolve application bottlenecks
  - Design and architecture issues
  - Code implementation issues
  - Poorly tuned queries
- Tune runtime "outside in"
  - HTTP server queues
  - Database connection pool
  - Application server (connections, heap sizes)
- Optimize hardware settings
  - Server system resources (memory, CPU)
  - Network health
- Adjust operating system settings

Resolve problems in order of severity

Figure 2-35. Results analysis and problem resolution WB868 / ZB8681.0

Notes:

You need to be sure to verify the results. Performance bottlenecks can come from a wide variety of sources. Identify bottlenecks by severity; then work to resolve them in the order of severity. By working in this way, you do not chase problems that result from larger, more significant problems.

Implement the changes to fix the problem and rerun the same test to verify the result of the modification. When that issue is resolved, move to the next issue in order of severity with the newly implemented changes.

This graphic shows a general outline of the order in which problems might be resolved. Your own order of resolution might be different from the order that is shown in the outline. Each application performance tuning exercise is unique.

2.4. Production monitoring and tuning

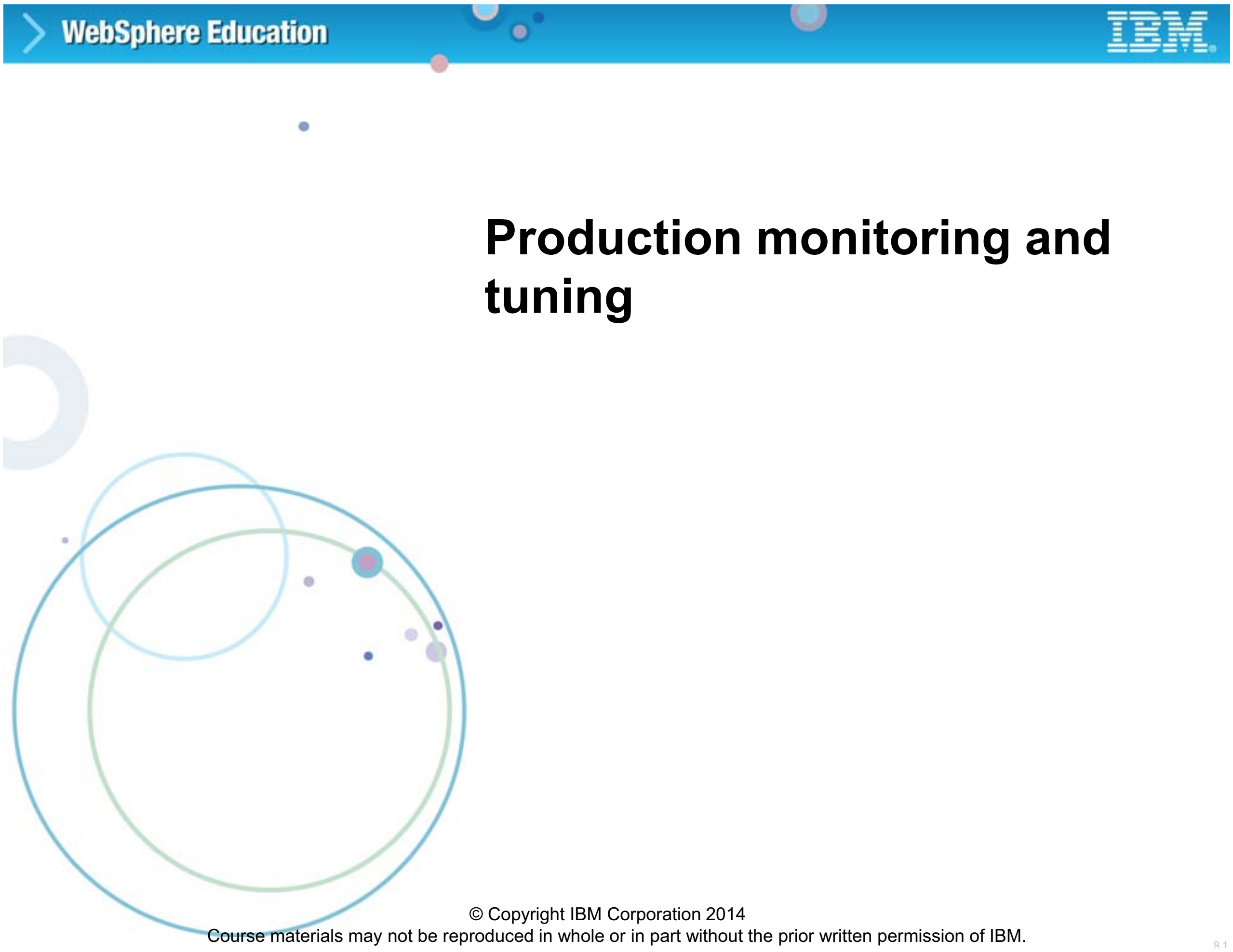


Figure 2-36. Production monitoring and tuning WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Performance monitoring overview

- The goal of performance monitoring is to ensure that your applications are running optimally
  - If not, to determine why
- Monitoring is central to performance diagnosis, tuning, and management solutions
  - Performance test and tuning
  - Problem identification and diagnosis
  - System health and availability management

Figure 2-37. Performance monitoring overview WB868 / ZB8681.0

Notes:

The goal of performance monitoring is to collect runtime statistics about your application and its environment to quantify their performance behavior. It allows you to determine whether your application meets its performance objectives and helps to identify any performance bottlenecks. Ultimately, monitoring helps to ensure that your applications are running optimally. Monitoring is at the core of the following performance-related activities:

- **Problem identification and diagnosis**: Java EE application designs are often complex. They typically include many interacting layers and use different types of component technologies such as servlets and Enterprise JavaBeans (EJBs), each with their own unique complexities. After deployment, the application runs in an environment that consists of firewalls, web servers, web application servers, databases, and many other necessary systems. This application and runtime architecture presents many areas for performance bottlenecks and single points of failure. You need to monitor each aspect to facilitate problem identification and diagnosis. When a problem occurs, monitoring helps to quickly identify the failing resource and determine its root cause.
- **Performance testing and tuning**: Your application and its runtime environment should be tuned optimally. This process entails conducting many iterations of a monitor-tune-test cycle.

Monitoring helps identify where to tune first (the trouble spots) and validate the results of your tuning changes with each iteration.

- **System Health and Availability Management**: System health and availability management solutions help ensure that resources are running within their established operational limits. When thresholds are exceeded, they can automatically generate alerts and even initiate corrective actions to resolve the problem. Monitoring provides the foundation for this proactive management of applications and resources. It also helps validate compliance with the performance terms defined in applicable service level agreements (SLAs).

WebSphere Education

Categories of parameters to monitor

- User view
  - Response time
  - Load
  - Throughput

Why is the system so slow?

- System view
  - CPU usage
  - Disk activity
  - Memory usage
  - JVM heap
  - Network

Is BPM running OK?

- Application view
  - Response time
  - Concurrent requests
  - Number of live sessions

Why is the Order process consuming so much time?

Figure 2-38. Categories of parameters to monitor WB868 / ZB8681.0

Notes:

In an environment with possibly thousands of different parameters, the key to monitoring is to logically partition the environment into different views. Each view looks at the environment from a different perspective, allowing you to drill down on a potential problem to find its root cause.

2.5. Key steps to Business Process Manager performance tuning

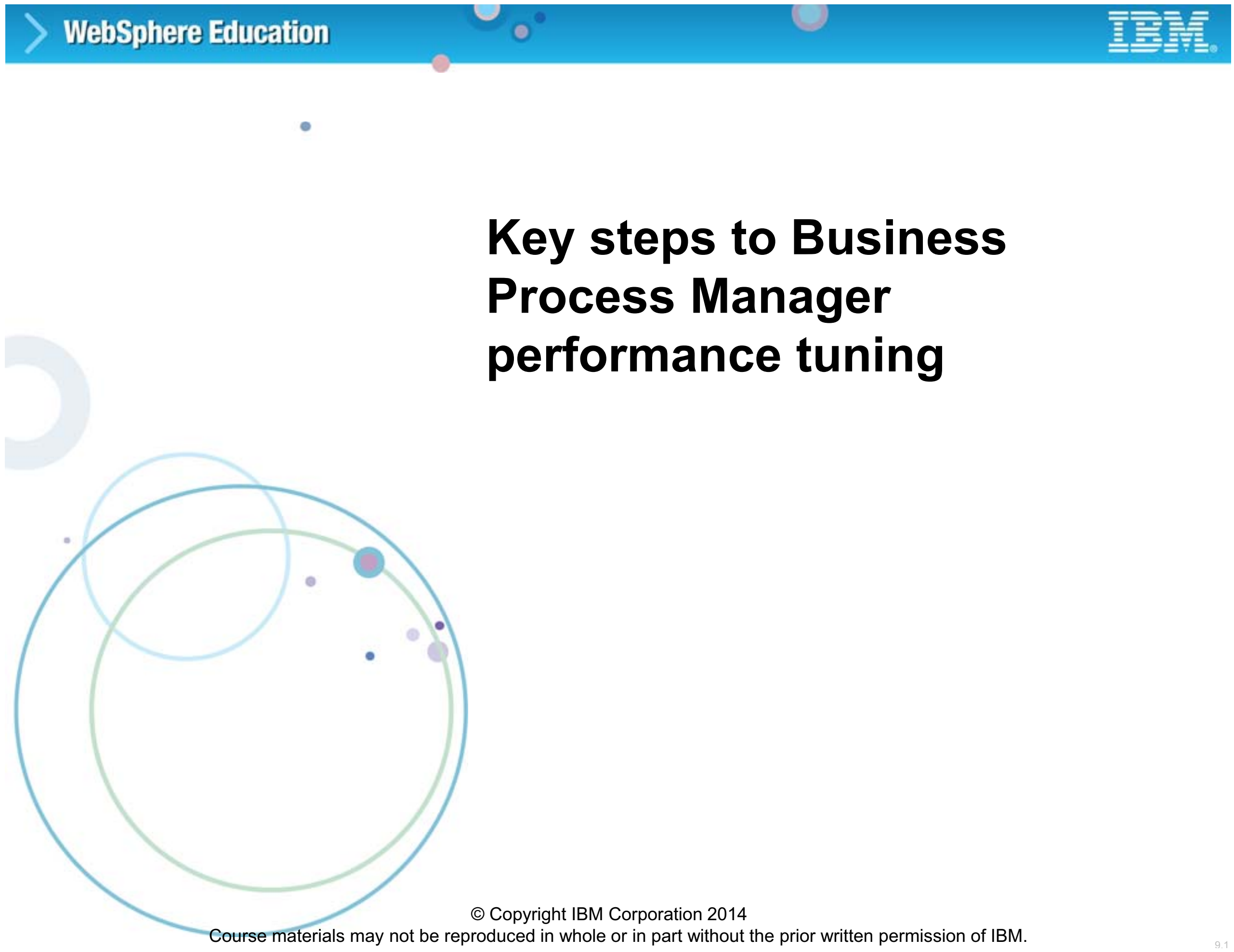


Figure 2-39. Key steps to Business Process Manager performance tuning WB868 / ZB8681.0

Notes:

WebSphere Education

Step 1: Apply a tuning methodology

- Establish clear, measurable performance objectives that are tied to specific business requirements
- Design workloads that are reliable and repeatable
- Perform a step-by-step tuning of the environment and document the changes
- Regularly monitor all involved systems and adjust the tuning as necessary
- Check for errors, exceptions, and timeouts in logs and reports

Figure 2-40. Step 1: Apply a tuning methodology WB868 / ZB8681.0

Notes:

WebSphere Education

Step 2: Review your applications

- Review your applications and apply performance optimizing practices
- Clear up variables once they are no longer used in business process definitions (BPDs) or services
- Avoid passing large business objects (BOs) among processes and linked processes, or through undercover agents
- Avoid a request-response design over asynchronous links
- Use multi-instance loops only to create tasks carefully

Figure 2-41. Step 2: Review your applications WB868 / ZB8681.0

Notes:

- Clear up variables as soon as they are no longer used in business process definitions (BPDs) or services. This task can reduce the amount of data that is stored in the databases.
- Avoid passing large business objects (BOs) among processes and linked processes, or through undercover agents. It is expensive to serialize and deserialize large BOs. Use the claim-check pattern instead.
- Avoid a request/response design over asynchronous links. For example, user interfaces can be built so that they initiate a BPD and then wait until the first task is created. This task might take a long time, which depends on what is in the BPD before the first human task is created.
- Be careful if you use multi-instance loops to create tasks. These loops can potentially create a huge number of tasks.

WebSphere Education IBM

Step 3: Tune the servers

- Ensure that the Java heap is sufficient and tune it properly
  - Monitor heap usage by using the verbose garbage collection log for the JVM
- Disable debug and trace messages for production servers
- Configure thread pools and connection pools to enable sufficient concurrency
  - Important for high-volume, highly concurrent workloads
  - Pool settings directly influence how much work the server can concurrently process
- Use fast, high-bandwidth network connections

Figure 2-42. Step 3: Tune the servers WB868 / ZB8681.0

Notes:

Business Process Manager servers use thread pools to manage concurrent tasks. You can set the Maximum Size property of a thread pool in the administrative console by selecting the server name whose thread pool you want to manage. Click **Additional Properties > Thread Pools** and then the thread pool name.

WebSphere Education IBM.

Step 4: Tune the databases

- Database performance is essential for overall system performance
- Use high-performance disk subsystems
  - For acceptable performance, you need a server-class disk subsystem on the tiers that host data stores
  - This step is the most important point when tuning the overall system
- Use reasonably sized buffer pools and caches
- Run the performance advisors from your database vendor

Figure 2-43. Step 4: Tune the databases WB868 / ZB8681.0

Notes:

Databases usually offer a wide variety of available choices when determining the best approach to access data. Statistics, which describe the shape of the data, are used to guide the selection of a low-cost data access strategy. Statistics are maintained in tables and indexes. Examples of statistics include the number of rows in a table and the number of distinct values in a certain column.

WebSphere Education IBM

Step 5: Maintain the system

- Regularly update database indexes
- Remove data from the database when it is no longer needed
  - For example, completed process instances
- Resize the system according to load growth, such as when you are deploying new applications and adding users

Figure 2-44. Step 5: Maintain the system WB868 / ZB8681.0

Notes:

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Describe performance monitoring and tuning methodologies
- Describe performance terminology
- Describe performance planning and design tasks that are performed during development
- Describe the goals of performance monitoring
- Identify key steps for Business Process Manager performance

Figure 2-45. Unit summary WB868 / ZB8681.0

Notes:

WebSphere Education

Checkpoint questions

1. True or false: Performance should not be considered until the application is monitored in a production environment.

2. True or false: Response time measures an individual user's wait for a request.

Figure 2-46. Checkpoint questions WB868 / ZB8681.0

Notes:

Write your answers here:

1.

2.

WebSphere Education

Checkpoint answers

1. False: Performance needs to be considered in every step from design through deployment.

2. True

Figure 2-47. Checkpoint answers WB868 / ZB8681.0

Notes:

Unit 3. Implementing for performance

What this unit is about

This unit provides an introduction to best practices for modeling, design, and development choices that are made when designing and implementing a Business Process Manager solution.

What you should be able to do

After completing this unit, you should be able to:

- Explain general development best practices
- Define the differences between BPMN and BPEL
- Explain Process Designer best practices
- Explain Integration Designer best practices

How you will check your progress

- Checkpoint

References

IBM Business Process Manager V8.5 IBM Knowledge Center

`http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/ditamaps/ic-homepage-bpm.html`

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Explain general development best practices
- Define the differences between BPMN and BPEL
- Explain Process Designer best practices
- Explain Integration Designer best practices

Figure 3-1. Unit objectives WB868 / ZB8681.0

Notes:

WebSphere Education

Topics

- Development overview
- IBM Process Designer authoring best practices
- IBM Integration Designer development best practices

Figure 3-2. Topics WB868 / ZB8681.0

Notes:

3.1. Development overview

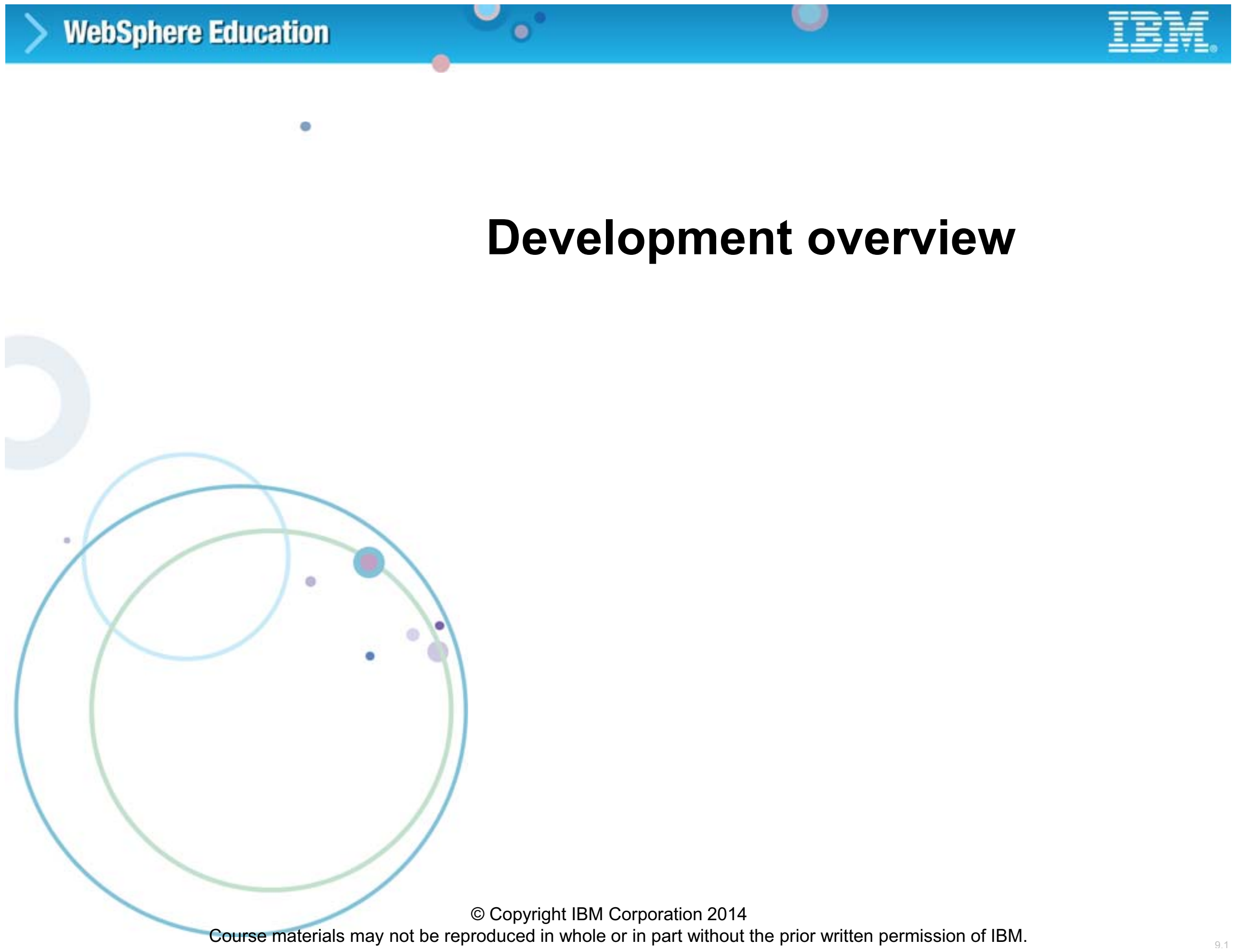


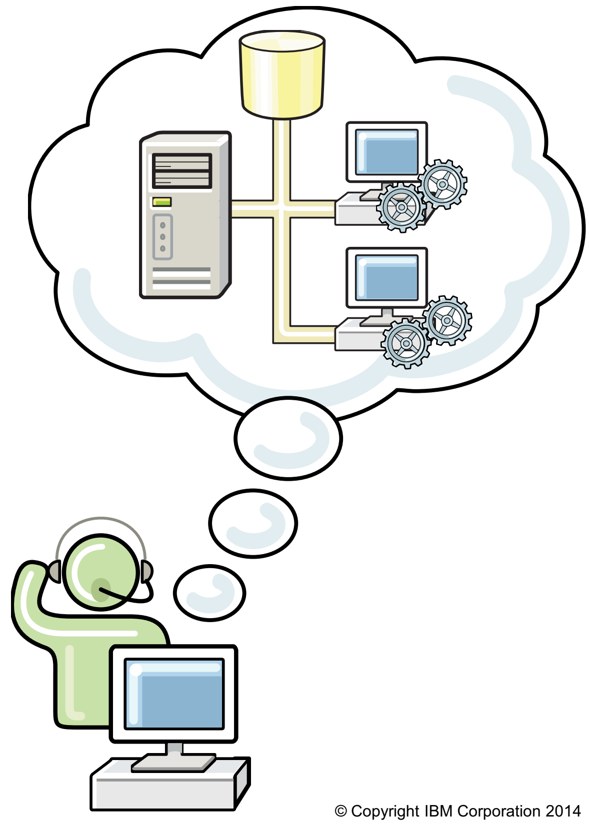
Figure 3-3. Development overview WB868 / ZB8681.0

Notes:

WebSphere Education

When to think about application performance

- Architecture design
- Development period
- Unit, component testing phase
- Performance testing phase
- Deployment and rollout
- Bottom line:
  - ***Always*** think about performance



© Copyright IBM Corporation 2014

Figure 3-4. When to think about application performance WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Development environments

- IBM Process Designer and IBM Integration Designer are two IDEs provided with IBM Business Process Manager Advanced
  – IBM Integration Designer is ***not*** provided with Express or Standard configurations
- IBM Process Designer is intended for **modeling**
  – Business analysts and development teams can use it for high-level business process solutions
  – Focuses on top-down human-centric processes and user interface creation
- IBM Integration Designer is intended for **technical development** that is used for low-level business process solutions, which offer:
  – Full SOA support
  – Focus on straight-through process creation and integration elements
  – Integration with other systems and services
  – Mediation module support for deployment to the WebSphere Enterprise Service Bus

Figure 3-5. Development environments WB868 / ZB8681.0

Notes:

IBM Process Designer is available in all editions of the product. IBM Business Process Manager Advanced also offers Integration Designer with its associated editors and adapters.

A process is the major unit of logic in IBM Business Process Manager. It is the container for all components of a process definition, including services, activities, and gateways; timer, message, and exception events; sequence lines, rules, and variables. When you model a process, you are creating a reusable business process definition (BPD). Both Process Designer and Integration Designer can create process models that can contain human tasks.

Integration Designer provides editors and aids to help developers create complex automated processes and services (such as SCA modules, mediations, and BPEL processes). They are available as part of the IBM Business Process Manager Advanced package or as a stand-alone toolset for other uses.

IBM Integration Designer is designed as a complete integration development environment for those building-integrated applications. Integrated applications are not simple. They can call applications on enterprise information systems (EIS), involve business processes across departments or enterprises, and invoke applications locally or remotely written in various languages and running on various operating systems. The components are created and assembled into other integrated applications (that is, applications that are created from a set of components) through visual editors. The visual editors present a layer of abstraction between the components and their implementations. A developer who uses the tools can assemble an integrated application without detailed knowledge of the underlying implementation of each component.

Comparing IBM Integration Designer and Process Designer

	IBM Integration Designer	IBM Process Designer
Container for integration artifacts	Module, which includes: • Integration logic (BPEL processes, human tasks, and business rules) • Data and interfaces • Transformations	Process application, which includes: • Processes (BPD, human tasks, and rules) • Data and services
Container for shareable artifacts	Library, which includes: • Integration logic • Data and interfaces • Transformations • Web Service Ports	Toolkit, which includes: • Processes • Data and services
Container for mediation services	Mediation module, which includes: • Mediation flows	N/A

Figure 3-6. Comparing IBM Integration Designer and Process Designer WB868 / ZB8681.0

Notes:

To model a process in Process Designer, you must create a business process definition (BPD). The business process definition can be based on an imported BPMN model.

A BPD is a reusable model of a process, defining what is common to all runtime instances of that process model. A BPD must contain a start event, an end event, at least one lane, and one or more activities.

A business process is a set of business-related activities that are invoked to achieve a business goal. A BPEL process is a business process that is defined in the Web Services Business Process Execution Language.

WebSphere Education

Workspaces in development environments

- IBM Process Designer
  - Requires a live connection to IBM Process Center
  - No local workspace; editors and artifacts are server-based
- IBM Integration Designer
  - Does not require a server connection
  - Requires connection to IBM Process Center only when reading from and writing to the repository
- IBM Integration Designer workspaces are code caches
  - Can be confusing when using repository and local workspaces
  - After changing, commit changes to the repository
  - Good practice to delete workspaces after committing changes
  - If you keep the workspace, always use “Refresh from Process Center”
  - Synchronization dialogs prevent conflicting changes during publish

Figure 3-7. Workspaces in development environments WB868 / ZB8681.0

Notes:

Using workspaces as primary source code repository can lead to confusion when multiple Integration Designer and Process Designer users work on the same artifacts. It is easy to forget to update from the repository and start changing the code that other developers already modified. Although compare and merge facilities exist in Integration Designer, it is simpler to always work with the newest code version.

After editing in Integration Designer, commit to the repository and delete the workspace. If you elect to keep the workspace around, always **Refresh from Process Center** before you start to change anything. When you do a **Refresh and Publish**, a Synchronization dialog window opens if other developers made conflicting changes.

WebSphere Education

Development roles

Role	Responsibilities
Integration programmer	• Is responsible for doing all of the integration work to support business processes that the business author creates • Focuses on building SOA and EAI solutions – Top-down, bottom-up, or meet-in-the-middle • Creates composite applications from integrated components • Has a basic understanding of business modeling • Expects authoring tools to simplify and abstract advanced implementation details • Is familiar with basic programming concepts • Understands business process choreography, workflow (including human interaction), WSDL, and BPEL • Creates mediation modules to implement connectivity logic • Uses IBM Integration Designer
Business author	• Is responsible for authoring all business processes • Able to use services but it not interested in the implementation details or how they work • Creates BPDs and uses Advanced Integration services (AISes) to collaborate with the integration programmer • Uses IBM Process Designer

Figure 3-8. Development roles WB868 / ZB8681.0

Notes:

A business author uses BPMN and JavaScript skills with a user interface focus.

An integration programmer uses BPEL, Java, and mediation skills with an integration and compensation focus.

WebSphere Education

Fit for purpose

- Deploying and running applications on the platform to offer the best performance at the best price
- Processes evolve constantly and might require more integration capabilities
  - Design your processes for change
  - Process Designer uses BPMN to create BPDs and service flows
  - Integration Designer creates BPEL processes
  - Use the features of both to create new solutions
- Guidance
  - Use BPMN for rich human interaction features
  - Use BPEL for high performance, business critical processes
  - Use BPEL over BPMN for service orchestration
  - Use mediation flows for integration and data translation

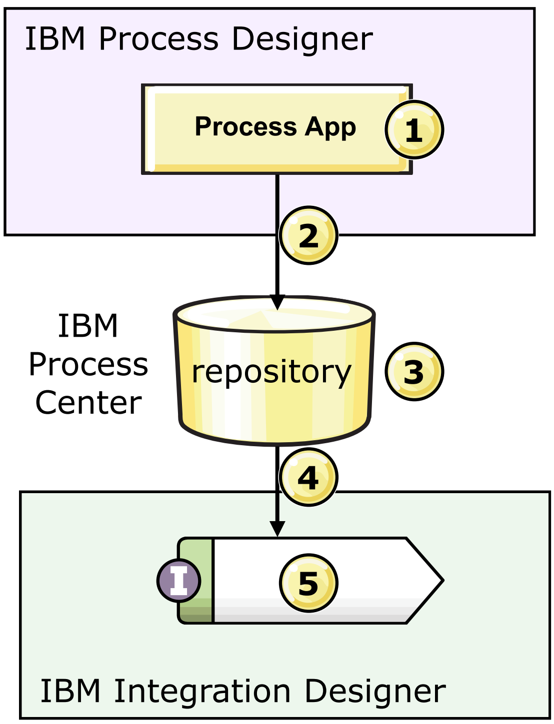
Figure 3-9. Fit for purpose WB868 / ZB8681.0

Notes:

A definition of *fit for purpose* is: something that is fit for purpose is good enough to do the job it was designed to do.

WebSphere Education

IBM Process Designer artifacts in IBM Integration Designer



1. Assets are built in IBM Process Designer.
2. A snapshot of asset is stored in IBM Process Center.
3. Developer uses Process Center perspective in IBM Integration Designer to read the repository.
4. Click **Open in Workspace**
   - Assets are added to synchronized project
5. Use business processes as export or import components.

Figure 3-10. IBM Process Designer artifacts in IBM Integration Designer WB868 / ZB8681.0

Notes:

Model: Business author

- Model the business process that is based on requirements
- Simulate “what if” conditions to refine the model
- Export the model for IBM Integration Designer

Assemble: Integration Designer

- Import process model
- Implement for runtime engine
- Test in Integration Designer

Deploy: System administrator

- Install the choreographed process model onto IBM Process Server

IBM Integration Designer artifacts in IBM Process Designer

1. Module is associated with process application or toolkit
2. "Make operations visible to IBM Process Designer" on import or export component
3. Changes published to repository
4. Open updated process application or toolkit
   - Imports and exports are displayed as "Advanced Integration service" implementations
   - Supporting artifacts, such as business objects, are read-only

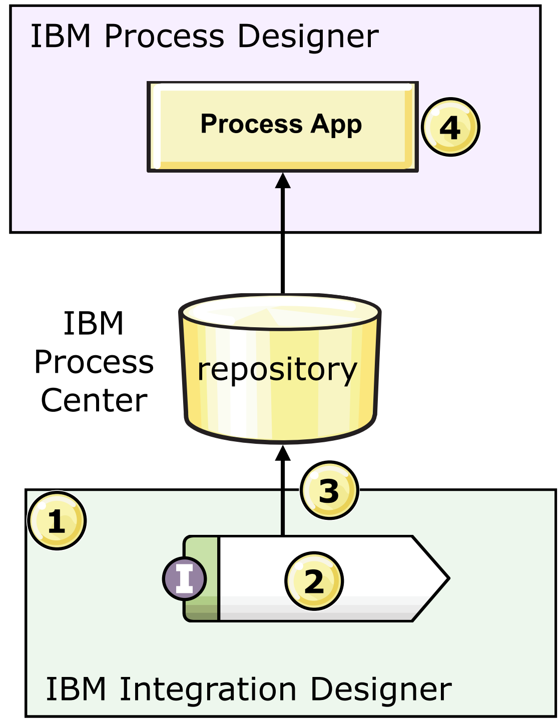


Figure 3-11. IBM Integration Designer artifacts in IBM Process Designer WB868 / ZB8681.0

Notes:

WebSphere Education

Using Advanced Integration services

- Assets from IBM Integration Designer, which are added to the repository are used in IBM Process Designer as “**Advanced Integration services (AIS)**”
- Use an AIS with caution:
  - Placing a many modules with many components in a toolkit is costly
  - Assets are deployed in an EAR ***by copy*** not *by reference*
  - Performance is negatively affected when many process applications use that toolkit
- Use a service facade pattern with an AIS:
  - Limits the excessive number of deployed EAR files
  - Promotes isolation of private service interfaces that are used for AIS implementations

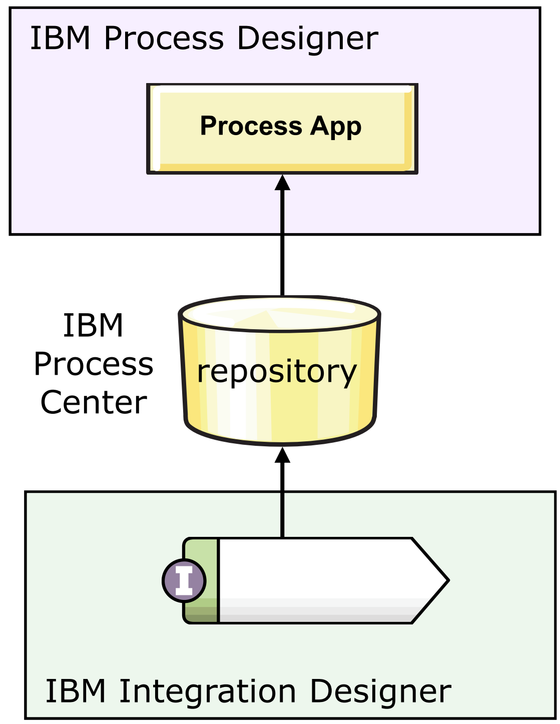


Figure 3-12. Using Advanced Integration services WB868 / ZB8681.0

Notes:

In IBM BPM, Advanced Integration services (AIS) can be used to author complex integration logic that can be reused in building BPM human centric processes. Reuse of AISes in BPM applications (Process Apps) is a desirable goal and should always be pursued when designing and packaging AISes.

When a Process App reuses an AIS, it must include the Toolkit where the AIS is implemented. Since each process application that is deployed to the runtime deploys its own copy of the Toolkits that it includes for implementing AISes, it might result in many deployed business-level applications (BLAs). In extreme cases, this result might negatively impact the server performance and make the Admin Console task more difficult.

This article describes best practices and a design pattern for efficient runtime implementation of reusable AISes.

Process application reuse results in an EAR being deployed because process applications use toolkits “by copy” not “by reference”.

The recommendation is to use the “Facade” pattern:

- To solve the excessive number of EAR deployed to Process Server

- To promote isolation of private service interfaces that are used for AIS implementations

For more information, see the developerWorks article: “Implementing the facade pattern using IBM Business Process Manager Advanced V7.5”:
`http://www.ibm.com/developerworks/websphere/bpmjournal/1112_pacholski/1112_pacholski.html`

WebSphere Education IBM

Common best practices

- Choose the appropriate granularity for a process
  - Use POJOs or Java snippets for logic without business significance
- Use events judiciously
  - Event emission can consume significant processor resources
  - Events that signify changes in state are important such as when long-running activities complete
  - Two events are enough for a microflow start and end
  - PMI is more appropriate for IT monitoring
- Use the right process for the job
  - For human centric processes, use BPMN
  - For straight-through processes that contain no human steps, use BPEL
  - For a process that does not need to wait on asynchronous services, use BPEL microflows
  - For a process that might span a long time, use BPEL macroflows

Figure 3-13. Common best practices WB868 / ZB8681.0

Notes:

A business process and its individual steps should have business significance and not try to mimic programming-level granularity. Use programming techniques such as plain old Java objects (POJOs) or Java snippets for logic without business significance.

WebSphere Education

Minimize conflicting changes

- Minimize conflicting changes between Process Designer and Integration Designer
  - Business objects that are defined in Process Designer can be modified in Integration Designer and vice versa
  - Although the Process Center can be used to synchronize these changes, concurrent updates might generate conflicts and confusion
- A solution to this problem is to establish an owner of the shared data models
  - Package data models in toolkits
  - Use data models by referencing toolkits

Figure 3-14. Minimize conflicting changes WB868 / ZB8681.0

Notes:

WebSphere Education

Expose only process-relevant data to Process Designer

- Resist the temptation to expose large data models in Process Designer resulting in exceptionally large process variables
  - This practice leads to hard-to-maintain processes and poor performance
- Ideally, process variables should contain only the information necessary for process navigation (decisions) and visibility (monitoring)
  - Everything else should be saved off to databases, or retrieved from the system of record
  - Use the Claim Check pattern; carry around only the primary key of large business objects
- Use Integration Designer
  - Then, you can keep your non-business data in Integration Designer and hide it by using the Mirrored Library concept

Figure 3-15. Expose only process-relevant data to Process Designer WB868 / ZB8681.0

Notes:

In a collaborative development environment between Integration Designer and Process Designer, artifacts like business objects are shared in libraries. Library mirroring means that when you put an artifact in your library within Integration Designer, it is made available to others. It is made available to others who are working with the same library in Process Designer and are using the same process application or toolkit.

When you bring a process application or toolkit into your workspace, the libraries that are associated with them have library mirroring enabled. You cannot change the setting.

When you associate a library in your workspace with a process application or toolkit that you brought into your workspace, library mirroring is not enabled by default. You can enable library mirroring if you want to share the artifacts in your library in Integration Designer with Process Designer.

3.2. Process Designer authoring best practices

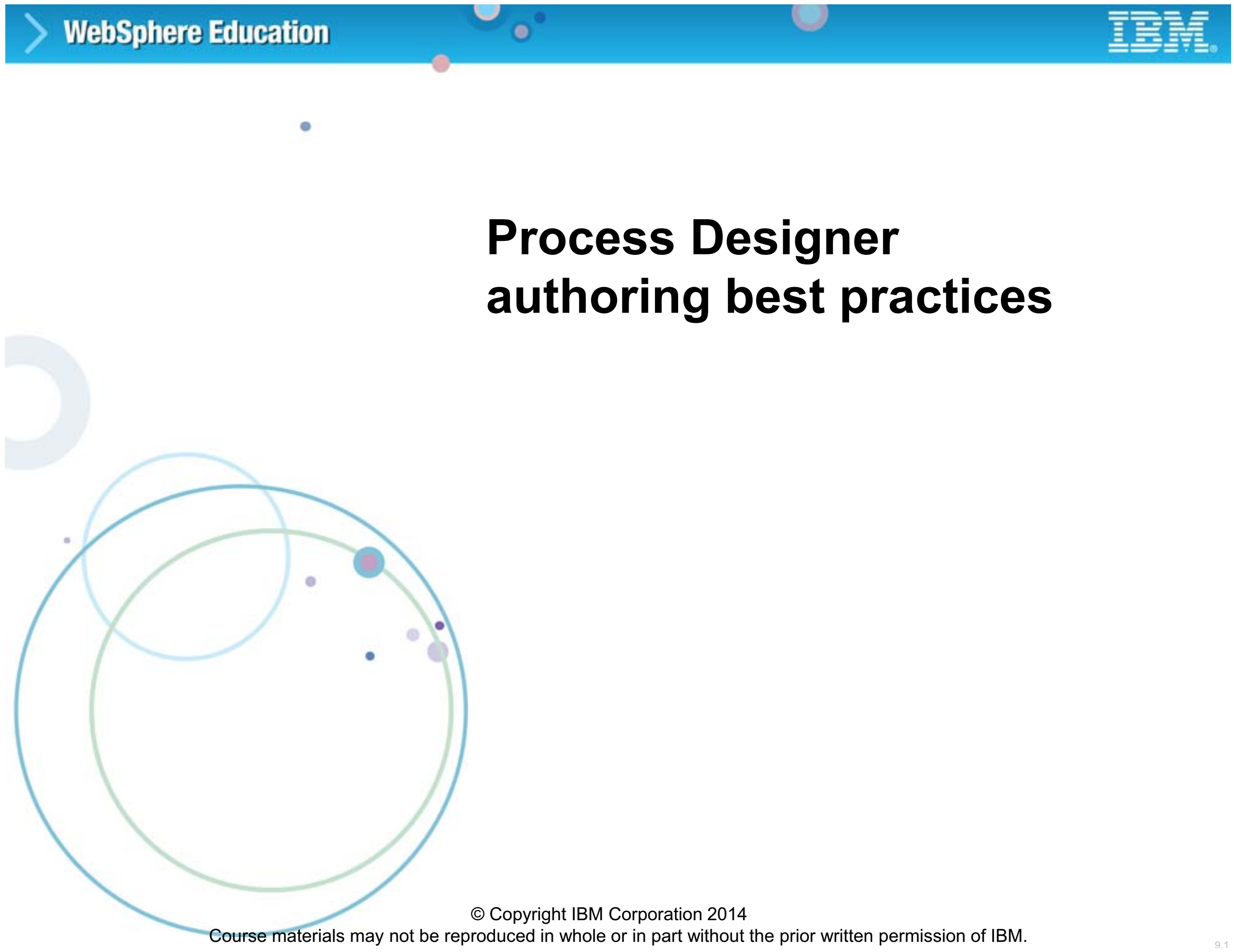


Figure 3-16. Process Designer authoring best practices WB868 / ZB8681.0

Notes:

WebSphere Education

Process modeling guidelines in IBM Process Designer

- A process diagram or model is called a business process definition (BPD) in IBM Process Designer
- In general, a BPD should be as simple an abstraction as you can make it
  - A highly conceptual BPD is resilient to change
- Make sure that you use the Documentation area in the Properties tab for each element in IBM Process Designer to include important requirement notes

Figure 3-17. Process modeling guidelines in IBM Process Designer WB868 / ZB8681.0

Notes:

To model a process, you must create a business process definition (BPD). A BPD is a reusable model of a process, defining what is common to all runtime instances of that process model.

WebSphere Education

Process Designer best practices (1 of 4)

- Manage variable usage
  - Clear variables in exposed human services that are not intended to end
  - Minimize the persistence cost and heap cost
  - Use searchable business variables judiciously
- Do not use multi-instance loops in the system lane or for batch activities
  - System lane activities lock an event manager engine thread until the activity is complete, so creating multiple instances of the loop can cause deadlock in the system until these activities complete
- Use sequential system lane activities efficiently
  - Each system lane activity is a new Event Manager task that adds a task transition in the server
  - Use one system lane task that wraps the others to minimize the extra resources

Figure 3-18. Process Designer best practices (1 of 4) WB868 / ZB8681.0

Notes:

Maintaining large variables is expensive. Each database save cost is high, and the heap cost is high. Execution context saves happen everywhere, such as when moving from a BPD to the service engine, at each coach, and at any point where a save execution context is checked. If you do not need a variable any more, null it out.

Multi-instance loops can happen when using a sub BPD as the activity of a multi-instance.

WebSphere Education IBM

Process Designer best practices (2 of 4)

- Provide specific guidelines for error handling
  - Avoid global error handling in a service, which uses an excessive amount of server processor use and can result in infinite loops in coaches
  - Do not route the error back to the same activity in a BPD, which causes the server to thrash, use large amounts of processor time, and increase database processing
- Use conditional joins only when necessary
  - The BPD engine evaluates all possible upstream paths to determine whether tokens can arrive at the join
  - Evaluation is expensive for large, complex BPDs
- Use a fast connection between Process Designer and Process Center
  - Minimize network latency to provide optimal response times
  - Try to place Process Center in the same physical location as the database
  - Avoid using a single shared Process Center

Figure 3-19. Process Designer best practices (2 of 4) WB868 / ZB8681.0

Notes:

If you have a geographically disperse development team, it is better to have regional Process Centers than to have a single shared Process Center that remote Process Designer authoring clients access.

The Process Designer interacts frequently with the Process Center for authoring tasks. For this reason, minimize network latency to provide optimal response times. Place the Process Center in or near the same physical location as the Process Designer users. Process Designer clients that connect to an especially remote Process Center might experience slow performance and dropped connections.

WebSphere Education

Process Designer best practices (3 of 4)

- Minimize the use of Service Tasks
  - Call system lane tasks with Service No Task
- Prevent WSDL validation from causing slow web service integrations
  - Two causes of delays are slow responses in retrieving the WSDL and deeply nested WSDP include structures
  - Store a local copy of the WSDL either in the local file system or somewhere where HTTP responses are fast
- Package AIS and business objects that change frequently in separate toolkits
  - Use the service facade pattern
  - Prevents the undeploying and redeploying of the BLA tip
  - Improves Process Center performance

Figure 3-20. Process Designer best practices (3 of 4) WB868 / ZB8681.0

Notes:

A business process definition (BPD) can include a lane for each system or team of users who participate in a process. A lane is the container for all the activities that a specific team or by a system carries out.

You can also create a swimlane as a system lane, indicating that the activities within it are for an IBM Business Process Manager system to complete. To make a swimlane a system lane, select the swimlane in the diagram and then select **Is System Lane** in the Properties view. Although you can create a system task in non-system lanes, any new tasks in the system lane are created as system tasks by default. After a system task is created, if you move the task to a non-system lane, for example a lane that is associated with a team, it retains a system task type.

WebSphere Education

Process Designer best practices (4 of 4)

- Disable auto-tracking in BPDs
  - Enabled by default for BPDs
  - Extra costs due to events that the Performance Data Warehouse is processing
  - Consider creating tracking groups to track key business events only
- Avoid BPDs that run perpetually
  - Uses numerous server processor resources by continually polling for new events
  - Use an undercover agent (UCA) instead of a BPD to do the polling
  - Design this BPD only if the capability is strictly required
- Minimize the use of large JavaScript scripts
  - Limit a JavaScript block to 50 lines
- Avoid direct SQL access to internal BPM tables

Figure 3-21. Process Designer best practices (4 of 4) WB868 / ZB8681.0

Notes:

Autotracking automatically captures data from tracking points at the entry and exit of each item in a BPD (for example, services, activities, and gateways).

WebSphere Education

Guidelines for using snapshots

- Snapshots can take up large amounts of space in a database
  - Agree on intervals in your organization and take snapshots at agreed-upon milestones
- Define a meaningful naming convention for snapshots and use that naming convention for all projects
- Work closely with IBM Business Process Manager administrators to come up with a snapshot deployment and activation plan

Figure 3-22. Guidelines for using snapshots WB868 / ZB8681.0

Notes:

Snapshots record the state of library items within a process application or track at a specific point in time. You can create snapshots in the Process Center Console or in the Designer view.

Snapshot management, such as installing, exporting, and archiving, is done in the Process Center Console.

WebSphere Education

Guidelines for Coach Views

- Coach Views are reusable asset that can be shared with multiple coaches or even multiple process applications
- Coach Views and coaches can share parts of their user interface with other Coach Views and coaches
- In general, create highly reusable Coach Views in toolkits and more specialized Coach Views in process applications
  - If the Coach View is in a toolkit and then someone edits it, the changes apply to all instances of the Coach View in all applications that use that toolkit
  - You cannot directly edit the definition of the coach view from within the parent coach or Coach View
  - Instead, you must open the Coach View definition first before you can change it
- Performance considerations for coaches are covered more in Unit 7

Figure 3-23. Guidelines for Coach Views WB868 / ZB8681.0

Notes:

Coach Views are reusable sets of user interfaces that Process Portal users use to interact with a business object or service. Coach Views can consist of one or more other Coach Views, data bindings, layout instructions, and behaviors.

Because Coach Views are reusable, Coach Views and coaches can share parts of their user interface with other Coach Views and coaches. For example, suppose you create a coach that has a Coach View that contains a set of address fields. If you create a second coach that needs address fields, you can reuse the Coach View from the first coach. In both cases, the coach is using an instance of the Coach View. You can edit the properties of each instance independently. For example, changing the label of one Coach View instance does not change the label of the other. Both instances of the Coach View use a reference to point to the Coach View definition. This approach means that if the Coach View definition changes, you can see the change reflected in the instances of the Coach View.

3.3. Integration Designer development best practices

WebSphere Education

IBM

Integration Designer development best practices

9.1

Figure 3-24. Integration Designer development best practices WB868 / ZB8681.0

Notes:

WebSphere Education

Integration Designer best practices (1 of 2)

- Choose microflows where possible
- Use share-by-reference libraries where possible
- Verify that the content in toolkits is needed for multiple applications
- Choose query tables for task list and process list queries
- Choose efficient metadata management
- Choose between business state machines and business processes

Figure 3-25. Integration Designer best practices (1 of 2) WB868 / ZB8681.0

Notes:

Microflows are short-lived BPEL processes. They can run either in a transaction, or in an activity session as specified on the SCA component of the microflow. Microflows that are executed as part of a transaction are explained here.

Microflows are not interruptible. Therefore, a microflow cannot contain activities that wait for an external event, or for a user interaction, for example, human task activities.

Microflows are transient. The process instance state of a microflow is held in memory, and not stored in the runtime database. However, the state of a microflow instance can be persisted in the audit log.

WebSphere Education

Integration Designer best practices (2 of 2)

- Identify performance considerations for:
  - Advanced content
  - Business object parsing modes
  - Service Component Architecture
  - BPEL business processes
  - Human tasks
  - Business processes and human task clients
  - Transactions
  - Invocation styles
  - Mediation flow components

Figure 3-26. Integration Designer best practices (2 of 2) WB868 / ZB8681.0

Notes:

WebSphere Education

Choose microflows where possible

- Microflows
  - Synchronous interactions
  - Typically last less than 2 minutes
  - State is transient
  - Provides significantly improved performance at run time
- Macroflows
  - Synchronous or asynchronous interactions
  - Fine-grained transaction control
  - Typically run for a long duration possibly weeks, months, or years
  - State is persisted to the database
- If macroflows are required for some capabilities, separate the processes
  - Use microflows for the most frequent scenarios
  - Use macroflows for exceptional cases

Figure 3-27. Choose microflows where possible WB868 / ZB8681.0

Notes:

A microflow runs in one transaction. However, the services that the microflow invokes can involve more than one transaction. This capability is because a service that is called through an invoke activity can either participate in the transaction of the microflow, or run in its own transaction.

A long-running BPEL process spans multiple transactions. Each transaction is triggered from either a Java Messaging Service (JMS) message or a work-manager-based implementation.

WebSphere Education IBM

Use share-by-reference libraries where possible

- Libraries are projects that are used to store shared resources that are shared between several modules
  - Libraries include dependencies, integration logic, data, interfaces, and transformations
- Libraries can be deployed in several ways:
  - With a process application
  - With the dependent module (default setting)
  - Globally
- Share-by-reference libraries are associated with the deployed process application
  - Only one copy is in memory
- Deploying libraries globally saves memory, but you must deploy each library independently

Figure 3-28. Use share-by-reference libraries where possible WB868 / ZB8681.0

Notes:

Often, interfaces, business objects, data maps, mediation subflows, relationships, roles, and web service ports need to be shared so that resources in several modules can use them. The library is a project that is used to store these resources.

Unlike modules or mediation modules, library projects can share resources. There are two kinds of sharing here, sharing of artifacts for development and sharing of artifacts within the solution at run time. In order for a module or mediation module to use the resources from a library at run time, the library must be added as a dependent to the module. You can add a library to the module and select to deploy it with the module, or you can deploy it as a global library. Also, you can add library dependencies to a library; for example, if one library uses resources in another library, then you would need to add the library dependency.

Scenarios that the share-by-reference library type supports:

- Module depends on a share-by-reference library
- Share-by-reference library depends on another share-by-reference library
- Share-by-reference library depends on a global library
- Module depends on a Java EE project

- Module depends on a Java project
- Global library depends on a Java project
- Module or Java project can contain customer JAR files

Scenarios that the share-by-value library type supports but not supported by share-by-reference library type:

- A JAR file can be inside a library
- Library depends on a Java project

WebSphere Education IBM

Choose query tables for task list and process list queries

- Query tables are designed to provide good response times for high-volume task lists and process list queries
  - Use composite query tables in place of the standard Business Process Choreographer query APIs
- Query tables offer improved query performance in the following ways:
  - Improved access to work items, reducing the complexity of the database query
  - Configurable high-performance filters for tasks, process instances, and work items
  - Composite query tables to bypass authorization through work items
  - Composite query tables that allow query table definitions that reflect information that is shown in task lists and process lists

Figure 3-29. Choose query tables for task list and process list queries WB868 / ZB8681.0

Notes:

Query tables support task and process list queries on data that is contained in the Business Process Choreographer database schema. This list includes human task data and BPEL process data that Business Process Choreographer manages, and external business data. Query tables provide an abstraction on the data of Business Process Choreographer that client applications can use. In this way, client applications become independent of the actual implementation of the query table. Query table definitions are deployed on Business Process Choreographer containers, and are accessible by using the query table API.

Query tables introduce a clean programming model for developing client applications that retrieve lists of human tasks and BPEL processes in Business Process Choreographer. Using query tables improves the performance. Information is provided about the query table API parameters and other factors that affect the performance.

WebSphere Education IBM

Choose between business state machines and business processes

- A business state machine is an implementation of a business process model that “executes” (moves from one state to another) based on real-time events
  - Focuses on the events that cause a transition from one state to another
  - State machines and business processes are both implemented in BPEL
- There are several types of transitions:
  - ***Automatic*** transitions are automatically followed
  - ***Operation*** (or event) transitions are triggered by calling an interface operation; a message that is sent from outside of the state machine triggers the transition
  - ***Duration*** transitions are triggered after a time that is set in a timeout
  - ***Expiration*** transitions are triggered after a deadline is elapsed (also by using a timeout)
- External even-driven state transitions are costly to performance
  - Minimize state transitions in business state machines
  - Automatic state transitions are much more costly than even-driven state transitions

Figure 3-30. Choose between business state machines and business processes WB868 / ZB8681.0

Notes:

A business state machine is an event-driven business application in which external operations trigger changes that guide the business state machine from one discrete mode to another. Each mode is an individual state, and this mode determines what activities and operations can occur.

The IBM Integration Designer tools are designed so that users can easily compose integrative business solutions without programming skills. To this end, you can easily create and develop business state machines in an intuitive graphical programming environment that is called the business state machine editor.

A transition is the movement that occurs through the recognition of an appropriate triggering event, and the evaluation of the conditions necessary for control to flow through it. It determines what actions can occur should processing be allowed. A transition is triggered when its triggering event occurs, and it is enabled when the business state machine enters the source state of the transition. Generally, a transition fires when it is enabled, triggered, and its condition evaluates to true. However, if the transition does not have a guard, then the first two conditions are sufficient.

WebSphere Education

Advanced content considerations

- Advanced content includes process applications or toolkits that are authored in Integration Designer
  - Takes much longer to deploy than standard content
  - Content is packaged as SCA modules and libraries
  - Deployed as BLAs and EARs to the runtime
- Deployments should be kept to a minimum
  - BLAs and EARs deployed consume resources on the server such as memory, disk space, and CPU cycles
- Consider the server operation costs during development
  - Minimize advanced content by integrating the content as late in the development cycle as possible
  - Try to minimize advanced content in toolkits
  - Include advanced content in toolkits only if every process application that references the toolkit needs the content

Figure 3-31. Advanced content considerations WB868 / ZB8681.0

Notes:

A process application has a business level application (BLA), which acts as a container for the process application and its assets (assets include things like SCA modules, toolkits, and libraries). Each process application snapshot has its own BLA. Many of the administration tasks for a snapshot (for example, stopping or starting it on a production server) are done at the level of the BLA. This task allows for quicker and simpler administration of the snapshot and all of its assets.

WebSphere Education

Business object parsing modes

- Set the parsing mode
  - Lazy parsing mode is the default setting
  - You can indicate either eager parsing, lazy parsing, or both
- Lazy parsing input XML stream is incrementally parsed
  - Properties are not populated until the application accesses them
  - Stream remains in memory through the lifetime of the corresponding business object
  - Delivers better performance than eager mode
- Eager parsing input XML stream is immediately and fully parsed
  - A complete in-memory data object representation
- Benchmark your application in both modes to determine which mode best suits your application characteristics

Figure 3-32. Business object parsing modes WB868 / ZB8681.0

Notes:

Integration Designer provides a property on modules and libraries that you can use to configure XML parsing mode for business objects to either eager or lazy.

- If the option is set to *eager*, XML byte streams are eagerly parsed to create the business object.
- If the option is set to *lazy*, the business object is created normally, but the actual parsing of the XML byte stream is deferred and partially parsed only when the business object properties are accessed.

In either XML parsing mode, non-XML data is always eagerly parsed to create the business object.

The business object parsing mode determines how XML data is parsed at run time. A business object parsing mode is defined on a module or library when it is created. You can change the parsing mode of the module or library; however, you should be aware of the implications.

WebSphere Education IBM

SCA development best practices

- Reduce the number of SCA modules when appropriate
  - Performance is key but consider maintainability, version requirements, and module ownership
  - Package modules in a single module when components are placed together in a single-server instance
- Cache the results of `ServiceManager.locateService()`
- Use synchronous SCA bindings across local modules
  - Modules are deployed within the same server JVM
- Use multi-threaded SCA clients to achieve concurrency
- Add quality of service qualifiers at the appropriate level
  - Do not use at the interface level if it is not needed for all operations of the interface

Figure 3-33. SCA development best practices WB868 / ZB8681.0

Notes:

Quality of service qualifiers are specified for interfaces, partner references, and implementations in the assembly editor. You can add qualifiers to supplement or replace the ones that the editor generates.

The system tries to anticipate your needs by generating qualifiers when you add interfaces and partner references and when you generate an implementation or synchronize from an implementation. You can also add your own qualifiers to the module.

WebSphere Education

BPEL modeling best practices

- Use the `Audit Logging property for Business Processes only` setting if you must log events in the database
- Do not use the `Enable persistence and queries of business-relevant data` property for the process and each activity
  - This property increases the load on the database and the amount of data that is stored for each process instance
- Use the setting `Participates` on all activities for better throughput performance
- Use human tasks only if necessary
  - Specify human tasks in business processes
  - Use group work items when multiple users are involved

Figure 3-34. BPEL modeling best practices WB868 / ZB8681.0

Notes:

SCA invocation styles review

- SCA components can call services synchronously or asynchronously, depending upon the preferred interaction style that is defined in the interface
  - The **synchronous** method is called: `as invoke`
  - The **asynchronous** method is called: `as invokeAsynch`
- A synchronous call waits for the return value before proceeding
- There are three kinds of asynchronous invocations:
  - **One-way**: No response is expected (“fire and forget”)
  - **Deferred response**: Caller fetches the response later by using a ticket
  - **Request with callback**: Callee sends the response back to the caller when the result becomes available

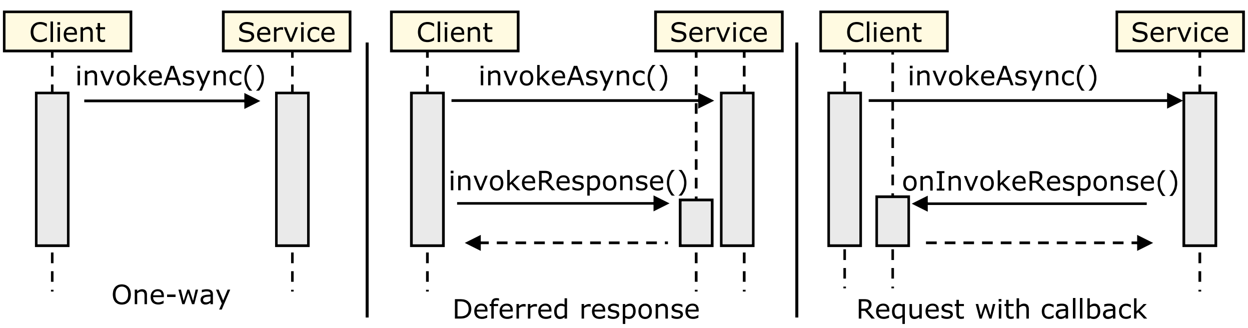


Figure 3-35. SCA invocation styles review WB868 / ZB8681.0

Notes:

One-way:

- A low number of clients can flood downstream components.
- A large downstream thread pool might be appropriate if CPU capacity is available.

Deferred response:

- Further processing on the calling thread is likely to be minimal (the calling thread is likely to block).
- Since calling threads eventually block, no flooding is possible.
- Implicitly, the number of calling threads is the right number of threads to handle responses (no additional configuration is required for responses).

Request with callback:

- Target thread configuration does not affect the number of threads to handle callbacks.
- If many downstream threads are configured, a similar number of threads might be required to handle the callbacks (responses).

WebSphere Education

BPEL considerations

- Avoid two-way synchronous invocations of long-running processes
  - This function ties up the threads and transactions of the caller until the process completes
  - Introduces difficulties when exceptions occur
- Ensure that callers of a two-way (request/response) interface to not use synchronous interactions
  - Use asynchronous interactions instead
  - Use a one-way synchronous call where no response is expected
- Minimize the number of size of BPEL variables and business objects
  - Each commit saves modified variables to the database
  - Smaller business objects are more efficient to process
  - Improve runtime performance by using data type variables

Figure 3-36. BPEL considerations WB868 / ZB8681.0

Notes:

When designing long-running business process components, ensure that callers of a two-way (request/response) interface do not use synchronous semantics because this function ties up the caller's resources (such as threads and transactions) until the process completes. Instead, start such processes either asynchronously, or through a one-way synchronous call, where no response is expected. In addition, calling a two-way interface of a long-running business process synchronously introduces difficulties when exceptions occur. Suppose that a non-interruptible process calls a long-running process by using the two-way request/response semantics, and the server fails after the long-running process completes, but before the caller's transaction is committed.

WebSphere Education IBM

Human task considerations

- Use group work items for large groups
  - Instead of individual work items for group members
- Use native properties rather the custom properties where possible
- Set the transaction behavior to `commit after` if the task is not part of a page flow
  - This setting improves the response time of task complete API calls

Figure 3-37. Human task considerations WB868 / ZB8681.0

Notes:

WebSphere Education

Business process and human task client considerations (1 of 2)

- Do not frequently call APIs that provide process and task details
- Do not put too much work into a single client transaction
  - Calls to the BPM and HTM APIs directly are typically not a concern for transaction size
  - Long-running transactions create long-lasting locks in the database
- Choose the protocol that best suits your needs
  - In a Java EE environment, use the BFM and HTM EJB APIs
  - If the client application is running on a BPM server, use the local EJB interface
  - In a Web 2.0 application, use the REST API
  - In an application that runs remote to the process container, use the web services API

Figure 3-38. Business process and human task client considerations (1 of 2) WB868 / ZB8681.0

Notes:

Do not frequently call APIs that provide task details and process details, such as `htm.getTask()`. Use these methods only when required; for example, to show the task details of a single task.

WebSphere Education IBM

Business process and human task client considerations (2 of 2)

- Clients that present both process and task lists should consider the following options:
  - Use query tables for queries
  - Do not loop over the tasks that are shown in the list and run an extra remote call for each object
  - Design the application so that all information is retrieved from a single query table

Figure 3-39. Business process and human task client considerations (2 of 2) WB868 / ZB8681.0

Notes:

Do not loop over the tasks that are shown in the task or process list and run an extra remote call for each object. This practice prevents the application from providing good response times and good scalability.

WebSphere Education

Invocation style considerations (1 of 2)

- Synchronous invocations are faster than asynchronous invocations since asynchronous bindings require queuing
  - Use asynchronous invocation judiciously
- Set the `Preferred Interaction Style` to synchronous invocation when possible
  - Synchronous cross-component invocations perform better
  - Use asynchronous for situations where you have a long-running process or where the target component requires it
- Minimize cross-component asynchronous invocations within a module
  - Intramodule asynchronous calls are **queued**
  - Intermodule asynchronous calls encounter more **queues** than for intramodule asynchronous calls: at the import and at the target (export)

Figure 3-40. Invocation style considerations (1 of 2) WB868 / ZB8681.0

Notes:

The Invocation Style property of the Service Invoke mediation primitive or the Callout node determines the invocation style that is used to invoke a service.

Within Integration Designer for a one-way operation that is using a Callout node or Service Invoke mediation primitive, the options for the invocation style property on the Details page are:

- As target
- Sync
- Async one way

WebSphere Education IBM

Invocation style considerations (2 of 2)

- Performance considerations for asynchronous invocation of synchronous services in a FanOut/FanIn block
  - Do not select deferred response for services with synchronous bindings unless there is an overriding need
  - In addition to performance implications, consider reliability and transactional aspects
  - If you need to guarantee that the service is called only once, do not use asynchronous invocation
  - To optimize the overall response time when calling services asynchronously, start the services in order of expected latency
  - Change the quality of service-on-service references to `Best Effort` to gain a substantial reduction in asynchronous processing time

Figure 3-41. Invocation style considerations (2 of 2) WB868 / ZB8681.0

Notes:

Do not select asynchronous (deferred response interaction) service invocations for services with synchronous bindings (for example, web services) unless there is an overriding need and the non-performance implications for this style of invocation are understood.

Apart from the performance implications of calling a synchronous service asynchronously, you must consider reliability and transactional aspects. Generally, use asynchronous callouts only for idempotent query type services. If you need to ensure that the service is called only one time, do not use asynchronous invocation.

WebSphere Education IBM

Transactional considerations

- Design the transaction boundaries as dictated by the needs of the application
- Take advantage of SCA transaction qualifiers
  - Reduce the number of transaction boundaries by allowing transactions to propagate across components
  - Use `SuspendTransaction=false` for the reference of the calling component
  - Use `joinTransaction=true` for the interface of the called component
  - Use `Transaction=any|global` for the implementation of both components
- Avoid two-way synchronous invocation of an asynchronous target
- Take advantage of transactional attributes for activities in macroflows
  - Change the scope and number of transactions
  - Settings include Commit before, Commit after, Participates, and Requires own

Figure 3-42. Transactional considerations WB868 / ZB8681.0

Notes:

The following are SCA transaction qualifiers that are specified for the process and the service that is called:

- The **suspendTransaction** qualifier on the reference of the process component specifies whether the transaction context of the process is propagated to the services to be invoked.
- The **joinTransaction** qualifier on the service interface specifies whether a service participates in the transaction of its caller if a transaction is propagated.

WebSphere Education

Mediation flow component considerations

- Use Extensible Stylesheet Language Transformation (XSLT) primitives versus business object maps
  - If no XSLT-specific function is required, use business object maps
- Use aggregation blocks design
  - It enables a single inbound request to map into multiple outbound service invocations
  - Responses can be aggregated into a single response to the original request
- Configure mediation flow component resources
  - Options include preconfigured to let the tool generate the resources
  - Preconfigured resources allow existing resources to be used and external creation and tuning scripts to be applied
  - Preconfigured resources allow for easier post-deployment adjustments
  - Tooling-created resources are suitable if there are no further needs for creating resources

Figure 3-43. Mediation flow component considerations WB868 / ZB8681.0

Notes:

The XSLT primitive offers two alternative transformation approaches in a mediation flow. If no XSLT-specific function is required, then it is generally better to use the business object map primitive, which can be faster. The exception is when a Mediation Flow Component is trivial in nature and the transformation is taking place at the body level of the service message object (SMO). In this case, XSLT is faster because the native XML bytes can be passed straight to the XSLT engine.

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Explain general development best practices
- Define the differences between BPMN and BPEL
- Explain Process Designer best practices
- Explain Integration Designer best practices

Figure 3-44. Unit summary WB868 / ZB8681.0

Notes:

WebSphere Education

Checkpoint questions

1. True or false: IBM Integration Designer requires a live connection to IBM Process Center.

2. True or false: Asynchronous one-way, asynchronous deferred response, and asynchronous callback are the three asynchronous invocation styles that SCA supports.

3. True or false: Advanced content includes process applications or toolkits that are authored in Integration Designer and are deployed as BLAs and EARs to the runtime.

Figure 3-45. Checkpoint questions WB868 / ZB8681.0

Notes:

Write your answers here:

1.

2.

3.

WebSphere Education

Checkpoint answers

1. False. IBM Process Designer requires a live connection to IBM Process Center. IBM Integration Designer does not require a connection to a server.

2. True

3. True

Figure 3-46. Checkpoint answers WB868 / ZB8681.0

Notes:

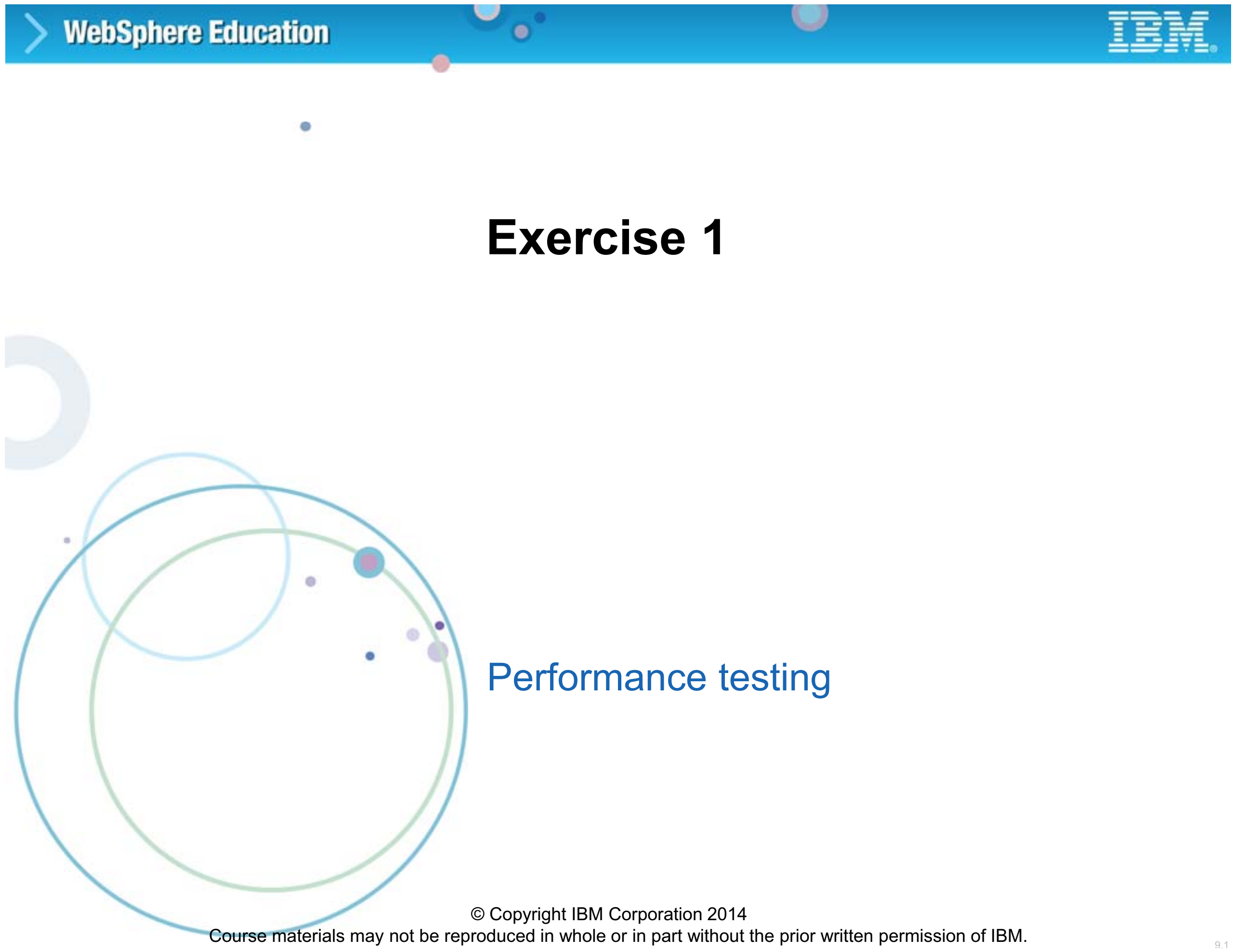


Figure 3-47. Exercise 1 WB868 / ZB8681.0

Notes:

WebSphere Education

Exercise objectives

After completing this exercise, you should be able to:

- Identify various aspects of valid performance benchmarks
- Describe the effect of threading in SCA applications that are running in the environment
- Explain the effect of target application response time
- Describe the differences in performance between synchronous and asynchronous SCA inter-component interactions

Figure 3-48. Exercise objectives WB868 / ZB8681.0

Notes:

Unit 4. WebSphere monitoring and tuning concepts

What this unit is about

This unit provides an introduction to the areas in WebSphere that can be monitored and tuned for Business Process Manager.

What you should be able to do

After completing this unit, you should be able to:

- Identify the areas of the runtime environment that need to be tuned
- Evaluate a list of the top ten monitoring considerations
- Detect performance bottlenecks
- Determine optimum queue sizes

How you will check your progress

- Checkpoint

References

IBM Business Process Manager V8.5 IBM Knowledge Center

`http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/ditamaps/ic-homepage-bpm.html`

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Identify the areas of the runtime environment that need to be tuned
- Evaluate a list of the top ten monitoring considerations
- Detect performance bottlenecks
- Determine optimum queue sizes

Figure 4-1. Unit objectives WB868 / ZB8681.0

Notes:

WebSphere Education

Topics

- Performance tuning in practice
- Top 10 monitoring hot list for WebSphere performance tuning
- Detecting bottlenecks
- System queues

Figure 4-2. Topics WB868 / ZB8681.0

Notes:

4.1. Performance tuning in practice

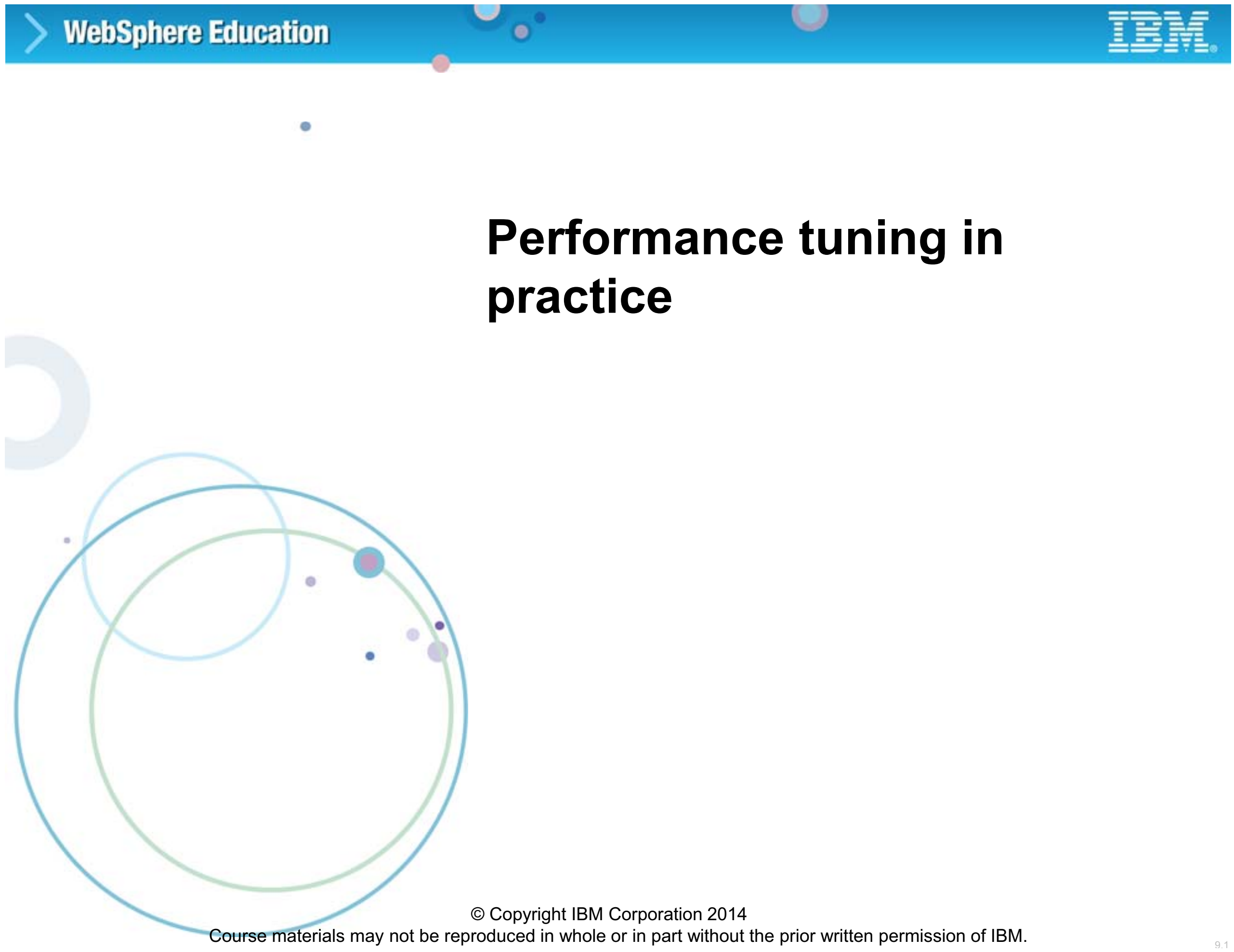


Figure 4-3. Performance tuning in practice WB868 / ZB8681.0

Notes:

WebSphere Education

Preparation for performance tuning

- Before you start tuning and managing the performance of any system
  - Understand the system components and how they can affect the overall performance
- To tune performance in a WebSphere Application Server environment:
  - Know every part of the system that is involved in request flow of your application
- Maintaining good performance requires you to:
  - Understand your deployment environment
  - Plan for scalability and capacity
  - Meet performance requirements for throughput and response time

Figure 4-4. Preparation for performance tuning WB868 / ZB8681.0

Notes:

Before you can start tuning and managing the performance of any system, you must understand your system's components and how they can affect system overall performance. To tune performance in a WebSphere Application Server environment, you must know every part of your system that is involved in this process. Maintaining good performance requires you to understand, plan, manage performance policies, and track them.

WebSphere Education

Performance tuning goals

- Improve throughput
  – Maximize the number of transactions that are served in a certain amount of time
- Improve response time
  – Minimize the time that it takes to respond to client requests
- Performance expectations are often expressed in terms of “number of clients”
  – Need to translate those figures in terms of throughput and response time

Figure 4-5. Performance tuning goals WB868 / ZB8681.0

Notes:

The goals of performance tuning are as follows:

- Improve throughput to maximize the number of transactions that are served in a certain amount of time.
- Improve response time to minimize the time it takes to respond to clients’ requests.
- Performance expectations are often expressed in terms of “number of clients.” You need to translate those figures in terms of transactions/hour and response time.

WebSphere Education

Application server parameter tuning

- The art of changing WebSphere Application Server settings with the goal of improving performance
- WebSphere Application Server offers an extensive list of tuning knobs and parameters that can be used to enhance the performance of an application
- Default values for the most commonly used tuning parameters in the server are set to ensure adequate performance, without customization, for the broadest range of applications
  - No two applications are alike and use the server in the same fashion
  - Cannot guarantee that a single set of tuning parameters is perfectly suited for every application
  - Therefore, it is important to conduct focused performance testing and tuning against your own application

Figure 4-6. Application server parameter tuning WB868 / ZB8681.0

Notes:

There are two types of tuning:

- **Application tuning** sometimes offers the greatest tuning improvements; this course first focuses on tuning individual performance parameters and the interactions between them.
- **Application server parameter tuning** is the art of changing WebSphere Application Server settings with the goal of improving performance.

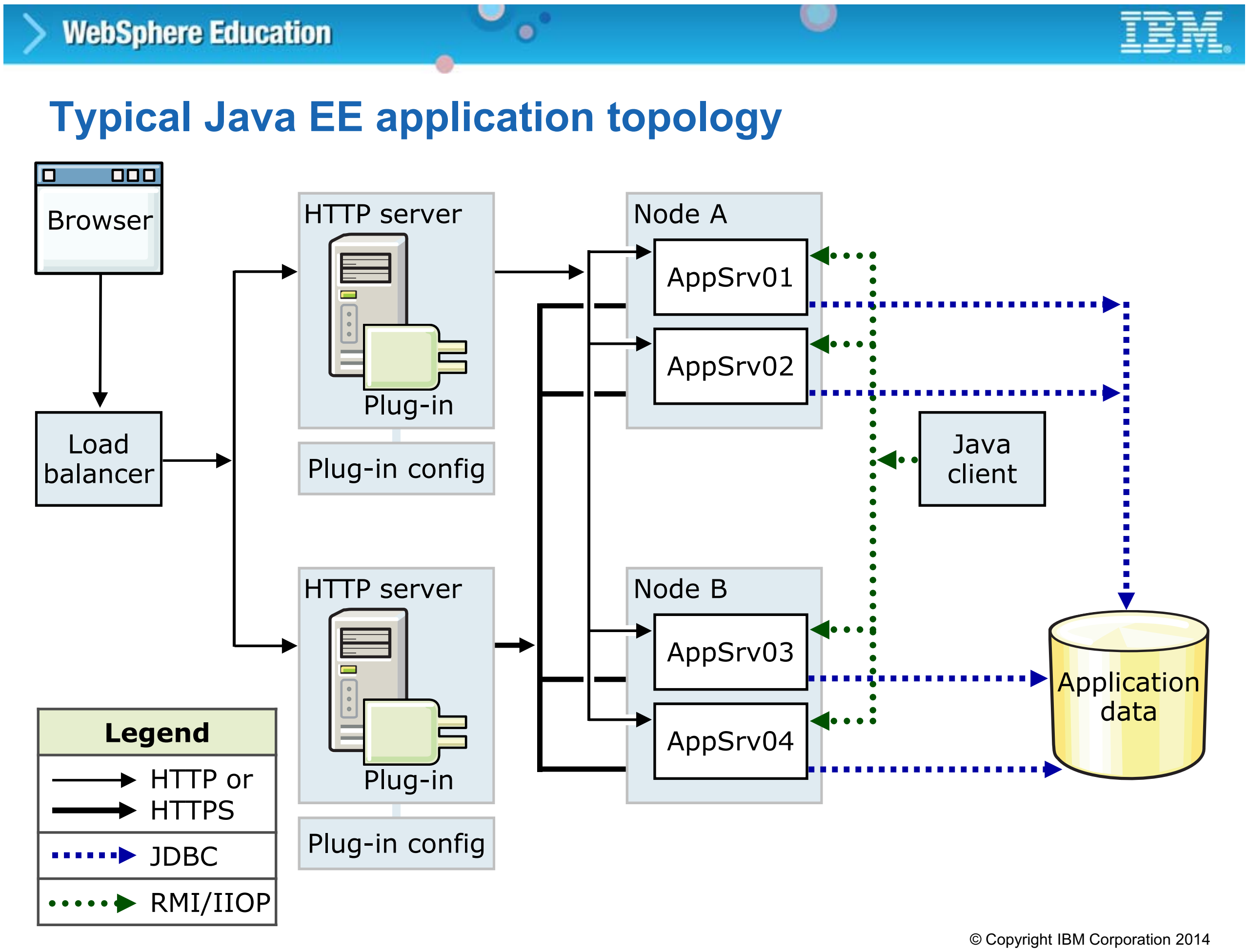


Figure 4-7. Typical Java EE application topology WB868 / ZB8681.0

Notes:

In a typical Java EE application topology, one or more of the following components can cause performance issues:

- Web server
- Load balancer
- Firewalls
- Hardware
- Network
- Scaling
- Application server JVM
- Application design
- EJBs
- Database

- JMS queues

The problem can be almost anywhere within an application. The problem can be network and hardware-related, back-end, or system-related. The problem can be actual product bugs, or often, application design issues.

With increasingly complex applications, it is getting more difficult to track down these problems. Typical e-business applications now have one or several of the following attributes:

- Multiple logical or physical tiers
- Mixture of operating system platforms
- Mixture of hardware architectures
- Fuzzy application boundaries

Software can produce massive quantities of numbers: CPU usage statistics, memory usage statistics, queuing statistics, transaction rate statistics, buffer pool statistics, and more. The problem is that the large number of statistics can be overwhelming and might hamper, rather than help, a person’s ability to understand and analyze performance data.

WebSphere Education

What influences tuning?

- The following are parameters that can affect the performance of WebSphere Application Server:
  - The application that is used
  - Hardware capacity and settings
  - Operating system settings
  - Web server
  - WebSphere Application Server process
  - Java virtual machine (JVM)
  - Database server
- Each parameter has its own tuning options, varying in importance and effect on performance

Figure 4-8. What influences tuning? WB868 / ZB8681.0

Notes:

The following are parameters that can affect the performance of WebSphere Application Server:

- The application that is used
- Hardware capacity and settings
- Operating system settings
- Web server
- WebSphere Application Server process
- Java virtual machine (JVM)
- Database

WebSphere Education

System view

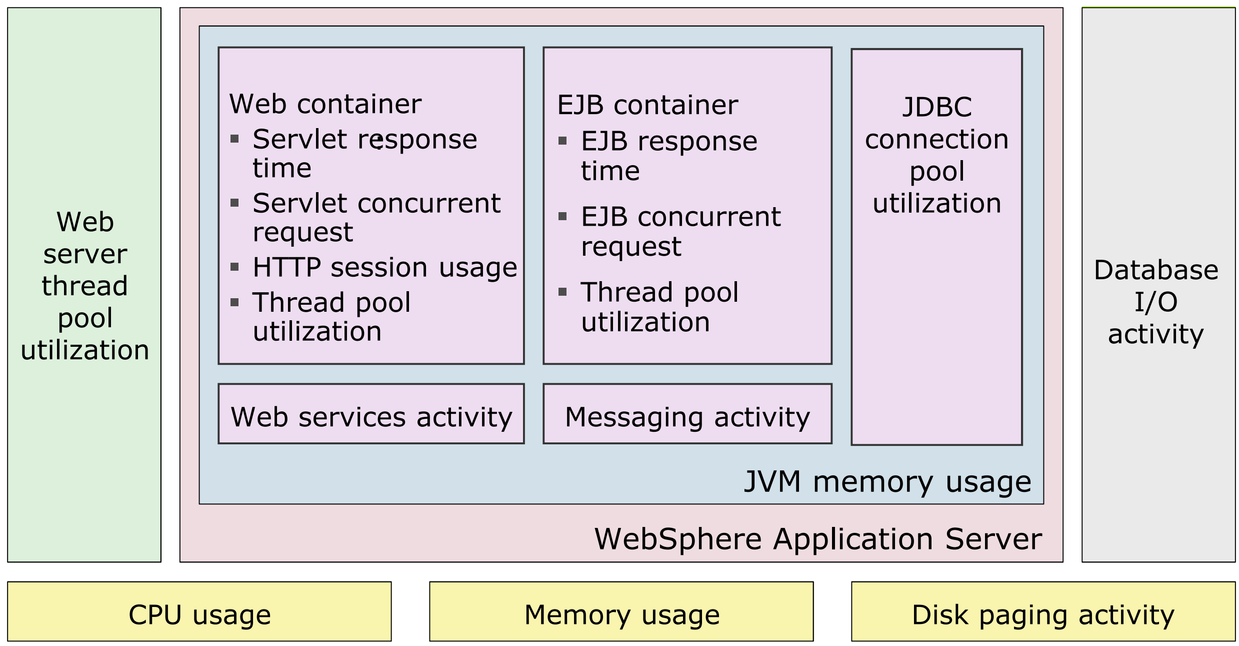


Figure 4-9. System view

WB868 / ZB8681.0

Notes:

The system view focuses on the overall health of the system by looking at critical system components and resources to make sure that they are running correctly. If a problem occurs, it helps identify which resources are constrained and should be investigated further.

WebSphere Education

System view concerns (1 of 2)

- Core system performance:
  - Is the CPU over-utilized?
  - Is the system paging memory to disk excessively?
  - Is much time spent accessing the disk?
- Web server health:
  - How many requests is the web server are currently processing (web server thread pool utilization)?
- WebSphere Application Server health:
  - How much time does the JVM spend doing garbage collection?
  - How many requests are the web or EJB containers currently processing (container thread pool utilization)?

Figure 4-10. System view concerns (1 of 2) WB868 / ZB8681.0

Notes:

Core system performance: Is the CPU over-used? Is the system paging memory to disk excessively? How much time is spent accessing the disk?

Web server health: How many requests is the web server currently processing (web server thread pool use)?

WebSphere Application Server health: How much time does the JVM spend in doing garbage collection? How many requests are the web or EJB containers currently processing (container thread pool use)?

WebSphere Education

System view concerns (2 of 2)

- Application resource health:
  - Is there a servlet or EJB component that runs unusually slowly (servlet or EJB response time)?
  - Is a servlet or EJB component over-utilized (servlet or EJB concurrent request)?
  - How many live HTTP session objects are currently allocated (HTTP session usage)?
  - Is a JDBC connection pool over-utilized?
- Database health:
  - Are the buffer pools sized adequately?
  - Are there enough listener agents?

Figure 4-11. System view concerns (2 of 2) WB868 / ZB8681.0

Notes:

Application resource health: Is there a servlet or EJB component that runs unusually slowly (servlet and EJB response time)? Is a servlet or EJB component over-utilized (servlet or EJB concurrent request)? How many live HTTP session objects are currently allocated (HTTP session usage)? Is a JDBC connection pool over-utilized?

Database health: Are the buffer pools sized adequately? Are there enough listener agents?

4.2. Top 10 monitoring hot list for WebSphere performance tuning

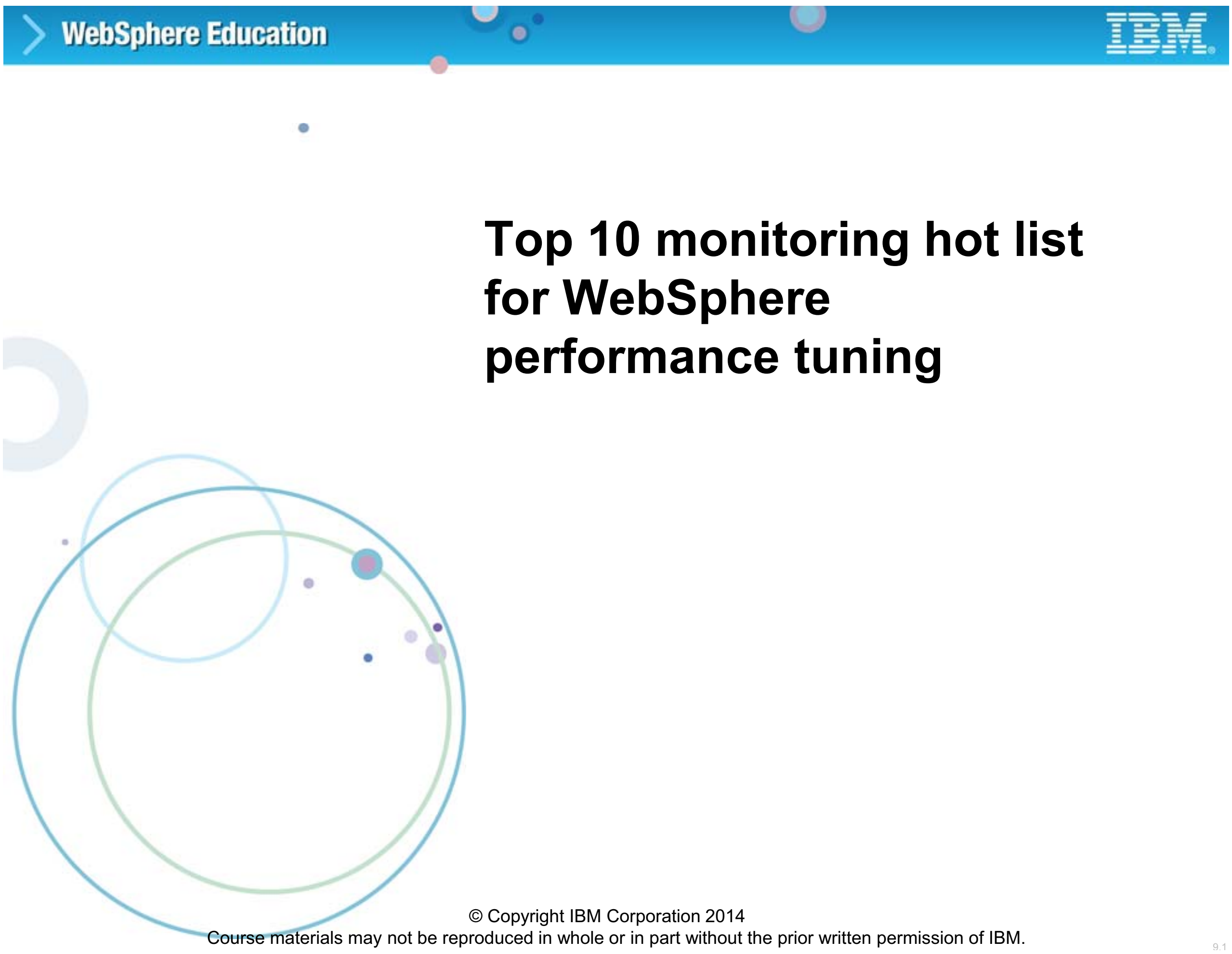


Figure 4-12. Top 10 monitoring hot list for WebSphere performance tuning WB868 / ZB8681.0

Notes:

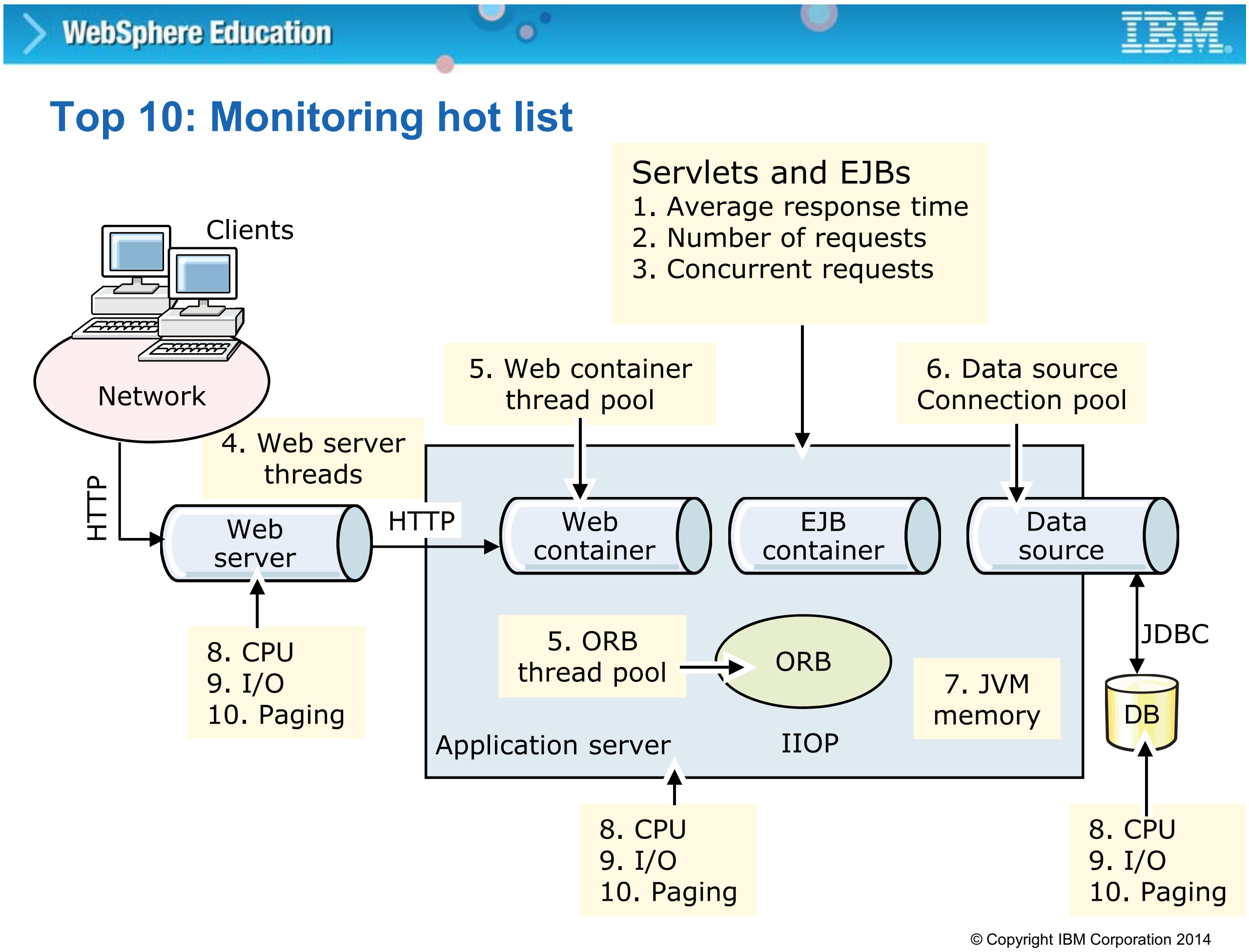


Figure 4-13. Top 10: Monitoring hot list WB868 / ZB8681.0

Notes:

From end-to-end, web browser client to back-end database, the top 10 list of things to monitor are:

1. Average response time for servlets and EJBs
2. Number of client requests for servlets and EJBs
3. Concurrent requests for servlets and EJBs
4. Number of web server threads used
5. Size of the web container thread pool
6. Size of data source connection pools
7. Size of JVM heap memory
8. CPU usage
9. Operating system I/O
10. OS paging

WebSphere Education

General tuning considerations (1 of 2)

- Because of the sheer magnitude of monitors and tuning parameters, it is difficult to know where to start
  - What should you monitor?
  - Which component should you tune first?
- When you experience performance problems:
  - Use the Tivoli Performance Viewer to walk through the top 10 monitoring items checklist
- Use the Performance Advisors together with Tivoli Performance Viewer
  - Load test your application
  - Monitor items by using Tivoli Performance Viewer and Health Center
  - Gather and implement performance advice

Figure 4-14. General tuning considerations (1 of 2) WB868 / ZB8681.0

Notes:

Because of the sheer magnitude of monitors and tuning parameters, it is difficult to know where to start.

- What should you monitor?
- Which component should you tune first?

When you experience performance problems:

- Use the Tivoli Performance Viewer to walk through the top 10 monitoring items checklist.

Use the Performance Advisors together with Tivoli Performance Viewer.

- Load test your application
- Monitor items with the Tivoli Performance Viewer
- Gather and implement performance advice

WebSphere Education

General tuning considerations (2 of 2)

- Make sure that you have a comprehensive understanding of the environment, the applications, and all components involved
- Before starting with WebSphere tuning, ensure that you do not have any performance bottlenecks at the OS level
  - Paging
  - Disk I/O
  - Network
- Focus your tuning actions on extraordinary behaviors such as:
  - An over-utilized thread pool
  - A JVM that spends 50% of its time in garbage collection during peak load
- Perform monitoring when the system is under typical production level load and record the observed values
  - Record Tivoli Performance Viewer sessions and save to the file system
  - Use the monitoring data for recurring analysis

Figure 4-15. General tuning considerations (2 of 2) WB868 / ZB8681.0

Notes:

Remember: A set of tuning parameters that work well for one application might not work the same way for another application. Tuning is specific to the application and the environment.

Server tuning cannot compensate for a poorly written application.

WebSphere Education IBM

How to find metrics for servlets and EJBs (1 of 2)

- Average servlet response time
  - In Tivoli Performance Viewer, click **Performance Modules > Web Applications** and view the value that is named `ServiceTime`
- Number of servlet requests (transactions)
  - In Tivoli Performance Viewer, click **Performance Modules > Web Applications** and view the value that is named `RequestCount`
- Concurrent servlet requests (extended statistics set)
  - In Tivoli Performance Viewer, click **Performance Modules > Web Applications** and view the value that is named `ConcurrentRequests`
- Live number of HTTP sessions
  - In Tivoli Performance Viewer, click **Performance Modules > Servlet Session Manager** and view the value that is named `LiveCount`

Figure 4-16. How to find metrics for servlets and EJBs (1 of 2) WB868 / ZB8681.0

Notes:

Definitions:

- ServiceTime: The average response time, in milliseconds, in which a servlet request is finished
- RequestCount: The total number of requests that a servlet processed
- ConcurrentRequests: The number of requests that are concurrently processed
- LiveCount: The total number of sessions that are currently live

WebSphere Education

How to find metrics for servlets and EJBs (2 of 2)

- From Tivoli Performance Viewer, select **Performance Modules > Web Applications**

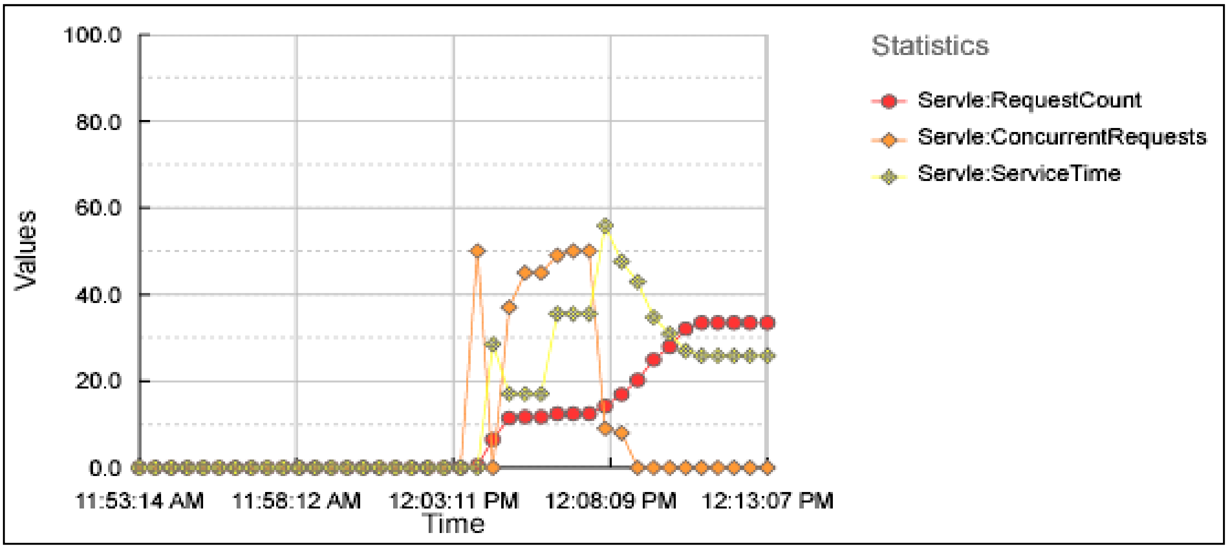


Figure 4-17. How to find metrics for servlets and EJBs (2 of 2) WB868 / ZB8681.0

Notes:

This graph was generated by running a load test of 50 current users. As you can see from the graph, the test ran for approximately 7 minutes, during which the service time gradually increased.

