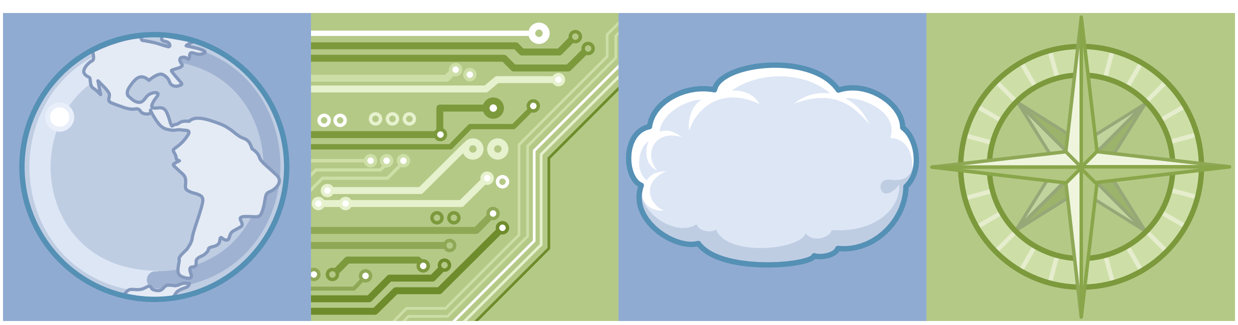
IBM Training

Student Notebook

IBM Business Process Manager V8.5 Problem Determination

Course code WB869 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®	alphaWorks®	Blueworks Live™
CICS®	DB™	DB2®
developerWorks®	Express®	IMS™
PartnerWorld®	Performance Inspector™	Rational Team Concert™
Rational®	Redbooks®	RequisitePro®
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.

Worklight® is a trademark or registered trademark of Worklight, an IBM Company.

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®	alphaWorks®	Blueworks Live™
CICS®	DB™	DB2®
developerWorks®	Express®	IMS™
PartnerWorld®	Performance Inspector™	Rational Team Concert™
Rational®	Redbooks®	RequisitePro®
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.

Worklight® is a trademark or registered trademark of Worklight, an IBM Company.

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

Course description

IBM Business Process Manager V8.5 Problem Determination

Duration: 5 days

Purpose

In this 5-day course, you learn how to troubleshoot problems that might appear in an IBM Business Process Manager V8.5.0.1 environment. It teaches you problem isolation techniques that enable you to be self-sufficient and effective in discovering and resolving runtime problems. IBM consultants and IBM support teams assist their customers by using many of the methodologies that are introduced in this course.

The course begins with an introduction to the tools and techniques that are used for both WebSphere Application Server and IBM Business Process Manager problem determination. It covers problem determination topics such as how to gather and analyze trace data and how to troubleshoot Business Process Manager runtime and user interface (UI) issues. You also learn how to troubleshoot problems with components such as WebSphere Adapters and the Business Process Choreographer.

Hands-on lab exercises throughout the course reinforce the concepts that you learn in the lectures, and enable you to practice your problem determination skills. The exercises cover skills such as collecting data with the IBM Support Assistant, troubleshooting Advanced Integration services, and troubleshooting problems with long-running business processes and Service Component Architecture (SCA) applications.

Audience

This course is designed for system administrators.

Prerequisites

Before taking this course, you should successfully complete course WB867 or ZB867, *Administration of IBM Business Process Manager Advanced V8.5*, or have equivalent experience.

Since WebSphere Application Server supports the core Java Platform, Enterprise Edition services, WebSphere Process Server administrators must know some WebSphere Application Server problem determination techniques. Therefore, this course contains some topics that are also covered in course WA591/VA591/ZA591, *WebSphere Application Server V8.5.5 Problem Determination*.