WebSphere Education

How to find metrics for thread pools and data sources

- Web server threads
  - Depends on the web server; consult the web server manuals
  - For IBM and Apache HTTP Server, use the “server-status” page
- Web container and EJB container thread pool
  - In Tivoli Performance Viewer, look at the Thread Pools summary report
- Data source connection pool size
  - In Tivoli Performance Viewer, look at the Connection Pools summary report
- To see the summary reports, open Tivoli Performance Viewer and expand Summary in the navigation tree

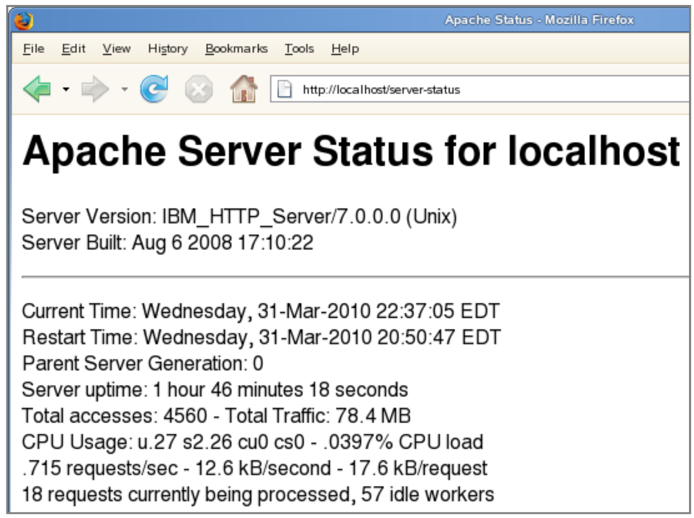


Figure 4-18. How to find metrics for thread pools and data sources WB868 / ZB8681.0

Notes:

Finding web server threads depends on the web server. Consult the web server manuals.

Use the Thread Pools summary report in Tivoli Performance Viewer to locate web container and EJB container thread pools.

Data source connection pool size can be found in the Connection Pools summary report.

To see the summary reports, open Tivoli Performance Viewer and expand Summary in the navigation tree.

For more information about tuning web servers, see the following article:

`http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp?topic=%2Fcom.ibm.websphere.nd.doc%2Fae%2Ftprf_tunewebserv.html`

WebSphere Education

How to find metrics for JVM and system resources

JVM memory and garbage collection statistics

- Some metrics depend on JVMTI
- Add `-agentlib:pmiJvmtiProfiler` as a Generic JVM argument
- In Tivoli Performance Viewer, go to **Performance Modules > JVM Runtime**
- Enable verbose GC for the application server
  - GC data is written to `native_stderr.log`

System resources on web, application, and database servers

- CPU utilization
- Disk and network I/O
- Paging activity
- Use OS-provided tools or third-party monitoring tools for system resource monitoring
- Per JVM process CPU utilization in Tivoli Performance Viewer Basic statistics set (`ProcessCpuUsage`)

Figure 4-19. How to find metrics for JVM and system resources WB868 / ZB8681.0

Notes:

JVM memory, garbage collection statistics:

- Some metrics depend on JVMTI.
- In Tivoli Performance Viewer, go to **Performance Modules > JVM Runtime**.
- Enable verbose GC for the application server. The data is written to `native_stderr.log`.

System resources on web, application, and database servers

- CPU utilization
- Disk and network I/O
- Paging activity
- Use of OS-provided tools or third-party monitoring tools for system resource monitoring
- Per-process CPU utilization in Tivoli Performance Viewer for Java processes

WebSphere Education

More considerations

- Disable functions that are not required
  - For example, if your application does not use the web services addressing (WS-Addressing) support, disabling this function can improve performance
- Ensure that the transaction log is assigned to a fast disk
  - Some applications generate a high rate of writes to the transaction log
  - Locating the transaction log on a fast disk or disk array can improve response time
- Enable the pass by reference option
  - Use applications that can take advantage of the pass by reference option to avoid the cost of copying parameters to the stack
- Tune the operating system
  - Operating system configuration plays a key role in performance
  - For example, adjustments such as TCP/IP parameters might be necessary for your application

Figure 4-20. More considerations WB868 / ZB8681.0

Notes:

Use this property with care because applications might require WS-Addressing MAPs to function correctly. Setting this property also disables WebSphere Application Server support for the following specifications, which depend on the WS-Addressing support: Web Services Atomic Transactions, Web Services Business Agreement, and Web Services-Notification.

4.3. Detecting bottlenecks

Detecting bottlenecks

Figure 4-21. Detecting bottlenecks WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Detecting bottlenecks (1 of 2)

- Always monitor OS resources: CPU, memory, and I/O
  - In general, your systems are CPU bound, meaning that adding resources (for example, another CPU or another node) can increase throughput
- If CPUs are not fully used under load, requests are probably waiting on some other resource
  - Are you sure that the HTTP server is running enough threads?
  - Database server might be undersized and therefore lowers overall performance
- Does the application establish network connections to external servers?
  - Those servers might be slow in response or not available on the network
  - Slow transactions on back-end systems consume application server resources (for example, web container threads)
  - Accommodate for such extra threads in thread pools or use async beans in code
- What about disk I/O?
  - Too much logging can kill performance
- Check the network performance

Figure 4-22. Detecting bottlenecks (1 of 2) WB868 / ZB8681.0

Notes:

Always monitor OS resources: CPU, memory, and I/O. Ideally, your systems are CPU bound, meaning that adding resources (for example, another CPU or another node) increases throughput.

CPU bound refers to a condition when the speed of the CPU determines the time that a computer takes to complete a task. During the task, CPU utilization is high or at 100% for several seconds. Interrupts that a peripheral device issues are processed slowly, or indefinitely delayed. When you determine that a computer is often CPU bound, this condition implies that upgrading the CPU can improve the overall computer performance.

If CPUs are not fully used under load, requests are probably waiting on some other resource. Are you sure that the HTTP server is running enough threads? The database server might be undersized and lowering overall performance.

Does the application establish network connections to external servers? Those servers might be slow in response or not available on the network at all. Slow transactions on back-end environments eat up resources (for example, web container threads).

Accommodate for such extra threads in thread pools or use async beans in code. What about disk I/O? Too much logging can impair performance. Be sure that you check the network performance.

WebSphere Education

Detecting bottlenecks (2 of 2)

- If CPU is fully used, does every process get enough CPU time?
  - Too many services and processes on a single machine compete for CPU time
  - SSL encryption of many connections consumes CPU resources
  - Bad application coding or bugs might consume CPU
  - Other running services (not WebSphere Application Server-related)

- Bottom line: Adding CPUs should help

Figure 4-23. Detecting bottlenecks (2 of 2) WB868 / ZB8681.0

Notes:

If CPU is fully used, does every process get enough CPU time?

- Too many services or processes on a single machine compete for CPU time
- SSL encryption of many connections eats up CPU resources
- Bad application coding
- Bugs in code
- Other running services (not WebSphere Application Server related)

Adding CPUs should help.

WebSphere Education IBM

Bottleneck considerations (1 of 4)

- Oversized queues and thread pools
  - Too many concurrent requests can flood the system and performance drops
- Lowering concurrent request processing usually gives better response times and thus increases throughput
- HTTP server is not accepting new connections
  - Increasing HTTP server processes (threads) might not help
  - Just postpones the problem
- Find the root cause of the bottleneck
  - Poor database performance
  - Security: Poor performance because of a slow authentication (LDAP) server
  - Garbage collection might be taking too long on application servers; tune the JVM heap
  - Hanging (synchronous) back-end connections

Figure 4-24. Bottleneck considerations (1 of 4) WB868 / ZB8681.0

Notes:

Oversized queues and thread pools: Too many concurrent requests can flood the system, and performance drops.

Lower concurrent request processing usually gives better response times and thus increases throughput and overall performance.

HTTP server is not accepting new connections: Increasing HTTP server processes or threads might not help it and thus can merely postpone the problem.

Find the root cause:

- Poor database performance
- Security: low performance because of slow authentication server
- Garbage collection is taking too long on application servers: tune the JVM heap
- Hanging (synchronous) back-end connections

WebSphere Education

Bottleneck considerations (2 of 4)

Database server

- Have your DBA check database performance

Most database servers have performance introspection tools

- CPU, memory, I/O performance
  - Databases perform better with multiple, fast disks and CPUs
  - Database cache ratio: should be high
- Watch for long-running SQL queries
  - SQL queries in application code can be formulated badly
  - Use database tools to analyze your statements (for example, **`db2 explain`**)
  - Index creation can help

OS tuning

- Tune the OS on each system as demanded by its usage
- Tuning OS for WebSphere and web server is different than for database servers

Figure 4-25. Bottleneck considerations (2 of 4) WB868 / ZB8681.0

Notes:

Tuning OS for WebSphere and web server is different from tuning database servers. For example, tune the AIX memory manager: A database system does not need large file system or buffer caches, which the AIX memory manager provides. Database software manages caches better itself.

What is the DB2 `explain` command? When an SQL query is executed against a DB2 database, the DB2 Optimizer tool defines the path that is used to access the data. The access path is defined based on table statistics that the DB2 runstats tool generates.

The `explain` command provides details of the access path and allows you to analyze how the data is accessed. Knowing these details, you might be able to improve the command's performance.

WebSphere Education

Bottleneck considerations (3 of 4)

Storage considerations:

- Extensive logging on each node and component can kill performance
- Set appropriate (low) log level for load testing and production machines
- Watch out for high disk I/O on web servers and application servers
- Use SAN storage if necessary

Caching:

- Can be done in various places
  – Understand when and which type of caching to use
- Do you serve static content from the application servers?
  – Perhaps a caching proxy server can increase speed without application changes

Figure 4-26. Bottleneck considerations (3 of 4) WB868 / ZB8681.0

Notes:

Storage considerations include:

- Extensive logging on each node and component can impair performance.
- Set appropriate (low) log levels for load testing and production.
- Watch out for high disk I/O on web servers and application servers.
- Using SAN storage is becoming increasingly popular, but it also becomes a black box (an abstraction of a device or system in which only its externally visible behavior is considered and not its implementation). You need a SAN storage expert to diagnose problems.

A storage area network (SAN) is a high-speed subnetwork of shared storage devices. A storage device is a dedicated machine that contains one or more disks for storing data. A SAN's architecture works such that all storage devices are available to all servers on a LAN or WAN.

WebSphere Education

Bottleneck considerations (4 of 4)

- Bugs in application code are easy to make but hard to find
  - Root cause analysis is best done together with developers
  - “You cannot tune your way out of a bad application”
- Use a monitoring tool that shows historical data
  - Compare system performance for different application releases
  - Look for changes in resource usage
  - Use OS monitoring tools as well as web server, application server, and database server monitoring tools
- Are you still having performance problems?
  - Use a profiler to look into the application code
  - Perform execution-time analysis to find out which parts of application take too long to execute
  - Profiling requires Java development and application knowledge

Figure 4-27. Bottleneck considerations (4 of 4) WB868 / ZB8681.0

Notes:

Bugs in application code are easy to make but hard to find. Root cause analysis is best done together with developers. But remember that you cannot tune your way out of a bad application.

Use a monitoring tool that shows historical data. Compare system performance at different application releases during the production phase to spot changes in resource usage. Use OS monitoring tools along with web, WebSphere, or DB monitoring tools.

If you are still having performance problems, use a profiler to look into the application code. Do execution-time analysis to find out which parts of your application execute too long, and dig down deeper. Correct analysis requires a high skill level in Java development and application knowledge.

4.4. System queues

System queues

Figure 4-28. System queues WB868 / ZB8681.0

Notes:

WebSphere Education

System queues: The queuing network

- WebSphere Application Server establishes a queuing network
- Goal: You must accept a high number of client connections, but the system usually performs better if not all clients are processed simultaneously
- The queuing network represents queues of requests that are waiting to use a resource
- The queues include:
  - Network
  - Web server
  - Web container
  - EJB container
  - Data source connection pools

Figure 4-29. System queues: The queuing network WB868 / ZB8681.0

Notes:

WebSphere Application Server establishes a queuing network. An application needs to accept a high number of client connections, but usually the system works better if not all clients are processed simultaneously.

The queuing network represents queues of requests that are waiting to use a resource. The queuing network includes the network, web server, web container, EJB container, and data source connection pools.

WebSphere Education

WebSphere tuning: Upstream queuing

- Upstream queuing attempts to allow more work to be done by limiting the number of connections at each tier of the application

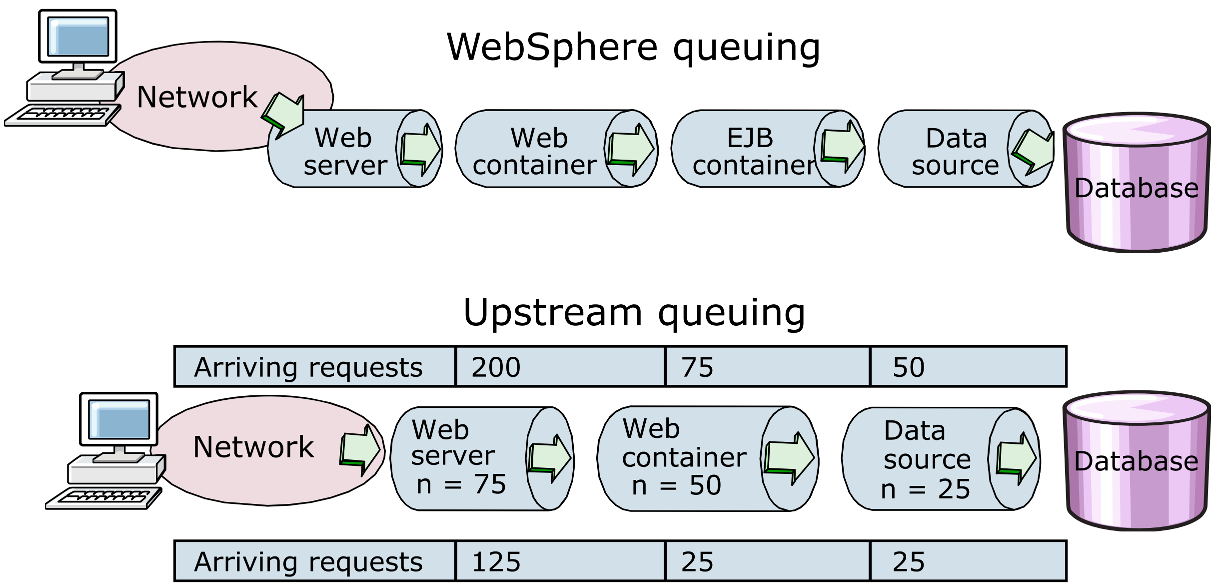


Figure 4-30. WebSphere tuning: Upstream queuing WB868 / ZB8681.0

Notes:

Design the queuing system to allow as much work on the database and application server as possible without over-stressing each area. It is important not to overload any one tier because then you have poor performance due to waiting threads. This result is one of the most difficult things to achieve during performance testing.

WebSphere Education

How to configure system queues (1 of 2)

- Most users should wait in the network
- Specify the queues farthest upstream (closest to the client) to be slightly larger, and specify the queues farther downstream (farthest from the client) to be progressively smaller

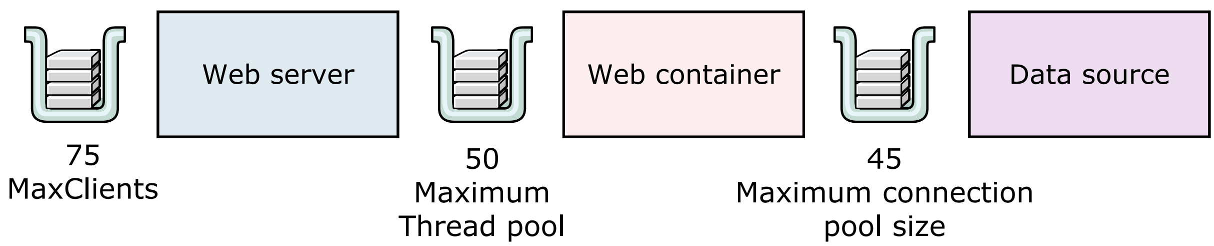


Figure 4-31. How to configure system queues (1 of 2) WB868 / ZB8681.0

Notes:

Most users should wait in the network; therefore, specify the queues farthest upstream (closest to the client) to be slightly larger, and specify the queues farther downstream (farthest from the client) to be progressively smaller. For example, given a Maximum Application Concurrency value of 48, start with system queues at the following values: web server 75, web container 50, data source 45. Do a set of extra experiments by adjusting these values slightly higher and lower to find the best settings.

WebSphere Education

How to configure system queues (2 of 2)

- IBM HTTP Server
  - In recent versions, the queue mechanism of the web server is greatly improved in its scalability, and tuning can focus primarily on the web server itself
  - **`MaxClients`** for UNIX and Linux is the maximum number of simultaneous client connections
  - Use **`ThreadsPerChild`** for Windows systems
- Web container
  - Thread pool maximum size
  - HTTP transport channel: maximum persistent requests
- ORB (object request broker)
  - Thread pool maximum size
- Data source
  - Maximum connection pool size
  - Prepared statement cache size

Figure 4-32. How to configure system queues (2 of 2) WB868 / ZB8681.0

Notes:

IBM HTTP Server: In V6, the queue mechanism of the web server was greatly improved in its scalability, and tuning can focus primarily on the web server itself. Configure the MaxClients parameter for UNIX and the ThreadsPerChild for Windows systems.

Web container: Configure thread pool maximum size, HTTP transport channel, and maximum persistent requests parameters.

ORB (object request broker): Configure thread pool maximum size.

Data source: Configure connection pool size and prepared statement cache size.

WebSphere Education IBM

Determining optimum queue sizes example (1 of 2)

Procedure:

1. Perform a number of load tests against the application server environment with large queue settings to ensure maximum concurrency through the system

2. Set the queue sizes for the web server, web container, and data source to an initial value; for example, 100

3. Simulate many typical user interactions that concurrent users enter in an attempt to fully load the WebSphere environment

   In this context, “concurrent users” means simultaneously active users who:

   - Send a request
   - Wait for the response
   - Immediately resend a new request

Figure 4-33. Determining optimum queue sizes example (1 of 2) WB868 / ZB8681.0

Notes:

The procedure should be to do a number of load tests against the application server environment with large queue settings to ensure maximum concurrency through the system.

1. Set the queue sizes for the web server, web container, and data source to some initial value, for example 100.

2. Simulate many typical user interactions that concurrent users enter in an attempt to fully load the WebSphere environment.

   In this context, “concurrent users” means simultaneously active users that send a request, wait for the response, and immediately resend a new request upon response reception, without think time.

WebSphere Education IBM

Determining optimum queue sizes example (2 of 2)

- Use any load testing tool to simulate this workload
- Measure overall throughput
  - Determine at what point the system capabilities are fully stressed (the saturation point)
- Repeat the process, increasing the user load each time
- After each run, record
  - Throughput (requests per second)
  - Response times (seconds per request)
  - Plot the throughput curve

Figure 4-34. Determining optimum queue sizes example (2 of 2) WB868 / ZB8681.0

Notes:

1. Use any stress tool to simulate this workload.
2. Measure overall throughput and determine at what point the system capabilities are fully stressed (the saturation point).
3. Repeat the process, and each time increase the user load.
4. After each run, record the throughput (requests per second) and response times (seconds per request) and plot the throughput curve.

WebSphere Education IBM

Unit summary

Having completed this unit, you should be able to:

- Identify the areas of the runtime environment that need to be tuned
- Evaluate a list of the top ten monitoring considerations
- Detect performance bottlenecks
- Determine optimum queue sizes

Figure 4-35. Unit summary WB868 / ZB8681.0

Notes:

WebSphere Education

Checkpoint questions

1. True or False: Performance bottlenecks at the OS level include Java heap size and database connection pool size.

2. True or False: The throughput of WebSphere Application Server is a function of the number of concurrent requests present in the total system.

3. True or False: Upstream queuing attempts to allow more work to be done by using equal queue and thread pool sizes along the request flow.

Figure 4-36. Checkpoint questions WB868 / ZB8681.0

Notes:

Write your answers here:

1.

2.

3.

WebSphere Education IBM

Checkpoint answers

1. False: Performance bottlenecks at the OS level include paging, swapping, and disk I/O

2. True

3. False: Upstream queuing attempts to allow more work to be done by limiting the number of connections at each tier of the application.

Figure 4-37. Checkpoint answers WB868 / ZB8681.0

Notes:

Unit 5. IBM Business Process Manager performance concepts

What this unit is about

This unit provides an introduction to the main areas of Business Process Manager that can be monitored and tuned for optimal performance.

What you should be able to do

After completing this unit, you should be able to:

- Define architecture best practices
- Identify top Business Process Manager tuning guidelines
- Explain the Business Process Manager performance tuning methodology
- Define common tuning parameters
- Describe Process Center tuning
- Describe how to tune BPEL business processes
- Explain how to tune Process Portal
- Explain how to tune the Process Center environment
- Explain how to use the Event Manager for monitoring
- Describe how to tune the Event Manager
- Describe how to tune participant groups

How you will check your progress

- Checkpoint

References

IBM Business Process Manager V8.5 IBM Knowledge Center

`http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/ditamaps/ic-homepage-bpm.html`

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Define architecture best practices
- Identify top Business Process Manager tuning guidelines
- Explain the Business Process Manager performance tuning methodology
- Define common tuning parameters
- Describe Process Center tuning
- Describe how to tune BPEL business processes
- Explain how to tune Process Portal
- Explain how to tune the Process Center environment
- Explain how to use the Event Manager for monitoring
- Describe how to tune the Event Manager
- Describe how to tune participant groups

Figure 5-1. Unit objectives WB868 / ZB8681.0

Notes:

WebSphere Education

Topics

- Best practices for BPM solutions
- Architecture best practices
- Optimizing BPM clients
- Tuning the default BPM settings for optimal performance
- Event Manager monitoring and tuning

Figure 5-2. Topics WB868 / ZB8681.0

Notes:

5.1. Best practices for BPM solutions

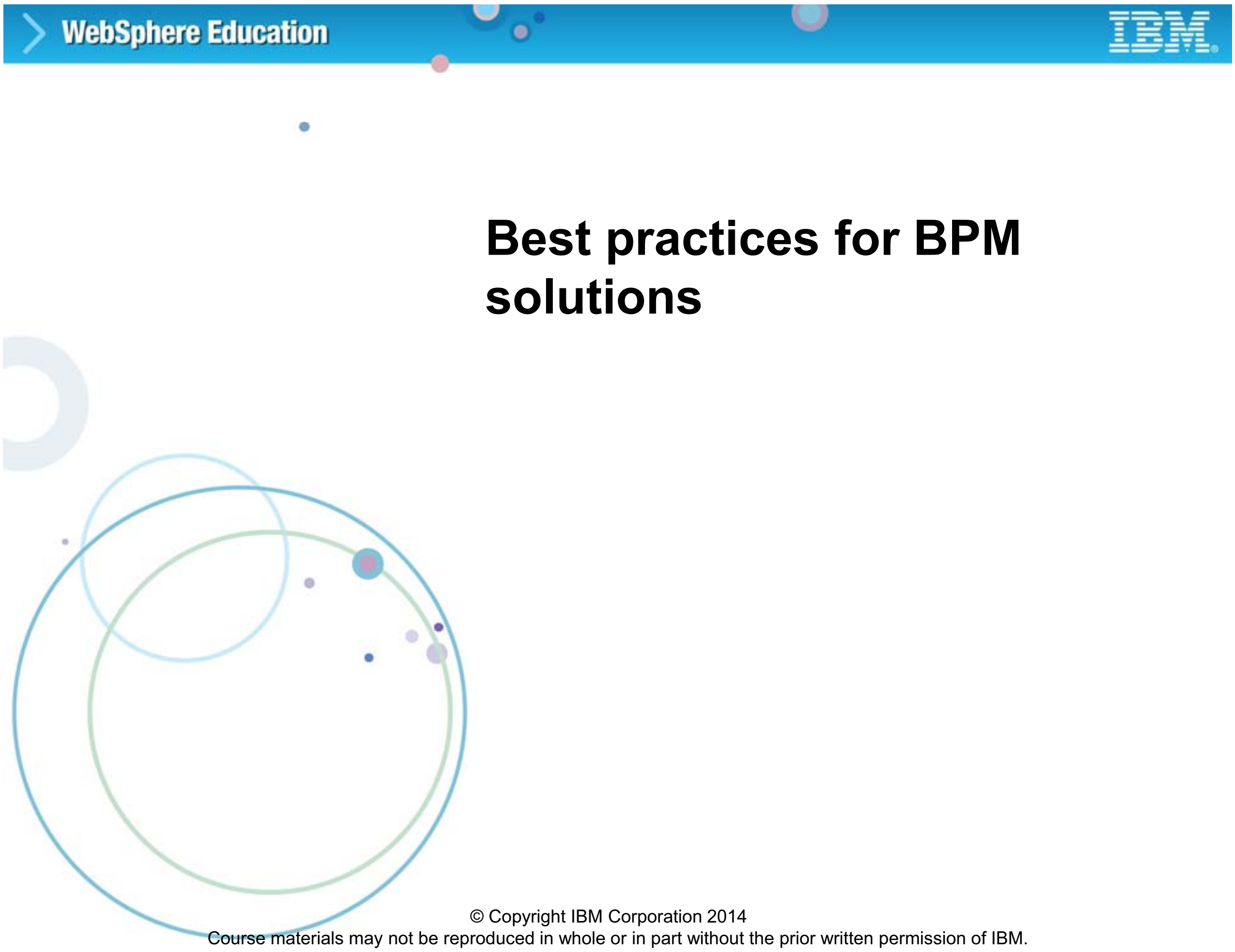


Figure 5-3. Best practices for BPM solutions WB868 / ZB8681.0

Notes:

WebSphere Education

BPM methodologies

- IT monitoring and tuning go together
  - Monitor resource utilization to determine the tuning that is required
- Examine every system in your environment, including physical resources that are backing virtual resources
- Use the following methodology steps:
  - Define performance objectives and how to assess them, such as response time SLAs, CPU utilization, and others
  - Choose a set of reasonable initial parameter settings and run the system
  - Monitor the system to obtain metrics to find out where performance is limited
  - Use monitoring data to guide further tuning or topology changes
  - Repeat until done

Figure 5-4. BPM methodologies WB868 / ZB8681.0

Notes:

Examine AppTarget clusters, databases, messaging engines, and the operating system.

WebSphere Education

Define performance objectives

- Establish clear performance objectives that are representative of business requirements
  - For example, refreshing the Task List in Process Portal should take less than 3 seconds when using IE9 with 100 ms of network latency between Process Portal and Process Server
- Design workloads that are reliable, produce repeatable results, and are easy and fast to run
- Ensure that the environment is warmed up before obtaining measurements
- For load tests:
  - Ensure that realistic think times are used between requests
  - Maintain a consistent flow of work into the system
  - Maintain steady state in the databases
  - Clean up at the end of the run, particularly queues, and the databases
- Check for errors, exceptions, and timeouts in logs and reports

Figure 5-5. Define performance objectives WB868 / ZB8681.0

Notes:

WebSphere Education

Top deployment guidelines

- Use a high-performance disk subsystem on tiers that host BPM data stores to achieve acceptable performance
  - The appropriate disk subsystem can be used to improve performance by several factors
- Use the most current release of the product with the most current fix pack
- Use a fast, high-bandwidth network connection in the environment
- Disable tracing, logging, and monitoring when possible
- If security is required, use application security, not Java2 security
- Use 64-bit JVMs for all servers
- Do not run production servers in development mode
- Deploy local modules in the same server
- Turn off state observers that are not needed

Figure 5-6. Top deployment guidelines WB868 / ZB8681.0

Notes:

If you plan to deploy modules on the same physical server, you can achieve better performance by deploying the modules to the same application server JVM. The server can then take advantage of this locality.

WebSphere Education

Top tuning guidelines

- Set an appropriate Java heap size for optimal throughput and response time
- Configure thread pools to enable sufficient concurrency
  - Important thread pools include the Default, web container, ORB, and Event Manager
- Configure the connection pool size for a data source and prepared statement cache size
- Increase the maximum number of connections in the data pool to greater than or equal to the sum of all maximum thread pool sizes
- Tune your database for optimal performance
- Tune external service providers and interfaces to ensure that they do not cause a system bottleneck
- Tune the Event Manager
- Tune for BPMN and BPEL processes

Figure 5-7. Top tuning guidelines WB868 / ZB8681.0

Notes:

One typical usage pattern for Business Process Manager is as an integration layer between incoming requests and back-end systems for the business (target applications or service providers). In these scenarios, the layer with the lowest throughput capacity limits the throughput.

Consider the simple case where there is only one target application. The Business Process Manager integration solution cannot achieve throughput rates higher than the throughput capacity of the target application. This inability to increase throughput applies regardless of the efficiency of the Business Process Manager implementation or the size or speed of the system that is hosting it. Thus, it is critical to understand the throughput capacity of all target applications and service providers and apply this information when designing the end-to-end solution.

Two key aspects of the throughput capacity of a target application or service provider are as follows:

- Response time, both for typical cases and exceptional cases
- Number of requests that the target application can process at the same time (concurrency)

WebSphere Education

BPMN tuning

- Tune the Process Server database
  - Increase the log file size
  - Enable file system caching
  - Exclude the table SIBOWNER from automatic runstats execution
  - Ensure that the database statistics are up-to-date
- Tune the Event Manager
- Tune participant groups
  - If you are not using the external email capability for participant groups, turn it off
  - This step reduces load on the Process Server and the database by eliminating unnecessary email lookups
  - Change the `send-external-email` parameter to false

Figure 5-8. BPMN tuning WB868 / ZB8681.0

Notes:

For participant groups with many users, if you are not using the external email capability (which is enabled by default), turn it off through the following change to `99Local.xml` in the profiles directory:

`<send-external-email>false</send-external-email>` (default is true)

This step reduces load on the Process Server and its database by eliminating unnecessary email lookups.

WebSphere Education IBM

BPMN tuning: Optimize business data search options

- Optimize business data search options if slow response times, or high database usage, are observed when performing business data searches
  - Ensure that no more than 10 business data variables are defined
  - Enable process search optimizations through the saved search acceleration tools for the Process Server
- Process search optimizations
  - This technique is often much faster, and uses fewer database resources, than the default mechanism
  - Optimize a process search only on runtime or production systems
  - Do not optimize process searches on systems that are used for development, or for systems where changes happen frequently to deployed applications

Figure 5-9. BPMN tuning: optimize business data search options WB868 / ZB8681.0

Notes:

You use the following two search acceleration tools to optimize a process search:

- SchemaGenerator tool: This tool generates two new tables: LSW_BPD_INSTANCE_VAR_NAMES (variables table), and LSW_BPD_INSTANCE_VARS_PIVOT (pivot table). It also generates the schema for each table.
- DataLoad tool: This tool populates the variables table data from the BPD instances that are currently in progress.

The pivot and variables tables

The pivot and variables tables ensure that your query runs efficiently as soon as your process search is optimized. The tables present data in the following format, with each item in its own column:

- The pivot table:
  - Instance ID
  - Every searchable business data variable that is defined in all currently deployed BPDs

  - Every business data variable that is defined in LSW_BPD_INSTANCE_VARIABLES
- The variables table:
  - Variable name
  - Column name
  - Data type

Process search optimization overview

A process search optimization can consist of three unique processes:

- Enabling an optimization
- Disabling an optimization
- Re-enabling an optimization

Enabling optimization

To enable process search optimization, you need to run the optimization tools.

Disabling optimization

To disable optimization, do the following steps:

1. Shut down the server.
2. Remove the pivot and variables tables from your database.
3. Restart the server.

Re-enabling optimization

To re-enable optimization, do the following steps:

1. Deploy all BPD updates.
2. Stop the server.
3. Remove the pivot and variables tables from your database.
4. Rerun the SchemaGenerator and DataLoad tools.
5. Restart the server.

WebSphere Education IBM

BPEL tuning

- When possible, use microflows instead of long-running processes
- Use composite query tables for task and process list queries
- Use Work Manager-based navigation to improve throughput for long running processes
  – Optimize the message pool size and inter-transaction cache size
- Avoid using asynchronous invocations unnecessarily
- Avoid overly granular transaction boundaries in SCA and BPEL
- Tune the database

Figure 5-10. BPEL tuning WB868 / ZB8681.0

Notes:

BPEL tuning is covered in more detail in Unit 9: Business Process Choreography best practices.

WebSphere Education IBM

Advanced content deployment considerations (1 of 2)

- Advanced content includes process applications or toolkits that are authored in Integration Designer
  - Takes much longer to deploy than standard content
  - Content is packaged as SCA modules and libraries
  - Deployed as BLAs and EARs to the runtime
- Deployments should be done the minimum number of times that is practical
  - BLAs and EARs deployed consume resources on the server such as memory, disk space, and CPU cycles
- Toolkits are copied by value into dependent process applications and toolkits
  - Content is duplicated on the server for every process application and toolkit that references it

Figure 5-11. Advanced content deployment considerations (1 of 2) WB868 / ZB8681.0

Notes:

A process application has a business-level application (BLA), which acts as a container for the process application and its assets (including SCA modules, toolkits, and libraries). Each process application snapshot has its own BLA. Many of the administration tasks for a snapshot (for example, stopping or starting it on a production server) are done at the level of the BLA. This task allows for quicker and simpler administration of the snapshot and all of its assets.

WebSphere Education

Advanced content deployment considerations (2 of 2)

- For Process Server deployments
  - Install snapshots during periods of low activity
  - Installation directories for the deployment manager and cluster members should be stored on a fast disk subsystem
- For Process Center server deployments
  - Make sure that the deployment manager and cluster members are installed on servers with multiple CPU cores
  - Use a dynamic load balancer that can detect CPU consumption on cluster members
- Purge data that is no longer needed
  - Data includes snapshots, process applications, and toolkit tips
  - Make sure to deactivate and undeploy advanced content

Figure 5-12. Advanced content deployment considerations (2 of 2) WB868 / ZB8681.0

Notes:

Purging data is covered in more detail in Unit 6: Purging data in IBM Business Process Manager.

WebSphere Education

Snapshot deployment governance

- A best practice is to create and use Snapshot Install Governance Process
  - Allows you to gain visibility and control over what gets installed on Process Servers
  - Ensures that proper business approvals are completed
  - Ensures that proper testing is done and all IT organization approvals are secured

- Snapshot Install Governance Process
  - Can prevent snapshot installation
  - Applies to both online and offline installations
  - Can query Requirement Management systems about the features included in a snapshot and then report the status
  - Can notify interested parties of a successful or aborted installation

Figure 5-13. Snapshot deployment governance WB868 / ZB8681.0

Notes:

When you create a snapshot or change status, an instance of the default Snapshot Status Change BPD from the System Governance toolkit is started. When a new snapshot of a process application or toolkit is created or when the status of a snapshot changes, a message event is triggered. If you created a governance process and bound to the snapshot event for the process application, you can use the governance process to track status changes and issue notifications.

5.2. Architecture best practices

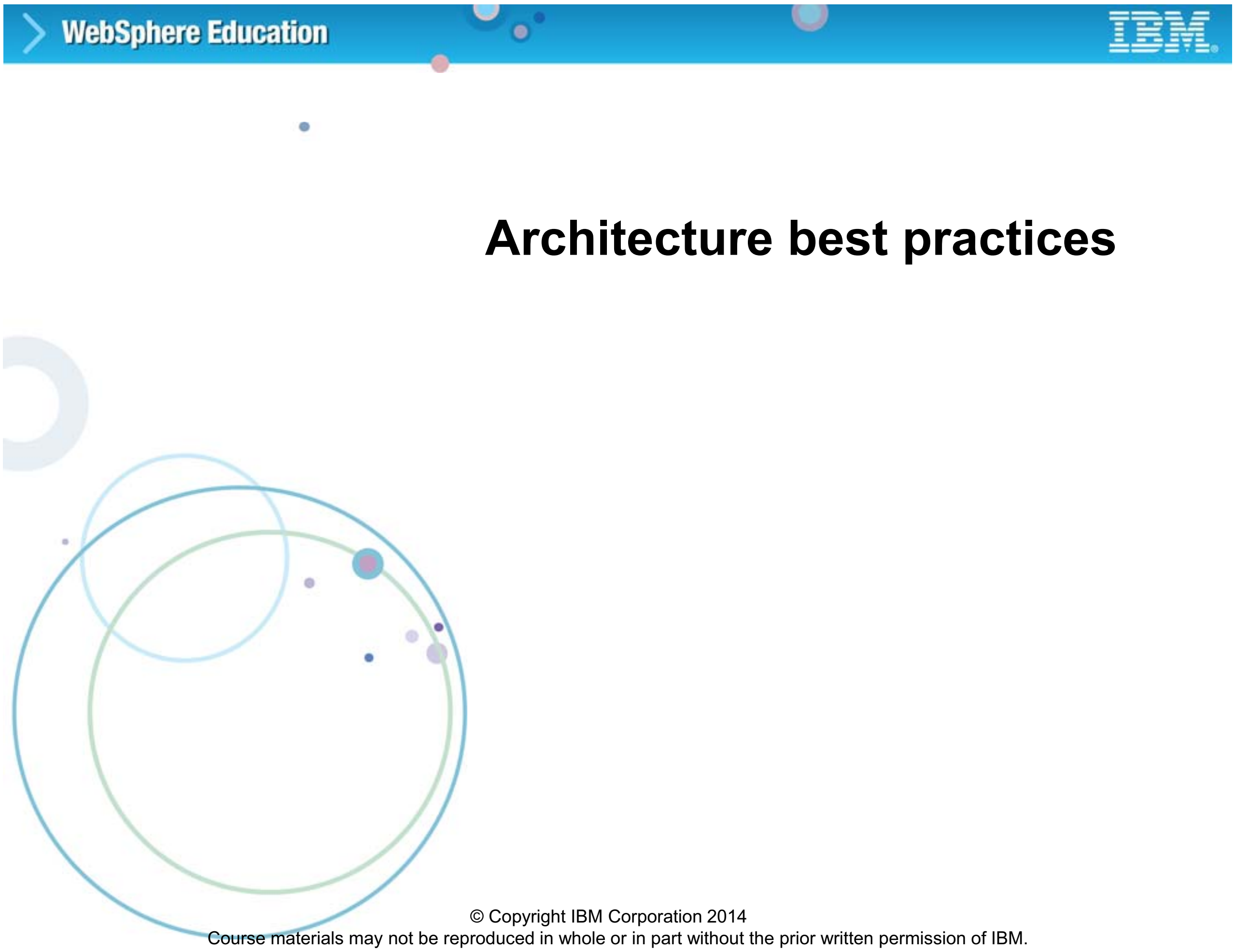


Figure 5-14. Architecture best practices WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Hardware (1 of 2)

Pick a hardware configuration that contains the resources necessary to achieve high performance

- Processor cores
  - Verify that the servers are installed on a modern server system with multiple processor cores
- Memory
  - Make sure that there is enough physical memory for all of the applications that are expected to run concurrently on the system
  - For 64-bit JVMs, 4 GB per JVM is needed if the max heap size is 3 GB or less
  - Add more physical memory for heap sizes larger than 3 GB
- Disk
  - Verify that systems have fast storage by using Redundant Array of Independent Disks (RAID) adapters with writeback caches

Figure 5-15. Hardware (1 of 2) WB868 / ZB8681.0

Notes:

For each physical machine in the topology, including front-end and back-end servers such as web and database servers, monitor the following processes:

- Processor core use
- Memory use
- Disk use
- Network use

Use the relevant OS tools such as vmstat, iostat, netstat, or their operating system-specific equivalents.

WebSphere Education

Hardware (2 of 2)

- Network
  - Verify that the network is fast enough not to create a system bottleneck
- Minimizing network latency and ensuring sufficient bandwidth is essential
  - Between Process Designer and Process Center
  - Between Process Center and the database
  - Between Process Center and online Process Servers
  - Between Process Portal and Process Server
  - Between Process Server and the database
- Virtualization
  - Ensure that there are sufficient physical processor, memory, and I/O recourses allocated to each virtual machine or logical partition
  - Avoid overcommitting resources

Figure 5-16. Hardware (2 of 2) WB868 / ZB8681.0

Notes:

WebSphere Education

Topology best practices

- Use the Application, Remote Messaging, and Remote Support deployment pattern for maximum flexibility in scaling
  - Allows for the use of separate clusters for applications and messaging engines
  - Provides independent control of resources to support the load on each of the components in the environment
- BPM servers scale well, both horizontally and vertically
  - Ensure that your performance test hardware matches production hardware
  - Start performance testing small and move up incrementally
  - Ensure that more hardware gives expected extra capacity
- Become familiar with recommended clustering topologies
  - Clustering for performance can give different approaches than high availability

Figure 5-17. Topology best practices WB868 / ZB8681.0

Notes:

The Application, Remote Messaging, and Remote Support topology pattern is an IBM-supplied topology pattern. In this pattern, the deployment environment functions are divided among three separate clusters.

The Application, Remote Messaging, and Remote Support topology pattern is the preferred topology for IBM Business Process Manager Standard and IBM Business Process Manager Advanced. It is also the default for the `BPMconfig` command, except for z/OS.

WebSphere Education

Clustered topology tuning (1 of 2)

- There are three primary reasons to consider when evaluating moving to a clustered topology from a single-server configuration
  – Scalability, high availability, and the ability to expand the environment
- If you are deploying more than one cluster member on a single physical system, keep in mind the following information:
  – It is important to monitor not just the resource utilization (CPU, disk, network) of the system as a whole, but also the utilization by each cluster member
  – This monitoring allows the detection of a system bottleneck due to a particular cluster member
- Use a minimum of two cluster members and a minimum of two processor cores per cluster member
  – Delivers high availability and ensures that a processor-intensive task that consumes significant CPU is less likely to affect other response times of users
  – Guideline applies to both Process Server and Process Center

Figure 5-18. Clustered topology tuning (1 of 2) WB868 / ZB8681.0

Notes:

In general, there are three primary issues to consider when evaluating whether to use a clustered topology instead of a single-server configuration:

- Scalability and load balancing to improve overall performance and throughput
- The ability to expand a configuration over time by adding more cluster members as more applications are deployed, or more users are added to the system
- High availability by using failover to another cluster member to prevent loss of service because of hardware or software failures

Although not mutually exclusive, each requires certain considerations.

WebSphere Education IBM

Clustered topology tuning (2 of 2)

- Effective scaling requires standard performance monitoring and bottleneck analysis techniques to be used
  - Examine if a singleton server or cluster member is the bottleneck
- For a messaging engine
  - Host the active cluster member on a more powerful hardware server
  - Removing extraneous load from the existing server
- For a database server
  - Consider hosting each database on a separate server if your database hosts multiple servers and if a single database is driving load
  - If a single database is driving load, consider a more powerful database server
  - Use database partitioning and clustering

Figure 5-19. Clustered topology tuning (2 of 2) WB868 / ZB8681.0

Notes:

As always, contact IBM TechLine or your IBM Sales Representative to define the specific configuration for your requirements.

WebSphere Education

Good practices for Process Center

- Avoid the use of a single shared Process Center
  - If you have a geographically dispersed BPM development team, it is better to have regional Process Centers
  - Regional is better than a single shared Process Center that remote Process Designer authoring clients access
- Process Designer interacts frequently with the Process Center for authoring tasks
  - For this reason, minimize network latency to provide optimal response times
  - Place the Process Center in or near the same physical location as the Process Designer users
  - Process Designer clients that connect to a highly remote Process Center might experience slow performance and dropped connections.

Figure 5-20. Good practices for Process Center WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Deployment options for Integration Designer

- Use Integration Designer with Process Center
  - Associate modules and libraries with process applications and toolkits in the Process Center repository
  - When the modules and libraries are published to the repository, EAR files are automatically generated and deployed for testing in the Process Center server
  - You can create a snapshot of the process application and use the Process Center Console to deploy the snapshot to a remote Process Server
  - You can still deploy and test BPEL processes or business integration services in the local Integration Designer test environment
- Use Integration Designer without Process Center
  - If you have older processes that were developed for WebSphere Process Server, you can still continue to develop and deploy applications to Process Server
  - Do not associate your modules with process applications or toolkits
  - Use a `serviceDeploy` command and an Ant script to deploy an EAR file to Process Server
  - You can also export EAR files from Integration Designer to either of the runtime environments

Figure 5-21. Deployment options for Integration Designer WB868 / ZB8681.0

Notes:

The idea is to deploy from Integration Designer to Process Center and from Process Center to your Process Server.

To successfully publish or deploy from Integration Designer to a Process Center server, the version of both tools must be the same.

5.3. Optimizing BPM clients

Optimizing BPM clients

Figure 5-22. Optimizing BPM clients WB868 / ZB8681.0

Notes:

WebSphere Education

Optimize BPM client environments (1 of 2)

- Important for optimizing Business Process Manager client environments
  - Includes Process Portal, Process Designer, and Business Space
- Performance of Asynchronous JavaScript and XML (Ajax) applications can be divided into a four-tier model
  - Each tier must be optimized to deliver a high-performing solution
- Ajax applications perform work on the client side, inside the browser
  - Such as building the user interface (UI)
  - Loads the content of a page dynamically as needed
- The four-tier model includes
  - Browser, network, servers, and databases

Figure 5-23. Optimize BPM client environments (1 of 2) WB868 / ZB8681.0

Notes:

The performance of Asynchronous JavaScript and XML (Ajax) applications, such as those used in Business Process Manager, can be divided into a four-tiered model. Each tier must be optimized to deliver a high-performing solution.

WebSphere Education

Optimize BPM client environments (2 of 2)

- Browser
  - Key to delivering excellent performance
  - Make sure to use a high-performing browser
  - Use modern desktop hardware
- Network
  - Delays in the network can significantly affect response times as it adds time to each message
  - Can add up to the most significant factor for the overall page-response time
  - Plays a role in communication between servers and the databases and between servers in a cluster
- Servers
  - Key servers include BPM servers and an HTTP server, if used in the topology
- Databases
  - Each database should be properly configured and tuned to support the anticipated load

Figure 5-24. Optimize BPM client environments (2 of 2) WB868 / ZB8681.0

Notes:

Ajax applications, by definition, do work (to some extent) on the client side, inside the browser. All typical client work, such as building the user interface (UI), is done on the client side, which differentiates these applications from classical web applications, where only the rendering of HTML is done in the browser. Optimizing the browser and client system is key to delivering excellent performance.

WebSphere Education

Tune Process Portal

- Use a high-performing browser and enable browser caching
- Use modern desktop hardware
  - Deploy modern desktop hardware with sufficient physical memory and high-speed processors with large caches and fast front-side buses
- Locate Process Portal physically close to the Process Server
- Configure the Process Portal index
- Use the WORK tab to refresh the Task List
  - Instead of refreshing the entire Process Portal browser page
- Apply search filters for the Process Portal Dashboard
  - Limit the data that is displayed in the process instances and tasks dashboard to improve the response time of the dashboard
- Disable or uninstall add-ins or extensions

Figure 5-25. Tune Process Portal WB868 / ZB8681.0

Notes:

The choice of browser technology is crucial to Business Process Manager client performance. It is possible that more recent versions of a browser perform better than the older versions of the same browser.

One factor that influences network latency is the physical distance between servers and clients. Another factor is the distance between the Business Process Manager servers in a cluster and their database (for example, between the Process Server and its databases). When practical, locate Business Process Manager servers physically close to each other and to the Business Process Manager clients to minimize the latency for requests.

WebSphere Education

Process Portal search indexing considerations

- There is one search index per node by default
  - On each node, at any time, a single cluster member gets a lock on the index and performs indexing
  - All cluster members on the same node read from the index of the node
  - A cluster member from each node indexes all of the cells data
- When there are multiple nodes in the topology, consider changing the search index configuration
  - Configure a single shared index across nodes
  - Only one of the cluster members across the entire node gets a lock on the index and performs indexing
  - All cluster members across the cell read from the shared index
- Options that you can change
  - Where the index is stored
  - Indexing behavior

Figure 5-26. Process Portal search indexing considerations WB868 / ZB8681.0

Notes:

The index allows process participants in Process Portal to search for tasks or process instances that contain particular metadata or instance data. The index is also used for historic data in the Process Performance and Task Performance dashboards.

WebSphere Education

Configuring the index

- Modify where the index is stored
  - The value of the `BPM_SEARCH_TASK_INDEX` environment variable in the cluster name directory determines the location of the index
  - Use a shared network storage solution for your index and change the value of the cell scoped `BPM_SEARCH_TASK_INDEX_ROOT` variable to point to the common location
- Indexing behavior
  - To change the index behavior, edit the `100Custom.xml` file for the appropriate server

```
<search-index>
<task-index-enabled>true</task-index-enabled>
<task-index-update-interval>5</task-index-update-interval>
<task-index-update-completed-tasks>false</task-index-update-completed-tasks>
<task-index-store-fields>false</task-index-store-fields>
<task-index-work-manager>wm/default</task-index-work-manager>
<task-index-include-system-tasks>true</task-index-include-system-tasks>
<process-index-instance-completion-best-effort>false</process-index-instance-completion-best-effort>
</search-index>
```


Figure 5-27. Configuring the index WB868 / ZB8681.0

Notes:

- `<task-index-enabled>`: This Boolean value determines whether indexing is enabled. The default value is true; if the index does not exist, it is created. To turn off indexing, change the value to false; if the index does not exist, it is not created. If indexing is turned off, the search field in the Process Portal user interface is hidden.
- `<task-index-update-interval>`: This integer value specifies the time between index updates in seconds. The specified interval determines when the state of the instance variables is captured for tasks that completed since the last index update. Only those tasks that are completed during the current interval are searchable with the newest instance data. The default value for the update interval is 5 seconds. The minimum value is 1 second.
- `<task-index-update-completed-tasks>`: This Boolean value controls whether the index is updated for completed tasks. The default value is false, which means that only information about open tasks is updated. If you change the value to true, instance-level updates, such as business data that is updated later in the process, are propagated to completed tasks.
- `<task-index-store-fields>`: This Boolean value controls whether the actual field values are stored as separate fields. The default value is false, which means that the actual field values are not stored as separate fields. You might want to change the value to true for debugging

purposes because it improves the readability for people and it allows queries by other search tools.

- `<task-index-work-manager>`: This string contains the JNDI name of the work manager that the indexing process uses to manage the search index. The default value is wm/default, which is the default work manager for WebSphere Application Server. To improve the performance of the index creation, in the administrative console you can create a dedicated work manager with a greater number of available threads. You can then use this tag to switch to the new work manager.
- `<task-index-include-system-tasks>`: This Boolean value controls whether system tasks are indexed. To enable system tasks to be displayed in Gantt charts in Process Portal, ensure that the value of this tag is set to true. If the value of this tag is false, system tasks are not displayed in Gantt charts.
- `<process-index-instance-completion-best-effort>`: This Boolean value controls whether completion dates are created when instances that are migrated from previous versions of IBM BPM are indexed. The default setting is false. If you change the value to true, the last completion date of the associated tasks is used for the instance completion date. If no associated tasks exist, the last modified time stamp of the instance is used as the completion date.

5.4. Tuning the default BPM settings for optimal performance

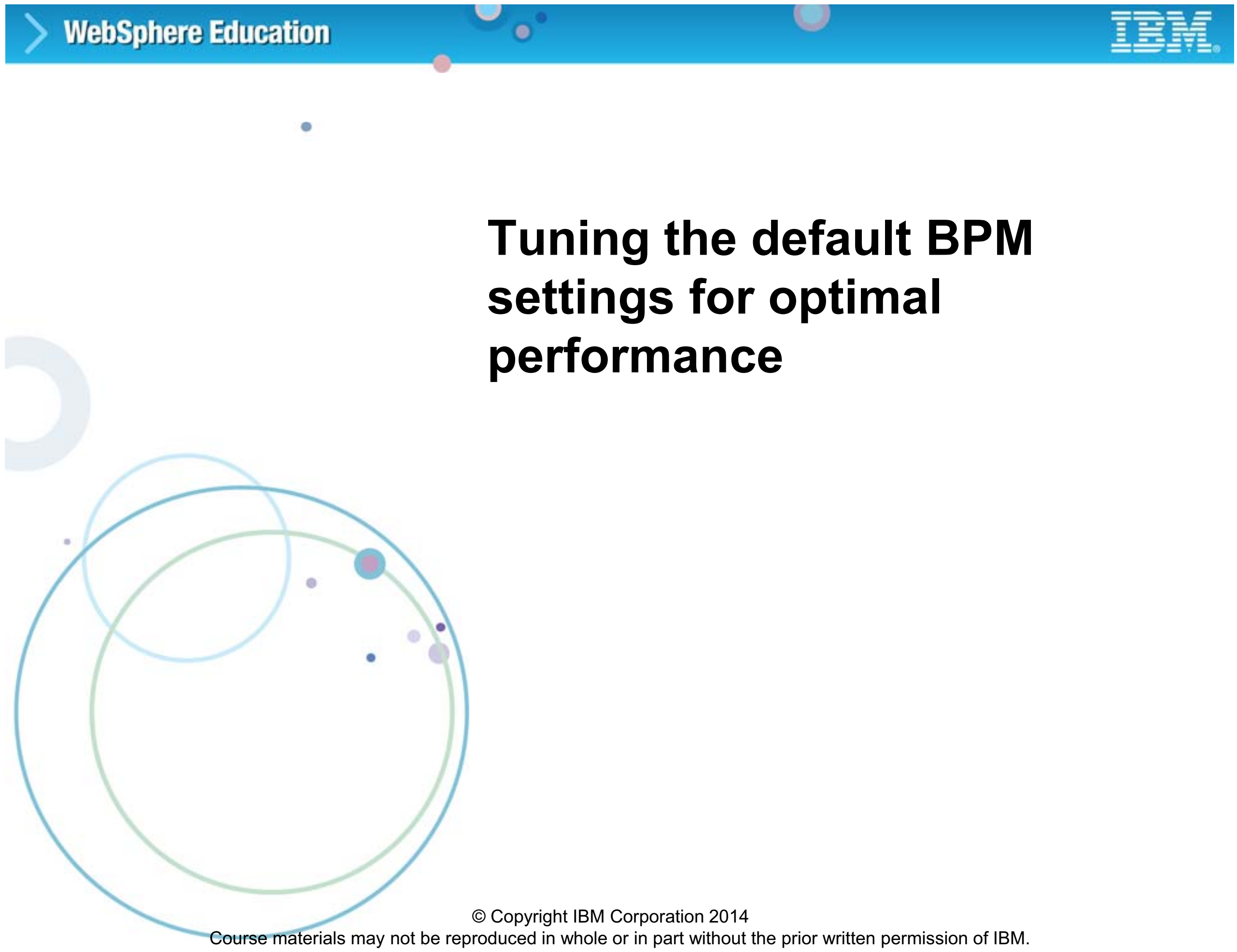


Figure 5-28. Tuning the default BPM settings for optimal performance WB868 / ZB8681.0

Notes:

WebSphere Education

Java tuning

- Set the heap and nursery size to manage memory efficiently
- Choose the appropriate garbage collection policy
  - The default garbage collection algorithm on platforms with an IBM JVM is a generational concurrent collector
  - Specified with `-Xgcpolicy:gencon` under the Generic JVM arguments on the JVM administrative console panel
  - This garbage collection policy usually delivers better performance with a tuned nursery size
- Enable verbosegc to obtain Java memory statistics for later analysis
  - There is essentially no administrative cost attributable to enabling verbosegc

Figure 5-29. Java tuning WB868 / ZB8681.0

Notes:

Java tuning is covered in more detail in Unit 13: Java memory issues.

WebSphere Education

Tuning thread pools

- Business Process Manager servers use thread pools to manage concurrent tasks
  - Set the **Maximum Size** property of a thread pool in the administrative console
- You typically must tune the following thread pools
  - Default
  - `ORB.thread.pool`
  - WebContainer
  - Event Manager (for BPMN solutions)
- You typically must tune the following Work Managers for BPEL solutions
  - DefaultWorkManager
  - BPENavigationWorkManager

Figure 5-30. Tuning thread pools WB868 / ZB8681.0

Notes:

Tuning thread pools is covered in more detail in Unit 8: Threading.

WebSphere Education

Tuning the Process Server Cache

- Tune the Time to Live setting by using the `cached-objects-ttl` parameter
  - Default value is 0 seconds
  - Appropriate for a Process Center development environment where you want changes to object to reflect updates immediately
  - Not appropriate for a Process Server runtime where changes are rare

- In a production environment, increase the value

- Change the number of objects in the cache
  - Use the `<default-unversioned-po-cache-size>` and `<default-versioned-po-cache-size>` parameters
  - For more complex environments with many process applications or coaches, increase this value so objects are held in the cache of their initial use
  - This increase can improve response time

Figure 5-31. Tuning the Process Server Cache WB868 / ZB8681.0

Notes:

Restarting the Process Server clears the caches. If code is deployed and it is required that the change is recognized immediately, the Process Server can be restarted. Or, the PO cache can be reset through the Process Admin Console under the Manage Caches link.

WebSphere Education

Monitoring the cache

- The effectiveness of these caches can be monitored through the Instrumentation Page in the Process Admin Console

IBM BPM Admin > Manage Caches

Name	Description	CA	UCA	UCP	Last A.	Status	Actions
E@GroupInfoCache	Stores UserGroup objects by GroupName and GroupId	4,526	66	0%	6:20 PM	ON	(Show) (Reset)
GroupCache	Caches group information and list of groups with information.	1	0	0%	10:27 AM	ON	(Show) (Reset)
E@GroupMembers	Stores group members (User IDs and Group IDs)	236	109	0%	11:57 AM	ON	(Show) (Reset)
ProfileCache	Caches user profile information	231	2	0%	1:42 PM	ON	(Show) (Reset)
Runtime TWClass Cache	Caches all business objects for use by the runtime engines	-	-	-	-	ON	(Reset)
PO Cache	Caches all library items	-	-	-	-	ON	(Reset)

Refresh View

Figure 5-32. Monitoring the cache WB868 / ZB8681.0

Notes:

The UCP column in the Manage Caches page of the Process Admin console can also be used to monitor how effectively the cache is working for a particular runtime server. The Process Admin console has online help that describes the column headings and their meanings.

User group membership information is also stored in a cache so a company with a large LDAP user and group population might require a larger setting for LDAP caches. This task is to avoid a performance impact when user or group information is accessed, such as during authorization validation for task assignment, login to Process Portal, or refreshing the Task List in Process Portal.

WebSphere Education IBM.

Overview of branch and snapshot cache sizes

- As process applications are developed, the changes that are made to versioned objects are tracked in the repository
  - Versioned objects consist of process models, service models, and coaches
  - These objects are noted as a three-part identifier: project, branch (or track), and snapshot
  - These objects are loaded into memory and metadata about the branch and snapshot is also loaded and cached
- The branch cache is used for each project-branch combination that is loaded into the memory cache
  - Each branch cache entry contains a snapshot cache
- The snapshot cache is used for metadata about the branch and the snapshot
- The branch and snapshot cache sizes can be tuned independently
  - For exceptionally large process applications, reducing these values can reduce the JVM heap memory usage

Figure 5-33. Overview of branch and snapshot cache sizes WB868 / ZB8681.0

Notes:

One entry in the branch cache is used for each project-branch combination that is loaded into the memory cache. For example, if a process application depends on two toolkits, three branch cache entries are required: one for the process application and one for each toolkit. Each entry in the branch cache contains a snapshot cache. Therefore, the number of entries in the branch cache is tunable, independently from the snapshot cache.

Branch cache entries are reused across project dependencies in the same way that the objects are reused across project dependencies. For example, if a process application has two snapshots that are deployed where both snapshots depend on the same two toolkit snapshots, the branch cache has four entries: one for each process application snapshot and one for each toolkit snapshot. The same is true if two different process applications each have one snapshot that is deployed, where both snapshots depend on the same two toolkits: four branch cache entries are created.

WebSphere Education

Tune the branch cache size (1 of 3)

- The branch cache is a Least Recently Used (LRU) cache
  - If the working set of branches is limited because the cache size is too small, more cache misses occur
  - Cache misses can cause a significant increase of database activity because artifacts must be loaded directly from the database
- Branch cache entries can vary widely in the amount of memory that they use
  - A snapshot cache is contained in a branch cache entry, and the size of a snapshot cache depends on the application
  - Can be difficult to precisely predict the amount of extra memory to be used when you are tuning the branch cache
- Use a tactical approach to monitoring cache activity in Process Server
  - Ensure that the SQL query response time on the `LSW_PO_VERSIONS` table does not suddenly degrade after you deploy a new process application snapshot

Figure 5-34. Tune the branch cache size (1 of 3) WB868 / ZB8681.0

Notes:

For example, a toolkit with one business object takes less than 1 MB of memory for the branch cache entry and the snapshot cache. The IBM BPM Portal Process application uses about 11 MB of memory. The system data toolkit uses several hundred MB of memory. There is a difference in memory usage because of the complexity of the relationships between the artifacts in the project and the volume of artifacts in each project. Therefore, it is difficult to precisely predict the amount of added memory that is used when you are tuning the branch cache.

WebSphere Education

Tune the branch cache size (2 of 3)

- Determine a rough calculation of the memory usage by using the following equation

  (# of unique process application snapshots + # of unique toolkit snapshots) x size_constant

  - The `size_constant` is a value in megabytes and requires careful consideration
  - When considering a change to the branch cache size, the effect on the JVM memory must be considered
- Example:
  - There are several process applications, where each application depends on different versions of about 10 large toolkits
  - If each process application also has a few snapshots on the system, then the branch cache could be nearing its default size of 64
  - Three process applications, times 2 snapshots each, times 10 toolkits snapshots is 60 branch cache entries; the snapshot cache within each entry is consuming a large amount of memory because the toolkits are large and complex
  - If a fourth process application is added to this system, not only the branch cache size needs to be increased, but there can be an increase in JVM heap usage

Figure 5-35. Tune the branch cache size (2 of 3) WB868 / ZB8681.0

Notes:

A process application depends on 70 toolkits. This process application requires 71 branch cache entries, a number that exceeds the default of 64. In this case, the branch cache size should be increased to avoid constant queries against the LSW_PO_VERSIONS table, which slows down the entire system. Note, however, that if this process application depends on 70 toolkits, then it is also likely that each toolkit does not contain many artifacts. The reason for having many toolkits is to create discrete packages of capability. Therefore, increasing the branch cache size does not have much impact on the memory requirements of the BPM JVMs.

WebSphere Education

Tune the branch cache size (3 of 3)

- You can configure the branch cache size in the `100Custom.xml` file

```
<server merge="mergeChildren">
    <repository merge="mergeChildren">
            <branch-context-max-cache-size merge="replace">64</branch-context-max-cache-size>
    </repository>
</server>
```

- The size of the branch cache is denoted as the number of branch cache entries that can be held in the cache

Figure 5-36. Tune the branch cache size (3 of 3) WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Tune the snapshot cache size (1 of 2)

- In a Process Server environment, each deployed snapshot is deployed to a unique branch
  - The snapshot cache (which is part of each branch cache entry) contains only one entry
- In Process Center, all the snapshots on a branch use the same branch cache entry
  - The number of entries in the snapshot cache might be more than one
  - The number of entries that are needed depends on the number of snapshots that are accessed
- Tuning the snapshot cache size is necessary only for the Process Center server, as based on anticipated usage patterns
- Snapshot cache entries can be large because they depend on application content
  - It is not uncommon for snapshot cache entries to reach several hundred MBs

Figure 5-37. Tune the snapshot cache size (1 of 2) WB868 / ZB8681.0

Notes:

In Process Center, all the snapshots on a branch use the same branch cache entry; therefore, the number of entries in the snapshot cache might be more than one. The number of entries that are needed depends on the number of snapshots that are accessed. Snapshots are accessed when the snapshot is activated or when Process Designer users view artifacts, play back artifacts, or do both at a specific snapshot version. Therefore, tuning the snapshot cache size is necessary only for the Process Center server. The necessity to tune the snapshot cache is based on anticipated usage patterns.

WebSphere Education IBM

Tune the snapshot cache size (2 of 2)

- You can configure the snapshot cache size in the `100Custom.xml` file

```
<server merge="mergeChildren">
   <repository merge="mergeChildren">
         <snapshot-cache-size-per-branch merge="replace">64</snapshot-cache-size-per-branch>
   </repository>
</server>
```

- The size of the snapshot cache is denoted as the number of snapshot cache entries that can be held in the cache
  - This value is directly related to the number of unique process application and toolkit snapshots that are being accessed

Figure 5-38. Tune the snapshot cache size (2 of 2) WB868 / ZB8681.0

Notes:

5.5. Event Manager monitoring and tuning

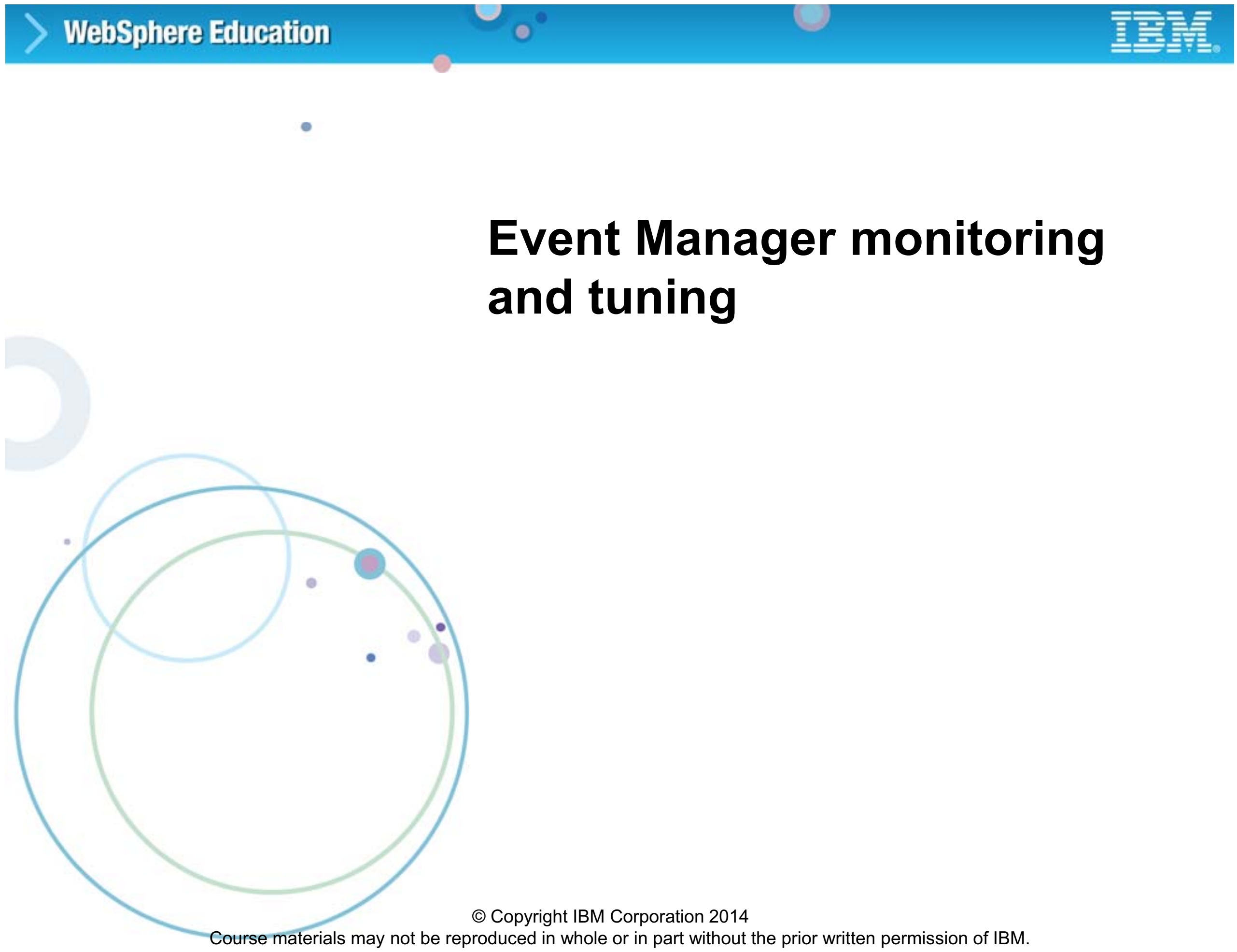


Figure 5-39. Event Manager monitoring and tuning WB868 / ZB8681.0

Notes:

WebSphere Education

Event Manager overview

- The Event Manager schedules execution of code
  - It does not execute the code, just schedules it with a corresponding Process Center or Process Server
  - For example, UCAs, BPD token movement, or system lane activities
  - Any work that is scheduled by a specific Event Manager runs on the local server
- The Event Manager
  - Uses queues to manage events and when asked to do work, the Event Manager reloads its queues from the database
  - Can be running or paused
  - Has a heartbeat, which is a separate thread that constantly updates a database table to tell other schedulers that it is alive
- The Event Manager queues include:
  - Asynchronous queues, which are executed as soon as possible with no guaranteed order
  - BPD asynchronous queues for BPD-related tasks such as system lane activities, BPD notifications, and timer events
  - Synchronous queues, which are executed serially

Figure 5-40. Event Manager overview WB868 / ZB8681.0

Notes:

All schedulers whose heartbeat ever ran are listed in `lsw_em_instance`, and are shown on the Event Manager Monitor page.

To prevent problems in a cluster, an Event Manager claims ownership of one or more sync queues when it starts.

WebSphere Education

Tokens

- Think of a token as a marker as to where in the BPD or service you are
  - Processes and services use tokens to mark where the current running instance is taking place
  - Each BPD token progression is a BPD task in the Event Manager
- Tokens can be passed from BPD to an activity
  - Process Inspector shows you the movement of the tokens between activities
- When tokens encounter an endpoint, the token returns to the parent or calling item or ends the instance
- Stranded tokens are a problem because you can do nothing with them
  - If you see stranded tokens, often you must terminate the instance that you see the stranded tokens in

Figure 5-41. Tokens WB868 / ZB8681.0

Notes:

WebSphere Education

Event Manager Monitor (1 of 2)

- The Event Manager Monitor displays information about the Scheduler for the Event Manager on your Process Center server or Process Server
  - Accessed by using the Process Admin Console
- When you access the Event Manager monitor
  - You can see the status for each scheduled job
  - The monitor displays all Schedulers in the cluster and the jobs for all Schedulers in the cluster
- The Event Manager monitor does not show historical information about undercover agents (UCAs) that are successfully run
  - You can capture this type of information in the `SystemOut.log` file
  - Set the log details level for the `WLE.wle_ucaexception` component to `*=info`

Figure 5-42. Event Manager Monitor (1 of 2) WB868 / ZB8681.0

Notes:

The Event Manager Monitor in the Process Admin Console displays information about the Scheduler for the Event Manager on your Process Center server or Process Server and about the various jobs that the Scheduler tracks.

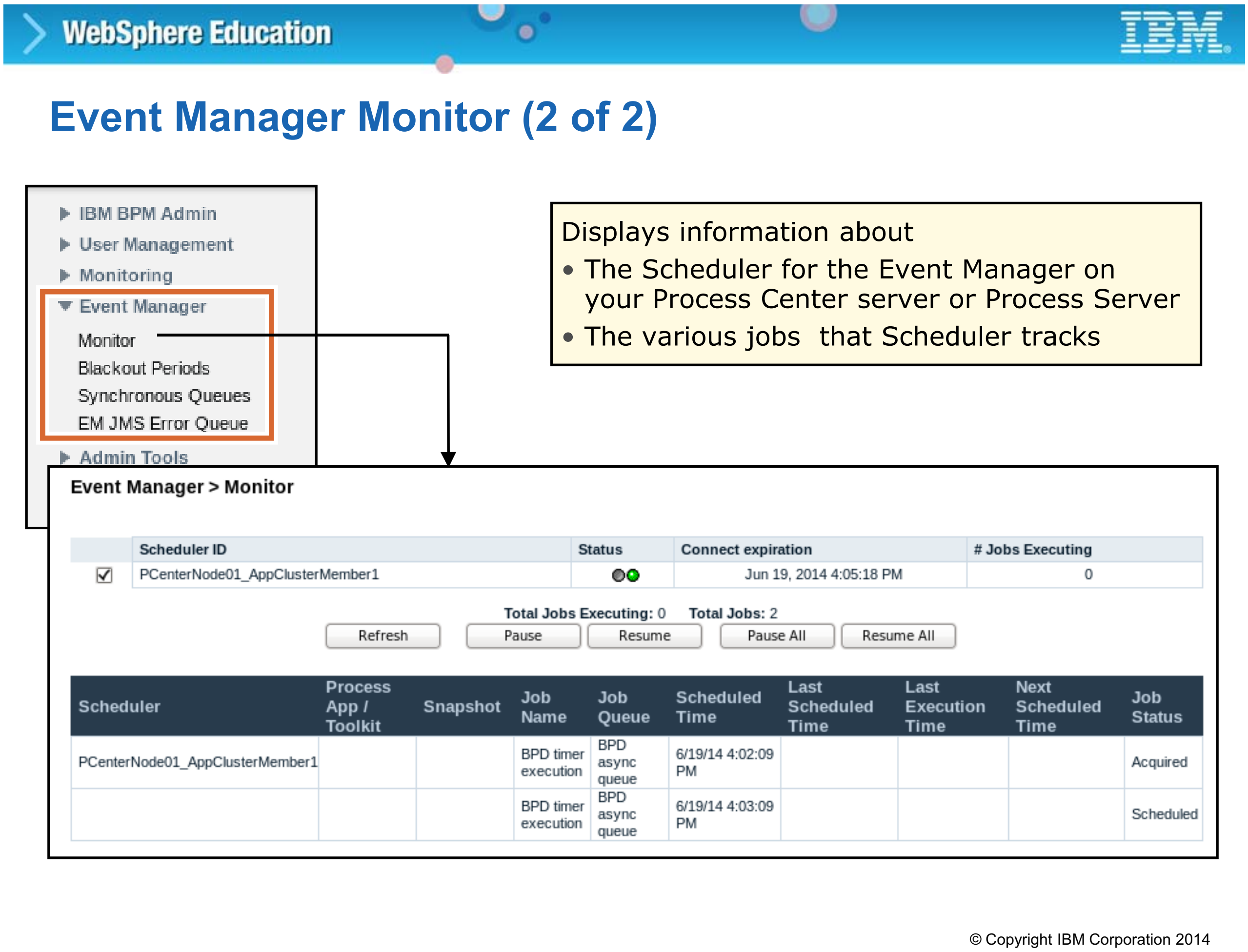


Figure 5-43. Event Manager Monitor (2 of 2) WB868 / ZB8681.0

Notes:

The Event Manager Monitor, included in the Process Admin Console, is useful for troubleshooting processes that are supposed to run automatically (through an undercover agent, for example) but fail to start. You can use the Event Manager monitor to identify underlying problems and also to control various aspects of Event Manager processing.

Administrators should establish blackout periods to specify times when events cannot be scheduled, for example, due to a holiday or when regular system maintenance is scheduled. The Event Manager takes blackout periods into account when scheduling and queuing events, event subscriptions, and undercover agents (UCAs).

Event Manager jobs are scheduled for processing by an execution queue. (If you look at the job listing in the Event Manager Monitor, you can see that each job is assigned to a job queue.)

WebSphere Education IBM

Tuning the Event Manager (1 of 3)

- You can tune the Event Manager by changing the parameter values in the configuration file
  - The default settings for the scheduler should be fine for most deployments
  - Configuration details are in the `80EventManager.xml` file but all changes should be made in the `100Custom.xml` file
- Tune the `bpd-queue-capacity` and `max-thread-pool-size` parameters to achieve optimal throughput and scaling
- The `bpd-queue-capacity` parameter specifies the number of Event Manager tasks that are loaded into memory for the BPD queue
  - To optimize throughput and scaling, start with a `bpd-queue-capacity` of 10 per physical processor core
  - For example, 40 for a 4-processor core configuration, with a maximum value of 80

Figure 5-44. Tuning the Event Manager (1 of 3) WB868 / ZB8681.0

Notes:

Throttling the Event Manager is done by decreasing the queue capacity. It is not done by decreasing the thread pool with the `max-thread-pool-size` parameter. The parameter specifies the number of threads in the pool only, not the actual number of threads that the Event Manager can use.

WebSphere Education

Tuning the Event Manager (2 of 3)

- The `max-thread-pool-size` parameter specifies the maximum number of threads for the Event Manager engine thread pool
  - To optimize throughput and scaling, start with a max-thread-pool-size size of 30 plus 10 per physical processor core
  - For example, 70 for a 4-processor core configuration, with a maximum value of 110
- The `use-was-work-manager` parameter specifies whether the WebSphere Application Server Work Manager is designated for the thread pool
  - Default value is true; changing this value to false is not recommended
  - Setting the value to false would disable visibility in WebSphere resource monitoring tools

Figure 5-45. Tuning the Event Manager (2 of 3) WB868 / ZB8681.0

Notes:

The total available IBM Business Process Manager database connections in the application server connection pool should be at least twice the number of the maximum Event Manager worker threads. The number of connections on the actual database server needs to be at least the sum of the maximum thread number for all nodes in the cluster.

If `<use-was-work-manager>` is true, the number of threads is configured in the work manager thread pool configuration. If `<use-was-work-manager>` is false, the maximum number of threads is the sum of the BPD queue capacity plus the asynchronous queue capacity plus the system queue capacity plus number of synchronous queues.

WebSphere Education IBM

Tuning the Event Manager (3 of 3)

- The `loader-advance-window` parameter indicates the number of timer events that are held in memory
  - For BPDs with many timers, reduce the amount of Event Manager activity by lowering this parameter
  - Default value is 6000
- The `loader-long-period` parameter indicates the milliseconds between major loads of the task loader
  - For every task loader long period, the Event Manager looks at each synchronous and asynchronous queue that it has access to and fills them to capacity
  - This setting should be less than or equal to the setting of the `loader-advance-window`
  - Default value is 15000

Figure 5-46. Tuning the Event Manager (3 of 3) WB868 / ZB8681.0

Notes:

The task loader is a dedicated thread of an event manager instance that loads event manager tasks into in-memory queues. The default parameter settings of the task loader are optimized for most usage patterns. Typically there is no need to tune any of these parameters.

WebSphere Education

Tune the Event Manager thread pool size

- The Event Manager thread pool is typically configured to use a Work Manager
  - Go to **Resources > Asynchronous beans > Work managers >** ***bpm-em-workmanager*** **> Thread pool properties**

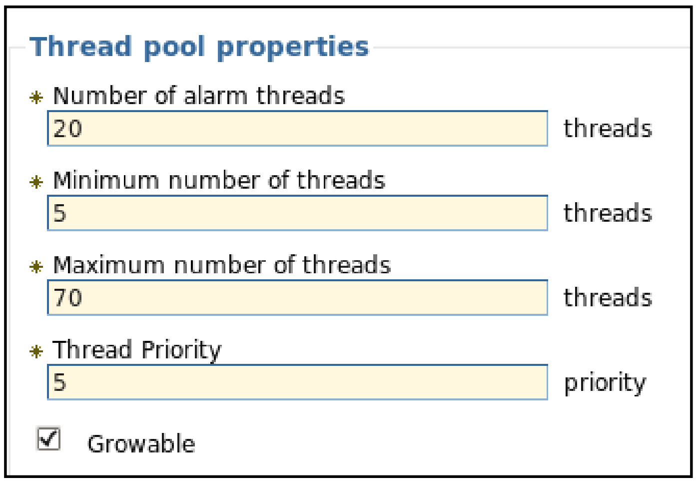


Figure 5-47. Tune the Event Manager thread pool size WB868 / ZB8681.0

Notes:

Number of alarm threads: Specifies the maximum number of threads that can be used for alarms.

Minimum number of threads: Specifies the minimum number of threads available in this work manager.

Maximum number of threads: Specifies the maximum number of threads available in this work manager.

Thread priority: Specifies the priority of the threads available in this work manager.

Growable: Specifies whether the number of threads in this work manager can be increased.

WebSphere Education IBM

Tips for the Event Manager

- The Event Manager is quick and efficient
  - Usually it is the tasks that it is executing that slow it down, not the Event Manager itself
- If you want to throttle the Event Manager, do not decrease the `max-thread-pool-size`, instead decrease the `bpd-queue-capacity`
- A synchronous queue can get stuck since it does not advance until the task completes
  - To help alleviate this problem, create multiple synchronous queues, which you can manage by using the Process Admin Console
- All the timestamps that the Event Manager scheduler uses are interpreted relative to the system clock of the database
  - Timestamps include the heartbeat expirations and the task's scheduled times
  - Keeping system clocks in sync is always a good idea

Figure 5-48. Tips for the Event Manager WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Unit summary

Having completed this unit, you should be able to:

- Define architecture best practices
- Identify top Business Process Manager tuning guidelines
- Explain the Business Process Manager performance tuning methodology
- Define common tuning parameters
- Describe Process Center tuning
- Describe how to tune BPEL business processes
- Explain how to tune Process Portal
- Explain how to tune the Process Center environment
- Explain how to use the Event Manager for monitoring
- Describe how to tune the Event Manager
- Describe how to tune participant groups

Figure 5-49. Unit summary WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Checkpoint questions

1. True or false: Tuning the snapshot cache size is necessary only for the Process Center server, as based on anticipated usage patterns.

2. True or false: You can configure Process Portal search indexing so only one of the cluster members across the entire node gets a lock on the index and does indexing.

3. True or false: The Event Manager schedules and executes code on a corresponding Process Center or Process Server.

Figure 5-50. Checkpoint questions WB868 / ZB8681.0

Notes:

Write your answers here:

1.

2.

3.

WebSphere Education

Checkpoint answers

1. True

2. True

3. False. The Event Manager schedules execution of code. It does not execute the code.

Figure 5-51. Checkpoint answers WB868 / ZB8681.0

Notes:

Unit 6. Purging data in IBM Business Process Manager

What this unit is about

This unit examines the areas of IBM Business Process Manager where data is collected and the methods available to purge that data to help with performance.

What you should be able to do

After completing this unit, you should be able to:

- Explain the need for purging data in the environment
- Archive, restore, and delete process applications and toolkits
- Archive and delete snapshots in the Process Center environment
- Delete unnamed snapshots
- Automate snapshot deletion
- Delete business-level applications
- Delete snapshots and instances in the Process Server environment
- Configure the cleanup service
- Define various business process choreography cleanup operations
- Explain more methods for purging data in the environment

How you will check your progress

- Checkpoint
- Lab exercise

References

IBM Business Process Manager V8.5 IBM Knowledge Center

`http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/ditamaps/ic-homepage-bpm.html`

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Explain the need for purging data in the environment
- Archive, restore, and delete process applications and toolkits
- Archive and delete snapshots in the Process Center environment
- Delete unnamed snapshots
- Automate snapshot deletion
- Delete business-level applications
- Delete snapshots and instances in the Process Server environment
- Configure the cleanup service
- Define various business process choreography cleanup operations
- Explain more methods for purging data in the environment

Figure 6-1. Unit objectives WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Topics

- Overview of purging concepts
- Purging data in the Process Center environment
- Purging data in the Process Server environment
- Business process choreography cleanup
- More areas to purge data

Figure 6-2. Topics WB868 / ZB8681.0

Notes:

6.1. Overview of purging concepts

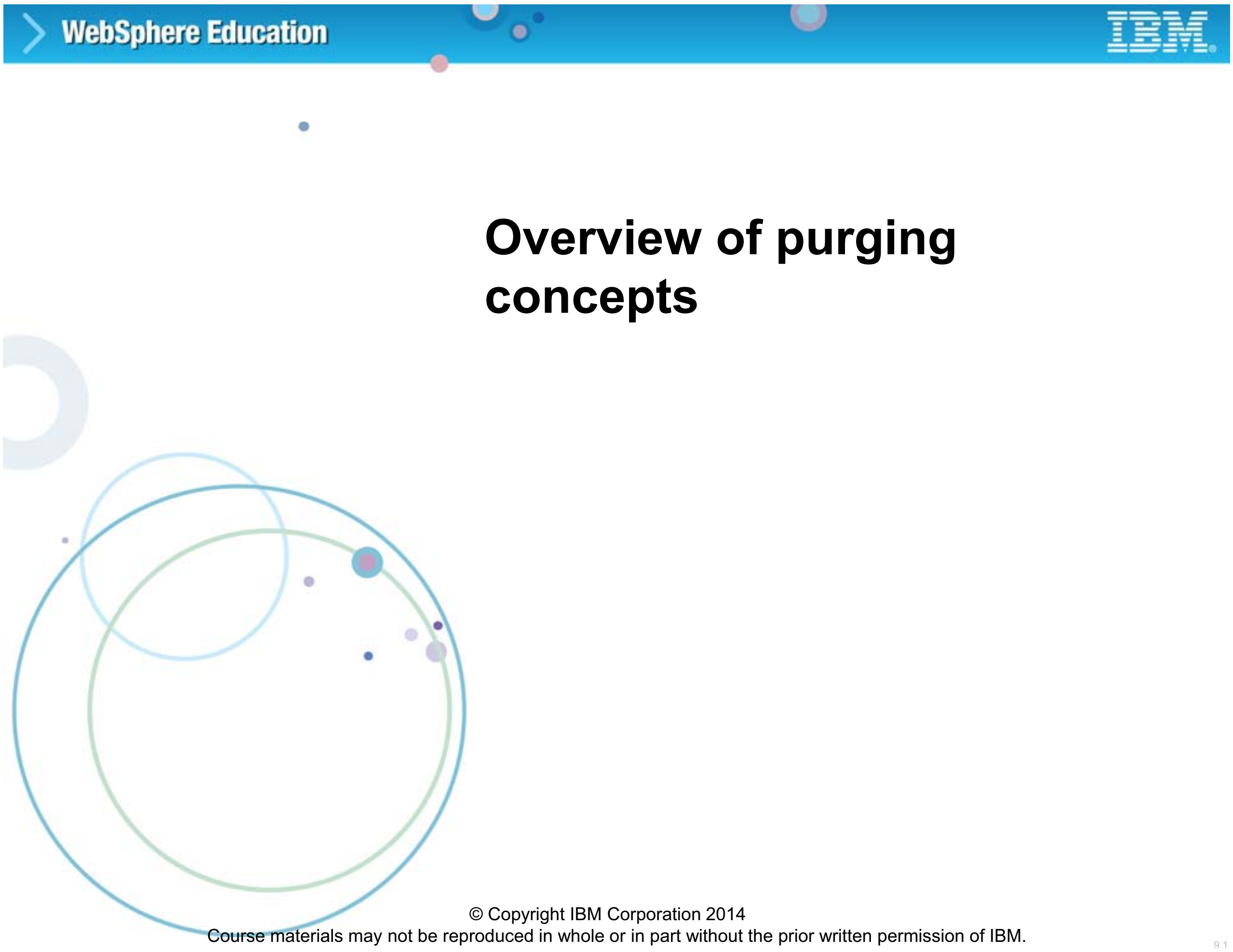


Figure 6-3. Overview of purging concepts WB868 / ZB8681.0

Notes:

WebSphere Education

Why do you need to purge data?

- Business Process Manager is a stateful product that accumulates data over time
  - The Process Server and Common databases can become large
  - Snapshots are regularly saved and deployed in the environment
  - Long-running processes store state information
- As data grows, it can lead to various problems
  - Database queries take longer to complete
  - Disk space issue
  - Performance-related issues
- You must design a strategy for the following tasks:
  - Deleting unneeded snapshots in Process Center
  - Deleting installed snapshots on Process Server
  - Deleting processes once they complete or terminate

Figure 6-4. Why do you need to purge data? WB868 / ZB8681.0

Notes:

When an instance completes and all of its associated tasks are closed, future work is not possible with this instance. You cannot restart the instance and assign it to someone or edit old work. When a user logs in to Process Portal, various database tables are queried to gather data on the active tasks for that user. This operation involves full table scans. Even if only 35% of the data is relevant, it is going to take a while to pull the tasks needed for that user. If the other 65% of the data is deleted, there is less data to scan.

If you do not delete the old completed instances, your team experiences slow performance on Process Portal and a potentially unusable state. Ignoring increases in database size causes an increase in backup time and disk space. Deleting old instances affects only the search for history items from the Process Portal inbox.

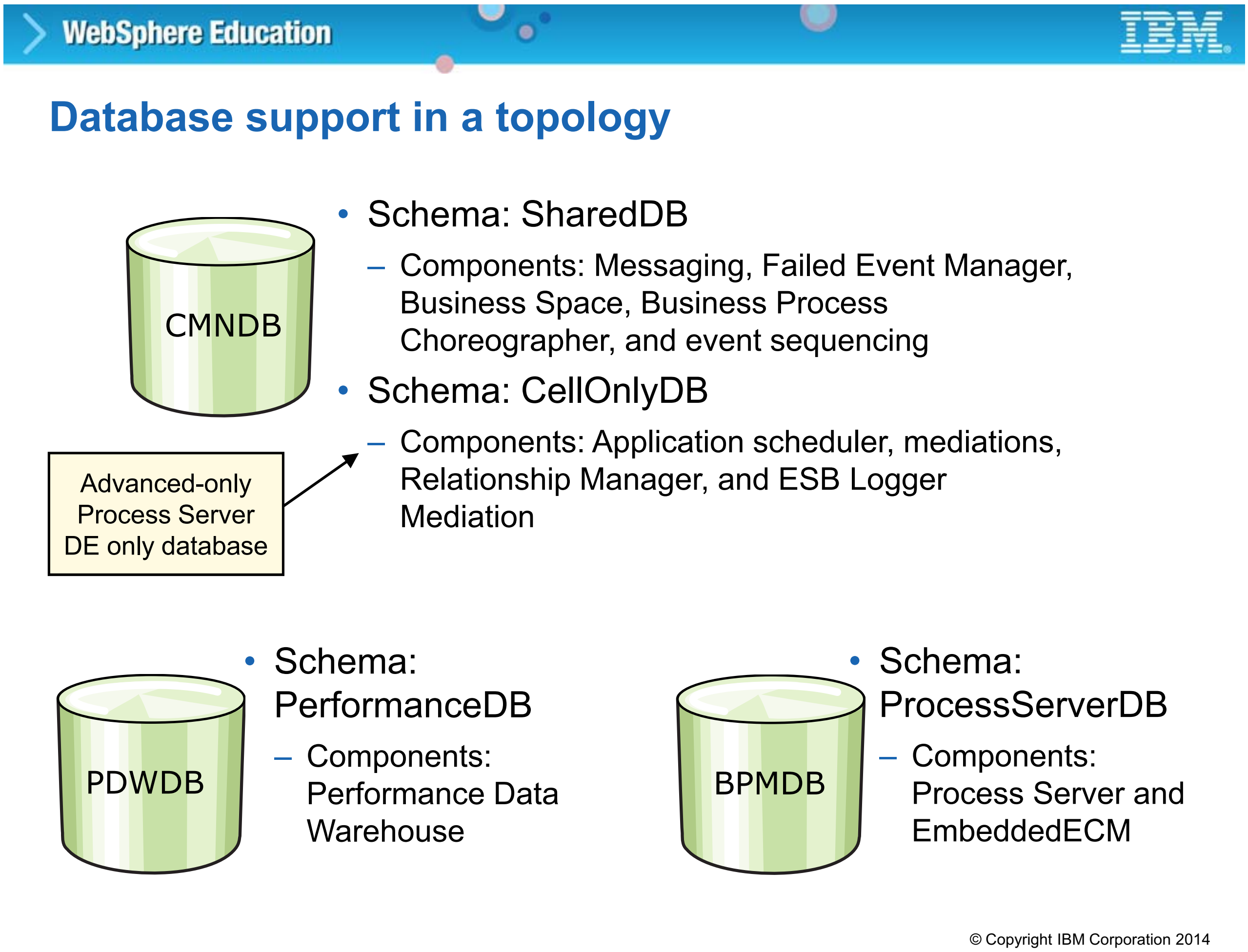


Figure 6-5. Database support in a topology WB868 / ZB8681.0

Notes:

The Process Server and Performance Data Warehouse components do not support case-sensitive databases. These databases must *not* be case-sensitive.

For Microsoft SQL Server and Oracle databases, the following restrictions apply:

- For Microsoft SQL Server databases, components other than Process Server or Performance Data Warehouse require that their databases be case-sensitive.
- For Oracle databases, the Process Server, Performance Data Warehouse, and Common database components must use a separate schema or user. They can use the same instance.

WebSphere Education

Perform regular housekeeping

- A best practice is to have a regular schedule of housekeeping for your Business Process Manager environment
  - Purge or archive data at regular intervals
- Especially important if you plan to do a migration
  - The number and size of objects to be migrated has a large impact on the complexity and overall duration of the migration
- Regularly clean up
  - Completed process instances and tasks
  - Snapshots that are no longer required
  - Performance Data Warehouse related data that is no longer required
  - Durable subscription messages that are already consumed
  - Failed events that might accumulate over time
  - Temporary directories
  - Log files, which include system, trace, and transaction logs

Figure 6-6. Perform regular housekeeping WB868 / ZB8681.0

Notes:

6.2. Purging data in the Process Center environment

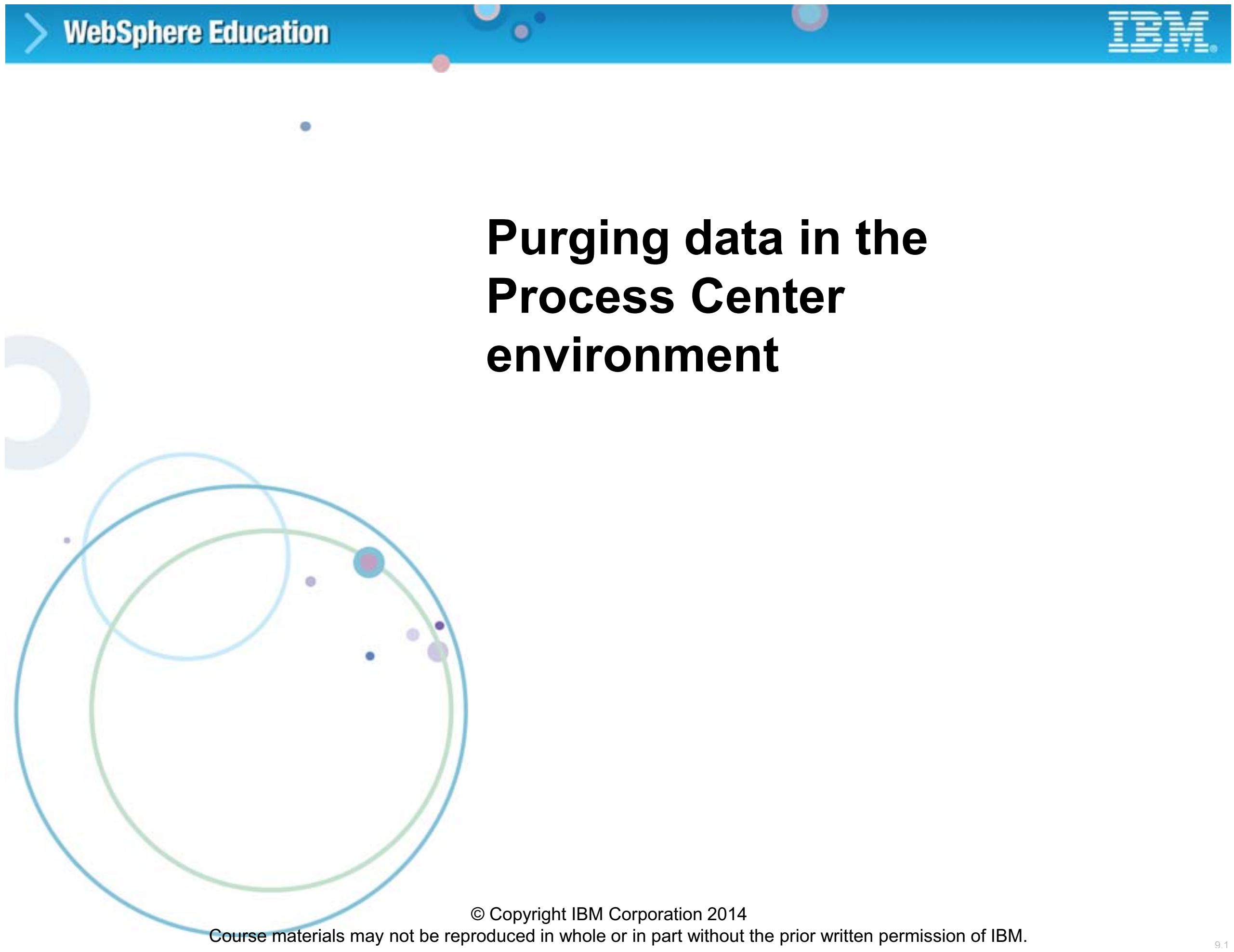


Figure 6-7. Purging data in the Process Center environment WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Process applications and toolkits (1 of 2)

- Archive a process application or toolkit
  - You can also archive snapshots
  - Does not delete the process application or toolkit, just removes it from the main view
  - Data still exists in the database
  - Restore the archived process application or toolkit later if needed
- Delete a process application or toolkit
  - Deletes the data from the repository, which includes snapshots and instances
  - You must archive first before deletion
  - No restore action available
- Archiving, restoring, and deleting can be done via the Process Center Console only

Figure 6-8. Process applications and toolkits (1 of 2) WB868 / ZB8681.0

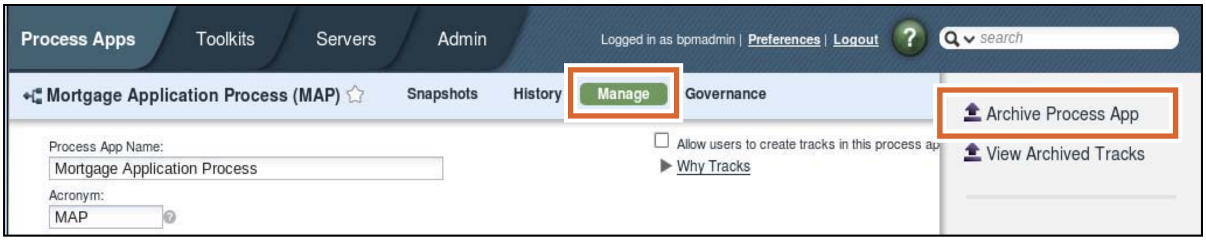
Notes:

Only the business process definitions of the process application are archived.

WebSphere Education

Process applications and toolkits (2 of 2)

- Archive a process application or toolkit



- Delete a process application or toolkit

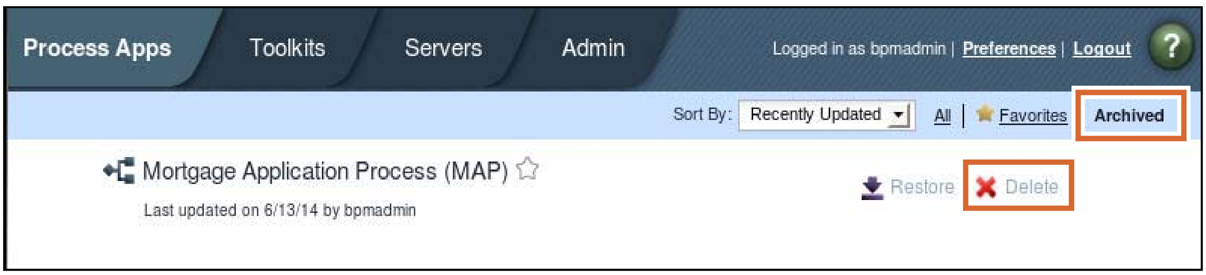


Figure 6-9. Process applications and toolkits (2 of 2) WB868 / ZB8681.0

Notes:

If a process application is no longer used, you can archive it. When you archive a process application, it no longer appears in the list of all process applications in the Process Center Console, and you must restore it before you can open it in the Designer view.

WebSphere Education

Snapshots

- Process Center holds snapshots of process applications and toolkits as they are developed
  - Includes named and unnamed snapshots
  - Named snapshots of a process application are deployed to a server runtime
  - Unnamed snapshots are created each time a business author saves in Process Designer
- Snapshots can be archived or deleted
  - Archive snapshots by using the Process Center Console
  - Delete snapshots by using either a wsadmin command or an automated method
  - Named snapshots must be archived first before deletion
- To delete snapshots, use the `BPMSnapshotCleanup` command
  - Deletes both named and unnamed snapshots

Figure 6-10. Snapshots WB868 / ZB8681.0

Notes:

Snapshots record the state of library items within a process application or track at a specific point in time. You can create snapshots in the Process Center Console or in the Designer view. Snapshot management, such as installing, exporting, and archiving, is done in the Process Center Console.

Use the `BPMSnapshotCleanup` command to delete unused snapshots of a process application when they are obsolete or older than a specified snapshot.

WebSphere Education

The BPMSnapshotCleanup command (1 of 2)

- You must archive named snapshots before you delete them
- You cannot delete the first snapshot of a process application or toolkit
  - The first snapshot contains original information about the snapshot that is displayed in the history panel in Process Designer
- Run the command when there are no operations on Process Center or connections to Process Designer
- Deleting snapshots in batch provides better performance
- For full capabilities, you must install fix pack 1 and the required interim fixes

Figure 6-11. The BPMSnapshotCleanup command (1 of 2) WB868 / ZB8681.0

Notes:

For details on the required interim fixes, go to the following URL:
`http://www.ibm.com/support/docview.wss?uid=swg21669992`

WebSphere Education

The BPMSnapshotCleanup command (2 of 2)

- The `BPMSnapshotCleanup` command parameters
  - `containerAcronym`: Where are the snapshots to be deleted
  - `[containerTrackAcronym]`: Where are the snapshots to be deleted
    `[containerSnapshotAcronyms]`: Where are the snapshots to be deleted
    OR
  - `[keptNumber]`: Number of unnamed snapshots to keep
    OR
  - `[createdBeforeLocal]`: Delete unnamed snapshots that are created before this date
  - `[createdAfterLocal]`: Delete unnamed snapshots that are created after this date
    `[createdBeforeSnapshotAcronym]`: Delete unnamed snapshots that are created before this name
  - `[deleteArchivedSnapshot]`: Delete archived named snapshots also; default is false
  - `[ignoreDependency]`: Delete even if other projects depend on this snapshot; default is false
  - `[outputFile]`: Fully qualified location to place the log file

Figure 6-12. The BPMSnapshotCleanup command (2 of 2) WB868 / ZB8681.0

Notes:

You must set at least one of the optional parameters such as `containerSnapshotAcronyms`, `keptNumber`, `createdBeforeLocal`, `createdAfterLocal`, or `createdBeforeSnapshotAcronym` as the filter for the `BPMSnapshotCleanup` command.

WebSphere Education

IBM

Automating snapshot deletion

- You can configure automatic cleanup of unnamed snapshots
  - Removes unnamed snapshots only, does not remove named snapshots
  - Runs only if the server is running
  - Introduced in fix pack 1 (V8.5.0.1)
- Randomly chooses which process applications and toolkits to work on
- Removes unnamed snapshots in chunks of 100 to limit database contention
- Done by adding entries to the `100Custom.xml` file
  - Examine the results by using the `SystemOut.log` file for the server

```
<unnamed-snapshots-cleanup-config>
   <enabled>true</enabled>
   <cleanup-start-time>23:59:59</cleanup-start-time>
   <cleanup-duration-minutes>5</cleanup-duration-minutes>
   <clean-after-number-named-snapshots>4</clean-after-number-named-snapshots>
</unnamed-snapshots-cleanup-config>
```


Figure 6-13. Automating snapshot deletion WB868 / ZB8681.0

Notes:

The entries in the file that are used include the following settings:

- **enabled**: Indicates whether to turn on the automatic deletion feature. The default setting is false.
- **cleanup-start-time**: Indicates the time of day when you want automated snapshot deletion to run. Local computer time is used, and the default time is midnight.
- **cleanup-duration-minutes**: Indicates the number of minutes that you want the process to run. The default duration is 5 minutes.
- **clean-after-number-named-snapshots**: Defines which snapshots you want to delete. The default setting is 4, which means that only unnamed snapshots that are older than the four most recently named snapshots are deleted.

WebSphere Education

Advanced content in Process Center (1 of 2)

- A process application might have content from Integration Designer, which indicates an Advanced Integration service (AIS)
  - AISes are implemented in SCA modules and deployed as EAR files
- For every process application or toolkit that contains an SCA module or library, a business-level application (BLA) is created
  - BLA is created for the current snapshot and every named snapshot
- Each snapshot corresponds to its own set of EAR files
- Proliferation can happen quickly, especially on the Process Center server
  - Created either when doing a playback from Process Designer or publishing to Process Center from Integration Designer
  - Can affect server start time, memory consumption, and general performance

Figure 6-14. Advanced content in Process Center (1 of 2) WB868 / ZB8681.0

Notes:

A process application has a business-level application (BLA), which acts as a container for the process application and its assets (including SCA modules, toolkits, and libraries). Each process application snapshot has its own BLA. Many of the administration tasks for a snapshot (for example, stopping or starting it on a production server) are done at the level of the BLA. This task allows for quicker and simpler administration of the snapshot and all of its assets.

WebSphere Education

Advanced content in Process Center (2 of 2)

- Have a regular practice of deleting the BLAs and EARs if they are no longer needed
  - You can delete all BLAs except the ones of the process application snapshot tip
  - There is no built-in method for purging this content
- Archiving a process application causes the BLA to be uninstalled and deleted
  - Uninstall of SCA BPELs causes instances to be deleted
- Best practice to use the AIS Facade pattern
  - You still get multiple copies of the facade EAR for each consuming process application
  - The EAR is smaller as it is only that small pass-through proxy

Figure 6-15. Advanced content in Process Center (2 of 2) WB868 / ZB8681.0

Notes:

For the suggested AIS Facade pattern, see **Best practices when using IBM Integration Designer and IBM Process Designer together** at:

`http://www.ibm.com/developerworks/websphere/bpmjournal/1106_taylor/1106_taylor.html`

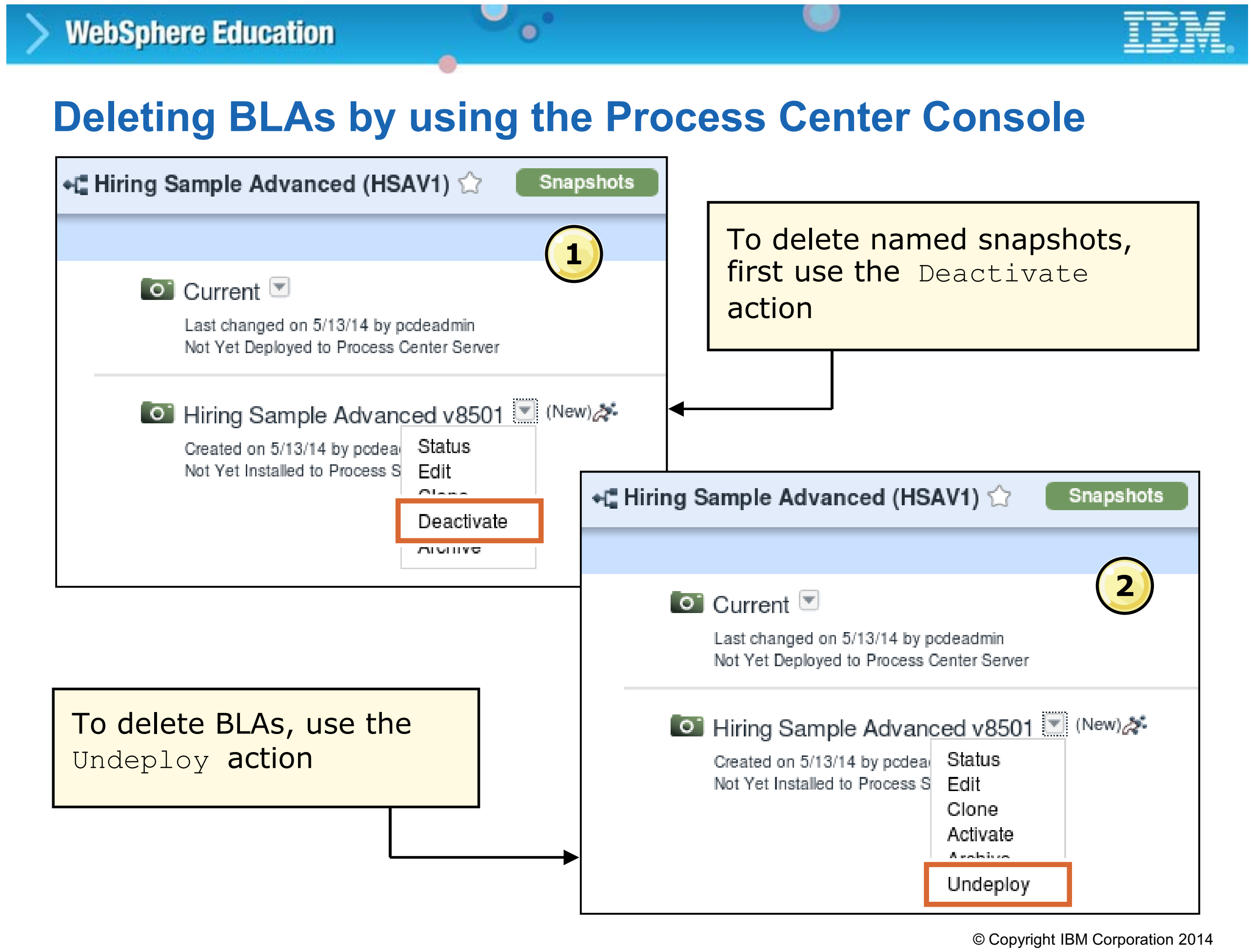


Figure 6-16. Deleting BLAs by using the Process Center Console WB868 / ZB8681.0

Notes:

If a process application snapshot contains Advanced Integration services (for example, SCA modules or BPEL processes), you can use the Process Center Console to undeploy a snapshot from the Process Center server.

When you activate a snapshot on a Process Center server, it is considered deployed on that server. To remove the Advanced Integration services artifacts and the associated business-level application (BLA) from the Process Center server, you must undeploy the snapshot.

6.3. Purging data in the Process Server environment

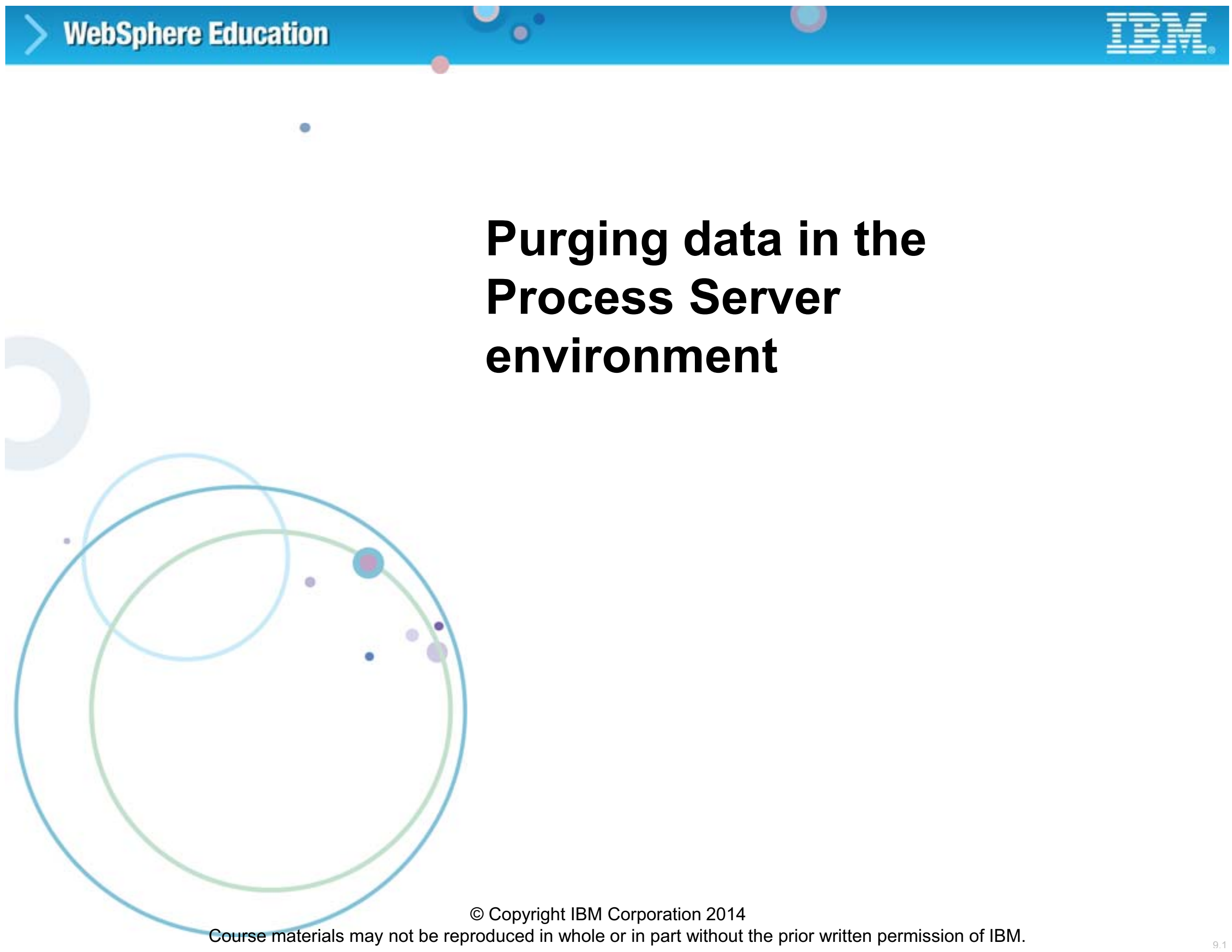


Figure 6-17. Purging data in the Process Server environment WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Snapshots

- You can delete inactive process application snapshots from a Process Server
- The process that you use varies depending on whether you use the Standard or Advanced editions of IBM Business Process Manager
- To delete snapshots on a process server, use the `BPMDeleteSnapshot` command
  – Required parameters include `containerAcronym` and `containerSnapshotAcronyms`
  – `containerAcronym`: Indicates the location of snapshots that are going to be deleted
  – `containerSnapshotAcronyms`: Indicates the named and archived snapshot

Figure 6-18. Snapshots WB868 / ZB8681.0

Notes:

The Process Server is where you install process application snapshots and where you run processes within them. What you need to think about primarily in the Process Server, in terms of accumulating content, is how to delete these installed snapshots, and how to delete processes as soon as they complete or terminate.

WebSphere Education

IBM.

Prerequisites for deleting a snapshot

- The snapshot must exist
- You cannot delete the default snapshot
- The snapshot must be inactive
- There must not be any running BPEL instances (Advanced only)
- The snapshot must not be deployed (Advanced only)

Figure 6-19. Prerequisites for deleting a snapshot WB868 / ZB8681.0

Notes:

You can delete an inactive process application snapshot on any test or production process server. The process that you use varies depending on whether you are using IBM Business Process Manager Advanced or IBM Business Process Manager Standard. You might want to delete snapshots and their dependencies if you no longer need them or if you have concerns about space on your system.

WebSphere Education

Steps for deleting a snapshot

- The snapshot must exist
  - Use the `BPMShowProcessApplication` command to verify
- You cannot delete the default snapshot
  - Use the `BPMShowSnapshot` command to see whether the snapshot is the default and whether there are running instances
- The snapshot must be inactive
  - Use the `BPMDeactivate` command to deactivate the snapshot
  - Also stops BPMN processes and allows running instances to acquiesce
- There must not be any running BPEL instances (Advanced only)
  - Use the `BPMStop` command
  - Stops the BPEL process for the snapshot and allows running instances to acquiesce
- The snapshot must not be deployed (Advanced only)
  - Use the `BPMUndeploy` command
  - Removes BLAs and EAR related to the snapshot
  - There must be no active BPEL instances or the command fails

Figure 6-20. Steps for deleting a snapshot WB868 / ZB8681.0

Notes:

When the default snapshot of a process application is deleted, the process application is removed. Toolkits that the process application depended on remain. Other toolkits and process applications might depend on those remaining toolkits. The toolkits cannot be deleted while those dependencies exist. In this case, you must:

- Deactivate the toolkit snapshot
- Stop the toolkit snapshot
- Get a list of toolkits and process applications that have dependencies on the toolkit snapshot
- Starting with the root of the list of reported dependencies, remove each dependency
- For each dependency that you want to remove, delete the snapshot of the toolkit or process application
- After you resolve all of the dependencies, delete the snapshot

WebSphere Education IBM

Instances in Process Server

- There are two types of instances to consider
  - User or human task instances
  - Process instances
- Two types of instances for both BPD and BPEL processes
  - Archiving is not supported for Standard task and process instances
  - Archiving is supported for Advanced BPEL process instances
- To delete BPD instances on a process server, use the `BPMProcessInstancesCleanup` command
  - Identify the specific instances to delete or the date range within which any instances that completed are going to be deleted
  - You can also indicate a state of completed, failed, or terminated
  - Enhanced in V8.5.0.1 to include the maximum deletion time and the size of the transaction for the delete operation
- Deleting BPEL processes is covered in the “Business process choreography cleanup” topic

Figure 6-21. Instances in Process Server WB868 / ZB8681.0

Notes:

Using the `BPMProcessInstancesCleanup` command deletes the BPD instance and its associated tasks for the instances that the command parameters specify. It also logs data to a standard `systemout.log` file to track which process applications were selected for deleting instance data.

To run this command, you must be a user in the DeAdmin role. Also, you can run this command only in the connected mode. In a network deployment environment, you must run this command on the node that contains the application cluster member that handles Process Server applications. Do not run this command from the deployment manager profile.

WebSphere Education IBM

Standard process and task instances

- When you delete process instances, task instances are also deleted
  - Archiving is not supported in the product
  - No method to delete task instances only
- System tasks also record entries in the task table, which can be avoided

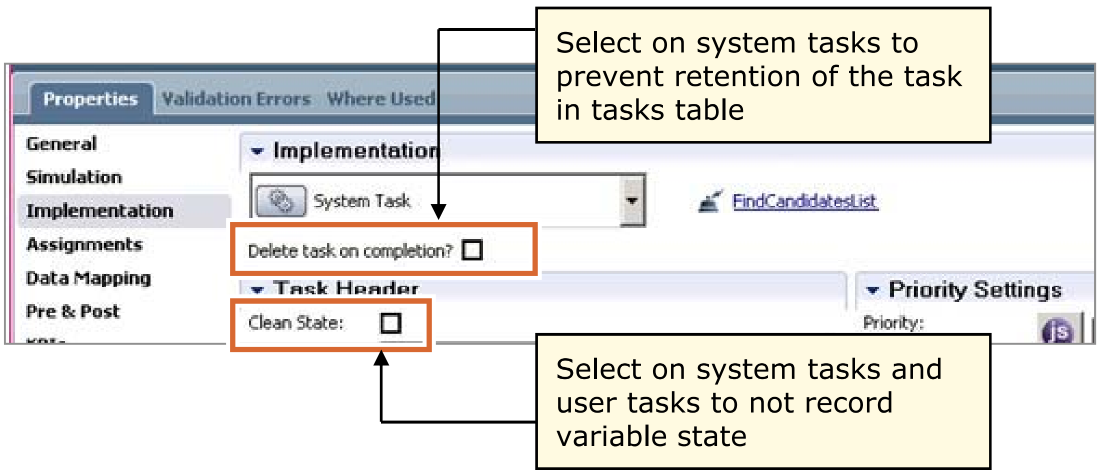


Figure 6-22. Standard process and task instances WB868 / ZB8681.0

Notes:

Note, however, that system tasks also record entries in the task table. These tasks can be avoided by using the check box **Delete task on completion** in the Implementation tab for its properties in Process Designer. Also, for both user tasks and system tasks, the check box **Clean slate** deletes the context (variable) information for tasks as they complete, which can have a significant impact on amount of data retained.

6.4. Business process choreography cleanup

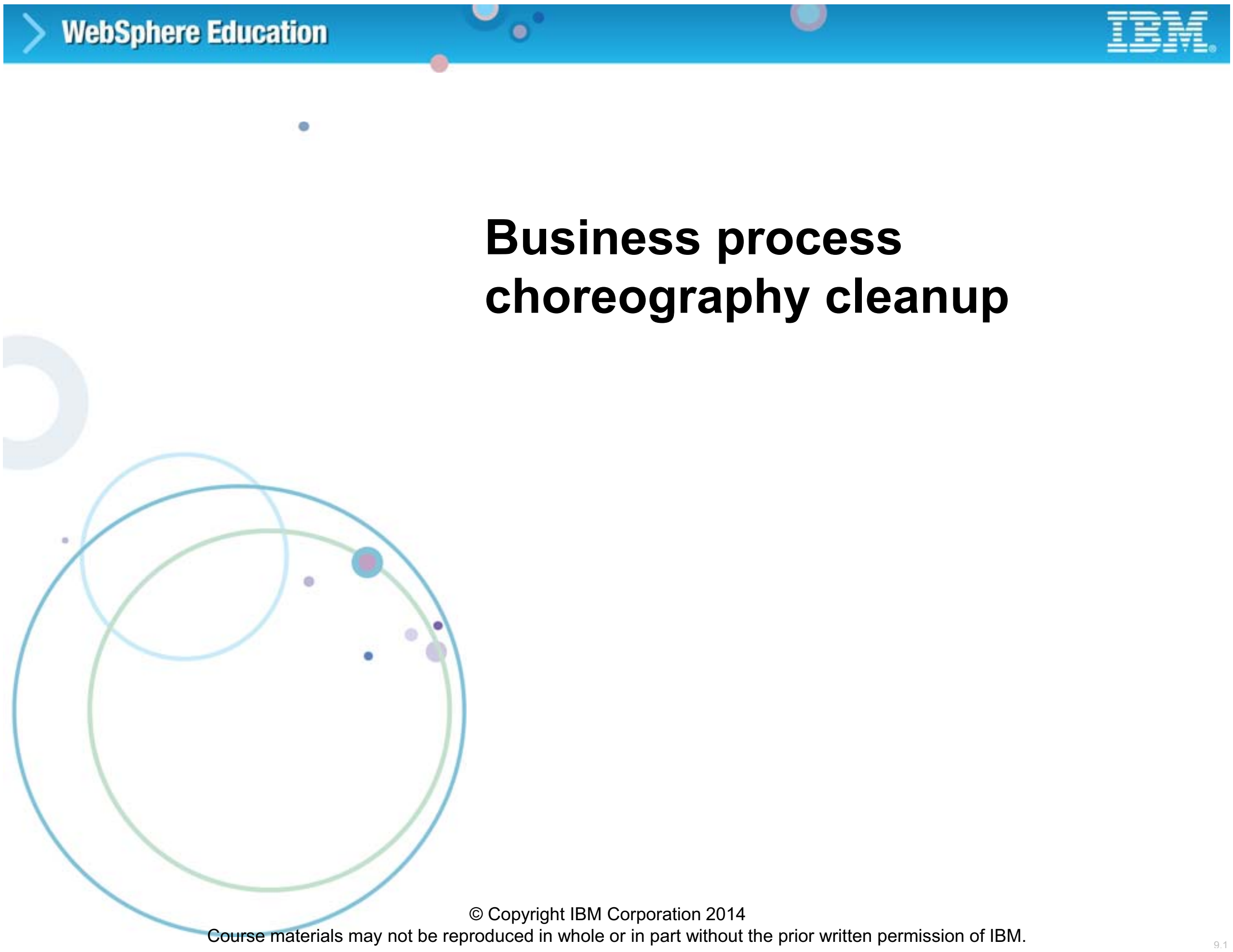


Figure 6-23. Business process choreography cleanup WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Periodic cleanup operations

- The database should contain running processes only, and completed processes should be deleted
  - Artifacts that are related to instances are deleted when the instance is deleted
- The number of processes in the database has an impact on performance
- Various cleanup procedures are available for the Process Server administrator to clean up the database
  - Cleanup service
  - Administrative console
  - Administrative scripts
  - Failed Event Manager
  - Business Process Choreographer Explorer
  - Business Process Choreographer APIs

Figure 6-24. Periodic cleanup operations WB868 / ZB8681.0

Notes:

Administrative console and Failed Event Manager: Messages that cannot be processed are placed on the hold queue, which includes messages for instances that are deleted. You can empty the hold queue by replaying the messages in the queue, which causes any messages for deleted instances to be discarded.

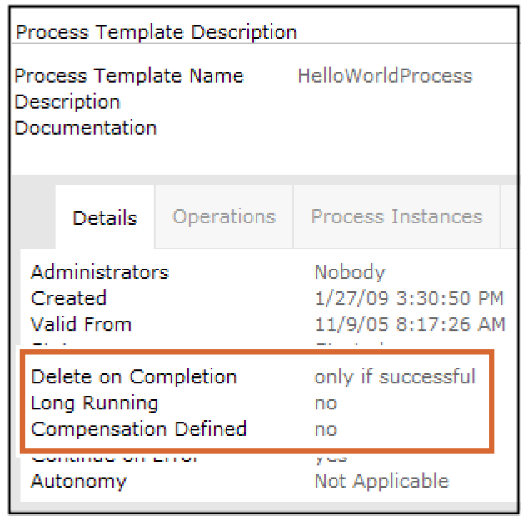
Business Process Choreographer Explorer: Use the Business Process Choreographer Explorer to delete tasks or processes individually.

Business Process Choreographer APIs: You can write your own cleanup tool that uses the Business Process Choreographer APIs to delete process instances, task instances, and task templates that were created at run time by using the APIs. Templates that are part of an enterprise application cannot be deleted by using the APIs.

WebSphere Education

Deletion of completed instances

- Deletion of completed process instances can be specified in the process or task properties
  - **Yes**: Delete instance if in FINISHED, TERMINATED, or FAILED state
  - **On successful completion** (default): Delete only if not in FAILED state
  - **No**: Do not delete
- Set properties in business model by using IBM Integration Designer



- Settings can be viewed by using Business Process Choreographer Explorer on process template **Details** tab

Figure 6-25. Deletion of completed instances WB868 / ZB8681.0

Notes:

The “Automatically delete the process after completion” setting in IBM Integration Designer is relevant only with long-running processes. It determines how the runtime environment will deal with the process instance that the process uses after it runs. You have the following options:

- **Yes**: Choose Yes to delete the data that is associated with this instance of the process after it completes. This setting removes the process instance, whether the process completed successfully or not.
- **On successful completion**: In this case, the data remains in the database when the process fails so that the problem can be traced and the process administrator can restart the process, if required.
- **No**: Choose No to not delete the data that is associated with this process after it completes.

WebSphere Education IBM

Cleanup service

- ***Cleanup service*** allows scheduled deletion of instances and tasks
  - Specify administratively which instances to delete and when
- ***Cleanup service*** configuration specifies:
  - When the cleanup service runs (during off-peak hours)
  - How long the cleanup service runs (maximum duration in minutes)
  - How many instances to delete in one transaction
  - Which instances to delete
- ***Cleanup job*** configuration specifies:
  - Template names and namespaces of instances to delete
  - End states of instances subject to the cleanup job
  - Time instances are kept after completion before they are eligible for deletion by cleanup service
  - Ordered list of jobs to run

Figure 6-26. Cleanup service WB868 / ZB8681.0

Notes:

Identify times of the day and days of the week, when it would be best to schedule the cleanup service; for example, when there is the lowest load on the database. For each BPEL process and human task that you want the cleanup service to delete, decide which states make an instance a candidate for deletion. Then, decide how long an instance must be in one of those states before the next scheduled cleanup deletes them.

WebSphere Education

Cleanup service configuration

- Separate cleanup services for business processes and human tasks
- Configuration that is done by using the administrative console
  - Business Flow Manager configuration panel for business processes
  - Human Task Manager configuration panel for human tasks

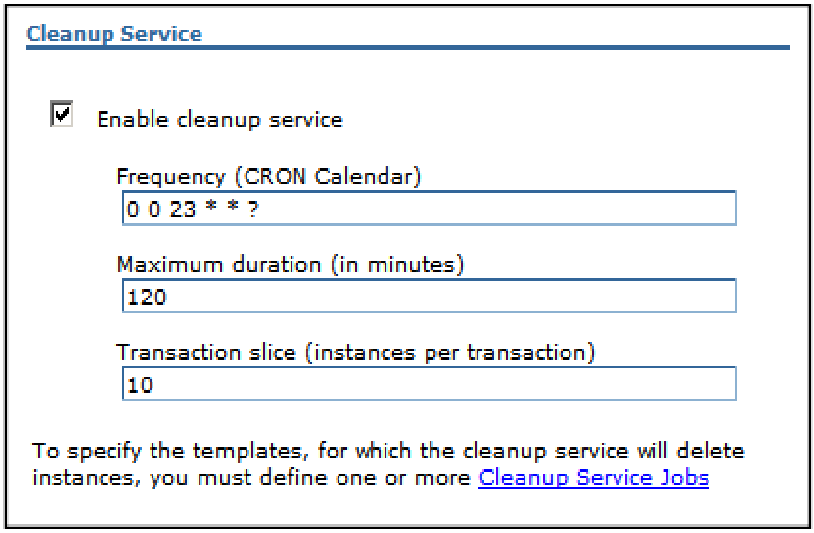


Figure 6-27. Cleanup service configuration WB868 / ZB8681.0

Notes:

You want to delete completed instances automatically after keeping them for a while. There is a separate cleanup service for the Business Flow Manager and for the Human Task Manager. For each of them, you must first enable the service and define the service parameters, such as the schedule, maximum duration of the cleanup, and the database transaction size. Then, you can define cleanup jobs for sets of templates and define the end states and the duration that an instance must be in to qualify for deletion.

The Human Task Manager cleanup service deletes stand-alone human tasks. When the Business Flow Manager cleanup service deletes a BPEL process, it also deletes all of the child processes and inline human tasks that are contained in the process. When security is enabled, the cleanup user ID specified for the Business Process Choreographer configuration must be in the business administrator role.

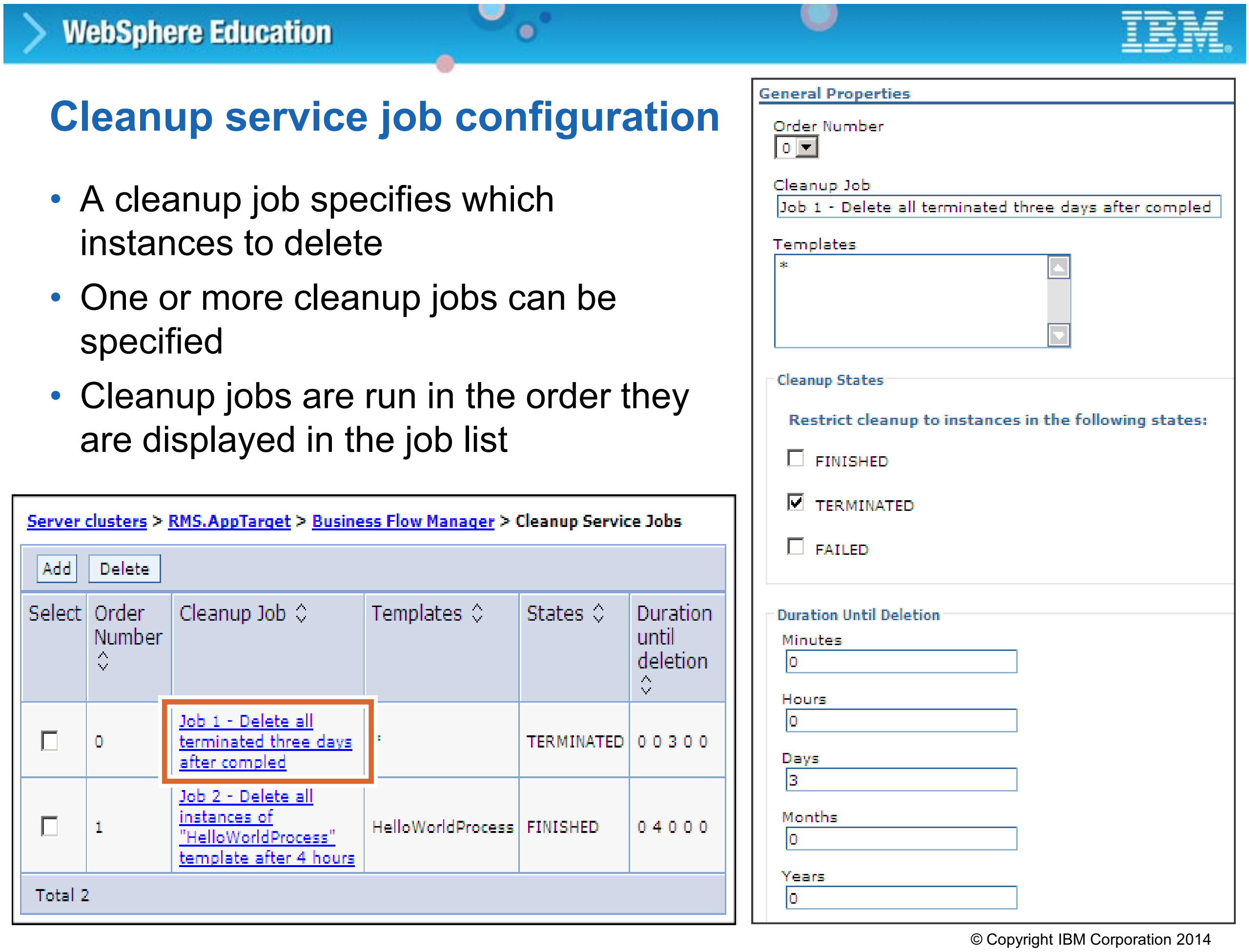


Figure 6-28. Cleanup service job configuration WB868 / ZB8681.0

Notes:

You activate the cleanup service and defined cleanup jobs to delete completed instances. When the cleanup service starts and finishes, the messages CWWBF0118I and CWWBF0119I are written to the `SystemOut.log` file. When one cleanup job starts and finishes, the messages CWWBF0116I and CWWBF0117I are written to the `SystemOut.log` file. Progress updates of the cleanup processing are written with message CWWBF0120I to the `SystemOut.log` file.

WebSphere Education

Deleting process templates by using scripting

- Use the `deleteInvalidProcessTemplate.py` script to remove, from the database, the templates, and all objects that belong to the template that are not contained in any corresponding valid application
  - Might occur if an application installation was canceled or not stored in configuration repository by the user
  - Use the `deleteInvalidTaskTemplate.py` script for human task templates
  - Location: ***<install_root>*/ProcessChoreographer/admin**

```
<install_root>/bin/wsadmin -f deleteInvalidProcessTemplate.py
 (([-node <node_name>] -server <server_name>] |
    -cluster <cluster_name>))
 -templateName <template_name>
 -validFromUTC <validFromString>
```


Figure 6-29. Deleting process templates by using scripting WB868 / ZB8681.0

Notes:

Use the `deleteInvalidProcessTemplate.py` script to remove from the database those templates that are not contained in any corresponding valid application in the configuration repository. This removal includes all of the objects that belong to these templates. This situation can occur if an application installation was canceled or not stored in the configuration repository by the user. These templates usually have no impact. They are not shown in Business Process Choreographer Explorer.

WebSphere Education IBM

Deleting process instances by using scripting

- Use the `deleteCompletedProcessInstances.py` script to delete completed process instances
  - Location: ***<install_root>*/ProcessChoreographer/admin**
  - Indicate the criteria to selectively delete process instances

```
<install_root>/bin/wsadmin -f -deleteCompletedProcessInstances.py
 [([-node <node_name>] -server <server_name> | (-cluster <cluster_name>)]
 (-all | -finished | -terminated | -failed)
 [-templateName <template_name> [-validFromUTC <timestamp>]]
 [-startedBy <userID>
 [(-completedAfterLocal <timestamp>) | (-completedAfterUTC <timestamp>)]
 [(-completedBeforeLocal <timestamp>)| (-completedBeforeUTC <timestamp>)]
```


Figure 6-30. Deleting process instances by using scripting WB868 / ZB8681.0

Notes:

Run this script when all completed process instances are deleted.

WebSphere Education IBM

Deleting task instances by using scripting

- Use the `deleteCompletedTaskInstances.py` script to delete completed task instances
  - Location: ***<install_root>*/ProcessChoreographer/admin**
  - Indicate the criteria to selectively delete task instances

```
<install_root>/bin/wsadmin -f -deleteCompletedTaskInstances.py
 [([-node <node_name>] -server <server_name> | (-cluster <cluster_name>)]
 (-all | [-finished] [-terminated] [-failed] [-expired])
 [-templateName <template_name> -nameSpace <name_space>
     [-validFromUTC <timestamp>]]
 [-createdBy <userID>
 [(-completedAfterLocal <timestamp>) | (-completedAfterUTC <timestamp>)]
 [(-completedBeforeLocal <timestamp>)| (-completedBeforeUTC <timestamp>)]
```


Figure 6-31. Deleting task instances by using scripting WB868 / ZB8681.0

Notes:

Run this script to delete completed task instances.

WebSphere Education IBM

Deleting audit log entries by using scripting

- Use the `deleteAuditLog.py` script to delete audit log entries from the database
  - Location: ***<install_root>*/ProcessChoreographer/admin**

```
<install_root>/bin/wsadmin -f -deleteAuditLog.py
 [([-node <node_name>] -server <server_name>) | (-cluster <cluster_name>)]
 (-all | -timeUTC <timestamp> | -timeLocal <timestamp>
       | -processtimeUTC <timestamp> | -processtimeLocal <timestamp>)
  [-slice <size>]
```


Figure 6-32. Deleting audit log entries by using scripting WB868 / ZB8681.0

Notes:

Use an administrative script to delete some or all audit log entries for the Business Flow Manager.

WebSphere Education

Archiving BPEL process instances

- The Business Process Archive Manager is an optional component that allows you to move completed BPEL process instances and human tasks to an archive database
  - A script is used to moved instances and tasks from the Business Process Choreographer database
- The BPEL process archive facility consists of the following components:
  - Business Process Archive Manager
  - Business Process Archive Explorer
  - Business Process Archive database
  - The `archive.py` script
  - Business Process Archive Manager EJB API

Figure 6-33. Archiving BPEL process instances WB868 / ZB8681.0

Notes:

To create a Business Process Archive Manager, you must add entries to the properties file before you create the deployment environment. Each Business Process Archive Manager requires its own database. The database must be of the same type and structure as is used for the Business Process Choreographer database. The default name for the archive database is BPARCDB.

For more information on this topic, go to:
`http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r5m0/topic/com.ibm.wbpm.imuc.doc/topics/c5_archive.html`

6.5. More areas to purge data

WebSphere Education

More areas to purge data

Figure 6-34. More areas to purge data WB868 / ZB8681.0

Notes:

WebSphere Education

Performance Data Warehouse (1 of 2)

- Support for purging Performance Data Warehouse data
  - Available in fix pack 1 (V8.5.0.1)
  - Currently, no support for archiving data
- Use the `prune` command that is part of the `perfDWTool`
  - Indicate the age of the data that should be removed by using the `prune` option
  - Run the command from an active node in the support cluster
  - Verify that are servers in the support cluster are running
  - Create a backup of the Performance Data Warehouse database

```
<install_root>/profiles/<profile_name>/bin/perfDWTool.sh
    -u user_name -p password -nodeName node_name prune
    -daysOld number-of-days-old
```


Figure 6-35. Performance Data Warehouse (1 of 2) WB868 / ZB8681.0

Notes:

BPD processes can support tracking, which means events are sent to the Performance Data Warehouse and logged in its database. How many events are sent is primarily determined by whether your BPD has autotracking turned on or not, which is the default for new BPDs.

WebSphere Education

Performance Data Warehouse (2 of 2)

- You can affect the running of the prune command by updating various settings in the `100Custom.xml` file

- `prune-batch-size`
  - Indicates the number of records to be deleted in a single prune operation
  - Default value is 10000
  - If the value is set too high, the increased load on the transaction log can result in errors
- `prune-operation-time-box`
  - Indicates the amount of time that the prune command runs
  - Default value is 10800 seconds
- `prune-operation-time-box-retry`
  - Specifies the number of times that the prune request can be tried
  - Default value is 4, which means it is tried again three times

Figure 6-36. Performance Data Warehouse (2 of 2) WB868 / ZB8681.0

Notes:

You can optionally change several configuration properties in the Performance Data Warehouse configuration file. You can set the batch size for pruning, indicate the timeout period for the prune operation, or specify how many times the operation can be tried again.

WebSphere Education

Durable subscription events

- Message events in an intermediate message activity of a BPD can be configured as durable
  - Configured within Process Designer
- Durable messages accumulate in the database and require occasional cleanup
  - Support available for purging durable subscription events by using the `BPMDeleteDurableMessages` command
  - Introduced in fix pack 1 (V8.5.0.1)

```
<install_root>/profiles/<profile_name>/bin/wsdamin.sh -c
    BPMDeleteDurableMessages -olderThan age_in_days
    -maximumDuration number_of_minutes
    -transactionSlice number_of_messages
```


Figure 6-37. Durable subscription events WB868 / ZB8681.0

Notes:

Historically, there was no built-in way to do this cleanup, but this problem is addressed in V8.5.0.1 with a new wsadmin command called `BPMDeleteDurableMessages`, which takes three parameters:

- `olderThan`: Only events older than this number of days are deleted.
- `maximumDuration`: The command runs only for this amount of time.
- `transactionSlide`: The number of events to delete per transaction.

WebSphere Education

Process Center index data

- BPM uses an index for optimizing searches that are done in Process Center
  - The data is automatically maintained but you can force an update if necessary
  - The index is stored on disk for each node
- You can configure if and when the index is automatically updated and where the data is saved
  - Update the `100Custom.xml` file
  - Modify the location of the index on disk by using WebSphere environment variables
- You can also rebuild or update the index
  - `artifactIndexFullReindex.sh`
  - `artifactIndexUpdate.sh`

Figure 6-38. Process Center index data WB868 / ZB8681.0

Notes:

The Process Center index is used to conduct searches on the Process Center repository. The index is automatically created and maintained when the server is started. After that, the system updates the index at regular intervals to reflect any changes that are made to the repository. You can configure the update interval and index location to better suit your installation. Also, commands are provided for you to manually re-create or update the index.

WebSphere Education

Process Portal index data

- Process Portal uses an internal in-memory index that allows process participants to search business processes for instance data
  - The index is also used for historic data in the Process Performance and Task Performance dashboards
  - A small amount of that index is stored on disk
- You can configure the indexing behavior and
  - Update the `100Custom.xml` file
- You can also rebuild or clean the index
  - `processIndexFullReIndex.sh`
  - `processIndexRemoveDeleted.sh`

Figure 6-39. Process Portal index data WB868 / ZB8681.0

Notes:

If a problem occurs with the Process Portal index, you might need to run a command to rebuild it. You can also update the index for an instance or task, or remove an instance or task from the index. The `processIndexFullReIndex.sh` command deletes the existing index and creates a new index. While the index is being built, the search facility in Process Portal is unavailable.

When tasks and instances are deleted from the database, they are also automatically deleted from the index. You can manually delete tasks and instances from the index that were previously deleted from the database. You can use the `processIndexRemoveDeleted.sh` command.

WebSphere Education

WebSphere logs

- The `SystemOut.log` file
  - Verify that there is sufficient disk space for your log file configuration
  - Log files automatically roll over and purge the oldest data
  - You can implement custom or third-party background batch processes to capture the oldest logs before rollover
  - Use the HPEL log format as a best practice
- Transaction log file
  - Never delete the transaction log file
  - The file is self-maintaining
  - Verify that the file is on a highly available file system
  - Option to store transaction log file in a database versus the file system

Figure 6-40. WebSphere logs WB868 / ZB8681.0

Notes:

The transaction (tranlog) log file stores critical transactional data that is written to databases. It is an internal file that WebSphere Application Server uses to manage in-flight transactions and attempt to recover them if the server locks up.

Deleting this file removes information on in-flight transactions from IBM Business Process Manager memory. Without the transaction log file, there is no functionality to recover transactional information. In addition, long-running processes remain in an inconsistent state and you cannot complete the process flow except by deleting running instances. Deleting running instances might cause you to lose operational or business-critical data, which makes the database inconsistent with the message destination. Other inconsistencies that might be caused by deleting the transaction log file include the following issues:

- Started transactions are not rolled back or committed.
- Artifacts remain in the Java virtual machine (JVM) since a transaction references or allocates them, but they are never garbage collected.
- Database contents (among others, navigation state of long-running BPEL processes) remains in the Business Process Choreographer-related tables and are never deleted

- Navigation messages of the business process engine (BPE) of long-running processes are never processed further.
- Service Component Architecture (SCA) messages that belong to a process navigation and transaction remain on SCA-related queues.

Deleting the transaction log from a development environment causes the same problems. Because you can re-create business processes, deleting the files from a test environment is not as damaging as deleting them from a production environment.

WebSphere Education

Temporary directories

- The system temp directory is used for a number of purposes by Business Process Manager and accumulates files
  – Used during the installation and also during the use of the Quick Start
  – When snapshots are exported, temporary files are stored as .zip files and quickly accumulate
- Periodically monitor and purge the system temp directory
  – You might want to archive the data before purging
- WebSphere creates three temporary directories under the *<profile_name>* directory
  – Directories include `temp`, `wstemp`, and `config/temp`
  – Delete the `temp` and `config/temp` directories after stopping all servers on the profile node
  – Delete the `wstemp` directories after stopping the deployment manager

Figure 6-41. Temporary directories WB868 / ZB8681.0

Notes:

WebSphere Education

Review

Data	Options available
Process applications and toolkits in Process Center	Archive and delete via Process Center Console
Snapshots in Process Center	`BPMSnapshotCleanup` command plus automated method in V8.5.0.1
Advanced BLA and EARs in Process Center	Undeploy by using the Process Center Console to remove BLA and EARs
Snapshots in Process Server	`BPMDeleteSnapshot` command
BPD process instances	`BPMProcessInstancesCleanup` command, which is updated in V8.5.0.1
BPEL process instances	`deleteCompletedProcessInstances.py` script or Cleanup Server
BPD durable events	`BPMDeleteDurableMessages` command in V8.5.0.1
Performance Data Warehouse	`prune` command in V8.5.0.1

Figure 6-42. Review WB868 / ZB8681.0

Notes:

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Explain the need for purging data in the environment
- Archive, restore, and delete process applications and toolkits
- Archive and delete snapshots in the Process Center environment
- Delete unnamed snapshots
- Automate snapshot deletion
- Delete business-level applications
- Delete snapshots and instances in the Process Server environment
- Configure the cleanup service
- Define various business process choreography cleanup operations
- Explain more methods for purging data in the environment

Figure 6-43. Unit summary WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Checkpoint questions

1. True or false: Named snapshots must be archived first before deletion.

2. True or false: The `BPMProcessInstancesCleanup` command can be used to clean up both BPMN and BPEL instances.

3. True or false: The cleanup service allows for scheduled deletion of process instances and tasks administratively.

Figure 6-44. Checkpoint questions WB868 / ZB8681.0

Notes:

Write your answers here:

1.

2.

3.

WebSphere Education

Checkpoint answers

1. True

2. False: The `BPMProcessInstancesCleanup` command can be used to clean up only BPMN, not BPEL instances.

3. True

Figure 6-45. Checkpoint answers WB868 / ZB8681.0

Notes:

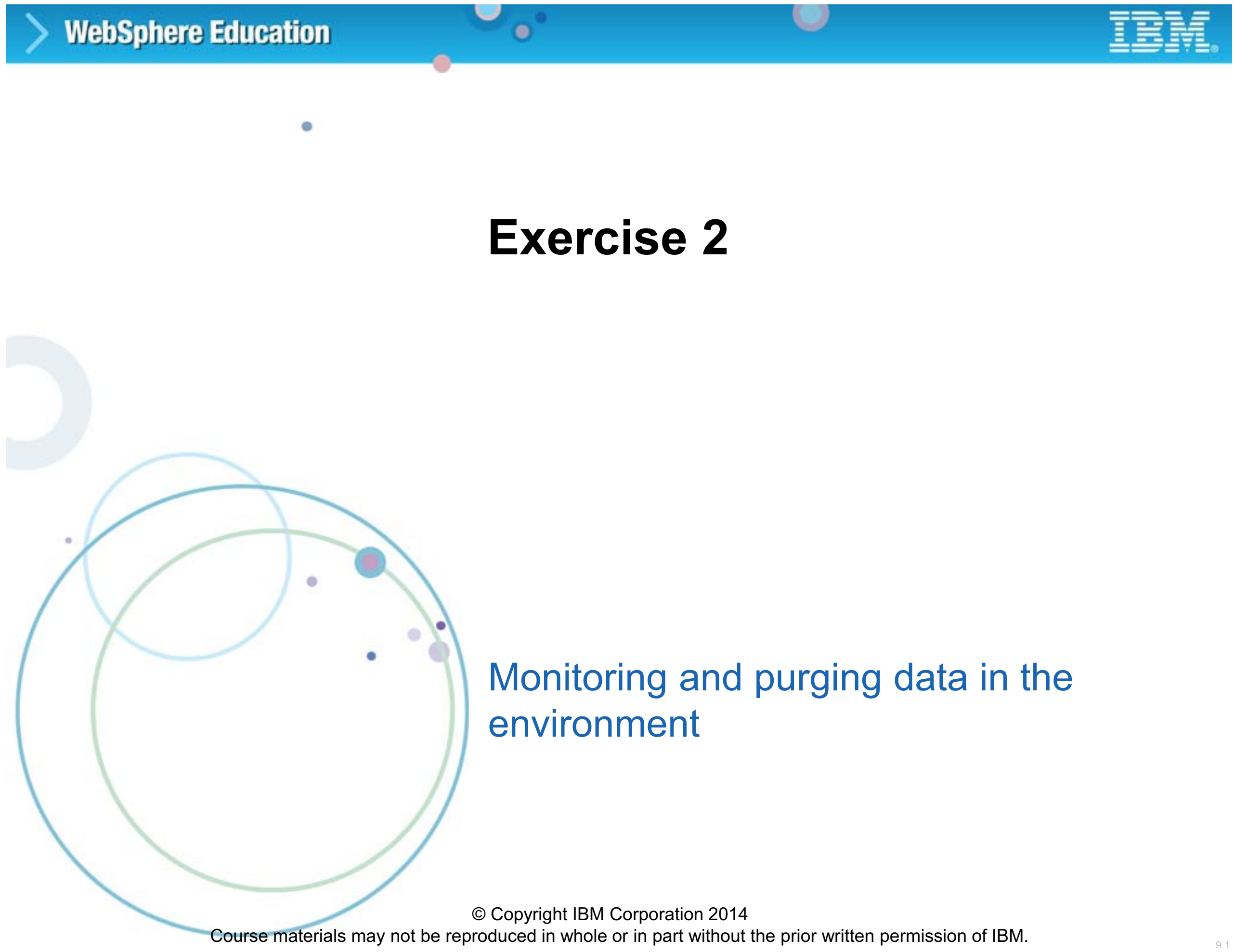


Figure 6-46. Exercise 2 WB868 / ZB8681.0

Notes:

WebSphere Education

Exercise objectives

After completing this exercise, you should be able to:

- Archive and delete process applications
- Manage and delete snapshots, both named and unnamed
- Configure an automated method for deleting unnamed snapshots
- Use the Process Admin Console to monitor the environment
- Use Process Portal to monitor process applications
- Purge data in the Process Center and Process Server environment
- Delete BPMN and BPEL instances in the Process Server environment
- Delete snapshots in the Process Server environment

Figure 6-47. Exercise objectives WB868 / ZB8681.0

Notes:

Unit 7. Performance considerations for coaches

What this unit is about

This unit introduces a number of guidelines for developing and deploying efficient coaches.

What you should be able to do

After completing this unit, you should be able to:

- Explain how coaches can benefit your organization
- Define good practices for developing Coach Views
- Identify the requirements for good response times when users interact with coaches
- Explain modeling capabilities that impact performance
- Describe deployment considerations for Coach Views

How you will check your progress

- Checkpoint

References

IBM Business Process Manager V8.5 IBM Knowledge Center

`http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/ditamaps/ic-homepage-bpm.html`

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Explain how coaches can benefit your organization
- Define good practices for developing Coach Views
- Identify the requirements for good response times when users interact with coaches
- Explain modeling capabilities that impact performance
- Describe deployment considerations for Coach Views

Figure 7-1. Unit objectives WB868 / ZB8681.0

Notes:

WebSphere Education

Topics

- Overview of coaches
- Good practices for design and development of coaches
- Deployment considerations for coaches

Figure 7-2. Topics WB868 / ZB8681.0

Notes:

External user interface considerations

- It is possible to use existing pages and integrate them with IBM Business Process Manager
  - Many organizations undertake studies to determine the cost benefit of replacing user interfaces with coaches
- Advantages of external user interfaces
  - Can use existing web pages, do not have to re-create screens in coaches
  - Web developers do not have to learn the Process Designer

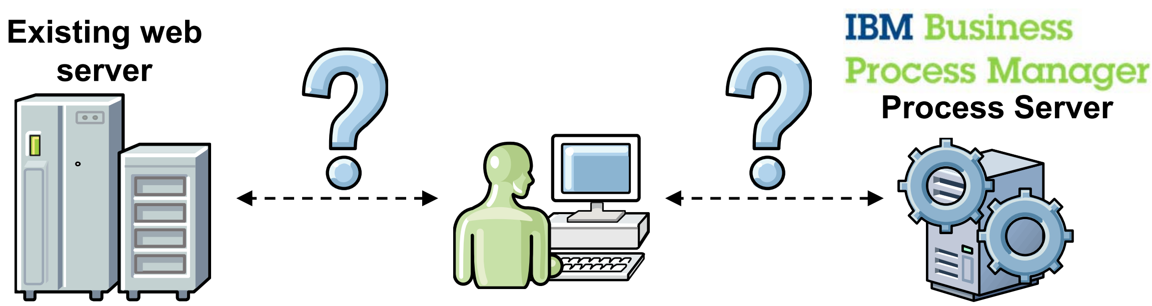


Figure 7-3. External user interface considerations WB868 / ZB8681.0

Notes:

When an organization adopts IBM Business Process Manager (IBM BPM), it is left with a decision to make. Should you use the existing infrastructure to serve web pages to users, or should you replace the existing infrastructure with the coach capabilities of IBM BPM? There are many studies on the cost benefits for both approaches. Both approaches require development time and effort, but which approach is the best for your organization?

The main advantages of using an existing web server and communicating back to IBM BPM revolve around the rework time that is required to re-create the user interfaces. Another consideration is the training time that is required for developers to become proficient with coaches.

WebSphere Education IBM

Disadvantages of external user interfaces

- BPM assets are now distributed among multiple systems
  - New versions of BPM assets and deployment must synchronize with the external web pages
  - One-click deployment inside of IBM Business Process Manager is no longer possible
  - Project asset lifecycle governance becomes more complex
- Must create integrations to the existing web server for every activity
- Must create integrations to IBM Business Process Manager from the existing system
- Security exchange, authentication, and authorization decisions become complex, and must be synchronized between systems
- Obtaining metrics on user performance inside their activities is impossible or complicated
- The process data system of record inside IBM Business Process Manager might not be authoritative

Figure 7-4. Disadvantages of external user interfaces WB868 / ZB8681.0

Notes:

There are numerous disadvantages to externalizing all user interfaces from IBM Business Process Manager. Here are a few problems developers must consider when designing an external user interface.

WebSphere Education

Advantages and disadvantages of coaches

- Advantages of coaches include:
  – WYSIWYG editor, rapid application development tool to create user interfaces
  – Coach data that is bound server-side without coding
  – Can be customized to match corporate themes with CSS
  – All advanced features (JavaScript, Ajax, and others) can be used
  – Centralizes all BPM assets, eliminating version conflicts and governance complexity
  – Enables the collection of BPM metrics at the service level to improve activity performance

- Disadvantages of coaches include:
  – New software requires training for web developers
  – All user interfaces must be re-created and tested

Figure 7-5. Advantages and disadvantages of coaches WB868 / ZB8681.0

Notes:

There are numerous advantages to eliminating or re-creating the existing user interfaces, and those benefits extend beyond the creation of the first release of development. The long-term sustainability of using coaches is dramatically improved if an organization decides to re-create its user interfaces inside of IBM BPM.

The disadvantages of coaches are the advantages of the external user interface. Re-creating the user interfaces in IBM BPM requires development time.

WebSphere Education

External user interface decision

- Organizations must make their own determination
- External user interface approach becomes two disjointed systems that must synchronize together
- In most cases, the disadvantages of an external user interface outweigh the advantages

Existing web server

IBM Business Process Manager Process Server

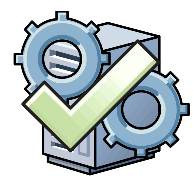


Figure 7-6. External user interface decision WB868 / ZB8681.0

Notes:

There are many advantages and disadvantages to each approach, and only the needs of the organization can determine which approach is best. Realize the pros and cons of using pre-existing user interfaces on existing systems. There are, however, occasions when the system must synchronize with existing information and the UI design within an organization. It requires customizing coaches to match the external UI and retrieve external data, or integrate with the existing UI altogether.

Project managers must determine the resource costs necessary to build, deploy, and maintain a more complex application if an external user interface is used. Many clients analyze the costs and determine that creating coaches from existing web pages comes at a much lower cost and reduced project timeline. Compare this approach to building a hybrid framework where an existing system and IBM Business Process Manager must coexist. Even if the project requires training for web developers to create and implement coaches, by using coaches instead of using an external UI, it ends up taking less time and costing less. Setting up the infrastructure, creating the artifacts to communicate between systems, and the service design and implementation time necessary to synchronize two systems with each other must be planned for before modeling or implementation can occur.

WebSphere Education

Coach overview

- Two types of user interface for human services:
  - Dashboards
  - Task completion
- Coaches provide:
  - A way for team members to input business and process data into the process in a task completion or dashboard
  - A wizard-like approach for team members to complete their tasks
- The coach is rendered in a browser for the team member when the coach service (the BPD activity of the participant) is run
- Benefits of using coaches:
  - Seamless integration of user interfaces (UIs) and process logic
  - Tailored UI components for your business

Figure 7-7. Coach overview WB868 / ZB8681.0

Notes:

The coach designer is also known as the next generation coach designer and supports Web 2.0 technology.

The coach designer allows web developers to create and distribute their custom Coach Views. It also allows business developers to define reusable sections and build richer user interfaces easily and quickly.

Coaches feature reusable **Coach Views**, which are reusable user interfaces that you can create and customize. Coach Views consist of one or more Coach Views, data bindings, layout information, and behaviors. Because Coach Views are reusable, you can create a library of common user interfaces and behaviors that you can use to rapidly develop new coaches. For greater flexibility in creating a service flow, Coach Views can broadcast boundary events that you use to connect nodes in the service.

WebSphere Education

Coach Views

- Coaches contain one or more Coach Views
- Coach Views are the reusable user interface building blocks that authors use to compose coaches
  - Provide the user interface elements and layout for the coach
  - Examples include components that allow users to edit text, push buttons, select items from a list, and select check boxes and radio buttons
  - Can be used to define the physical layout of a coach, which includes vertical and horizontal sections, tables, and tab controls
- BPM includes various Coach Views for presenting and manipulating process data
  - IBM, Business Partners, and clients are creating and distributing Coach Views
  - Dramatically increases the power of the coach framework
  - Stock content controls are Coach Views that are included with BPM in the Content Management (SYSCM) toolkit

Figure 7-8. Coach Views WB868 / ZB8681.0

Notes:

Contained within a page are visual building blocks that, when aggregated together, form the content of the page. IBM BPM calls each of these building blocks a Coach View. Other UI technology products (such as Dojo) would call them widgets or components.

7.1. Design and development considerations for coaches

WebSphere Education

Design and development considerations for coaches

9.1

Figure 7-9. Design and development considerations for coaches WB868 / ZB8681.0

Notes:

WebSphere Education

Design for application responsiveness

- A good design criterion for any application is the page responsiveness
  - Balance the responsiveness needs and functionality enhancements to create a great user experience

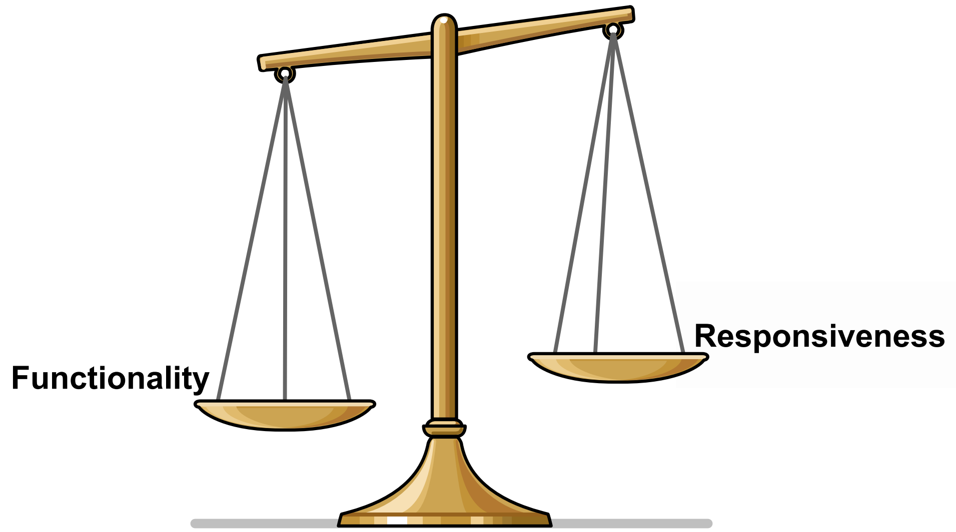


Figure 7-10. Design for application responsiveness WB868 / ZB8681.0

Notes:

A good design criterion for any application is the page responsiveness. For example, a large amount of data that is retrieved before a coach screen is shown can affect page responsiveness and the overall user experience. In such a case, a delayed user experience is less than desirable.

However, an example of good functional responsiveness is a "type ahead" feature that can be added to your user interface design. It gives the perceived value of increased responsiveness from your application to the user. It increases the positive user experience without a huge maintenance cost.

Using design tools like YSlow and the Firebug Profiler can help achieve fast user interfaces. These tools can also help identify individual elements in a coach that take longer to load, and assist developers in addressing these deficiencies.

Ajax can assist developers in increasing responsiveness and decreasing load times of coaches. If you have a large coach, or maybe a coach with multiple tabs, it might be advantageous to load some of the page immediately to allow users to start working. In the background, or as the users continue their work, you can use Ajax when the users need it. Avoid loading data that is hidden or data that is not needed at the current time.

Be sure to balance the user's responsiveness needs with interactivity enhancements to create a positive user experience.

WebSphere Education

User interface design approaches

- User-focused design concentrates on the people who use the system and information architecture
  - Frames design in terms of information
  - Centered around each display or page
- Activity-focused design concentrates on human activity
  - Focus shifts to the activity the user must accomplish, not just the individual pages
  - Activity context and transitions between pages are enhanced
- Use activity-focused design when creating coaches

Figure 7-11. User interface design approaches WB868 / ZB8681.0

Notes:

BPM allows you to apply custom functions and style to the standard user interface (coach). A key consideration for customization is how well the design of the interface is applied so as not to interfere with the task.

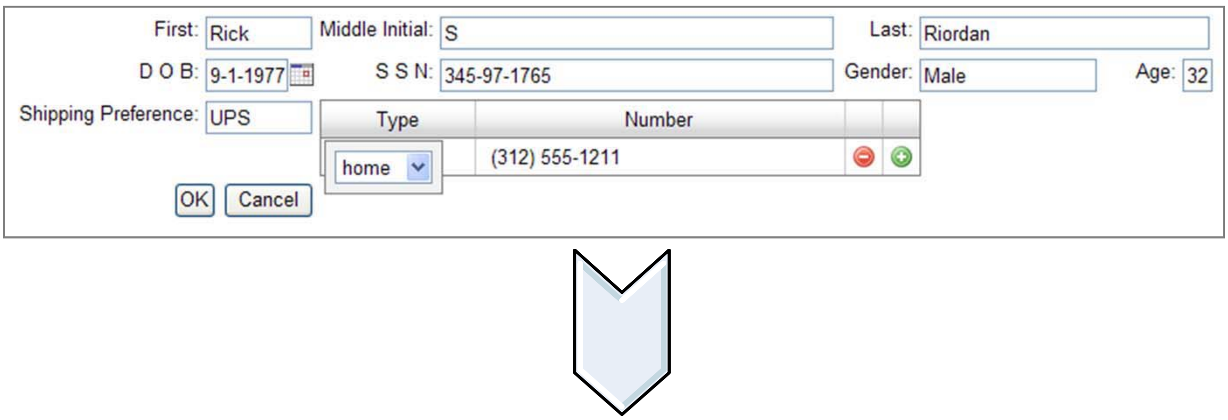
The best design consideration for a coach is that of an activity-focused design. User interface customization must stay within the parameter of the activity requirements and not detract from it. With a balance of good design and function to the customizations that developers and business users seek, comprehensive coach refinement can overcome the limitations in the standard coach controls. The development effort must center around an activity-focused user interface.

Activity-focused design focuses on human activity, as opposed to user-centered design, which frames design in terms of the specific people who use it. It also differs from information architecture, which frames design in terms of information. When faced with the choice between task, user, or information, choose the task as the central focus of the coach design.

WebSphere Education

Design for application maintainability

- Any custom user interface effort must also consider application maintainability
- Do not sacrifice maintainability for the added functionality that is brought about with customization



MAINTENANCE

Figure 7-12. Design for application maintainability WB868 / ZB8681.0

Notes:

Any custom user interface design effort must be created with application maintainability in mind. This consideration must be at the forefront of the design so that developers do not sacrifice this important factor for the added features that are brought about with customization. For example, it can be enticing to use custom HTML blocks in the Coach Designer for almost everything, but then the coach is not easily editable or readable by other developers. The power and ease of use of the Coach Designer are thus limited.

WebSphere Education IBM

Browser considerations when designing coaches

- Rendering speed and efficiency vary significantly by browser vendor and release
- Consider the browser that is used in production during development of a coach
  - The more complicated the coach user interface, the longer it takes to render on older browsers
- Specific issues include:
  - Large Document Object Model (DOM) trees
  - Large JavaScript scripts
  - Deeply nested Coach Views
- Use browser monitor tools to check the resource loading actions and time consumption
  - You can use Fire Bug or HTTP Watch

Figure 7-13. Browser considerations when designing coaches WB868 / ZB8681.0

Notes:

The Document Object Model (DOM) is an application programming interface (API) for valid HTML and well-formed XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated. In the DOM specification, the term “document” is used in the broad sense. Increasingly, XML is being used as a way of representing many different kinds of information that might be stored in diverse systems. Much of this XML would traditionally be seen as data rather than as documents. Nevertheless, XML presents this data as documents, and the DOM might be used to manage this data.

With the Document Object Model, programmers can build documents, navigate their structure, and add, modify, or delete elements and content. Anything that is found in an HTML or XML document can be accessed, changed, deleted, or added by using the Document Object Model, with a few exceptions. In particular, the DOM interfaces for the XML internal and external subsets are not yet specified.

WebSphere Education

Adopt a layered Coach View architecture

- Coach Views are the building blocks of human services as well as other Coach Views
- What is the best way to design a Coach View hierarchy to achieve the most efficient reuse and enable rapid agile changes?
  - Create a hierarchy of different Coach Views that are based on reuse
  - The layered Coach View architecture is driven by skill level that is required to build a Coach View
  - The layered Coach View architecture promotes agility and reuse
  - Enables frequent and rapid UI changes with minimum loss of productivity
- Layered Coach Views include
  - Atomic: A fundamental Coach View
  - Data: Data entry
  - Template: Contains elements that are not changeable
  - Composite: Derived from template and data Coach Views; requires customization

Figure 7-14. Adopt a layered Coach View architecture WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Table and Tab controls

- Table controls are powerful and are the appropriate control for many business scenarios
  - Can be expensive to render on some browsers
- Verify that the total number of cells is no more than necessary when using table controls
- Tab controls are a good way to structure information within a coach
  - It allows a page to be displayed where the user can select from a set of Coach Views to be shown at one time
  - A coach with several tabs can take a significant amount of time to render
  - If several tabs are required, do not load the tab until the user selects the tab, which delays the rendering until the user is ready to interact with it

Figure 7-15. Table and Tab controls WB868 / ZB8681.0

Notes:

The table control provides a mechanism to show a list of business objects in a row-column format. When the table control is added to the canvas, you can then insert further controls as child processes. Each of these child processes is used to visualize a particular column of data. The control is repeated in the same column for each row that is displayed.

WebSphere Education

Bound variables

- Coach Views do not perform well when Human Services have too many or too large variables that are bound to Coach Views

- You can have as many variables in a Human Service as you like
  - Keep the number and size of variables you bind to Coach Views to a minimum
  - Bound variables should contain only variables that are used in Coach Views
  - Use reusable service or server script for mapping

Figure 7-16. Bound variables WB868 / ZB8681.0

Notes:

WebSphere Education

Coach View boundary events

- A boundary event creates a flow from the coach on the coach service
  - Client-side browser events that you might want to propagate back to the server
- Buttons that require the coach to move to the next state must trigger a boundary event
  - When a user clicks the button, it broadcasts the event
- Boundary events enable Ajax elements of a Coach View to access server-side services
  - Boundary events are recognized in the Human Services editor
  - At authoring time, you can add wires to boundary events to execute the client request
- Key uses of boundary events
  - Navigate to the next step in Human Service flow
  - Perform server-side validation
  - Get more or new data from the server

Figure 7-17. Coach View boundary events WB868 / ZB8681.0

Notes:

A Coach View must have the check box next to “Can Fire a Boundary Event” selected on the Overview tab if it wants to broadcast the event and create a flow from the coach. This flow is created from the boundary event. These boundary events also enable Ajax on a coach. There are specific rules about Coach Views:

- Only one boundary event can be declared per view
- You must use a JavaScript API to wire a Coach View to the next state by using: `this.context.trigger(callback);`

When you have multiple flows, the human service editor contains a wiring dialog to help find the correct boundary event on a form.

WebSphere Education

Coach View boundary event limitations

- Coach View boundary events react only to the bound business object
  - Can respond to data updates on only one business object
  - Can write to and can read from a single business object
  - Coach View onChange event relates only to the “bound” variable
- Wrap all controls that must communicate in a composite Coach View
  - This way, the view catches the events and manages visibility, control changes, and focus changes on the browser
- Minimize the number of boundary events
  - A network request is made to the server, and the current state of the coach is persisted
  - Both of these operations are expensive
  - Excessive use might cause performance issues if many concurrent users are interacting with a coach that is frequently generating boundary events

Figure 7-18. Coach View boundary event limitations WB868 / ZB8681.0

Notes:

There are also some limitations to boundary events. When you bind a variable to a Coach View, any change to the business object causes the Coach View to react to those changes, but you can bind only one object to each Coach View. You can read and write from only a single business object, and the changes to the business object trigger the onChange event on the Coach View.

If you need a Coach View to react to multiple variables, create multiple Coach Views that are bound to the individual variables and then wrap all of the Coach Views in a composite Coach View.

WebSphere Education

Minimize the size of business objects

- Business objects that are bound to Coach Views are persisted when boundary events occur
  - A costly operation, especially for large business objects
- If the business object used in the process flow is large and complex:
  - Create a separate business object that contains only fields that are relevant to the coach user interface
  - Bind that business object to the Coach View
- After the coach actions complete, merge the contents of the coach-bound business object into the process flow business object
  - More efficient than persisting large business objects in coaches when boundary events occur

Figure 7-19. Minimize the size of business objects WB868 / ZB8681.0

Notes:

WebSphere Education

Minimize network requests to the server

- For optimal performance, you must make as few network requests as possible to the Process Server
- The following techniques can be used to minimize network requests:
  - Combine multiple JavaScript files into one (or a few)
  - Combine multiple Cascading Style Sheets (CSS) files into one
  - Use CSS image sprites to reduce the number of requests to retrieve image files
  - Design coarse-grained API calls that minimize the number of API calls that flow over the network to the Process Server

Figure 7-20. Minimize network requests to the server WB868 / ZB8681.0

Notes:

For optimal performance, you must make as few network requests as possible to the IBM Process Server. This task is true if the production runtime environment has high network latency between the coach clients and Process Server, or the Process Server and its database. This rule is also important if the production runtime environment uses a relatively low-bandwidth network.

The number of network requests can be determined by using Firebug, if Firefox is the browser, or a similar tool for other browsers.

WebSphere Education

Delay time for auto-complete fields

- Auto completion uses an auto completion service that is called after a certain delay to search for auto completion options
  - These searches are performed against the server
- The delay option specifies the time to wait after the user finishes typing in the field until the request is sent to the server
- A short delay allows for several requests to be sent before the user finishes typing
  - Causes unnecessary load against the server and delays the user response time in the coach
- Set a delay value that ensures that the user finished typing
  - Several hundred milliseconds is a good compromise between responsiveness and reducing server load

Figure 7-21. Delay time for auto-complete fields WB868 / ZB8681.0

Notes:

WebSphere Education

More development tips

- For optimal performance, execute long-running calls asynchronously so that the control can be returned to the user as quickly as possible
- Maximize browser concurrency
  - Load CSS files before JavaScript files, and load CSS and JavaScript files from HTML
- Load CSS files before JavaScript files
  - If while processing a JavaScript file such a dependency is found and the CSS file is not yet loaded, processing is stopped for all JavaScript files on all browser channels
  - Can produce higher and more variable coach load times
- Load CSS and JavaScript files from HTML
  - Browsers have visibility only to files that are loaded directly from the parent HTML file
  - Avoid loading JavaScript files from other JavaScript files as it results in sequentially loading each JavaScript file on the same browser channel

Figure 7-22. More development tips WB868 / ZB8681.0

Notes:

When making service calls that are expected to take a long time to complete, such as calls to back-end systems, perform the calls asynchronously so control can be returned to the user as quickly as possible.

WebSphere Education IBM

Guidelines for using CSS in custom Coach Views

- Coach Views that are supplied with IBM have a certain look to them
  - Standard CSS defines the look, which means that the look can be changed by changing the applied CSS
- Make all styles extensible
  - Show the look and feel of your Coach Views through CSS classes
  - Users can easily change the look and feel of the Coach View without having to change the source code
- When creating a new Coach View, it is a good practice to match the look and feel of the Stock Coach
  - Emulate an HTML-based Stock Coach (Button Coach) or Dojo Dijit-based (Text Coach) View

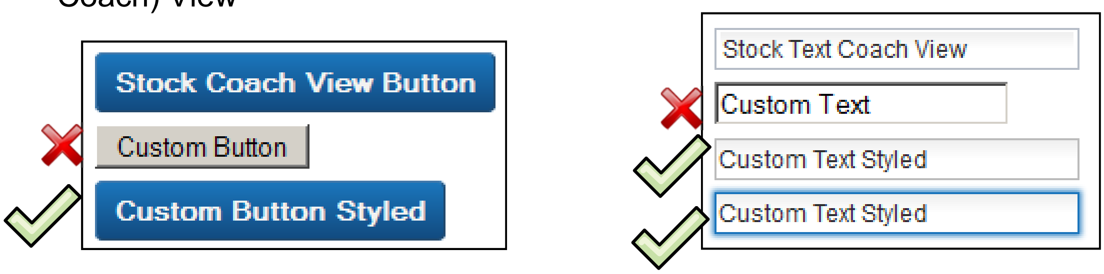


Figure 7-23. Guidelines for using CSS in custom Coach Views WB868 / ZB8681.0

Notes:

The Coach Views that are supplied with IBM have a certain look to them. Since these Coach Views eventually resolve to HTML sent to the browser, you find that standard Cascading Style Sheets (CSS) defines the look. Since CSS is used, it also means that the look can be changed by changing the CSS applied.

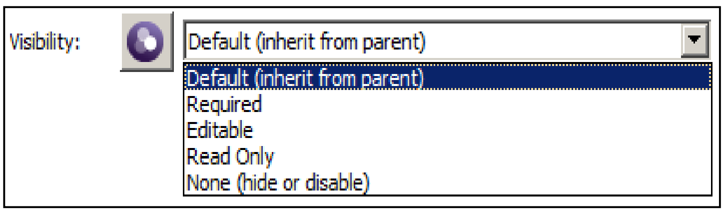
The graphic on the left emulates an HTML-based Stock Coach button. The first one is a Stock Coach View button. The second one is a Custom Button Coach View without any CSS styling applied. The third one is a Custom Button Styled Coach View with CSS classes from the `coach_ng_control.css` file that is applied so that the custom Coach View mimics the appearance and behavior of the Stock Coach Views.

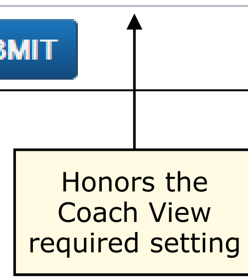
The graphic on the right emulates a Dojo dijit-based text coach. The first one is a Stock Text Coach View. The second one is a Custom Text Coach View without any CSS styling applied. The third one is a Custom Text Styled Coach View with CSS classes from the `coach_ng_control.css` file that is applied so that the custom Coach View mimics the appearance and behavior of the Stock Coach Views. The fourth one is also a Custom Text Styled Coach View with CSS classes applied, which shows that the selected appearance and behavior mimics the Stock Text Coach View.

WebSphere Education IBM

Coach View visibility

- When you create a Coach View, the visibility property automatically is made available
  - Each Coach View has a set of standard properties, which are general, visibility, and HTML attributes
  - Your Coach View must honor these settings





- Use custom visibility sparingly
- If you do use Coach View visibility, plan for it
  - Handle it in `view()`
  - Delete to `view()` to handle `change()`
  - Borrow the code from a Stock Coach View

Figure 7-24. Coach View visibility WB868 / ZB8681.0

Notes:

When you select a Coach View in a layout, the Properties area displays the General, Visibility, HTML Attributes, and Configuration pages. These pages contain the properties of that Coach View instance. The General, Visibility, and HTML Attributes pages display properties that all Coach View instances have. The Configuration page has properties that are defined in the Coach View definition, and they are specific to Coach Views that are bound to that definition. Stock controls have examples of configuration properties.

The visibility of items in coaches is more flexible in V8.5. It now allows rules and scripts to determine visibility at run time. The visibility properties are now on a separate tab in the Properties window to accommodate the new styles for specifying visibility.

WebSphere Education

Providing Coach View information

- When using a custom Coach View, supply basic information to make it easier to find and use
  - Tags: Used for categorizing the Coach View on the palette
  - Coach Editor: Provide meaning icons for the Coach Editor palette icon and a layout image
- Provide Coach View configuration options
  - Define a configuration data type
  - Make configuration options more usable
  - Promote your configuration options
- You cannot directly edit the definition of the Coach View from within the parent coach or Coach View
  - You must open the Coach View definition first before you can change it
  - Make the needed changes on the Overview page

Figure 7-25. Providing Coach View information WB868 / ZB8681.0

Notes:

The **Overview** page displays the Coach View name, information about the Coach View, the images that are used to represent the Coach View during design time, and how the Coach View is used. You can also tag your Coach View to make it easier to find in the library and on the palette.

- Defining a configuration data type: Configuration options can be a set of specific values. For example, your custom Coach View's configuration option might require a currency symbol from a predefined set of supported currency symbols. In such cases, do not use the built-in types, such as String. Instead, create a Business Object that is called CurrencyType, and for Definition Type use Simple Type.
- Making configuration options more usable: To make configuration options more usable, complete the multiple steps. In the Label field, provide a meaningful display name for the configuration property. Use the Documentation field to provide hover help text to help users understand how to use the settings. Provide a Group Name to group several related configuration options.
- Promoting your configuration options: You can promote your custom Coach View's configuration options. A promoted configuration option appears in higher-level Coach Views; this option vastly simplifies and enables Coach View reuse.

WebSphere Education IBM

Provide label support (1 of 2)

- If your Custom Coach View has a label, you must add support for it to ensure that your Coach View behaves as other Stock Coach Views do
  - Added for runtime support and at authoring time in the Coach Editor
- Runtime label support
  - When a custom Coach View is included in the Table or Tab Coach View, the label should appear only once
  - Enable label hiding in the container type of Coach Views by adding code to your custom Coach View

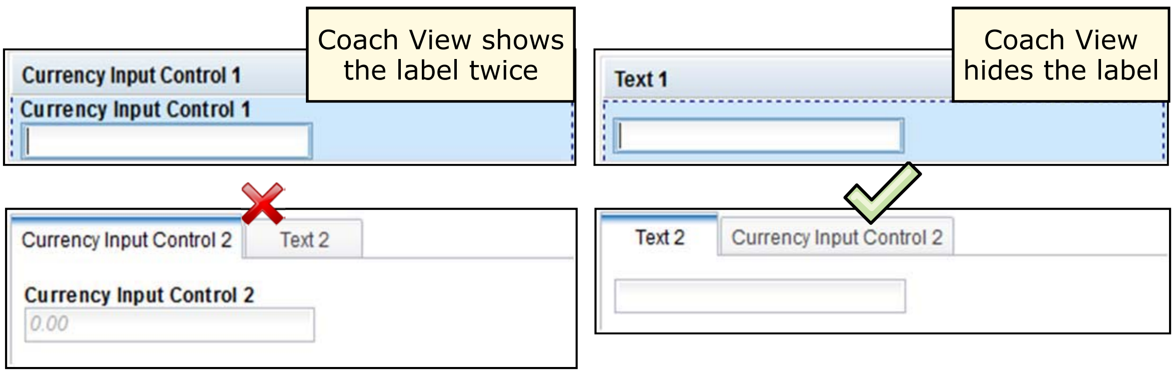


Figure 7-26. Provide label support (1 of 2) WB868 / ZB8681.0

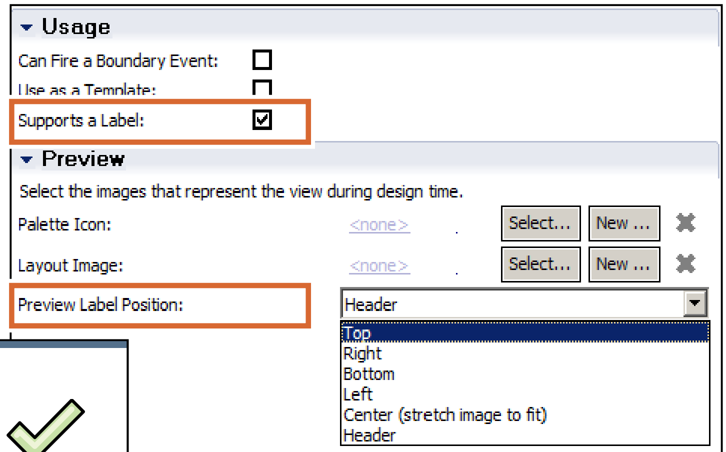
Notes:

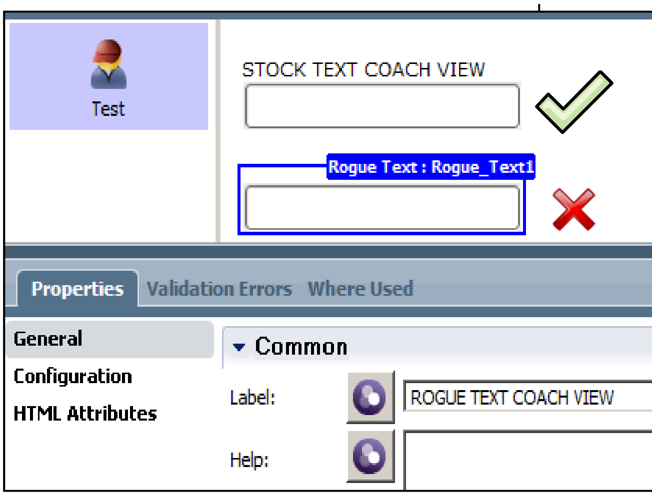
Many IBM-supplied controls have a label that is associated with them. This label can be shown above the new Coach View.

WebSphere Education

Provide label support (2 of 2)

- Authoring label support
  - If your Custom Coach View has a label, the label does not appear in the Coach Editor or Coach View Editor by default





  - For the label to be visible at authoring time, you must select the **Supports a Label** check box in Coach View Editor
  - Also, specify the default label position by using the **Preview Label Position** drop-down menu

Figure 7-27. Provide label support (2 of 2) WB868 / ZB8681.0

Notes:

The **Preview Label Position** setting specifies only the authoring time preview support. If you have a configuration option to adjust the label position, the design time authoring does not reflect the setting; it should show the default. If you do not supply a configuration option for your label position and you have a hardcoded runtime label position, make sure that the preview label support matches the runtime support.

WebSphere Education

Runtime considerations for the label placement

- In your new Coach View, make sure that you implement the label style that is shown in the Coach Editor
- The Coach Editor label (authoring) and runtime label should be the same
  - Example: Header style in Horizontal Section Stock Coach View at authoring time is the same as at run time

Figure 7-28. Runtime considerations for the label placement WB868 / ZB8681.0

Notes:

Many IBM-supplied controls have a label that is associated with them.

WebSphere Education

Know the programming model

- The Coach Views programming model is based on standard web technologies and Dojo
- To be a proficient designer of Atomic Coach Views, you must be proficient with these web technologies
- Specifically
  - CSS and HTML
  - JavaScript Object Notation (JSON): `http://www.json.org/`
  - JavaScript: `http://www.w3schools.com`
  - Dojo: `http://dojotoolkit.org/` and `http://dojotoolkit.org/reference-guide/1.7/`
- You must understand how the Coach View editor maps to key elements of the standard web programming models

Figure 7-29. Know the programming model WB868 / ZB8681.0

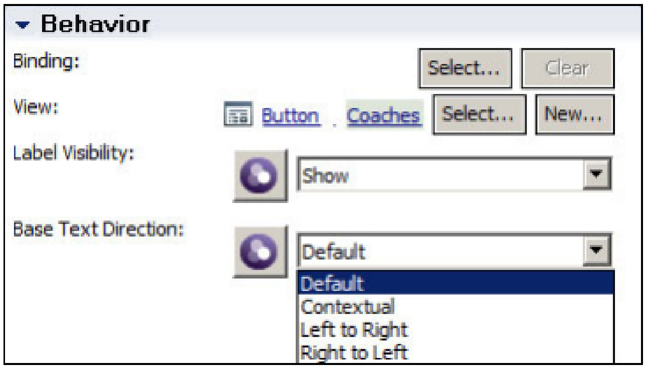
Notes:

Experienced web developers, who already know the web programming model and are engaged in development of advanced Atomic Coach Views, need to understand how the various Coach View editor elements correspond to the generated HTML.

WebSphere Education

Base Text Direction settings

- BPM supports languages that are written from right to left and languages that are written from left to right
- Enable Base Text Direction in Process Designer and the Behavior window



- To enable Base Text Support in custom Coach Views, you must retrieve the value of the Base Text Direction attribute and write code

Figure 7-30. Base Text Direction settings WB868 / ZB8681.0

Notes:

The Base Text Direction property is available only if you enable the Base Text Direction preference. The default value is Default, which means the Coach View uses the text direction that is set in the profile of the user. When you set the text direction as Left to Right or Right to Left, the text is locked to the specified direction. When you set the text direction as Contextual, the first strong directional character in a string determines its text direction. This rule applies to all text elements that a Coach View displays. For example, a Text stock control has an Arabic label, but its field contents are English. In this case, the text in the label is from right to left, although the text in the field is from left to right.

WebSphere Education IBM.

Collaboration and validation

- Process Portal allows you to request help from experts and collaborate with experts and other users in real time to complete work on a task
  - The collaboration feature provides feedback behavior when multiple people work on the same coach at the same time
- In most cases, Coach Views can rely on the Coach Framework to handle Coach Collaboration in IBM Process Portal
  - There are a few exceptions, and they are usually associated with the container type of a custom Coach View
- Enable collaboration through the collaboration handler or the collaboration API
- If your custom Coach View receives user input, you must take special steps to enable server-side validation
  - To ensure that your custom Coach View is consistent in the way it displays validation errors, use the Stock Coach Views for examples of presentation logic

Figure 7-31. Collaboration and validation WB868 / ZB8681.0

Notes:

IBM BPM Coach Framework provides a server-side validation capability that you can use to validate the data that is in the coach before the flow proceeds to the next step in the service flow. To validate the data in a coach before the flow proceeds to the next step in the service flow, you can add a validation node.

Runtime performance considerations

- Avoid firing unnecessary change events
  - Always check the value of configuration or binding before setting a new value
- Avoid putting too many tabs with many Coach Views in a single coach
  - It takes too much time to render all the content
- Avoid showing many rows and columns when you use the Table Coach View
  - The pagination support is only client-side
- Avoid the use of too many Ajax calls during the initial load to populate data
  - Browsers have a limited number of concurrent connections available, and their logs vary
- Reduce the Coach View memory footprint
  - Enable the prototype-level event handler

Figure 7-32. Runtime performance considerations WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Developing and coding tips for custom Coach Views

- Always test your Coach, Human Services
  - While developing Coach Views, it can be useful to use debuggers to see what is going on
- Use `bind()` and `bindAll()` APIs correctly
  - Avoid the use of event handler `change()` as the callback; instead, use a different function as a callback handler
  - Do no use lifecycle event `change()` as the callback handler
  - Add rebind logic when the base object is changed
- Implement a Loading Curtain as a Coach View to avoid flickering
  - Depending on the size and the complexity of a web page, JavaScript and CSS might take a long time to complete initialization
  - During this initialization, a web page is not ready for user interactions and might flicker as the page elements are loaded and initialized

Figure 7-33. Developing and coding tips for custom Coach Views WB868 / ZB8681.0

Notes:

Most of the browsers available today have debugging tools that are built into them, including Firefox, IE, and Chrome. Using these debuggers, you can set breakpoints and examine the content of the DOM tree and JavaScript.

When debugging a Coach View, you might find that the code shown in the debugger appears compressed and unreadable. The IBM-supplied code, including Dojo, is what is called “minified.” What this means is that the interpreted JavaScript that comprises the IBM code and Dojo code is compressed by shrinking variable names and white space (including line breaks). The reason for compressing is that it becomes smaller in size, and hence there is less data to be moved over the network when the libraries are being used in the browser. This technique is a standard technique. However, when debugging your own code, it can be confusing. Fortunately, IBM provides a solution. A debugging flag can be enabled which causes a different version of the JavaScript libraries to be used. That alternative version is not minified and hence does not show as compressed in the debuggers.

On the Resource Environment Provider panel, select **Mashups_ConfigService** and set a Custom Property of **isDebug**. Change the value from false to true (or true to false to disable again) and click **OK**. Your debug changes will not start happening until after a server restart. As soon as the server is restarted, code now shows up as not minified. Notice that it might have an impact on overall performance, but this effect is a small price to play during Coach View debugging. If you are using a shared environment with other developers, be sensitive to their needs as much as your own.

7.2. Deployment considerations for coaches

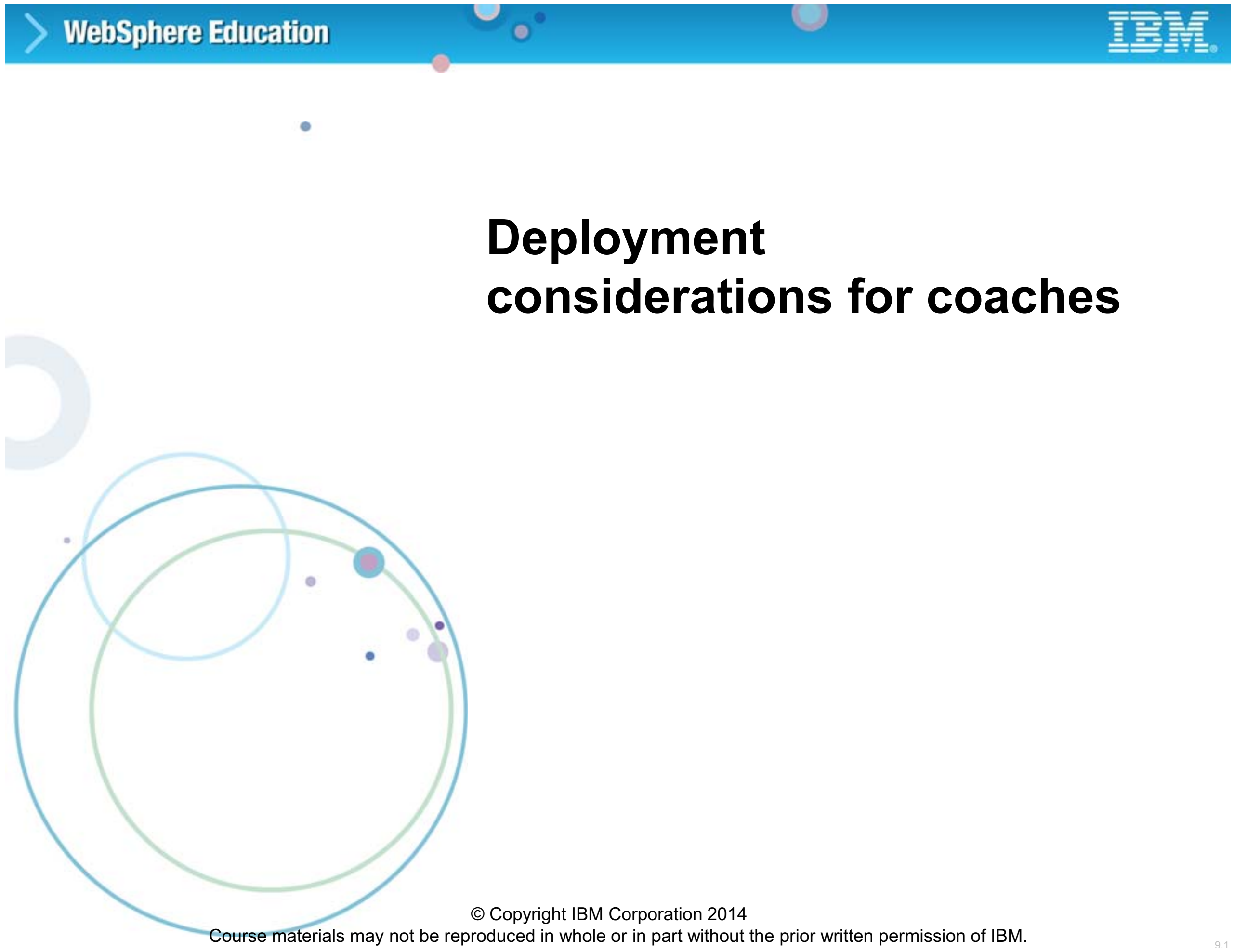


Figure 7-34. Deployment considerations for coaches WB868 / ZB8681.0

Notes:

WebSphere Education

Browser considerations

- Use a high performing browser
  - Use the most recent releases of the web browsers such as Chrome, Firefox, Internet Explorer, and Safari to load your coaches
  - Performance can vary depending on which browser is used
- BPM allows the browser to cache static content such as style sheets and JavaScript files for 24 hours
  - The browser might be configured to not cache static files
  - Results in the same resources that are being loaded repeatedly causing long page load times
- Enable browser caching
  - Coaches rely heavily on Ajax and JavaScript code; their performance depends heavily on the efficiency of the browser in processing this code
  - Examine and adjust both HTTP server and client browser settings

Figure 7-35. Browser considerations WB868 / ZB8681.0

Notes:

For optimal performance, use the most recent releases of the web browsers such as Chrome, Firefox, Internet Explorer, and Safari to load your coaches.

Ensuring the browser cache is enabled in Internet Explorer:

To enable caching, use the following steps:

1. Click **Tools > Internet Options > General > Browsing History**.
2. Set the cache size, indicated by Disk space to use, to at least 50 MB.

To disable the browser cache clearing on logout, use the following steps:

1. Click **Tools > Internet Options**.
2. Under the General tab, make sure that the *Delete browser history on exit* check box is cleared.
3. Under the Advanced tab, ensure that the *Empty Temporary Internet Files folder when browser is closed* check box is cleared. Click **OK** to save the settings.
4. Restart your browser to ensure that the changes take effect.

Ensuring the browser cache is enabled in Firefox:

To enable caching, use the following steps:

1. Click **Tools > Options > Advanced > Network > Offline Storage**.
2. Set the *Offline storage* to at least 50 MB.

To ensure that caching is enabled, use the following steps:

1. Click **Tools > Options**.
2. Click the **Privacy** tab and make sure that it reads *Firefox will: Remember History*. Click **OK**. This setting enables caching. If you select, *Firefox will: Never remember history*, caching is disabled and you must change the setting.

   If you select *Use custom settings for history*, you have more options that still allow caching:

   - If you select *Automatically start Firefox in a private browsing session*, caching is enabled. However, after the browser is closed, everything is erased (not only cache, but also browsing history) as though you were never using the browser.
   - If private browsing is not selected, make sure the *Clear history when Firefox closes* check box is cleared.
3. Restart the browser to apply all changes.

WebSphere Education

Optimize network quality

- Verify that you have an optimal network connection between browser clients and servers
  - Multiple network requests are made between the coach in the browser and the Process Server
  - This task is crucial to performance
- Examine the physical distance of components to minimize network latency
  - Locate BPM servers physically close to one another when possible
  - Check your servers and clients and the servers in a cluster and their databases
- Examine the quality of the network
  - Look at the speed and the bandwidth of the connection
  - Verify that the browser client is using the fastest possible network with the highest bandwidth

Figure 7-36. Optimize network quality WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Warm-up frequently used coaches

- Coaches are generated in two stages on the Process Server
  - Stage 1: Does most of the generation of the coach runtime code; only done once during the lifetime of an instance
  - Stage 2: Does the substitution of dynamic data in the coach runtime code that is generated during stage1; done every time that a coach is opened
- Coach generation is done only once during a Process Server runtime instance
  - The first usage of the coach takes much longer than subsequent uses
- Warm up commonly used coaches one time when you start the Process Server
  - Open an instance of the coach in a client browser
  - No need to complete the coach; stage 1 code generation is done when the coach instance is opened
  - Might not be practical for business reasons

Figure 7-37. Warm-up frequently used coaches WB868 / ZB8681.0

Notes:

WebSphere Education

Tune the HTTP server

- Coaches typically are deployed to an environment that includes an HTTP server
  - Coaches send multiple requests to their Process Server, which an HTTP server handles
- Tune the HTTP server to handle the load and achieve good user response time
  - Verify that caching is configured in the HTTP server
  - Change the expiration time of static files in the HTTP server header
- Update the caching and compression settings in the `httpd.conf` file
  - `ExpiresActive on`
  - `ExpiresByType image/gif A86400`
  - `ExpiresByType image/jpeg A86400`
  - `ExpiresByType image/bmp A86400`
  - `ExpiresByType image/png A86400`
  - `ExpiresByType application/x-javascript A86400`
  - `ExpiresByType text/css A86400`

Figure 7-38. Tune the HTTP Server WB868 / ZB8681.0

Notes:

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Explain how coaches can benefit your organization
- Define good practices for developing Coach Views
- Identify the requirements for good response times when users interact with coaches
- Explain modeling capabilities that impact performance
- Describe deployment considerations for Coach Views

Figure 7-39. Unit summary WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Checkpoint questions

1. True or False: The business data mapping on a Coach View can map only one business object.

2. True or False: Service calls in coaches that are expected to take a long time to complete, such as calls to back-end systems, need to be performed asynchronously.

3. True or False: It is a good practice to enable caching in the web browser that is used for loading the coaches and in the HTTP server that handles the coach requests.

Figure 7-40. Checkpoint questions WB868 / ZB8681.0

Notes:

Write your answers here:

1.

2.

3.

WebSphere Education

Checkpoint answers

1. True

2. True

3. True

Figure 7-41. Checkpoint answers WB868 / ZB8681.0

Notes:

Unit 8. Threading

What this unit is about

This unit provides an introduction to the threading architecture of Business Process Manager and methods on tuning thread pools.

What you should be able to do

After completing this unit, you should be able to:

- Describe the service component architecture (SCA) threading architecture
- Configure thread pools
- Tune thread pools that are based on binding types
- Explain some common thread issues

How you will check your progress

- Checkpoint

References

IBM Business Process Manager V8.5 IBM Knowledge Center

`http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/ditamaps/ic-homepage-bpm.html`

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Describe the service component architecture (SCA) threading architecture
- Configure thread pools
- Tune thread pools that are based on binding types
- Explain some common thread issues

Figure 8-1. Unit objectives WB868 / ZB8681.0

Notes:

Performance is often measured as the number of throughputs (BTPS). To maximize the throughputs, applications must be running in multiple threads to process a number of requests concurrently. If you are dealing with a slow target, it might become necessary to control the data flow by adjusting the thread number in the source. The configuration of the thread pool is specific to its binding types. When you are integrating numerous applications and processes, it is important to understand the threading architecture of Business Process Manager. You must avoid performance limitation that is caused from insufficient thread numbers.