Objectives

After completing this course, you should be able to:

- Describe key problem determination steps
- Describe error prevention strategies
- Identify and collect MustGather data for IBM Business Process Manager
- Use the server logs viewer tool to read logging and tracing files
- Query failed events through failed event manager and resubmit the events after the problem is resolved
- Describe the exception types from the SCA programming model: service business exceptions and service runtime exceptions
- Examine the trace log to follow the navigation steps inside the business process engine
- Isolate and troubleshoot WebSphere Adapter-related runtime problems
- Explain how to troubleshoot Advanced Integration services (AIS)
- Use Business Process Choreographer Explorer to examine problems with running business process instances

Agenda

Day 1

Course introduction
Unit 1: IBM Business Process Manager troubleshooting methodology
Unit 2: WebSphere Application Server and IBM Business Process Manager systems and components
Unit 3: Gathering diagnostic data
Exercise 1: Gathering diagnostic data

Day 2

Unit 4: MustGather data and IBM Support Assistant
Exercise 2: Using IBM Support Assistant
Unit 5: WebSphere Application Server problem determination refresher
Unit 6: Configuration problems
Exercise 3: Troubleshooting an online Process Server

Day 3

Unit 7: Troubleshooting process application deployment
Exercise 4: Troubleshooting process application deployment
Unit 8: IBM Business Process Manager Standard runtime problems
Exercise 5: Troubleshooting Business Process Manager Standard runtime problems
Exercise 6: Troubleshooting user interface

Day 4

Unit 9: Troubleshooting SCA runtime problems
Exercise 7: Troubleshooting an SCA application with runtime errors
Unit 10: Business Process Choreographer problems
Exercise 8: Troubleshooting a long-running business process

Day 5

Unit 11: Troubleshooting the advanced deployment environment
Exercise 9: Troubleshooting Advanced Integration services
Unit 12: Business Process Manager performance problems
Unit 13: WebSphere Adapter problems
Exercise 10: Troubleshooting WebSphere Adapters
Unit 14: Problem prevention and best practices
Unit 15: Course summary

Unit 1. IBM Business Process Manager troubleshooting methodology

What this unit is about

This unit serves as an introductory unit for IBM Business Process Manager problem determination, and some of the topics are covered in more detail in later units. When you encounter a problem, you must answer several questions to separate runtime problems from design problems, or perhaps, environment configuration problems. In this unit, you learn the basic troubleshooting checklist for IBM Business Process Manager that can help you clearly identify the problem.

What you should be able to do

After completing this unit, you should be able to:

- Describe the key steps in problem determination
- Characterize a problem from its symptoms
- Implement a relief or recovery plan
- Describe the basic troubleshooting checklist for IBM Business Process Manager
- Describe a problem efficiently to IBM support

How you will check your progress

- Checkpoint

WebSphere Education IBM

Unit objectives

After completing this unit, you should be able to:

- Describe the key steps in problem determination
- Characterize a problem from its symptoms
- Implement a relief or recovery plan
- Describe the basic troubleshooting checklist for IBM Business Process Manager
- Describe a problem efficiently to IBM support

Figure 1-1. Unit objectives WB8691.0

Notes:

WebSphere Education

Topics

- Problem determination overview
- Problem prevention
- Organize the investigation
- Relief options
- Initial investigation
- In-depth investigation

Figure 1-2. Topics WB8691.0

Notes:

1.1. Problem determination overview

This topic is introductory. First, common types of problem symptoms are discussed. An ideal strategy for problem prevention is to monitor the system regularly. However, monitoring the system comes with a performance cost. Therefore, the benefit you can gain from monitoring the system should be weighed against the performance cost.

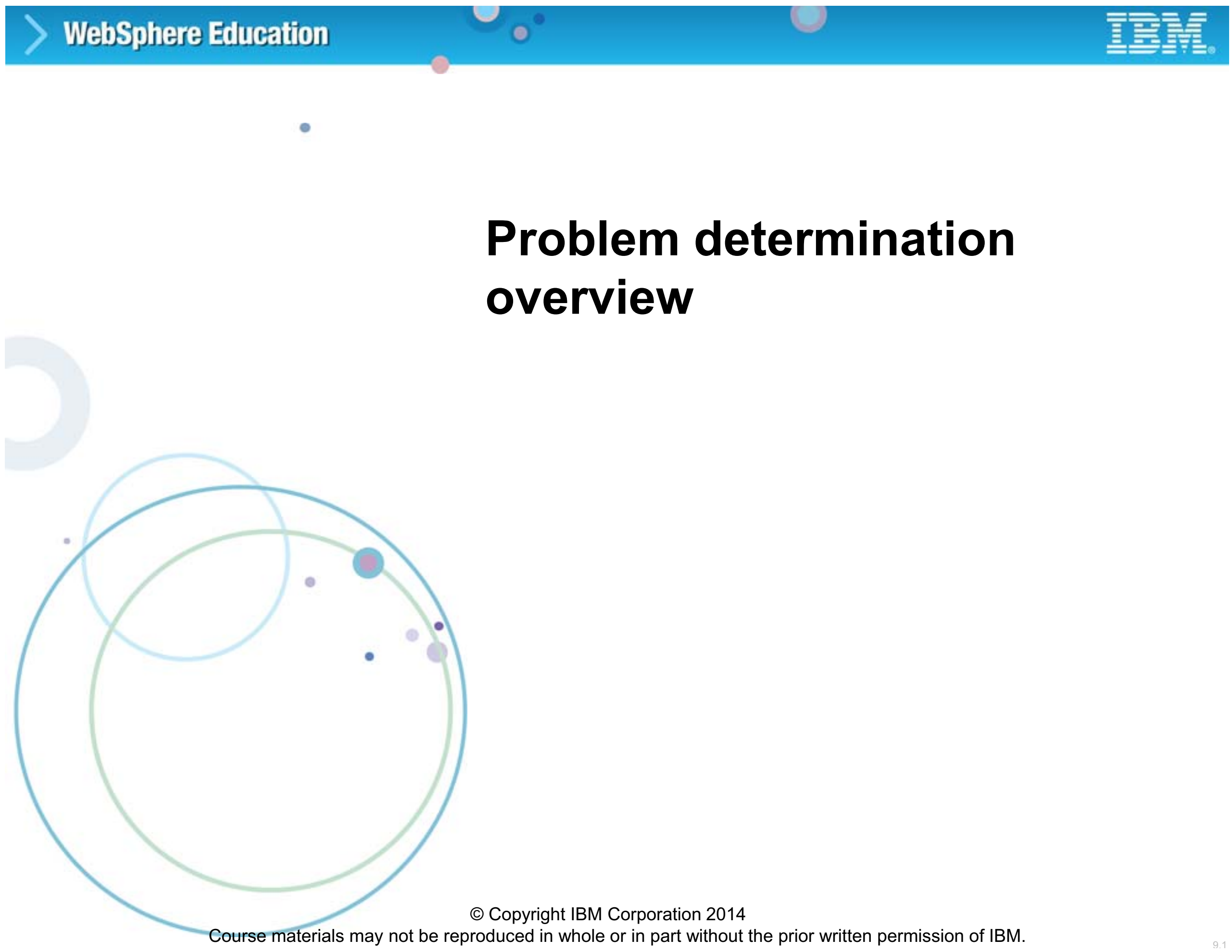


Figure 1-3. Problem determination overview WB8691.0

Notes:

General observations about problem determination

- Problem determination is ***not*** an exact science
- Problem determination is ***not*** incomprehensible
- Problem determination is often a cooperative and iterative process
- The biggest obstacle to problem determination is poor communication
- Not every problem requires the most complex problem determination skills and techniques

> Problem determination is not magic, but a matter of common sense, thoroughness, and clear communication

Figure 1-4. General observations about problem determination WB8691.0

Notes:

A major challenge of problem determination is dealing with unanticipated problems. It is much like detective work: finding clues, making educated guesses, verifying suspicions, and other considerations. The most important skills are common sense, focus, thoroughness, and rigorous thinking.

WebSphere Education

Problem determination goals

In any troubleshooting situation, you have three goals:

1. Quickly provide a temporary solution (relief)
   – This way, affected users can get back to work while you look for the permanent solution
2. Find and implement the correct permanent solution
3. Try to make sure that similar problems do not recur
   – If the issue does recur, make sure that you are prepared to deal with it based on what you learned