WebSphere Education

Topics

- Service Component Architecture (SCA) threading architecture
- Threading considerations

Figure 8-2. Topics WB868 / ZB8681.0

Notes:

8.1. Service Component Architecture (SCA) threading architecture

This topic discusses how to tune the thread concurrency and the size of each thread pool for each binding type.

WebSphere Education

Service Component Architecture (SCA) threading architecture

Figure 8-3. Service Component Architecture (SCA) threading architecture WB868 / ZB8681.0

Notes:

WebSphere Education

SCA threading architecture overview (1 of 2)

- Thread pools are the provider of threads that SCA modules and components use
  - Subject to the threading configuration that is specified in activation specifications
- Resource adapter configurations specify the thread pool that the associated activation specifications use
- SCA modules and components might run on threads from the following preconfigured thread pools, which depend on their implementation and configuration:
  - Default thread pool: Asynchronous SCA binding
  - SIBJMSRAThreadPool thread pool: JMS binding
  - ORB thread pool: Synchronous SCA inter-Java virtual machine (JVM)
  - Web container thread pool: Web services binding

Figure 8-4. SCA threading architecture overview (1 of 2) WB868 / ZB8681.0

Notes:

WebSphere Education

SCA threading architecture overview (2 of 2)

- SCA modules and components can run on threads from any user-defined thread pool
- New thread pools can be created and resource adapters can be configured to use the newly created thread pool
- Bindings that can be configured to use any thread pool:
  - Asynchronous SCA: Platform Messaging Component SPI Resource Adapter
  - JMS to SIBus destinations: SIB JMS Resource Adapter
  - Web services (WebSphere Application Server transport chain facility)
  - Web services SOAP/JMS to SIBus destinations: SIB JMS Resource Adapter
  - WebSphere MQ: WebSphere MQ Resource Adapter

Figure 8-5. SCA threading architecture overview (2 of 2) WB868 / ZB8681.0

Notes:

WebSphere Education

Thread pools

Select	Name	Description	Minimum Size	Maximum Size
You can administer the following resources:				
☐	Default		20	20
☐	ORB.thread.pool		10	50
☐	SIBFAPInboundThreadPool	Service integration bus FAP inbound channel thread pool	4	50
☐	SIBFAPThreadPool	Service integration bus FAP outbound channel thread pool	4	50
☐	SIBJMSRAThreadPool	Service Integration Bus JMS Resource Adapter thread pool	35	41
☐	TCPChannel.DCS		20	20
☐	WMQJCAResourceAdapter	WebSphere MQ Resource Adapter thread pool	10	50
☐	WebContainer		50	50
☐	server.startup	This pool is used by WebSphere during server startup.	1	3

Figure 8-6. Thread pools WB868 / ZB8681.0

Notes:

WebSphere Education

Synchronous SCA

- Intra-JVM (same JVM)
  - Target SCA components or SCA modules that are run on the thread of the caller
  - The behavior of synchronous SCA intra-JVM is equivalent for both intramodule and intermodule
  - Bindings have performance characteristics similar to direct Java to Java calls
- Cross-JVM
  - Uses RMI and target SCA modules that are run on ORB threads
  - Not applicable to intramodule (component to component) bindings
  - SCA modules are enterprise archives (EARs), so intramodule bindings are forced to be intra-JVM
  - Bindings have performance similar to other asynchronous, queue-based, SCA bindings
- The fastest SCA binding is intra-JVM synchronous SCA
  - Both intramodule and intermodule

Figure 8-7. Synchronous SCA WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Asynchronous SCA (1 of 2)

- SCA module activation specifications should be configured considering all asynchronous SCA bindings that the module uses
  - SCA exports
  - SCA imports (request with callbacks)
  - Internal asynchronous SCA bindings (invocations and callbacks)
- Intramodule
  - Asynchronous SCA is the only asynchronous binding available intramodule
  - Target SCA components run on threads that are created in the context of the associated SCA module's activation specification
  - Callbacks run on threads that are created in the context of the associated SCA module's activation specification
  - The number of threads is limited to the maximum number of threads that the associated module's activation specification (consumer) allocates
  - The number of threads is limited to the maximum number of threads available in the target thread pool that the associated resource adapter (provider) specifies

Figure 8-8. Asynchronous SCA (1 of 2) WB868 / ZB8681.0

Notes:

WebSphere Education

Asynchronous SCA (2 of 2)

- Intermodule
  - Target SCA modules run on threads that SCA exports create in the context of the associated SCA module’s activation specification
  - Callbacks run on threads that SCA imports create in the context of the associated SCA module’s activation specification
  - The number of threads is limited to the maximum number of threads that the associated module’s activation specification (consumer) allocates
  - The number of threads is limited to the maximum number of threads available in the target thread pool that the associated resource adapter (provider) allocates
- The performance characteristics of asynchronous SCA bindings are equivalent for both intra-JVM and inter-JVM bindings

Figure 8-9. Asynchronous SCA (2 of 2) WB868 / ZB8681.0

Notes:

SCA exports and SCA imports do not have activation specifications that are associated with them.

WebSphere Education IBM

Asynchronous SCA: SCA modules and resource adapters

- The maximum size of the thread pool for the platform messaging resource adapter is applied against all SCA module activation specifications
  - Activation specifications for SCA modules are the consumers of threads that the target thread pool provides that the associated resource adapter specifies

- By default, the Platform Messaging Component SPI Resource Adapter uses the user-defined thread pool named **Default**

Figure 8-10. Asynchronous SCA: SCA modules and resource adapters WB868 / ZB8681.0

Notes:

WebSphere Education

Platform Messaging Component SPI Resource Adapter (1 of 2)

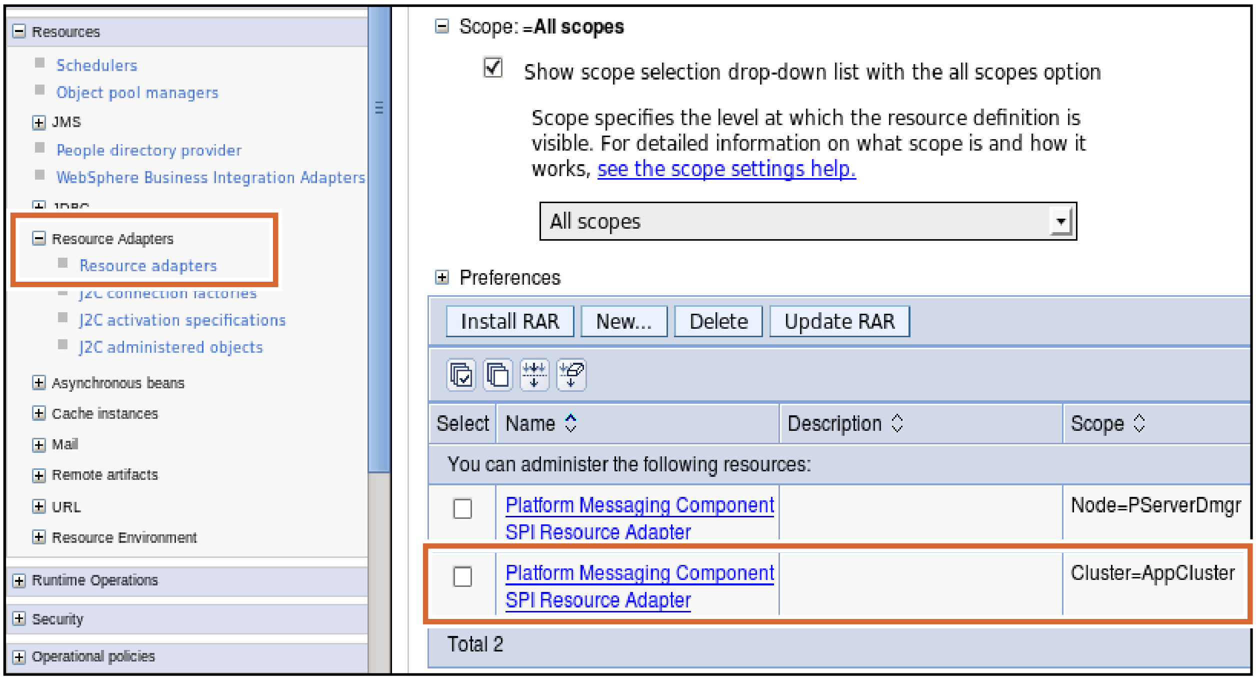


Figure 8-11. Platform Messaging Component SPI Resource Adapter (1 of 2) WB868 / ZB8681.0

Notes:

WebSphere Education

Platform Messaging Component SPI Resource Adapter (2 of 2)

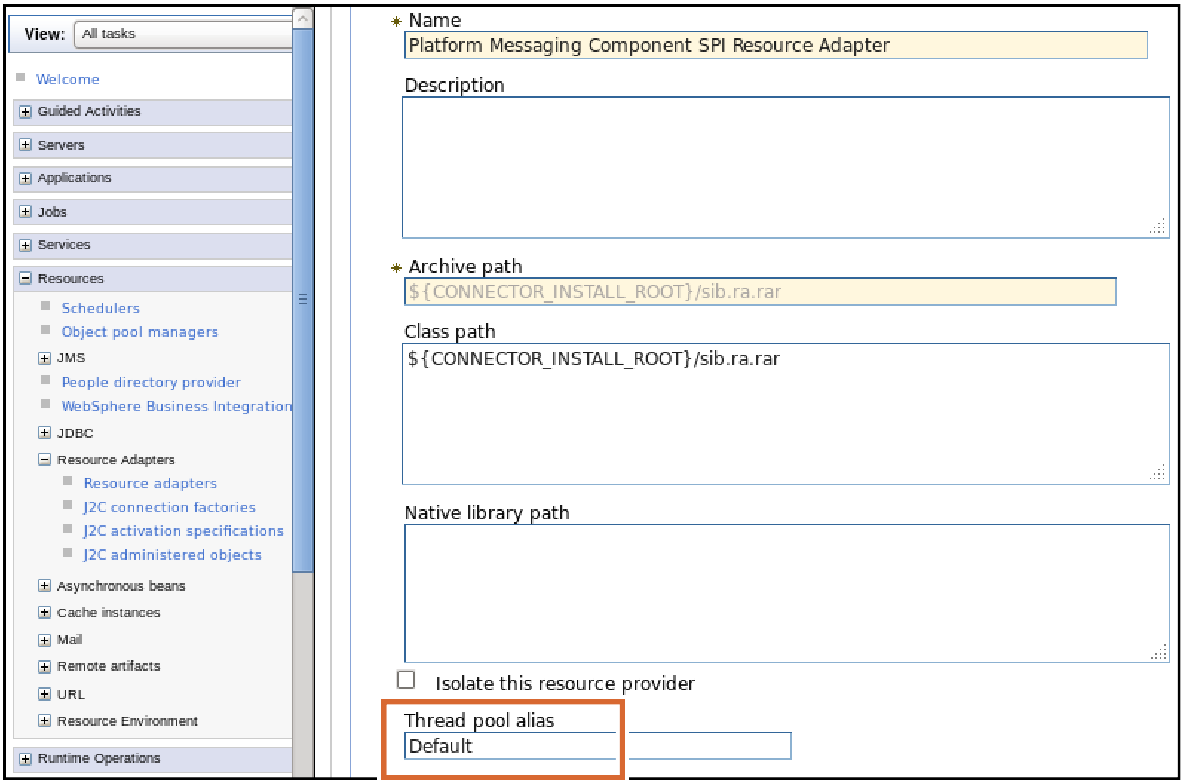


Figure 8-12. Platform Messaging Component SPI Resource Adapter (2 of 2) WB868 / ZB8681.0

Notes:

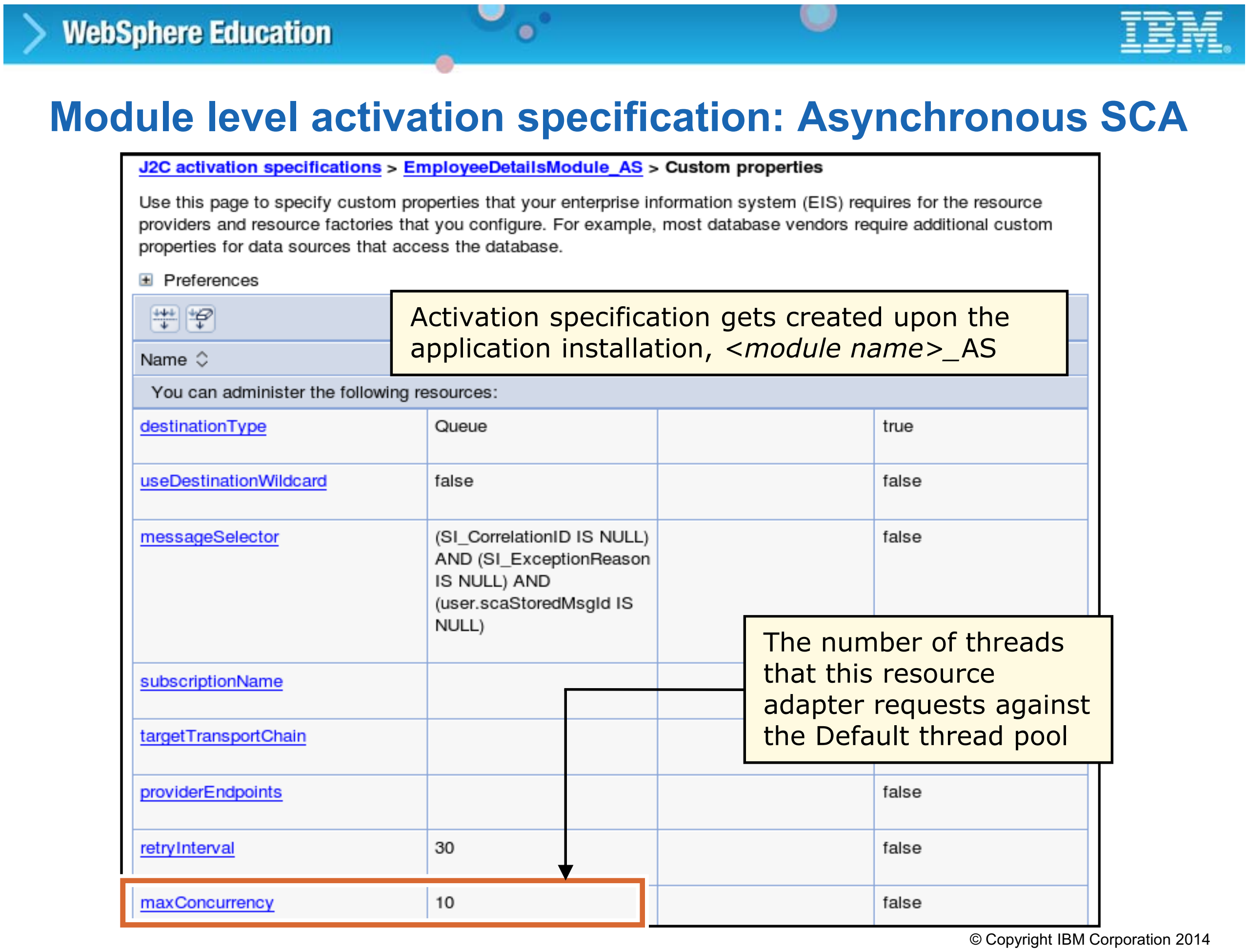


Figure 8-13. Module level activation specification: Asynchronous SCA WB868 / ZB8681.0

Notes:

The *Platform Messaging Component SPI Resource Adapter* is created as part of an application installation (*<module name>_AS*). It provides an EIS with the ability to communicate with message-driven beans configured on the server. Message processing for an application uses properties that are defined under an activation specification for this resource adapter. The `maxConcurrency` custom property under the JCA activation specification for the resource adapter is used to specify the number of threads available to process messages by the application. (In this example, the number of threads that are requested in the `maxConcurrency` property is set to 10.)

WebSphere Education

Default thread pool configuration

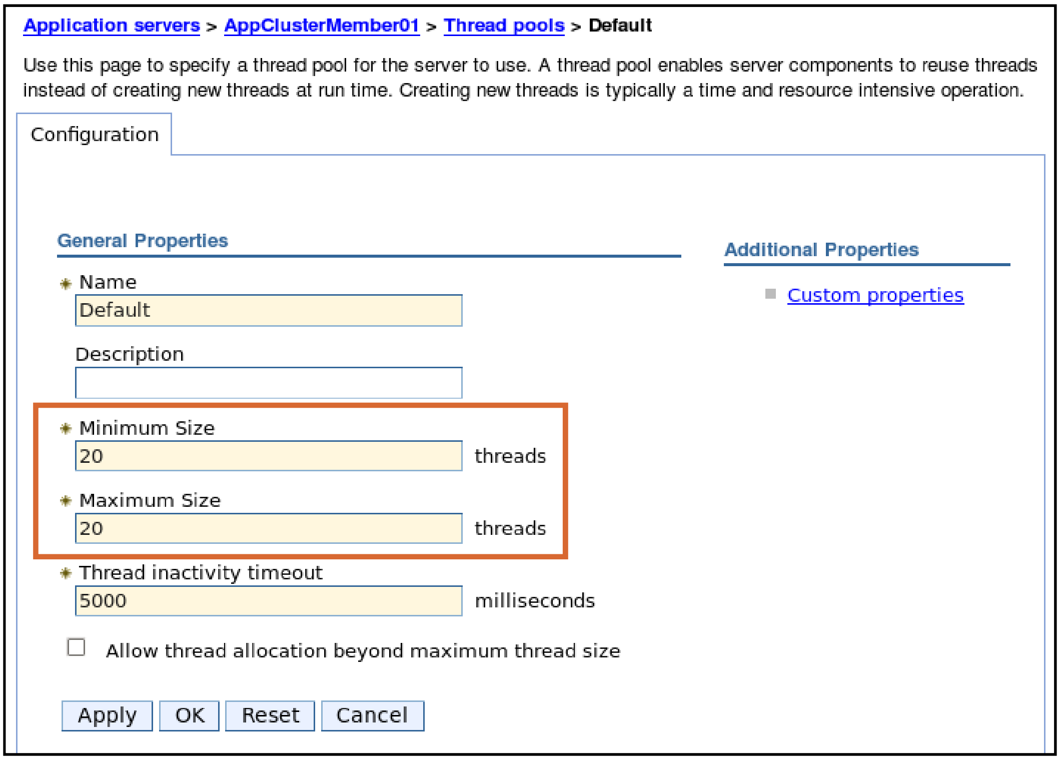


Figure 8-14. Default thread pool configuration WB868 / ZB8681.0

Notes:

WebSphere Education IBM

JMS based on service integration bus (SIBus)

- The default resource adapter for JMS bindings is the SIB JMS Resource Adapter
- Downstream modules that are accessed via JMS exports that are run on SIB JMS Resource Adapter threads that apply to the activation specification of the target JMS export
  - Each individual JMS export has its own activation specification
  - JMS exports create threads when they are accessed asynchronously (SIBus queue-based)
  - Threads that JMS exports create do not count against the parent SCA module's activation specification
- Callbacks are run on SIB JMS Resource Adapter threads that apply to the activation specification of the calling JMS import
  - Each individual JMS import has its own activation specification, and threads that JMS imports create to process incoming callback responses
  - Threads that JMS imports create do not count against the parent SCA module activation specification

Figure 8-15. JMS based on service integration bus (SIBus) WB868 / ZB8681.0

Notes:

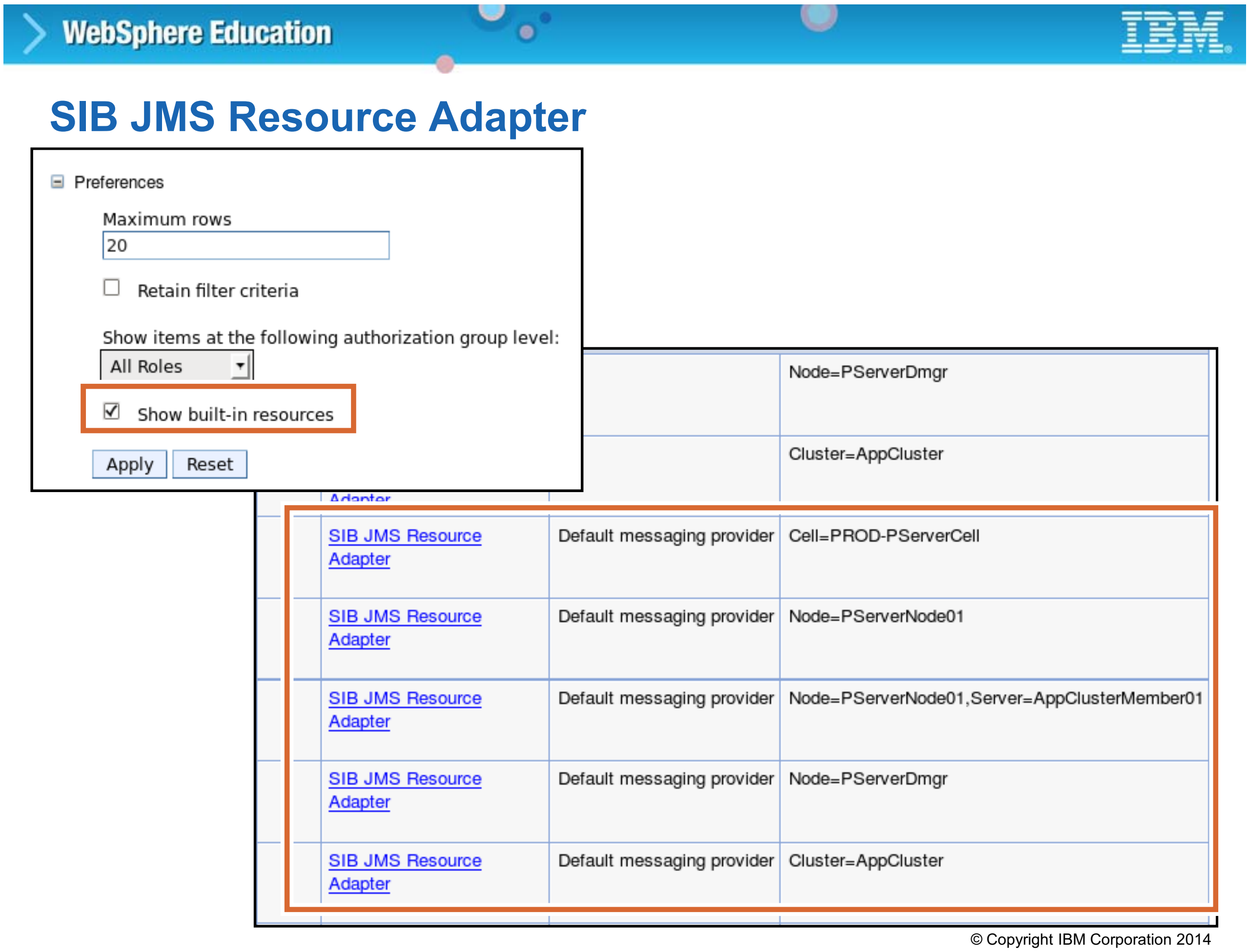


Figure 8-16. SIB JMS Resource Adapter WB868 / ZB8681.0

Notes:

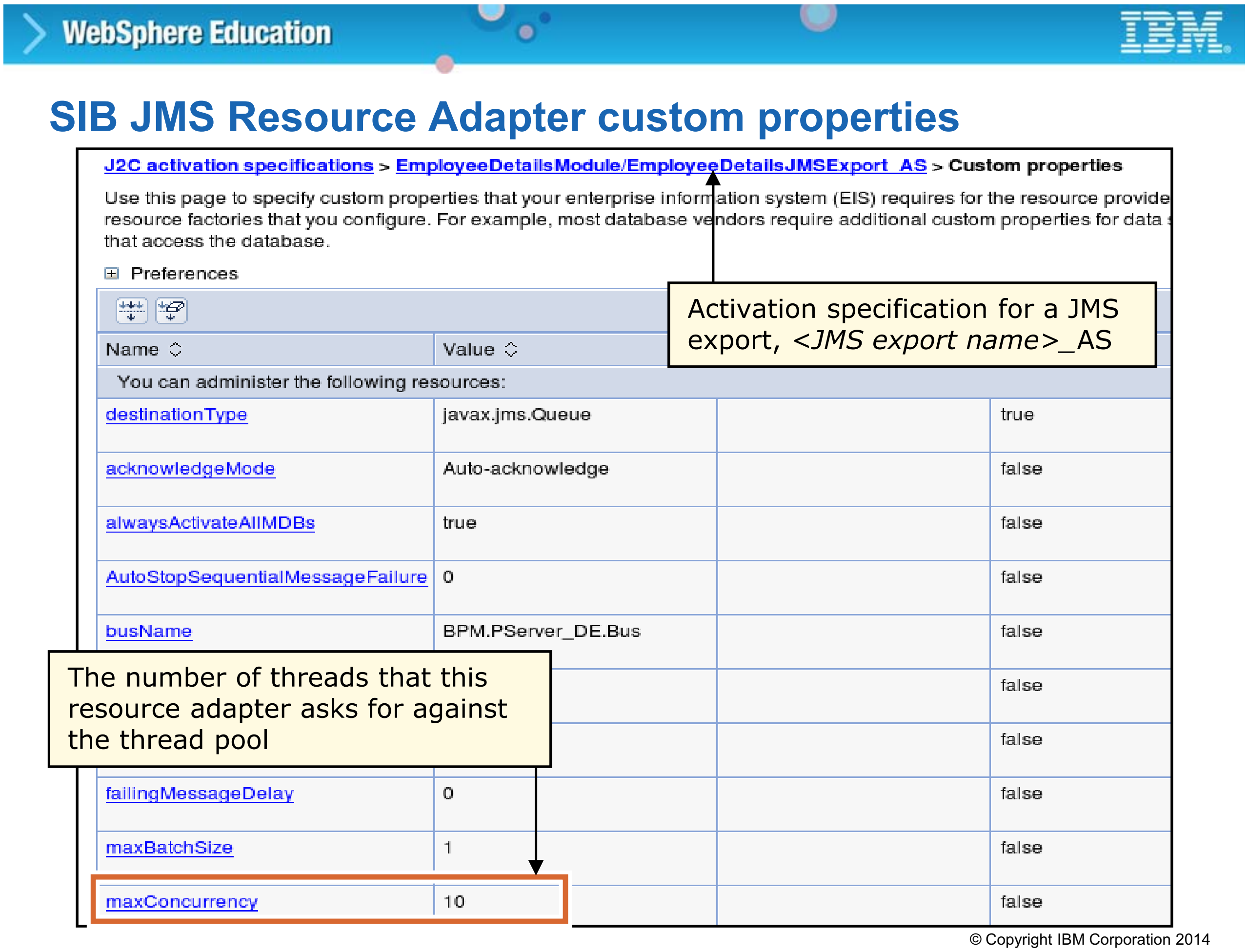


Figure 8-17. SIB JMS Resource Adapter custom properties WB868 / ZB8681.0

Notes:

WebSphere Education

JMS based on SIBus

- The maximum size of the thread pool that is assigned to the SIB JMS Resource Adapter is also applied
- The thread pool that is assigned to the SIB JMS Resource Adapter is shared among all JMS usage
  - JMS exports and imports
  - Asynchronous web services (SOAP/JMS) exports and imports
  - Internal business process engine (BPE) queue that is used for long-running processes
- By default, the SIB JMS Resource Adapter uses the user-defined thread pool named **SIBJMSRAThreadPool**
- The BPE internal queue is assigned to the SIB JMS Resource Adapter in the BPE default operational mode
  - The JMS threading configuration directly affects concurrency for long-running processes

Figure 8-18. JMS based on SIBus WB868 / ZB8681.0

Notes:

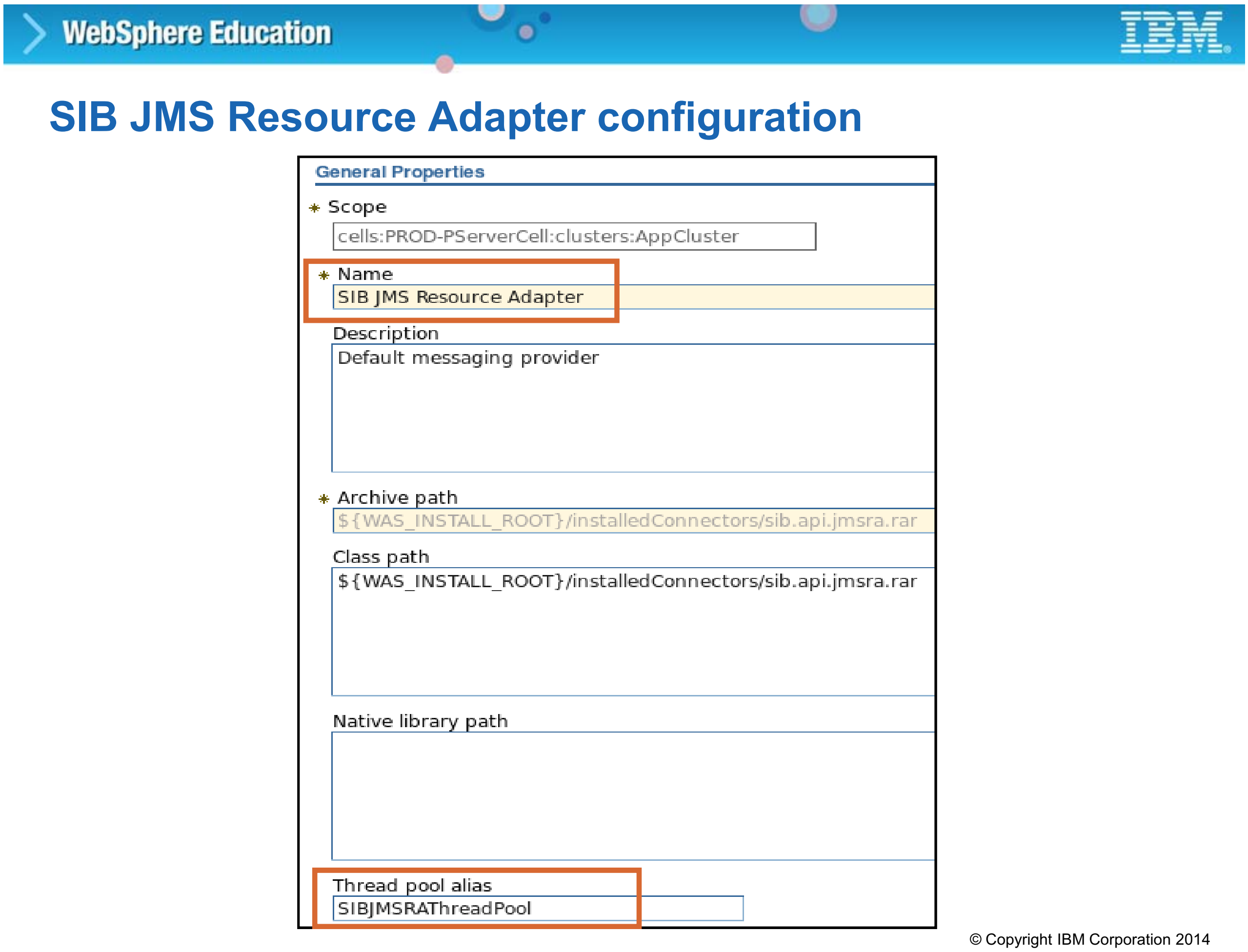


Figure 8-19. SIB JMS Resource Adapter configuration WB868 / ZB8681.0

Notes:

The areas noted on this screen capture show that, under General Properties, the Name field is set to **SIB JMS Resource Adapter** and the Thread pool alias is set to **SIBJMSRAThreadPool**.

WebSphere Education

SIBJMSRAThreadPool thread pool configuration

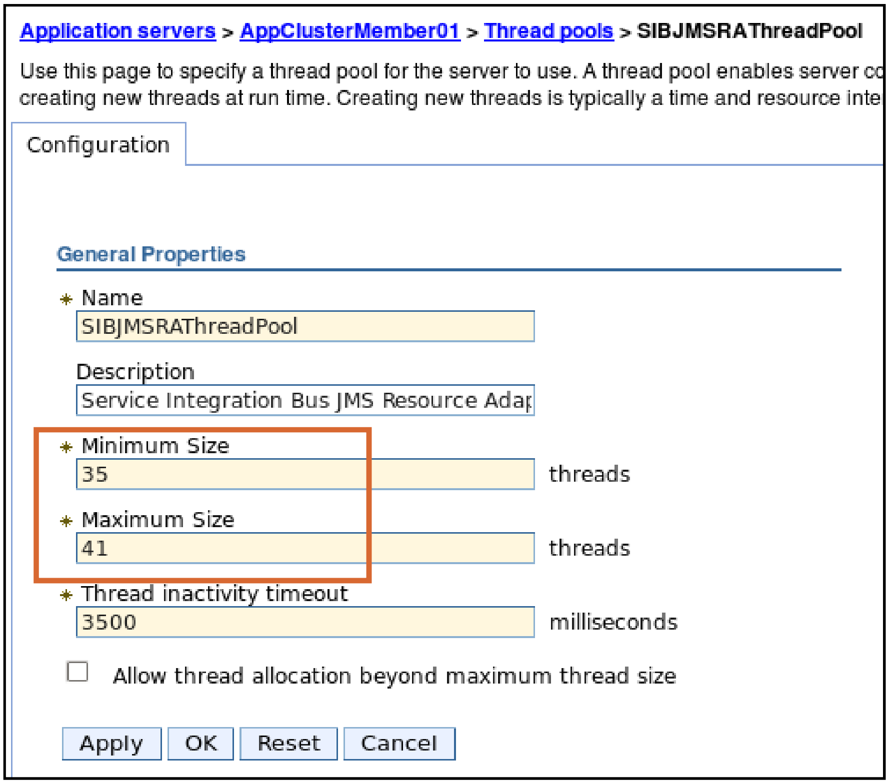


Figure 8-20. SIBJMSRAThreadPool thread pool configuration WB868 / ZB8681.0

Notes:

WebSphere Education IBM

JMS and other messaging providers

- Business Process Manager supports generic JMS
- Generic JMS binding supports independent vendor JMS providers
  - Oracle AQ, TIBCO, SonicMQ, WebMethods, BEA WebLogic, and WebSphere MQ
- Other JMS providers that the **generic Java Message Service** binding supports are tuned by using the previous JMS tuning procedures
- Tuning for messaging providers that the generic JMS binding supports depends on the implementation of each individual messaging provider

Figure 8-21. JMS and other messaging providers WB868 / ZB8681.0

Notes:

WebSphere Education

SOAP/HTTP: Synchronous web services

- Downstream modules that are accessed by using synchronous web services exports that are run on web container threads
  - Threads that the web services exports create
  - Threads that web services exports create do not count against the activation specification of the parent SCA module
  - **Exception**: If the target service is in the same JVM on WebSphere Application Server and if web services optimization is enabled, the target service runs on the thread of the caller
  - SCA support for synchronous web services bindings does not include callback semantics
- The maximum size of the web container thread pool is applied
  - The web container thread pool is a shared resource
  - All synchronous web services; web services exports and regular Integration Designer web services
  - Any HTTP requests including HTML and other servlets or JSPs hosted by the same Business Process Manager instance

Figure 8-22. SOAP/HTTP: Synchronous web services WB868 / ZB8681.0

Notes:

WebSphere Education

WebContainer thread pool configuration

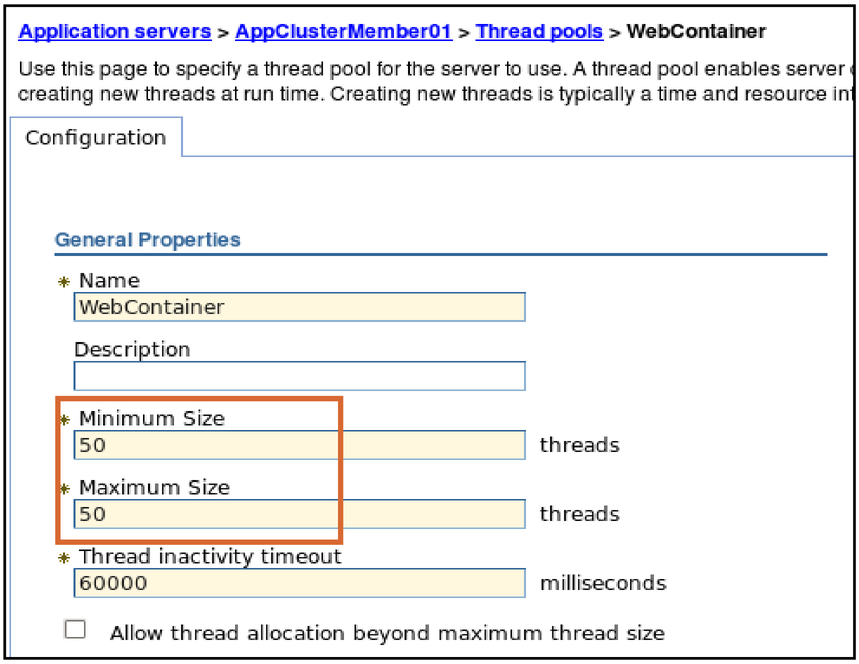


Figure 8-23. WebContainer thread pool configuration WB868 / ZB8681.0

Notes:

WebSphere Education IBM

JMS: Asynchronous web services

- Downstream modules are run on SIB JMS Resource Adapter threads
- SCA support for synchronous web services bindings does not include callback semantics
- The maximum size of the thread pool that is assigned to the SIB JMS Resource Adapter is applied
- The thread pool that is assigned to the SIB JMS Resource Adapter is shared among all JMS usage
  - All JMS exports and imports
  - All asynchronous web services exports and imports
  - Internal BPE queue that is used for long-running processes
- By default, the SIB JMS Resource Adapter uses the user-defined thread pool named SIBJMSRAThreadPool
- The BPE internal queue is assigned to the SIB JMS Resource Adapter
  - The JMS threading configuration directly affects concurrency for long-running processes

Figure 8-24. JMS: Asynchronous web services WB868 / ZB8681.0

Notes:

WebSphere Education

Business processes

- Microflows
  – Generally all activities run on the thread of the caller
- Long-running business processes
  – Processes move from activity to activity when a message is received on an internal BPE queue
  – The next activity runs on the thread that is created to work on the internal queue message
  – If internal queue threads are not available, no long-running business process can progress to its next activity
  – Response processing two ways invokes run on the “callback” thread of the binding
- Exceptions
  – Java snippets are invoked synchronously and run on the thread that is created to complete processing of the previous activity
  – If an activity is configured to “participate”, that activity gets invoked synchronously and runs on the thread that is created to complete processing of the previous activity
  – If the activity is an invoke activity, the thread blocks wait for the response, which assumes that the invoke targets respond quickly

Figure 8-25. Business processes WB868 / ZB8681.0

Notes:

WebSphere Education

Invoking business processes

- Invoking microflows
  - Prefer synchronous but asynchronous is allowed
  - Invoke with either a one-way or request/response operation
  - All bindings are appropriate
- Invoking long-running business processes
  - Must be asynchronous
  - Invoke by using a one-way operation is preferred
  - The calling thread blocks, waiting for the response if a long-running process is invoked by using a request/response operation without explicitly using callback semantics upstream

Figure 8-26. Invoking business processes WB868 / ZB8681.0

Notes:

WebSphere Education

Invoking services from business processes

- Invokes from long-running processes

Intramodule

– SCA bindings only

– Preferred interaction style of the target SCA component is asynchronous SCA

– Callback semantics are used for asynchronous, request/response operations

Intermodule and external (non-SCA) services

– Threading module of the target import, export, or adapter is used

– Synchronous SCA is used if bindings are SCA and the target has a preferred interaction style of “synchronous”

– Callback semantics are used for asynchronous, request/response operations

– Asynchronous bindings are required if making multiple, parallel invocations

Figure 8-27. Invoking services from business processes WB868 / ZB8681.0

Notes:

8.2. Threading considerations

To achieve higher throughput, you need to run the applications in a multi-threaded environment. This topic introduces some of the best practices that are specific to threading.

WebSphere Education

Threading considerations

9.1

Figure 8-28. Threading considerations WB868 / ZB8681.0

Notes:

WebSphere Education

Long-running processes case study

- If all of the threads available to the internal BPE queue activation specification are used and blocked, no long-running process will be able to progress to its next activity
  – In a thread exhaustion, no long-running process will be able to progress to its next activity
- The internal BPE queue uses JMS bindings on the SIBus destination
- Tune the thread pools

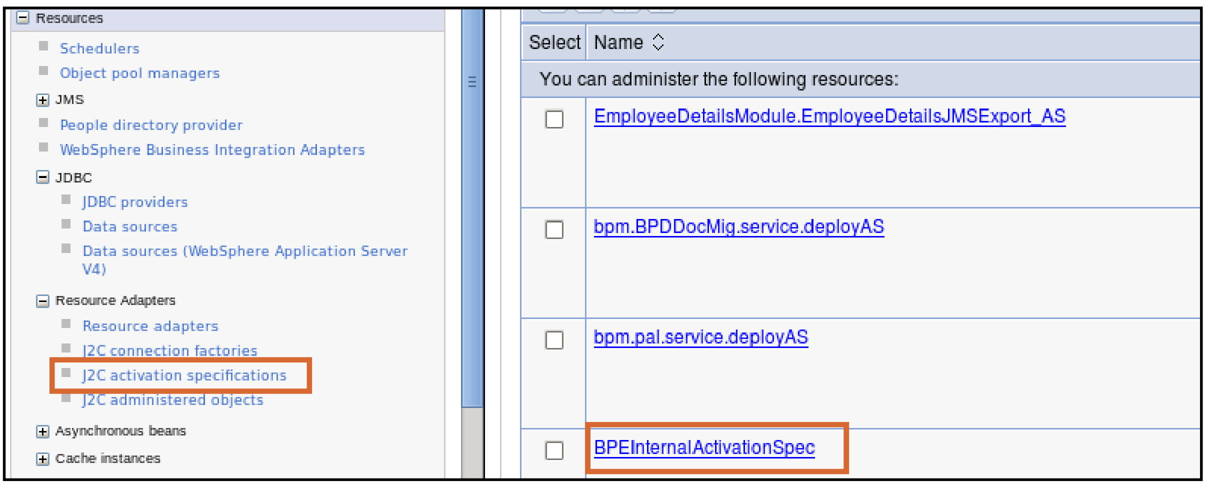


Figure 8-29. Long-running processes case study WB868 / ZB8681.0

Notes:

If all of the threads available to the internal BPE queue activation specification are used and blocked, no long-running process is able to progress to its next activity. To avoid this problem, consider the following practices:

- Tune the internal BPE queue activation specification, considering all activity for the transition of long-running processes in aggregate (not in-flight long-running processes).
- Design solutions to specifically avoid blocking internal BPE queue threads because solutions are likely to have many more in-flight long-running processes than activity for the transition of long-running processes.

WebSphere Education

Threading case studies

- An intramodule, one-way asynchronous binding can end up consuming all threads from a module's activation specification
  - Avoid the use of one-way, asynchronous bindings between SCA components in an SCA module
  - Use one-way asynchronous bindings between SCA modules
  - Use either synchronous SCA bindings or request/response asynchronous bindings between SCA components inside an SCA module

Figure 8-30. Threading case studies WB868 / ZB8681.0

Notes:

WebSphere Education

Threading case studies: JMS resource adapter

- Many blocked threads that are going against the resource adapter that is handling JMS bindings can consume all of the threads that the JMS resource adapter allocates
- This consumption of threads can prevent long-running processes from progressing from activity to activity
  - Avoid threads that block for long periods of time
  - Use callback when possible when downstream calls have high latencies
  - Configure the JMS resource adapter for a number of threads at least equal to the sum of the number of threads that are specified in all activation specifications that apply to the JMS resource adapter

Figure 8-31. Threading case studies: JMS resource adapter WB868 / ZB8681.0

Notes:

WebSphere Education

Threading case studies: Synchronous web services bindings

- Many clients that access BPM by using synchronous web services bindings can consume all of the WebContainer threads, preventing user interface applications that are running in the same application server instance from being available
  - Example: Business Process Choreographer Explorer
  - Front the application server with a web server (IBM HTTP Server) and limit the number of HTTP requests between the web servers and the application server
  - All HTTP requests should come through the web servers

Figure 8-32. Threading case studies: Synchronous web service bindings WB868 / ZB8681.0

Notes:

WebSphere Education

Overall threading consideration

- How many threads run in a single JVM?
  - **Platform Messaging Component SPI Resource Adapter**: Considering all asynchronous SCA bindings in aggregate
  - **ORB Thread Pool**: Considering all synchronous SCA bindings that are coming in from other JVMs
  - **SIBus JMS Resource Adapter**: Considering all JMS bindings by using SIBus destinations in aggregate
  - **Web Container Thread Pool**: Considering the maximum number of web services (SOAP over HTTP) clients that each JVM services, and the maximum number of Java EE HTTP requests that each JVM services, such as Business Process Choreographer Explorer
  - **WebSphere MQ Resource Adapter**: Considering all WebSphere MQ and MQ/JMS bindings in aggregate
  - Other thread pools: Third-party messaging providers
- Establish a target maximum number of threads per JVM and use this limit to drive towards asymmetrical deployments across multiple instances
  - The maximum number of threads per JVM is one of the system resource limits that need to be considered when deploying applications

Figure 8-33. Overall threading consideration WB868 / ZB8681.0

Notes:

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Describe the service component architecture (SCA) threading architecture
- Configure thread pools
- Tune thread pools that are based on binding types
- Explain some common thread issues

Figure 8-34. Unit summary WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Checkpoint questions

1. True or false: For an asynchronous request/response inter-JVM invocation, the callbacks run on threads that SCA imports create in the context of the associated SCA module's activation specification.

Figure 8-35. Checkpoint questions WB868 / ZB8681.0

Notes:

Write your answer here:

1.

WebSphere Education

Checkpoint answers

1. True

Figure 8-36. Checkpoint answers WB868 / ZB8681.0

Notes:

Unit 9. Business Process Choreography best practices

What this unit is about

This unit describes the runtime environment of a Business Process Choreographer engine, and the difference between microflows and long-running business processes is addressed in terms of transactions. It is important to understand how messages are persisted for a long-running business process. This unit points out some of the best practices for business process choreography.

What you should be able to do

After completing this unit, you should be able to:

- Describe the difference between long-running and microflow (short-running) business processes
- Explain the runtime architecture of long-running business processes
- Explain how events are persisted in long-running business processes
- Calculate a realistic database table growth rate for each solution, and consult with database administrators to establish a database strategy for sizing
- Identify the key best practices for the Business Process Choreographer
- Describe the runtime architecture of the Human Task Manager
- List and describe methods to optimize work item creation

How you will check your progress

- Checkpoint

References

IBM Business Process Manager V8.5 IBM Knowledge Center

`http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/ditamaps/ic-homepage-bpm.html`

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Describe the difference between long-running and microflow (short-running) business processes
- Explain the runtime architecture of long-running business processes
- Explain how events are persisted in long-running business processes
- Calculate a realistic database table growth rate for each solution, and consult with database administrators to establish a database strategy for sizing
- Identify the key best practices for the Business Process Choreographer
- Describe the runtime architecture of the Human Task Manager
- List and describe methods to optimize work item creation

Figure 9-1. Unit objectives WB868 / ZB8681.0

Notes:

WebSphere Education

Topics

- Business processes runtime architecture
- Business processes runtime considerations
- Business Process Choreographer best practices
- Optimizing work item creation

Figure 9-2. Topics WB868 / ZB8681.0

Notes:

9.1. Business processes runtime architecture

This topic explains the runtime architecture of event flow in Business Process Choreographer. The basic message navigation mechanism of the business flow manager is discussed.

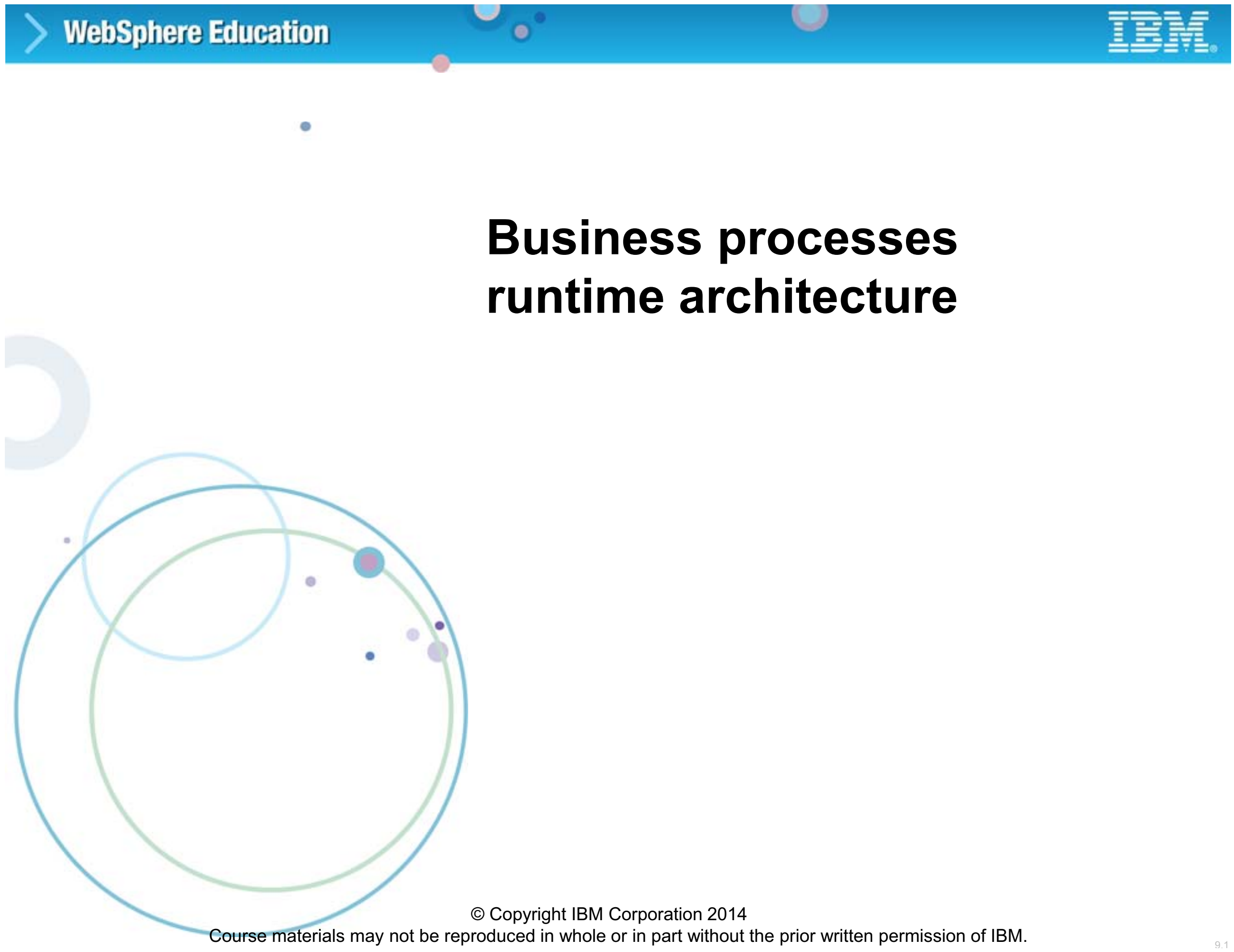


Figure 9-3. Business processes runtime architecture WB868 / ZB8681.0

Notes:

WebSphere Education

Microflows versus long-running processes

- Process Server runs business processes as either microflows (short-running) or long-running processes
  - Known as non-interruptible and interruptible processes
- A microflow process is used for running short business processes or small units of work within a larger business process (subprocesses)
  - Microflow completes or fails and is not persisted
- A long-running process might run for hours, days, or weeks
  - Frequently involve components with lengthy response times such as human tasks
  - State of the process must be persisted

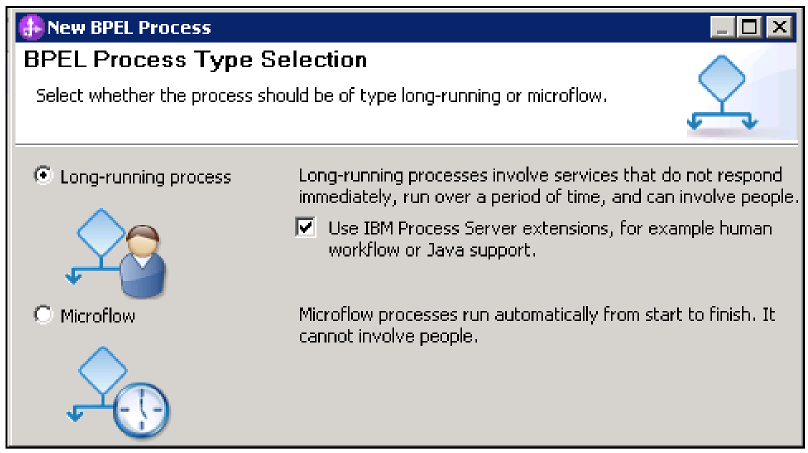


Figure 9-4. Microflows versus long-running processes WB868 / ZB8681.0

Notes:

In general, microflows choreograph the activities of target applications that respond relatively quickly. As such, microflows can be invoked synchronously where the "client" blocks and waits for the response from the business process, including all target applications. As such, microflows that are called synchronously run in the calling component's execution context, including its Java thread. This implementation tends to be the most efficient and yields the best performance both in response time and throughput.

More specifically, if a client synchronously invokes a microflow, the invocation uses standard EJB calling semantics. Since SCA modules are contained in a single EAR file, intramodule invocations are in the same JVM. Similarly, intermodule invocations might or might not be made in the same JVM depending on whether the EAR file is deployed in the same or different application server instances.

Microflows can also be invoked asynchronously. In this case, an MDB picks up the incoming request and runs the business process on an MDB-created thread. The MDB might or might not create a new thread that is for the business process, and such implementation details might change from release to release. However, the effect is the same in that the business process runs in a different execution context from the caller.

In general, long-running processes choreograph the activities of target applications that respond relatively slowly. As such, long-running processes are usually invoked asynchronously and invoke downstream services and components asynchronously as well.

When any SCA component is invoked asynchronously, the request is queued and picked up by an MDB. As a result, the SCA component (or as the case might be, the long-running process) runs in an execution context different from the caller.

WebSphere Education IBM

Microflow and long-running processes at run time

- The runtime behavior of a business process depends upon whether it is long-running or microflow (non-interruptible)

	Microflow (non-interruptible)	Long-running (interruptible)
Transactions	One transaction for the entire process	Transaction boundaries can be set between activities
Persistence	None	Process activity and state information is persisted to a database
Crash recovery	None (execution is transient; if the server crashes, process state is lost)	Process can resume from the last checkpoint
Parallelism	None, strictly single-threaded	Activities can run in parallel
Interruptible	No	Yes
Asynchronous	No	Yes
Correlation	No	Yes
Handlers	No support for fault, event, or compensation handlers	Handlers are supported

Figure 9-5. Microflow and long-running processes at run time WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Human interactions

- Work must be assigned an individual or group of individuals
- Business processes must halt execution and wait for work to be complete
- When the work is completed, the business process continues with execution
- Administrators or supervisors must be able to override, transfer, and manage tasks

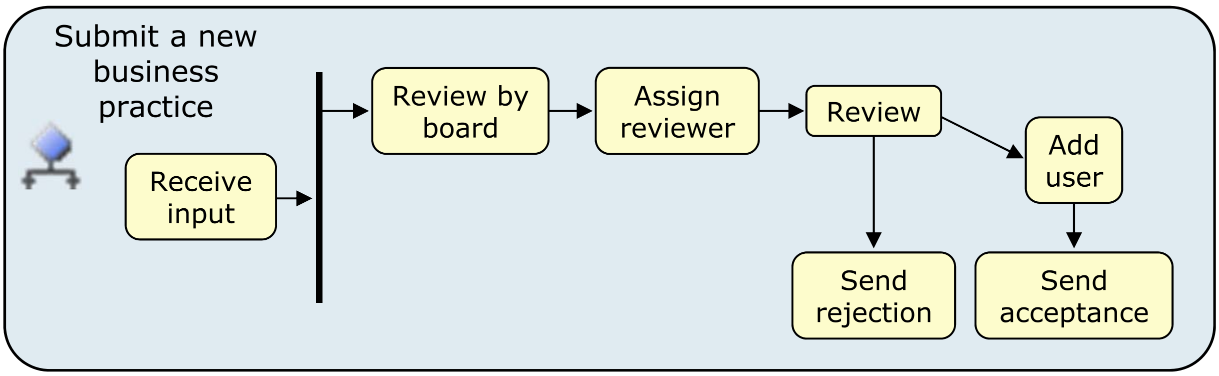


Figure 9-6. Human interactions WB868 / ZB8681.0

Notes:

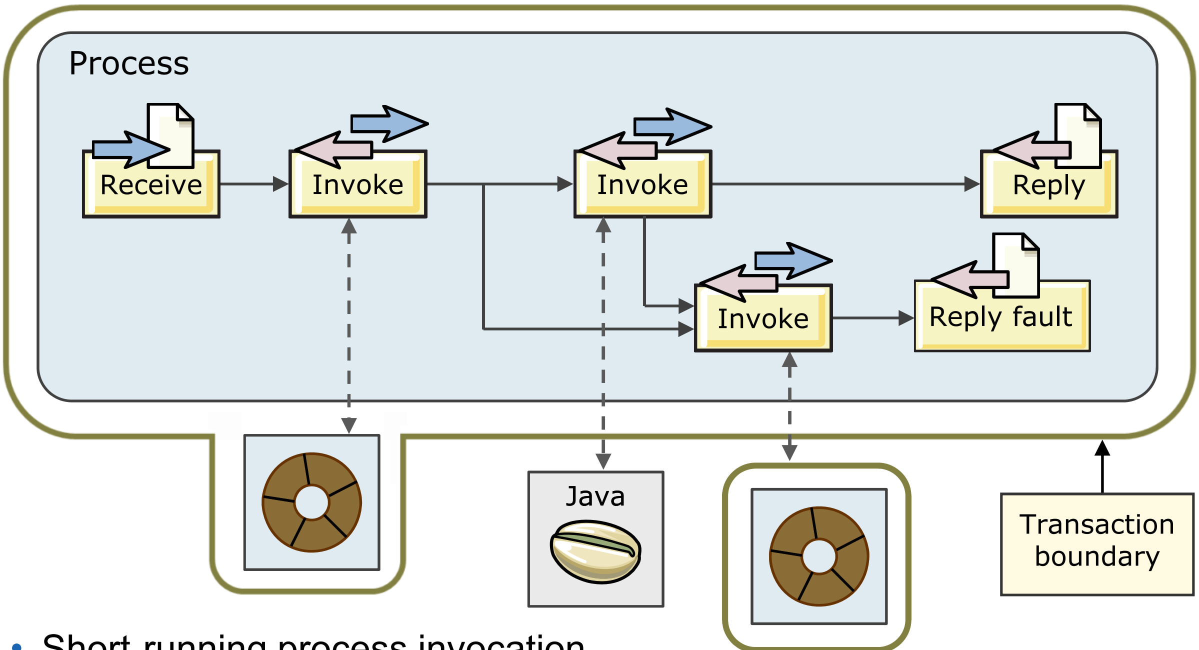
Although a business process can automate many steps, it might still be necessary for a human to be part of the business process. For most business processes, an individual views information that the business process generates, and acts upon that information. The actions of the individual might then be used within the business process and affect the execution path of the business process.

To have a certain individual do a specific task in a business process, a work item with certain permissions and capabilities is generated for the individual by the runtime environment. It is up to the individual to then check and retrieve any work items that might be created. Based on the permission level, the individual might be able to input information and complete the activity, or read the activity.

Administrators of the business process can override tasks (work items), which are assigned to an individual, by completing the activity.

WebSphere Education IBM

SCA invocation architecture: Microflow



- Short-running process invocation
  - Synchronous short-running process (microflow) invocation
  - Asynchronous long-running process invocation
  - For short-running processes, all activities run in the same transaction

Figure 9-7. SCA invocation architecture: Microflow WB868 / ZB8681.0

Notes:

Synchronous short-running process (microflow) invocation:

- The caller invokes the EJB (SCA) interface.
- Activities are run in the caller’s execution context (EJBs).
- Same JVM: All components in an SCA module that is run in the same JVM.

Asynchronous long-running process invocation:

- The MDB (SCA) invokes the process (EJB) when the event arrives.
- All activities run on a thread that the SCA MDB in the EJB container creates.
- Same JVM: All components in an SCA module that is run in the same JVM.
- This thread is a different thread from the caller’s thread.

For short-running processes, all activities run in the same transaction.

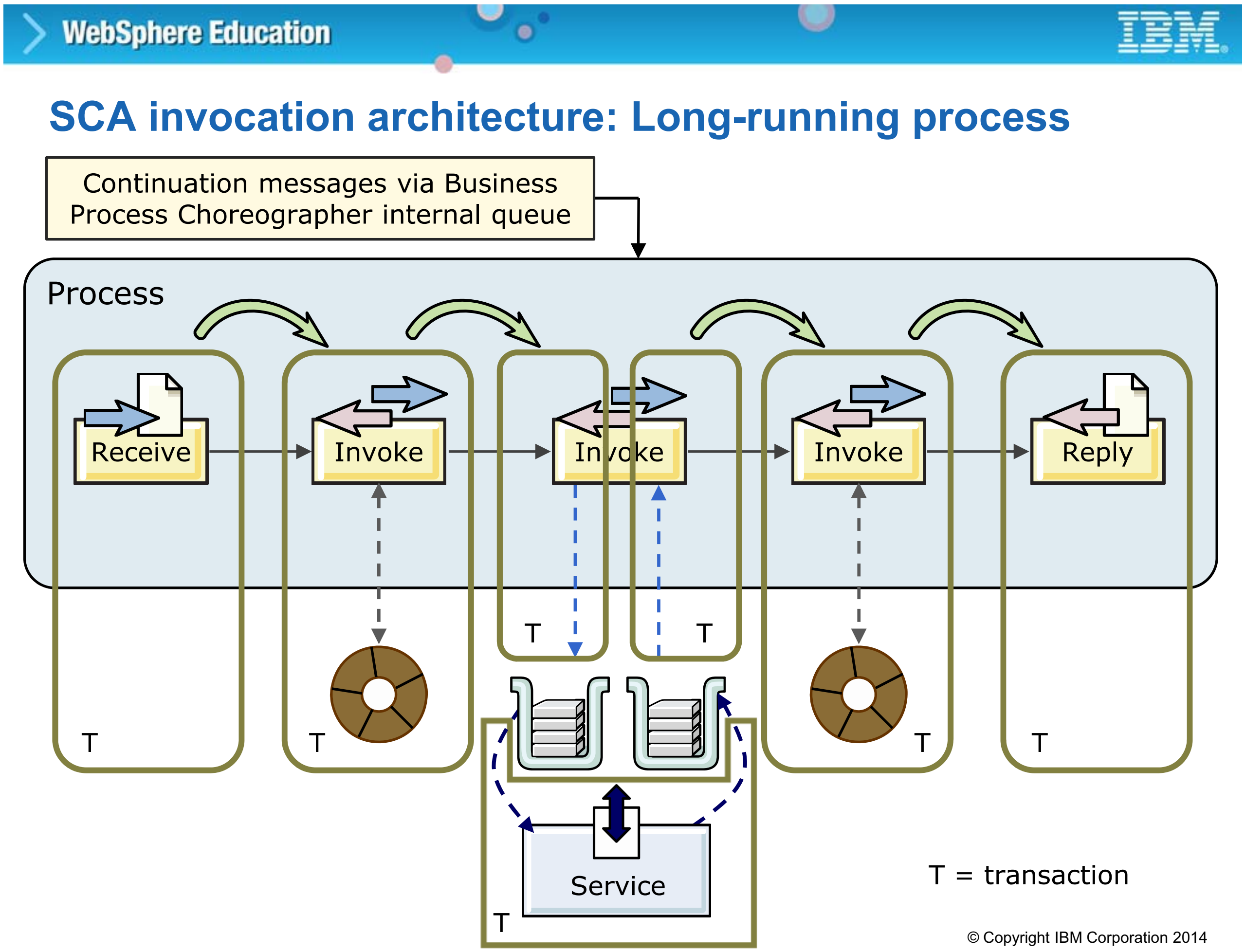


Figure 9-8. SCA invocation architecture: Long-running process WB868 / ZB8681.0

Notes:

The following attributes apply to long-running processes:

- Performance:
  - Lower throughput when compared to microflows because they interact with queues and a database to reliably hold the state of the process. (It can be up to 100 times slower than microflows.)
- Appropriate when:
  - The lifetime of the process is longer than the transaction timeout of the system.
  - The process requires human interaction or event activities.
  - The process invokes services asynchronously.
  - The process contains activities that need to be transactionally decoupled from other activities.
  - The process communicates with another process.

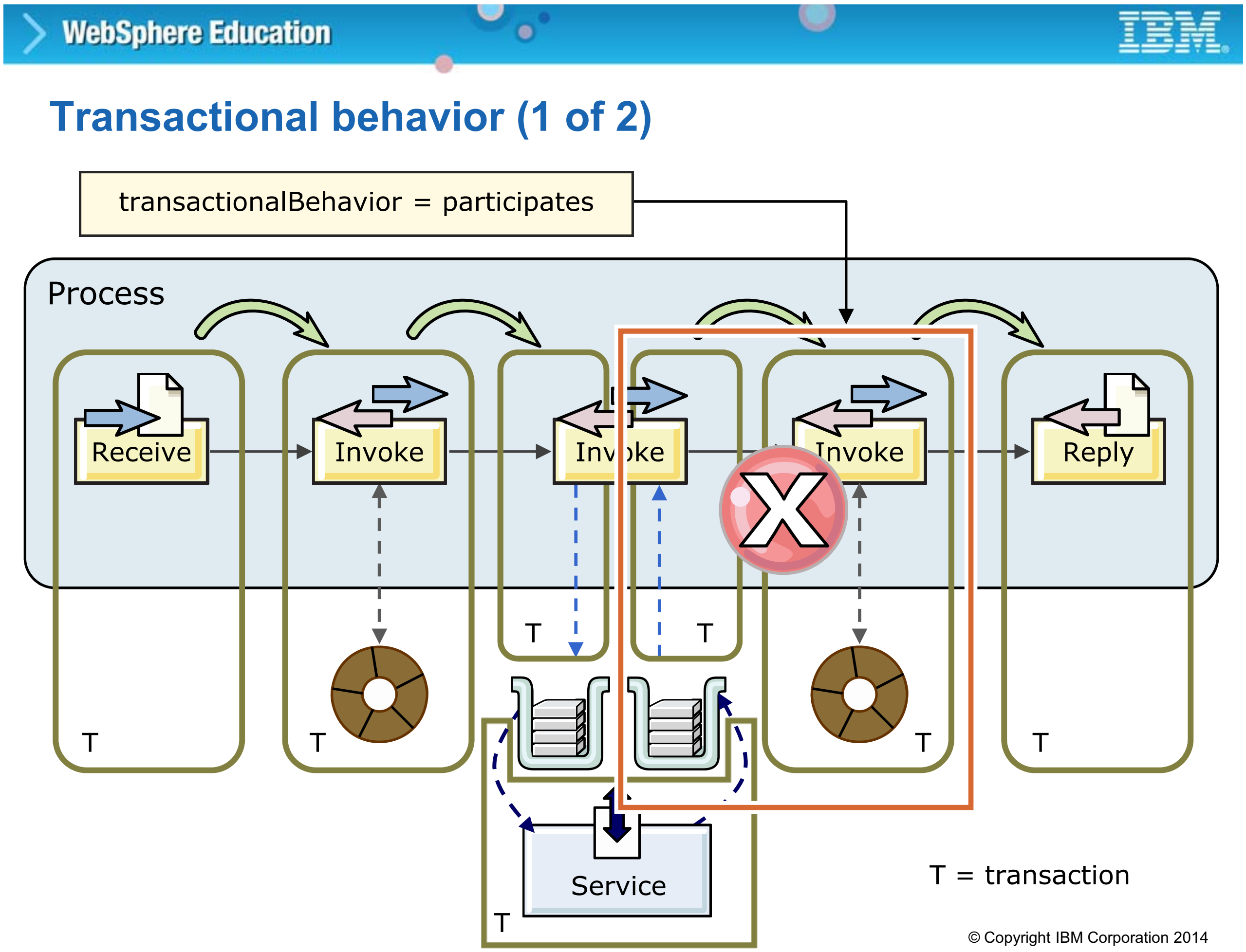


Figure 9-9. Transactional behavior (1 of 2) WB868 / ZB8681.0

Notes:

Use "**Commit after**" for inline human tasks to increase responsiveness to human users. When a human user issues a task completion, the thread or transaction handling for that action is used to resume navigation of the human task activity in the process flow. The user's task completion action does not complete until the process engine commits the transaction. If the Participates setting is used, the commit gets delayed and forces longer response time to the user. This response is a classic response time versus throughput trade-off.

Certain Business Process Execution Language (BPEL) activities have user-defined transaction attributes:

- Commit Before ensures that the current transaction ends and a new transaction begin before executing the activity.
- Commit After ensures that the current transaction ends as soon as the activity finishes.
- Participates transaction behavior depends on the previous and next activities.
  - If the previous activity did not commit the transaction, execute the activity in the same transaction.

- If the previous activity committed the transaction, start a new transaction before executing the activity.

- Requires Own is a combination of commit before and commit after; the container starts a new transaction, executes the activity, and then ends the transaction.

WebSphere Education

Transactional behavior (2 of 2)

- Commit before
  - The transactions that precede it must be fully committed before it can begin
  - This activity still tolerates being in a transaction with other activities that follow it
- Commit after
  - The activity must be committed immediately after it is completed
  - This activity still tolerates being in a transaction with other activities that precede it
- Participates
  - The activity does not require a commit to occur either before or after it, or where it can coexist within a transaction where one or more other activities will be invoked
  - Provides best throughput performance
- Requires own
  - The activity must be isolated within its own transaction

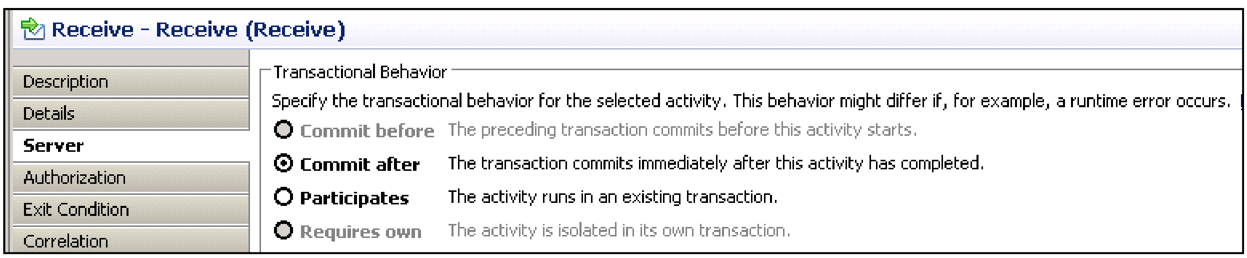


Figure 9-10. Transactional behavior (2 of 2) WB868 / ZB8681.0

Notes:

With **Participates**, the activity participates in the current transaction. More transaction boundaries are not set, either before nor after the activity.

In the following situations, this setting allows the transaction to continue with the navigation of the following activities, which depend on the values of their settings of the transactional behavior attributes.

- If the invoke activity invokes the service asynchronously, the arrival of the response message triggers a new transaction. The transaction is short because it commits immediately after the status of the invoke activity is updated.
- In a sequence of human task activities, two transactions are needed for each human task activity, one to activate the human task activity and another to complete the human task activity. If you change the setting to Participates, you can reduce the number of transactions to one for each human task activity. This option works because the completion of the previous human task activity and the activation of the following activity are done in the same transaction.
- To enable server-controlled page flows that use the completeAndClaimSuccessor API.

9.2. Business processes runtime considerations

WebSphere Education

Business processes runtime considerations

9.1

Figure 9-11. Business processes runtime considerations WB868 / ZB8681.0

Notes:

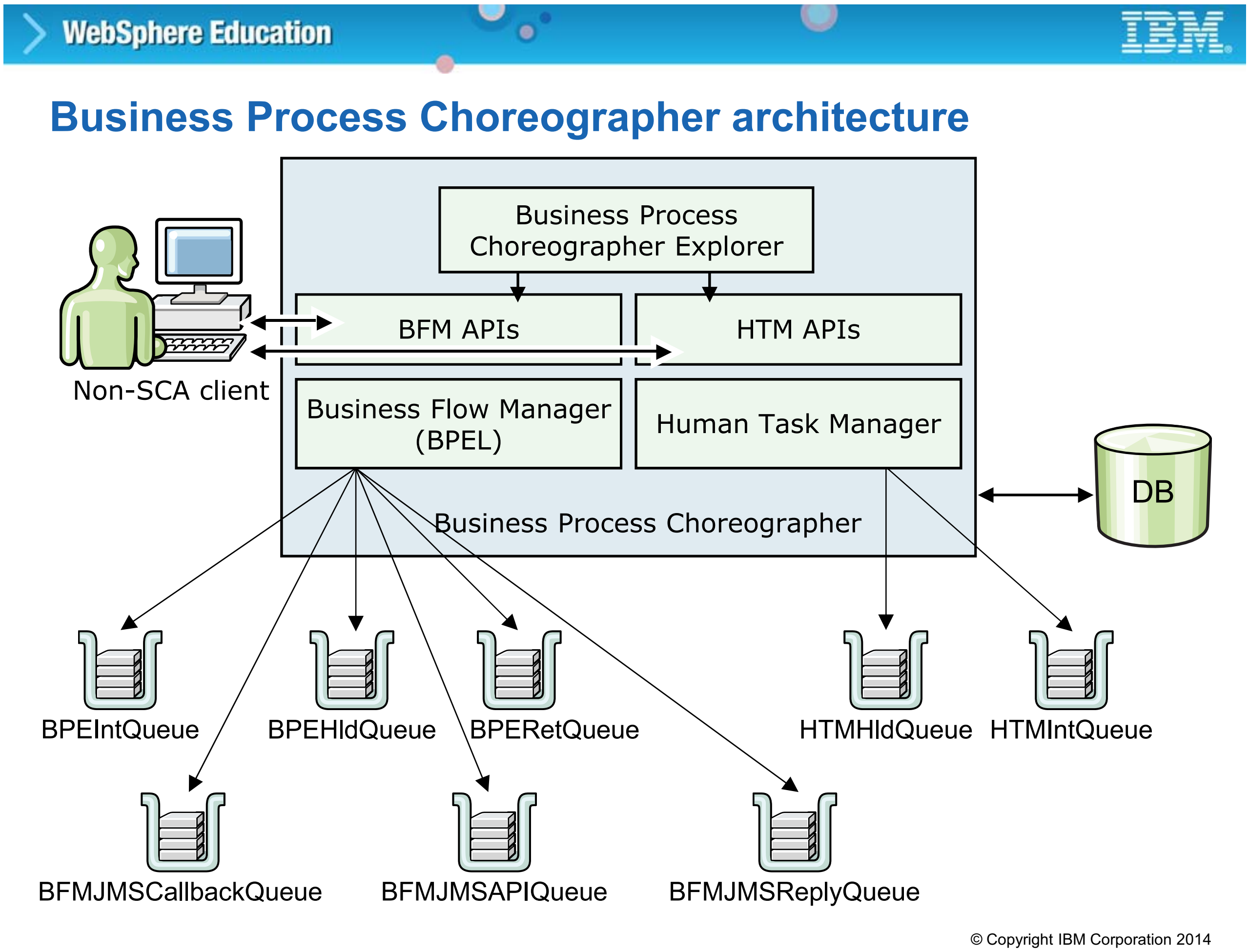


Figure 9-12. Business Process Choreographer architecture WB868 / ZB8681.0

Notes:

There is an internal BPE queue for long-running processes. The MDB listening to the internal BPE queue configured in the **BPEIntQ Activation Spec** controls concurrency for long-running processes. Target concurrency capabilities become the critical issue. The amount of time that is spent in transition between activities is negligible.

WebSphere Education

Long-running process: Persistent navigation

- Navigating through a long-running process is done by using persistent messaging and database access
  - An activity is started from an incoming (Business Process Choreographer internal) continuation message, which also starts the transaction in which the activity is processed
  - The state of the process instance and activity instance is read from the database
  - The activity is executed (for example, the web service of an invoke activity is invoked)
  - Changes to process instance and activity instance are written back into the database
  - For each outgoing link, a continuation message is created and sent
  - The transaction commits
- In case something goes wrong during the processing of the activity and a rollback is required, only the in-flight work of the current activity is lost

Figure 9-13. Long-running process: Persistent navigation WB868 / ZB8681.0

Notes:

WebSphere Education

Business process templates and instances

- Process templates describe the business process model
  – The Business Flow Manager uses the template to create instances of the business process at run time
  – Process templates are deployed and installed on IBM Process Server
- Process instances are entities that exist at run time
  – An instance represents one running business process, its specific data, and its state
  – The Business Flow Manager can run multiple process instances at the same time
- Creation of template version is supported by using the template's valid-from date
  – The valid-from date is used to decide which process template to use when creating a process instance
  – When the instance is created, it runs against that version of the template
  – If a new template is deployed, in-flight process instances can be upgraded
- Process templates and instances are persisted to a database
  – In IBM Integration Designer, the database is in DB2 Express
  – For production environments, database creation scripts are provided per platform

Figure 9-14. Business process templates and instances WB868 / ZB8681.0

Notes:

A BPEL process template is a process definition that is deployed and installed in the runtime environment.

Process properties are specified when the process is defined. In the runtime environment, properties for process templates are stored in the runtime database. They can be accessed by using the Business Process Choreographer database views, such as the PROCESS_TEMPLATE view, or by using query tables.

In addition, an installed BPEL process can also have one of the following states:

- **Started**: When a process template is created and started, new instances of the template can be started.
- **Stopped**: When a process template is in the stopped state, no new instances of this template can be created and started. Existing instances of the template continue to run until they complete.

WebSphere Education

Understand your processes

- Every time an activity executes, the Business Process Choreographer container accesses a dozen or more tables
- It is critical to plan and to size the database properly
- To plan the database, you need to understand how much data you are storing in the database

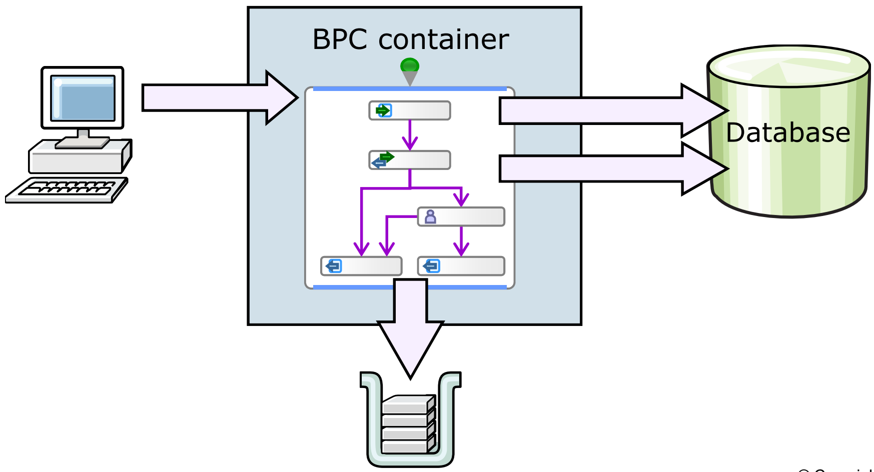


Figure 9-15. Understand your processes WB868 / ZB8681.0

Notes:

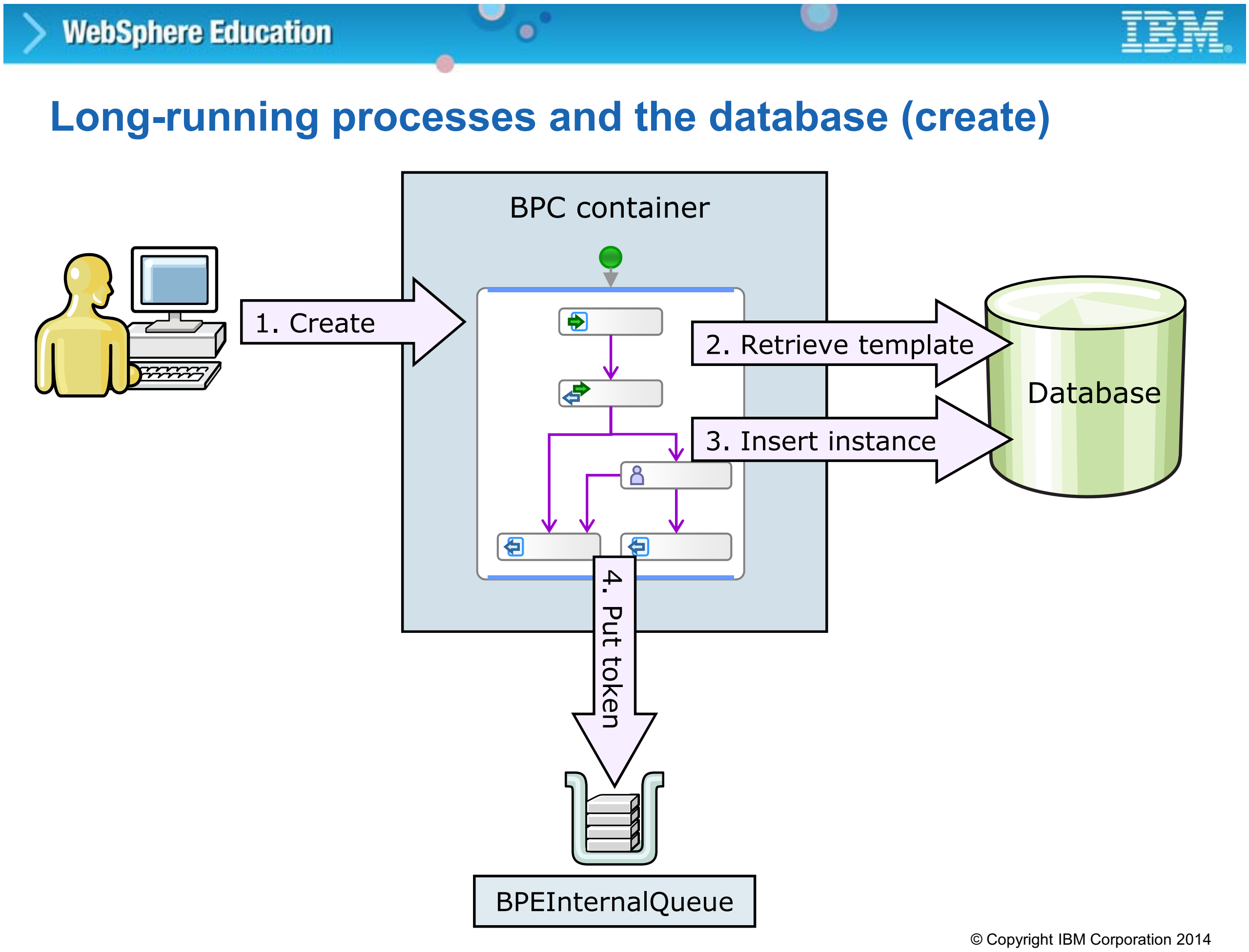


Figure 9-16. Long-running processes and the database (create) WB868 / ZB8681.0

Notes:

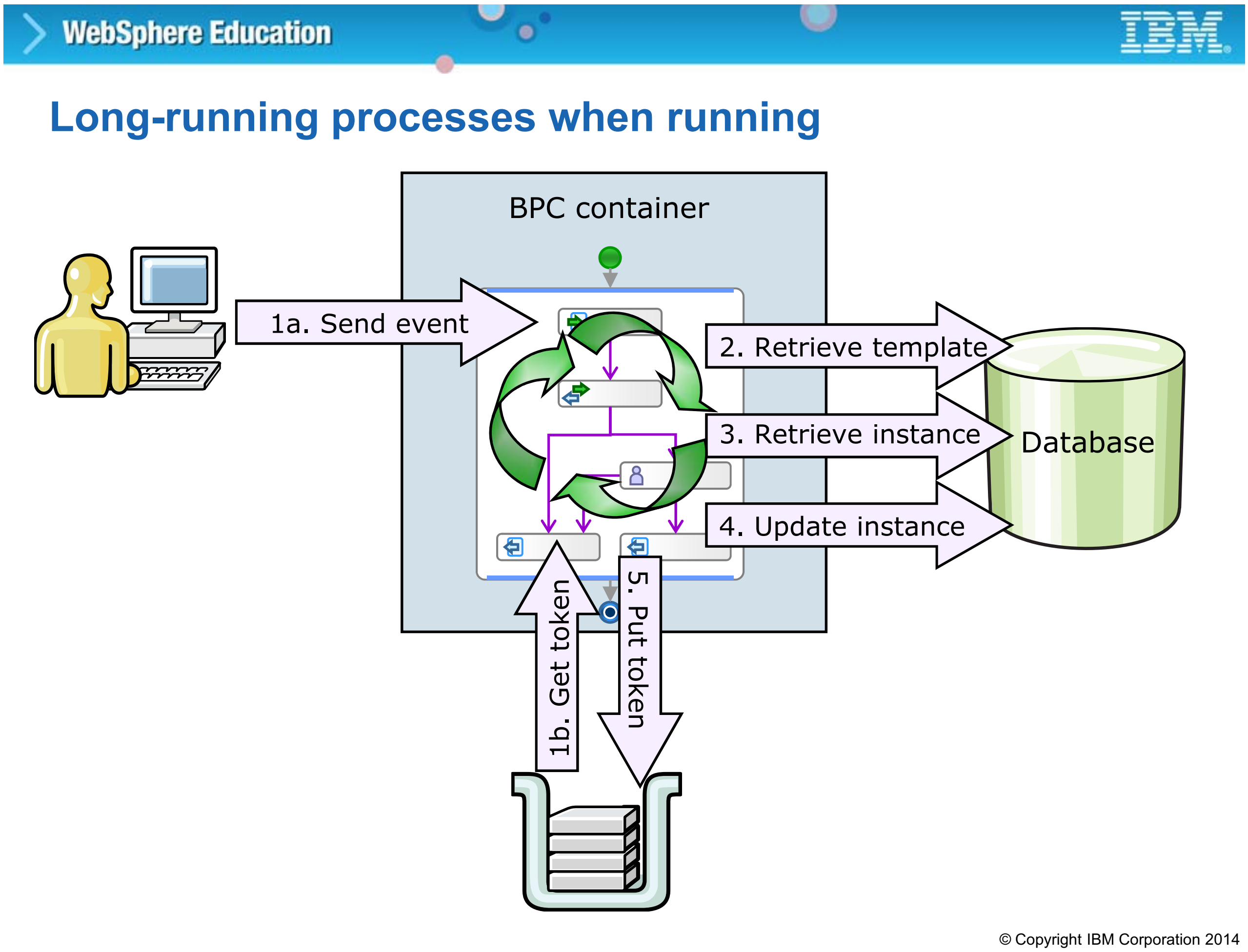


Figure 9-17. Long-running processes when running WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Calculating process instances for sizing (1 of 3)

- The number of in-flight instances is critical to:
  - Your database configuration
  - Performance of your process container
- Calculating in-flight instances:
  - How many new instances do you add per day?
  - What is the average life span of your instances?
  - In-flight = (new instances) * (life span)
- **Example:**
  - 30,000 documents per day (one instance per document)
  - 30-day average life per instance
  - In-flight = 900,000 instances in the database

Figure 9-18. Calculating process instances for sizing (1 of 3) WB868 / ZB8681.0

Notes:

WebSphere Education

Calculating process instances for sizing (2 of 3)

- Some processes have happy path and error path
  - Update your calculation appropriately
- Calculating in-flight instances (less simple)
- 30,000 instances per day
  - 95% is happy path and have a 5-day duration
  - 5% is error path and have a 30-day duration
  - In-flight = (30,000 * 0.95 *5) + (30,000 * 0.05 *30)
  - In-flight = **187,500 instances**

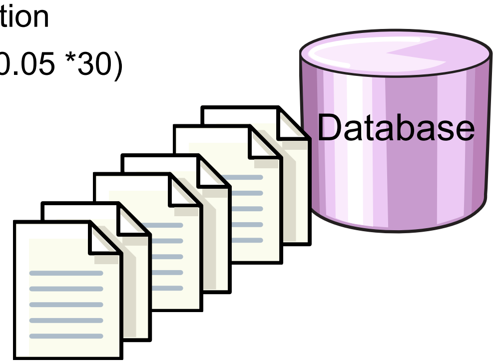


Figure 9-19. Calculating process instances for sizing (2 of 3) WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Calculating process instances for sizing (3 of 3)

- Some companies want to retain their instances after completion for contract purposes
  - Some customers retain instances for monitoring purposes
  - Instead, consider a "passivate" service where you call it to write all of the process information out as the last activity in the flow, and then delete the instance upon completion
- In-flight process count has to include both running and completed but not deleted process instances
- If previous example has a 30-day retention period:
  - In-flight = (new Instances) * (happy %) * (happy duration)
    + (new Instances) * (error %) * (error duration)
    + (new Instances) * (retention period)
  - In-flight = (30,000 * 0.95 *5) + (30,000 * 0.05 *30) + (30 K * 30 days)
  - In-flight = **1.08 million** instances

Figure 9-20. Calculating process instances for sizing (3 of 3) WB868 / ZB8681.0

Notes:

WebSphere Education

Process instances for sizing best practice

- A process instance can end with **finished**, **failed**, or **terminated** state
- The recommendation is to delete the process instance data upon the successful completion
  - ***Yes*** option can be problematic if the process instance ends in a failed state
  - ***No*** option quickly fills out the database

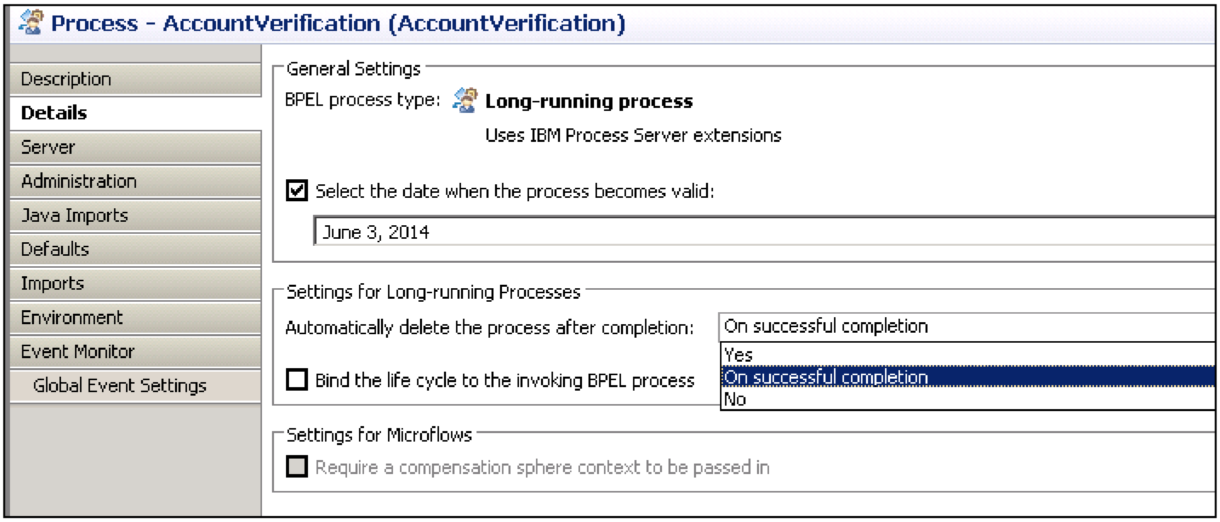


Figure 9-21. Process instances for sizing best practice WB868 / ZB8681.0

Notes:

For business processes:

The property **Automatically delete the process after completion** can have the value Yes, No, or On successful completion. If this property has the value No or On successful completion, it makes sense to configure a cleanup job to delete the process instances.

For human tasks:

The property **Auto deletion mode** can have the value On completion, or On successful completion (which is the default). Only deletion takes place, and you can change the value for Auto deletion mode, if the property Duration until task is deleted has either the value Immediate or a defined interval. If the property Duration until task is deleted has the value Never, automatic deletion is disabled. The Auto deletion mode property cannot be changed, and it makes sense to configure a cleanup job to delete the human tasks. Otherwise, if **Duration until task is deleted** does not have the value Never, and **Auto deletion mode** has the value On successful completion, then it makes sense to define a cleanup job. This cleanup job should delete the human tasks that do not complete successfully.

WebSphere Education IBM

Process instances are cumulative (1 of 2)

- The number of in-flight process instances can be up to five times the number of documents read, depending on the path taken
  - Credit check gets called on risky customers only (about 30%)
  - 10% of risky customers fail the credit check, so ManageInventory and BillCustomer are not called (for example, they get called only 97% of the time)
  - The customer averages 50,000 documents per day with 20 days average lifecycle, 30 days retention

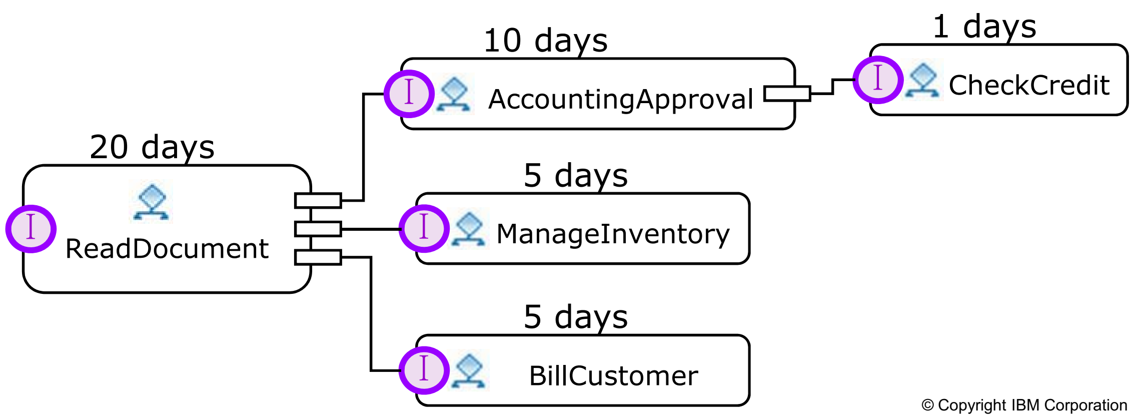


Figure 9-22. Process instances are cumulative (1 of 2) WB868 / ZB8681.0

Notes:

If this example is a single large BPEL, the total in-flight instances would be about 2+ million. The maintenance would become nearly impossible.

WebSphere Education

Process instances are cumulative (2 of 2)

- If you take the earlier assumptions, your calculations can end up like this example:
  - **ReadDocument** = 50,000 * 20 days + 50,000 * 30 days retention
  - **AccountingApproval** = 50,000 * 10 days + 50,000 * 30 days retention
  - **CreditCheck** = 50,000 * 30 % * 1 days + 50,000 * 30% * 30 days retention
  - **ManageInventory** = 50,000 * 97% * 5 days + 50,000 * 97% * 30 days retention
  - **BillCustomer** = 50,000 * 97 % * 5 days + 50,000 * 97% * 30 days retention
  - **Total in-flight = 8.86 million predicted process instances**

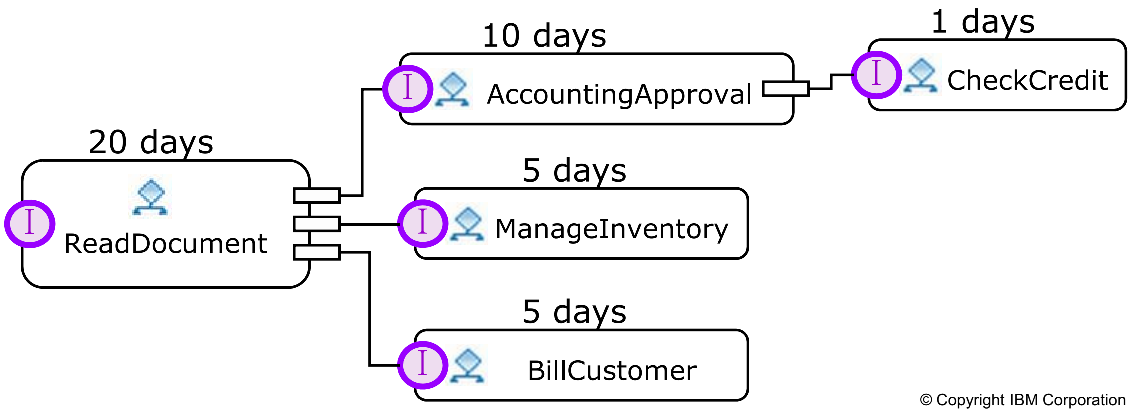


Figure 9-23. Process instances are cumulative (2 of 2) WB868 / ZB8681.0

Notes:

If this example is a single large BPEL, the total in-flight instances would be about 2+ million. The maintenance would become nearly impossible.

WebSphere Education

Instance tables

When you create a new process instance, components of that instance are included in about a dozen tables

- In general, the main table is **`PROCESS_INSTANCE_B_T`**
  - There is one row in this table for every process instance
- **`SCOPED_VARIABLE_INSTANCE_B_T`**
  - Each process instance variable is stored in its own row with serialized data of the variable's message
  - If you have 30 variables and 1 million process instances, you have a table of 30 million rows
  - If you have large messages, it can make your table huge
  - 1 million instances, 30 variables, average message is 1 KB, table size is 30 GB
- **`ACTIVITY_INSTANCE_B_T`** and **`SCOPE_INSTANCE_B_T`**
  - When an activity is in the middle of executing, its instance information is stored in this table
  - When the activity is marked as being "business-relevant," then you have to worry
  - These activities are kept in the database after they execute, which clients can observe (Business Process Choreographer Explorer)
  - The number of rows in the table grows as a multiple of the process instance count
  - Some activities such as invoke have an implicit scope around them
  - Keep an eye out for the SCOPE table if you mark activities business-relevant because it grows with the activity table; sometimes there can be large messages in the SCOPE table

Figure 9-24. Instance tables WB868 / ZB8681.0

Notes:

Business-relevant data is activity-related data that gets deleted after the completion of its execution. The business-relevant data is not persisted, and exists in the memory only during the activity execution unless the developer specifically tells it to be persisted.

If the "**delete the instances upon completion**" option is turned on, the business-relevant data gets deleted as well.

WebSphere Education

Property tables

- Some property tables that are in essence name-value paired data that is related to either variables, activities or process instances
- Tables:
  - `PROCESS_INSTANCE_ATTRIBUTE_T`
  - `ACTIVITY_INSTANCE_ATTRIBUTE_T`
  - `QUERYABLE_VARIABLE_INSTANCE_T`
- Query-able variable table has one row for every variable property that is marked as query-able
  - The table size can be calculated as: (In-flight) * (properties that are exposed as query-able)
- Process and activity tables have one row per property set via the process API
  - Because customers use these properties to help search and monitor the instances, these tables can grow large

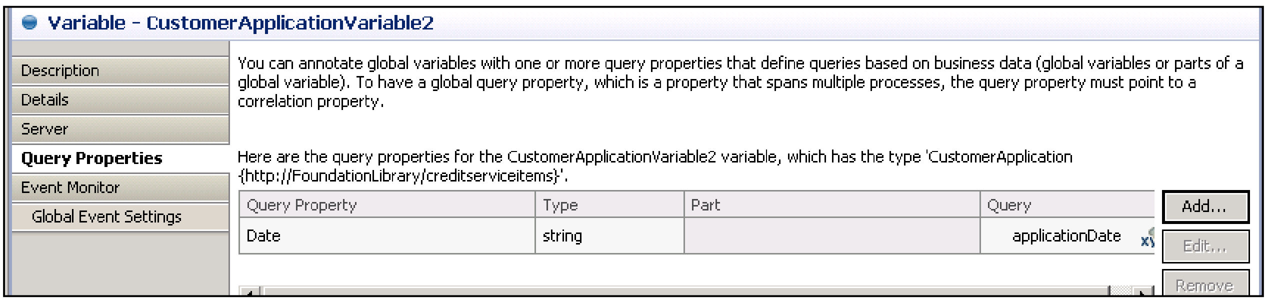


Figure 9-25. Property tables WB868 / ZB8681.0

Notes:

WebSphere Education

Utility and log tables

The BPEL container uses some internal tables to manage the system

- **`SCHEMA_VERSION`**: Is used to track the schema version
- **`GRAPHICAL_PROCESS_MODEL_T`**: Stores the graphics that are shown in the Business Process Choreographer Explorer's **View Process State** page
- **`PROCESS_CELL_MAP_T`**: Maps on which cell a process is installed
- **`PROCESS_CONTEXT_T`**: Holds the calling ***thread information*** and ***work area*** There is one row for every in-flight instance, and this table can be large

Log tables: **`AUDIT_LOG`** and **`TASK_AUDIT_LOG`**

- These tables are insert-only: they constantly grow as the processes add more data to them
- The Business Process Choreographer container generates on average about two rows of audit log data for each activity
- Information about the event like the variable data is stored in the table
- If you have large messages, the AUDIT LOG table gets large
- The administrator is responsible for purging these tables at regular intervals

Figure 9-26. Utility and log tables WB868 / ZB8681.0

Notes:

Do not use AUDIT_LOG if possible. If you need it, take the top 10,000 entries and delete the rest every day, or come up with a strategy to reduce the size of the table regularly.