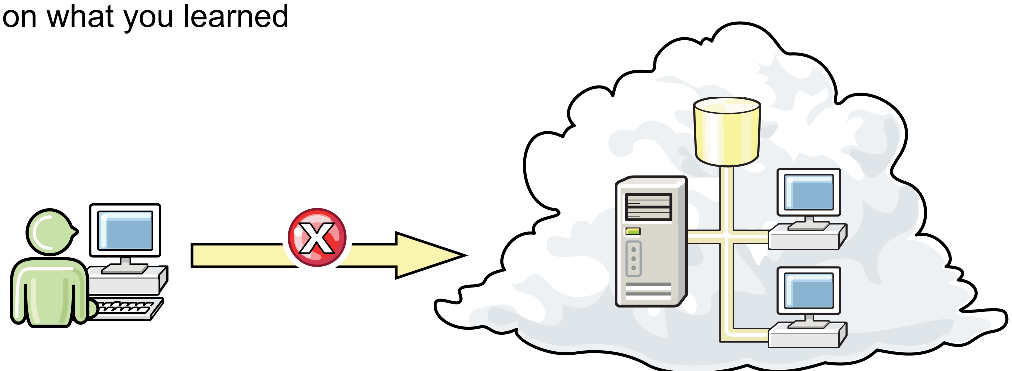


Figure 1-5. Problem determination goals WB8691.0

Notes:

Often, problems go undetected for a long time, or get detected indirectly through a secondary effect. You need tools and a plan to effectively detect problems or any potential anomalies when they emerge.

When a problem does occur, the three main goals of troubleshooting are:

- Provide relief to affected users
- Find a permanent resolution to the problem
- Try to make sure that the problem does not recur

WebSphere Education

Key steps for problem determination

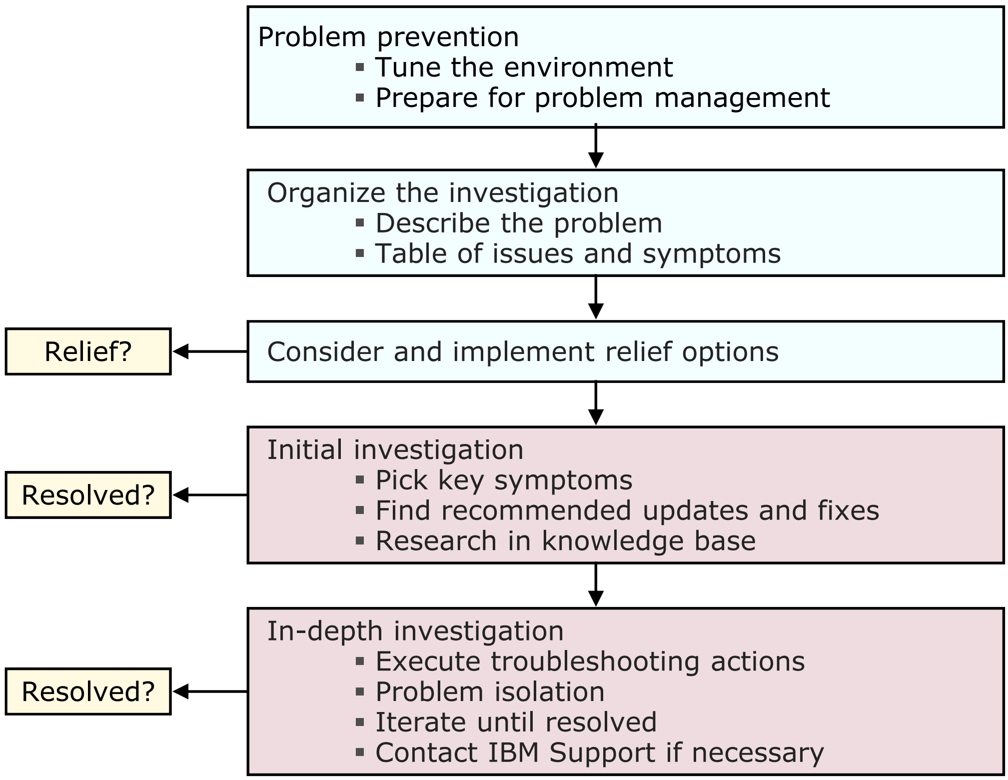


Figure 1-6. Key steps for problem determination WB8691.0

Notes:

The diagram on this slide shows the main steps in the problem determination process. Each of these steps is covered in more detail later in this unit.