You should ***not*** be accessing the database directly to query how many instances exist. The process monitoring should be done by using external monitoring tools.

9.3. Business Process Choreographer best practices

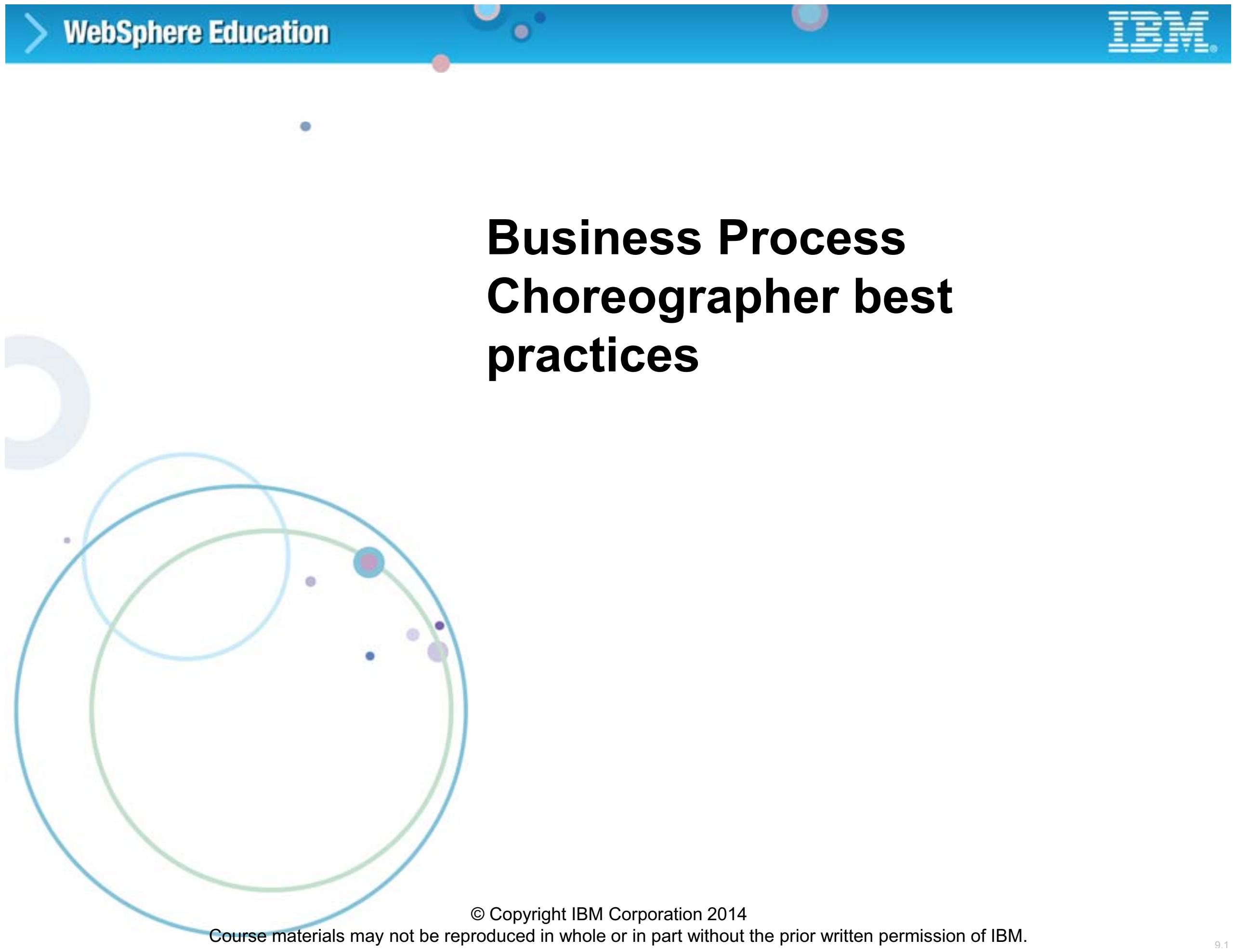


Figure 9-27. Business Process Choreographer best practices WB868 / ZB8681.0

Notes:

WebSphere Education

Business Process Choreographer tuning checklist

- Use work-manager-based navigation for long-running processes
  - For work-manager-based navigation, optimize message pool size and inter-transaction cache size
- Optimize Business Flow Manager resources:
  - Database connection
  - Activation specification (BPEInternalActivationSpec)
  - JMS connection (BPECF and BPECFC)
- Optimize the database configuration for Business Process Choreographer database
- Optimize indexes for SQL statements that result from task and process list queries by using database tools
- Turn off state observers that are not needed
  - For example, turn off the audit logging

Figure 9-28. Business Process Choreographer tuning checklist WB868 / ZB8681.0

Notes:

Work-manager-based navigation is the default navigation mode for BPEL business processes (rather than JMS-based navigation). Work-manager-based navigation provides two methods to optimize performance, keeping the quality of service of (JMS-based) process navigation consistent with persistent messaging:

- **Work Manager-based navigation**: WorkManager is a thread pool of Java Platform, Enterprise Edition threads. WorkManager process navigation takes advantage of an underlying capability of WebSphere Application Server to start the processing of ready-to-navigate business flow activities without using messaging from JMS providers.
- **InterTransactionCache**: This cache is a part of the work-manager-based navigation mode that holds process instance state information in memory, reducing the need to retrieve information from the database.

WebSphere Education

JMS versus work-manager-based navigation

- Traditional Java Message Service (JMS)-based navigation (default) for business processes follows the well-known pattern of chained transactions with persistent messages
  - After each transaction, the thread context is thrown away, and cannot be used for optimizations
- Enable lightweight navigation style, work manager-based navigation
  - Work manager threads are used for thread creation
  - The `interTransactionCache` holds process instance state information in memory, reducing the need to retrieve information from the database
  - In clusters, workload management occurs for incoming requests only
  - The quality of service is the same as in JMS-based navigation
  - Communication to the database is ***reduced***
  - Performance improvement in extreme cases were a factor of 2

Figure 9-29. JMS versus work-manager-based navigation WB868 / ZB8681.0

Notes:

A work manager is a thread pool for Java Platform, Enterprise Edition threads. The use of a work manager allows the Business Process Choreographer engine to use an underlying capability of WebSphere Application Server to start the processing of ready-to-navigate business flow activities without using messaging as provided by JMS providers.

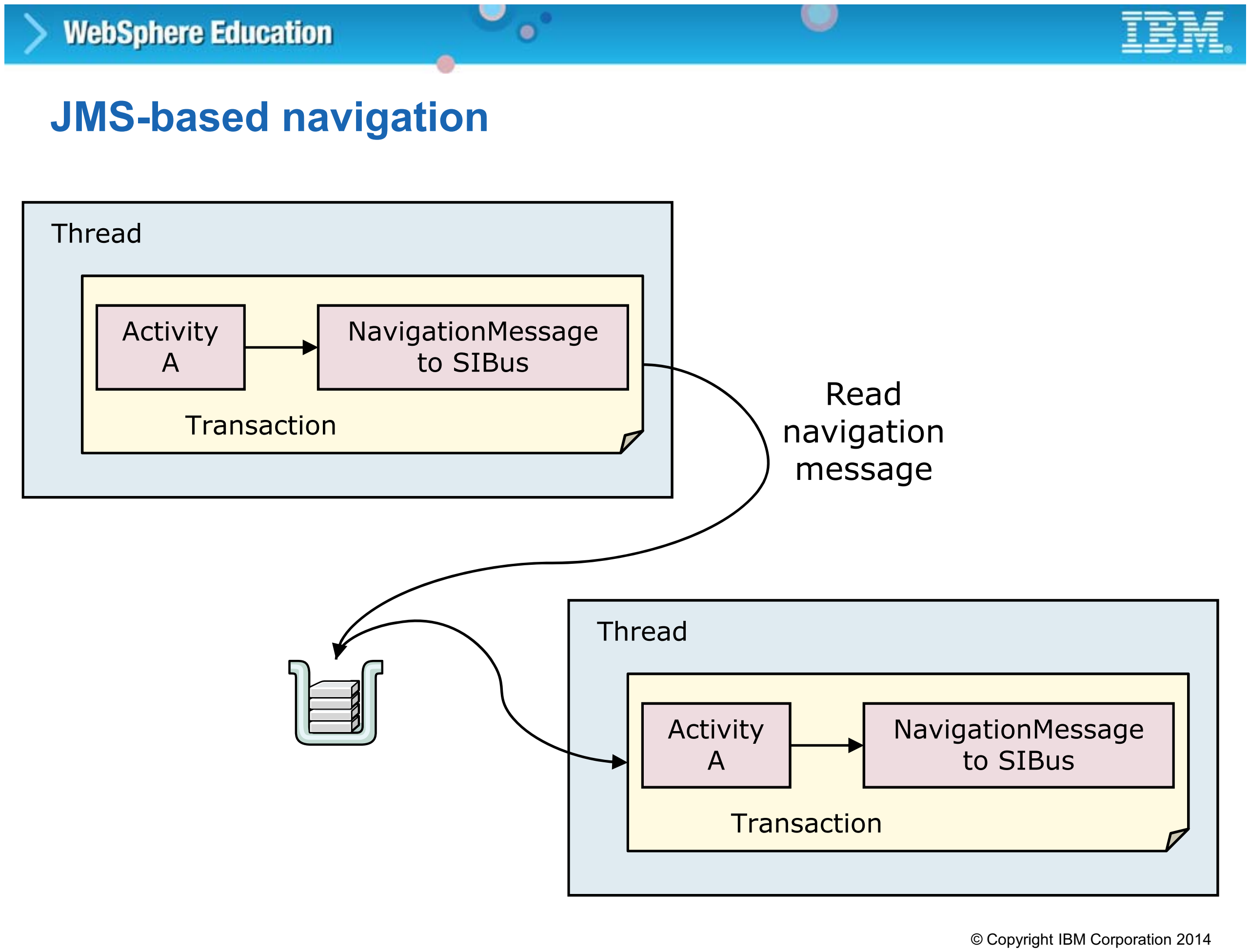


Figure 9-30. JMS-based navigation WB868 / ZB8681.0

Notes:

After each transaction, the thread context is discarded, and cannot be used for optimizations.

WebSphere Education IBM

Work-manager-based navigation

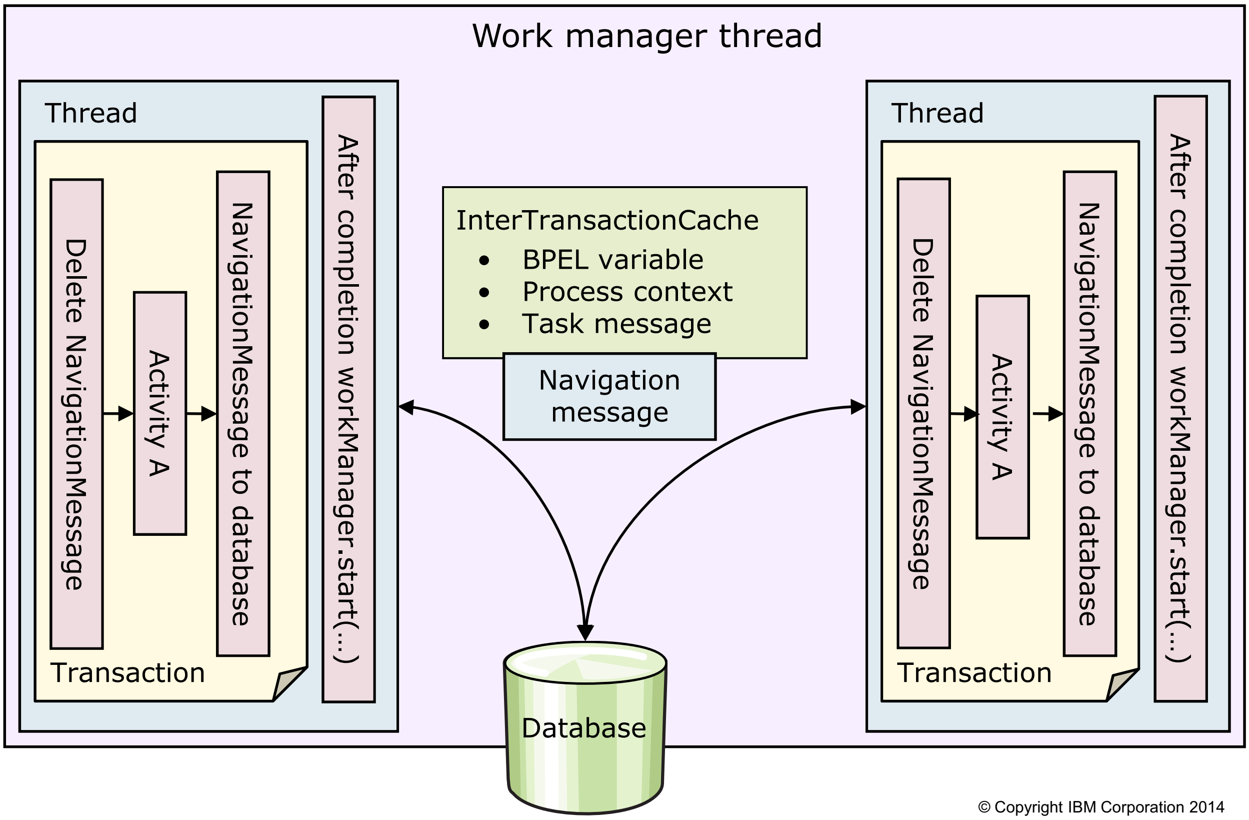


Figure 9-31. Work-manager-based navigation WB868 / ZB8681.0

Notes:

Work-manager-based navigation provides two performance optimizations, keeping the quality of service of process navigation with persistent messaging (JMS-based navigation):

- Work-manager-based navigation. A WorkManager is a thread pool for Java Platform, Enterprise Edition threads. WorkManager process navigation uses an underlying capability of WebSphere Application Server to start the processing of ready-to-navigate business flow activities without using messaging as provided by JMS providers.
- The InterTransactionCache, a part of the WorkManager-based navigation mode, holds process instance state information in memory, reducing the need to retrieve information from the database.

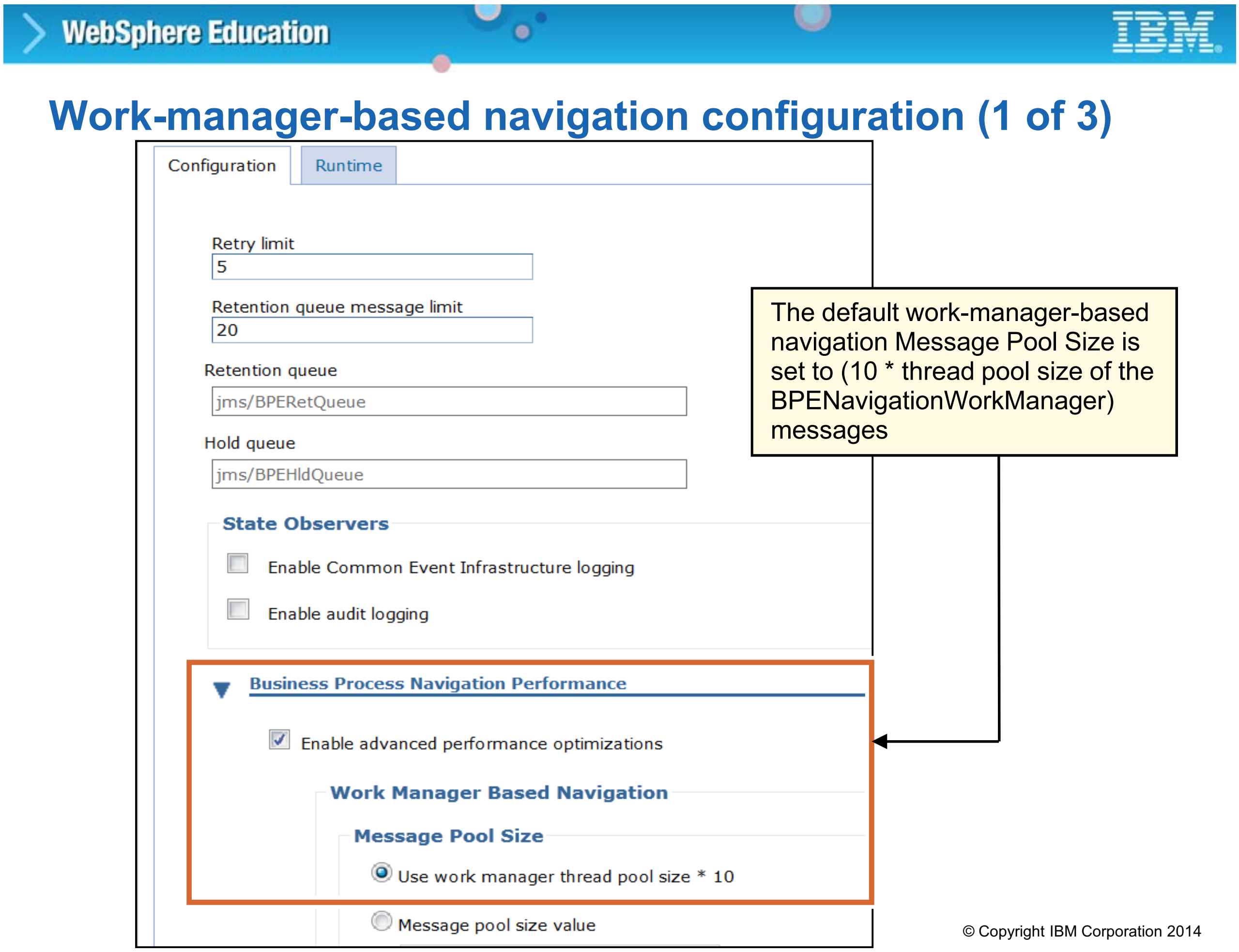


Figure 9-32. Work-manager-based navigation configuration (1 of 3) WB868 / ZB8681.0

Notes:

Check **Enable advanced performance optimization** to enable both the work-manager-based navigation and interTransactionCache optimizations.

The key parameters are:

- Check **Enable advanced performance optimization** to enable both the work-manager-based navigation and interTransactionCache optimizations.
- Work Manager Based Navigation Message Pool Size: This property specifies the size of the cache that is used for BPE navigation messages that cannot be processed immediately, provided work-manager-based navigation is enabled. The cache defaults to a size of (10 * thread pool size of the BPENavigationWorkManager) messages. If this cache reaches its limit, the Business Process Choreographer engine uses JMS-based navigation for new messages.
- InterTransaction Cache Size: This property specifies the size of the cache that is used to store process state information that is written to the database. It should be set to twice the number of parallel running process instances. The default value for this property is the thread pool size of the BPENavigationWorkManager.

WebSphere Education

Work-manager-based navigation configuration (2 of 3)

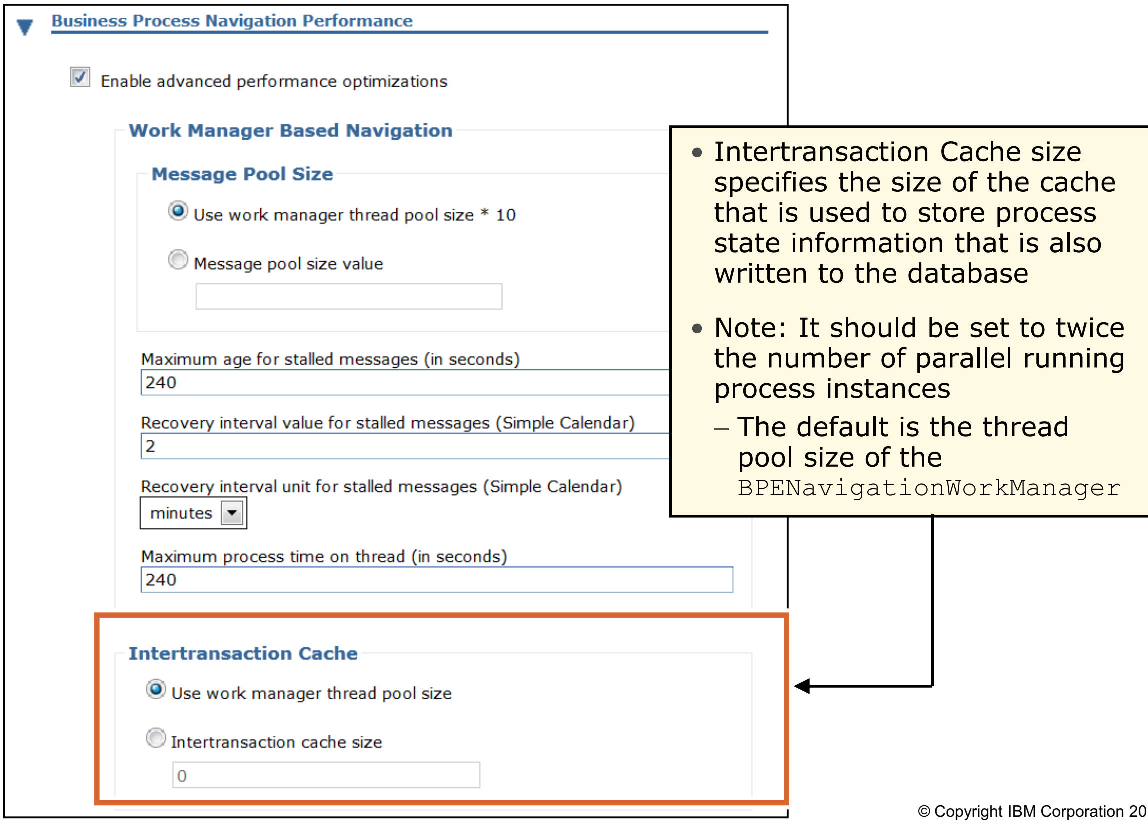


Figure 9-33. Work-manager-based navigation configuration (2 of 3) WB868 / ZB8681.0

Notes:

The intertransaction cache stores database objects such as BPEL variables and process contexts between transactions.

Select either **Use work manager thread pool size** to use the default size or **Intertransaction cache size** to enable the input field so that you can specify a value.

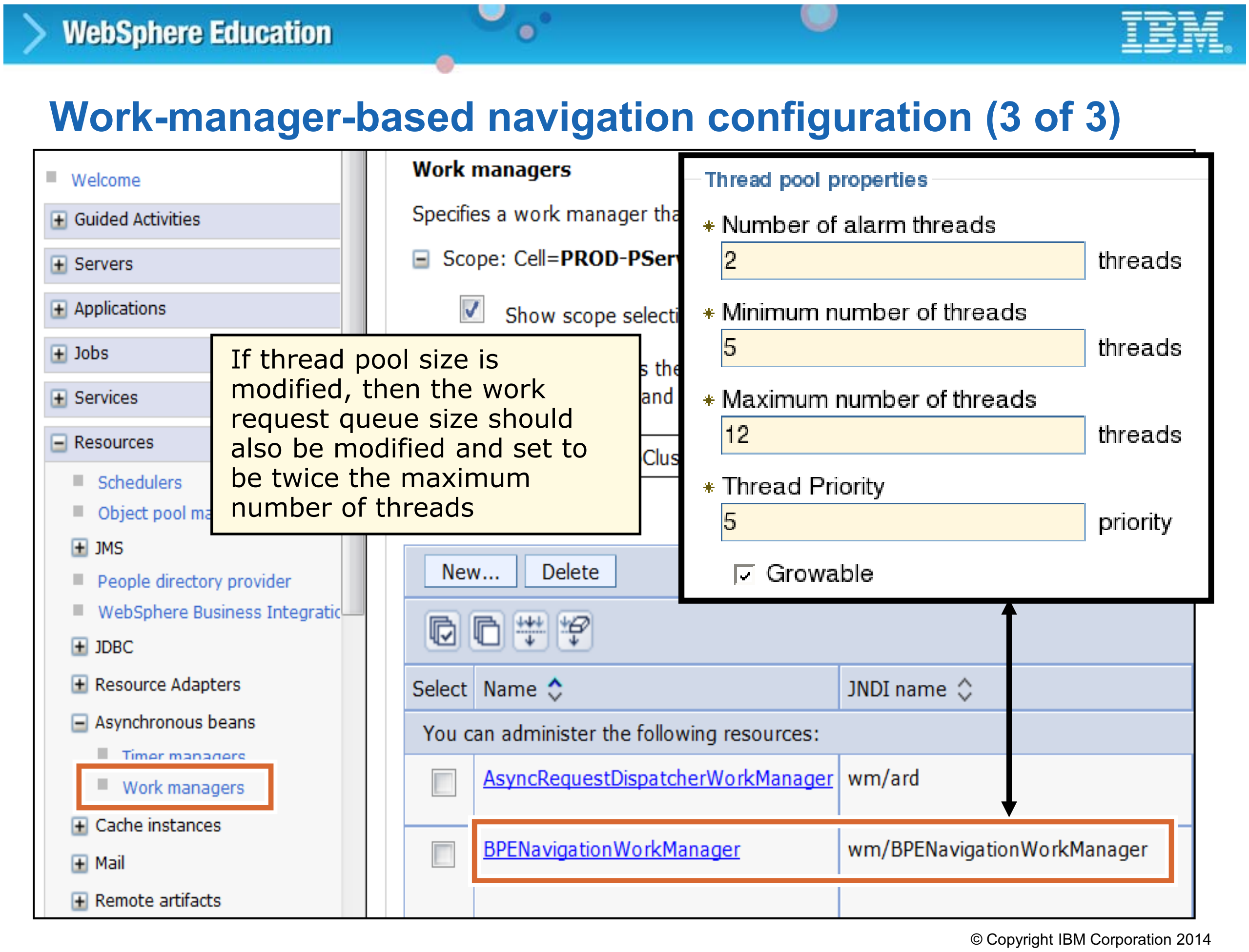


Figure 9-34. Work-manager-based navigation configuration (3 of 3) WB868 / ZB8681.0

Notes:

WebSphere Education

Tuning for JMS-based navigation (1 of 2)

If JMS-based navigation is configured, the following resources need to be optimized for efficient navigation of business processes:

- Activation specification **`BPEInternalActivationSpec`**
  - The *maximum concurrent endpoints* parameter specifies the parallelism that is used for process navigation across all process instances
  - Increase the value of this parameter to increase the number of business processes that are executed concurrently
- JMS connection factory **`BPECFC`**
  - Set the connection pool size to the number of threads in the **`BPEInternalActivationSpec`** + 10%

Figure 9-35. Tuning for JMS-based navigation (1 of 2) WB868 / ZB8681.0

Notes:

Set the connection pool size to the number of threads in the BPEInternalActivationSpec plus 10%. Find this resource in the administrative console by clicking **Resources > JMS > Connection factories > BPECFC > Connection pool properties**. This connection factory is also used when work-manager-based navigation is in use, but only for error situations or if the server is highly overloaded.

WebSphere Education

Tuning for JMS-based navigation (2 of 2)

J2C activation specifications > eis/BPEInternalActivationSpec > **Custom properties**

Use this page to specify custom properties that your enterprise information system (EIS) requires fo
resource providers and resource factories that you configure. For example, most database vendors r
additional custom properties for data sources that access the database.

Preferences

Name	Value	Description	Required
You can administer the following resources:			
destinationType	javax.jms.Queue		true
acknowledgeMode	Auto-acknowledge		false
alwaysActivateAllMDBs	true		false
AutoStopSequentialMessageFailure	0		false
busName	BPM.PServer_DE.Bus		false
clientId			false
destination			false
durableSubscriptionHome			false
failingMessageDelay	0		false
maxBatchSize	1		false
maxConcurrency	10		false

Figure 9-36. Tuning for JMS-based navigation (2 of 2) WB868 / ZB8681.0

Notes:

WebSphere Education

Best practices: Business Flow Manager

- Use microflows instead of long-running processes if possible
  - If targets respond quickly, use short-running processes
  - Use long-running processes only if a process includes human tasks, and targets respond slowly
  - Carefully plan long-running process use, and consider impacts to capacity plan
- Avoid overusing the process engine
  - A process and its individual steps should have a business significance
  - Use normal implementation techniques such as servlets and plain old Java objects (POJO) for logic that has no business significance

Figure 9-37. Best practices: Business Flow Manager WB868 / ZB8681.0

Notes:

Microflows are designed to orchestrate business processes that involve back-end facilities that respond relatively quickly. As a result, the microflow facilities can use techniques that are lightweight, including executing all activities in a flow in a single transaction.

Long-running processes are designed to orchestrate business processes that involve back-end facilities that respond relatively slowly. As a result, microflows must execute in such a way as to not tie up system resources for long periods of time. For example, each activity in a microflow is executed in its own transaction, and a transaction is not held while waiting on a long-running, back-end request. A good example of a long-running, back-end request is one that requires human interaction.

Since microflows can use implementation techniques that are more efficient than long-running processes, business processes that are implemented as microflows have better performance than business processes implemented as long-running processes.

The process engine should be used to implement processes that have business significance. If a function does not have business significance, consider the use of standard Java Platform, Enterprise Edition programming techniques, including servlets and EJBs.

WebSphere Education

Best practices: Transactional behavior and interaction style

- Set **transactional behavior** property to **participate** when possible
  - This setting allows the current activity to run in the transaction context of the previous activity
  - The default setting for **transactional behavior** is **commit after**, which causes each activity to run in its own transaction
- When calling an SCA component from a long-running process, set **preferred interaction style** property to synchronous
  - If a long-running process calls a component that is configured for **any** interaction style, SCA asynchronous is used

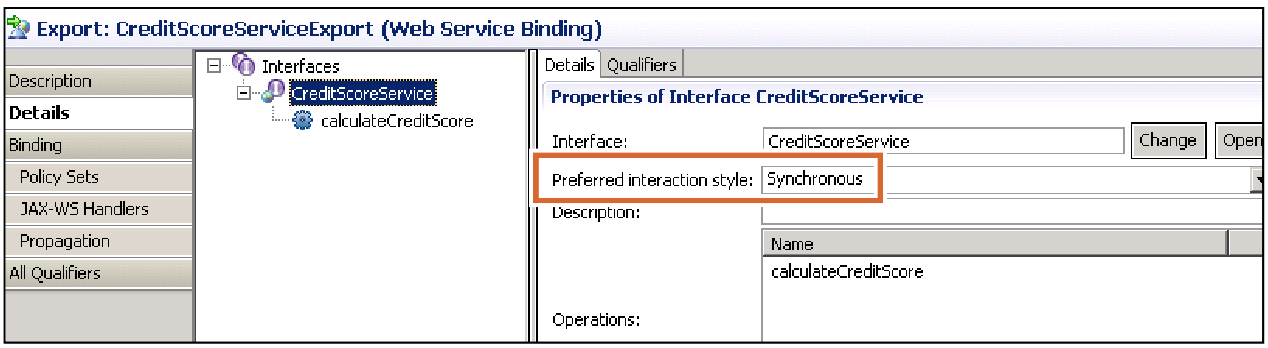


Figure 9-38. Best practices: Transactional behavior and interaction style WB868 / ZB8681.0

Notes:

When designing long-running business process components, ensure that callers of a two-way (request/response) interface do not use synchronous semantics, as it ties up the caller's resources (for example, threads or transactions) until the process completes. Instead, such processes should be invoked either asynchronously, or through a "one-way" synchronous call, where no response is expected.

In addition, calling a two-way interface of a long-running business process synchronously introduces difficulties when exceptions occur. Suppose that a non-interruptible process calls a long-running process by using the two-way request/response semantics, and the server fails after the long-running process completes, but before the caller's transaction is committed.

If a persistent message starts the caller, upon server restart the caller's transaction is rolled back and then tried again. However, the result of the execution of the long-running process on the server is not rolled back, since it was committed before the server failure. As a result, the long-running process on the server is executed twice.

This duplication causes functional problems in the application unless it is corrected manually.

If a persistent message does not start the caller, and the response of the long-running process was not submitted yet, it ends in the failed event queue.

WebSphere Education

Best practices: Cache size

Configure statement cache for long-running processes:

- Long-running processes use the data source for the database well beyond the default capacity of the data source's statement cache
- If you are using long-running processes, the size of the data source statement cache should be increased
- Make sure that the prepared statement cache for each data source is sufficiently large
  - For example, set the data source prepared statement cache size to at least 300

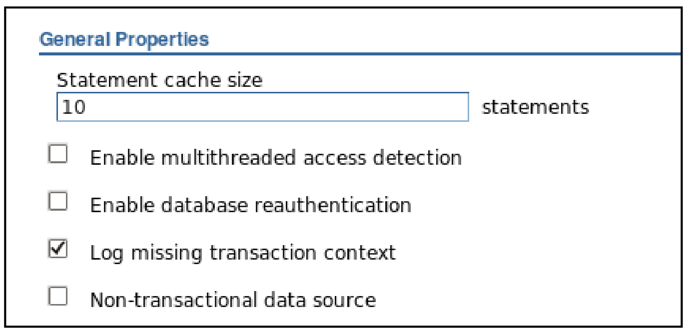


Figure 9-39. Best practices: Cache size WB868 / ZB8681.0

Notes:

Prepared statements that are cached complete faster than those statements not cached. However, for Oracle databases, each prepared statement can take a significant amount of Java heap space (140 KB per statement for some cases that are seen in Business Process Manager solutions).

WebSphere Education

Best practices: Logging

- Use the **audit logging** property for business processes ***only if*** you need to log events in the database
  - Audit logging property can be set at the activity or process level
  - If this property is set at the process level, all activities inherit the setting
- For long-running processes, disable the "**Enable persistence and queries of business-relevant data**" flag
  - Enabling this flag causes details and history of the activity execution to be stored in the database
  - This setting should be used only if the specific information needs to be retrieved later

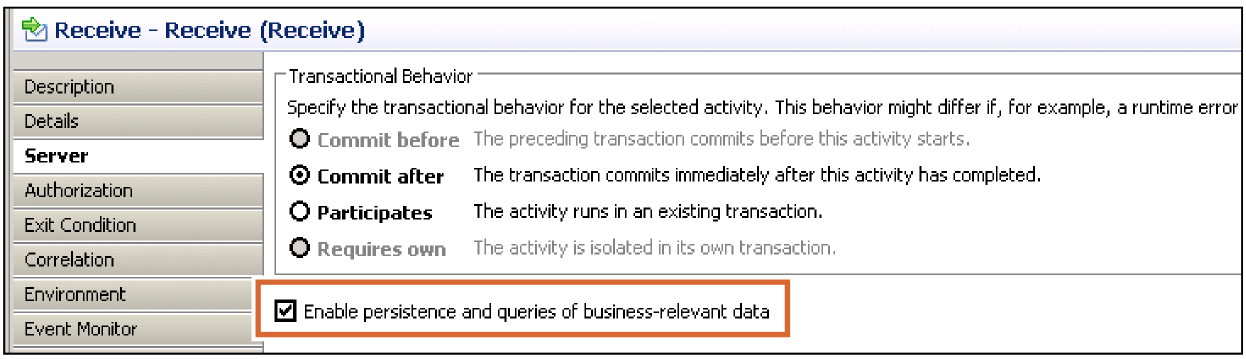


Figure 9-40. Best practices: Logging WB868 / ZB8681.0

Notes:

Enable persistence and queries of business-relevant data:

When this check box is enabled, the details and history of the processing of this activity are stored in the runtime environment for later reference. When it is clear, then no data is kept.

WebSphere Education

Best practices: Variables and business objects

- Considerations to minimize the number and size of BPEL variables and business objects: use as few variables as possible and minimize the size and the number of business objects used
  – Each commit saves modified variables to the database, and with multiple variables or large business objects this saving of variables becomes costly in long-running processes
  – Smaller business objects are more efficient to process when emitting monitor events
- Use transformations (such as maps and assigns) to produce smaller business objects by only mapping fields necessary
- Specify variables as "**Data type variables**"
  – This specification improves runtime performance

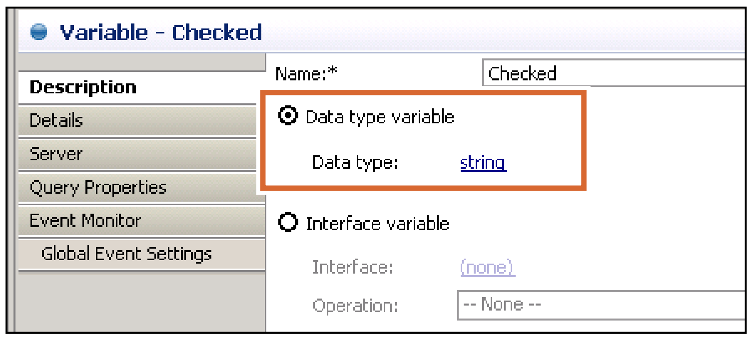


Figure 9-41. Best practices: Variables and business objects WB868 / ZB8681.0

Notes:

Some considerations are as follows:

- Use as few variables as possible and minimize the size and the number of business objects (BOs) used. In long-running processes, each commit saves modified variables to the database (to save context), and with multiple variables or large BOs these save actions become costly. Smaller BOs are also more efficient to process when emitting monitor events.
- Specify variables as data type variables, which improves runtime performance.
- Use transformations (maps or assigns) to produce smaller BOs by mapping only fields that are necessary for the business logic.

9.4. Optimizing work item creation

WebSphere Education

Optimizing work item creation

9.1

Figure 9-42. Optimizing work item creation WB868 / ZB8681.0

Notes:

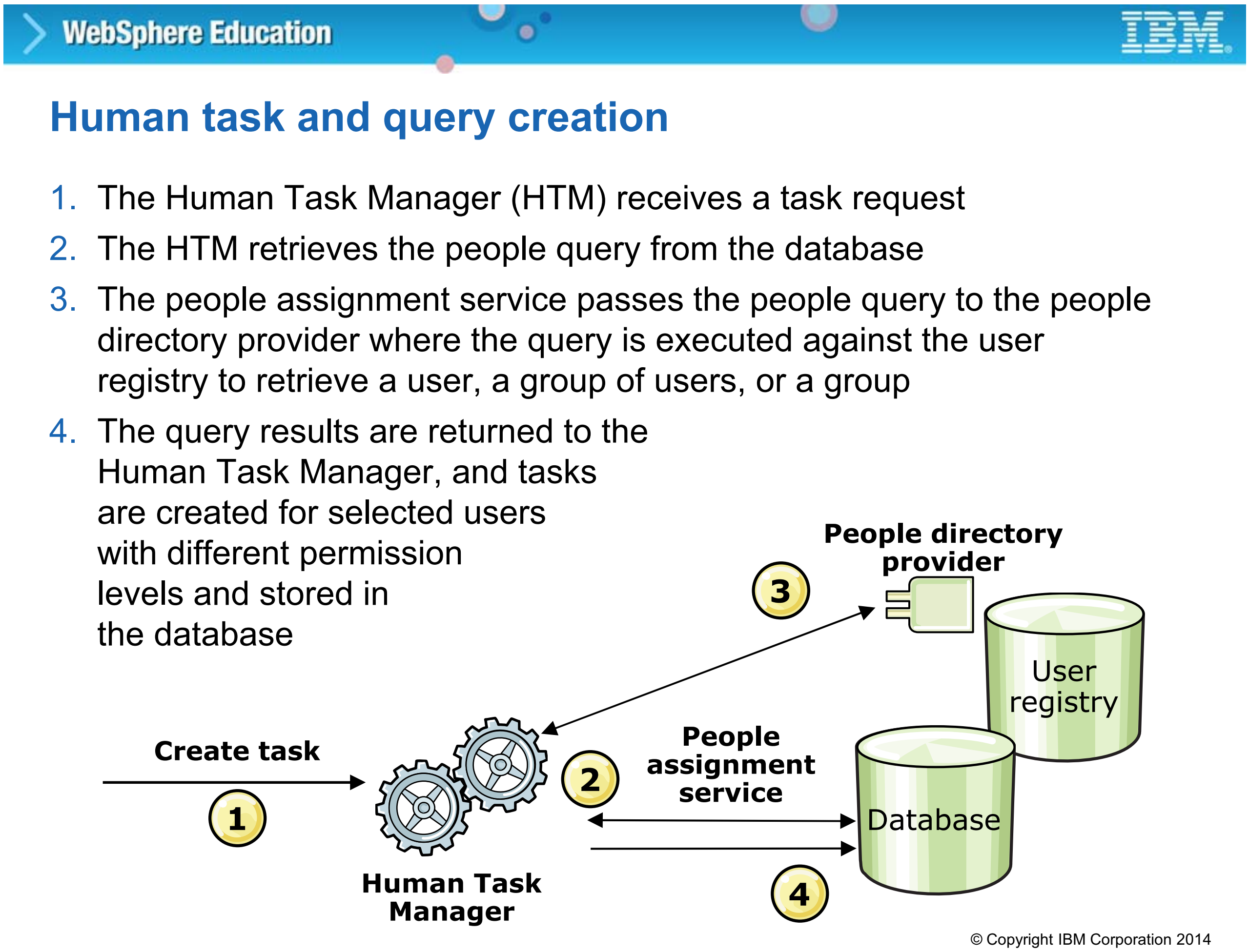


Figure 9-43. Human task and query creation WB868 / ZB8681.0

Notes:

Authorization is the mechanism by which certain people are enabled to do selected actions on task templates, task instances, and escalations. Authorization roles are used to define sets of actions available to specific roles. People can be assigned to system-level roles by using Java EE mechanisms, or to task instance roles by using people assignment criteria.

People assignment criteria are constructs that are used in the task model to identify sets of people that can be assigned to an instance-based authorization role. At run time, the people resolution uses the people assignment criteria to retrieve the user IDs and other user information from the people directory, for example, for composing emails. The runtime also uses people assignment criteria when task models are created programmatically.

People query example

- In this example, the “Potential owner” role is assigned to the “Approvers” group
- Both John and Jane, as members of the approvers group, can claim the task
  - One work item is created for the group
  - When the work item is claimed, the database is populated with the user name
- A people assignment post-processor plug-in is available to further modify the query result after it is returned to the Human Task Manager

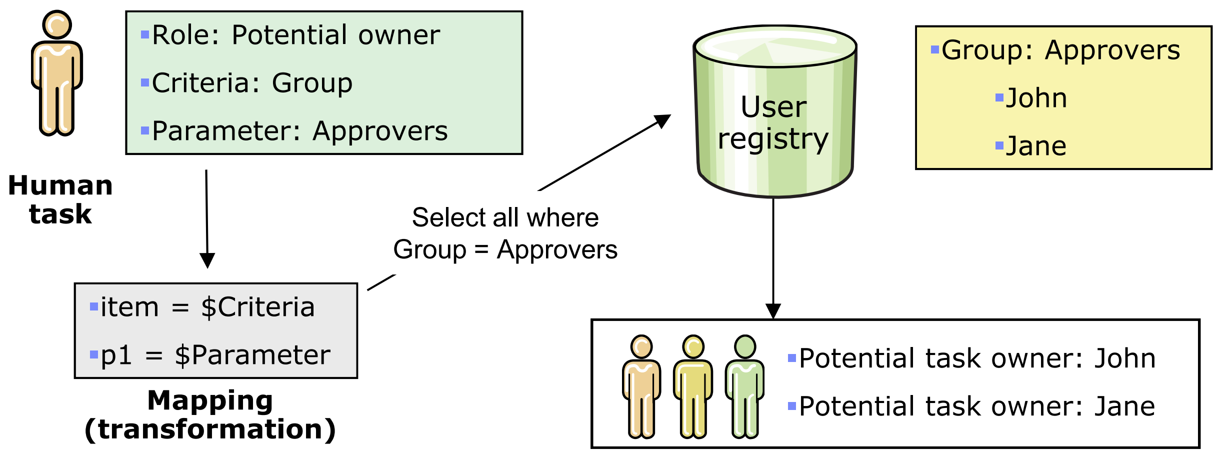


Figure 9-44. People query example WB868 / ZB8681.0

Notes:

A key aspect of the Human Task Manager is the ability to assign tasks to personnel who are defined in a user registry of an organization. Rapidly changing organizational structures are a fact of life. It is important to be able to manage the interface into the user registry in a way that it is flexible enough to respond to change. To assign a task, you define parameters that the HTM uses to dynamically look up authorized personnel at run time. Any changes to the structure of an organization are automatically picked up as soon as the changes are reflected in the user registry.

WebSphere Education

People assignment

- Users and groups are assigned to an instance-based role by using an assignment criteria (formerly called a staff verb) which is resolved to a specific people query (formerly a staff query) for a specific directory
- The assignment criteria defines the members of the role and is narrowed by using a set of parameters, creating an authorization rule
- People queries can be tested in the human task editor

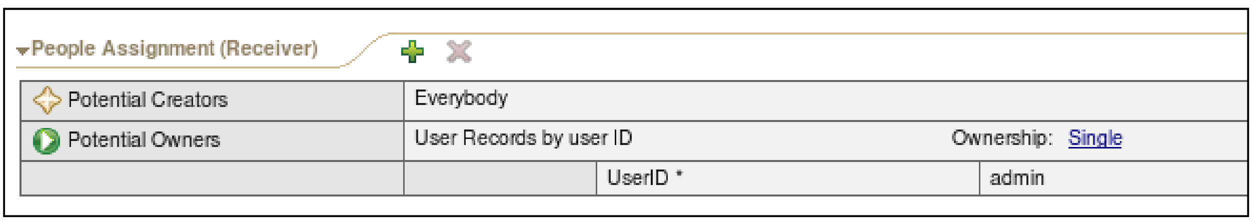


Figure 9-45. People assignment WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Translating people assignments into work items

- Assigning multiple individuals as potential owners, or all group members as potential owners:
  - There can be many work items
  - Requires ***one table row per potential owner*** in the work item view
  - More rows are required for other roles such as administrator
- Simple math shows table growth:
  - Twenty thousand task instances are created per day
  - Average life of five days
  - With two hundred users per role and three assigned roles:
    20,000 * 5 * 200 * 3 = **60,000,000 work item rows**
- Work items grow quickly per task:
  - Every task has one or more administrators (always one implicit administrator even if none is assigned)
  - Every task has an originator work item (tracks who created the instance)
  - Every claimed task has a “claimed” work item

Figure 9-46. Translating people assignments into work items WB868 / ZB8681.0

Notes:

WebSphere Education

Query result size in LDAPTransformation.xsl

- Each plug-in provider has an XSL transformation file that creates a registry-specific query from the people assignment that is identified in the role properties in Integration Designer
- The default threshold setting returns 1,000,000 people from a group
- Adjusting the threshold might be necessary for the performance

```
...
<!-- Begin global variables, adapt if needed -->
 <xsl:variable name="Threshold">1000000</xsl:variable>

 <xsl:variable name="DefaultPersonClass">inetOrgPerson</xsl:variable>
 <xsl:variable name="DefaultUserIdAttribute">uid</xsl:variable>
 <xsl:variable name="DefaultMailAttribute">mail</xsl:variable>
 <xsl:variable
        name="DefaultLocaleAttribute">preferredLanguage</xsl:variable>
 <xsl:variable name="DefaultManagerAttribute">manager</xsl:variable>
...
```


Figure 9-47. Query result size in LDAPTransformation.xsl WB868 / ZB8681.0

Notes:

WebSphere Education

Effect of people assignment criteria on number of work items

- The chosen people assignment criteria has an impact upon how many work items are created, and thus, impacts system performance
- Use **Group** work item for large groups instead of individual work items for **Group members**

Figure 9-48. Effect of people assignment criteria on number of work items WB868 / ZB8681.0

Notes:

WebSphere Education

Reducing collision between potential owners

- If concurrent access is high, limit the number of users who can access a particular human task
- Avoid unnecessary human task queries from clients, by using intelligent claim mechanisms
  - Try to claim another item from the list if the first claim is unsuccessful
  - Always claim a random human task
- Reduce the number of potential owners for the task
  - For example, assign the task to a group with fewer members
- Limit the size of the task assignment list by specifying a threshold on the query that is used to retrieve the role members in **`transformation.xsl`**

```
<!-- Begin global variables, adapt if needed -->
 <xsl:variable name="Threshold">20</xsl:variable>
 <xsl:variable name="DefaultPersonClass">inetOrgPerson</xsl:variable>
 <xsl:variable name="DefaultUserIdAttribute">uid</xsl:variable>
 <xsl:variable name="DefaultMailAttribute">mail</xsl:variable>
```


Figure 9-49. Reducing collision between potential owners WB868 / ZB8681.0

Notes:

WebSphere Education

Storage of query results

- During staff resolution, the Human Task Manager (HTM) executes a staff query:
  - Staff query results are cached during run time
  - It might become invalid due to changes in the organizational directory (staff repository)
- **`RETRIEVED_USER_T`**: Retrieved user table
  - Provides a query cache by storing the list of users that a staff query returns
  - Storage of query results allows Human Task Manager to reuse earlier staff queries
  - New task instances simply reference the retrieved user table
- Caching staff query results reduces load on configured user directory
  - Especially when HTM is configured to use a production organizational directory
- Use of dynamic parameter values (late binding) in staff queries prevents query results from being reused
  - The user repository (LDAP directory) is queried for every new task instance
  - Retrieved user table grows as each task instance runs the staff query and stores its results in the retrieved user table

Figure 9-50. Storage of query results WB868 / ZB8681.0

Notes:

WebSphere Education

Scheduling a staff query refresh

- Automatic staff query refresh specifies the time after which a staff query result expires and should be refreshed
- Refresh applies organizational changes to HTM role assignments
  - Scheduled by using timer-controlled daemon
  - Staff query post-processing plug-in is called during staff query refresh
- Staff query refresh schedule
  - Specifies how often staff query refresh daemon should run; frequency affects performance
  - Specified by using CRON syntax → **Default:** `0 0 1 * * ?` (every day at 1 a.m.)

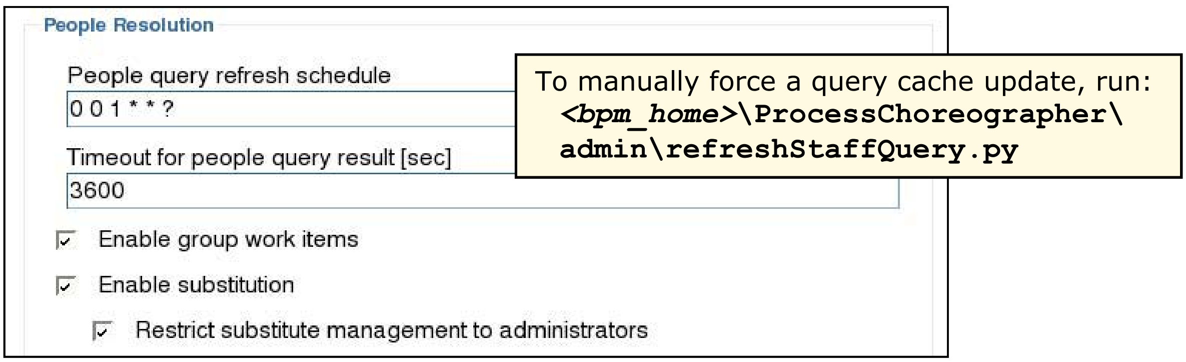


Figure 9-51. Scheduling a staff query refresh WB868 / ZB8681.0

Notes:

Use the `refreshStaffQuery.py` administrative script to refresh people queries because the query results are static.

The following conditions must be met:

- Run the script in the connected mode; that is, do not use the wsadmin `-conntype none` option.
- At least one cluster member must be running.
- If your user ID does not have operator authority, include the wsadmin `-user` and `-password` options to specify a user ID that has operator authority.
- If you are not working with the default profile, use the wsadmin `-profileName profile` option to specify the profile.

WebSphere Education

Query tables in Business Process Choreographer

- Instance data for long-running business processes and human tasks are stored persistently in the database and are accessible by queries
- Query tables support task and process list queries on data that is contained in the Business Process Choreographer database schema
  - Human task data and business process data that Business Process Choreographer manages, and external business data
  - Query tables provide an abstraction on the data of Business Process Choreographer that client applications can use
  - Query table definitions are deployed on Business Process Choreographer containers, and are accessible by using the query table API
- There are three types of query tables:
  - Predefined query tables
  - Supplemental query tables
  - Composite query tables

Figure 9-52. Query tables in Business Process Choreographer WB868 / ZB8681.0

Notes:

Query tables support task and process list queries on data that is contained in the Business Process Choreographer database schema. This data includes human task data and BPEL process data that Business Process Choreographer manages, and external business data. Query tables provide an abstraction on the data of Business Process Choreographer that client applications can use. In this way, client applications become independent of the actual implementation of the query table. Query table definitions are deployed on Business Process Choreographer containers, and are accessible by using the query table API.

Predefined query tables provide access to the data in the Business Process Choreographer database. They are the query table representation of the corresponding predefined Business Process Choreographer database views, such as the TASK view or the PROCESS_INSTANCE view. These predefined query tables enhance the functionality and performance of the predefined database views because they are optimized for running process and task list queries.

The predefined query tables can be queried directly by using the query table API. When you access the tables by using the query table API, you are offered more options for configuration than when you use the query API.

Supplemental query tables in Business Process Choreographer expose to the query table API business data that Business Process Choreographer does not manage. With supplemental query tables, this external data can be used with data from the predefined query tables when retrieving BPEL process instance information or human task information.

Composite query tables in Business Process Choreographer do not have a specific representation of data in the database; they consist of a combination of data from related predefined and supplemental query tables. Use a composite query table to retrieve the information for a process instance list or task list, such as My To Dos.

WebSphere Education

Query performance overview

- Query tables enhance the predefined database views and the existing query interfaces of Business Process Choreographer, and they:
  - Are optimized for running process and task list queries, by using performance optimized access patterns
  - Simplify and consolidate access to the information needed
  - Allow for the fine-grained configuration of authorization and filter options
- Parameters with impact on query performance
  - Number of retrieved rows
  - Number of rows to be evaluated; if order by is specified, the database must sort
  - Number of joins between database tables or views
  - Complexity of selection criteria (where)
- Database techniques for improving query performance
  - Indexes
  - Statistics on tables, columns, and indexes
  - Materialized query tables
  - Threshold on the number of rows to be retrieved

Figure 9-53. Query performance overview WB868 / ZB8681.0

Notes:

Query tables introduce a clean programming model for developing client applications that retrieve lists of human tasks and BPEL processes in Business Process Choreographer. Using query tables improves the performance. Information is provided about the query table API parameters and other factors that affect the performance.

WebSphere Education IBM

Unit summary

Having completed this unit, you should be able to:

- Describe the difference between long-running and microflow (short-running) business processes
- Explain the runtime architecture of long-running business processes
- Explain how events are persisted in long-running business processes
- Calculate a realistic database table growth rate for each solution, and consult with database administrators to establish a database strategy for sizing
- Identify the key best practices for the Business Process Choreographer
- Describe the runtime architecture of the Human Task Manager
- List and describe methods to optimize work item creation

Figure 9-54. Unit summary WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Checkpoint questions

1. True or false: Both microflows and long-running business processes are constructed of a set of stratified transactions.

2. True or false: The use of JMS-based navigation is the preferred navigation mode since it provides some performance optimization and is easy to configure.

3. On average, you are expecting 10,000 requests per day with life span of 20 days. The retention period is 30 days. How many in-flight process instances do you expect in the database?
   a) (10,000 * 20) = 200,000 in-flight instances
   b) (10,000 * 30) = 300,000 in-flight instances
   c) (10,000 * 20 * 30) = 6,000,000 in-flight instances
   d) (10,000 * 20) + (10,000*30) = 200,000 + 300,000 = 500,000 in-flight instances

Figure 9-55. Checkpoint questions WB868 / ZB8681.0

Notes:

Write your answers here:

1.

2.

3.

WebSphere Education

Checkpoint answers

1. False. Microflows run in a single transaction, and single-threaded.

2. False. Work-manager-based navigation is the preferred navigation mode, and it optimizes the performance.

3. d

Figure 9-56. Checkpoint answers WB868 / ZB8681.0

Notes:

Unit 10. Performance tools

What this unit is about

This unit describes performance data and performance data analysis tools.

What you should be able to do

After completing this unit, you should be able to:

- Describe important performance data for monitoring Business Process Manager environments
- Identify useful performance tools for monitoring performance data
- Use the Tivoli Performance Viewer tool to monitor the runtime performance of an application that is running
- Describe the performance tools available in the IBM Support Assistant

How you will check your progress

- Checkpoint
- Lab exercise

References

IBM Business Process Manager V8.5 IBM Knowledge Center

`http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/ditamaps/ic-homepage-bpm.html`

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Describe important performance data for monitoring Business Process Manager environments
- Identify useful performance tools for monitoring performance data
- Use the Tivoli Performance Viewer tool to monitor the runtime performance of an application that is running
- Describe the performance tools available in the IBM Support Assistant

Figure 10-1. Unit objectives WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Topics

- Performance data
- Operating system tools
- Administrative console tools
- Business Process Manager tools
- IBM Support Assistant overview
- IBM Support Assistant tools

Figure 10-2. Topics WB868 / ZB8681.0

Notes:

10.1.Performance data

Performance data

Figure 10-3. Performance data WB868 / ZB8681.0

Notes:

WebSphere Education IBM

High-level performance data areas

- Hardware
- Operating system
- JVM
- Application servers
- Messaging
- Databases

Figure 10-4. High-level performance data areas WB868 / ZB8681.0

Notes:

For each system in your topology, including front-end and back-end servers such as web servers and database servers, you should monitor various areas on an ongoing basis.

WebSphere Education

Hardware performance data

- Number of physical processors
- Amount of physical memory
- Total disk space, free disk space, and temp disk space
- I/O channel utilization
- Network interface card utilization and configuration

Figure 10-5. Hardware performance data WB868 / ZB8681.0

Notes:

The hardware resources highly impact BPM performance. It is important to select an appropriate hardware resource configuration to achieve high performance capability.

WebSphere Education

Operating system performance data

- System configuration
  - Logical processors per physical processor
- CPU utilization
  - Whole partition
  - Per process
  - Per logical CPU
- Physical memory utilization
  - Paging
  - Swap space utilization

Figure 10-6. Operating system performance data WB868 / ZB8681.0

Notes:

Normally, for Process Center, one core CPU can support 5 - 10 Process Designer connections; for Process Server or Process Center Server, one core CPU can support 50 - 100 current users. However, the CPU core utilization depends on the process and service implementation. If the implementation is intensive CPU usage, such as complex coach view or complex result calculation, more CPU cores must be added.

JVM performance data

- Heap utilization as in current heap size versus maximum heap size
  - Current nursery size versus maximum nursery size
  - Current tenured size versus maximum tenured size
- Live object utilization
  - Current size of live versus current heap size
  - Current nursery that is used versus current nursery size
  - Recent minimum tenured used versus current tenured size
- Garbage collection (GC) cost
  - Time in GC versus time between GCs
- Level of heap fragmentation, as in “free before GC”
- Average and peak size of requested objects

Figure 10-7. JVM performance data WB868 / ZB8681.0

Notes:

You must ensure that the heap sizes are large enough that garbage collections do not occur too often. For example, suppose that large complex business objects are used in your BPM implementation, and meanwhile on the high workload environment, the resources such as thread pool, connection pool, and others, occupy more memory. Generally, BPM servers run most efficiently on a 64-bit JVM instance because of the much larger amount of memory that is accessible in this mode.

WebSphere Education

Server performance data

- Data source utilization levels
  - Current number of database connections per data source versus maximum number of database connections per data source
- Thread pool utilization levels
  - Current number of active threads per thread pool versus maximum number of threads per thread pool
- Source of application server performance data
  - Specialized tools are required to monitor data sources, caching, and thread pools
  - You can use the Tivoli Performance Viewer
- Time synchronization between servers
  - Verify system clocks between your servers and database servers

Figure 10-8. Server performance data WB868 / ZB8681.0

Notes:

The correct timer behavior highly depends on the time synchronized. Also, the Event Manager scheduler uses all the time stamps. The heartbeat expirations and the task scheduled times are interpreted relative to the system clock for the database machine. Keeping system clocks in sync is a good idea.

WebSphere Education IBM

Messaging performance data

- Names of queues
- Location (server) for queues
- Queue utilization levels
  - Current queue depth versus maximum queue depth
- Average and peak message sizes
- Current and peak queue depths
  - Queue depth trends
- Message arrival rates
- Sources of messaging performance data
  - Administrative console
  - Tivoli Performance Viewer (SIB service)

Figure 10-9. Messaging performance data WB868 / ZB8681.0

Notes:

WebSphere Education

Database performance data

- Names (and counts) of databases and schemas
- CPU usage, disk space, I/O, cache size
- Database connection utilization levels
  - Current number of database connections versus maximum number of database connections
- Longest running SQL queries
- Buffer utilization levels
- Sources of database performance data
  - Specialized tools are required for database performance monitoring
  - Typically provided by the database product

Figure 10-10. Database performance data WB868 / ZB8681.0

Notes:

Examples of conditions to monitor

- The JVM heap reaches 70% of maximum heap and stays there for over 5 minutes
  - Typically characteristic of an impending out-of-memory error
- The CPU remains at over 90% for 5 minutes
- The hard disk space capacity increases by 20% in less than 5 minutes
  - Rapid consumption of hard disk space typically indicates that an issue is occurring that produces numerous log file entries is using up disk space
- Garbage collection happens frequently without much success in freeing up memory
  - Usually means that there is more demand for memory than is available, which creates conditions for a possible out-of-memory error

Figure 10-11. Examples of conditions to monitor WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Performance monitoring tools

- IBM offers high-end tools for real-time problem determination
  - IBM Tivoli for Composite Application Management suite
  - Other non-IBM products such as CA Wily Introscope
- Most tuning and debugging can be performed with free tools
  - Are already available or can be installed easily
  - Focus on the problem at hand instead of the tools
- Many tuning and problem determination tools can be downloaded with the IBM Support Assistant

Figure 10-12. Performance monitoring tools WB868 / ZB8681.0

Notes:

10.2. Operating system tools

WebSphere Education

Operating system tools

Figure 10-13. Operating system tools WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Analysis and monitoring tools (1 of 3)

- `vmstat` is a lightweight monitoring tool for key system resources and is especially useful to monitor CPU usage
- Available on all major UNIX operating systems and Linux distributions
  - Use **`mpstat`** on multi-processor AIX systems to monitor all logical processor statistics
- Identify various key metrics
  - CPU, memory, paging, context switching, and system calls
- Command format:
  - **`vmstat [-options] [interval [count]]`**
  - Where **`interval`** is the number of seconds between measurements and **`count`** is the number of measurements performed

Figure 10-14. Analysis and monitoring tools (1 of 3) WB868 / ZB8681.0

Notes:

CPU:

- Is the system quiet before the test begins?
- How much CPU does the test use?
- Is it spent productively (mostly in the “User” component of CPU)?

Memory:

- Watch the free list for sufficient memory.
- Use ps ev for more detailed process memory reports.

Paging:

- The JVM must stay in memory during run time.

Context switching and system calls:

- High numbers indicate issues with excessive disk I/O or network problems.

WebSphere Education IBM

Analysis and monitoring tools (2 of 3)

- To print system resource usage every 10 seconds, run the following command:

```
# vmstat 10
procs -----------memory---------- ---swap-- -----io---- -system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 4  0  99864  60000  34404 451260    0    2    40    32  269  927  6  7 86  1  0
 0  0  99848  60572  34416 451524    0    0     0    87  264 1514 17  9 74  1  0
 0  0  99828  60356  34428 451728    6    0     6    50  263 1479 13  6 81  1  0
10  0 103020  51880  34372 446612    0  302     0   376  267 1259 12 35 53  0  0
 2  0 111644  53328  26968 443712    6  844     7   880  265 1396 24 72  4  0  0
40  0 111636  53884  26992 442916    0    0     0    80  262 1460 17 10 69  5  0
 0  0 111612  52724  27008 443412    3    0     3    74  264 1576 13  7 80  0  0
 5  0 111568  51532  27020 444420    6    0     6    70  266 1614 20 17 63  0  0
```

- Output from a Linux system

Figure 10-15. Analysis and monitoring tools (2 of 3) WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Analysis and monitoring tools (3 of 3)

- `netstat` is a lightweight monitoring tool that displays network connection, routing tables, and a number of network interface and protocol statistics
- `iostat` is used for monitoring system input/output device loading by observing the time that the devices are active in relation to their average transfer rates
  - Generates reports on CPU utilization, device utilization, and the network file system report
- `ps` provides you a snapshot of the current processes that are running
  - Uses numerous options to retrieve specific data
  - Provides CPU, memory, and virtual memory usage and other details

Figure 10-16. Analysis and monitoring tools (3 of 3) WB868 / ZB8681.0

Notes:

`netstat` (**net**work **stat**istics) is a command-line tool that displays network connections (both incoming and outgoing), routing tables, and a number of network interface (network interface controller or software-defined network interface) and network protocol statistics.

The `iostat` command is used for monitoring system input/output device loading by observing the time that the devices are active in relation to their average transfer rates. The `iostat` command generates reports that can be used to change system configuration to better balance the input/output load between physical disks.

`ps program` (short for “**p**rocess **s**tatus”) displays the currently running processes. A related UNIX utility that is named **top** provides a real-time view of the running processes.

WebSphere Education IBM.

Analysis and monitoring tools: nmon

- Free and powerful tool to monitor AIX and Linux systems
- Can be run in interactive mode and non-interactive mode
- Monitors most operating system resources that include:
  - CPU
  - Memory
  - Network I/O
  - Disk I/O
- Useful on platforms that support hardware virtualization
  - Monitors LPAR resources on AIX System p machines
- Non-interactive mode probably most useful during performance test runs
  - Monitors and logs to `.csv` file
  - Use Excel Analyzer spreadsheet to import the `.csv` file and generate graphs

Figure 10-17. Analysis and monitoring tools: nmon WB868 / ZB8681.0

Notes:

This nmon tool gives you a huge amount of information all on one screen. Even though IBM does not officially support the tool and you must use it at your own risk, you can get a wealth of performance statistics.

The nmon tool is designed for AIX and Linux performance specialists to use for monitoring and analyzing performance data.

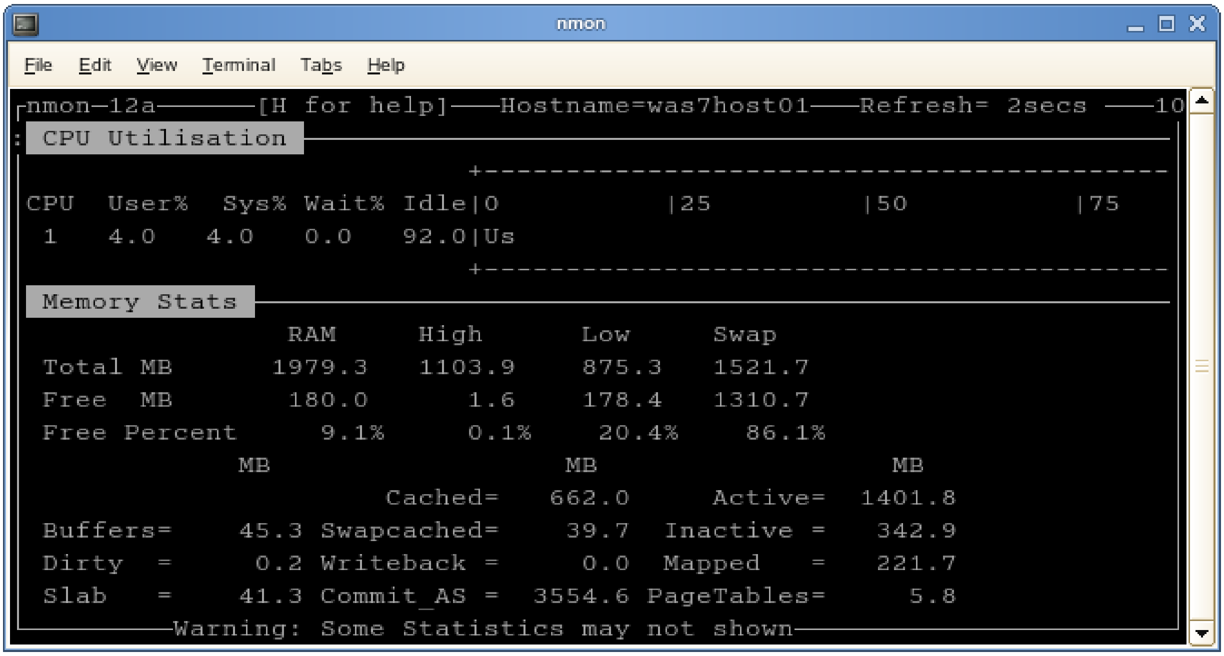
See more information about this tool at:

`http://www.ibm.com/developerworks/aix/library/au-analyze_aix/index.html`

WebSphere Education IBM

nmon output of CPU and memory statistics

- `nmon cm` provides CPU utilization and memory statistics



```
nmon
File  Edit  View  Terminal  Tabs  Help
┌nmon─12a──────[H for help]───Hostname=was7host01───Refresh= 2secs ───10
│ CPU Utilisation ─────────────────────────────────────────────────────────
│                            +---------------------------------------------
│CPU  User%  Sys% Wait% Idle|0               |25           |50            |75
│ 1   4.0   4.0   0.0   92.0|Us
│                            +---------------------------------------------
│ Memory Stats ────────────────────────────────────────────────────────────
│                RAM      High       Low      Swap
│ Total MB      1979.3    1103.9     875.3    1521.7
│ Free  MB       180.0       1.6     178.4    1310.7
│ Free Percent     9.1%      0.1%     20.4%     86.1%
│            MB                  MB                  MB
│                     Cached=    662.0      Active=  1401.8
│ Buffers=     45.3 Swapcached=    39.7  Inactive =   342.9
│ Dirty  =      0.2 Writeback =     0.0  Mapped   =   221.7
│ Slab   =     41.3 Commit_AS =  3554.6 PageTables=     5.8
└─────────Warning: Some Statistics may not shown──────────────────────
```


Figure 10-18. nmon output of CPU and memory statistics WB868 / ZB8681.0

Notes:

The nmon tool is helpful in presenting all the important performance tuning information on one screen and dynamically updating it. This efficient tool works on any dumb screen, telnet session, or even a dial-up line. In addition, it does not consume many CPU cycles, usually below two percent. On newer machines, CPU usage is well below one percent.

Data is displayed on the screen and updated once every 2 seconds, by using a dumb screen. However, you can easily change this interval to a longer or shorter time period. If you stretch the window and display the data on X Window System, VNC, PuTTY, or similar, the nmon tool can output a great deal of information in one place.

The nmon tool can also capture the same data to a text file for later analysis and graphing for reports. The output is in a spreadsheet format (`.csv`).

10.3. Administrative tools

Administrative tools

Figure 10-19. Administrative tools WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Performance Monitoring Infrastructure (PMI)

- When PMI is enabled for an application server, monitoring of individual WebSphere components can be enabled or disabled dynamically
- WebSphere PMI provides four predefined statistic sets that can be used to enable a set of statistics

Statistics sets	Description
Basic	• Statistics that are specified in Java EE specification, as well as top statistics like CPU usage and live HTTP sessions, are enabled • This set is enabled by default and provides basic performance data about runtime and application components (performance cost of up to 2%)
Extended	Basic statistics plus key statistics from various WebSphere components like workload management (WLM) and dynamic caching (performance cost of up to 3%)
All	All statistics are enabled (performance cost of up to 6%)
Custom	Allows the user to enable or disable statistics selectively

- PMI levels can be set via the administrative console or the `wsadmin` command

Figure 10-20. Performance Monitoring Infrastructure (PMI) WB868 / ZB8681.0

Notes:

Performance measurements are available for service component event points, and are processed through the Performance Monitoring Infrastructure. You configure a server to gather performance metrics from service component event points. You can also collect performance statistics that are specific to Service Component Architecture, directly from service invocations of applications.

Whether you are tuning service components for optimal efficiency or diagnosing a poor performance, it is important to understand how the various runtime and application resources are behaving from a performance perspective. The Performance Monitoring Infrastructure (PMI) provides a comprehensive set of data that explains the runtime and application resource behavior. Using PMI data, the performance bottlenecks in the application server can be identified and fixed. PMI data can also be used to monitor the health of servers.

The PMI is included in the base WebSphere Application Server installation.

WebSphere Education

IBM

Tivoli Performance Viewer: Monitoring capabilities (1 of 2)

- Performance Monitoring Infrastructure (PMI) is an application programming interface (API) for WebSphere
- Three PMI categories apply to a BPM environment, including:
  - SIB Service
  - WBIStats.RootGroup
  - JCA Connection Pools
- WBIStats.RootGroup includes performance statistics on:
  - Business rules (BR), business state machines (BSM), maps, mediation, recovery, SCA, selector (SEL), business process engine (BPE), human task
  - GoodRequests, BadRequests, ResponseTime WebSphere Process Server particular categories might appear in previously available categories
  - For example, enterprise beans, JDBC connection pools, and web services
- Once monitors are enabled, Tivoli Performance Viewer interaction can be performed

Figure 10-21. Tivoli Performance Viewer: Monitoring capabilities (1 of 2) WB868 / ZB8681.0

Notes:

When viewing these statistics, do not mix counter-type statistics with duration-type statistics. Counters are cumulative, and the scales against which they are graphed can quickly grow depending on your application. Duration statistics, in contrast, tend to remain within a certain range because they represent the average amount of time that it takes your system to process each event. The disparity between the statistics and their relative scales can cause one or the other type of statistic to appear skewed in the viewer graph.

WebSphere Education

Tivoli Performance Viewer: Monitoring capabilities (2 of 2)

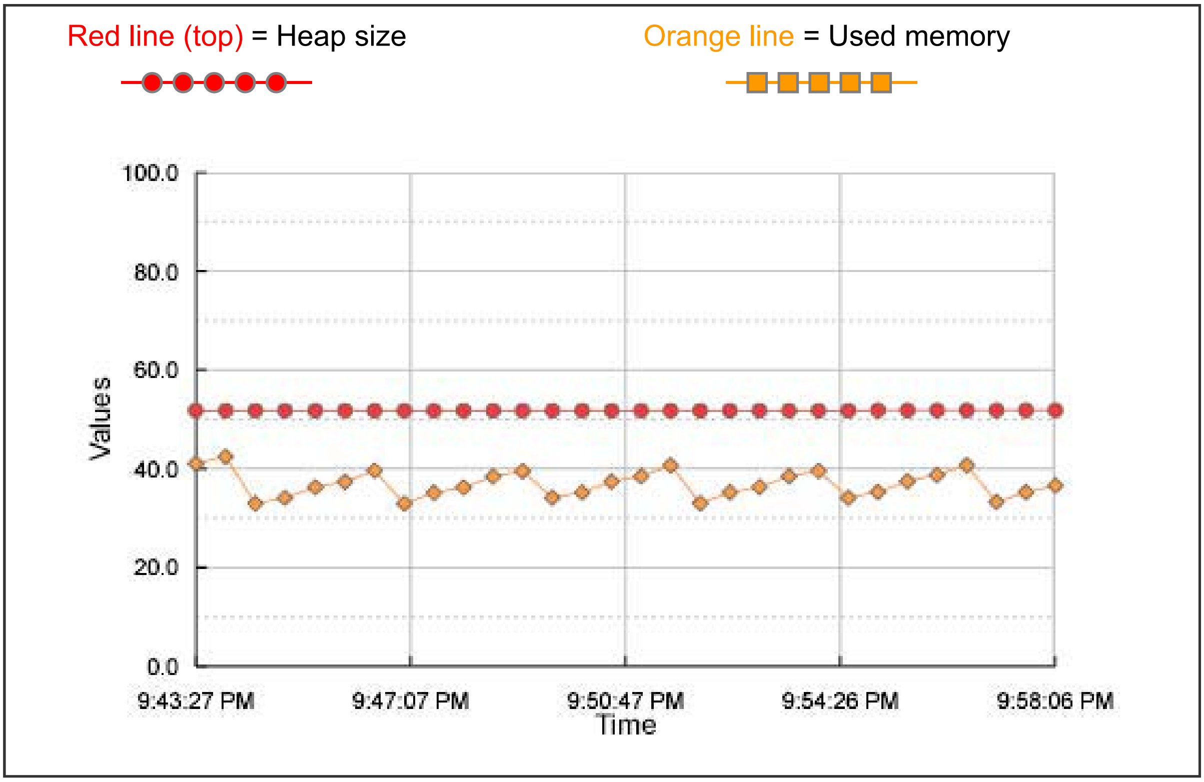


Figure 10-22. Tivoli Performance Viewer: Monitoring capabilities (2 of 2) WB868 / ZB8681.0

Notes:

The Tivoli Performance Viewer is a powerful application that allows you view various details of about the performance of your server. The section that is entitled “Monitoring performance with Tivoli Performance Viewer” in the WebSphere Application Server IBM Knowledge Center contains details about how to use this tool for various purposes. It includes the resource for complete instructions on using this program. This section is limited to discussing the viewing of performance data for events specific to IBM Business Process Manager Advanced.

The performance viewer enables administrators and programmers to monitor the current health of IBM Business Process Manager. Because the collection and viewing of data occurs on the process server, performance is affected. To minimize performance impacts, monitor only those servers whose activity you want to monitor.

Tivoli Performance Viewer usage

- Monitor real-time resource performance
  - Servlet request response times
  - Enterprise bean method calls
  - Data sources in use
  - Wait time on data sources
  - Concurrent waiters on data sources
- Gauge application server load
  - Percentage of pools in use
  - CPU load
  - Java virtual machine (JVM) heap size and amount in use
- Determine optimal resource configurations
  - Allocated memory
  - Database connection pool size
  - Enterprise bean objects cache size
- Detect trends by analyzing logs of data over time

Figure 10-23. Tivoli Performance Viewer usage WB868 / ZB8681.0

Notes:

Performance monitoring statistics are available for most server events. You can use performance monitoring statistics to monitor the counts of successful and unsuccessful invocation requests, and the time that is taken to complete events.

WebSphere Education IBM

Sources of JVM performance data

- JVM configuration, such as maximum heap size, are Java command-line arguments
  - For example: `-Xmx512m`
- Statistics regarding garbage collection behavior are available via verbose garbage collection output

```
<gc type="scavenger" id="1" totalid="1" intervalms="0.000">
  <flipped objectcount="20386" bytes="1146496" />
  <tenured objectcount="0" bytes="0" />
  <finalization objectsqueued="16" />
  <scavenger tiltratio="50" />
  <nursery freebytes="5321632" totalbytes="6553600" percent="81" tenureage="10" />
  <tenured freebytes="38817912" totalbytes="39321600" percent="98" >
    <soa freebytes="36851832" totalbytes="37355520" percent="98" />
    <loa freebytes="1966080" totalbytes="1966080" percent="100" />
  </tenured>
  <time totalms="7.753" />
</gc>
```

  - Graphing verbose garbage collection output is the best way to view garbage collection behavior

Figure 10-24. Sources of JVM performance data WB868 / ZB8681.0

Notes:

WebSphere Education IBM.

JVM performance tools: Verbose garbage collection

- The best way to analyze garbage collection behavior is to enable and collect verbose GC data and graph it
  - Garbage collection time data
- The IBM Support Assistant includes a tool that is called Garbage Collection and Memory Visualizer (GCMV), which will graph verbose GC
- GCMV uses a powerful statistical analysis engine that provides guidance in the following areas:
  - Memory leak detection
  - Optimize garbage collection performance
  - Fine-tuning of Java heap size

Figure 10-25. JVM performance tools: Verbose garbage collection WB868 / ZB8681.0

Notes:

The IBM Monitoring and Diagnostic Tools for Java - Garbage Collection and Memory Visualizer (GCMV) is a tool that allows you to visualize and analyze the memory usage and garbage collection activity of your Java application.

IBM Monitoring and Diagnostic Tools for Java - Garbage Collection and Memory Visualizer (GCMV) provides analysis and views of your application’s verbose GC output. GCMV displays the data in both graphical and tabulated form. It provides a clear summary and interprets the information to produce a series of tuning recommendations.

WebSphere Education

JVM performance tools: Thread dumps (1 of 2)

- Thread dumps provide a snapshot of the running JVM
- Useful when:
  - CPU utilization is not as expected (lower or higher)
  - Throughput hits a plateau
- Thread dump produces a stack trace of each thread's activity
- Things to keep in mind:
  - A thread dump is a snapshot
  - Take several thread dumps, giving the JVM time to recover after each
  - They vary in format and detail among platforms

Figure 10-26. JVM performance tools: Thread dumps (1 of 2) WB868 / ZB8681.0

Notes:

A javacore dump, or a thread dump as it is also called, is one of the primary problem determination documents that an application server creates.

WebSphere Education IBM

JVM performance tools: Thread dumps (2 of 2)

- Analyzing thread dumps with a plain editor can be difficult
  - Use a thread dump viewer
- IBM Thread and Monitor Dump Analyzer for Java
  - Simplifies the process of analyzing a thread dump by presenting a list of the threads that existed at the time the thread dump was triggered
  - Simplifies the process of examining the Java call stack from the list of threads
  - Provides information about locking
  - Shows the threads that own locks
  - Shows the threads that are blocked waiting on a lock

Figure 10-27. JVM performance tools: Thread dumps (2 of 2) WB868 / ZB8681.0

Notes:

IBM Thread and Monitor Dump Analyzer for Java is a tool that allows identification of hangs, deadlocks, resource contention, and bottlenecks in Java threads. During the run time of a Java process, some Java virtual machines (JVMs) might not respond predictably and often seem to hang for a long time or until JVM shutdown occurs. It is not easy to determine the root cause of these sorts of problems.

By triggering a javacore when a Java process does not respond, it is possible to collect diagnostic information that is related to the JVM. Also, a Java application might be captured at a particular point during execution. For example, the information can be about the operating system, the application environment, threads, native stack, locks, and memory. The exact contents depend on the platform on which the application is running.

On some platforms, and in some cases, javacore is known as “javadump.” The code that creates javacore is part of the JVM. It can be controlled by using environment variables and runtime switches. By default, a javacore occurs when the JVM terminates unexpectedly. A javacore can also be triggered by sending specific signals to the JVM. Although javacore or javadump is present in Sun Solaris JVMs, IBM adds much of the content of the javacore; therefore, this content is present only in IBM JVMs.

IBM Thread and Monitor Dump Analyzer for Java analyzes javacore and diagnoses monitor locks and thread activities to identify the root cause of hangs, deadlocks, and resource contention or monitor bottlenecks.

10.4.Business Process Manager tools

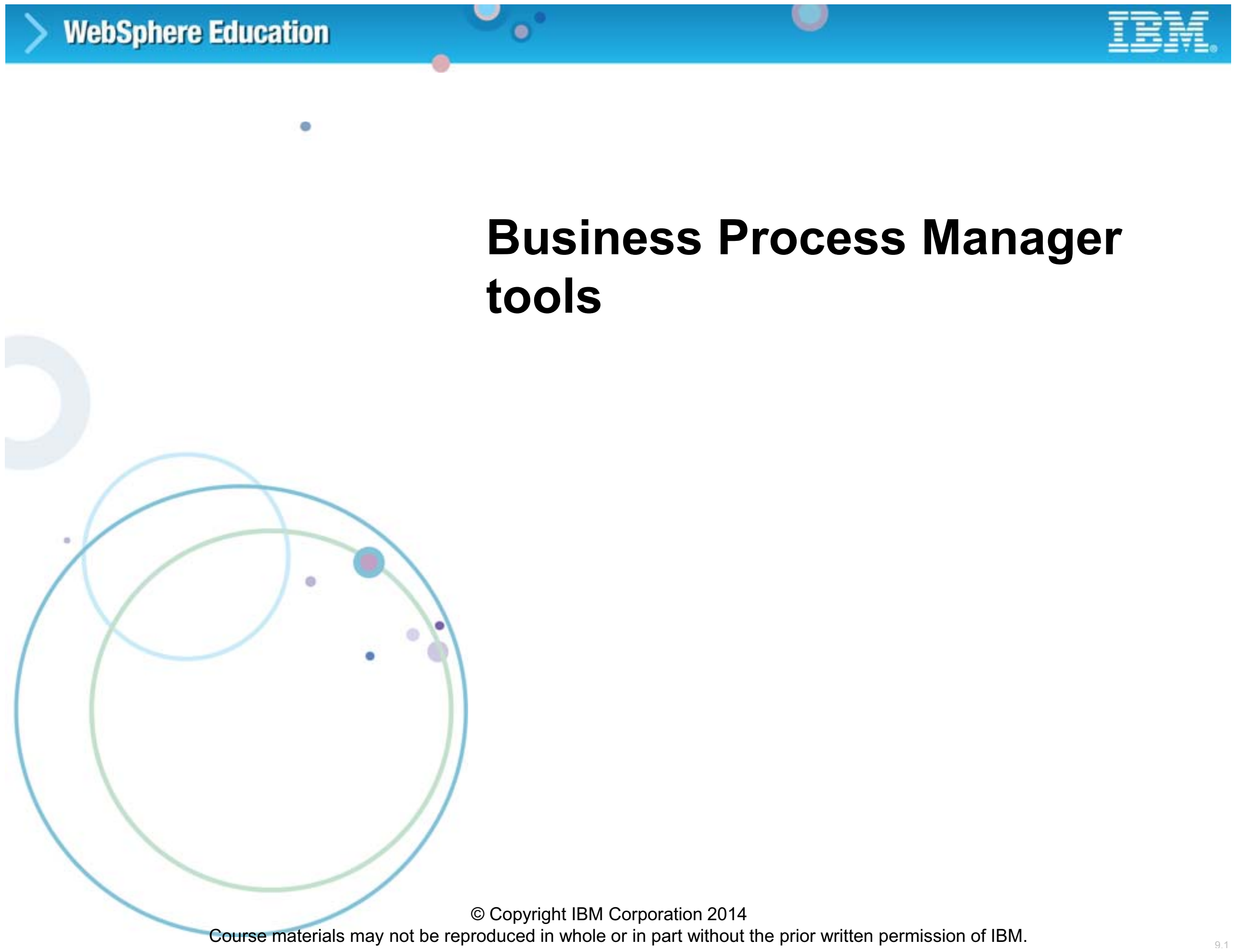


Figure 10-28. Business Process Manager tools WB868 / ZB8681.0

Notes:

WebSphere Education

Service Integration Bus Browser

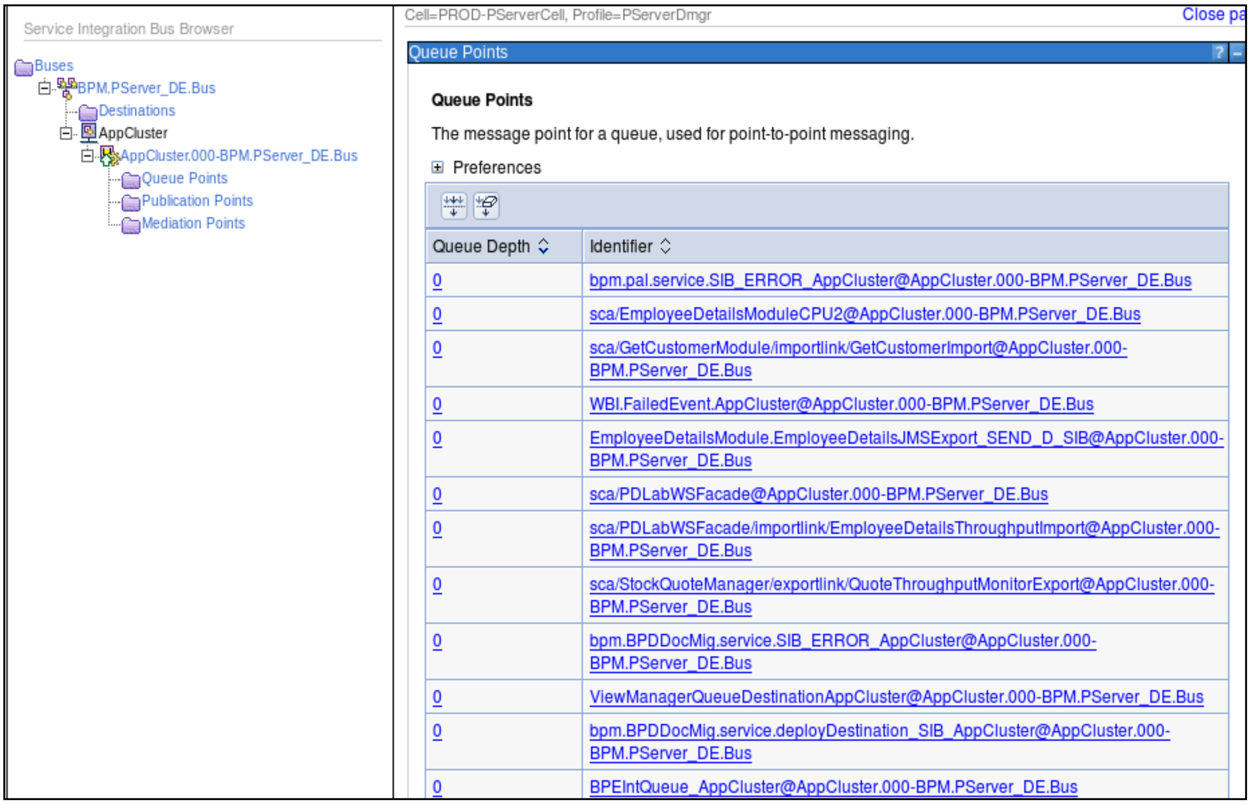


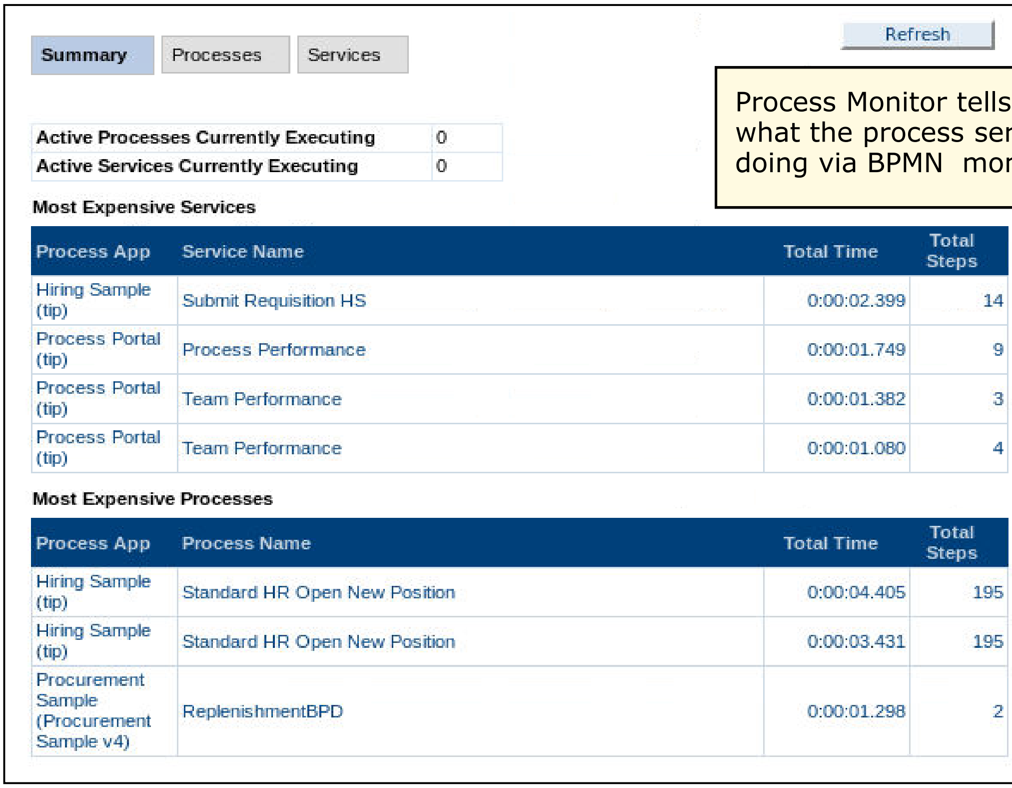
Figure 10-29. Service Integration Bus Browser WB868 / ZB8681.0

Notes:

The Service Integration Bus Browser has two panes. The first pane (referred to in the help as the tree pane) presents a navigation tree where you can browse the service integration buses and their components that are configured on the system. The second pane (referred to in the help as the content pane) contains the collection and detail pages for the buses and their individual components. It includes messaging engines, queue points, destinations, publication points, and mediation points. When you click an item in the navigation tree pane, its corresponding collection or detail page opens in the content pane.

WebSphere Education

Process Monitor



Summary | Processes | Services | Refresh

Active Processes Currently Executing	0
Active Services Currently Executing	0

Most Expensive Services

Process App	Service Name	Total Time	Total Steps
Hiring Sample (tip)	Submit Requisition HS	0:00:02.399	14
Process Portal (tip)	Process Performance	0:00:01.749	9
Process Portal (tip)	Team Performance	0:00:01.382	3
Process Portal (tip)	Team Performance	0:00:01.080	4

Most Expensive Processes

Process App	Process Name	Total Time	Total Steps
Hiring Sample (tip)	Standard HR Open New Position	0:00:04.405	195
Hiring Sample (tip)	Standard HR Open New Position	0:00:03.431	195
Procurement Sample (Procurement Sample v4)	ReplenishmentBPD	0:00:01.298	2

Process Monitor tells you what the process server is doing via BPMN monitoring

Figure 10-30. Process Monitor WB868 / ZB8681.0

Notes:

Process Monitor provides details on what services and BPDs are running and how long each step is taking. It allows you to identify items that are consuming large amounts of resources or are stuck in an infinite loop due to process modeling bus. You can end a looping service if needed. Process Monitor gives you the ability to understand bottlenecks in your application. Process Monitor is per JVM, not cluster wide. You must go to each process server in the cluster to check the status of the server.

WebSphere Education

Instrumentation Monitor

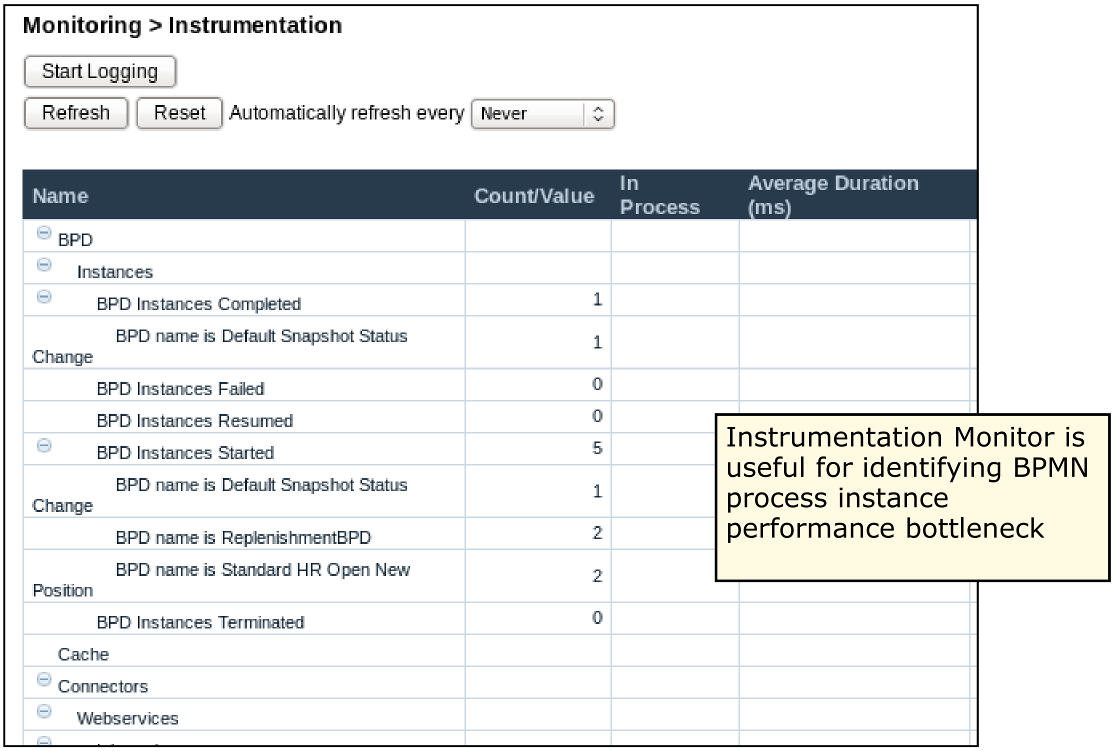


Figure 10-31. Instrumentation Monitor WB868 / ZB8681.0

Notes:

Instrumentation Monitor enables you to display and collect instrumentation data. The Instrumentation Monitor is useful for identifying BPMN process instance performance bottlenecks and also for capturing instrumentation data. Instrumentation Monitor is available in the Process Admin Console.

WebSphere Education

Process Inspector

- Process Inspector shows information and details about failing instances
- Actions available to interact with processes
- In the Process Admin Console and in Process Designer
- Provides BPMN monitoring

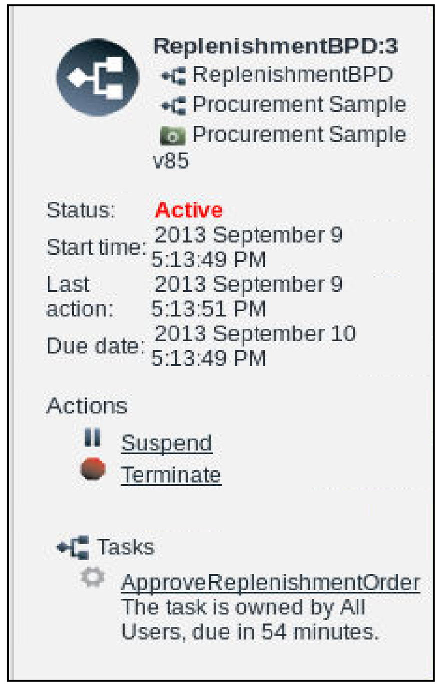


Figure 10-32. Process Inspector WB868 / ZB8681.0

Notes:

Process Inspector shows information and details about failing instances. It also helps you to suspend running instances and helps to narrow down the issues that you are experiencing.

WebSphere Education

Event Manager Monitor

- The Event Manager Monitor displays tasks and activities that are successfully scheduled, initiated, and running in the Event Manager
  - Displays processes that are in the queue, running, or paused
  - Provides BPMN monitoring

Event Manager > Monitor

	Scheduler ID	Status	Connect expiration	# Jobs Executing
☑	PServerNode01_AppClusterMember01		Jun 12, 2014 1:37:11 PM	0

Total Jobs Executing: 0 **Total Jobs:** 1

Refresh | Pause | Resume | Pause All | Resume All

Scheduler	Process App / Toolkit	Snapshot	Job Name	Job Queue	Scheduled Time	Last Scheduled Time	Last Execution Time	Next Scheduled Time	Job Status
	Process Portal	8.5.0.1	Execute UCA Periodic SLA Update, on set schedule	SYNC_QUEUE_1	6/12/14 1:45:00 PM	6/12/14 1:30:00 PM	6/12/14 1:30:00 PM	6/12/14 2:00:00 PM	Scheduled

Figure 10-33. Event Manager Monitor WB868 / ZB8681.0

Notes:

The Event Manager monitor, included in the Process Admin Console, is useful for troubleshooting processes that are supposed to run automatically (through an undercover agent, for example) but fail to start. You can use the Event Manager monitor to identify underlying problems and also to control various aspects of Event Manager processing.

The Event Manager is the part of the Process Server that handles event scheduling and queuing. For example, when Process Server receives an event, that event becomes a job in the Event Manager. Each job in the Event Manager is routed through a Scheduler, which schedules and tracks the execution of its assigned jobs.

WebSphere Education

Performance Admin Console

- Used to view load, errors, and helping to identify bottlenecks environment
- Captures instrumentation data that can be used to further analyze any performance issues

Figure 10-34. Performance Data Warehouse console WB868 / ZB8681.0

Notes:

The Business Performance Data Warehouses in your IBM Business Process Manager configuration retrieve and store tracked performance data, which allows users in IBM Process Designer to create reports and also analyze processes by using the Optimizer.

As part of system maintenance, you might want to view the Performance Data Warehouse load queue to determine which records have yet to be loaded to the database. You might also want to view the error queue to determine whether any errors occurred while data was being loaded from the Process Server to the Performance Data Warehouse.

WebSphere Education

Business Process Choreographer Explorer

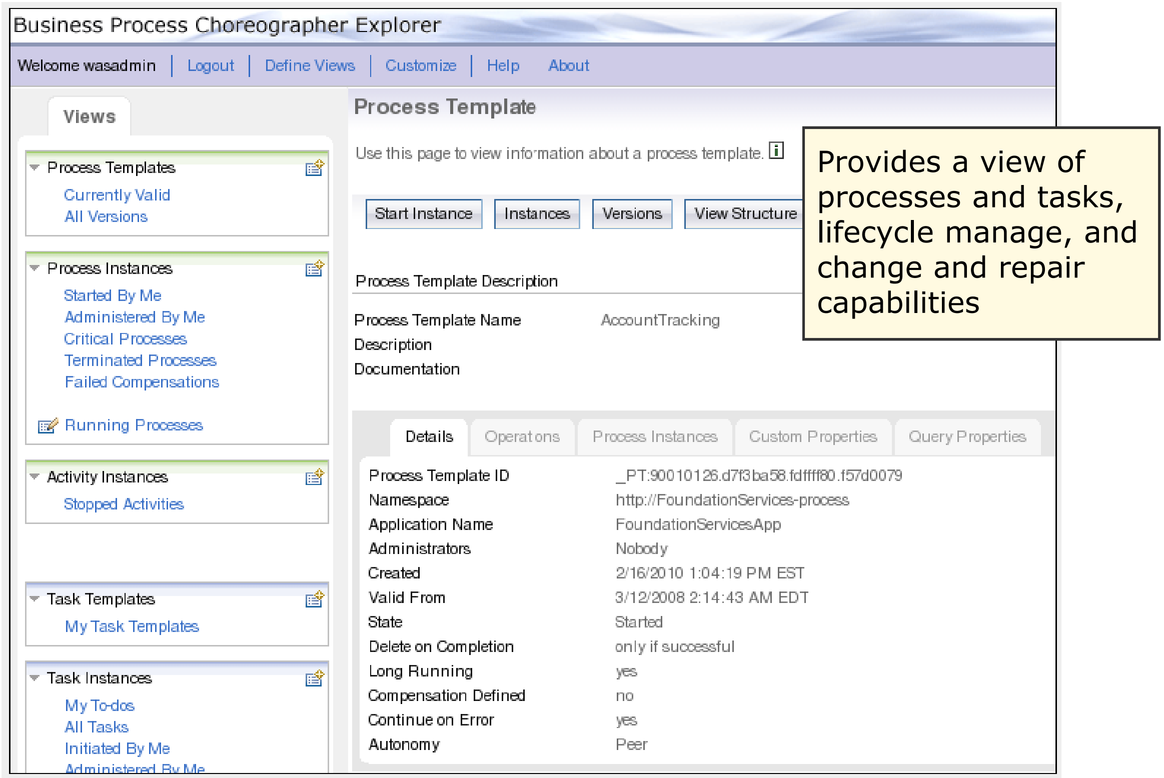


Figure 10-35. Business Process Choreographer Explorer WB868 / ZB8681.0

Notes:

You can use Business Process Choreographer Explorer to view information about the started templates.

WebSphere Education

Failed Event Manager

- The Failed Event Manager can be used to find and manage failed events on all servers in a deployment environment
  - Provides SCA and BPEL monitoring

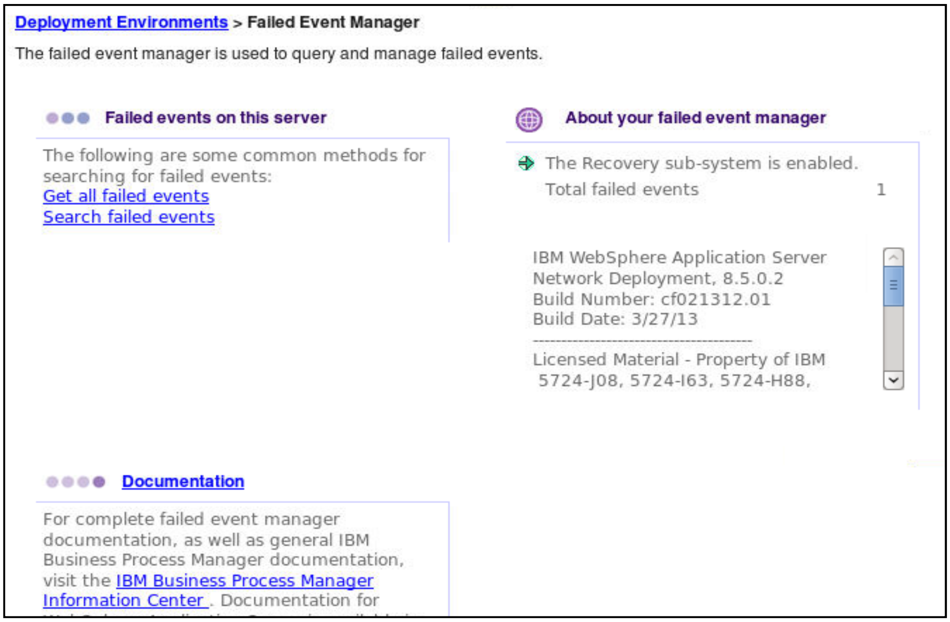


Figure 10-36. Failed Event Manager WB868 / ZB8681.0

Notes:

You can use the Failed Event Manager to find failed events in IBM Business Process Manager. You can search for all failed events or for a specific subset of events on all the servers within the deployment environment. Click any failed event to see all of the data that is associated with it.

If an event fails, it is stored in a database in the Failed Event Manager. Use the Failed Event Manager to search for and handle failed events.

Actions for handling failed events include examining the types of data that is associated with the event (business, trace, or expiration data) to determine the cause of the failure. Actions also include editing the data, resubmitting the event, or both.

WebSphere Education

Business Space System Health widget

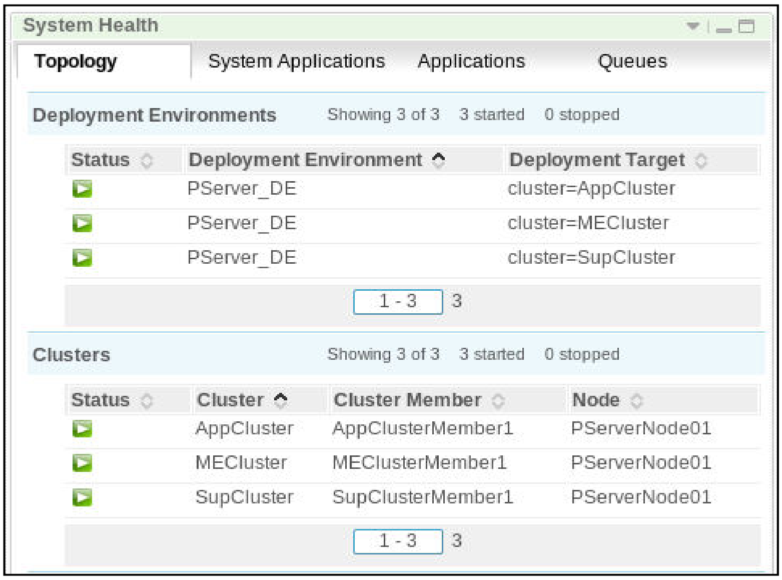


Figure 10-37. Business Space System Health widget WB868 / ZB8681.0

Notes:

The System Health widget is organized into the following four tabs:

- **Topology**: Displays the status of deployment environments, clusters, stand-alone servers, and node agents
- **System Applications**: Provides the status of system applications, data sources, and messaging engines
- **Application**: Lists the status of enterprise applications and failed events
- **Queues**: Displays the queue depth for the messaging point

10.5.IBM Support Assistant overview

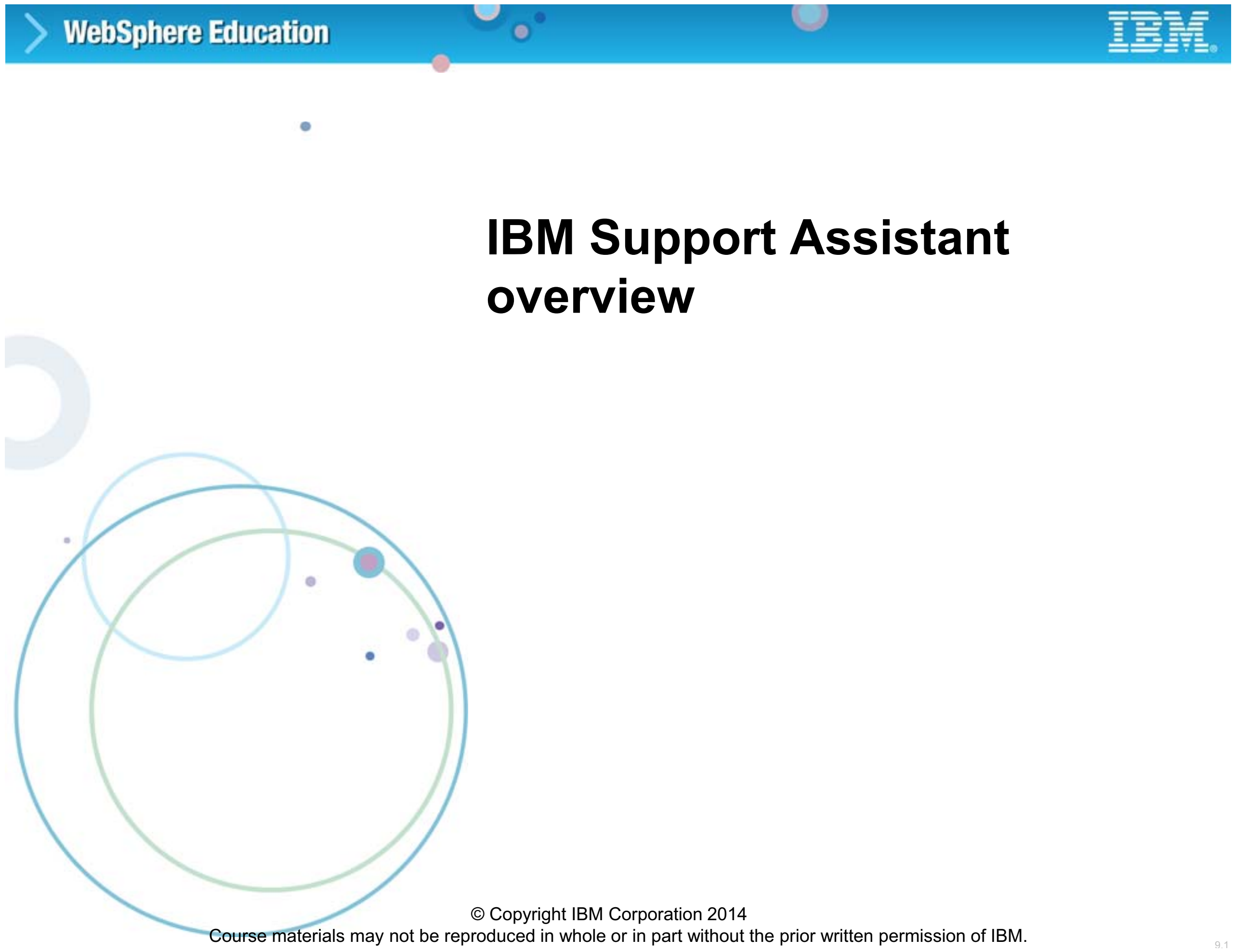


Figure 10-38. IBM Support Assistant overview WB868 / ZB8681.0

Notes:

WebSphere Education

What is the IBM Support Assistant Team Server 5.0?

- Free self-help problem determination application
  - Cloud based
  - Multiple installation options
- Use to organize diagnostic files into cases
  - Store diagnostic files by problem incident or other categories
  - Simplifies collaboration
- Provides a growing collection of problem determination tools
  - Report generators
  - Interactive web-based tools
  - Desktop tools
- Provides automated analysis of diagnostic files
- Download IBM Support Assistant Team Server 5.0

  `http://www.ibm.com/software/support/isa/teamserver.html`

Figure 10-39. What is the IBM Support Assistant Team Server 5.0? WB868 / ZB8681.0

Notes:

The IBM Support Assistant provides a framework for IBM software products to deliver customized self-help information into the different tools within it. Customize your IBM Support Assistant client by using the built-in Update capability to find and install new product features or support tools.

Being server-based provides several advantages over IBM Support Assistant V4.

- Processing resources are moved from your workstation to the server.
- No installation is required on user workstations. All that you need is a browser.
- Diagnostic files and analysis reports are available to all members of the team for each case.
- Report generator tools and interactive web tools are supported along with the desktop tools that you used with IBM Support Assistant V4.

WebSphere Education

Installing IBM Support Assistant 5: Overview

- Installation option: Use the Installation Manager
  - Choose which problem determination tools to install
  - Makes it easy to keep IBM Support Assistant and tools up-to-date
- Installation option: All-in-one compressed file
  - Includes IBM Support Assistant and all available problem determination tools
  - Ideal for environments that are disconnected from the Internet

- Deployment option: Embedded server (Preferred)
  - Embedded server (Liberty profile)
  - Configuration ready to run
- Deployment option: EAR deployment to an existing WebSphere Application Server environment
  - Any full profile application server instance
  - Flexibility

Figure 10-40. Installing IBM Support Assistant 5: Overview WB868 / ZB8681.0

Notes:

There are two installation options and two deployment options. These slides cover the use of the Installation Manager to install, and stand-alone or embedded server deployment.

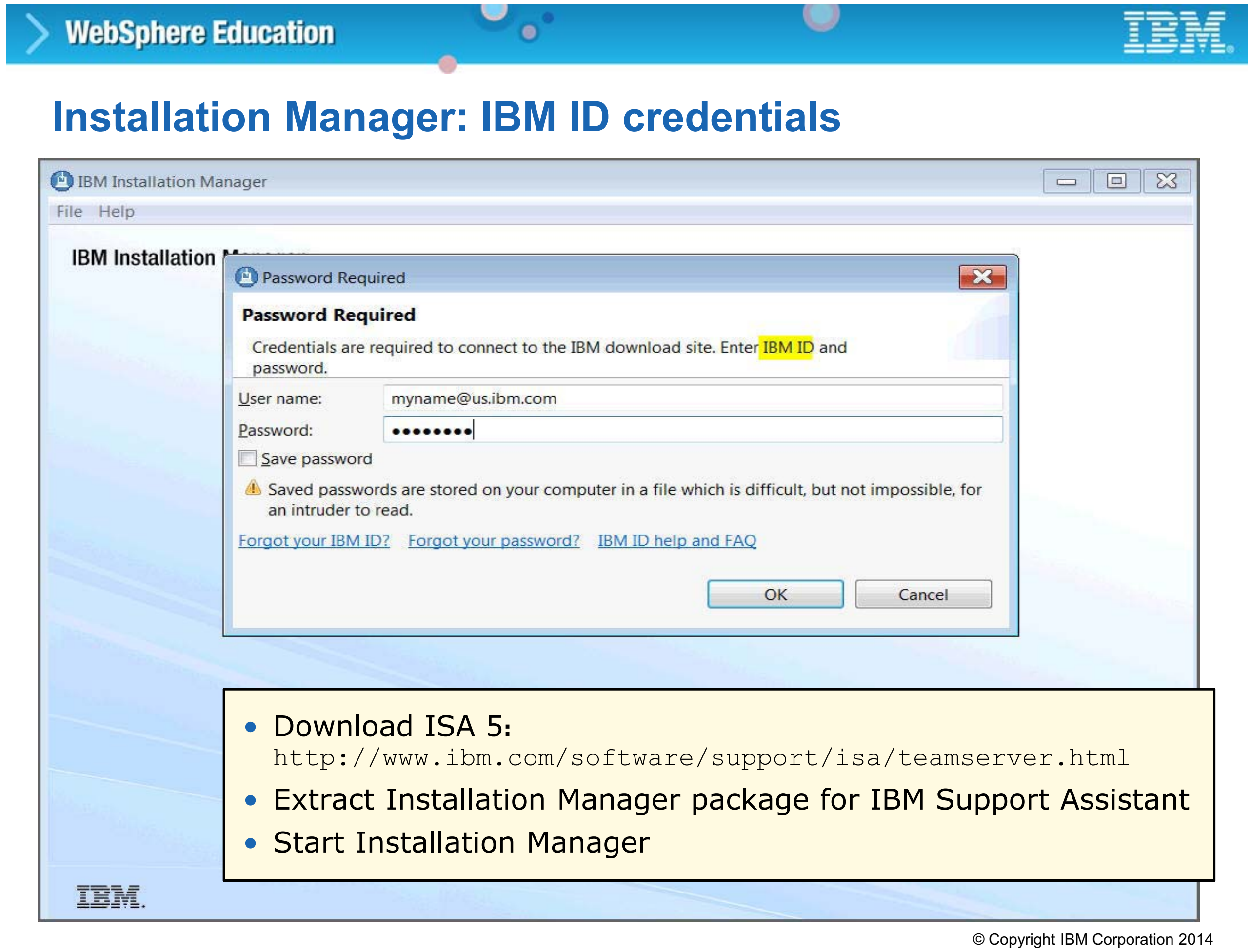


Figure 10-41. Installation Manager: IBM ID credentials WB868 / ZB8681.0

Notes:

When you download IBM Support Assistant 5 for installing with Installation Manager, the repository location of IBM Support Assistant is configured for you. When you start Installation Manager, you are prompted for your IBM ID and password.

WebSphere Education

Getting started with IBM Support Assistant 5

Team Server URL: `http://<hostname>:10911/isa5`

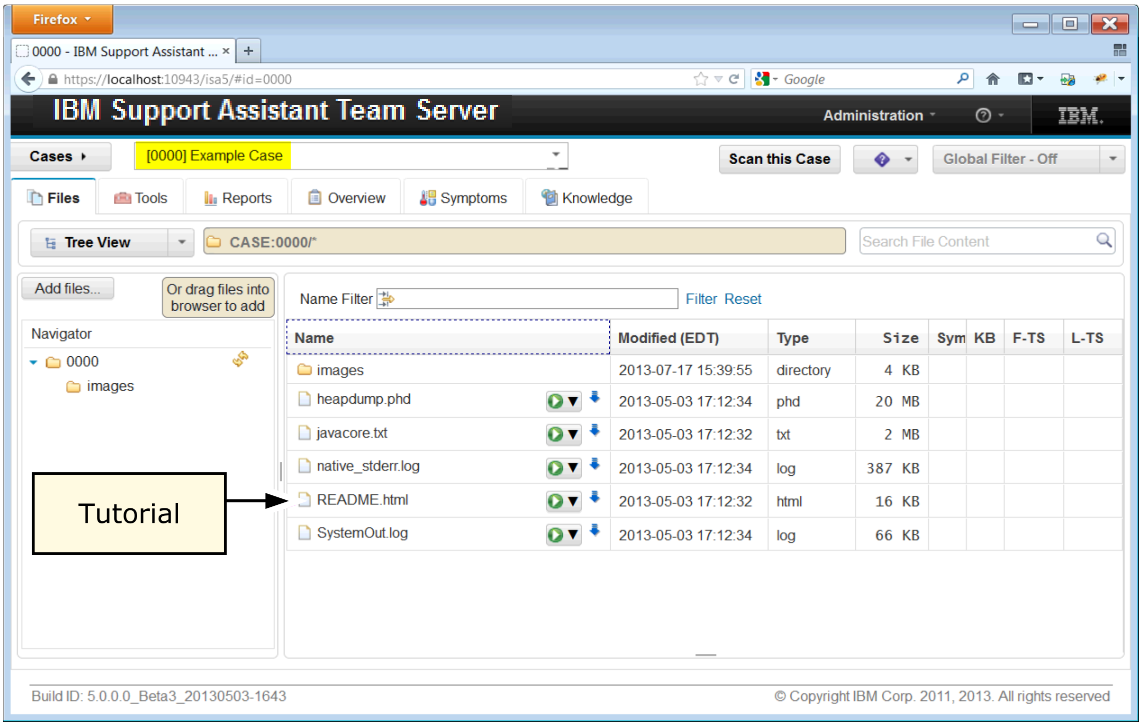


Figure 10-42. Getting started with IBM Support Assistant 5 WB868 / ZB8681.0

Notes:

If you are new to IBM Support Assistant 5, then you should start with the tutorial in the example case. Start the IBM Support Assistant server and then open a browser and point it to **http://<hostname>:10911/isa5**. After IBM Support Assistant loads, choose the **Example Case** from the **Cases** drop-down list under the black banner. The files for the case load. Double-click **README.html** to open the tutorial. It introduces you to IBM Support Assistant's features by guiding you through a troubleshooting workflow.

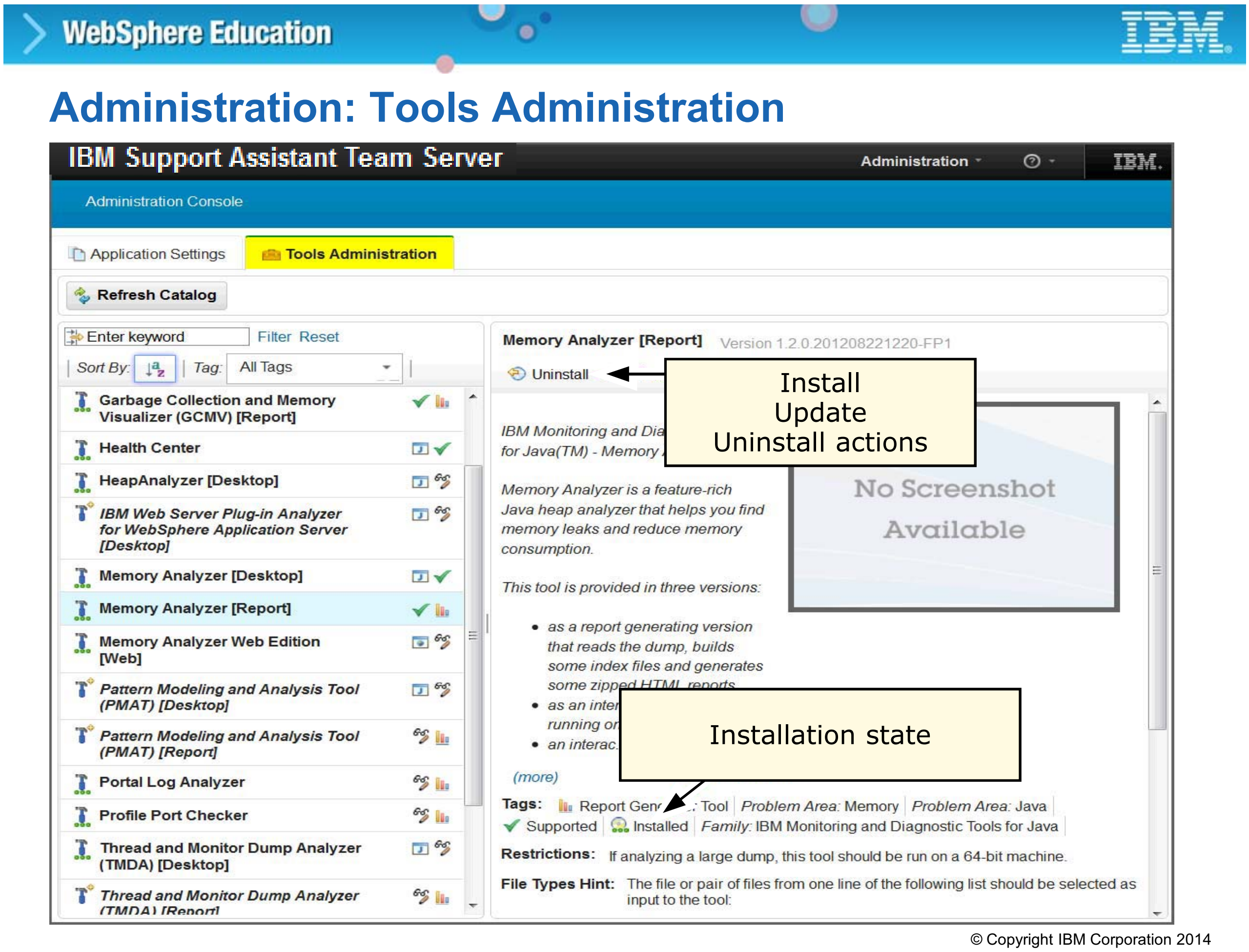


Figure 10-43. Administration: Tools Administration WB868 / ZB8681.0

Notes:

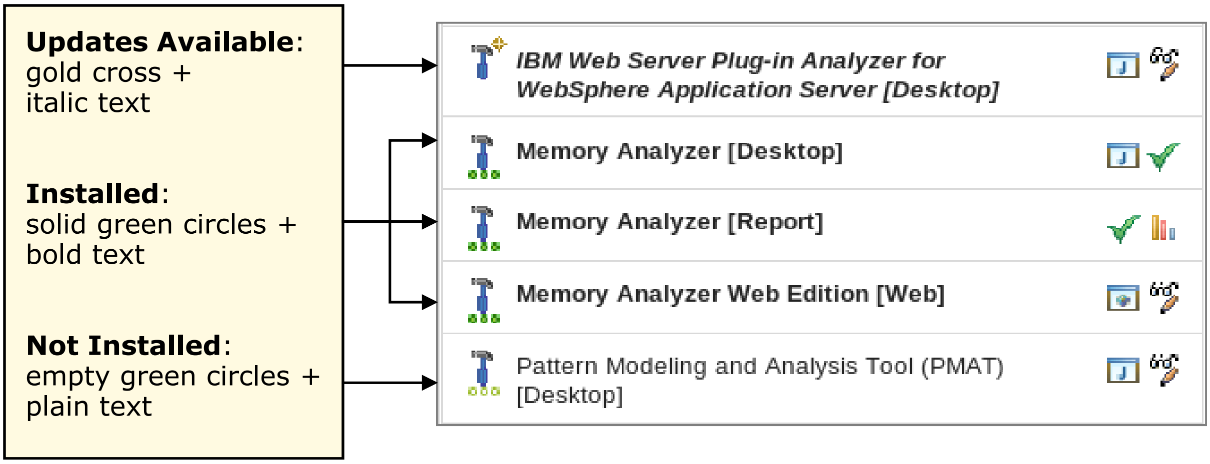
The Tools Administration tab shows you a list of all problem determination tools that are available for IBM Support Assistant 5. You can use this tab to install, update, and uninstall problem determination tools. The column on the left shows you the complete list of tools. Under the tool title above the details pane, you find buttons for installing, updating, or uninstalling tools that depend on the current installation state.

The next slides show you the various ways to recognize the installation state.

WebSphere Education

Administration: Tool installation state indicators

- Visual cues to tell you whether a tool is installed, uninstalled, or installed with an update available



- In addition, a set of icons show state in the tags list and in the tags that are shown on the details pane for each tool

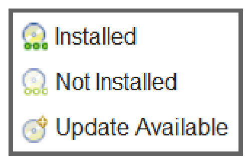


Figure 10-44. Administration: Tool installation state indicators WB868 / ZB8681.0

Notes:

WebSphere Education

Case management

- Manage problem incidents with cases
  – Containers for logically grouping files, reports, and other information
  – Save notes about each case
  – Upload diagnostic files to store in cases

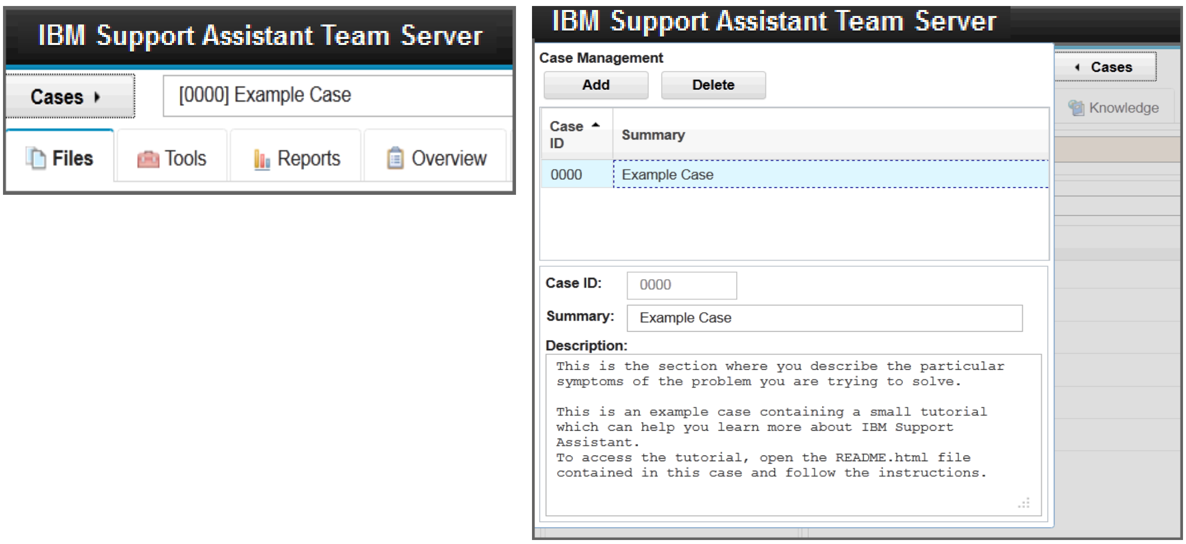


Figure 10-45. Case management WB868 / ZB8681.0

Notes:

Before you begin troubleshooting a problem, you should create a case to manage related diagnostic files. You use cases to organize diagnostic artifacts by problem incident, failing system, time, or any other logical categories that you like. When you create a case, you assign a summary and you can optionally add details.

Click **Cases** above the Files tab to open a “Case Management” slider where you can add, edit, and delete cases.

After you have a case, you are ready to start adding files to it.

WebSphere Education

Files tab: Add files to a case

- Click **Add files**
- Browse the file system or drag files into the browser

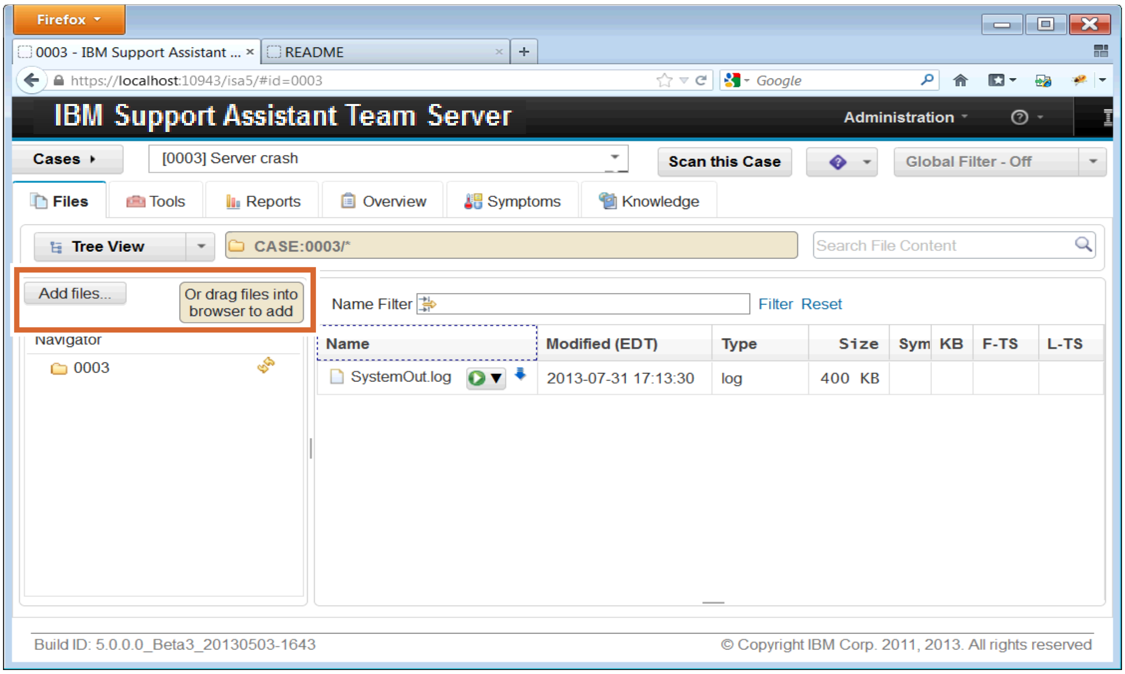


Figure 10-46. Files tab: Add files to a case WB868 / ZB8681.0

Notes:

There are two ways to add files to a case: click **Add files...** or “drag”. Clicking **Add files...** opens the browser’s file selection window where you can select one or more files. After you select your files, a progress bar appears, and when the upload is complete, you see a yellow notification message at the top of the IBM Support Assistant UI.

Use “drag” to grab files from your OS desktop and drop them into the selected case. As with **Add files**, you see a progress bar and a notification message when the upload is complete.

10.6.IBM Support Assistant tools

Figure 10-47. IBM Support Assistant tools WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Editor's choice and suggested tools

- Memory Analyzer
  – Java heap analyzer that helps you find memory leaks and reduce memory consumption (report, web, desktop)
- Thread and Monitor Dump Analyzer (TMDA)
  – Compares each thread dump and monitor dump and automatically detects hangs, resource contention, Java monitor ownership directional graph structure, and deadlocks.
- Garbage Collection and Memory Visualizer (GCMV)
  – Provides graphical display of a wide range of verbose GC data values together with tuning suggestions and detection of problems such as memory leaks
- Health Center
  – A lightweight tool that monitors active IBM virtual machines for Java with minimal performance administrative cost and provides live tuning suggestions and observations.
- Interactive Diagnostic Data Explorer (IDDE)
  – Use to more easily explore and examine system dump files that the JVM produces

Figure 10-48. Editor's choice and suggested tools WB868 / ZB8681.0

Notes:

Interactive Diagnostic Data Explorer (IDDE) is a GUI-based alternative to the dump viewer (`jdmpview` command). IDDE provides the same functionality as the dump viewer, but with extra support such as the ability to save command output.

Use IDDE to more easily explore and examine dump files that the JVM produces. Within IDDE, you enter commands in an investigation log to explore the dump file.

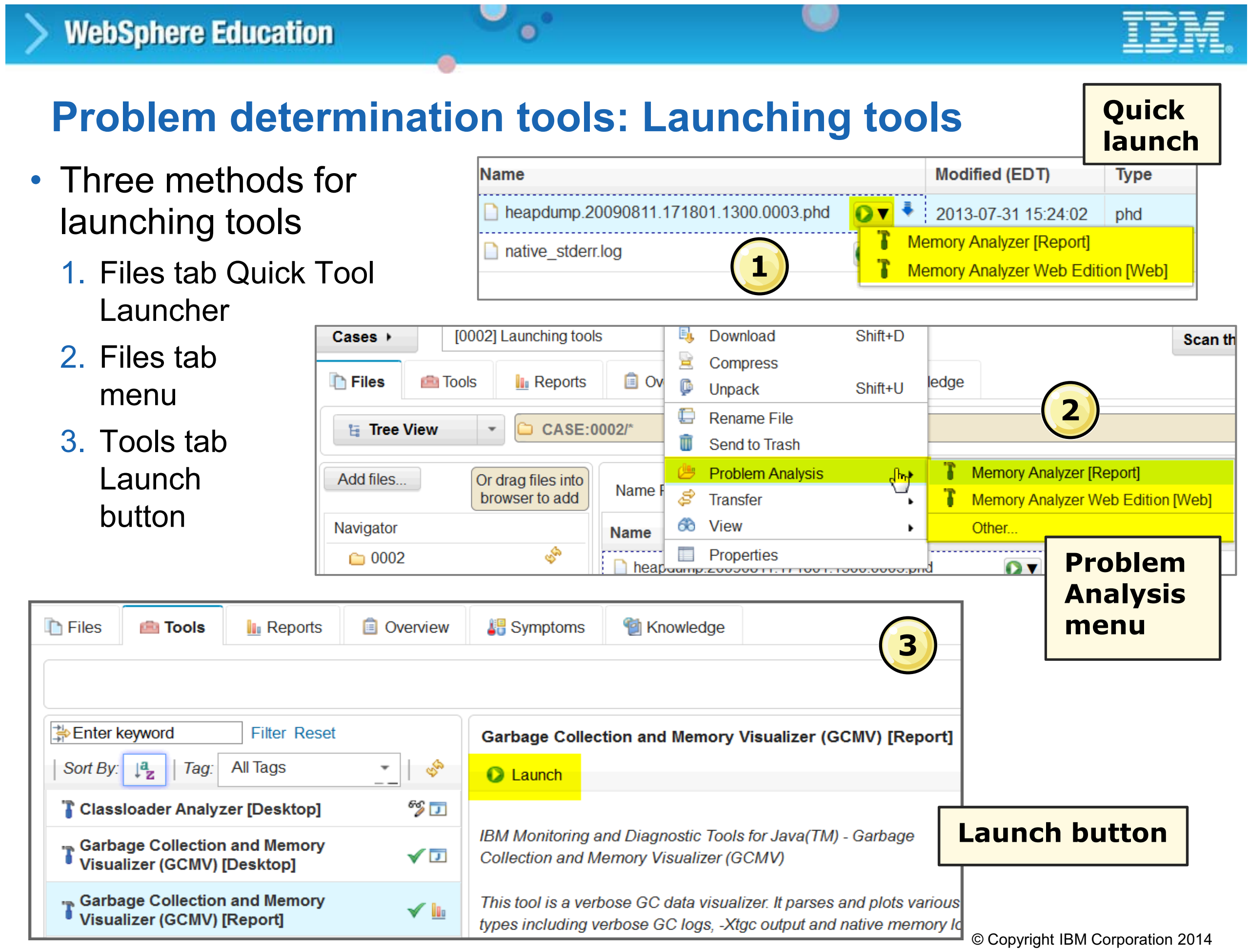


Figure 10-49. Problem determination tools: Launching tools WB868 / ZB8681.0

Notes:

There are three methods for launching tools. On the Files tab, you can choose the QuickLaunch icon next to a file name or right-click one or more files and choose a tool from the **Problem Analysis** item of the pop-up menu. The lists are filtered based on the type of file that is selected to show you which tools might be applicable. If the tool you need is not listed, then you can select **Problem Analysis > Other...** to choose from the complete list of installed tools.

You can also launch tools from the Tools tab. Under the title of each tool on its details pane is a Launch button. When you click launch for report and web tools, you see a browse button on the input dialog, with which you can choose the case files for analysis. The desktop tools are available only from the Tools tab.

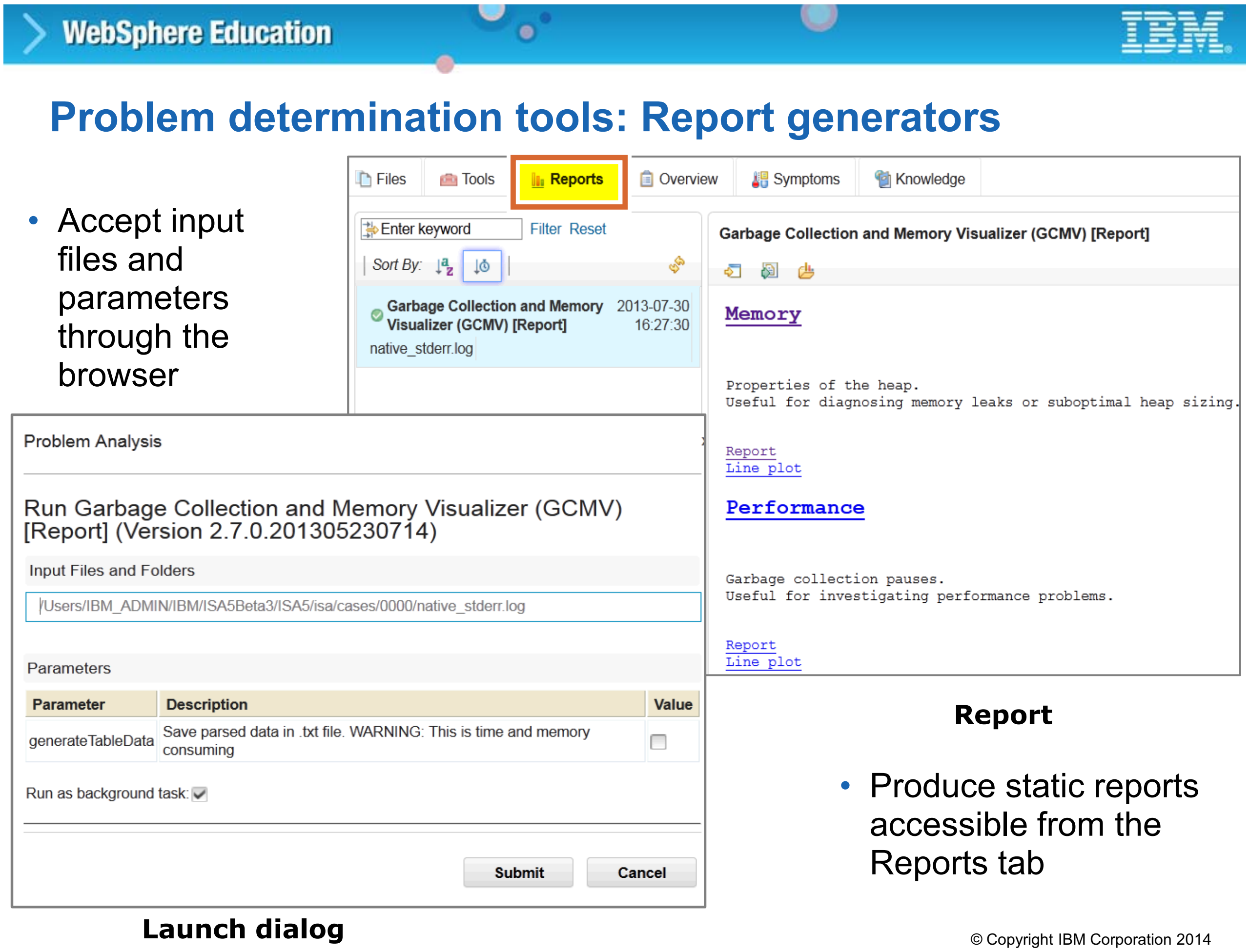


Figure 10-50. Problem determination tools: Report generators WB868 / ZB8681.0

Notes:

Report generators accept one or more files or folders as input along with any additional parameters the tool needs to run. It does its work, and when it is finished, you can view the results on the Reports tab.

Reports are organized according to case (you see reports for the case that is open).

The status icon to the left of the report name in the list of reports indicates success (green check mark), failure (red X), or in progress (hourglass). Other information in the report list includes the time that the report was run and the input files. When you hover over an item, a tooltip opens showing input files and input parameters.

You can filter the list by using keywords, and you can sort it alphabetically or by time. The default sorting is by time with the most recent report at the top of the list.

The larger right pane shows the report. Use the icons beneath the report title and above the report to take more actions.

- Use the left icon to open the report in a new browser tab.
- The middle icon switches to the Files tab and navigates to the folder that contains the input files.

- The right icon (graph in a folder) switches to the Files tab and navigates to the folder that contains the report files and logs.

WebSphere Education

Problem determination tools: Desktop tools

- Select a desktop tool from the Tools tab
- Tool downloads (via Java Web Start) and runs on your local workstation

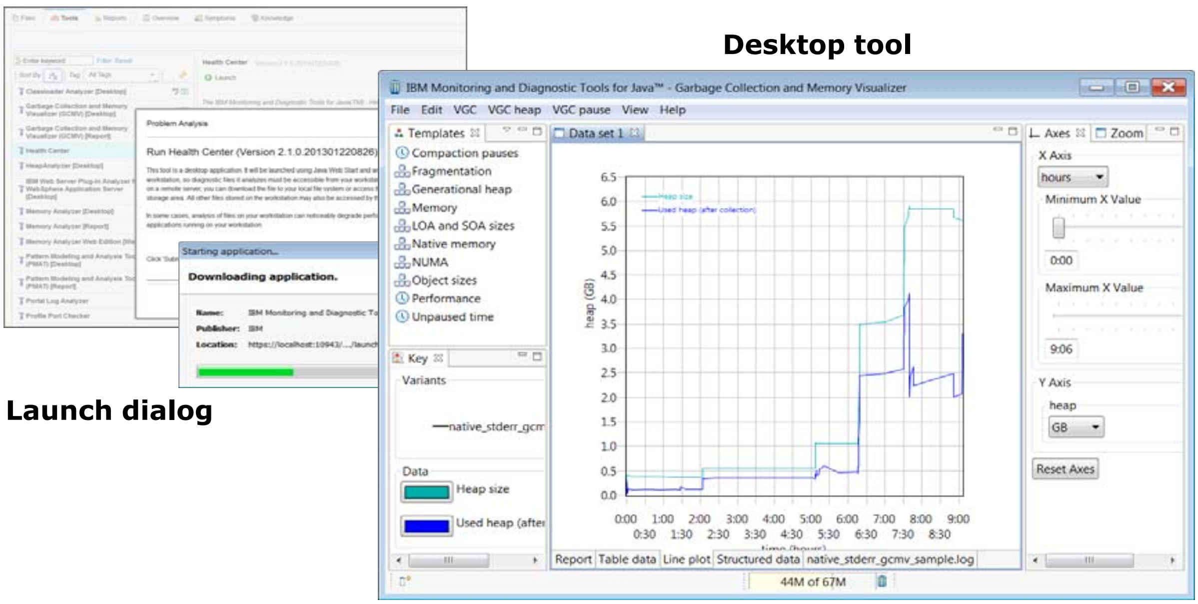


Figure 10-51. Problem determination tools: Desktop tools WB868 / ZB8681.0

Notes:

IBM Support Assistant uses Java Web Start technology to support desktop tools. When you choose to launch a desktop tool, the browser either asks you what to do when you click “Submit” on the confirmation dialog or it automatically starts downloading the tool and then launches it. This process does not require a Java plug-in or any other type of browser extension; it uses the built-in download capability of the browser.

Desktop tools allow you to work only with diagnostic files that are stored locally on your workstation or on a mounted file system. If the cases file system is not mounted for your workstation, then you would need to first download case files to your workstation to analyze them.

Note: The purpose of this slide is to illustrate the general appearance of the tool. Do not be concerned about reading the details of the screen captures.

WebSphere Education

JVM tuning tools (1 of 2)

- GCMV
  - Tool to analyze Java verbose GC logs
  - Graphs suggestions to guide you towards uses that might be limiting performance
  - Show garbage collection and Java heap stats over time
  - Not only for memory errors, it is good for performance tuning

- Diagnostic collector
  - At JVM start, it runs a diagnostic configuration check
  - Runs as a separate process when the JVM detects a dump event
  - Java heap out-of-memory error
  - Knows all possible dump locations and searches to gather all dumps into a single compressed file
  - Collects system dumps, Java dumps, heap dumps, verbose GC logs

Figure 10-52. JVM tuning tools (1 of 2) WB868 / ZB8681.0

Notes:

GCMV parses and plots data from various types of log, including the following types:

- Verbose garbage collection logs
- Trace garbage collection logs, generated by using the `-Xtgc` parameter
- Native memory logs, generated by using the `ps`, `svmon`, or `perfmon` system commands

The tool helps to diagnose problems such as memory leaks, analyzes data in various visual formats, and provides tuning recommendations.

WebSphere Education IBM

Tuning tools

- Memory analyzer
  - Eclipse project for analyzing heap dumps and identifying memory leaks from the JVM
  - Works with IBM system dumps, heap dumps, and Sun hprof binary dumps
  - Provides memory leak detection and footprint analysis
- Health Center
  - Live monitoring tool with low operation cost
  - Provides an understanding of how your application is behaving
  - Provides access to information about method profiling, GC, class loading, locking, and environment data
  - Diagnose potential problems with suggestions
  - Works at the JVM level

Figure 10-53. Tuning tools WB868 / ZB8681.0

Notes:

Health Center is a diagnostic tool for monitoring the status of a running Java virtual machine (JVM).

The tool is provided in two parts:

- The Health Center agent that collects data from a running application.
- An Eclipse-based client that connects to the agent. The client interprets the data and provides recommendations to improve the performance of the monitored application.

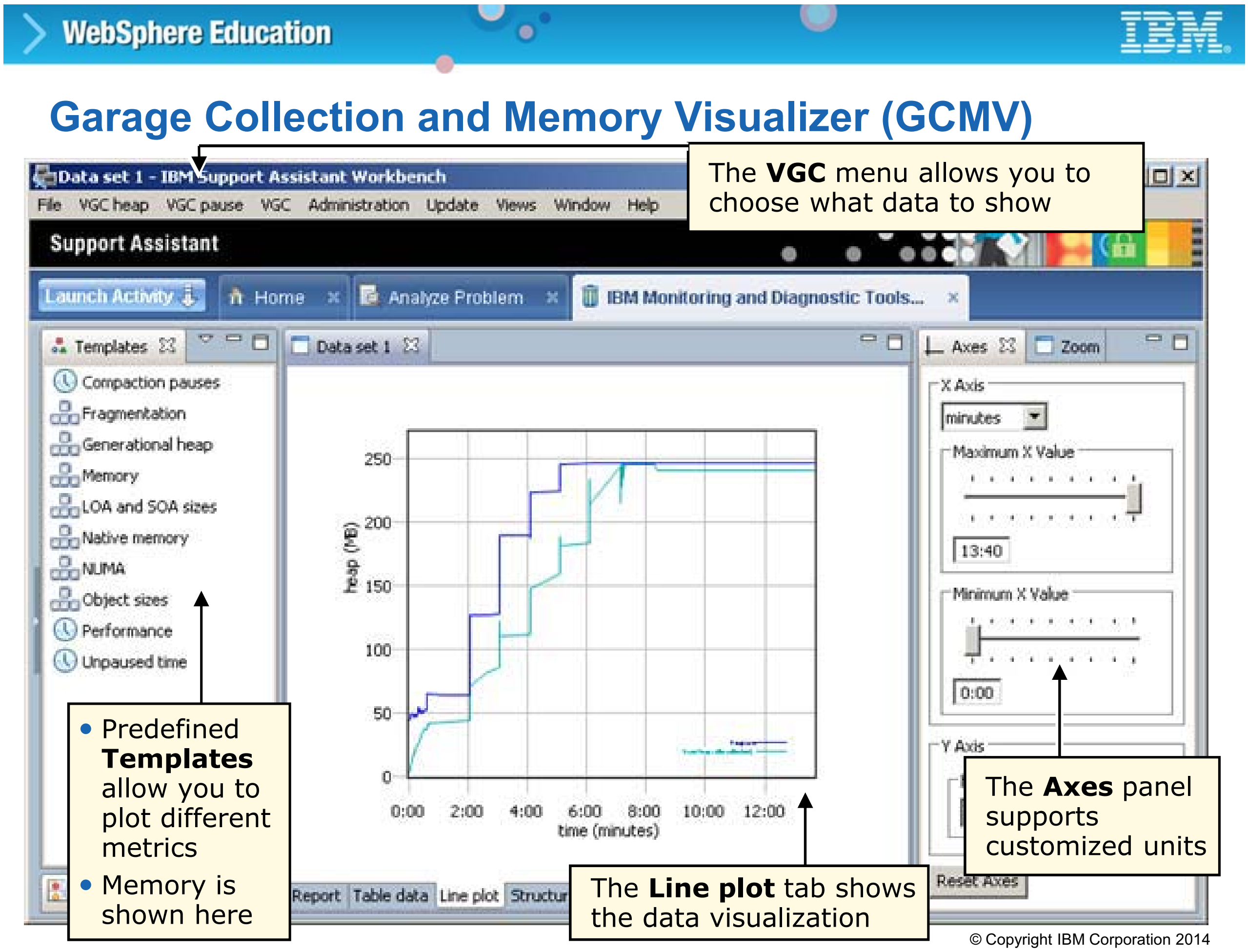


Figure 10-54. Garage Collection and Memory Visualizer (GCMV) WB868 / ZB8681.0

Notes:

WebSphere Education

Thread and Monitor Dump Analyzer (TMDA) (1 of 2)

- Import one or more javacore files into TMDA and view analysis

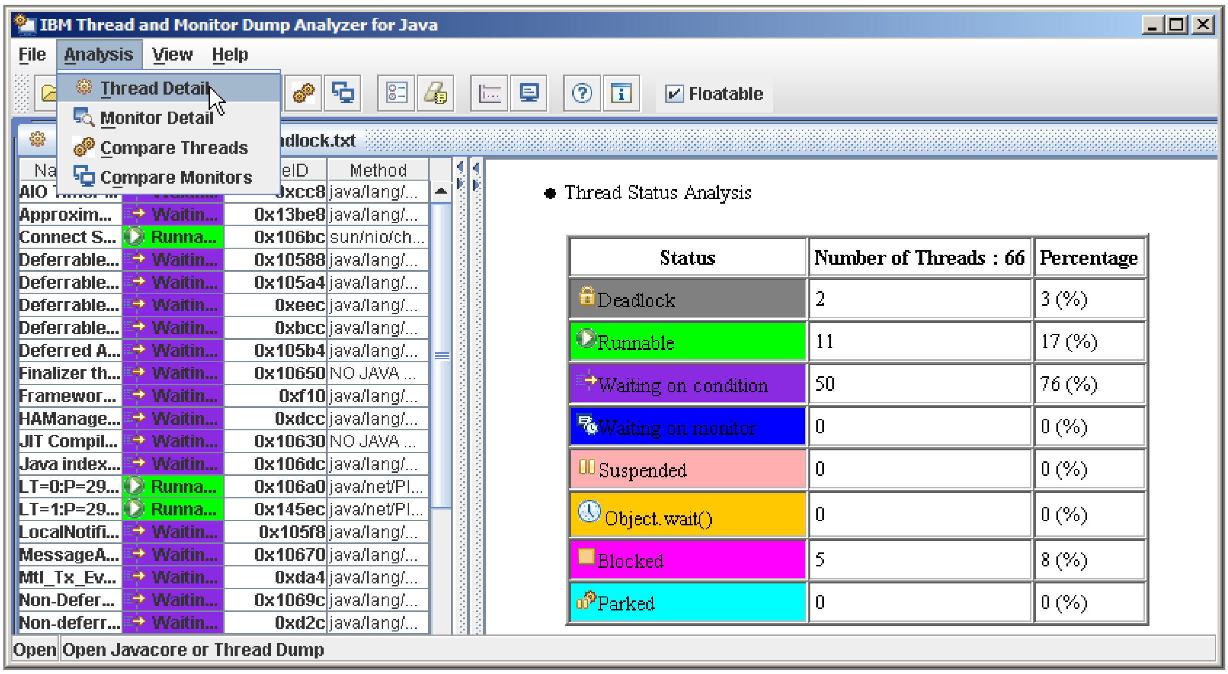


Figure 10-55. Thread and Monitor Dump Analyzer (TMDA) (1 of 2) WB868 / ZB8681.0

Notes:

WebSphere Education

Thread and Monitor Dump Analyzer (TMDA) (2 of 2)

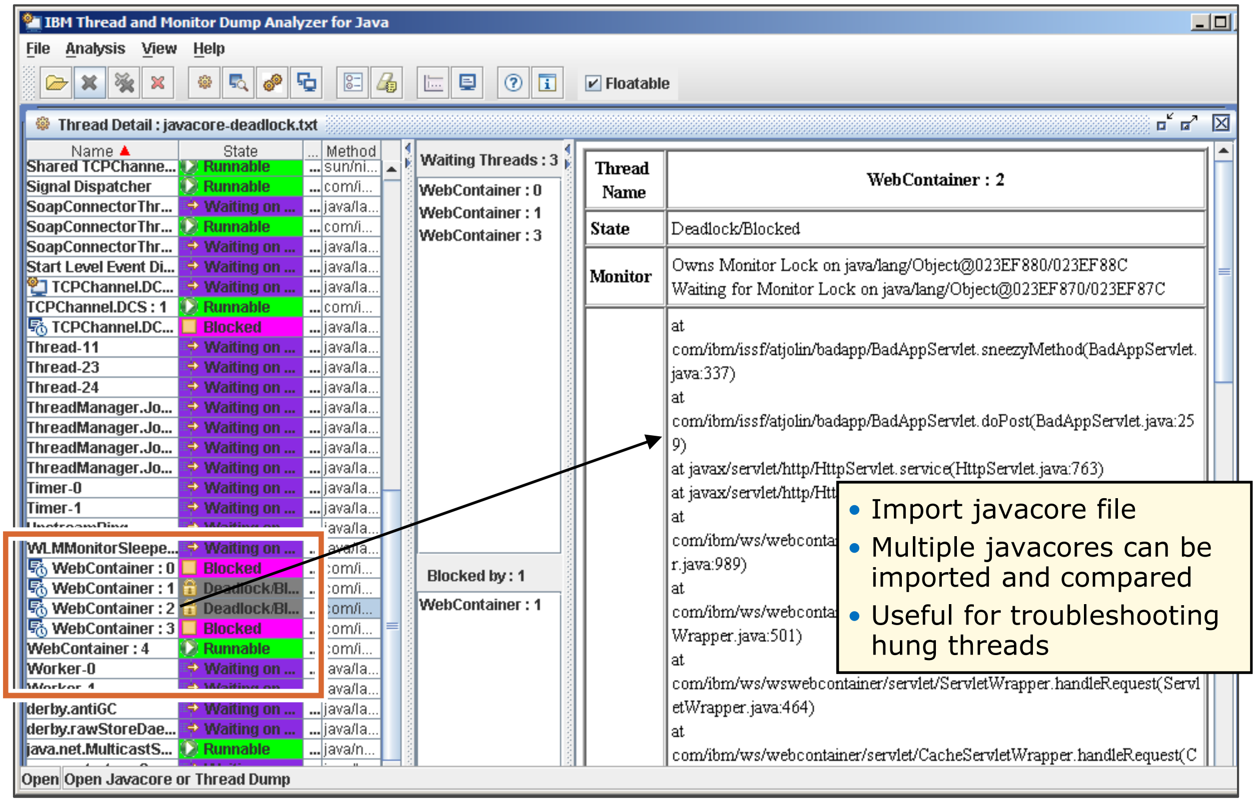


Figure 10-56. Thread and Monitor Dump Analyzer (TMDA) (2 of 2) WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Health Center

- The IBM Monitoring and Diagnostic Tools for Java - Health Center is a diagnostic tool for monitoring the status of a running JVM
  – Available in the IBM Support Assistant
- The Health Center uses a small amount of processor time and memory, thus a small impact on the application's performance
- The tool is provided in two parts:
  – The Health Center agent that is part of the WebSphere runtime and collects data from an application server
  – An Eclipse-based client that connects to the agent
- The client interprets the data and provides suggestions to improve the performance of the monitored application server

Figure 10-57. Health Center WB868 / ZB8681.0

Notes:

Health Center client

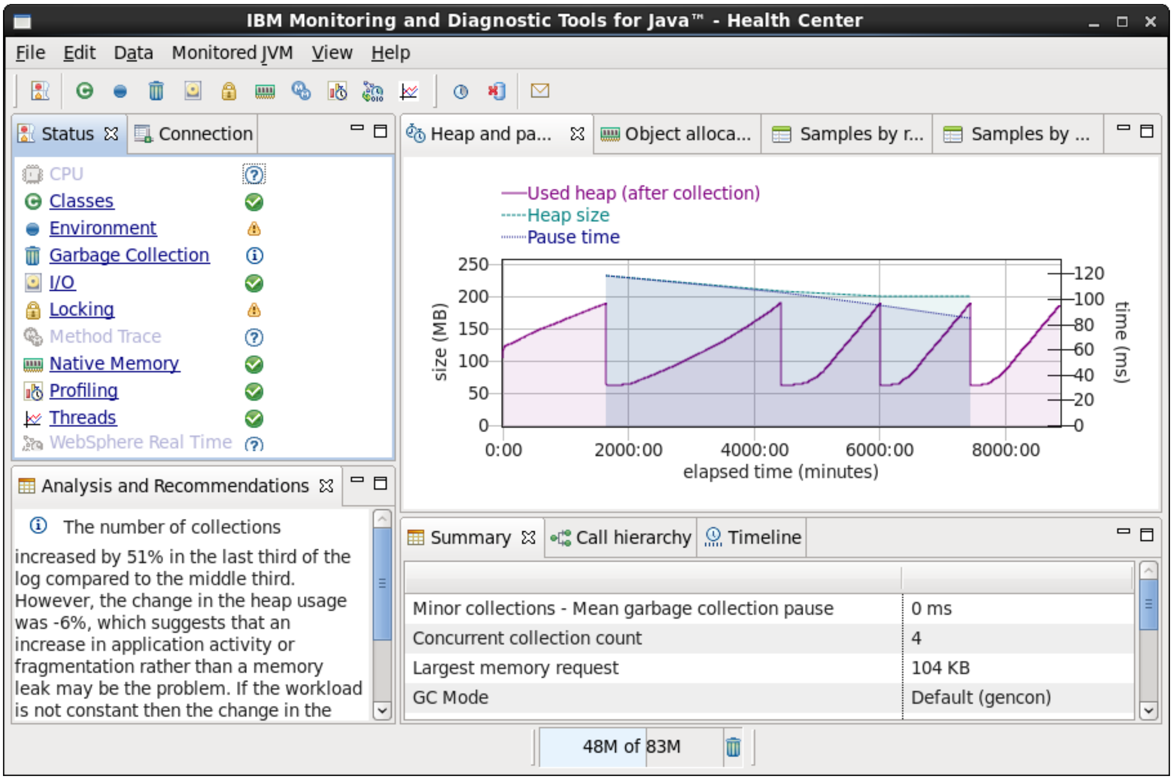


Figure 10-58. Health Center client WB868 / ZB8681.0

Notes:

WebSphere Education

A final word on performance tools

- The tools that are discussed in this unit are only a small subset of the tools that are used to monitor performance
- Investigate all available tools and use tooling that makes your team the most productive
- Use the right tool at the right time
  – Use the tool with which you are familiar
  – Use the tool that does not require extensive setup
  – Use the tool that is cost effective

Figure 10-59. A final word on performance tools WB868 / ZB8681.0

Notes:

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Describe important performance data for monitoring Business Process Manager environments
- Identify useful performance tools for monitoring performance data
- Use the Tivoli Performance Viewer tool to monitor the runtime performance of an application that is running
- Describe the performance tools available in the IBM Support Assistant

Figure 10-60. Unit summary WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Checkpoint questions

1. True or False: Tivoli Performance Viewer monitors the performance of the shared WebSphere Application Server resources. Thus, it does not monitor the BPEL activity.
2. True or False: Critical hardware performance data includes amount of physical memory, free disk space, and I/O channel utilization.
3. True or False: JVM performance data includes connection pool size, HTTP response time, and native heap size.
4. True or False: Some of the key OS metrics that the vmstat tool provides are gathered from CPU, memory, and paging.
5. True or False: There are currently no tools available for analyzing verbose garbage collection data. Analysis of verbose GC must be done manually.

Figure 10-61. Checkpoint questions WB868 / ZB8681.0

Notes:

Write your answers here:

1.

2.

3.

4.

5.

WebSphere Education

Checkpoint answers

1. False. Tivoli Performance Viewer in Business Process Manager has other categories that are specific to SCA components.
2. True
3. False: JVM performance data includes verbose GC data, thread dumps, and Java heap dumps.
4. True
5. False: There are several tools available. The Garbage Collection and Memory Visualizer (GCMV) is one example.

Figure 10-62. Checkpoint answers WB868 / ZB8681.0

Notes:

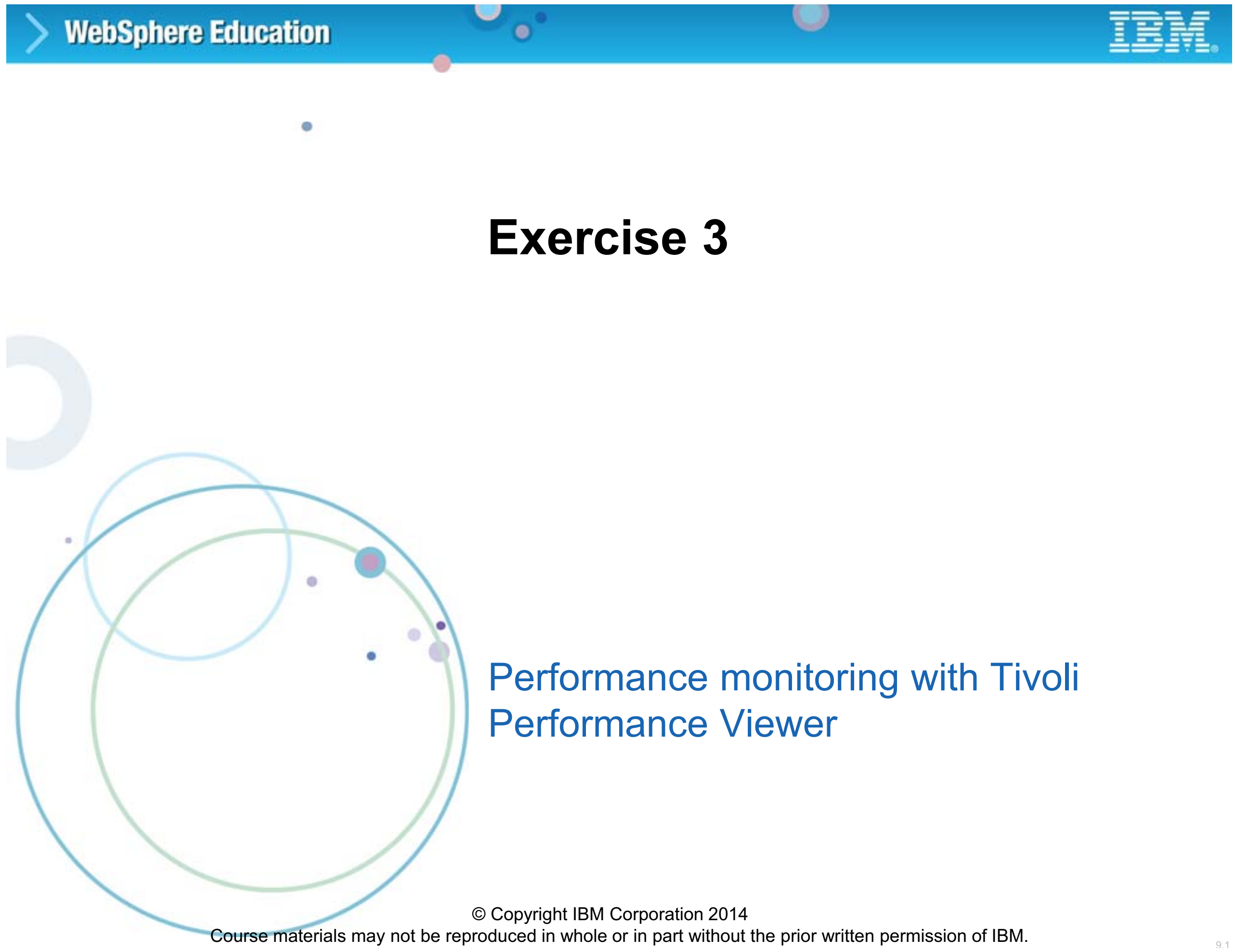


Figure 10-63. Exercise 3 WB868 / ZB8681.0

Notes:

WebSphere Education IBM.

Exercise objectives

After completing this exercise, you should be able to:

- Enable Performance Monitoring Infrastructure (PMI) on SCA components
- View the performance statistics by using the Tivoli Performance Viewer

Figure 10-64. Exercise objectives WB868 / ZB8681.0

Notes:

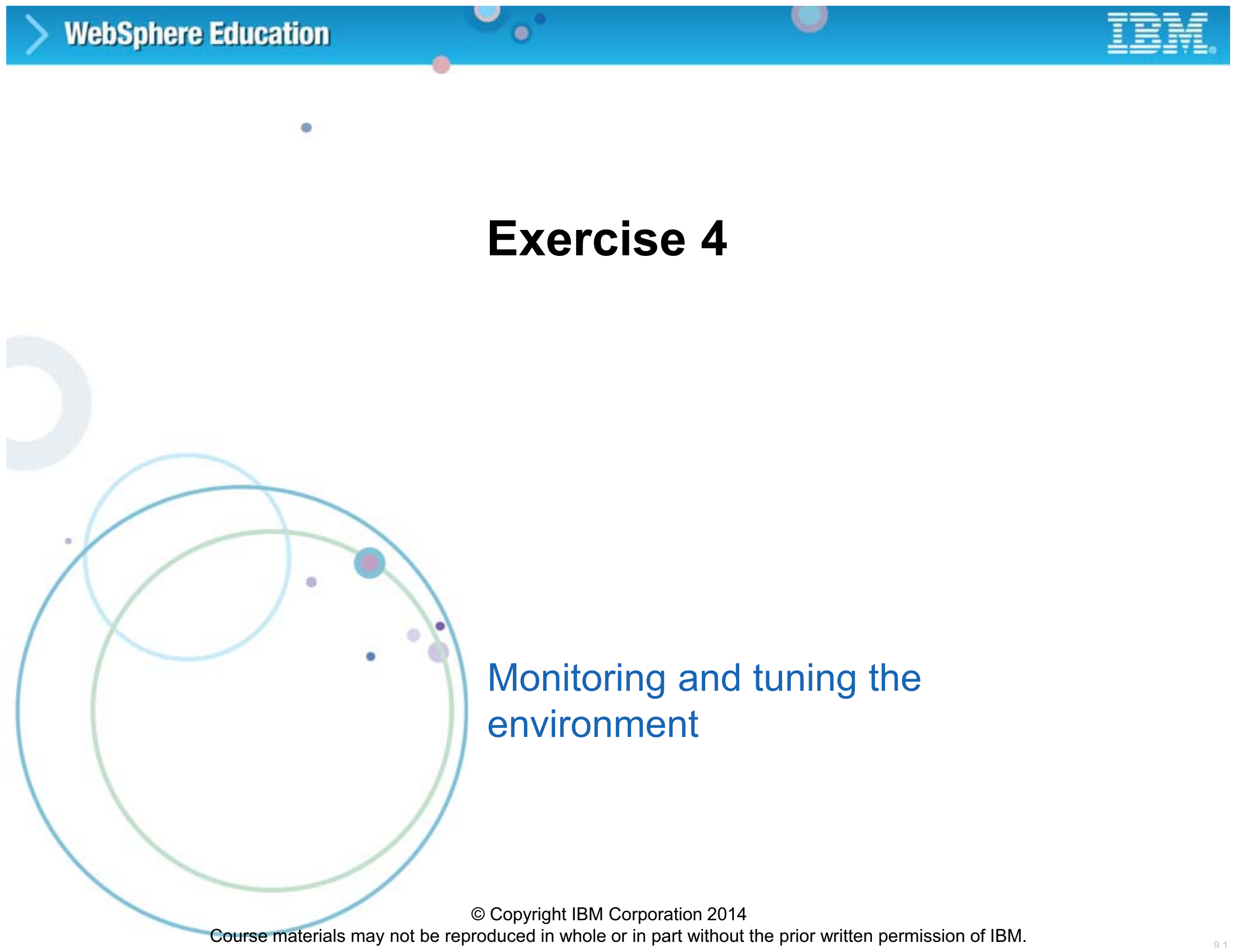


Figure 10-65. Exercise 4 WB868 / ZB8681.0

Notes:

WebSphere Education

Exercise objectives

After completing this exercise, you should be able to:

- Explore an application with various bindings by using IBM Integration Designer
- Examine various tuning parameters
- Tune the Event Manager
- Use operating system commands to monitor the environment
- Explore the `performanceTuning.properties` file
- Use the IBM Support Assistant Health Monitor tool to monitor the environment

Figure 10-66. Exercise objectives WB868 / ZB8681.0

Notes:

Unit 11. Performance problem determination

What this unit is about

This unit describes the IBM Business Process Manager performance-related problem determination methodology. You also examine the tools available to help you troubleshoot performance problems, how to detect and troubleshoot a hang condition, and preventive measures for the environment.

What you should be able to do

After completing this unit, you should be able to:

- Describe the nature of a performance problem
- Describe the types of data that is needed to troubleshoot performance problems
- Identify integration performance concerns
- Apply high-level problem determination methods to a performance problem
- Detect a hang condition
- Trigger and analyze javacore files for hangs
- Use the WebSphere Application Server hang detection facility
- Use the IBM Thread and Monitor Dump Analyzer for Java
- Identify preventive measures for the Business Process Manager environment

How you will check your progress

- Checkpoint
- Lab exercise

References

IBM Business Process Manager V8.5 IBM Knowledge Center

`http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/ditamaps/ic-homepage-bpm.html`

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Describe the nature of a performance problem
- Describe the types of data that is needed to troubleshoot performance problems
- Identify integration performance concerns
- Apply high-level problem determination methods to a performance problem
- Detect a hang condition
- Trigger and analyze javacore files for hangs
- Use the WebSphere Application Server hang detection facility
- Use the IBM Thread and Monitor Dump Analyzer for Java
- Identify preventive measures for the Business Process Manager environment

Figure 11-1. Unit objectives WB868 / ZB8681.0

Notes:

WebSphere Education

Topics

- Preparation: Before problems occur
- Organize the investigation
- Gathering data
- Cross-component trace capabilities
- Hung threads issue
- IBM Thread and Monitor Dump Analyzer
- Preventive measures

Figure 11-2. Topics WB868 / ZB8681.0

Notes:

11.1. Preparation: Before problems occur

When you open a PMR, the IBM support team asks for the MustGather data. The purpose of this topic is to explain what the MustGather data is, and what files are included in the MustGather data collection. For collecting some of the general MustGather data, the steps are described so that you can use some of the log files during troubleshooting.

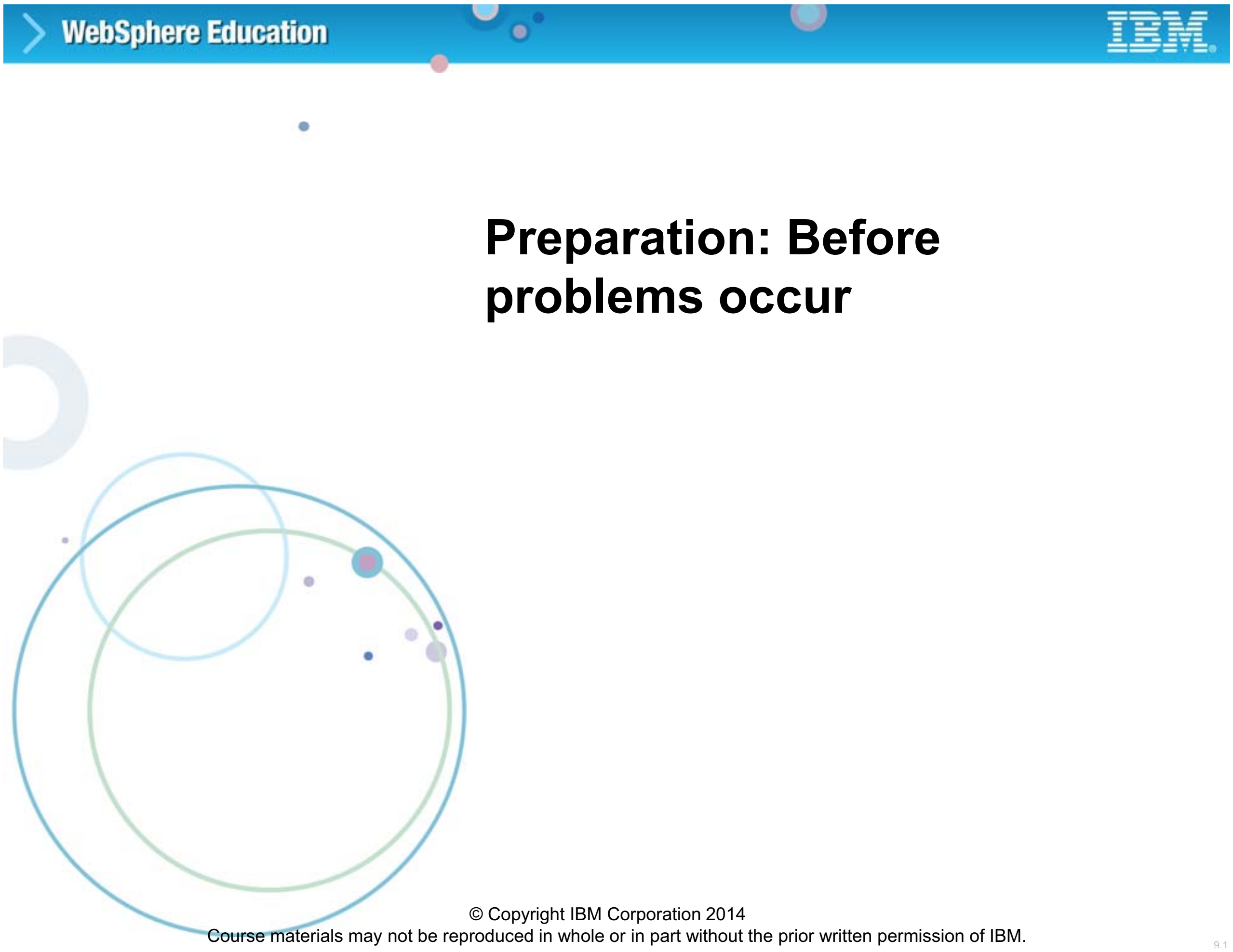


Figure 11-3. Preparation: Before problems occur WB868 / ZB8681.0

Notes:

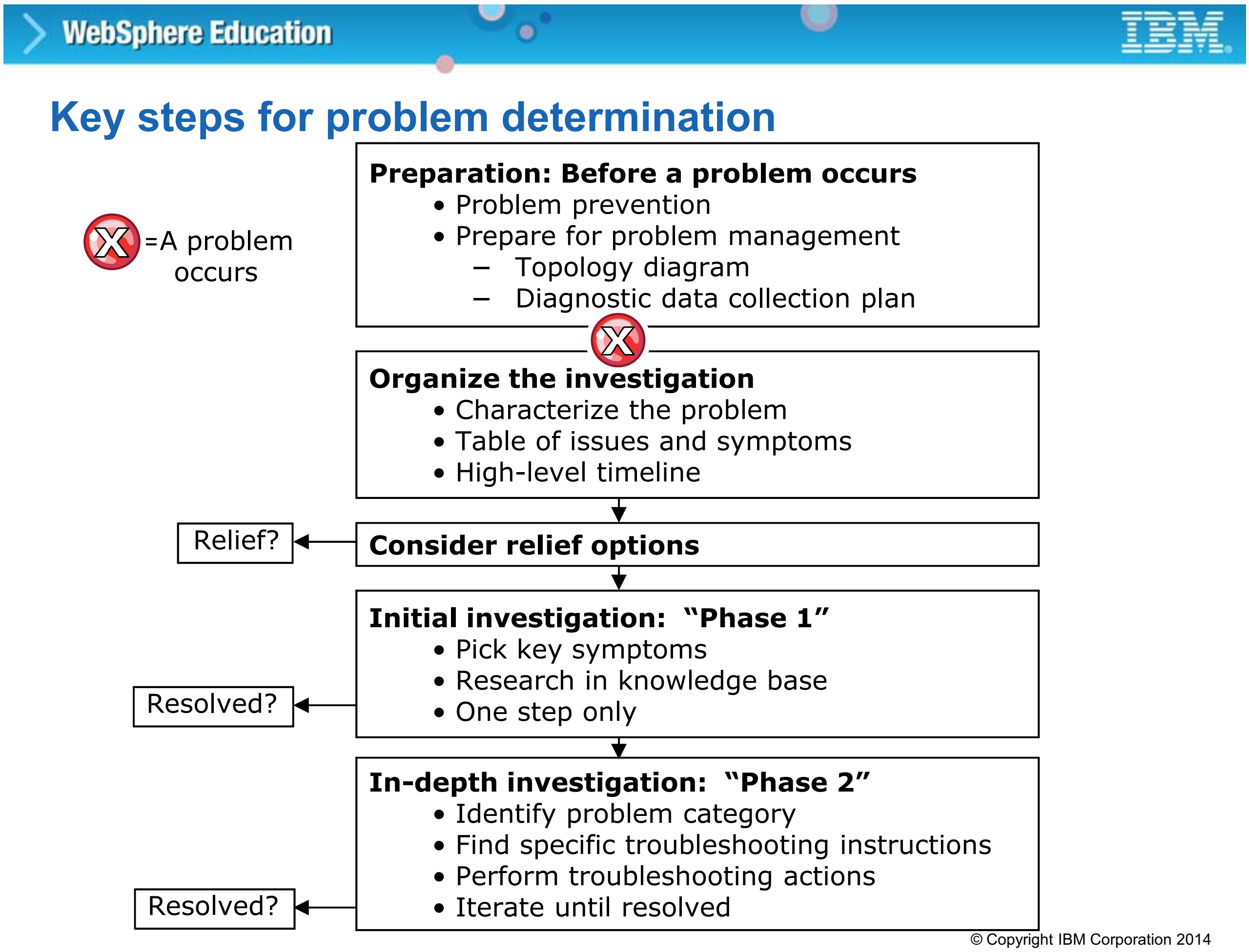


Figure 11-4. Key steps for problem determination WB868 / ZB8681.0

Notes:

The red X represents the point at which a problem occurs. The job of a troubleshooter does not begin after a problem occurs. Plenty of preparation work should be done well before a problem occurs, if you want to run a good IT shop and be able to do good problem determination when it becomes needed.

One pitfall that many troubleshooters fall into is that they see a problem (for example, the system fails), and they dive into long and complex analysis, maybe for days and weeks. Meanwhile, the system is down and users are seeking short-term relief. Some relief options are explained later in this presentation.

Relief is something that you should keep in mind always throughout the problem determination process, not just one time as suggested in this chart. In any troubleshooting situation, you have three goals:

- Quickly provide a temporary solution so that users can get back to work and not have the problem continue to affect them, while you spend more time in looking for the permanent solution.
- Find and implement the right, permanent solution.

- Try to make sure that a similar problem does not occur again in the future. Or, if it does, you want to be as prepared to deal with it as possible after what you learned from this problem.

WebSphere Education IBM

Prepare for effective production troubleshooting

- Good problem determination starts long before anything bad happens
- Implement problem prevention “best practices”
- Perform monitoring and problem detection
- Keep good system documentation
- Have a diagnostic data collection plan
- Have a relief or recovery plan
- Keep a maintenance plan: scheduled and emergency
- Keep a change log

Figure 11-5. Prepare for effective production troubleshooting WB868 / ZB8681.0

Notes:

See the developerWorks article: “The Support Authority: 12 ways you can prepare for effective production troubleshooting”:

`http://www.ibm.com/developerworks/websphere/techjournal/0708_supauth/0708_supauth.html`

Keep a change log:

An important aspect of most troubleshooting exercises is to figure out what is different between a working system and a broken one, and using a baseline is one way to help address the question. Another action that can help you determine system differences is to keep a rigorous log of all changes that are applied to the system over time. When a problem occurs, you can look back through the log for any recent changes that possibly contributed to the problem. You can also map these changes to the various baselines that were collected in the past to ascertain how to interpret differences in these baselines.

Your log should at least track all upgrades and software fixes that are applied in every software component in the system, including both infrastructure products and application code. It should also track every configuration change in any component. Ideally, it should also track any known changes in the pattern of usage of the system; for example, expected increases in load or a different mix of operations that users invoke.

In a complex IT environment where many teams contribute to different parts of the environment, the task of maintaining an accurate, up-to-date, and global change log can be surprisingly difficult. You can use tools and techniques to help this task, from collecting regular snapshots of the configuration with simple data collection scripts, which are used to collect diagnostic data, to sophisticated system management utilities.

It is important to know the concept of change control, and keeping a change log is generally broader than the troubleshooting arena. Change control is also considered one of the key best practices for managing complex systems to prevent problems, as opposed to troubleshooting them.

Problem prevention best practices are as follows:

- Providing a sufficient test environment
- Doing load or stress testing
- Capacity planning
- Keeping the system operating within the capacity plan
- Having a production traffic profile (network, for example)
- Having a process for rolling out changes into production
- Keeping a record of changes
- Doing application review and best practices
- Providing education
- Having a migration plan
- Having a current architecture plan

Monitoring: Often, problems go undetected for a long time, or get detected only indirectly through some secondary effect. You need tools and a plan to effectively detect problems or any potential anomalies when they emerge.

System documentation: When investigating a problem, you need plenty of information about the system, and especially about how to system works when there is no problem. This information should be collected in advance. There are two key devices for formalizing this information:

- A topology or flow diagram
- A series of baselines for the normal behavior of the system

Diagnostic data collection: When a problem does occur, you must quickly and effectively gather information that allows you to diagnose the problem. To do so, you need a plan that is set up *in advance* for what diagnostics you collect and how.

Relief or recover plan: When a problem occurs, one of the main priorities, independent of any investigation, should be to restore function to the users. The relief or recovery plan lays out, in advance, the steps that you undertake to restore this function, without knowing in advance exactly what problem is occurring.

Maintenance: Applying regular maintenance is one of the key factors to reduce the probability and impact of problems. The maintenance plan establishes how you do so regularly. In addition to regular scheduled maintenance, you also must make emergency changes or maintenance to the system, in response to a newly diagnosed problem. The emergency maintenance plan outlines how to do so safely and effectively.

Change log: Your log should at least track all upgrades and software fixes that are applied in every software component in the system, including both infrastructure products and application code. It should track every configuration change in any component. Ideally, it should also track any known changes in the pattern of usage of the system; for example, expected increases in load, or a different mix of operations that users invoke.

WebSphere Education

Monitoring and problem detection

- Often, problems go undetected for a long time, or get detected only indirectly through some secondary effect
- Monitoring tools and a plan are needed to effectively detect problems or anomalies when they emerge
  – Trade-off: Detect important events, but do not degrade performance
- Passive monitoring versus active monitoring
  – **Passive**: Examine logs, PMI statistics, heap size, verbose GC data
  – **Active**: Send a test transaction from end to end
- Watch low-level operating system and network metrics
  – Overall processor and memory usage for the entire server
  – Paging and disk I/O activity
  – Rate of network traffic between various components
- Be prepared to actively generate more diagnostic data when a problem occurs

Figure 11-6. Monitoring and problem detection WB868 / ZB8681.0

Notes:

Monitoring tools and a plan are needed to effectively detect problems or anomalies when they emerge. Monitoring is a trade-off. You want to detect important events, and yet not adversely impact the normal operation of the system. Monitoring is an entire technical area in itself, different from problem determination.

Passive monitoring can be done at all levels: network, operating system, application server, and application. Dependent systems such as databases and LDAP directories can also be monitored. Monitor the main system log files for errors and events. For example, you might detect application server restarts that indicate that the server is failing. Some tools for passive monitoring include the Tivoli Performance Viewer and IBM Tivoli Composite Application Manager for Application Diagnostics.

Active monitoring goes beyond passive monitoring. Periodically test the operation of the entire system from end to end. One technique that can be used is the pinging of system components, such as one server or one database connection. Another technique is end-to-end pinging: periodically sending an entire “dummy” transaction through the system and verifying that it completes. Some tools for active monitoring include IBM Tivoli Composite Application Manager for Transactions and web-based load-generating programs like Rational Performance Tester.

Be prepared to actively generate more diagnostic tests when a problem occurs. In addition to dealing with diagnostic artifacts that are present when an incident occurs, your troubleshooting plan should consider any additional explicit actions to take as soon as an incident is detected. You want these actions to take place before the data disappears or the system is restarted.

Here are some examples of explicit actions to generate more diagnostic messages:

- Actively trigger various system dumps, if they are generated automatically (such as Java dump, heap dump, system dump, WebSphere Application Server Diagnostic Provider dumps, or other memory dumps that various products and applications might provide). For example, when a system is believed to be “hung,” it is common practice to collect three consecutive Java dumps for each potentially affected JVM process.
- Take a snapshot of key operating system metrics, such as process states, sizes, and processor usage.
- Enable and collect information from the WebSphere Application Server Performance Monitoring Infrastructure instrumentation.
- Dynamically enable a specific trace, and collect that trace for a specified interval while the system is in the current unhealthy state.

WebSphere Education

System architecture or topology diagram

- Show all the components and all the main flows in the system
  - Identify the various troubleshooting points in the system
  - Identify computers, applications, databases, topology diagram, and network
- Be specific and detailed
  - Indicate software versions, releases, and fixes applied, server names, and IP addresses, if possible
- Useful for:
  - Communicating with all parties involved in the problem determination effort
  - Identifying discrepancies between the expected environment and the current reality
  - Identifying points where monitoring or health checking can be done
  - Identifying points where diagnostics data can be collected

Figure 11-7. System architecture or topology diagram WB868 / ZB8681.0

Notes:

One suggestion is to meet with all the parties involved in the deployment and operation of the system together to draw or redraw the system topology diagram.

- Often, the mere fact of having multiple groups compare their understanding of the topology of the system yields important clues and dependencies that the various groups maybe did not know about previously.
- Then, you can work together, from the big picture, to decide where you should start looking for anomalies in the system that might cause the current problem.

WebSphere Education

Establish baselines

- What does a “normal” system look like?
- What are the typical values for common system metrics?
  - Processor usage
  - Memory size
  - PMI statistics
- What is a typical load?
- What is a typical response time for various operations?
- What should you normally see in the logs during operation of the system?
- Understand and document the system baseline

Figure 11-8. Establish baselines WB868 / ZB8681.0

Notes:

Much of what you do for problem determination is to observe various aspects of the behavior of the system. Then, try to determine whether these behaviors are normal or if they represent a symptom of a possible problem.

In many actual systems, it is not unusual to see various benign “errors” (both WebSphere Application Server and application-level) during normal operation. Learn to recognize them, or better yet, eliminate as many of these benign errors from the implementation of the system as possible.

Examples of baseline information are as follows:

- Copies of the various log files and trace files, over a representative time period in the normal operation of the system, such as a full day.
- Copies of a few Java dumps, heap dumps, system dumps, or other types of artifacts that are normally generated “on demand.” You can combine this activity with the earlier recommendation to test the generation of these artifacts on a system that is working correctly, before a problem occurs.
- Information about the normal transaction rates in the system and response times.

- Various operating system level statistics on a correctly running system, such as processor usage, memory usage, and network traffic.
- Copies of any other artifacts, information, or normal expected results from any of the special diagnostic collection actions, suggested earlier, for each anticipated type of problem.

11.2. Organize the investigation

WebSphere Education

IBM

Organize the investigation

9.1

Figure 11-9. Organize the investigation

WB868 / ZB8681.0

Notes:

WebSphere Education

Solving performance problems

An iterative process:

1. Load test the system
2. Monitor and collect performance data
3. Identify bottlenecks
4. Tune parameters to eliminate the most severe bottleneck
5. Repeat

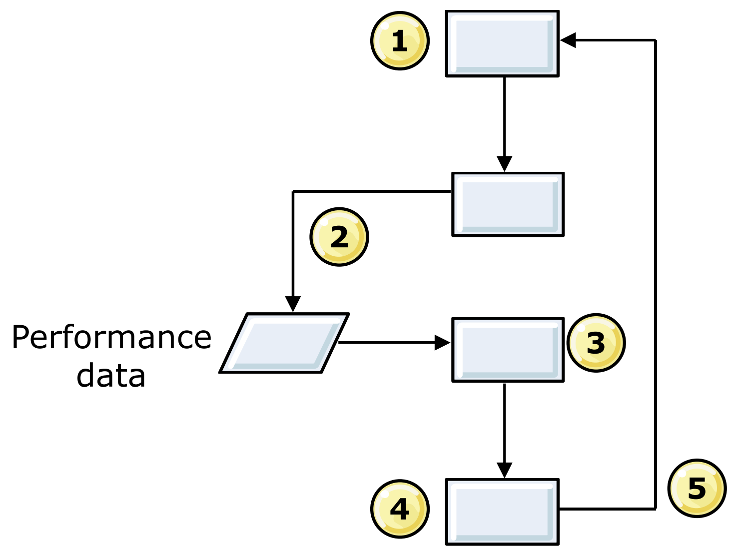


Figure 11-10. Solving performance problems WB868 / ZB8681.0

Notes:

There is a good summary in G. Polya, “How to Solve It”, Second edition, Princeton University Press, 1957, ISBN 0-691-08097-6, at the following website:

`http://www.math.utah.edu/~alfeld/math/polya.html`

It explains a basic methodology for problem solving.

WebSphere Education

The lifecycle of performance monitoring and tuning

- Load testing with the stress tools and benchmarking applications
  - Rational Performance Tester, Jmeter, and others
- Monitor and collect performance data
  - PMI, request metrics, IBM Tivoli for Composite Application Management
- Displaying server data
  - Tivoli Performance Viewer, performance monitoring servlet, user-written PMI clients
- Analyzing server data
  - Performance advisors
- Tuning the servers
  - IBM Support Assistant tools
  - Health Center

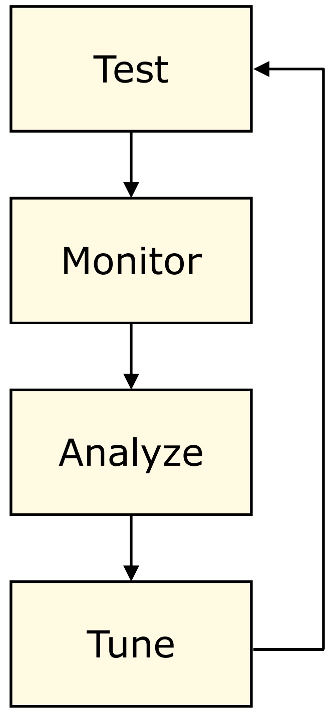


Figure 11-11. The lifecycle of performance monitoring and tuning WB868 / ZB8681.0

Notes:

The lifecycle of performance monitoring and tuning includes the following steps:

- Load testing with the stress tools and benchmarking applications by using tools such as Rational Performance Tester, JMeter, Load Runner, DayTrader.
- Collect application server data by using PMI, request metrics, and IBM Tivoli for Composite Application Management.
- Display application server data by using Tivoli Performance Viewer, performance monitoring servlet, and user-written PMI clients.
- Analyze application server data. The Tivoli Performance Viewer Performance Advisors might be helpful.
- Tune the application servers. The Java Health Center might be helpful.

WebSphere Education

Get an overall nature of the performance problem

- Take the time to carefully understand the problem and its context
- Listen and ask questions
  - In many cases, asking questions is all that it takes to solve simple problems
  - For complex problems, failure to do so results in considerable delays
- Ask: **What? Where? When?** and **Why?**
- Develop a clear and specific description of **what** happened, error messages, observed abnormal behavior, and other symptoms
- **Where** specifically are you seeing performance problems?
  - Authoring environment
  - User experience: launching a BPD, progressing to the next coach in a flow
  - Custom solution code
  - Impacting all areas
- **When** exactly did the problem occur?
- Consider **why** this problem occurred here and now, and not in the past

Figure 11-12. Get an overall nature of the performance problem WB868 / ZB8681.0

Notes:

Before you start troubleshooting a performance problem, you must assess whether it impacts a specific component or the entire environment.

In many cases, you might be surprised to discover how often careful consideration of these questions gets you very close to solving the problem, without even needing much further investigation.

And, in the rest of the cases, giving careful consideration to these questions can avoid much of the confusion, miscommunication, and false starts that often plague complex problem investigations.

WebSphere Education IBM

Quantify the performance issue

- How long does a specific action take?
- Do you have logging that shows where the time is being spent?
- Do you have performance tools to help analyze or collect data?
- Is it a new problem or have you always had it in this environment?
- Do you see this same problem in other environments on the network?
- Is the performance problem consistent or does it change?
- Is performance is worse at certain times of day?
- Is performance is bad under load?
- Does performance improve after a restart of the application server?

Figure 11-13. Quantify the performance issue WB868 / ZB8681.0

Notes:

11.3. Gathering data

When you open a PMR, the IBM support team asks for the MustGather data. The purpose of this topic is to explain what the MustGather data is, and what files are included in the MustGather data collection. For collecting some of the general MustGather data, the steps are described so that you can use some of the log files during troubleshooting.

WebSphere Education IBM

Gathering data

Figure 11-14. Gathering data WB868 / ZB8681.0

Notes:

WebSphere Education

Initial investigation (1 of 2)

The goal at this stage is to:

- Locate known problems and solutions
- Get a starting point for further investigation if necessary

Make an inventory of all pertinent anomalies and potential symptoms:

- Errors, warnings, exceptions, out-of-range statistics, any other unusual behavior
  - Scan (grep) through available logs
- Ideally, you should check everything
- Research that is guided by understanding the flows in the topology diagram
- Integrate into the table of issues and symptoms
- Assess and prioritize symptoms
- Not a perfect science; hard to tell which symptom is a cause, and which symptom is a consequence of the original problem
- Use the topology diagram and baselines prepared earlier

Figure 11-15. Initial investigation (1 of 2) WB868 / ZB8681.0

Notes:

Make sure that the problem is not a well-known issue. Do the research first.

WebSphere Education

Initial investigation (2 of 2)

- Build a low-level, detailed timeline of events for one incident to clarify what might be the cause of what
  - Include pertinent but normal events, in addition to anomalies and symptoms
  - Monitor multiple components in parallel and attempt to correlate
- Research the top symptoms in the WebSphere Knowledge Base
  - Search for relevant IBM technotes
  - Search for APARs
- Answer and verify a favorite question of all troubleshooters: **What changed recently?**
- Clearly communicate between the various parties that are involved in the troubleshooting task
  - Inside your organization
  - When trying to explain a complex environment to IBM support

Figure 11-16. Initial investigation (2 of 2) WB868 / ZB8681.0

Notes:

If you are lucky, at this stage you find an article in the knowledge base that addresses the problem or symptoms you are seeing, and then you can apply the suggested solution.

Authorized program analysis report (APAR) tracks software defects reported by customers.

WebSphere Education

Gathering information

- Gather information about the environment
  - Hardware specifications of IBM Business Process Manager servers (amount of CPU, RAM)
  - Clustered, type of topology?
  - Hardware specifications of the database server
  - VM or not? (IBM Business Process Manager server and database server)
  - Other hardware that is being used

- Gather information about the software
  - IBM Business Process Manager version, any interim fixes, fix packs that are installed
  - Version of the operating system
  - Database server, what is the RDBMS, what is the version
  - VMware or any other VM, what is the version, what type

Figure 11-17. Gathering information WB868 / ZB8681.0

Notes:

WebSphere Education

Gathering resource consumption information

- Performance monitor logs from the operating system level
  - Examine disk space, CPU load, disk I/O, and other details
- Garbage collection details
- Database server resource consumption logs
  - Snapshot monitor on DB2
  - AWR report on Oracle
  - Dashboard report on MSSQL
- Is the database a shared resource or not?
- Examine basic WebSphere configuration settings
  - Data source configuration, maximum connection, and others
- Did you include application logging in the process application or SCA applications?

Figure 11-18. Gathering resource consumption information WB868 / ZB8681.0

Notes:

WebSphere Education

Types of data to gather for performance-related problems

- Thread dumps: Give a runtime snapshot of all the threads that are running at the time when the problem or slowness occurs
- Heap dumps: Needed only in case there is an out-of-memory problem
- Java GC logging: Can help in understanding your application in terms of:
  - Memory usage (object count and size)
  - Heap size (initial and over time)
  - GC metrics (frequency, pause time, and amount freed)
- Solution code logging: Custom logging in the solution code helps to determine where the performance problems lie
- Instrumentation logging: Writes logging data and timing in a lightweight binary format that has minimal to no impact on the performance of the system

Figure 11-19. Types of data to gather for performance-related problems WB868 / ZB8681.0

Notes:

With solution code logging, custom logging in the solution code helps to determine where the performance problems lie. It not only helps identify areas in the solution code that might benefit from performance tuning but also helps highlight the IBM BPM product areas to focus on during diagnosis.

Instrumentation logging writes logging data and timing in a lightweight binary format that has minimal to no impact on the performance of the system. These logs are writing threads also; you might compare threads in thread dumps with these logs, and it gives you a good picture of what is happening.

WebSphere Education

What should you do for BPM problems? (1 of 3)

- Use the Event Manager
  - Examine the Event Manager parameters that are configured
  - Look for any long-running tasks that are in an executing state
  - A number of tasks can show up there, and the type of task gives a hint to the problem
- Use the Process Monitor
  - Indicates whether you have a service or a BPD in a loop
  - Get a list of active services and processes that are currently running and consuming CPU
  - Find out how long a step of a process takes and the most complex process
- Examine the settings in the `TeamworksConfiguration.running.xml` file
  - Look for any Event Manager related parameters
  - Check the cache, snapshot, and interval parameters

Figure 11-20. What should you do for BPM problems? (1 of 3) WB868 / ZB8681.0

Notes:

After monitoring the operating system, start with the event manager. Look for any long-running tasks that are in an executing state. A number of tasks can show up there, and the type of task gives a hint to the problem. BPDNotification and DBNotification are tasks in the BPD engine. They should generally execute in a small amount of time. SystemTask and UCATask are generally services. If these services are long-running, then the associated services are likely “stuck”. If you see items rapidly popping up and being completed, you might be in a loop that is moving between the BPD engine and the service engine. For instance, you might have a system lane task that is failing. If you have a catch exception on that system lane activity that loops back to itself, then the system task fails. Go back to the BPD engine in the catch, and then immediately back to the service engine to fail again.

The Process Monitor tells you if you have a service or a BPD in a loop. If you go there, you see a list of active services and processes. Active services and processes are jobs that are currently running and consuming CPU. You can click them to find out which service or BPD is involved and then find out what step (or steps) you are stuck in. If you find a loop in the Process Monitor or Service Monitor, it is likely that the loop is what is causing CPU and memory usage. If you do not find a loop, you need to look elsewhere.

Thread dumps are the right next step.

WebSphere Education

What should you do for BPM problems? (2 of 3)

- Use the Instrumentation Monitor
  - Display and collect instrumentation data
  - Useful for identifying BPMN process instance performance bottlenecks
  - You can see whether functions such as EJB API, caches, and database queries are taking longer than usual
  - Allows you to log instrumentation data and play later
- It is always good to examine Process Monitor and Instrumentation Monitor data together
- Explore the Process Inspector
  - Provides information about events and processes in your entire system
  - You can view detailed activity information about process instances and perform troubleshooting tasks
  - Interfaces via Process Admin Console and integrated in Process Designer

Figure 11-21. What should you do for BPM problems? (2 of 3) WB868 / ZB8681.0

Notes:

For a BPMN-based application (developed in the Process Designer), you can turn on process instrumentation so that you can see which services and processes take the longest time to complete.

Business Process Manager provides the integrated Process Inspector as part of Process Designer. With the integrated Process Inspector, you can debug a process in Process Designer. When you select the **Inspector** tab on the Process Designer view, it switches to the Process Inspector. Click **Start** to start a process, simulate it in Process Designer, and monitor the state of the process.

WebSphere Education

What should you do for BPM problems? (3 of 3)

- Use the Failed Event Manager
  - Administered through the Failed Event Manager application in the BPM Advanced administrative console
- Failed Event Manager monitors and logs failed events for:
  - Runtime faults of asynchronous SCA, JMS, or WebSphere MQ invocations
  - Long-running BPEL process failures
  - Business Flow Manager infrastructure failures

Figure 11-22. What should you do for BPM problems? (3 of 3) WB868 / ZB8681.0

Notes:

WebSphere Education

What if you are running low on heap?

- At a high level, when you run low of Java memory (heap space) your CPU suddenly spikes and your application stops responding
  - Java garbage collection is expensive when you are running low on memory and it can block the other threads in the JVM and prevent them from moving
  - Might be due to a process or service that is running in a loop
  - Get heap dumps
- Garbage Collection logging can also be useful
  - GC logs give you the ability to graph memory utilization over time
  - Spot trends such as if the heap grows slowly you might have a leak
  - If it spikes every day at 3 p.m., you might have a user who is running a report that is consuming too much memory
- Examine basic WebSphere settings
- Analyze the solution code that is causing the issue

Figure 11-23. What if you are running low on heap? WB868 / ZB8681.0

Notes:

At a high level, when you run low on Java memory (heap space), your CPU suddenly spikes and your application stops responding. This response happens because Java garbage collection is expensive when you are running low on memory, and it can block the other threads in the JVM and prevent them from moving. One possibility for running out of heap is an IBM BPM process that is running in a loop. You probably already ruled that out in the process monitor step that is described previously, so now you need heap dumps. You can use heap dumps to see what was in memory when you saw the problem.

Garbage Collection Logging can also be useful. GC logs give you the ability to graph memory utilization over time. These logs can be helpful to spot trends. For instance, if the heap grows slowly, you might have a leak. If it spikes every day at 3 p.m., a user might be running a report that is consuming too much memory.

WebSphere Education IBM

Other log analysis

- Instrumentation log analysis
  - These files are organized by thread, not by time; when you transform them to test, you get the timestamps in your local time zone
  - Look at the time stamps, not just the ms values that are recorded in the text of the log
  - Start with looking at the instrumentation periods that take 1000 ms or more (4 or more decimal digits)
  - Compare instrumentation logs threads with thread dump threads
- IBM Java cores generation and analysis
  - Generate a thread dump if needed
- Regular log analysis
  - Hung threads and `SystemOut.log`
  - Compare hung threads with what you have in thread dumps
  - Compare problematic periods in the logs (where many errors are thrown, for example) with the instrumentation logs

Figure 11-24. Other log analysis WB868 / ZB8681.0

Notes:

Remember that threads in the logs are the same as threads in the thread dumps, so if you see a suspicious thread in the logs, check for the same thread in all of the thread dumps.

Instrumentation logs do not affect performance. Gathering such a log file is expensive only from the disk space standpoint, and you would not want to leave it turned on for 24 hours. Typically for performance issues, you need to gather instrumentation logs and Java cores for a short time, but again, instrumentation logs do not affect performance of the server as traces would. By enabling “wle all” on a badly performing server, you would likely stop it, whereas with instrumentation logs there would be no administrative cost as with verboseGC in WebSphere Application Server.

11.4. Cross-component trace capabilities

This topic introduces the basic concepts of troubleshooting performance problems. The key elements that affect runtime performance are introduced.

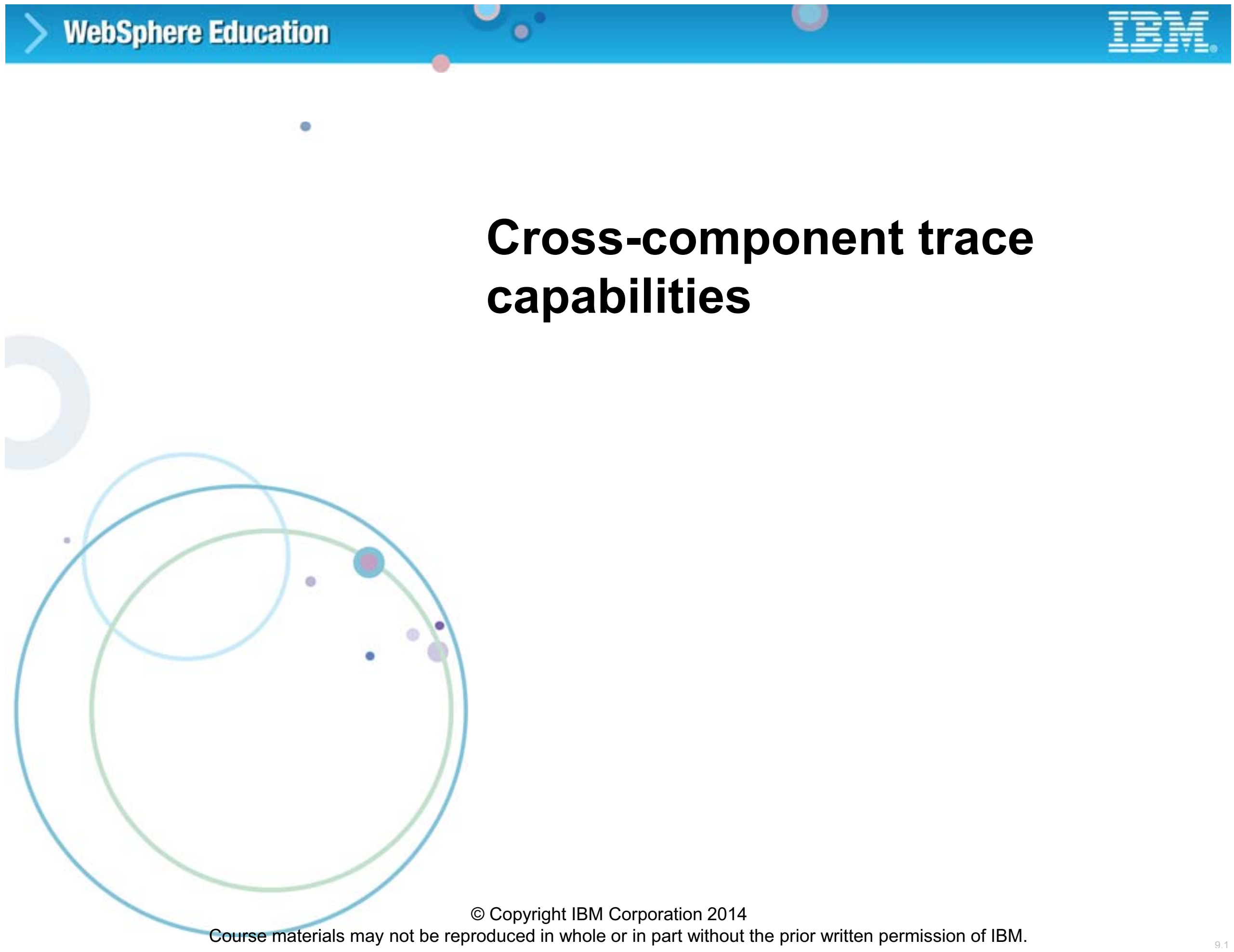


Figure 11-25. Cross-component trace capabilities WB868 / ZB8681.0

Notes:

WebSphere Education

What is cross-component trace (XCT)?

- XCT is a feature that annotates the logs so that entries that are related to a request are identified as belonging to the same unit of work
  – The request might traverse more than one thread, process, or server
- XCT helps identify the root cause of problems across components, which provides the following benefits:
  – Enables administrators and support teams to follow the flow of a request from end to end as it traverses thread or process boundaries, or travels between stack products and WebSphere Application Server
  – Helps to resolve questions about which component is responsible for a request that fails
- Do not use XCT in production or while obtaining performance data
  – Incurs significant performance resource usage

Figure 11-26. What is cross-component-trace (XCT)? WB868 / ZB8681.0

Notes:

Depending on the nature of your applications, multiple threads within an application server can be used to handle requests, such as HTTP requests or JMS requests. More than one application server might handle some requests, such as when one application server makes a request to another application server for a web services request.

Applications that are built by using distributed architectures, such as service-oriented architecture, can benefit from XCT, since XCT helps facilitate problem determination across multiple services on different systems.

WebSphere Education

Cross-component trace capabilities

- Cross-component trace (XCT) maps **`SystemOut.log`** and **`trace.log`** records back to the SCA programming model
  - Logs message process sequence from module to module and the entering time and exiting time of an SCA invocation
- Supports all SCA call patterns
  - Asynchronous one way
  - Asynchronous with callback
  - Asynchronous with deferred response
  - Synchronous call
- Works with business objects, or simple data types
- Supports a network deployment environment
  - Load files from each server
  - Might load only some of the files from all the servers
- Create IBM Integration Designer Test Client trace from log record data

Figure 11-27. Cross-component trace capabilities WB868 / ZB8681.0

Notes:

The cross-component trace (XCT) is available only in IBM BPM Advanced. XCT maps `SystemOut.log` and `trace.log` records back to the SCA programming model by using Integration Designer.

WebSphere Education

Use cross-component trace for problem determination

- Supports Business Process Execution Language (BPEL) microflows
- Support for long-running business processes
  - Follow flow and correlate log records to long-running processes
  - Supports multiple “pick” and “receive” activities
  - Follow internal thread work
  - Follow callouts to partners
- Supports HTTP, JMS, WebSphere MQ, and MQ JMS bindings
  - Another step toward end-to-end call chains
  - Captures work done as part of export: Function selectors and data bindings
- Rolling log support
  - SystemOut and trace files are often split over multiple files
  - Load all the files in for the full call chain

Figure 11-28. Use cross-component trace for problem determination WB868 / ZB8681.0

Notes:

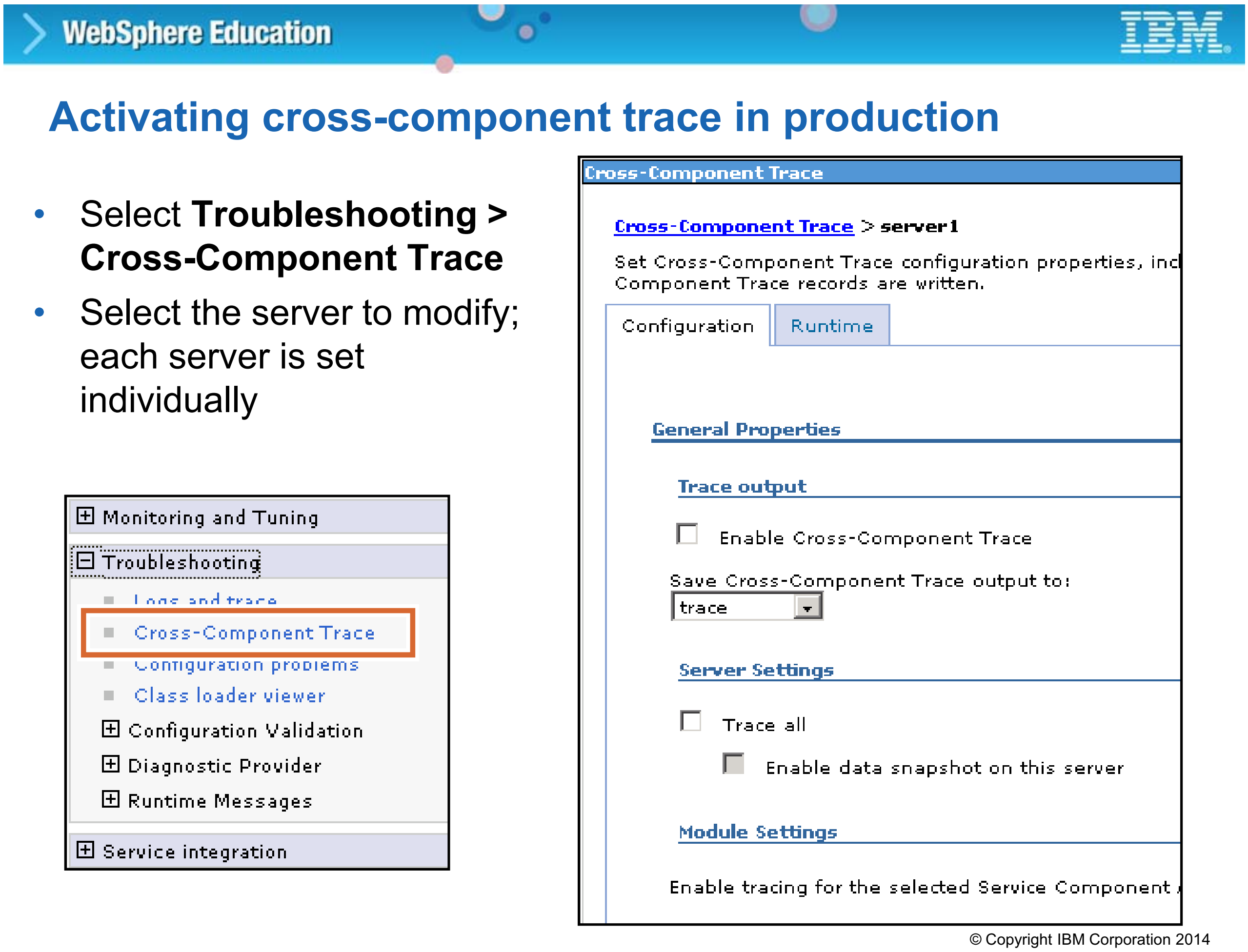


Figure 11-29. Activating cross-component trace in production WB868 / ZB8681.0

Notes:

Enable XCT to include request IDs in log and trace files when you want to see which log and trace entries, in all threads and application server processes, are related to the same request. Request IDs are recorded only when using HPEL log and trace mode, and they can be seen or used for filtering by using the `logViewer` command.

Enable XCT to create correlation log records when you want to log how requests branch between threads and processes, and see extra information about each request. Enabling XCT to create correlation log records might have a significant performance impact on your system, so is best suited to test and development environments.

Enable XCT to capture data snapshots when you want to store entire request and response bodies to the file system. Enabling XCT to capture data snapshots might have a significant performance impact on your system, so is best suited to test and development environments. XCT captures data snapshots for message requests and responses that the SIBus handles.

WebSphere Education

Activating cross-component trace in testing

- Cross-component tracing allows developers to examine flows in a business process step by step
- When you configure cross-component trace with data snapshot, generated invocation records contain the input and output data that was passed between the components during processing

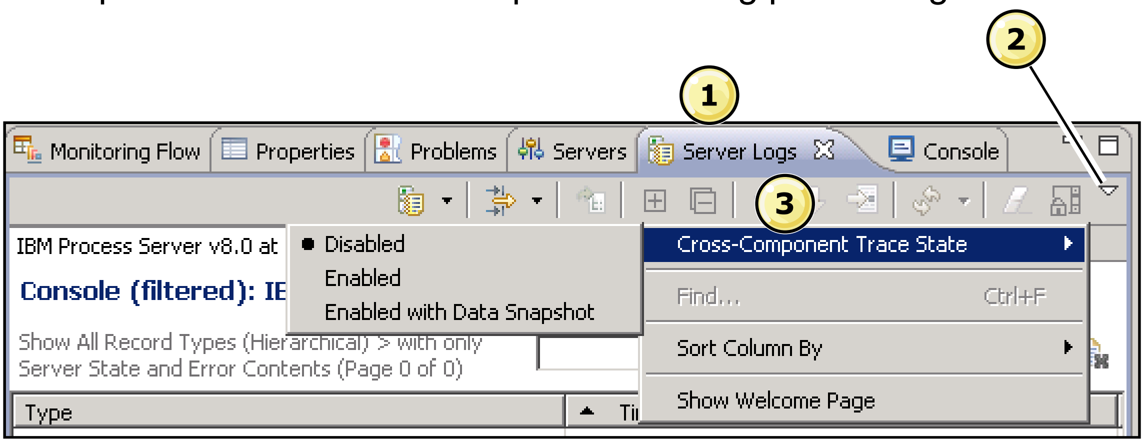


Figure 11-30. Activating cross-component trace in testing WB868 / ZB8681.0

Notes:

By default, the Server Log view displays standard server console and log records. However, if you enable cross-component tracing, the Server Log view also displays invocation records that can contain the invocation data that is passed between the components in your application. The invocation records are displayed in hierarchical format in the Server Log view, which allows you to more easily understand the relationships that exist between the records. When you enable cross-component tracing, the Server Log view becomes an even more powerful tool for problem determination.

When you enable cross-component tracing on a server, invocation records are generated during SCA processing of modules and components. The invocation records include information about any errors or events that occur during processing, such as runtime exceptions. If you choose to enable cross-component tracing with the data snapshot feature, the generated invocation records also contain the invocation input and output data that is passed between the components during processing.

You can enable or disable cross-component tracing for a server from either the Server Log view or the server administrative console. If you enable cross-component tracing from the Server Log view, the tracing is enabled only during the server session. When you next stop or restart the server, the cross-component trace state is automatically disabled by default. By comparison, if you enable cross-component tracing for a server from the server administration console, the cross-component tracing remains enabled for all sessions of the server until you choose to disable it again.

When you enable or disable cross-component tracing, you can choose from one of the following options:

- **Disabled**: This option disables cross-component tracing. No invocation records are generated in the server console and logs.
- **Enabled**: This option enables cross-component tracing. Invocation records are generated in both the server console and the `SystemOut.log` and `trace.log` files, but the record properties do not include any invocation input and output data. The `SystemOut.log` and `trace.log` files are in the server `log` directory.
- **Enabled with data snapshot**: This option enables cross-component tracing with the data snapshot feature. Invocation records are generated in both the server console and the `SystemOut.log` and `trace.log` files, and the record properties include invocation input and output data. This data is captured in input and output files under `logs\xfa`.

11.5. Hung threads issue

One of the common runtime issues is hung thread. This topic discusses how to troubleshoot a potential hung thread problem.

WebSphere Education

IBM

Hung threads issue

9.1

Figure 11-31. Hung threads issue WB868 / ZB8681.0

Notes:

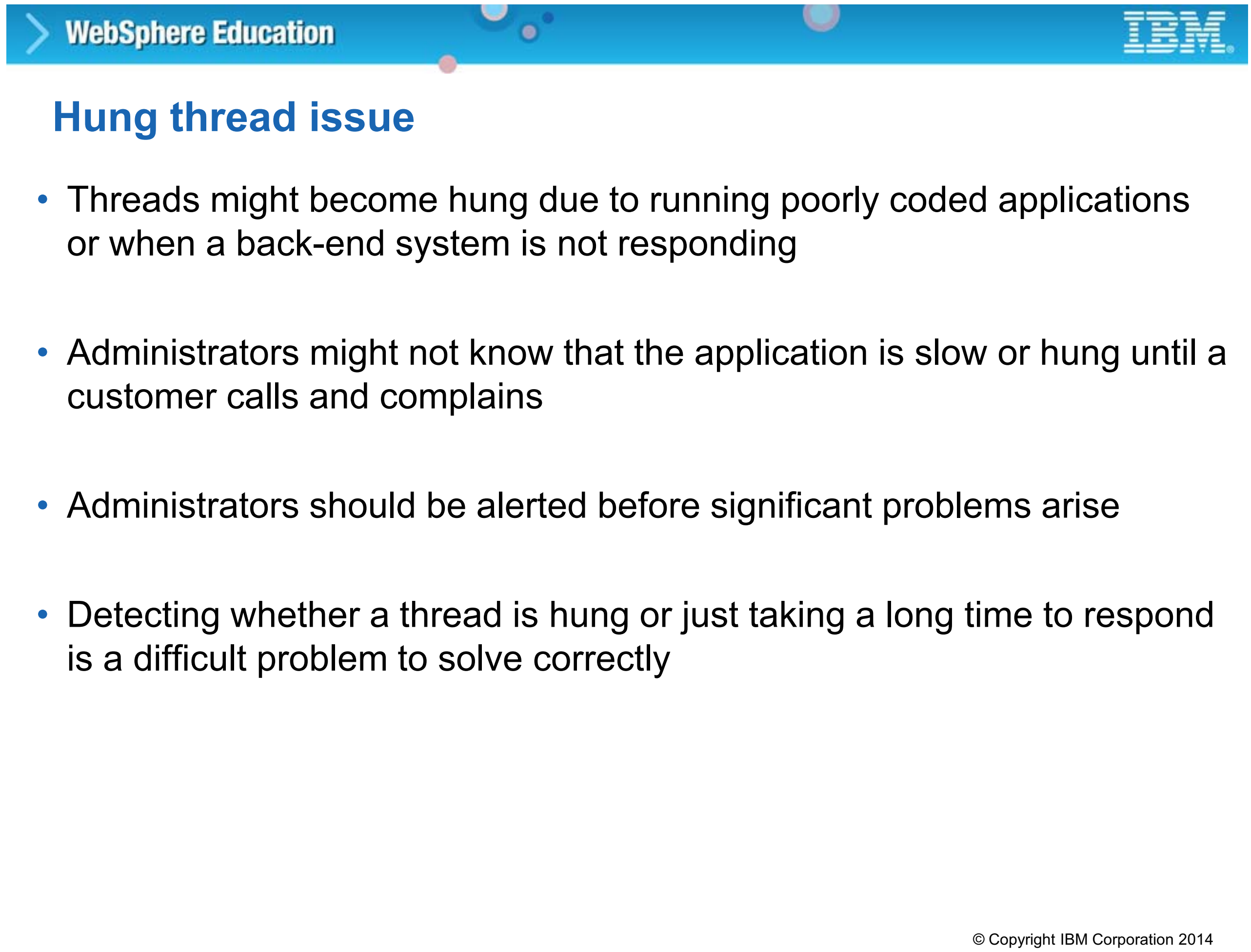


Figure 11-32. Hung thread issue WB868 / ZB8681.0

Notes:

There are tools available, and WebSphere has a built-in monitor to help identifying hung threads. These tools and the monitor are reviewed later in the unit. Using these facilities, it can be a simple process to identify when threads are hung.

Keep in mind that not all hangs are the result of application coding problems. They can be the result of problems in the JVM with WebSphere.

WebSphere Education IBM

Process hangs detection steps

1. Collect OS statistics, processor usage by other processes, and virtual memory paging
   – The JVM might be hung because OS resources are exhausted
2. When hung threads are suspected, manually trigger a thread dump
   – Use wsadmin or OS facilities; see next slide
   – Create a script or use script in IBM Support Assistant to collect MustGather data when the process that is suspected hangs
   – Distinguish the 100% CPU cases from idle CPU cases
3. For a typical hang, collect three javacore dumps a few minutes apart
   – To see whether anything is moving within the process (but slowly)
4. Examine the thread dumps manually or with tools
   – Look for deadlocks
   – Look for large number of threads that are blocked
   – Look for threads that are waiting after sending a request to some other process, now awaiting a response

Figure 11-33. Process hangs detection steps WB868 / ZB8681.0

Notes:

Java dumps are enabled by default. Java dump production can be turned off using `-Xdump:java:none`. Disabling a Java dump is a bad idea because Java dumps are essential diagnostics tools. Use the `-Xdump:java` option to give more fine-grained control over the production of Java dumps with IBM JDK on AIX, Linux, or Windows.

Note: The use of `–Xdump` is introduced in IBM JDK 5. For earlier versions of the IBM JDK the Java dump can be disabled by setting the `DISABLE_JAVADUMP` environment variable to `TRUE`.

By default, a Java dump is triggered when one of the following occurs:

- An unrecoverable native exception occurs in the JVM (not a Java Exception).
- The JVM has insufficient memory to continue operation; often caused by heap expansion and compaction.
- You send a signal to the JVM from the operating system.
- You use the `JavaDump()` method within Java code that is run.

The exact conditions in which you get a Java memory dump vary depending on whether the default memory dump agents are overridden. An unrecoverable exception is one that causes the JVM to fail. The JVM handles this situation by producing a javacore file and then stopping the process. In the user-controlled cases (the latter two), the JVM pauses, creates the memory dump, and then continues execution.

WebSphere Education

IBM

How to manually trigger a thread dump

- Use operating system facilities:
  - kill -3 <WAS PID> (UNIX or Linux)
- Use the Java core button in the administrative console:
  - Click **Troubleshooting > Java dumps and cores > *server_name* > Java core**
- Explicitly tell WebSphere to generate a thread dump
  - From a command line, start the wsadmin shell
  - Specify -lang jython on the command line
  - Run the following Jython commands:

```
jvm = AdminControl.completeObjectName(
              'type=JVM,process=<server_name>,*')
AdminControl.invoke(jvm, 'dumpThreads')
```


Figure 11-34. How to manually trigger a thread dump WB868 / ZB8681.0

Notes:

The default location for the placement of the javacore file is: <*BPM_install_root*>/profiles/<*profile*>. See the JVM introduction unit for complete details. For Solaris or HP-UX JDK, a thread dump is printed in native_stdout.log.

A javacore file can be generated on demand through the wsadmin command-line interface: From the command prompt, enter the command wsadmin.bat to get a wsadmin command prompt.

Note: If security is enabled or the default SOAP ports are changed, you must pass more parameters to the batch file to get a wsadmin prompt. For example:

```
wsadmin.bat [-host host_name] [-port port_number] [-lang jacl/jython] [-user
userid] [-password password]
```

Note: You can connect wsadmin to any of the server JVMs in the cell. After running the wsadmin command, it will display the server process to which it is attached. Depending on the process to which it is attached, you can get thread dumps for various JVMs. If wsadmin is connected to the deployment manager, then you can get thread dumps for any JVM in that cell. If it is attached to a node agent, then you can get thread dumps for any JVM in that node. If it is attached to a server, then you can get thread dumps only for the server to which is connected.

Get a handle to the problem application server. (Note: the contents in brackets "`[...]`", along with the brackets, are not optional. The text must be entered to set the JVM object. There is a space between `completeObjectName` and `type`.)

```
wsadmin> set jvm [$AdminControl completeObjectName type=JVM,process=server1,*]
```

Where `server1` is the name of the application server that does not respond (or is *hung*). If wsadmin is connected to a deployment manager and if the server names in the cell are not unique, then you can qualify the JVM with a node attribute in addition to the process.

Generate the thread dump:

```
wsadmin>$AdminControl invoke $jvm dumpThreads
```

You can keep the wsadmin session and use the `$AdminControl invoke $jvm dumpThreads` command multiple times to generate multiple thread dumps.

For Jython commands, replace steps 2 and 3 with the following commands:

```
jvm = AdminControl.completeObjectName('type=JVM, process=server1,*')
AdminControl.invoke(jvm, 'dumpThreads')
```

WebSphere Education

Thread dump analysis

What are the significant pictures in the thread dump?

- Many threads are waiting in the same method for some resource
  - Probably a synchronization issue
  - Might be an outage or slow response from remote server
- No activity
  - Server is not receiving traffic for some reason
  - Check front-end resources, networks, and test clients
  - Also, check timing of the thread dump
- Hundreds of threads
  - Shared resource not available
  - Customer must control web container threads better
  - Might need more capacity

Figure 11-35. Thread dump analysis WB868 / ZB8681.0

Notes:

Thread dumps are snapshot of the running JVM. They vary in format and detail among operating systems.

IBM JDK generates a separate javacore file. Solaris and HP-UX JDK print thread dumps in the `native_stdout.log` file.

WebSphere Education IBM

Javacore hang indicators

- Look for the string “Deadlock detected”
- JVM monitor information:
  - Shows synchronization locks
  - Indicates blocked threads
- Active threads
  - Look for `state:R`, which indicates runnable threads
  - This example shows that the thread is doing I/O
  - If this thread is doing the same operation across multiple javacore files, there might be a network interface issue

```
"Servlet.Engine.Transports:239" (TID:0x34B94018,sys_thread_t:0x7CD4E008,
state:R, native ID:0x10506) prio=5 at
java.net.SocketInputStream.socketRead(Native Method)
at java.net.SocketInputStream.read(SocketInputStream.java(Compiled Code))
at
com.ibm.ws.io.Stream.read(Stream.java(Compiled Code))
at
com.ibm.ws.io.ReadStream.readBuffer(ReadStream.java(Compiled Code))
```


Figure 11-36. Javacore hang indicators WB868 / ZB8681.0

Notes:

The monitor information shows what synchronization locks are held by which threads. It also shows which threads the monitors are blocking. This information is useful for determining the cause of a deadlocked or hung JVM. The monitor information is in a section entitled LOCKS subcomponent dump routine. It is before the thread dump of all the threads of the JVM.

Many blocked threads on a monitor do not mean that a deadlock occurred. It might mean that a monitor (synchronization lock) is causing a backlog of work to be completed.

The javacore processing dumps the current stack for every thread in the JVM. It shows the current state of the thread and produces a stack trace.

Thread states: The thread state indicates whether the thread is runnable or not. If the thread state is state:R, the thread is runnable. If the thread state is state:CW (conditioned wait), then the thread is in a wait state. If too many of the threads are in the CW state, you should focus on the monitor section to look for synchronized hangs.

The values of state can be:

- R – runnable: The thread is able to run when given the chance.
- CW – condition wait: The thread is waiting. For example, it might be because:

  - A `sleep()` call is made.
  - The thread is blocked for I/O.
  - A synchronized method of an object that is locked by another thread was called.
  - The thread is synchronizing with another thread with a `join()` call.
- S – suspended: The thread is suspended by another thread.
- Z – zombie: The thread is killed.
- P – parked: The new concurrency API (java.util.concurrent) parked the thread.
- B – blocked: The thread is waiting to obtain a lock that something else currently owns.

Java stack: The call stack under the thread header is the Java stack. The stack shows the Java calls that are made to get the thread to its current state. The first line in the Java stack is the last Java method call that was made. It was from that location that a call into a native method might be called. It is typically identified with the phrase “native method” showing the location in the Java program that was called.

Native stack: The native stack shows what native methods (procedures) were called after the thread entered the native code. The first line in the native stack shows what the thread was doing in native code when the javacore was taken.

WebSphere Education

Javacore hang symptoms (1 of 2)

- Check to see whether threads are blocked waiting on monitors
  - Might indicate bottleneck on unavailable resources or poor synchronization logic
  - Deadlocks within the process are noted in the javacore
- If threads are in a running state:
  - Check method across multiple javacore files
  - If individual threads in the same method, might indicate looping logic
- If threads are in wait states, might indicate that a resource (local or remote) is causing the hang
- Technical note:
  - Some threads that are Runnable are shown as in a Conditional Wait
  - Since IBM JVM version 5, when a javacore is taken, some threads in a Runnable (R) state enter Conditional Wait (CW) state during the javacore
  - This behavior is by design and it is meant to maximize internal consistency during the process of creating the javacore
  - This behavior increases the quality of the javacore and lessens the potential for a crash or data corruption

Figure 11-37. Javacore hang symptoms (1 of 2) WB868 / ZB8681.0

Notes:

Threads waiting on monitors are not usually deadlocks. Most of them are bottleneck or synchronization issues where the active thread is in some type of long timeout or resource shortage issue.

You must follow a chain of blocked threads to get to the root cause. For example, thread A might block multiple threads, while thread B blocks thread A, and thread B is waiting for a response from a remote server.

Java cores contain much information and cover dozens of threads. It is good to use tools to process the javacore, such as the Thread and Monitor Dump Analyzer. If such a tool is not available, important information can still be gained, though the process can be a bit tedious. Most tools will provide the capability to compare javacore files, making it easier to check the status of threads over time.

If a hung thread is suspected, also check for log messages from the hang detection facility included with WebSphere.

WebSphere Education

Javacore hang symptoms (2 of 2)

- The JVM might be hung because it is spending too much time in garbage collection
- An IBM javacore file can be loaded directly into the Garbage Collection and Memory Visualizer tool to review the garbage collection usage for the last few cycles before the javacore was triggered
  - Enable the Total Pause Time statistic in the Memory template

Total pause time

Mean	Minimum	Maximum	Total
time (ms)	time (ms)	time (ms)	time (ms)
12.8	4.88	32.5	372

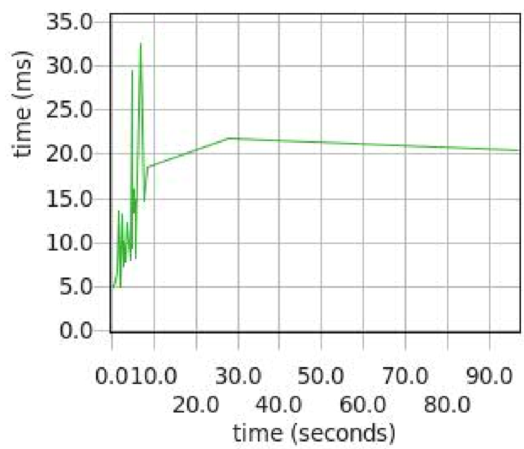


Figure 11-38. Javacore hang symptoms (2 of 2) WB868 / ZB8681.0

Notes:

In general, you would load verbose GC data into the Garbage Collection and Memory Visualizer tool and analyze it. Using the javacore file can help when you are troubleshooting hangs to determine whether excessive garbage collection, rather than hung threads, caused the hang.

WebSphere Education IBM

Hang detection tools

- **ThreadMonitor**: ThreadMonitor architecture was created to monitor thread pools within WebSphere
- Notification of potentially hung threads is logged
- Monitored pools include:
  - Web container thread pool
  - ORB thread pool
  - Others
- IBM Thread and Monitor Dump Analyzer
  - GUI-based tool
  - Gathers and analyzes thread dumps from a WebSphere Application Server
  - Provides tuning recommendations that are based on analysis
- IBM Monitoring and Diagnostic Tools for Java – Garbage Collection and Memory Visualizer
  - A verbose GC data visualizer
  - Parses and plots various log types that include verbose GC logs, `-Xtgc` output, native memory logs (output from `ps`, `svmon`, and `perfmon`)

Figure 11-39. Hang detection tools WB868 / ZB8681.0

Notes:

For the ThreadMonitor, the administrator can determine the threshold of how much time a thread can run before it is considered hung. By default, the monitoring is always on and has a threshold of 10 minutes and a check interval of 3 minutes. The performance degradation for this monitoring is less than 1%.

WebSphere Education IBM

Hung thread detection configuration

- Custom properties for hung thread detection configuration
  - Select **Servers > Application Servers > server_name**
  - Under Server Infrastructure, click **Administration > Custom Properties**
  - Add the following properties (or change if present):

Property	Units	Default	Description
`com.ibm.websphere. threadmonitor.interval`	seconds	180	The interval at which the thread pools are polled for hung threads
`com.ibm.websphere. threadmonitor.threshold`	seconds	600	The length of time that a thread can be active before being marked as "potentially hung"
`com.ibm.websphere. threadmonitor.false.alarm. threshold`	N/A	100	The number of false alarms that can occur before automatically increasing the threshold by 50%
`com.ibm.websphere. threadmonitor.dump.java`	N/A	False	Set to true to cause a javacore to be created when a hung thread is detected and a WSVR0605W message is printed

Figure 11-40. Hung thread detection configuration WB868 / ZB8681.0

Notes:

When the thread pool dispatches work to a thread, it sends a notification to the thread monitor, which notes the thread ID and the time in a list.

At user-configurable intervals, the thread monitor looks at the active threads, and compares them to the list, to determine how long each thread has been active. If a thread is active longer than the user-specified threshold, the thread is marked as "potentially hung", and the notifications are sent.

The performance impact of this monitoring is minimal, less than 1%.

These custom properties are not present unless added to the configuration. If not present, the default values take effect. To change the behavior, add these properties that provide values that you want.

The hang detection policy can be configured by creating custom properties for the application server.

com.ibm.threadmonitor.interval is the interval at which the thread pools are polled for hung threads (in seconds). It defaults to 180 seconds, which is 3 minutes.

com.ibm.websphere.threadmonitor.threshold is the length of time that a thread can be active before being marked as "potentially hung". The default value is 10 minutes.

com.ibm.websphere.threadmonitor.false.alarm.threshold is the number of false alarms that can occur before automatically increasing the threshold by 50%. The default value is 100. Automatic adjustment can be disabled altogether by setting this property to zero.

If com.ibm.websphere.threadmonitor.dump.java is set to true, javacore is taken when a hung thread is detected. This javacore can be helpful to discover a hung thread in an action automatically.

The application server must be restarted for these changes to take effect. To adjust the hang detection policy proactively, use wsadmin. Refer to the information center for instructions.

To disable the hang detection option, set the com.ibm.websphere.threadmonitor.interval property to less than or equal to zero.

11.6. IBM Thread and Monitor Dump Analyzer

IBM Thread and Monitor Dump Analyzer

Figure 11-41. IBM Thread and Monitor Dump Analyzer WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Thread and Monitor Dump Analyzer (TMDA)

- TMDA is a tool to analyze thread dumps
  - Available from IBM Support Assistant
- Used for:
  - Analyzing threads and monitors in javacores
  - Comparing multiple javacores from the same process
- Friendlier interface for novice thread dump readers
  - Provides graphical interface to view contents of the thread dump
- Used to analyze threads for the following situations:
  - Performance bottlenecks
  - Determining whether deadlocks exist
  - Determining whether threads are being blocked on monitors (might not be a deadlock)

Figure 11-42. Thread and Monitor Dump Analyzer (TMDA) WB868 / ZB8681.0

Notes:

The ThreadAnalyzer tool is still available through IBM Support Assistant. It is deprecated and not actively maintained.

WebSphere Education

TMDA: Open a thread dump

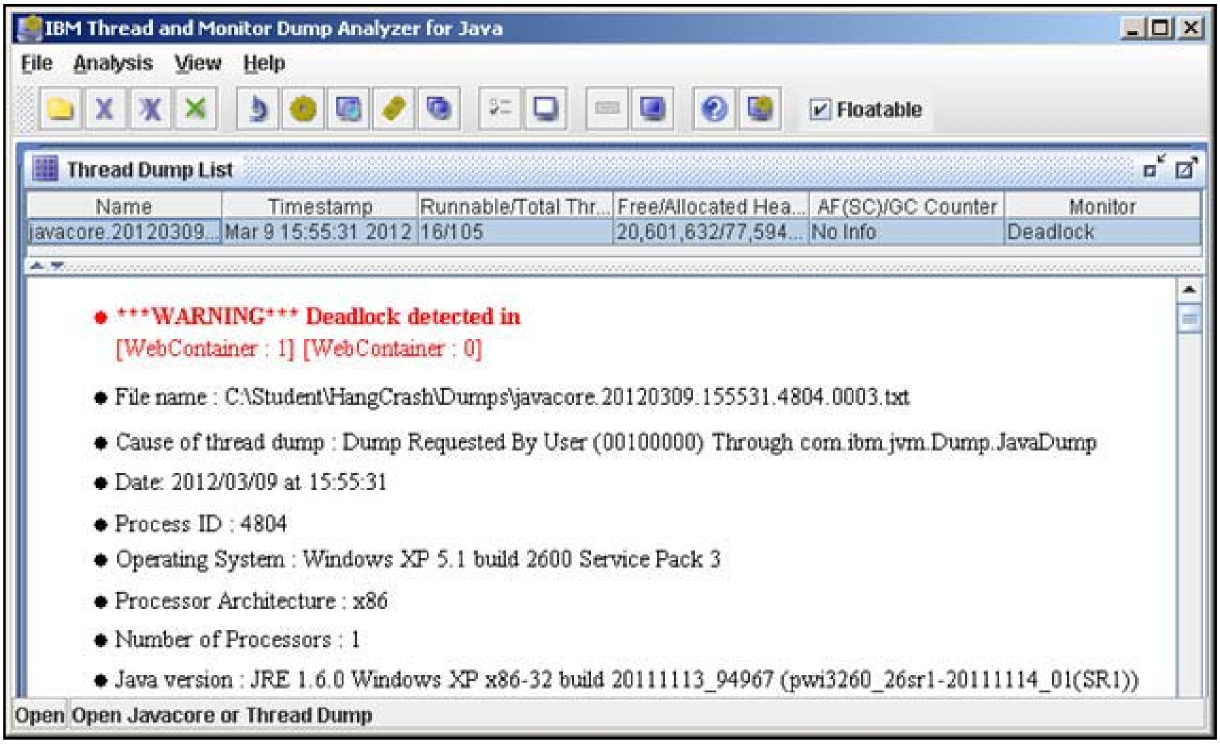


Figure 11-43. TMDA: Open a thread dump WB868 / ZB8681.0

Notes:

You can search the local file system for one or more javacore files. Each file is loaded into the tool and analyzed.

When a thread dump is loaded and selected, TMDA displays a summary information of the selected thread dump.

The following information is displayed for IBM javacore:

- Warning if any deadlocked threads are found
- File name
- Cause of thread dump
- Timestamp of the javacore
- Process ID
- Java version
- Java heap information:
  - Maximum Java heap size

  - Initial Java heap size
  - Garbage collector counter
  - Allocation failure counter
  - Free Java heap size
  - Allocated Java heap size
- Memory segment analysis (only for IBM JDK 5 and above)
- Current thread name
- Number of loaded classes in Java heap
- Suggested size of cluster (applicable only to IBM SDK 1.4.2 and 1.3.1 SR7 or later)
- Number of classloaders on the Java heap
- Java command line
- Thread status analysis – group threads by state, such as runnable, waiting
- Thread method analysis – group threads by the current method
- Thread aggregation analysis – group threads by thread function, such as ORB threads or web container threads

For Solaris and HP-UX javacore, only the file name, thread status analysis, thread method analysis, and thread aggregation analysis are provided.

WebSphere Education

Thread detail: Thread status analysis

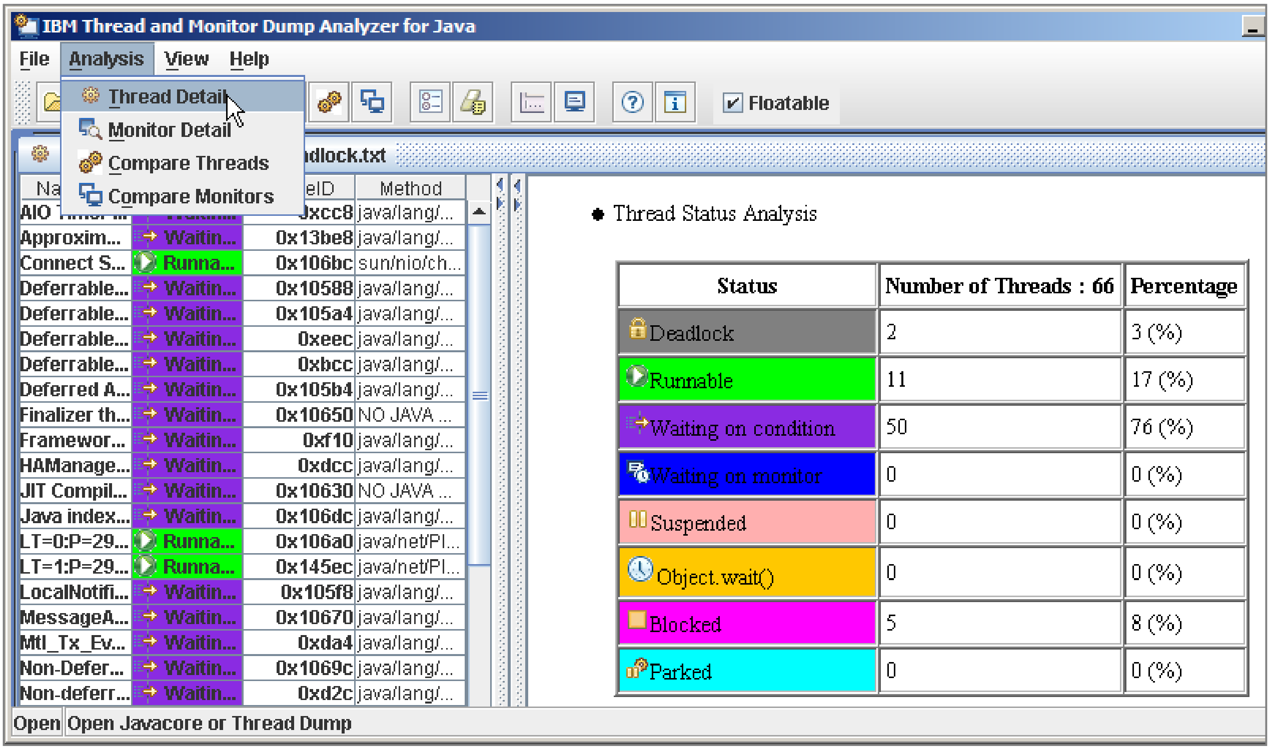


Figure 11-44. Thread detail: Thread status analysis WB868 / ZB8681.0

Notes:

The Analysis menu allows you to display thread and monitor details for a single javacore. If you open multiple javacores, you can display a comparative thread or monitor analysis.

The Thread Detail Analysis displays:

- Thread Status Analysis
- Thread Method Analysis
- Thread Aggregation Analysis

Thread Status Analysis shows the number of threads in each state: deadlocked, runnable, or blocked, for example.

The left pane lists all thread in the thread dump. Threads can be sorted by thread name, state, native ID, or current method.

When you select a thread from the left pane, the right pane shows Thread Detail View. Thread Detail View provides the following information:

- Thread name: The name of a thread

- Thread state: The state of a thread; for example, runnable, waiting, or suspended
- Method name: The most recent method that is run or predefined status or stack trace pattern; for example, IDLE, LISTEN, and KEEP-ALIVE
- Java Stack Trace: Java Stack Trace is shown when a thread is selected
- Native Stack Trace: Native Stack Trace is shown after the Java Stack Trace if it is available

The thread states can be:

- R – runnable: The thread is able to run when given the chance.
- CW – condition wait: The thread is waiting; for example, because:
  - A `sleep()` call is made.
  - The thread is blocked for I/O.
  - A synchronized method of an object that is locked by another thread was called.
  - The thread is synchronizing with another thread with a `join()` call.
- S – suspended: Another thread suspended the thread.
- Z – zombie: The thread is killed.
- P – parked: The new concurrency API (java.util.concurrent) parked the thread.
- B – blocked: The thread is waiting to obtain a lock that something else currently owns.

Note the following information:

- An IDLE thread is a thread that is ready to receive work but does not have a connection that is established.
- A KEEP-ALIVE thread is an idle thread that is ready to receive work and does have a connection that is established.
- A LISTEN thread listens on a port.

Thread detail: Thread method analysis

Method Name	Number of Threads : 66	Percentage
java/lang/Object.wait(Native Method)	42	64 (%)
java/lang/Thread.sleep(Native Method)	7	11 (%)
java/net/PlainSocketImpl.socketAccept(Native Method)	3	5 (%)
sun/nio/ch/WindowsSelectorImpl$SubSelector.poll0(Native Method)	2	3 (%)
com/ibm/issf/atjolin/badapp/BadAppServlet.sneezyMethod(BadAppServlet.java:332)	2	3 (%)
com/ibm/io/async/AsyncLibrary.aio_getioev2(Native Method)	2	3 (%)
NO JAVA STACK	2	3 (%)
com/ibm/jvm/Dump.JavaDump(Native Method)	1	2 (%)
com/ibm/issf/atjolin/badapp/BadAppServlet.sneezyMethod(BadAppServlet.java:337)	1	2 (%)
com/ibm/issf/atjolin/badapp/BadAppServlet.dopeyMethod(BadAppServlet.java:320)	1	2 (%)
java/net/PlainDatagramSocketImpl.receive0(Native Method)	1	2 (%)
java/net/SocketInputStream.socketRead0(Native Method)	1	2 (%)
com/ibm/misc/SignalDispatcher.waitForSignal(Native Method)	1	2 (%)

Figure 11-45. Thread detail: Thread method analysis WB868 / ZB8681.0

Notes:

This table groups threads by the current threads. It is a convenient view to find out whether many threads are blocked in the same method.

WebSphere Education

Thread detail: Memory segment analysis

Memory Type	# of Segments	Used Memory(bytes)	Used Memory(%)	Free Memory(bytes)	Free Memory(%)	Total Memory(bytes)
Internal	102	6,567,172	98.24	117,500	1.76	6,684,672
Object	1	65,131,520	100	0	0	65,131,520
Class	1,090	77,451,880	95.1	3,988,936	4.9	81,440,816
JIT Code Cache	7	0	0	3,670,016	100	3,670,016
JIT Data Cache	5	2,214,476	84.48	406,964	15.52	2,621,440
Overall	1,205	151,365,048	94.87	8,183,416	5.13	159,548,464

Figure 11-46. Thread detail: Memory segment analysis WB868 / ZB8681.0

Notes:

The memory segment analysis is available only for IBM JDK 5 and above. It provides information of memory usage by JVM process. It can be useful in debugging native memory problems.

WebSphere Education

Multiple dump comparative analysis

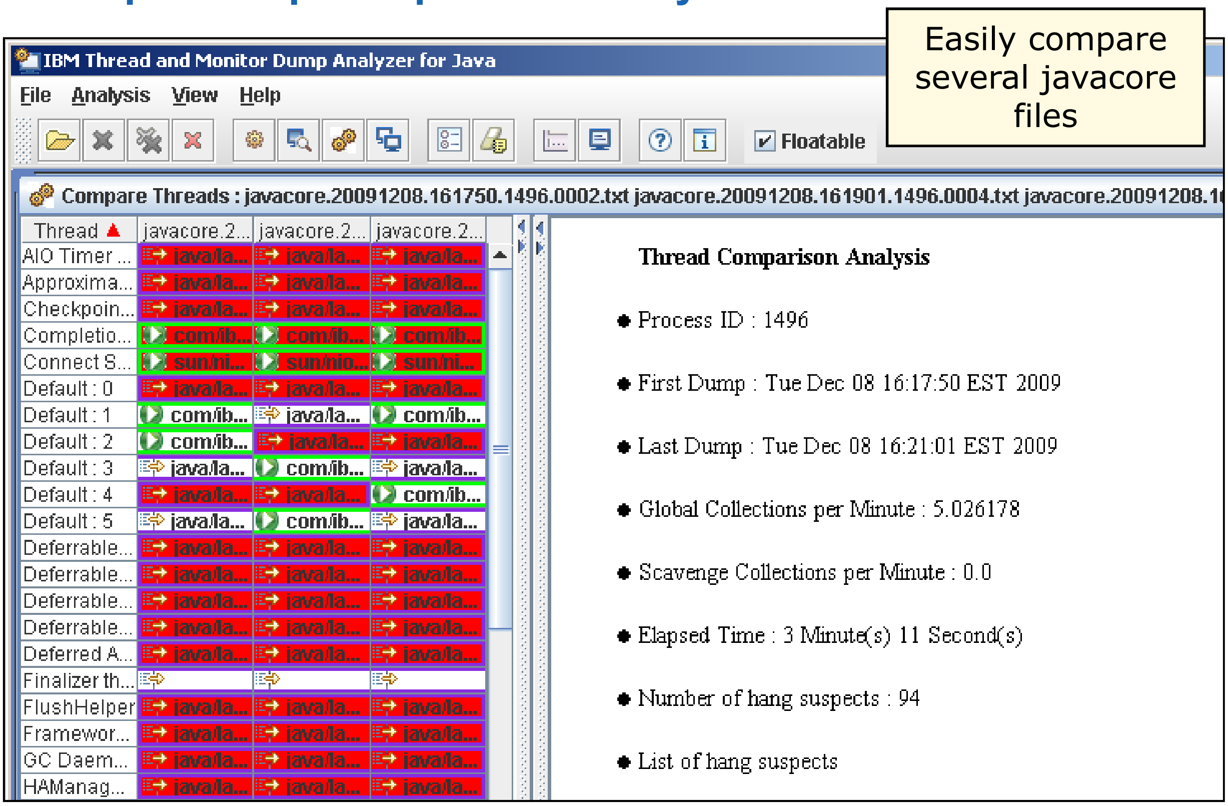


Figure 11-47. Multiple dump comparative analysis WB868 / ZB8681.0

Notes:

The Compare Threads View for two or more javacores consists of the following information:

- Summary is displayed in the right pane. Thread name and method name are displayed from each javacore on the table.
- Process ID.
- First Dump: Timestamp of the first javacore.
- Last Dump: Timestamp of the last javacore.
- Garbage Collections per Minute: Number of garbage collections per minute.
- Allocation Failures per Minute: Number of allocation failures per minute.
- Elapsed Time: Time between the first javacore and the last javacore.
- Number of hang suspects.
- List of hang suspects. Many threads are in idle state, and they are hang suspects.

WebSphere Education

Thread analysis: Deadlocked thread details

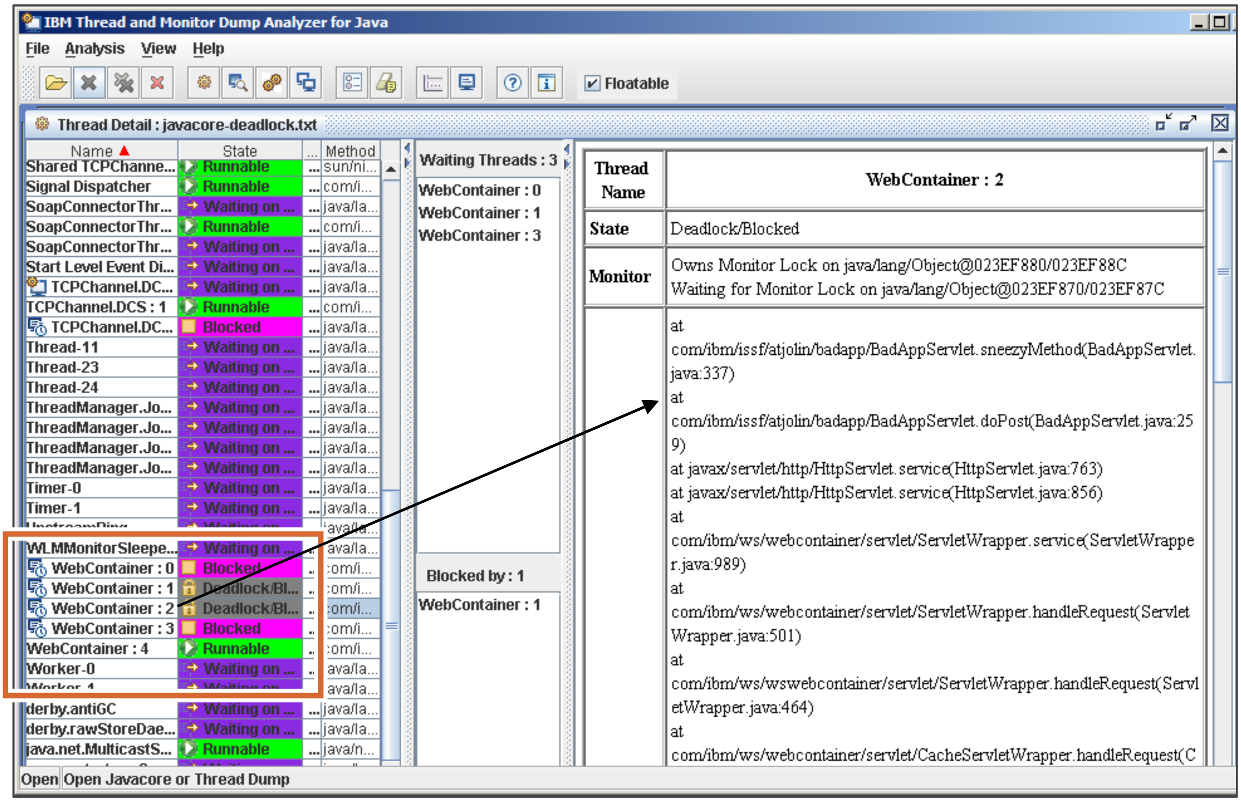


Figure 11-48. Thread analysis: Deadlocked thread details WB868 / ZB8681.0

Notes:

In the left pane, each thread name can be selected, and the details of the thread are displayed in the right pane.

Notice that in the web container, two threads are selected in the screen capture. The middle pane shows threads that are waiting and the threads that web container 2 is blocking.

Thread analysis: Monitor details

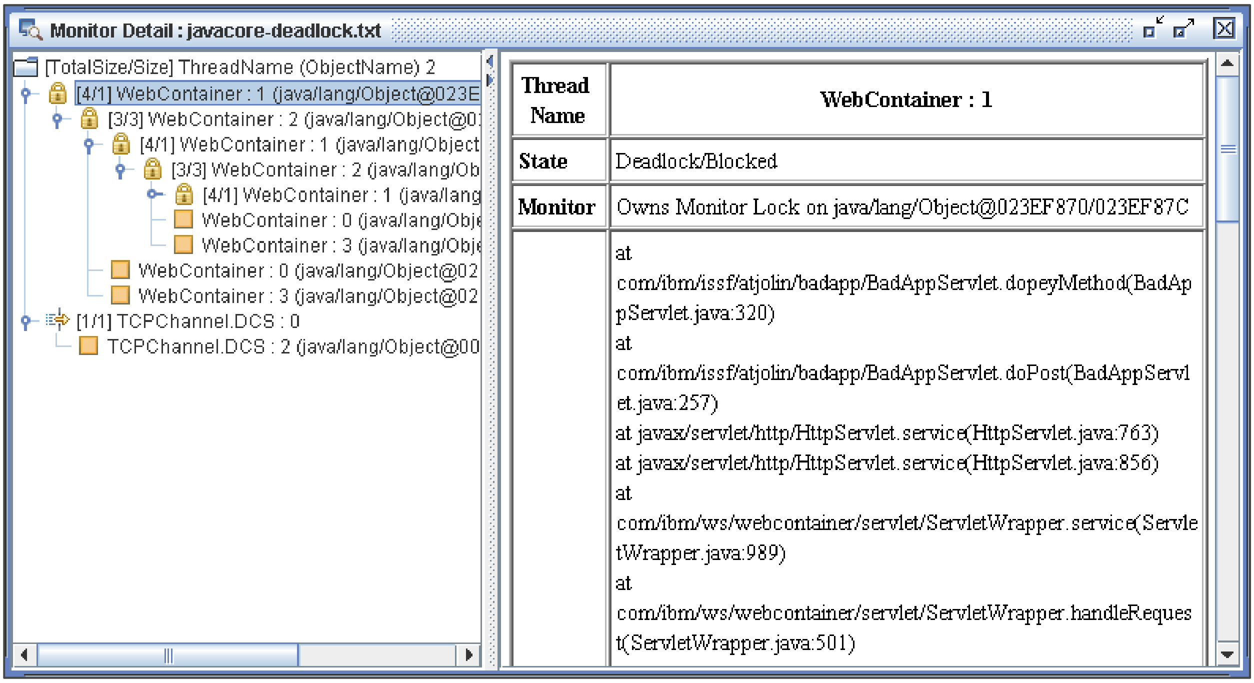


Figure 11-49. Thread analysis: Monitor details WB868 / ZB8681.0

Notes:

The Monitor Detail view provides a hierarchical tree of the threads.

Each top-level thread holds a lock. By clicking each thread to expand the tree, in the hierarchy you can see information about the threads that this thread is blocking.

11.7. Preventive measures

Preventive measures

Figure 11-50. Preventive measures WB868 / ZB8681.0

Notes:

WebSphere Education

Daily basis tasks

- Verify that all error queues are empty
  - For SCA-based applications, check the Failed Event Manager for any failed events
  - If you defined any exception queue for the connections component that is used on SCA import and export, check those queues also
  - For BPMN-based applications, check the Process Inspector to see whether there are any failed instances
- Check the server log files for errors and exceptions
  - Account for the error if it cannot be eliminated
- Monitor the memory and CPU spike
- Look for failed instances and identify the root cause of the failure
- Check the FFDC log files
- Monitor the average growth in instances and task count
  - For BPMN processes, compare the changes in values in the LSW_PRI_KEY table

Figure 11-51. Daily basis tasks WB868 / ZB8681.0

Notes:

Preventing issues is always better than fixing them. Regular maintenance helps you better separate the normal behavior from the abnormal behavior. That maintenance can help you find the system error quickly and help you determine the exception area or possible cause more quickly.

WebSphere Education

Weekly basis tasks

- Check for product maintenance fixes
- Check the product technotes for any issues or exceptions that are found in the BPM log files
- Check for database log files and follow the recommendations for your database product to maintain a healthy database
- Review the database performance report (take a 4-hour sample)
- Remove unnamed snapshots and AIS business-level applications that are no longer needed

Figure 11-52. Weekly basis tasks WB868 / ZB8681.0

Notes:

WebSphere Education

Monthly basis tasks

- Run regular process data cleanup to remove old or completed information
  - Over time it can lead to disk space issues and performance issues as database queries take longer and longer to process
  - It is important to have a policy of continuously removing older data
- Remove artifacts that are no longer needed
  - Unnamed snapshots
  - AIS business-level applications
  - Named snapshots
  - Old or completed process information
  - Tracking data

Figure 11-53. Monthly basis tasks WB868 / ZB8681.0

Notes:

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Describe the nature of a performance problem
- Describe the types of data that is needed to troubleshoot performance problems
- Identify integration performance concerns
- Apply high-level problem determination methods to a performance problem
- Detect a hang condition
- Trigger and analyze javacore files for hangs
- Use the WebSphere Application Server hang detection facility
- Use the IBM Thread and Monitor Dump Analyzer for Java
- Identify preventive measures for the Business Process Manager environment

Figure 11-54. Unit summary WB868 / ZB8681.0

Notes:

WebSphere Education

Checkpoint questions

1. True or False. Heap dumps are needed only in case there is an out-of-memory problem.

2. True or False. GC logs give you the ability to graph memory utilization over time

3. True or false: The Thread Monitor facility is turned on by default for every application server.

4. True or false: The Thread and Monitor Dump Analyzer is part of the administrative console.

Figure 11-55. Checkpoint questions WB868 / ZB8681.0

Notes:

Write your answers here:

1.

2.

3.

4.

WebSphere Education IBM

Checkpoint answers

1. True

2. True

3. True

4. False: The Thread and Monitor Dump Analyzer runs in the IBM Support Assistant.

Figure 11-56. Checkpoint answers WB868 / ZB8681.0

Notes:

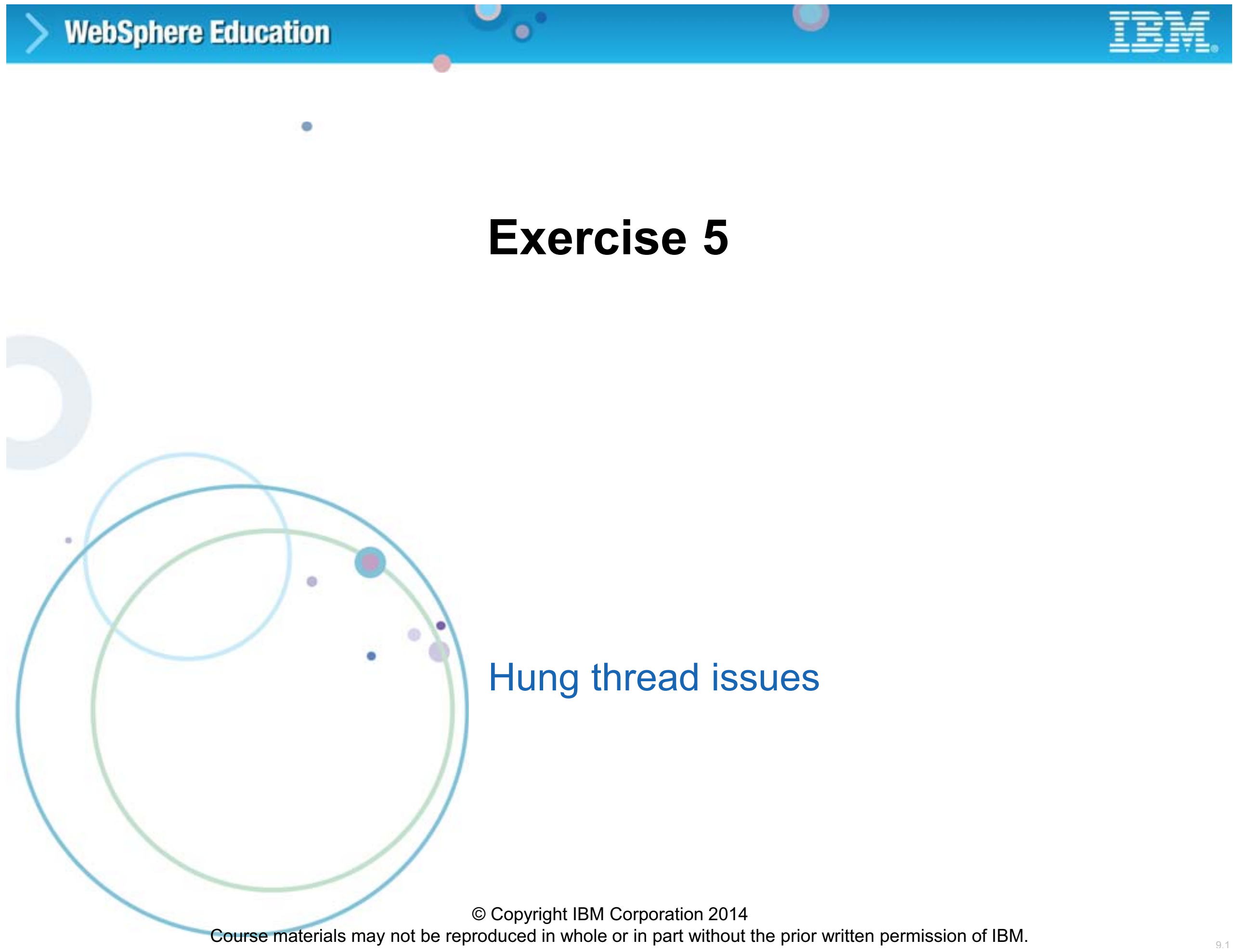


Figure 11-57. Exercise 5 WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Exercise objectives

After completing this exercise, you should be able to:

- Use IBM Integration Designer to explore SCA modules and components
- Use wsadmin to trigger a javacore file
- Analyze a thread dump by using the IBM Support Assistant Thread and Monitor Dump Analyzer for Java tool

Figure 11-58. Exercise objectives WB868 / ZB8681.0

Notes:

Unit 12. Database tuning

What this unit is about

This unit introduces database tuning considerations for the Business Process Manager environment. You examine common tuning tasks for DB2 and Oracle databases.

What you should be able to do

After completing this unit, you should be able to:

- Explain general database tuning considerations
- Identify tuning practices that are specific to DB2
- Identify tuning practices that are specific to Oracle
- Describe best practices for BPMN workloads
- Describe best practices for BPEL workloads

How you will check your progress

- Checkpoint

References

IBM Business Process Manager V8.5 IBM Knowledge Center

`http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/ditamaps/ic-homepage-bpm.html`

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Explain general database tuning considerations
- Identify tuning practices that are specific to DB2
- Identify tuning practices that are specific to Oracle
- Describe best practices for BPMN workloads
- Describe best practices for BPEL workloads

Figure 12-1. Unit objectives WB868 / ZB8681.0

Notes:

WebSphere Education

Topics

- General database tuning considerations
- DB2 specific database tuning
- Oracle specific database tuning
- Tuning for BPMN and BPEL workloads

Figure 12-2. Topics

WB868 / ZB8681.0

Notes:

12.1.General database tuning considerations

WebSphere Education IBM

General database tuning considerations

9.1

Figure 12-3. General database tuning considerations WB868 / ZB8681.0

Notes:

WebSphere Education

The importance of the database for BPM

- Nearly all customer data and internal processing is stored in the database
  – Used for persisting the state of business processes and tasks
  – Leverages the multi-user concurrency and transactional features of the database
  – BPM performance depends on database performance

- Common database tasks and configuration parameters
  – Table spaces
  – Transaction logs
  – Memory
  – Statistics
  – Tuning queries
  – Purge data
  – Data source configuration

Figure 12-4. The importance of the database for BPM WB868 / ZB8681.0

Notes:

WebSphere Education

Initial database tuning and parameter settings (1 of 2)

- Place log files and table space containers on separate physical disks
  - Place log files on a fast disk subsystem
- Increase the log file size for the Process Server database
- Set the buffer pool or cache size to a minimum of 2 GB
  - Larger is better, assuming sufficient physical memory to hold the buffer pool
- Enable file system caching for the Process Server database
- Ensure that the database statistics are up-to-date
- Provide sufficient physical memory

Figure 12-5. Initial database tuning and parameter settings (1 of 2) WB868 / ZB8681.0

Notes:

Accessing data in memory is much faster than reading it from disk. Because 64-bit hardware is readily available and memory prices continue to fall, the sensible approach is to provision enough memory to avoid most disk reads in steady state for many performance-critical workloads.

Take great care to avoid virtual memory paging on the database computer. The database manages its memory with the assumption that it is never paged, and does not cooperate well if the operating system swaps some of its pages to disk.

WebSphere Education

Initial database tuning and parameter settings (2 of 2)

- Monitor top SQL statements
  - Use database vendor tools to discover expensive SQL statements
- Add indexes as required
- Archive or delete completed process instances
- Increase the **Maximum Connections** property of data sources
  - Make the value large enough to allow concurrent access to the database from all threads
  - Verify that the database can handle the number of incoming connections
- Increase the prepared statement cache size
- Exclude SIBOWNER from the automatic statistics collection

Figure 12-6. Initial database tuning and parameter settings (2 of 2) WB868 / ZB8681.0

Notes:

Business Process Manager products typically provide a reasonable set of indexes for the database tables used. In general, creating indexes involves a tradeoff between the cost of queries and the cost of statements that insert, update, or delete data. For query-intensive workloads, providing a rich variety of indexes as required to allow rapid access to data makes sense. For update-intensive workloads, a helpful approach is to minimize the number of indexes that are defined because each row modification might require changes to multiple indexes. Indexes are kept current even when they are infrequently used.

WebSphere Education IBM

Place log files and table space containers on separate physical disks

- When changes are made to table data, the changes might not be written immediately to disk but written to the database log
  - Database log files might be heavily used
- Updates are populated from the log file to disk when you reach any of the following conditions
  - The log buffer fills
  - The transaction-commit time
  - After a maximum interval of time
- The log-write holds pending commit operations, and the application is synchronously waiting for the write to complete
  - The performance of write access to the database log files is critical to overall system performance
- To achieve best I/O performance, separate the database log, database control file, table space container, and index container
  - Also improves recoverability when log archival is used

Figure 12-7. Place log files and table space containers on separate physical disks WB868 / ZB8681.0

Notes:

A basic strategy for all database storage configurations is to place the database logs on separate physical disk devices from the table space containers.

This strategy reduces disk access contention between input and output to the table space data sets, and input and output to the database logs, and preserves the mostly sequential access pattern for the log stream.

A basic strategy for all database storage configurations is to place the database logs on dedicated physical disks, ideally on a dedicated disk adapter. This placement reduces disk access contention between I/O to the table space containers and I/O to the database logs and preserves the mostly sequential access pattern of the log stream. Such separation also improves recoverability when log archival is employed.

WebSphere Education

Ensure that the database statistics are up to date

- Database statistics, which describe the shape of the data, can be used to guide the selection of a low-cost data access strategy
  - Statistics are maintained on tables and indexes
  - Examples of statistics include the number of rows in a table and the number of distinct values in a certain column
- Updating statistics ensures that database operation compile with up-to-date statistics and optimize the execution plan
- On DB2, you can use `runstats` to update the table statistics, which ensures that the correct statistics are collected and maintained
- On Oracle, you can gather and use optimizer statistics with the `DBMS_STATS` package

Figure 12-8. Ensure that the database statistics are up to date WB868 / ZB8681.0

Notes:

For information on managing optimizer statistics on Oracle, see:
`http://docs.oracle.com/cd/E11882_01/server.112/e16638/stats.htm#PFGRF003`

WebSphere Education

Add indexes as required

- BPM creates a set of database indexes that are appropriate for many installations, but more indexes are required in some circumstances
  - If database processor or disk usage is high, or there are concerns with database response times, it might be helpful to consider changes to indexes
- Creating indexes involves a tradeoff between the cost of queries and the cost of statements that insert, update, or delete data
- A database environment that requires more indexes often displays performance degradation over time
  - Environments that need more indexes often exhibit heavy read I/O on devices that hold the table space containers
- To help determine which extra indexes could improve performance, use your specific database vendor tools

Figure 12-9. Add indexes as required WB868 / ZB8681.0

Notes:

In this area, DB2 databases assist by analyzing indexes in the context of a workload. Recommendations are given to add, modify, or remove indexes. One caveat is that if the workload does not capture all relevant database activity, a necessary index might appear unused, leading to a recommendation that it be dropped. If the index is not present, future database activity might suffer as a result.

WebSphere Education IBM

Monitoring and troubleshoot database performance issues

- The database information is always there and you can gain insights, which might be not achievable with other means
  - When troubleshooting, sometimes a trace can be more useful, depending on the problem scenario
  - If you want to troubleshoot a problem, which rarely occurs or happens on a high load production system, you might not be in a position to gain information with a trace
- Obtain MustGather information for troubleshooting
  - Database server CPU, disk subsystem, and network utilization
  - DB2: Statement event monitor or db2support tool
  - Oracle: Automatic workload repository (AWR) report

Figure 12-10. Monitoring and troubleshoot database performance issues WB868 / ZB8681.0

Notes:

Database monitoring is a vital activity for the maintenance of the performance and health of your database management system.

Monitoring table functions and snapshot routines return the values of monitor elements at the specific point in time that the routine is run. This information is useful when you want to check the current state of your system. However, you might not always want to monitor points in time. There are many times when you need to capture information about the state of your system at exactly the time that a specific event occurs. Event monitors serve this purpose.

12.2.DB2 database specific tuning

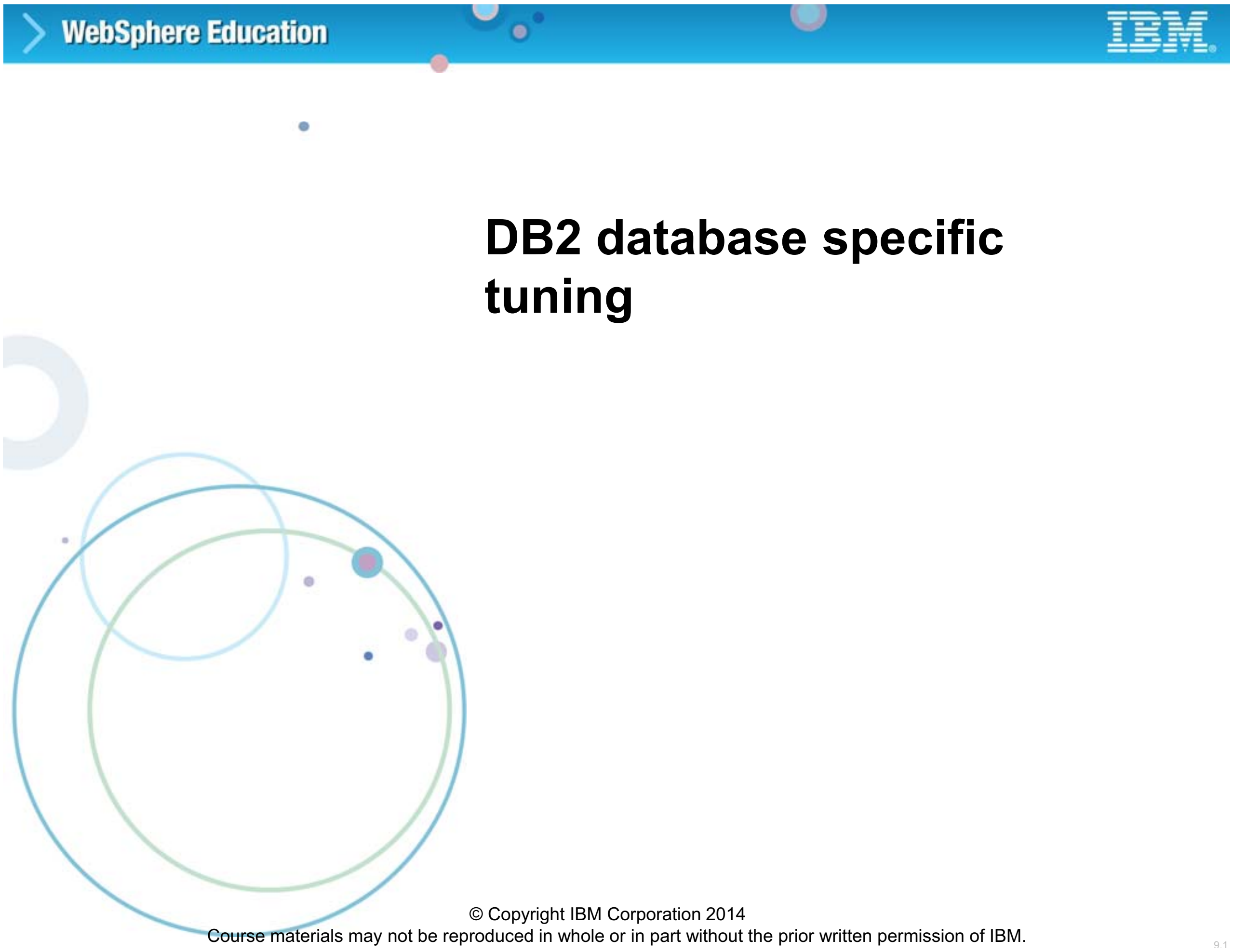


Figure 12-11. DB2 database specific tuning WB868 / ZB8681.0

Notes:

WebSphere Education

DB2 tuning information

- DB2 provides a number of automatic, or self-tuning, features to help managing and tuning a database system
  - Best practice to use these features
  - Examples include database memory, maximum storage for lock lists, package cache size, and other memory tuning features

- You can enable more self-tuning features
  - For individual buffer pools, use **`SIZE AUTOMATIC`** with the **`CREATE`** or **`ALTER`** commands

- Complete DB2 documentation is available at
  - For V9.7: `http://www.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/`
  - For V10.1: `http://www.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/`

Figure 12-12. DB2 tuning information WB868 / ZB8681.0

Notes:

Self-tuning memory simplifies the task of memory configuration by automatically setting values for memory configuration parameters and sizing buffer pools. When enabled, the memory tuner dynamically distributes available memory resources among the following memory consumers:

- Buffer pools
- Locking memory
- Package cache
- Sort memory

Self-tuning memory is enabled through the `self_tuning_mem` database configuration parameter.

WebSphere Education IBM.

Best practices for DB2 (1 of 2)

- Manage large objects separate from other data to achieve optimal performance
  - Use the **NO FILE SYSTEM CACHING** setting on the table spaces that do no have large objects
  - Move large objects to table spaces with the **FILE SYSTEM CACHING** setting
- The **SET INLINE LENGTH** value on large object columns provides significant performance improvements
- Set the appropriate buffer pool size
  - Set it manually or enable self-tuning
  - Monitor the database container I/O activity to determine the settings
  - One of the most important tuning parameters
- Use the self-tuning memory features to automatically set values for memory parameters and sizing buffer pools

Figure 12-13. Best practices for DB2 (1 of 2) WB868 / ZB8681.0

Notes:

A buffer pool provides working memory and cache for database pages.

Buffer pools improve database system performance by allowing data to be accessed from memory instead of from disk. Because most page data manipulation takes place in buffer pools, configuring buffer pools is the single most important tuning area.

WebSphere Education IBM

Best practices for DB2 (2 of 2)

- Verify that sufficient locking resources are available
  - Problems can lead to database deadlocks
  - Obtain values for `locklist` and `maxlocks` and determine whether these parameters need to change
  - The `locklist` parameter controls locks that are allocated from a common pool
  - The `maxlocks` parameter bounds the percentage of the lock pool that is held by a single application
- Maintain correct table indexing
  - Use the DB2 Design Advisor for suggestions on schema changes, including changes to indexes
- Exclude the table SIBOWNER from the DB2 Health monitor
  - You might encounter transaction log full errors when you enable the automatic `runstats` execution

Figure 12-14. Best practices for DB2 (2 of 2) WB868 / ZB8681.0

Notes:

Use the `maxlocks` parameter to define a percentage of the lock list that is held by an application that must be filled before the database manager performs lock escalation.

The use of indexes can significantly improve the performance of your queries.

WebSphere Education IBM

Troubleshooting

- Minimize the impact of troubleshooting
  - Collect diagnostic data only where the problem is occurring
  - Collect only the diagnostic data that you need
- Use the DB2 Design Advisor to identify all of the objects that are needed to improve the performance of your workload
- Use the **db2support** tool to collect needed information
  - The tool collects configuration information, database log files, and system diagnostic information
- The **db2exfmt** command collects access plan information and statistics for the query in question
  - The command is useful for a performance problem of a specific query

Figure 12-15. Troubleshooting WB868 / ZB8681.0

Notes:

When it comes to collecting information for a DB2 problem, the most important DB2 utility you need to run is **db2support**. The db2support command automatically collects all DB2 and system diagnostic information available. It also has an optional interactive “Question and Answer” session, which poses questions about the circumstances of your problem.

Using the db2support utility avoids possible user errors, as you do not need to manually type commands such as `GET DATABASE CONFIGURATION FOR` *`database-name`* or `LIST TABLESPACES SHOW DETAIL`. Also, you do not require instructions on which commands to run or files to collect; therefore it takes less time to collect the data.

12.3. Oracle database specific tuning

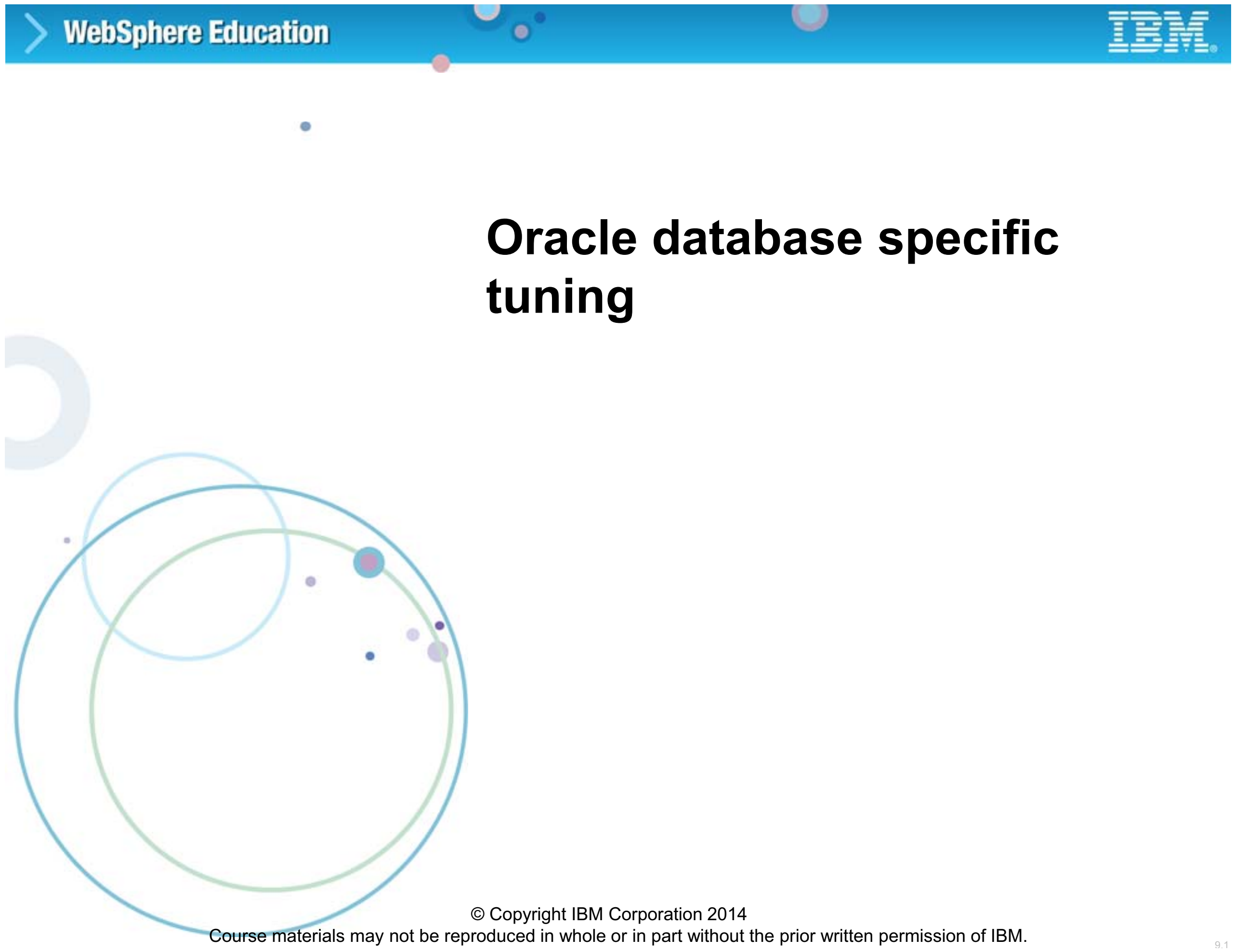


Figure 12-16. Oracle database specific tuning WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Oracle tuning information

- Oracle provides a number of guides to help you improve the performance of your database system
  - Examine the Performance Tuning Guide
- Specific Oracle tasks to examine include
  - Proactively monitor the condition of the database and take preventive or corrective actions, as required
  - Monitoring and tuning for database performance
- Complete Oracle documentation is available at
  - For 11g Release 2: `http://docs.oracle.com/cd/E11882_01/index.htm`
- Oracle Architecture and Tuning on AIX reference white paper:
  - `http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100883`

Figure 12-17. Oracle tuning information WB868 / ZB8681.0

Notes:

The Oracle performance method can be applied until performance goals are met or deemed impractical. Because this process is iterative, some investigations might have little impact on system performance. It takes time and experience to accurately pinpoint critical bottlenecks quickly.

WebSphere Education IBM

Best practices for Oracle (1 of 3)

- Modify the number of table spaces
  – Consider minimizing the value when creating table spaces by setting initial and autoextended sizes
  – This change can help produce fewer spikes in database utilization under peak load
  – Manually extend the value during periods of low activity
- Manage large objects separate from other data to achieve optimal performance
  – Disable concurrent I/O at the database level
  – Mount the file system that contains data files that do not have large objects with options to enable the use of concurrent I/O
  – Move large objects to table spaces that reside on file systems that are mounted with options to use file system cache
  – Partition large objects to reduce file locking contention

Figure 12-18. Best practices for Oracle (1 of 3) WB868 / ZB8681.0

Notes:

Large objects are a set of data types that are designed to hold large amounts of data. A large object can hold up to a maximum size that ranges 8 - 128 terabytes depending on how your database is configured. Storing data in large objects enables you to access and manipulate the data efficiently in your application.

WebSphere Education IBM

Best practices for Oracle (2 of 3)

- Size log files appropriately
  - Make sure log files are large enough to avoid frequent checkpoint operations
- Configure automatic memory management for buffer caches
- Ensure that the database statistics are up-to-date
- Identify and improve the performance of long-running SQL statements
  - Identify long running SQL statements by using an automatic workload repository (AWR) report or the Oracle Enterprise Manager
  - Access the statement
  - Run the SQL Tuning Advisor against the long-running SQL statements
  - Evaluate the recommendations from the SQL Tuning Advisor

Figure 12-19. Best practices for Oracle (2 of 3) WB868 / ZB8681.0

Notes:

The transaction log of an Oracle instance is stored in several files that are used in a round-robin fashion. The active log files are switched when one becomes full, allowing the last active log to be archived. Because switching log files is an expensive operation, size the log files so that these switches occur infrequently; 750 MB is a good starting value. Then, monitor the transaction rate and the average log size, and adjust this value as needed.

WebSphere Education

Best practices for Oracle (3 of 3)

- Maintain correct table indexing
  - Use the SQL Access Advisor for recommendations
- Examine and tune specific database settings for BPM
  - memory_mx_target: 25G
  - memory_target: 25G
  - processes: 500
  - open_cursors: 1000
  - undo_retention: 200
  - _undo_autotune: FALSE
- Modify the max-idle parameter in the `98Database.xml` file
  - Indicate to keep more threads active in the connection pool to better manage peak throughput

Figure 12-20. Best practices for Oracle (3 of 3) WB868 / ZB8681.0

Notes:

OPEN_CURSORS specifies the maximum number of open cursors (handles to private SQL areas) that a session can have at one time. You can use this parameter to prevent a session from opening an excessive number of cursors.

UNDO_RETENTION specifies (in seconds) the low threshold value of undo retention. For AUTOEXTEND undo table spaces, the system retains undo for at least the time that is specified in this parameter, and automatically tunes the undo retention period to satisfy the undo requirements of the queries. For fixed-size undo table spaces, the system automatically tunes for the maximum undo retention period, which is based on undo table space size and usage history, and ignores UNDO_RETENTION unless retention guarantee is enabled.

WebSphere Education IBM

Troubleshooting

- Tools for any Oracle database-related problems are AWR and ADDM reports and `alert.logs`
- Every hour, AWR collects a snapshot of the system workload by default:
  – Corresponding AWR reports can be created afterward by default for seven days
  – You define which snapshots should be included in the report
  – Manual snapshots can be taken and included in the report
  – The data includes slow performing queries as well as hints where the most time is spent
- ADDM (Automatic Database Diagnostic Monitor)
  – Based on the same snapshots that are taken as part of the AWR recommendations, which are provided to improve the general system performance and behavior

Figure 12-21. Troubleshooting WB868 / ZB8681.0

Notes:

Alert logs contain important information about error messages and exceptions that occur during database operations. Each Oracle database for Windows instance has one alert log; information is appended to the file each time you start the instance. All threads can write to the alert log.

For example, when automatic archiving of redo logs is halted because no disk space is available, a message is placed in the alert log. The alert log is the first place to check when something goes wrong with the database and the cause is not immediately obvious.

12.4.Tuning for BPMN and BPEL workloads

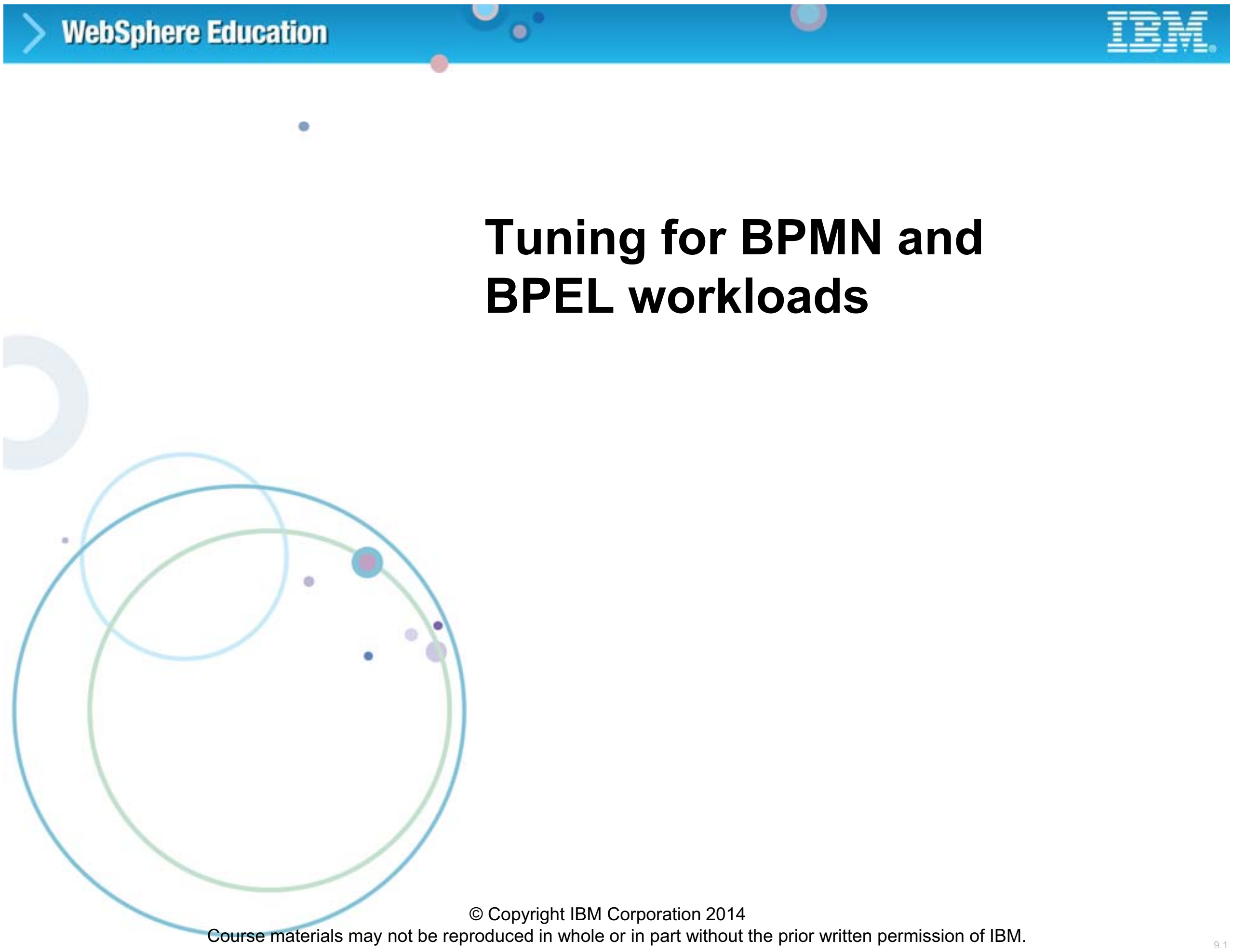


Figure 12-22. Tuning for BPMN and BPEL workloads WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Database tables

- A few of the essential database tables for BPD instances
  - LSW_BPD_INSTANCE for process instances
  - LSW_TASK for tasks instances
  - LSW_SNAPSHOT for snapshot information
  - LSW_USR_GRP_XREF for user information

- A few of the essential database tables for BPEL instances
  - PROCESS_INSTANCE_B_T for process instances
  - ACTIVITY_INSTANCE_B_T for activity instances
  - TASK_INSTANCE_T for task instances
  - WORK_ITEM_T for work items
  - SCOPE_INSTANCE_B_T for more scope information

Figure 12-23. Database tables WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Best practices for BPMN workloads

- Verify that optimizer statistics are current for hot spot tables
  - Verify that access paths are optimized for SQL queries that are reading from the hot spot tables
- Tune parameters to optimize process searches (task list queries)
  - Verify that no more than 10 business data variables are defined
  - Configure search optimizations through the Saved Search Accelerator Tools for the Process Server
  - Tools include the `SchemaGenerator` and `DataLoad` to optimize a process search
- Update the indexes for the Performance Data Warehouse
- Oracle considerations for processes and task list queries
  - Examine more indexes as a starting point for defining indexes in the Process Server database

Figure 12-24. Best practices for BPMN workloads WB868 / ZB8681.0

Notes:

Hot spot tables include:

- LSW_TASK
- LSW_TASK_EXECUTION_CONTEXT
- LSW_BPD_INSTANCE
- LSW_BPD_INSTANCE_DATA
- LSW_BPD_INSTANCE_VARIABLES
- LSW_EM_TASK
- LSW_USR_XREF
- LSW_USR_GRP_XREF
- LSW_PO_VERSIONS

For systems that have a large amount of business data and are used in searches (for example 50+) the Process Portal tasks and instance queries might operate more slowly than expected. When you use the save search acceleration tools to optimize your process searches, Process Portal searches that constrain on business data are faster.

Tip: You optimize a process search only on runtime or production systems. It is suggested that you do not optimize process searches on systems that are used for development, or for systems where changes happen frequently to deployed applications.

WebSphere Education IBM

Best practices for BPEL workloads

- Verify that optimizer statistics are current for hot spot tables
  - Monitor and establish a periodic reorg check for various hot spot tables
- Use the query table API for better query response times
- Use inline custom properties for the most efficient filtering
- Limit the number of queryable variables and custom properties
- Create indexes and drop unutilized indexes
- Apply database optimization profiles

Figure 12-25. Best practices for BPEL workloads WB868 / ZB8681.0

Notes:

Hot spot tables include:

- PROCESS_CONTEXT_T
- PROCESS_INSTANCE_B_T
- SCOPED_VARIABLE_INSTANCE_B_T
- SCOPE_INSTANCE_B_T
- ACTIVITY_INSTANCE_B_T
- TASK_INSTANCE_T
- WORK_ITEM_T
- SWI_T
- EVENT_INSTANCE_B_T

WebSphere Education IBM

Unit summary

Having completed this unit, you should be able to:

- Explain general database tuning considerations
- Identify tuning practices that are specific to DB2
- Identify tuning practices that are specific to Oracle
- Describe best practices for BPMN workloads
- Describe best practices for BPEL workloads

Figure 12-26. Unit summary WB868 / ZB8681.0

Notes:

WebSphere Education

Checkpoint questions

1. True or false: It is a good practice to place log file and table space containers on separate physical disks

2. True or false: The most important DB2 utility you need to run is **db2support** to collect troubleshooting data.

Figure 12-27. Checkpoint questions WB868 / ZB8681.0

Notes:

Write your answers here:

1.

2.

WebSphere Education IBM

Checkpoint answers

1. True

2. True

Figure 12-28. Checkpoint answers WB868 / ZB8681.0

Notes:

Unit 13. Java memory issues

What this unit is about

This unit provides an overview of the JVM, garbage collection policies, and methods on analyzing verbose garbage collection data.

What you should be able to do

After completing this unit, you should be able to:

- Explain the basic components of garbage collection (GC): mark, sweep, and compact
- Explain the concept of Java memory management
- Describe some of the myths that surround Java memory management
- Collect verbose GC trace logs from a server at run time
- Analyze verbose GC trace logs to diagnose potential Java memory management issues

How you will check your progress

- Checkpoint
- Lab exercise

References

IBM Business Process Manager V8.5 IBM Knowledge Center

`http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/ditamaps/ic-homepage-bpm.html`

WebSphere Education IBM

Unit objectives

After completing this unit, you should be able to:

- Explain the basic components of garbage collection (GC): mark, sweep, and compact
- Explain the concept of Java memory management
- Describe some of the myths that surround Java memory management
- Collect verbose GC trace logs from a server at run time
- Analyze verbose GC trace logs to diagnose potential Java memory management issues

Figure 13-1. Unit objectives WB868 / ZB8681.0

Notes:

WebSphere Education

Topics

- Java garbage collection
- Java memory diagnostics that use verbose GC traces
- Sizing the Java heap
- JVM tools

Figure 13-2. Topics WB868 / ZB8681.0

Notes:

13.1.Java garbage collection

This topic covers the basic concepts of a Java garbage collector. The garbage collection stages are explained: mark, sweep, and compact.

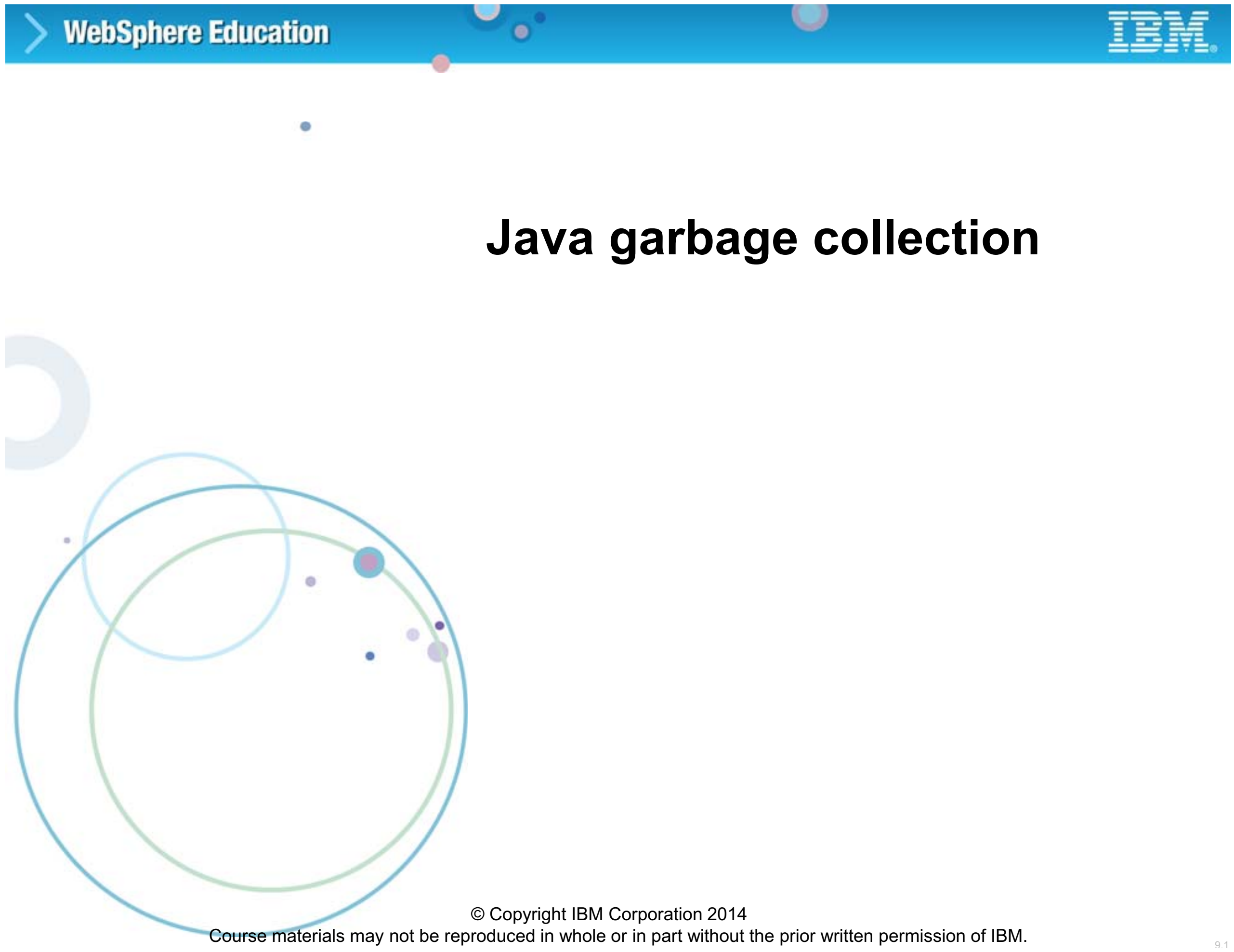


Figure 13-3. Java garbage collection WB868 / ZB8681.0

Notes:

WebSphere Education

Memory issues

- Memory issue symptoms:
  - Application response time is getting slower, or gives strange results such as blank pages, or error pages
  - You see **Out Of Memory** or **Totally Out Of Space** errors in **`SystemOut.log`** or **`SystemErr.log`** files
  - Process size is growing, as shown by operating system tools
  - Amount of free memory as seen in Tivoli Performance Viewer trends downward
  - A heap dump file was created

> What to do?
>
> Set an appropriate Java heap size to deliver optimal throughput and response time

Figure 13-4. Memory issues WB868 / ZB8681.0

Notes:

There are two types of memory problems, which throw an OutOfMemoryError:

- If a Java object cannot be allocated
- If there is no more memory available for native code allocations (c `malloc`) or threads

The first case is called Java heap exhaustion and the second, a native memory issue.

Finding the users of native memory can be a lengthy process, as there is no way of knowing which code allocated the native memory. There are no comparable heap dumps for native memory, so you must use trial and error, going through each of the potential native memory users until you find the one that is causing the problem.

WebSphere Education

Just-in-time compiler (JIT) basics

- The just-in-time compiler (JIT) is essential for a high-performing Java application
  – Java is write-once-run-anywhere; thus it is interpretive by nature and without the JIT, cannot compete with native code applications
- The JIT compiler is enabled by default for an application server
- The JIT works by compiling bytecode that is loaded from the Classloader when an application accesses it
  – Because different operating systems have different JIT compilers, there is no standard procedure for when a method is compiled
  – As your code accesses methods, the JIT determines how frequently specific methods are accessed
  – Methods that are used often are compiled to optimize performance

Figure 13-5. Just-in-time compiler (JIT) basics WB868 / ZB8681.0

Notes:

All the JVMs that are currently used commercially come with a supplementary compiler that takes bytecode and outputs platform-dependent computer code. This compiler works with the JVM to select parts of the Java program that would benefit from the compilation of bytecode. It replaces these areas of bytecode with concrete code for the JVM. This process is called just-in-time (JIT) compilation.

WebSphere Education IBM

JVM version

- WebSphere supports several JVMs based on version and operating system type
  - Windows, AIX, and Linux: IBM supplied
  - Can use only IBM SDK that zWSAS provides on z/OS
  - Solaris and HP-UX: Hybrid of IBM add-ons and vendor supplied JVM
- For a comprehensive list of the supported JVMs, check
  - For Standard: `http://www.ibm.com/support/docview.wss?uid=swg27023007`
  - For Advanced: `http://www.ibm.com/support/docview.wss?uid=swg27023005`
- To determine the JVM version in use:
  - Look in the `SystemOut.log` file of one of the profile instances

Figure 13-6. JVM version WB868 / ZB8681.0

Notes:

JVMs differ on the available tools that are supported and can have different behavior, even between different versions that are targeted for the same platform. Always look at the documentation for the specific JVM for behavioral descriptions and the options to control the JVM.

IBM Business Process Manager V8.5 supports the Java Development Kit (JDK) Version 6.0. V8.5 does not work with the Java Development Kit (JDK) Version 7.0.

For Oracle Solaris, WebSphere Application Server contains an embedded copy of the Oracle Solaris JVM along with some IBM add-ons, such as security, XML, and ORB packages. The WebSphere Application Server Solaris SDK is, therefore, a hybrid of Oracle and IBM products. However, the core JVM and JIT are Oracle Solaris.

For HP-UX, WebSphere Application Server contains an embedded copy of the HP JVM alongside some IBM add-ons, such as security packages. The WebSphere Application Server HP SDK is, therefore, a hybrid of HP and IBM products. However, the core JVM and JIT are HP software.

WebSphere Education IBM

Java virtual machine (JVM) memory management

- Java virtual machine (JVM) is responsible for memory (object) allocation and deallocation
  - Allocation requests for new objects are detected, and a memory management routine is invoked
  - Deallocation is more complex; JVM does not know when an object becomes free
  - JVM implements garbage collection; an allocation failure requires the JVM to gather all the unused objects in storage (heap) and merge them whenever possible
  - If the allocation is satisfied, operation continues; otherwise, the JVM fails with an `OutofMemory` exception
- Configuration and tuning of the JVM heap and **garbage collector (GC)** are key to a well-performing system
  - Many variations of memory management via garbage collection are implemented
  - A new method of garbage collection: Generational collection is the preferred implementation of memory management

Figure 13-7. Java virtual machine (JVM) memory management WB868 / ZB8681.0

Notes:

The Java virtual machine (JVM) is an interpretive computing engine that is responsible for running the bytecode in a compiled Java program. The JVM translates the Java bytecode into the native instructions of the host computer. The application server, being a Java process, requires a JVM to run and to support the Java applications that are running on it. JVM settings are part of an application server configuration.

The JVM is called “virtual” because it provides an interface that is independent of the underlying operating system and computer hardware architecture. This independence from hardware and operating system is a cornerstone of the write-once-run-anywhere value of Java programs. Java programs are compiled into bytecode that targets the abstract virtual machine; the JVM is responsible for running the bytecode on the specific operating system and hardware combinations.

WebSphere Education

Java memory management basics

- Live set versus garbage:
  - Live set is objects that are currently referenced
  - Garbage is objects that are not currently referenced
- Fragmentation:
  - You should not expect to allocate an object that is as large as the amount of free space in the heap immediately after a GC cycle
- Contiguous memory:
  - Java objects must occupy contiguous memory
- Thread local heaps (TLHs):
  - Threads reserve a chunk of free heap to allocate from; TLH reduces contention on allocation lock, and keeps code running in a straight line (fewer failures)
  - Available for objects less than 512 bytes in size
- Finalizer:
  - Finalizers should be avoided

Figure 13-8. Java memory management basics WB868 / ZB8681.0

Notes:

Fragmentation exists on both IBM and Sun JVMs.

WebSphere Education

Garbage collection basics: Mark, sweep, compact

- **Mark** phase identifies objects that are not used
  - Objects no longer have references to them from active objects
  - Objects no longer reachable
- **Sweep** phase collects garbage objects to meet garbage collection target (satisfy the allocation request)
- **Compact** phase collects and compacts the resulting freed-up spaces

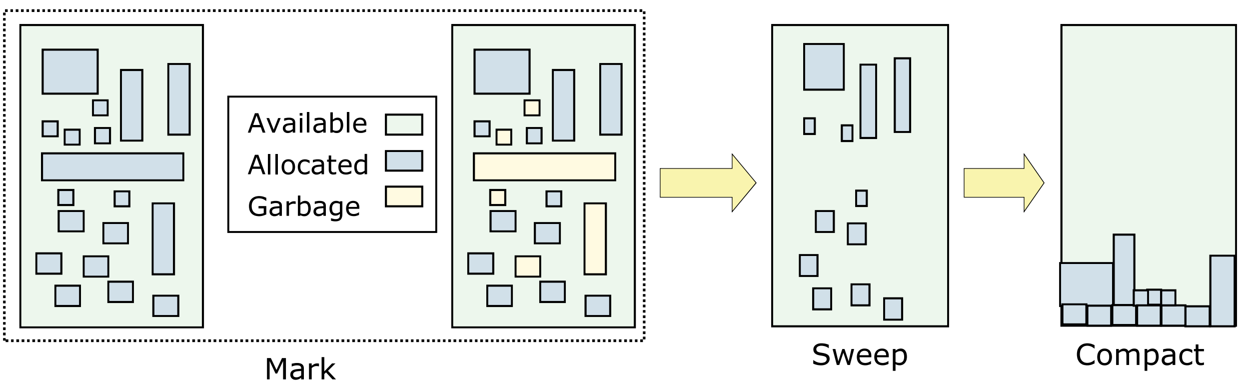


Figure 13-9. Garbage collection basics: Mark, sweep, compact WB868 / ZB8681.0

Notes:

When the JVM cannot allocate an object from the current heap due to the lack of contiguous space, a memory allocation fault occurs, and the GC is invoked. The first task of the GC is to collect all of the garbage that is in the heap. This process starts when any thread calls the GC either indirectly as a result of allocation failure, or directly by a specific call to `System.gc()`. The first step is to acquire exclusive control on the virtual machine to prevent any further Java operations. Garbage collection can then begin.

WebSphere Education

IBM JDK GC process

- **Mark**: Recursively marks all the live objects, starting with the registers and thread stacks
- **Sweep**: Frees all the objects that were not marked in the mark phase
- **Compaction**: Reduces heap fragmentation
  - This phase attempts to move all live objects to one end of the heap, freeing up large areas of contiguous free space at the other end
  - Compaction stops JVM activity while it occurs
  - Not every GC cycle results in a compaction
- **Parallel** mark and sweep process uses main and multiple helper threads (number of processors minus one) to process tasks
- **Concurrent** mark and sweep can be configured
  - It starts a concurrent marking and sweeping phase before the heap is full
  - The mark and sweep phase runs while the application is still running

Figure 13-10. IBM JDK GC process WB868 / ZB8681.0

Notes:

The IBM Garbage Collector is by default a “stop-the-world” (STW) operation because all application threads are stopped while the garbage is collected.

Concurrent mark can be configured. This configuration starts a concurrent marking phase before the heap is full. The mark phase runs while the application is still running, in effect, attempting to trade application performance for possible smaller garbage collection times.

Mark stack overflow (MSO) is a rare event that can occur. Because the mark stack has a fixed size, it can overflow. This overflow has a negative impact on pause time:

- Mark process resumes where the overflow occurred
- Repeats <n> times until no more objects require marks

A parallel version of garbage collector mark exists. The time spent marking objects is decreased through the addition of helper threads and a facility that shares work between those threads.

Incremental compaction is a way of spreading compaction work across a number of garbage collection cycles, reducing pause times. Another important task for incremental compaction is the removal of dark matter. Dark matter is the term for small pieces of free space (currently less than

512 bytes in size) that are not on the free list and therefore are not available for allocation of objects.

WebSphere Education IBM

Java memory management myths

- **Myth**: The garbage collector is going to reclaim that memory eventually
  - **Truth**: If an object is not garbage, it cannot be collected
- **Myth**: Objects never need to be cleaned out of **`Vectors`** and **`Hashtables`**
  - **Truth #1**: If the parent object of a **`Vector`** or **`Hashtable`** is long-lived, objects added to its **`Vectors`** or **`Hashtables`** must be removed or “nulled out” when that object is no longer required
  - An example would be an ORB servant
  - **Truth #2**: A **`Hashtable`** or **`Hashmap`** does not become garbage until all of its elements are removed
- **Myth**: All finalizers need to be eliminated
  - **Truth**: Finalizers should be ***avoided***
  - If you encounter an **out of memory** with BPM, it is unlikely that finalizers are the root cause of your memory issue

Figure 13-11. Java memory management myths WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Garbage collection policies

Memory management is configurable by using four different policies with varying characteristics

- **Generational concurrent**: (New default) divides heap into “nursery” and “tenured” segments that provide fast collection for short-lived objects
  – Can provide maximum throughput with minimal pause times
- **Optimize for throughput**: Flat heap collector that is focused on maximum throughput
- **Optimize for pause time**: Flat heap collector with concurrent mark and sweep to minimize GC pause time
- **Balanced**: New policy
  – Uses a region-based layout for the Java heap
  – These regions are individually managed to reduce the maximum pause time on large heaps and increase the efficiency of garbage collection
- **Subpool**: This option is now deprecated and is treated as an alias for optimize for throughput

Figure 13-12. Garbage collection policies WB868 / ZB8681.0

Notes:

The former default policy was “optimize for throughput”. It is a “stop-the-world” policy where the JVM does not do any other work. To optimize the throughput, the garbage collection does all of its work during the GC cycle and does not affect the application when not active.

The `-Xgcpolicy` command-line parameter is used to enable and disable concurrent mark:

`-Xgcpolicy:<optthruput | optavgpause | gencon | subpool>`

The `-Xgcpolicy` options have these effects:

- `Optthruput` disables concurrent mark. If you do not have pause time problems (as seen by erratic application response times), you get the best throughput with this option. *Optthruput* is the default setting.
- `Optavgpause` enables concurrent mark with its default values. If you are having problems with erratic application response times that normal garbage collections cause, you can reduce those problems at the cost of some throughput, by using the `optavgpause` option.
- `Gencon` requests the combined use of concurrent and generational GC to help minimize the time that is spent in any garbage collection pause.

- `Balanced` is a new policy that uses a region-based layout for the Java heap. These regions are individually managed to reduce the maximum pause time on large heaps and increase the efficiency of garbage collection. The policy also uses a different object allocation strategy that improves application throughput on large systems that have nonuniform memory architecture (NUMA) characteristics
- `Subpool` is used before version 8. It is a flat heap technique to help increase performance on large SMP systems with 16 or more processors by optimizing the object allocation and is available only on IBM pSeries and zSeries.

This option is now deprecated. The subpool option is treated as an alias for optimize for throughput. Therefore, if you use this option, the effect is the same as optimize for throughput.

WebSphere Education IBM

Generational concurrent garbage collector

Generational concurrent garbage collector routines classify objects by age

- Activate the generational concurrent garbage collector with the **`Xgcpolicy:gencon`** command-line option

IBM JVM has two generations:

- Young generation:
  - Most objects in any program have a short lifespan
  - They are created, used, and abandoned rapidly (these objects are responsible for most heap fragmentation)
  - Generational collector identifies and clears objects from memory early
- Tenured generation:
  - Aged objects are promoted to the tenured (old) generation
  - The collector tries to avoid garbage collection in this generation
  - Collection in the tenured generation tends to be expensive
  - Long-lived objects are not collectable and must be compacted to obtain space

Figure 13-13. Generational concurrent garbage collector WB868 / ZB8681.0

Notes:

The active state of the JVM is made up of the set of stacks that represent the threads, the static objects that are inside Java classes, and the set of local and global JNI references.

The GC scans the thread stacks and looks for pointers that appear to be pointing to Java objects (that is, within the Java heap range). It is likely that there are attributes (for example, floats) being pushed onto the stack that look like Java object references; that is why GC marks them as “dosed” to make them unmovable during compaction.

The root set also includes pinned objects, JNI references, and more.

All the objects that pointers reference are reachable objects; they are listed in a mark vector, which is used to compare with the allocbits vector. The allocbits vector contains the information that the GC needs for all of the objects created. From the difference between the mark vector and the allocbits vector, the GC can determine what needs to be collected.

WebSphere Education

Generational concurrent garbage collector (gencon)

- Young objects are created and kept in a **nursery** area
  - Cheaply and frequently reclaiming their space as they die
  - Nursery is split into **allocate** and **survivor** spaces
  - Objects are born in the allocate space
- Objects that survive into adulthood move to the **survivor** area
  - Garbage collection runs infrequently: run only on objects most likely to be collected
  - Garbage collection operating cost is spread across frequent, shorter calls: avoids long compaction delays

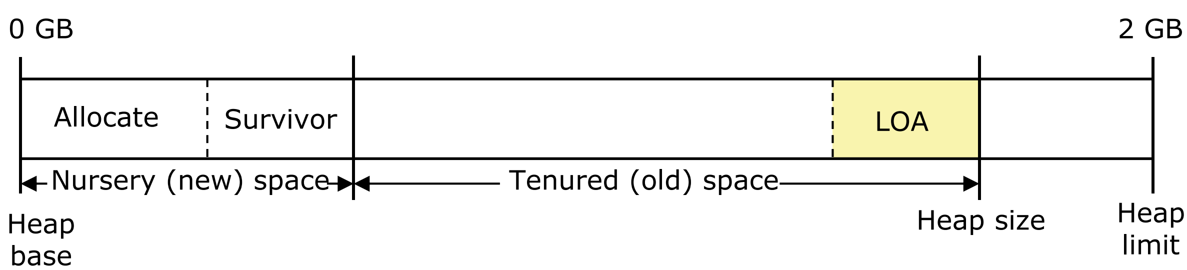


Figure 13-14. Generational concurrent garbage collector (gencon) WB868 / ZB8681.0

Notes:

The generational concurrent garbage collector was introduced in Java 5.0 from IBM. A generational garbage collection strategy is suited to an application that creates many short-lived objects, as is typical of many transactional applications.

WebSphere Education

Generational concurrent GC: Scavenge (1 of 2)

- When the allocate space is full, a GC is triggered
- The allocate space is traced
- Objects that reach the tenured age (survived a specific number of scavenge operations; maximum is 14) are copied into the tenured area

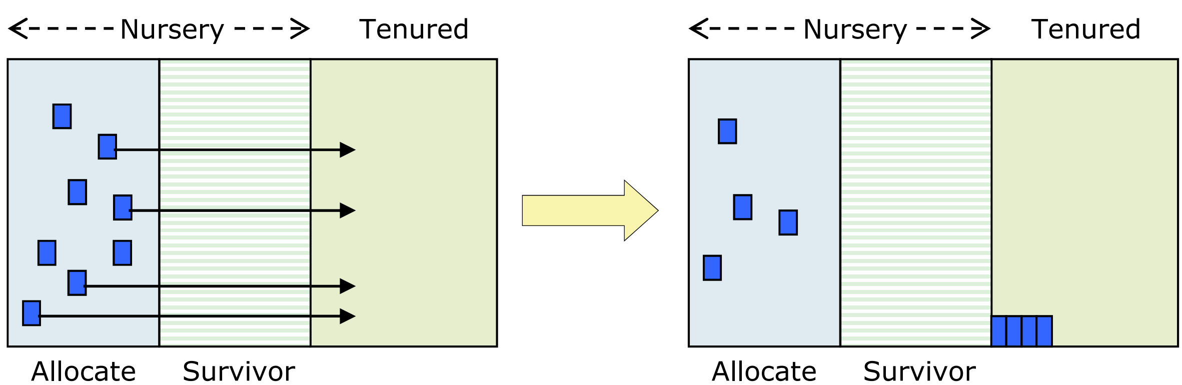


Figure 13-15. Generational concurrent GC: Scavenge (1 of 2) WB868 / ZB8681.0

Notes:

Different algorithms are used in each generation. For young generations, compaction or by-copy algorithms are used. For older tenured generations, the normal mark and sweep algorithms are used.

The Java heap is split into two areas, a new (or nursery) area and an old (or tenured) area. Objects are created in the new area, and if they live long enough, they are moved or promoted into the old area. Objects are promoted when they reach a certain age (known as the tenure age). This age is a count of the number of garbage collections for which they are alive.

Tenure age is a measure of the object age at which it should be promoted to the tenure area. The JVM dynamically adjusts this age, and reaches a maximum value of 14. The age of an object is incremented on each scavenge. A tenure age of x implies that, if the object survives x flips between survivor and allocate space, it is promoted. The threshold is adaptive and adjusts the tenure age that is based on the percentage of space that is used in the new area.

WebSphere Education

Generational concurrent GC: Scavenge (2 of 2)

- Remaining live objects are copied from the allocate space into the survivor space, and a count of the number of times each object was scavenged is incremented
- The copied objects are placed contiguously in the survivor space so that an effective compaction is performed
- At the end of scavenge, the survivor space and allocate space pointers are flipped
  - The survivor space becomes the new allocate space and is empty

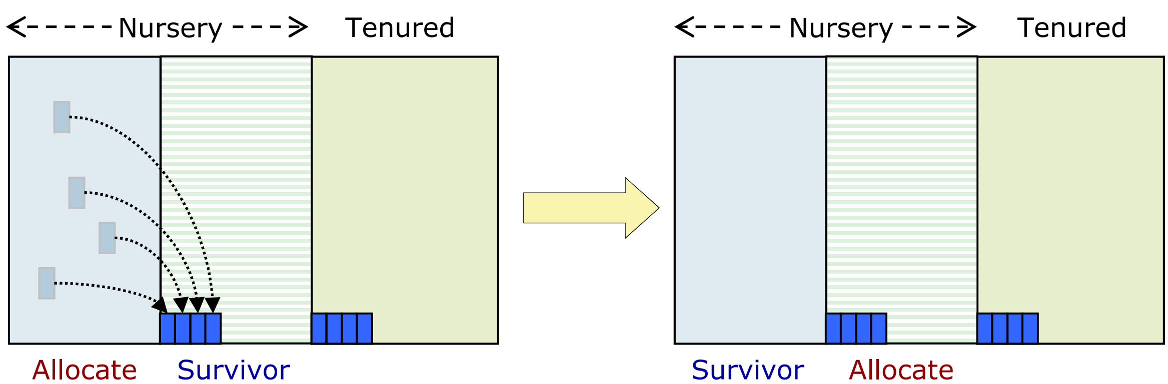


Figure 13-16. Generational concurrent GC: Scavenge (2 of 2) WB868 / ZB8681.0

Notes:

The new area is split into two logical spaces: allocate and survivor. Objects are allocated into the allocate space. When that space is filled, a GC process that is called scavenge is triggered. During a scavenge, live objects are copied either into the survivor space or into the tenured space if they are old enough to reach the tenured age. Then, the roles are reversed: the survivor space becomes the allocate space.

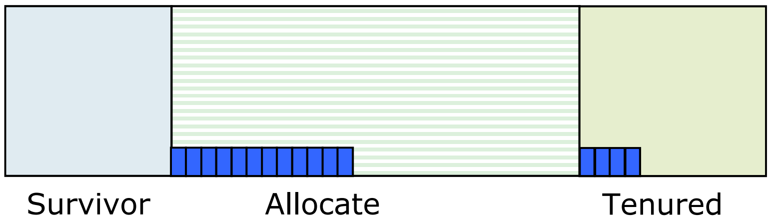
Dead objects in the nursery remain untouched. When all the live objects are copied, the spaces in the new area switch roles. The new survivor space is now entirely empty of live objects and is available for the next scavenge. Any objects that were not moved into the old or tenured space are now in the “new” allocate space.

A technique that is called *tilting* maximizes the size of the allocate space in the new area. Tilting controls the relative sizes of the allocate and survivor spaces. Based on the amount of data that survives, the ratio can be adjusted to make the survivor space smaller.

WebSphere Education

Generational concurrent GC: Self-adjusting

The scavenger automatically adjusts the relative sizes of the allocate and survivor spaces as based on history and predictions (*tilt*)



The generational collector is designed to respond dynamically to varying conditions so that its behavior is adaptive

- These mechanisms are:
  - Concurrent collection runs in the tenured area
  - In many cases, it means that one rarely or never sees a full classic collection (mark, sweep, compact) in the tenured area
  - The sizes of the young and tenured generations will self-adjust over time as based on memory pressure
  - The scavenger adjusts the flip count (number of scavenges that an object must survive in order to tenure); it is based on history
  - The tilt ratio self-adjusts as based on history

Figure 13-17. Generational concurrent GC: Self-adjusting WB868 / ZB8681.0

Notes:

A technique that is called tilting maximizes the size of the allocate space in the new area. Tilting controls the relative sizes of the allocate and survivor spaces. Based on the amount of data that survives, the ratio can be adjusted to make the survivor space smaller.

13.2. Java memory diagnostics that use verbose GC traces

WebSphere Education

Java memory diagnostics that use verbose GC traces

9.1

Figure 13-18. Java memory diagnostics using verbose GC traces WB868 / ZB8681.0

Notes:

WebSphere Education

Choosing the right GC policy

There are four GC policies; each optimizes for different scenarios

- `-Xgcpolicy:gencon` (Default) Optimized for highly transactional workloads
  - Use when fragmentation can be an issue (applications create large objects that are short lived)
- `-Xgcpolicy:optthruput` Optimized for batch type applications
- `-Xgcpolicy:optavgpause` Optimized for applications with responsiveness criteria
- `-Xgcpolicy:balanced` Optimized to reduce the maximum pause time on large heaps and increase the efficiency of garbage collection

How do you know which policy to use?

- Monitor GC activity by using **verbose GC** trace log
- Verbose GC output analysis can be done with *GC and Memory Visualizer* (GCMV), or *IBM Pattern Modeling and Analysis Tool* (PMAT) which are available through IBM Support Assistant tool
- Look for certain heap characteristics (such as frequency of allocation failures and time to process allocation failures)

Figure 13-19. Choosing the right GC policy WB868 / ZB8681.0

Notes:

Setting gcpolicy to optthruput disables concurrent mark. If you do not have pause time problems, denoted by erratic application response times, you should get the best throughput by using this option. Setting gcpolicy to optavgpause enables concurrent mark with its default values. This setting alleviates erratic application response times that result during normal garbage collection. However, this option might decrease the overall throughput.

WebSphere Education

Garbage collection debugging and analysis tools (verbose:gc)

- Always run your application server with **`-verbose:gc`** (GC trace log) turned on, even in production
  - Cost is minimal (at least on the IBM JVM)
  - Invaluable source of information, since the data was collected directly from the JVM runtime

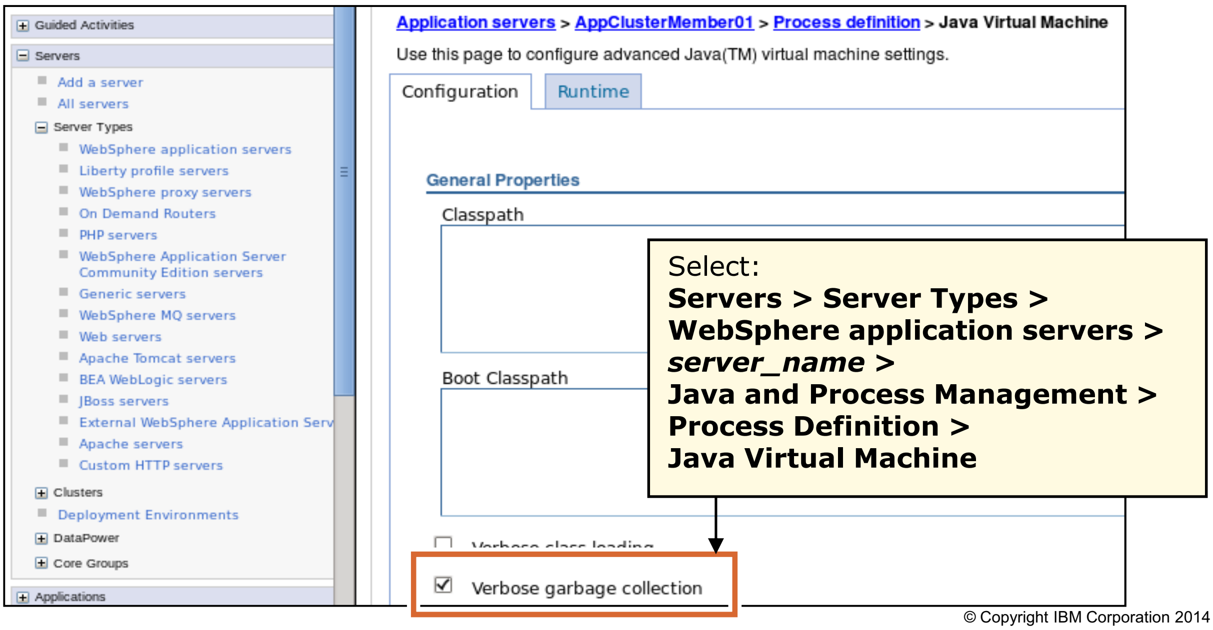


Figure 13-20. Garbage collection debugging and analysis tools (verbose:gc) WB868 / ZB8681.0

Notes:

The garbage collector is enabled by entering `–verbose:gc` on the Java command line.

Advantages:

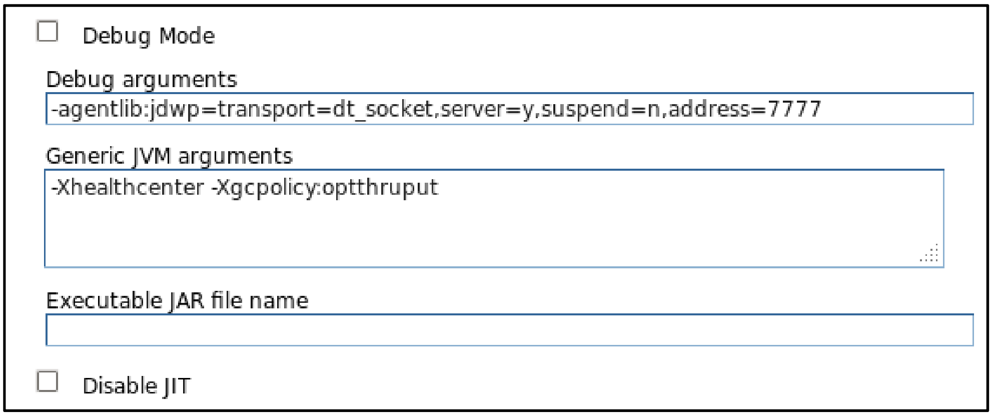
- Provides detailed low-level information for serious debugging, enough for initial investigation
- Readily available and it is free

Disadvantages:

- Have to restart your server, which is not suitable for production environments
- Does not give object-level information for further analysis

JVM configuration

- Administrative console JVM configuration tab



- The **Generic JVM arguments** window is where you can specify:
  - GC policy (**`-Xgcpolicy`**)
  - Use several garbage collection threads (**`-Xgcthreads`**)
  - Disable class garbage collection (**`-Xnoclassgc`**)
  - Disable explicit garbage collection (**`-Xdisableexplicitgc`**)
  - Many more

Figure 13-21. JVM configuration WB868 / ZB8681.0

Notes:

The Generic JVM arguments window is where you specify command-line arguments to pass to the Java virtual machine code that starts the application server process. You can enter the optional command-line arguments in the Generic JVM arguments field. If you enter more than one argument, enter a space between each argument.

A manual call to the garbage collector (for example, through the `System.gc()` call) suggests that a garbage collection cycle runs. In fact, the call is interpreted as a request for a full garbage collection scan unless a garbage collection cycle is already running or explicit garbage collection is disabled by specifying: `-Xdisableexplicitgc`

`-Xnoclassgc`: By default, the JVM unloads a class from memory whenever there are no live instances of that class left. The cost of loading and unloading the same class multiple times can decrease performance.

You can use the `-Xnoclassgc` argument to disable class garbage collection. However, the performance cost of class garbage collection is typically minimal. Turning off class garbage collection in a Java EE-based system might effectively create a memory leak of class data, and cause the JVM to throw an out-of-memory exception. If you use this option, whenever you redeploy an application, you should always restart the application server to clear the classes and static data from the previous version of the application.

WebSphere Education

Key Java GC statistics

- Java heap size
- Live set size and free space size
- Memory released
- Time in GC
- Time since last GC
- Size of allocation request
  - Size of allocation request versus size of free space

native_stderr.log


```
<?xml version="1.0" ?>

<verbosegc xmlns="http://www.ibm.com/j9/verbosegc"
version="R26_Java626_SR6_Ifix_20130823_2006_B162690_CMPRSS">

<initialized id="1" timestamp="2014-05-27T16:52:28.267">
  <attribute name="gcPolicy" value="-Xgcpolicy:gencon" />
  <attribute name="maxHeapSize" value="0x140000000" />
  <attribute name="initialHeapSize" value="0x30000000" />
  <attribute name="compressedRefs" value="true" />
  <attribute name="compressedRefsDisplacement" value="0x0" />
  <attribute name="compressedRefsShift" value="0x3" />
  <attribute name="pageSize" value="0x1000" />
  <attribute name="pageType" value="not used" />
  <attribute name="requestedPageSize" value="0x1000" />
  <attribute name="requestedPageType" value="not used" />
  <attribute name="gcthreads" value="1" />
  <attribute name="numaNodes" value="0" />
```


Figure 13-22. Key Java GC statistics WB868 / ZB8681.0

Notes:

Garbage collection can be used to evaluate application performance health. By monitoring garbage collection during the execution of a fixed workload, users gain insight about whether the application is over-utilizing objects. Garbage collection can even be used to detect the presence of memory leaks.

Use the garbage collection and heap statistics in Tivoli Performance Viewer to evaluate application performance health. You must enable JVMTI to get detailed garbage collection statistics. By monitoring garbage collection, memory leaks and overly used objects can be detected.

WebSphere Education

Important characteristics for choosing GC policy

- Rate of GC
  - High rates of object burn point to large numbers of transitional objects, and therefore the application might benefit from the use of **`gencon`**
- Large object allocations
  - The allocation of very large objects adversely affects **`gencon`** unless the nursery is sufficiently large
  - The application might benefit from **`optavgpause`**
- Large heap usage variations
  - The **`optavgpause`** algorithms are best suited to consistent allocation profiles
  - Where large variations occur, **`gencon`** might be better suited

Rule of thumb: If GC operating cost is larger than 10%, you most likely chose the wrong policy

Figure 13-23. Important characteristics for choosing GC policy WB868 / ZB8681.0

Notes:

Setting gcpolicy to optthruput disables concurrent mark. If you do not have pause time problems, denoted by erratic application response times, you should get the best throughput by using this option. Setting gcpolicy to optavgpause enables concurrent mark with its default values. This setting alleviates erratic application response times that result during normal garbage collection. However, this option might decrease the overall throughput.

WebSphere Education IBM

Issues that are illustrated with garbage collection statistics

Analyzing garbage collection statistics (verbose GC traces) can diagnose potential Java memory issues

- Memory leaks:
  - Indication: Live set increases over time
- Large objects:
  - Indication: Live set much smaller than heap size at time of out of memory crash
- Excessive garbage collection:
  - Time that is spent in GC can be reduced by reducing the rate at which GC is invoked
  - The rate at which GC is invoked can be reduced by increasing the heap size
  - Increasing the heap size increases the pauses associated with the GC process
  - 50% heap occupancy is a good tradeoff point

Figure 13-24. Issues illustrated by garbage collection statistics WB868 / ZB8681.0

Notes:

An application server that does not respond might be hung, or the process might be terminated. Users see hung applications or are not able to access new applications.

An application server can suffer performance degradation when an application server is repeatedly failing and being restarted automatically.

If CPU activity is low but the application server is not terminated, you most likely have a hang or deadlock situation. If CPU activity is high and the application server is using the cycles, you most likely have a loop or inefficient code.

An often encountered condition is out-of-memory (OOM), which manifests itself either as an unexpected process exit with an “OutOfMemoryException” or just a list of errors and exceptions but no immediate process exit. This situation can occur when the application server is running low on memory, due to an application problem such as a memory leak, a hardware memory failure, or higher than usual demand.

WebSphere Education

Java memory leak: Cause of the memory leak

- Objects that are referenced by another object cannot be cleaned
- Usual code issues: Objects that are referenced from a `Collection`:
  - The `Collection` is never cleaned
  - Objects that are created in a loop
  - Objects that are referenced from a static variable
- Usual reaction to a memory leak is to raise the maximum heap size (`-Xmx`)
  - This action does not fix the problem, and the system crashes later
  - The larger the heap, the more work for GC at mark time

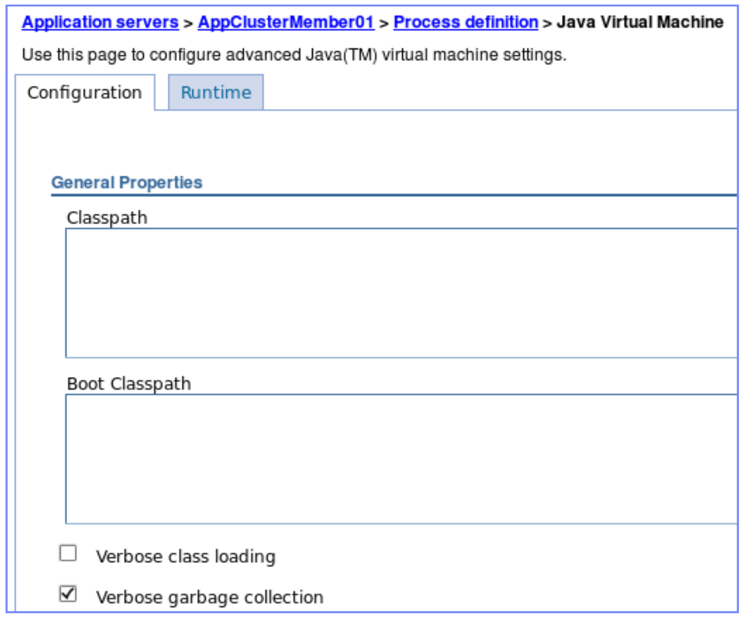


Figure 13-25. Java memory leak: Cause of the memory leak WB868 / ZB8681.0

Notes:

An OutOfMemoryError (OOM) is a Java exception that occurs when the JVM is unable to allocate memory to satisfy a request. There are four potential reasons for an OOM to occur. The first reason is that the Java heap is too small for the amount of workload. Second is that the memory in the Java heap is fragmented and there is not enough contiguous free space to satisfy the request. Third is that some part of the Java code is leaking memory and the heap space is exhausted. Finally, the JVM might not have enough native memory to call operations that are associated with the Java object allocation. In any case, the first artifact to analyze is the javacore that is created by default whenever an OOM occurs; however, verboseGC logs and heap dumps are also valuable in resolving OOMs.

WebSphere Education

Common causes of memory leak (1 of 2)

- Listeners
  - By installing a listener, you effectively attach your object to a static reference
  - This object cannot be collected while the listener is active
  - You must explicitly uninstall a listener when you finish using the object to which you attached it
- Hash tables
  - Anything that is added to a hash table creates a reference to your object from the hashed object
  - Hashed objects cannot be collected unless they are explicitly removed from any hash table to which they were added
- Static class data
  - Static class data exists independently of instances of your object
  - Anything that it points to cannot be collected even if no instances of your class are present that contain the static data

Figure 13-26. Common causes of memory leak (1 of 2) WB868 / ZB8681.0

Notes:

WebSphere Education

Common causes of memory leak (2 of 2)

- Java Native Interface (JNI) references
  - Objects that are passed from the JVM to native code across the JNI have a reference to them that is held in the JNI code of the JVM
  - The native code application must explicitly clear references before they can be collected
- Objects with finalizers
  - Objects that have finalizers cannot be collected until the finalizer runs

Figure 13-27. Common causes of memory leak (2 of 2) WB868 / ZB8681.0

Notes:

WebSphere Education

Shrinking heap issue: Conditions for heap shrinkage

- Heap occupancy is under 40%
- The following is ***not*** true:
  - Heap was recently expanded (last three GC cycles)
  - GC is a result of a `System.GC()` call
- Compaction occurs if:
  - An object exists in the area that is being shrunk
  - GC did not shrink on the previous cycle
- Compaction is therefore likely to occur

Figure 13-28. Shrinking heap issue: Conditions for heap shrinkage WB868 / ZB8681.0

Notes:

The first probable cause for an OOM is that the heap is not large enough for the workload. This problem often surfaces shortly after the JVM begins to process the workload, and the heap usage increases until the maximum available space is used. A steady increase in heap use can also be an indicator of a memory leak; you are going to explore the differences when you get to the topic of memory leaks. For now, if you suspect the heap is too small, the size can be increased under the Java virtual machine section of the server's Process Definition area of the administrative console.

WebSphere Education

Doublers issue

- Many Java object types double when they need to expand
  - `java.util.Vector` (owns an array of `Objects`)
- Size is the number of objects in the container
- Capacity is the number of objects the container can hold
- When size exceeds capacity, the container doubles capacity
- The `Vector` doubles the capacity of its array of `Objects`
- An array that can hold more than 100 K objects or primitives is going to be megabytes in size, even if it is empty

Figure 13-29. Doublers issue WB868 / ZB8681.0

Notes:

WebSphere Education

Large objects in WebSphere products

- Processing large objects:
  - In general, objects of size 5 MB or larger are considered large
  - Objects of size 100 MB or larger are exceptionally large, and generally require significant tuning to be processed successfully
  - Large objects exist in both the server and adapters
- Factors affecting large object size processing:
  - Java heap size limitations
  - Size of in-memory business objects
  - Number of concurrent objects that are being processed

Figure 13-30. Large objects in WebSphere products WB868 / ZB8681.0

Notes:

WebSphere Education

Large object design patterns

- Batch inputs
  - Decompose large business objects into smaller objects, and submit them individually
- Claim check pattern: When the process or mediation truly needs a few attributes, use the claim check pattern
  1. Detach the data payload from the message
  2. Extract the required attributes into a smaller control business object
  3. Persist the larger data payload to a data store, and store the claim check as a reference in the control business object
  4. Process the smaller control business object, which has a smaller memory footprint
  5. When the whole large payload is needed, check out the large payload from the data store
  6. Delete the large payload from the data store
  7. Merge the attributes in the control business object with the large payload

Figure 13-31. Large object design patterns WB868 / ZB8681.0

Notes:

What factors affect memory performance the most

- Memory management:
  – How efficiently does the system manage memory?
  – GC policy
- Total available memory:
  – Is there enough memory to satisfy every request for memory?
  – Heap size
- Allocation rate:
  – How often does the application issue requests for memory?
  – Application design
- Object size:
  – How large are these objects?
  – Application design
- Object lifetime:
  – How long do these objects stay reserved by the application?
  – Application design

Figure 13-32. What factors affect memory performance the most WB868 / ZB8681.0

Notes:

Enterprise applications that are written in the Java language involve complex object relationships and use large numbers of objects. Although the Java language automatically manages memory that is associated with object lifecycles, understanding the application usage patterns for objects is important. In particular, verify that the following conditions exist:

- The application is not leaking objects.
- The Java heap parameters are set properly to handle a specific object usage pattern.
- The application is not over-utilizing objects.

13.3. Sizing the Java heap

WebSphere Education

IBM

Sizing the Java heap

Figure 13-33. Sizing the Java heap WB868 / ZB8681.0

Notes:

WebSphere Education

Memory basics

Two heaps: Native heap and Java heap

- Native heap (system heap)
  - Has objects whose lifetime is the same as the JVM (such as class file)
  - Contains objects that are allocated via JNI; for example, database (type 2) connections
  - Thread stacks
  - Not controlled by the garbage collector
- Java heap
  - Contains all Java objects
  - `-Xmx` and `-Xms` options bound the Java heap

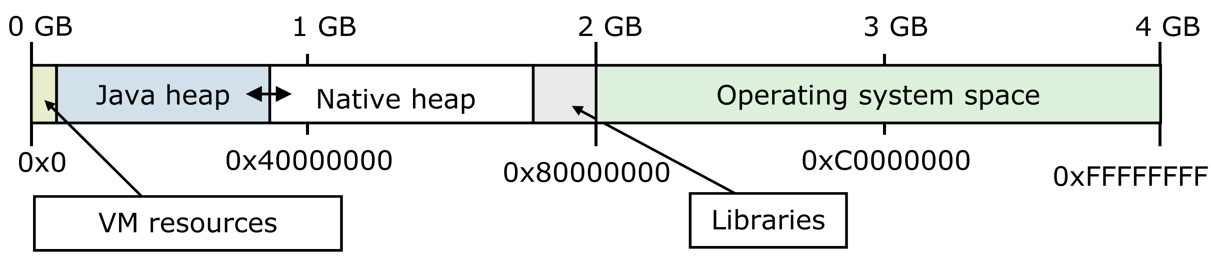


Figure 13-34. Memory basics WB868 / ZB8681.0

Notes:

The native heap available to an application is different from one operating system to another operating system.

WebSphere Education

Garbage collector (GC) managed Java heap sizing

- GC adapts heap size to keep occupancy between 40% and 70%
  - Heap occupancy over 70% causes frequent GC cycles, which generally means reduced performance
  - Heap occupancy below 40% means infrequent GC cycles, but cycles longer than they need to be
  - The result is longer pause times than necessary, and reduced performance
- The maximum heap size setting should therefore be 43% larger than the maximum occupancy of the application
  - (Maximum occupancy) + (43%) = occupancy at 70% of total heap

    Example: For 70 MB occupancy, 100 MB Max heap is required, which is 70 MB + 43% of 70 MB

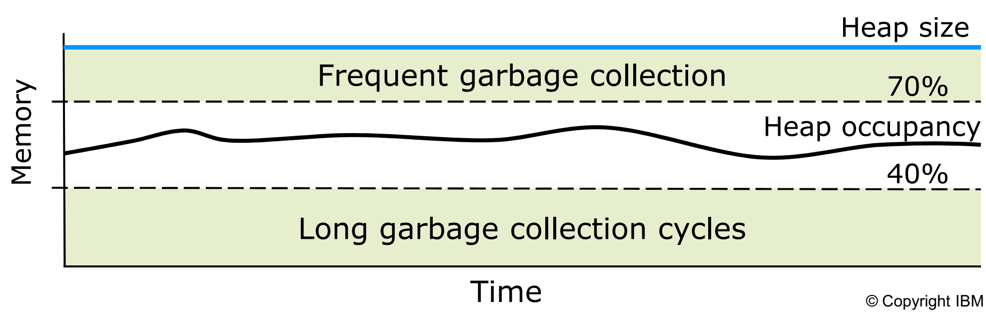


Figure 13-35. Garbage collector (GC) managed Java heap sizing WB868 / ZB8681.0

Notes:

Math lesson:

- MH = maximum heap size
- MO = maximum heap occupancy

You want MH such that 70% of MH = MO, or 0.7 MH = MO.

Doing the math,

MH = 1.43 MO

MH = MO + 43% of MH

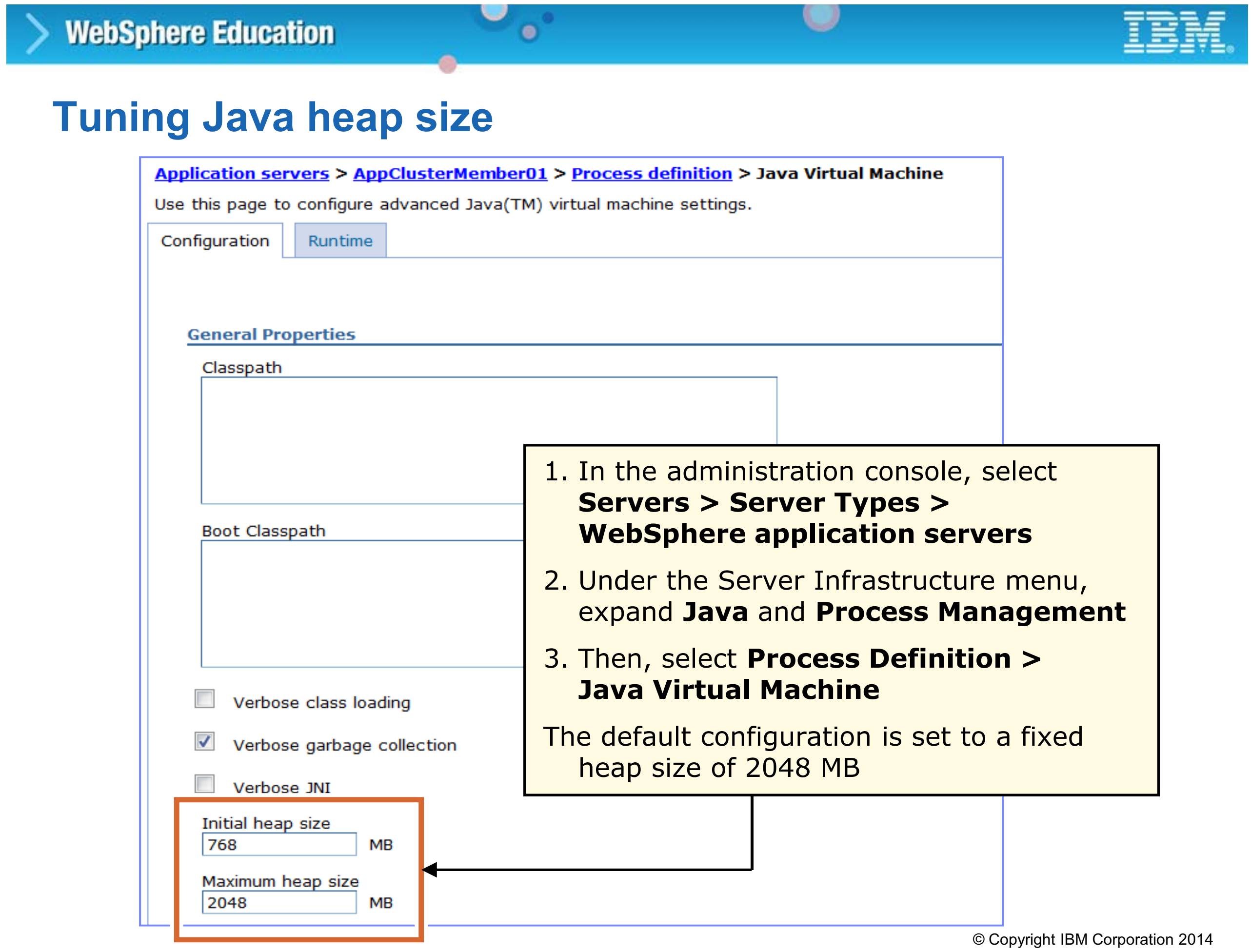


Figure 13-36. Tuning Java heap size WB868 / ZB8681.0

Notes:

You need to select the **Verbose garbage collection** check box on the Runtime tab. Selecting this check box enables verbose GC logging immediately without having to restart the application server.

The initial heap size specifies, in megabytes, the initial heap size available to the JVM code. If this field is left blank, the default value is used.

The maximum heap size specifies, in megabytes, the maximum heap size that is available to the JVM code. If this field is left blank, the default value is used.

The default maximum heap size is 256 MB. This default value applies for both 32-bit and 64-bit configurations.

Increasing the maximum heap size setting can improve startup. When you increase the maximum heap size, you reduce the number of garbage collection occurrences with a 10% gain in performance.

Increasing this setting usually improves throughput until the heap becomes too large to store in physical memory. If the heap size exceeds the available physical memory, and paging occurs, there is a noticeable decrease in performance. Therefore, it is important that the value you specify for this property allows the heap to be contained within physical memory.

WebSphere Education IBM

Fixed heap sizes versus variable heap sizes

- Fixed heap size:
  - Minimum heap size (-**Xms**) = Maximum heap size (-**Xmx**)
  - Fixed heap size does not expand or shrink the Java heap
  - Advised for maximum performance to avoid compact GC phase, which must run each time that the heap shrinks
- Variable heap sizes:
  - GC adapts the heap size to keep occupancy between 40% and 70%
  - Expands and shrinks the Java heap as necessary
  - Allows for scenario where usage varies over time, and variations would take usage outside the 40% – 70% window
  - Advised for large object situations
  - Creates head room such that heap can be grown with contiguous heap space if a large object needs to be allocated
  - If heap size equals maximum (**-Xmx**) heap size, you are subject to fragmentation issues and are at risk of encountering an out of memory for a large object
- Each option has advantages and disadvantages, and you must select which is right for the particular application

Figure 13-37. Fixed heap sizes versus variable heap sizes WB868 / ZB8681.0

Notes:

It is generally suggested to set the initial and maximum heap sizes to the same value to prevent the JVM from dynamically resizing the heap. This setting not only makes GC analysis easier by fixing a key variable to a constant value, but also avoids the performance cost that is associated with allocating and deallocating more memory. The trade-off for setting the minimum and maximum heap sizes to the same value is that the initial startup of the JVM is slower since the JVM must allocate the larger heap. There are scenarios where setting the minimum and maximum heap settings to different values is advantageous.

Refer to: WebSphere Application Server Performance Tuning Case Study at:

`http://www.ibm.com/developerworks/websphere/techjournal/0909_blythe/0909_blythe.html`

WebSphere Education IBM

Heap expansion and shrinkage

- The act of heap expansion and shrinkage is relatively cheap
- However, a **compaction** of the Java heap is sometimes required
  - **Expansion**: GC might already be compacted to try to allocate the object before expansion for some expansion
  - **Shrinkage**: GC might need to compact to move objects from the area of the heap that is shrinking
- Expansion and shrinkage optimize the heap occupancy; however, it usually does so ***at the cost of the compaction cycle***

Figure 13-38. Heap expansion and shrinkage WB868 / ZB8681.0

Notes:

When and why does the Java heap expand?

The JVM starts with a small default Java heap, and it expands the heap as a result of an application's allocation requests until it reaches the value that `-Xmx` specifies. Expansion occurs after GC if GC is unable to free enough heap storage for an allocation request, or if the JVM determines that expanding the heap is required for better performance.

When does the Java heap shrink?

Heap shrinkage occurs when GC determines that there is large amount of free heap storage, and releasing some heap memory is beneficial for system performance. Heap shrinkage occurs after GC, but when all of the threads are still suspended.

WebSphere Education IBM

IBM JVM tuning parameters

Introduction to **-Xmaxf** and **-Xminf**

- The **-Xmaxf** and **-Xminf** settings control the 40% – 70% occupancy bounds
  - **-Xmaxf**: Maximum heap space free before shrinkage (default is 60%)
  - **-Xminf**: Minimum heap space before expansion (default is 70%)
- Can be used to move optimum occupancy window if required; for example, lower heap utilization is required for more infrequent GC cycles
- Can be used to prevent heap shrinkage
  - **-Xmaxf1.0** means shrinkage only when heap is 100% free

Introduction to **-Xmaxe** and **-Xmine**

- The **-Xmaxe** and **-Xmine** settings control the bounds of the size of each expansion step
  - **-Xmaxe**: Maximum amount of memory to add to the heap size in the case of expansion (default is unlimited)
  - **-Xmine**: Minimum amount of memory to add to the heap size in the case of expansion (default is 1 MB)
- Can be used to reduce (or prevent) compaction due to expansion
  - Reduce expansions by setting a large **-Xmine**

Figure 13-39. IBM JVM tuning parameters WB868 / ZB8681.0

Notes:

GC managed heap sizing

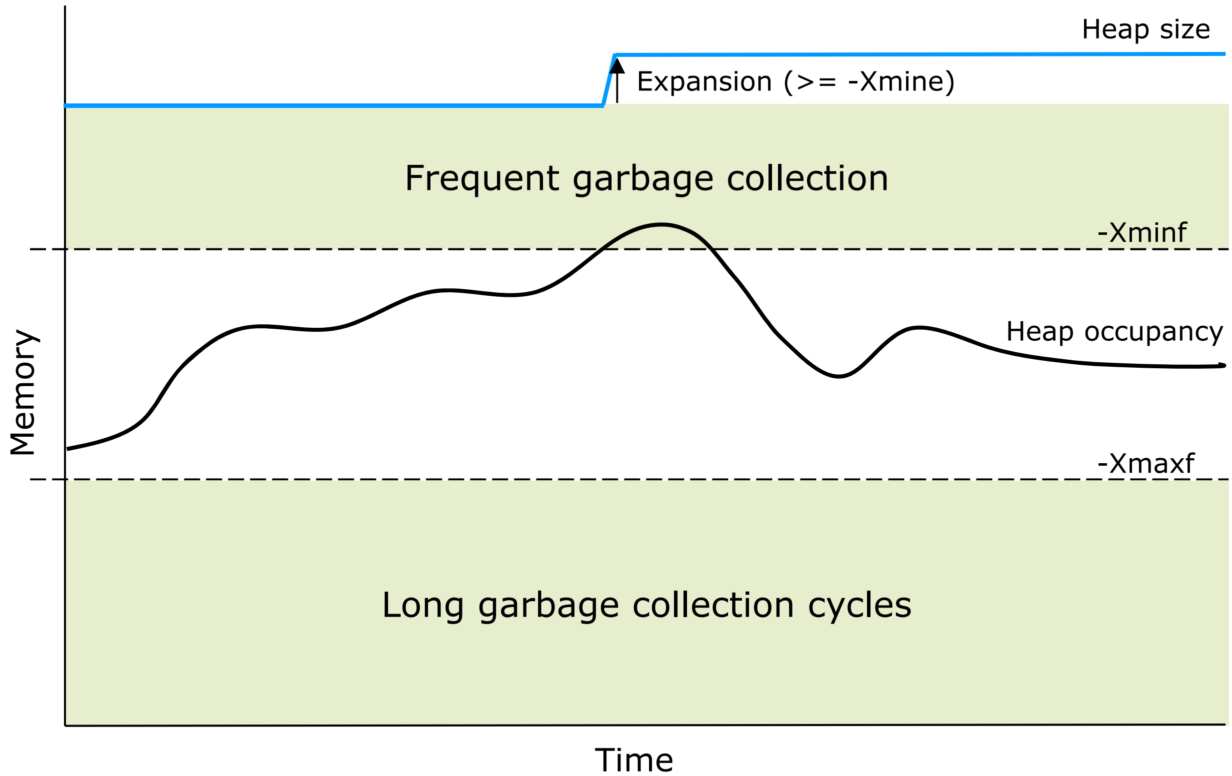


Figure 13-40. GC managed heap sizing WB868 / ZB8681.0

Notes:

How to tune a generational concurrent GC setup

- Tenured space must be large enough to hold all persistent data of the application
  - Too small causes excessive GC or even out-of-memory conditions
  - For a typical WebSphere Application Server application: 100 – 400 MB
- One way to determine the tenure space size:
  - The amount of free heap that exists after each GC in default mode
  - (% free heap) x (total heap size)
- Analyze GC logs
  - Understand how frequently the tenured space gets collected
  - An optimal generational application has infrequent collection in the tenured space

Figure 13-41. How to tune a generational concurrent GC setup WB868 / ZB8681.0

Notes:

Size the tenure space:

Sizing the tenure space is analogous to sizing overall heap size in a mark, sweep, compact collector.

- For a fixed size, use `-Xmo` with `xx` megabytes.
- For a dynamic sized tenure space within set boundaries, use `-Xmos` and `-Xmox` to set the minimum and maximum tenure sizes.

The fixed size options are not compatible with the minimum and maximum settings. If you specify both, the JVM does not start and it throws an error message.

There is no specific control on the flip count. The scavenger adjusts the flip count as needed.

There is no specific control on the tilt ratio. The scavenger adjusts the tilt ratio as needed.

WebSphere Education IBM

How to tune a generational concurrent GC nursery generation space

- Large nursery:
  - Large nursery is good for throughput
  - If it is too large, it delays the scavenge operations
- Small nursery:
  - Small nursery is good for low pause times
  - If it is too small, it causes frequent scavenge operations over a short span of time
- Good WebSphere performance (throughput) requires a large nursery
  - A good starting point is 512 MB
  - Move up or down to determine optimal value
  - Measure throughput and response times
- Analyze GC logs to understand frequency and length of scavenges

Figure 13-42. How to tune a generational concurrent GC nursery generation space WB868 / ZB8681.0

Notes:

If the nursery is too small, there are many scavenge operations over short spans of time. The result is high processor usage. Because there is a maximum flip count of 14, many objects reach the tenure age quickly. Many of these objects are short-lived and are tenured incorrectly. This occurrence fills and fragments the old generation, and forces a full garbage collect in tenure space (mark, sweep, and the nearly always deadly compaction).

If the nursery is too large, it delays scavenge operations. Most objects are no longer live and can be reclaimed. Many long-lived objects need to be copied to the survivor space. Since scavenge runs infrequently, it takes too long for these objects to reach the flip count and get tenured.

WebSphere Education

Java heap sizing tips

For some applications, the default settings might not give the best results

- **Symptom**: The frequency of garbage collection is too high until the heap reaches a steady state
  - **Tip:** Use `verbose:gc` to determine the size of the heap at a steady state and set `-Xms` to this value
- **Symptom**: The heap is fully expanded and the occupancy level is greater than 70%
  - **Tip:** Increase the `-Xmx` value so that the heap is not more than 70% occupied; but for best result performance, try to ensure that the heap never pages
  - The maximum heap size should be able to be contained in physical memory to avoid paging
- **Symptom**: At 70% occupancy, the frequency of garbage collections is too great
  - **Tip**: Change the setting of `-Xminf`
  - The default is 0.3, which tries to maintain 30% free space by expanding the heap
- **Symptom**: Pause times are too long
  - **Tip**: Try using: `-Xgcpolicy:optavgpause`
  - As a result, the pause times are reduced and more consistent when the heap occupancy rises
  - It does, however, reduce throughput by approximately 5%, although this value varies with different applications

Figure 13-43. Java heap sizing tips WB868 / ZB8681.0

Notes:

13.4.JVM tools

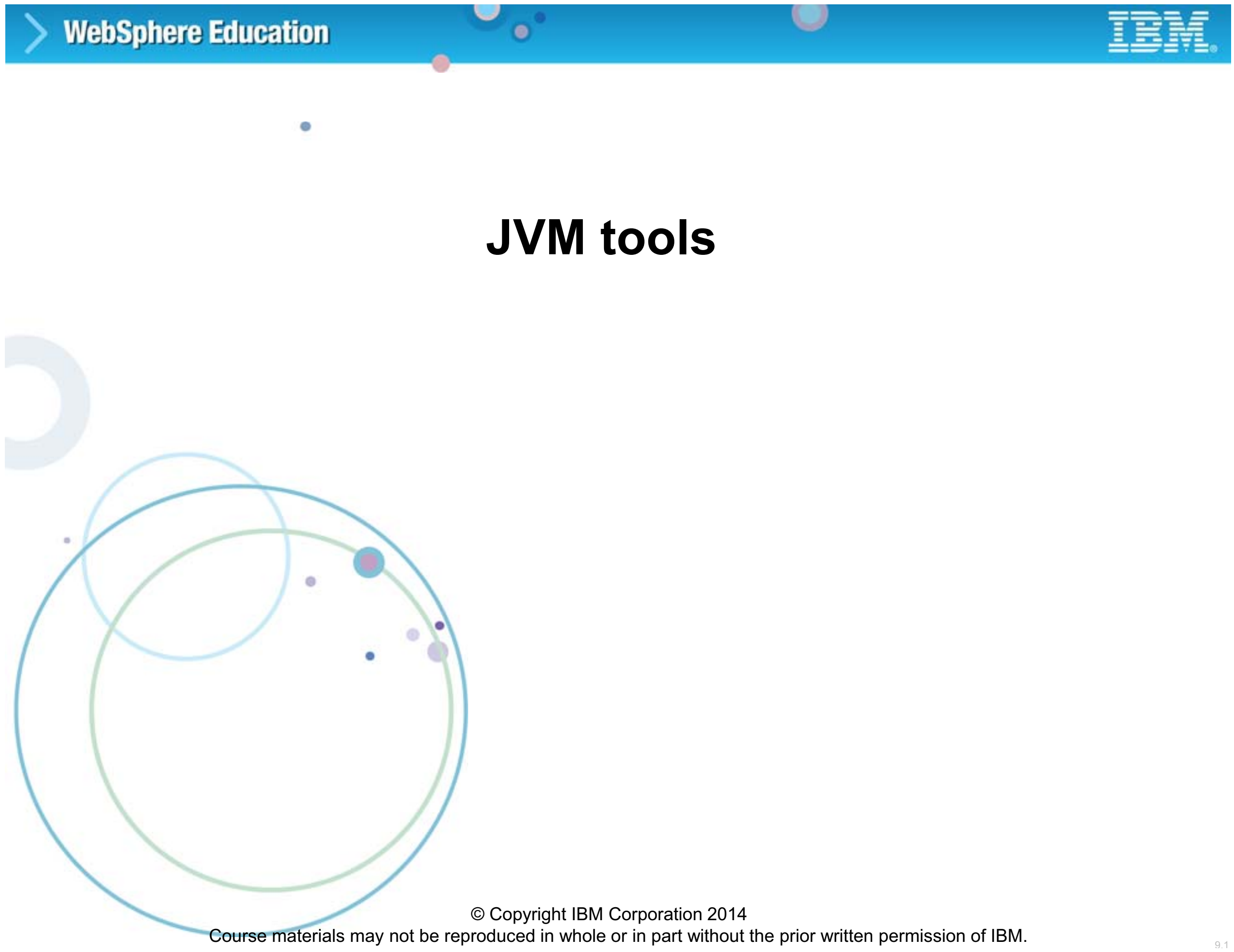


Figure 13-44. JVM tools WB868 / ZB8681.0

Notes:

WebSphere Education IBM

JVM tool overview

- Assess the status of a running Java application
  - IBM Monitoring and Diagnostic Tools for Java – Health Center
  - The Health Center can also be used to monitor processor usage and lock contention
- WebSphere internal thread pools and heap usage statistics
  - Tivoli Performance Viewer
- Thread activity snapshot: Use javacore files
  - IBM Thread and Monitor Dump Analyzer for Java
- Memory and garbage collection: Use verbose GC output
  - IBM Monitoring and Diagnostic Tools for Java – Garbage Collection and Memory Visualizer (GCMV)
  - IBM Pattern Modeling and Analysis tool for Java Garbage Collector
- Memory use by object: Use heap dumps
  - Memory Analyzer Tool (MAT)
  - IBM HeapAnalyzer

Figure 13-45. JVM tool overview WB868 / ZB8681.0

Notes:

Most tuning and debugging can be accomplished with free tools that many customers have on hand or ones available using IBM Support Assistant. In addition, high-end tools can also be used, such as IBM Tivoli Composite Application Manager for WebSphere or other third-party products.

There are tools available to parse system core files.

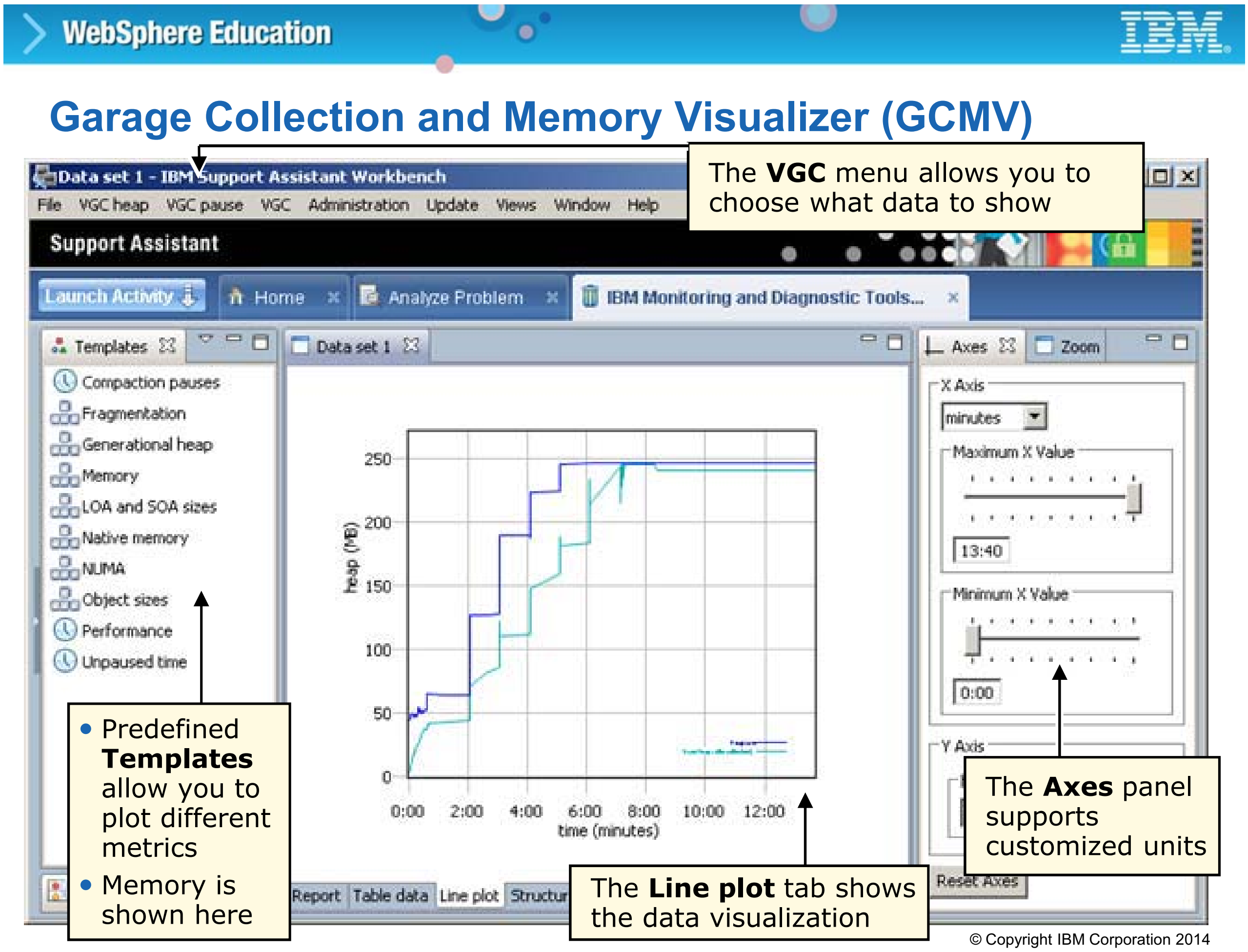


Figure 13-46. Garage Collection and Memory Visualizer (GCMV) WB868 / ZB8681.0

Notes:

Templates

The IBM Monitoring and Diagnostic Tools for Java - Garbage Collection and Memory Visualizer (GCMV) provides the following templates, which cover a range of common collections of data types to plot:

- **Memory**: Plots Heap size, Used heap (after collection), and JVM restarts. It is useful for diagnosing memory leaks or poor choice of heap size.
- **Performance**: Plots Mark, Sweep, Pause, Compact times, and JVM restarts. It is useful for investigating performance issues.
- **Unpaused Time**: Plots Intervals between garbage collection triggers and Pause times. This template can help determine whether an application is spending long periods of time in garbage collection rather than running.
- **Object Sizes**: Plots Heap size, Used heap (after collection), and Requested object sizes that trigger allocation failures. This template can help determine whether an application is requesting large objects and triggering excessive garbage collection.

- **LOA and SOA Sizes**: Plots Heap size, Used SOA (small object area), Used LOA (large object area), Used tenured heap, and Used nursery heap (after collection). This template can help determine whether there is a problem with the `-Xloratio` setting.
- **Generational Heap**: Plots Heap size, Free heap, Tenured heap size, Free tenured heap, Nursery size, Free nursery heap, and Tenure age. It is useful for diagnosing memory leaks or poor nursery or tenured heap sizings.
- **Fragmentation**: Plots Heap size, Free heap (before collection), and Free heap (after collection). A large amount of free heap before collection might indicate heap fragmentation. A small amount of free heap after collection might indicate that the heap size is too small or the application is holding onto too much data.
- **Compaction Pauses**: Plots Heap size, Used heap (after collection), Compact times, and Pause times. This template can help identify whether a restricted heap is causing excessive time that is spent in compaction.
- **Native Memory**: Plots native memory usage. It is useful for diagnosing out-of-memory errors.
- **NUMA**: Non-uniform memory architecture information.

WebSphere Education

Pattern Modeling and Analysis Tool (PMAT) (1 of 2)

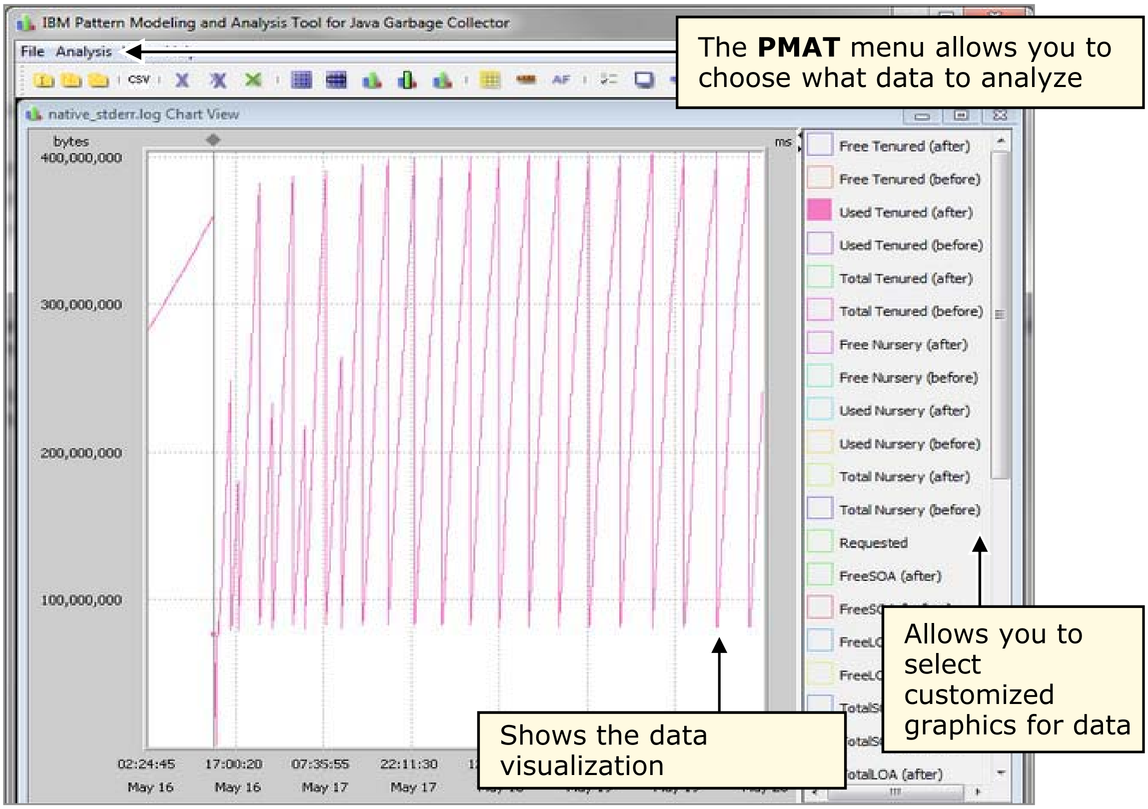


Figure 13-47. Pattern Modeling and Analysis Tool (PMAT) (1 of 2) WB868 / ZB8681.0

Notes:

Pattern Modeling and Analysis Tool for IBM Java Garbage Collector (PMAT) parses verbose GC trace, analyzes Java heap usage, and suggests key configurations that are based on pattern modeling of Java heap usage.

When the JVM (Java virtual machine) cannot allocate an object from the current heap because of lack of space, a memory allocation fault occurs, and the garbage collector is invoked. The first task of the garbage collector is to collect all the garbage that is in the heap. This process starts when any thread calls the garbage collector either indirectly as a result of allocation failure or directly by a specific call to `System.gc()`. The first step is to get all the locks needed by the garbage collection process. This step ensures that other threads are not suspended while they are holding critical locks. All other threads are then suspended. Garbage collection can then begin. It occurs in three phases: mark, sweep, and compaction (optional).

WebSphere Education

Pattern Modeling and Analysis Tool (PMAT) (2 of 2)

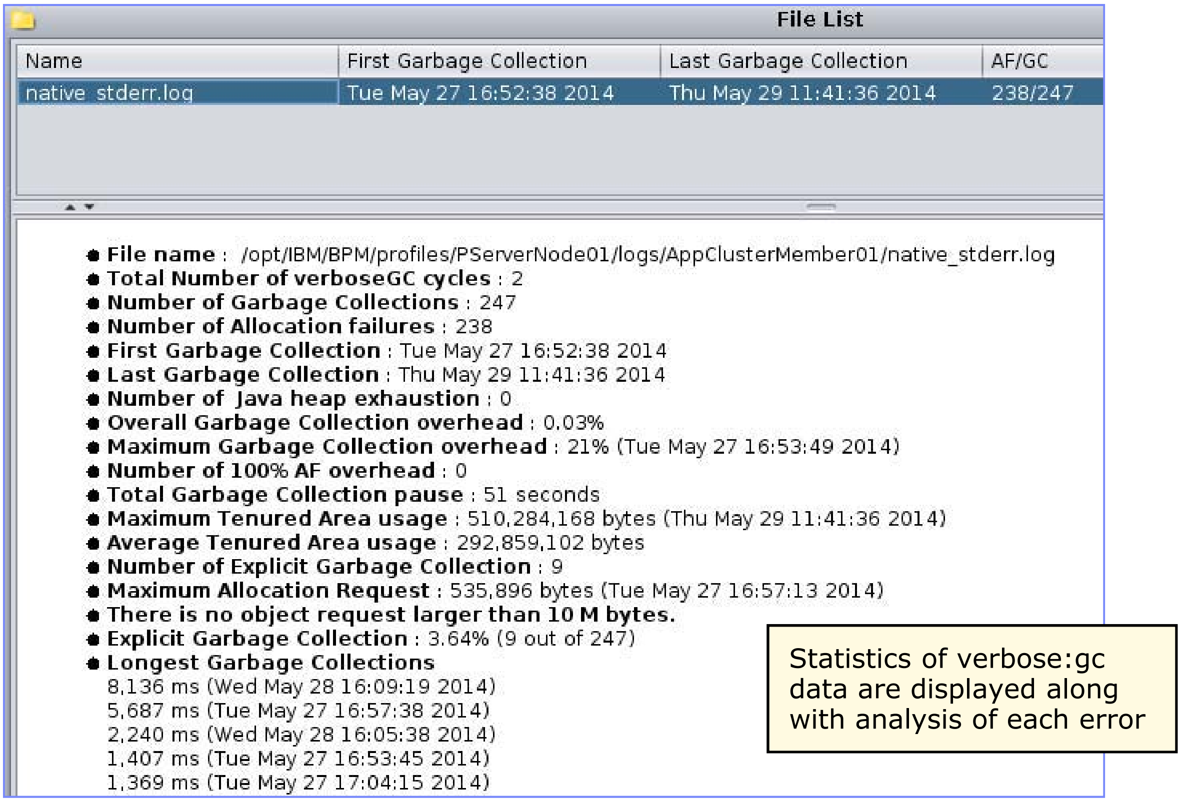


Figure 13-48. Pattern Modeling and Analysis Tool (PMAT) (2 of 2) WB868 / ZB8681.0

Notes:

PMAT analyzes verbose GC traces by parsing the traces and building pattern models. PMAT suggests key configurations by executing a diagnosis engine and pattern modeling algorithm. If any errors are related to Java heap exhaustion or fragmentation in the verbose GC trace, PMAT can diagnose the root cause of failures. PMAT provides rich chart features that graphically display Java heap usage.

The following features are included:

- GC analysis
- GC table view
- Allocation failure summary
- GC usage summary
- GC duration summary
- GC graph view
- GC pattern analysis

- Zoom in, out, selection, center of chart view
- Option of changing chart color

WebSphere Education IBM

Unit summary

Having completed this unit, you should be able to:

- Explain the basic components of garbage collection (GC): mark, sweep, and compact
- Explain the concept of Java memory management
- Describe some of the myths that surround Java memory management
- Collect verbose GC trace logs from a server at run time
- Analyze verbose GC trace logs to diagnose potential Java memory management issues

Figure 13-49. Unit summary WB868 / ZB8681.0

Notes:

WebSphere Education

Checkpoint questions

1. Which statement is true about the JVM with generational concurrent garbage collection policy?
   a) JVM has two generations: young generation and tenured generation.
   b) JVM has flat heap collector that is focused on maximum throughput.
   c) JVM provides maximum throughput with minimal pause times.
   d) Both a and c

2. True or false: When you are tuning the JVM heap, monitor the GC activities and find some characteristics; then, choose the policy that would best suit the type of applications that you have. If GC operation cost is larger than 10%, you most likely chose the wrong policy.

3. True of false: Listeners, hash tables, static class data, JNI references, and objects with finalizers are common causes of memory leaks in Java.

Figure 13-50. Checkpoint questions WB868 / ZB8681.0

Notes:

Write your answers here:

1.

2.

3.

WebSphere Education

Checkpoint answers

1. d
2. True
3. True

Figure 13-51. Checkpoint answers

WB868 / ZB8681.0

Notes:

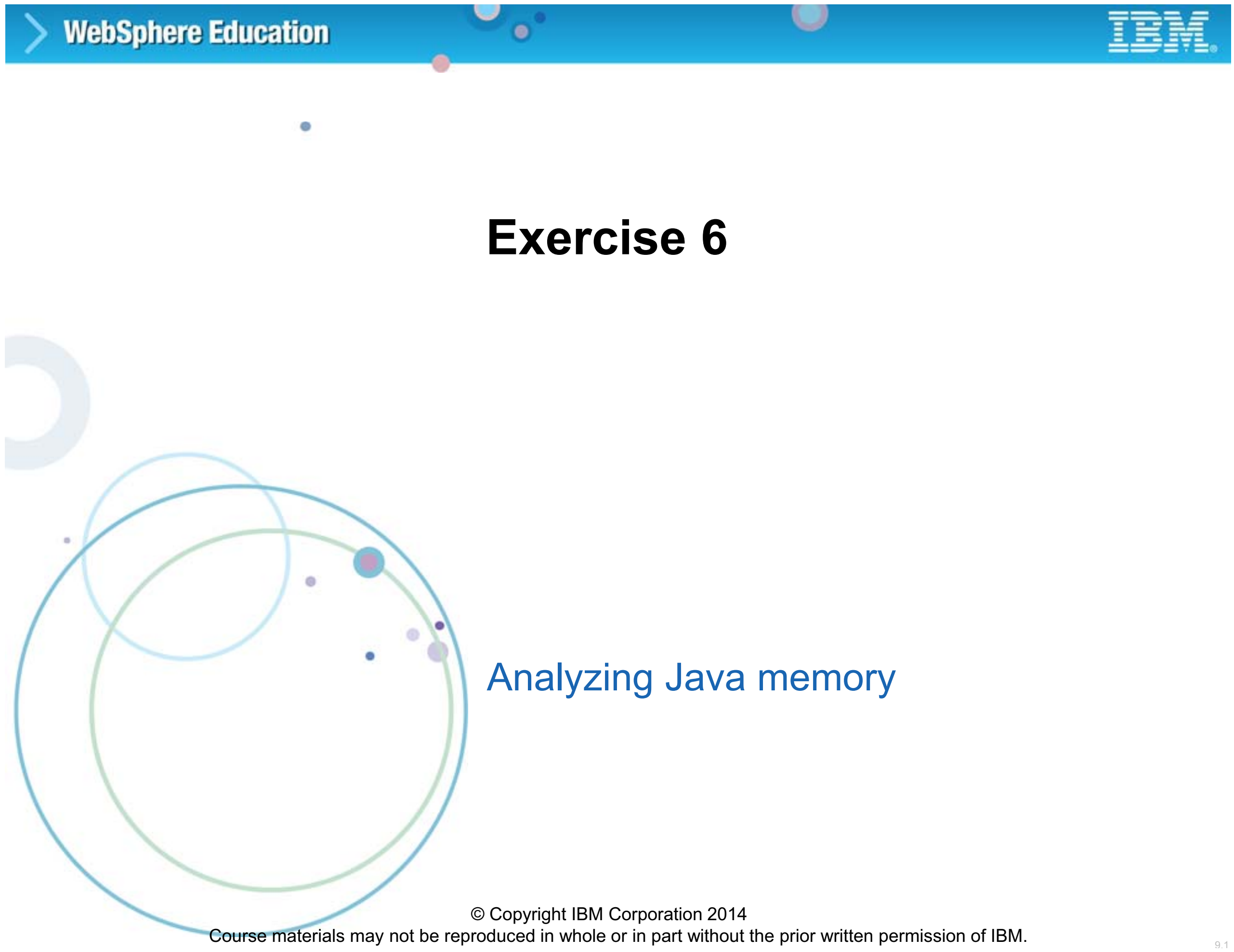


Figure 13-52. Exercise 6 WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Exercise objectives

After completing this exercise, you should be able to:

- Enable verbose GC and modify JVM heap settings
- Analyze Java heap statistics by collecting verbose:gc output for large live sites, large objects, out-of memory-cases, and slow targets
- Use the IBM Monitoring and Diagnostic Tools for Java – Garbage Collection and Memory Visualizer (GCMV) tool from the IBM Support Assistant

Figure 13-53. Exercise objectives WB868 / ZB8681.0

Notes:

Unit 14. Course summary

What this unit is about

This unit summarizes the course and provides information for future study.

What you should be able to do

After completing this unit, you should be able to:

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

Figure 14-1. Unit objectives WB868 / ZB8681.0

Notes:

WebSphere Education IBM

Course learning objectives (1 of 2)

After completing this course, you should be able to:

- Explain the architecture and components for a typical IBM Business Process Manager deployment
- Explain basic performance concepts and methodologies
- Apply the Business Process Manager performance checklist and configure the server for better performance
- Identify key development best practices for IBM Process Designer and IBM Integration Designer
- Implement best practices for general WebSphere runtime performance
- Tune the target modules for various bindings
- Purge data that is no longer needed from the Business Process Manager environment
- Create efficient Coaches and Coach Views
- Name the key best practices for the Business Flow Manager (BFM)

Figure 14-2. Course learning objectives (1 of 2) WB868 / ZB8681.0

Notes:

WebSphere Education

Course learning objectives (2 of 2)

After completing this course, you should be able to:

- Identify key WebSphere monitoring facilities
- Evaluate various bottleneck patterns and determine a possible solution that is based on your observations
- Collect verbose GC trace logs from the runtime and analyze them to diagnose potential Java memory management issues
- Monitor application server performance by using WebSphere and the IBM Support Assistant
- Monitor and tune the JVM for optimum throughput and response time
- Monitor and tune connection pools for optimum performance
- Use the IBM Health Center tool to profile and tune Java EE applications
- Troubleshoot Business Process Manager performance problems

Figure 14-3. Course learning objectives (2 of 2) WB868 / ZB8681.0

Notes:

WebSphere Education IBM

To learn more on the subject

- IBM Training website:
  - www.ibm.com/training
- The online version of the IBM Business Process Manager Advanced V8.5 Knowledge Center is a good source of product information
  - http://publib.boulder.ibm.com/infocenter/dmndhelp/v8r5m0/index.jsp
- Business Process Management (BPM) enabled by SOA:
  - http://www.ibm.com/software/info/bpm/
- IBM Business Process Manager Advanced V8.5 home page:
  - http://www.ibm.com/software/integration/business-process-manager/advanced/
- Numerous articles that are listed in IBM developerWorks
  - http://www.ibm.com/developerworks/

Figure 14-4. To learn more on the subject WB868 / ZB8681.0

Notes:

WebSphere Education

References

- IBM Redbooks
  - *IBM Business Process Manager V8.0 Performance Tuning* and *Best Practices* (REDP-4935-00)
  - *Business Process Management Guide Using IBM Business Process Manager V8.5* (SG24-8175-00)
  - *Leveraging the IBM BPM Coach Framework in Your Organization* (SG24-8210-00)
  - *IBM Business Process Manager Security: Concepts and Guidance* (SG24-8027-00)
  - *WebSphere Application Server V8.5 Administration and Configuration Guide for the Full Profile (SG24-8056-01)*
  - *WebSphere Application Server Network Deployment V6: High Availability Solutions* (SG24-6688-00)
  - *Techniques for Managing Large WebSphere Installations* (SG24-7536-00)

Figure 14-5. References WB868 / ZB8681.0

Notes:

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

Figure 14-6. Unit summary

WB868 / ZB8681.0

Notes:

Appendix A. List of abbreviations

ADDM	Automatic Database Diagnostic Monitor
AIS	Advanced Integration service
AIX	Advanced IBM UNIX
AMI	Application Messaging Interface
APAR	authorized program analysis report
API	application programming interface
ARM	Application Request Measurement
AST	Application Server Toolkit
AWR	automatic workload repository
BFM	Business Flow Manager
BLA	business-level application
BO	business object
BPC	Business Process Choreographer
BPCIVT	Business Process Choreographer Installation Verification Test
BPD	business process definition
BPD	business process diagram
BPE	business process engine
BPEL	Business Process Execution Language
BPM	business process management
BPMN	Business Process Model and Notation
BRM	Business Rules Manager
BR	business rules
BSM	business state machine
BTPS	***IEG, p. 1-24***
CA	cache access
CEI	Common Event Infrastructure
CGBI	core group bridge interface
CGBS	core group bridge service
CICS	Customer Information Control System
CIFS	Common Internet File System
COBOL	Common Business Oriented Language

CPU	central processing unit
CSS	Cascading Style Sheets
CVS	Concurrent Versions System
CW	conditioned wait
DBA	database administrator
DCS	Distribution and Consistency Services
DDL	Data Definition Language
DDT	database design tool
DE	***PPT #1-40***
DNS	Domain Name System
DOM	Document Object Model
DRS	data replication service
EAI	Enterprise Application Infrastructure
EAR	enterprise archive
EE	Enterprise Edition
EIS	enterprise information system
EJB	Enterprise JavaBeans
ESB	enterprise service bus
FFDC	first-failure data capture
FQDN	fully qualified domain name
FTP	File Transfer Protocol
GC	garbage collector
GCMV	Garbage Collection and Memory Visualizer
GUI	graphical user interface
HA	high availability
HAM	high availability manager
HACMP	High-Availability Cluster Multi-Processing
HP	Hewlett Packard
HP-UX	Hewlett Packard UNIX
HTM	Human Task Manager
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDDE	Interactive Diagnostic Data Explorer

IIOP	Internet Inter-ORB Protocol
ILT	instructor-led training
ILO	instructor-led online
IMS	Information Management System
I/O	input/output
IP	Internet Protocol
ISMP	Install Shield MultiPlatform
IT	information technology
IVT	installation verification test
J2C	J2EE Connector architecture
J2EE	Java 2 Platform, Enterprise Edition
JAAS	Java Authentication and Authorization Service
JAR	Java archive
JCA	Java Connector Architecture
JDBC	Java Database Connectivity
JDK	Java Development Kit
JEE	Java Enterprise Edition
JIT	just-in-time
JMS	Java Messaging Service
JMX	Java Management Extensions
JNDI	Java Naming and Directory Interface
JNI	Java Native Interface
JNLP	***IEG, p. 6-30***
JSF	JavaServer Faces
JSON	JavaScript Object Notation
JSP	JavaServer Pages
JVM	Java virtual machine
JVMTI	***IG, p. 4-42***
KB	kilobyte
KPI	key performance indicator
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LOA	large object area

LOB	line of business
LRU	least recently used
LSD	location service daemon
LTPA	Lightweight Third Party Authentication
MAP	message addressing property
MAP	Mortgage Application Process
MAT	Memory Analyzer Tool
MDB	message-driven bean
ME	messaging engine
MEDB	messaging engine database
MQ	Message Queue
MQI	Message Queue Interface
MSO	mark stack overflow
ND	Network Deployment
NFS	Network File System
NUMA	nonuniform memory access
OASIS	Organization for the Advancement of Structured Information Standards
ODM	***PPT #1-13***
OOM	out-of-memory
ORB	Object Request Broker
OS	operating system
PC	Process Center
PD	problem determination
PDW	Process Data Warehouse
PHD	Portable Heap Dump
PID	process identifier
PMAT	Pattern Modeling and Analysis Tool
PMI	Performance Monitoring Infrastructure
PMR	problem management report
PO	***IG, p. 5-70***
POC	proof of concept
POJO	plain old Java object
PuTTY	***IG, p. 10-40***

QCF	queue connection factory
QoS	quality of service
RA	resource adapter
RAID	Redundant Array of Independent Disks
RAM	random access memory
RAR	resource adapter archive
RDBMS	relational database management system
REST	Representational State Transfer
RMI	Remote Method Invocation
RMI/IIOP	Remote Method Invocation over Internet InterORB Protocol
RMM	Reliable Multicast Messaging
ROI	return on investment
SAN	storage area network
SCA	Service Component Architecture
SCDL	Service Component Definition Language
SDK	software development kit
SDO	Service Data Object
SEL	selector
SIB	service integration bus
SIBus	service integration bus
SLA	service level agreement
SMO	service message object
SMP	***IG, p. 13-28***
SOA	service-oriented architecture
SOA	small object area
SQL	Structured Query Language
SSL	Secure Sockets Layer
STW	stop-the-world
SWAM	Simple WebSphere Authentication Mechanism
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TLH	thread local heap
TM	transaction manager

TMDA	Thread and Monitor Dump Analyzer
UCA	undercover agent
UCA	unrefreshed cache access
UCP	unrefreshed cache percentage
UI	user interface
UML	Unified Modeling Language
UNIX	Uniplexed Information and Computing System
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTE	unit test environment
UUID	Universally Unique Identifier
VGC	***PPT #13-48***
VM	virtual machine
VMM	virtual member manager
VNC	***IG, p. 10-40***
WAN	wide area network
WAR	web archive
WLM	workload management
WS-BPEL	Web Services Description Language
WSDL	Web Services Description Language
WYSIWYG	what you see is what you get
XCT	cross-component trace
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformation
z/OS	z Series Operating System
zWSAS	z Series Operating System

Appendix B. Resource guide

Completing this WebSphere Education course is a great first step in building your WebSphere, CICS, and SOA skills. Beyond this course, IBM offers several resources to keep your WebSphere skills on the cutting edge. Resources available to you range from product documentation to support websites and social media websites.

Training

- **IBM Training website**
  - Bookmark the IBM Training website for easy access to the full listing of IBM training curricula. The website also features training paths to help you select your next course and available certifications.
  - For more information, see: `http://www.ibm.com/training`
- **IBM Training News**
  - Review or subscribe to updates from IBM and its training partners.
  - For more information, see: `http://bit.ly/IBMTrainEN`
- **IBM Certification**
  - You can demonstrate to your employer or clients your new WebSphere, CICS, or SOA mastery through achieving IBM Professional Certification. WebSphere certifications are available for developers, administrators, and business analysts.
  - For more information, see: `http://www.ibm.com/certify`
- **Training paths**
  - Find your next course easily with IBM training paths. Training paths provide a visual flow-chart style representation of training for many WebSphere products and roles, including developers and administrators.
  - For more information, see: `http://www.ibm.com/services/learning/ites.wss /us/en?pageType=page&c=a0003096`

Social media links

You can keep in sync with WebSphere Education, including new courses and certifications, course previews, and special offers, by going to any of the following social media websites.

- **Twitter**
  - Receive short and concise updates from WebSphere Education a few times each week.
  - Follow WebSphere Education at: `twitter.com/websphere_edu`
- **Facebook**:
  - Become a fan of IBM Training on Facebook to keep in sync with the most recent news and career trends, and to post questions or comments.

  - Find IBM Training at: `facebook.com/ibmtraining`
- **YouTube**:
  - Go to the IBM Training YouTube channel to learn about IBM training programs and courses.
  - Find IBM Training at: `youtube.com/IBMTraining`

Support

- **WebSphere Support portal**
  - The WebSphere Support website provides access to a portfolio of support tools. From the WebSphere Support website, you can access several downloads, including troubleshooting utilities, product updates, drivers, and Authorized Program Analysis Reports (APARs). To collaboratively solve issues, the support website is a clearing house of links to online WebSphere communities and forums. The IBM support website is now customizable so you can add and delete portlets to the information most important to the WebSphere products you work with.
  - For more information, see: `http://www.ibm.com/software/websphere/support`
- **IBM Support Assistant**
  - The IBM Support Assistant is a local serviceability workbench that makes it easier and faster for you to resolve software product issues. It includes a desktop search component that searches multiple IBM and non-IBM locations concurrently and returns the results in a single window, all within IBM Support Assistant.
  - IBM Support Assistant includes a built-in capability to submit service requests; it automatically collects key problem information and transmits it directly to your IBM support representative.
  - For more information, see: `http://www.ibm.com/software/support/isa`
- **WebSphere Education Assistant**
  - IBM Education Assistant is a collection of multimedia modules that are designed to help you gain a basic understanding of IBM software products and use them more effectively. The presentations, demonstrations, and tutorials that are part of the IBM Education Assistant are an ideal refresher for what you learned in your WebSphere Education course.
  - For more information, see: `http://www.ibm.com/software/info/education/assistant/`

WebSphere documentation and tips

- **IBM Redbooks**
  - The IBM International Technical Support Organization develops and publishes IBM Redbooks publications. IBM Redbooks are downloadable PDF files that describe installation and implementation experiences, typical solution scenarios, and step-by-step “how-to” guidelines for many WebSphere products. Often, Redbooks

include sample code and other support materials available as downloads from the site.

- For more information, see: `http://www.ibm.com/redbooks`

- **IBM documentation and libraries**
  - Information centers and product libraries provide an online interface for finding technical information on a particular product, offering, or product solution. The information centers and libraries include various types of documentation, including white papers, podcasts, webcasts, release notes, evaluation guides, and other resources to help you plan, install, configure, use, tune, monitor, troubleshoot, and maintain WebSphere products. The WebSphere information center and library are located conveniently in the left navigation on WebSphere product web pages.
- **developerWorks**
  - IBM developerWorks is the web-based professional network and technical resource for millions of developers, IT professionals, and students worldwide. IBM developerWorks provides an extensive, easy-to-search technical library to help you get up to speed on the most critical technologies that affect your profession. Among its many resources, developerWorks includes how-to articles, tutorials, skill kits, trial code, demonstrations, and podcasts. In addition to the WebSphere zone, developerWorks also includes content areas for Java, SOA, web services, and XML.
  - For more information, see: `http://www.ibm.com/developerworks`

WebSphere Services

- IBM Software Services for WebSphere are a team of highly skilled consultants with broad architectural knowledge, deep technical skills, expertise on suggested practices, and close ties with IBM research and development labs. The WebSphere Services team offers skills transfer, implementation, migration, architecture, and design services, plus customized workshops. Through a worldwide network of services specialists, IBM Software Service for WebSphere makes it easy for you to design, build, test, and deploy solutions, helping you to become an on-demand business.
- For more information, see:
  `http://www.ibm.com/developerworks/websphere/services/`