1.2. Problem prevention

Problem prevention

Figure 1-7. Problem prevention WB8691.0

Notes:

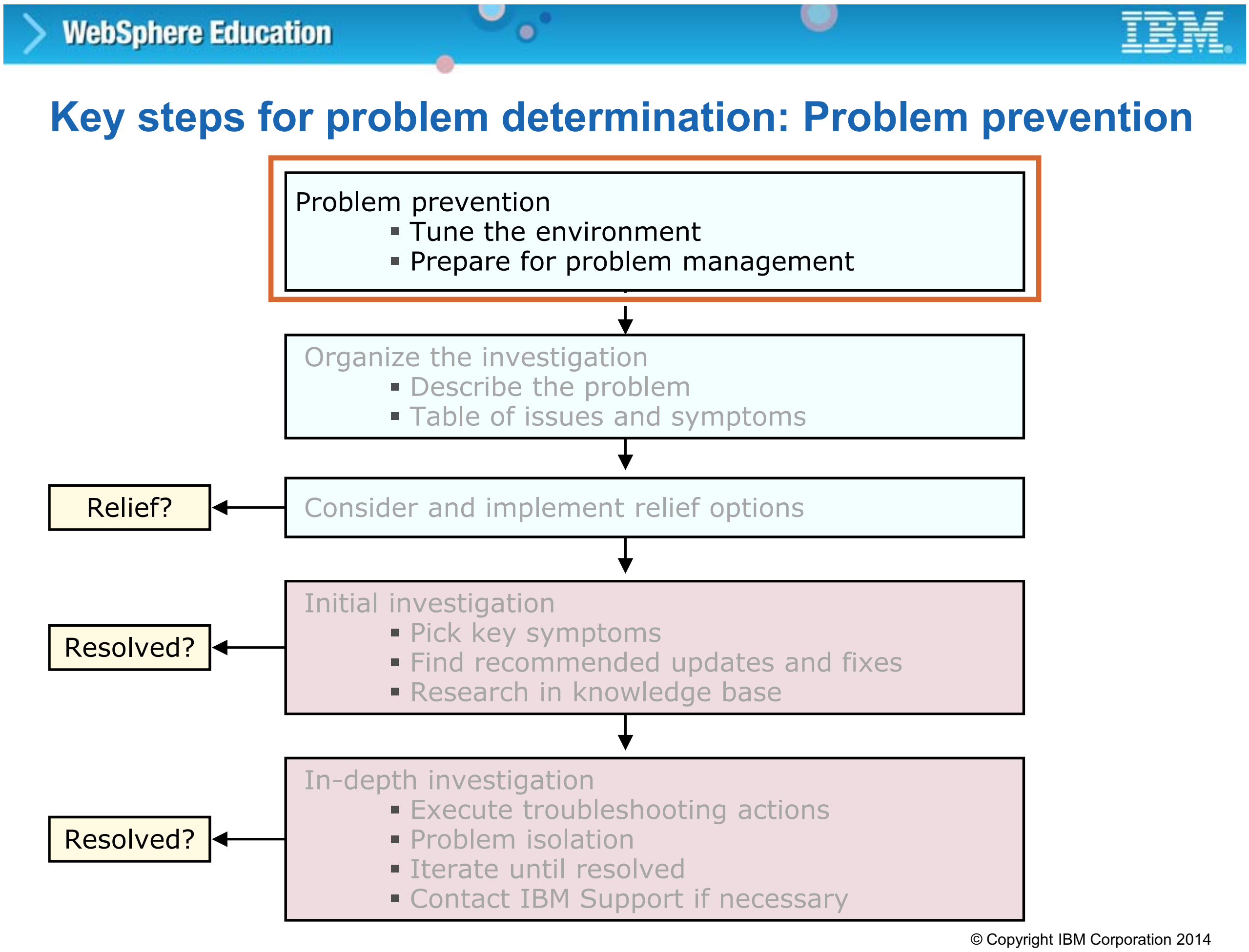


Figure 1-8. Key steps for problem determination: Problem prevention WB8691.0

Notes:

This topic provides an overview of problem prevention.

The job of a troubleshooter does not begin after a problem occurs. Thorough preparation should be done well before a problem occurs.

One problem that some troubleshooters face is that they see a problem (for example, the system crashes) and start conducting many long and complex analyses, maybe for days and weeks. Meanwhile, the system is not working, and users are seeking short-term relief. Relief is something that one should always keep in mind throughout the problem determination process.

WebSphere Education

Types of problem symptoms

- Here are the common types of symptoms that you might see:
  - The system is not responding
  - An application failed to start
  - An application does not respond to incoming requests
  - An application produces unexpected results
  - An application cannot connect to an external system or resources
  - An application performs slowly or its performance degrades over time

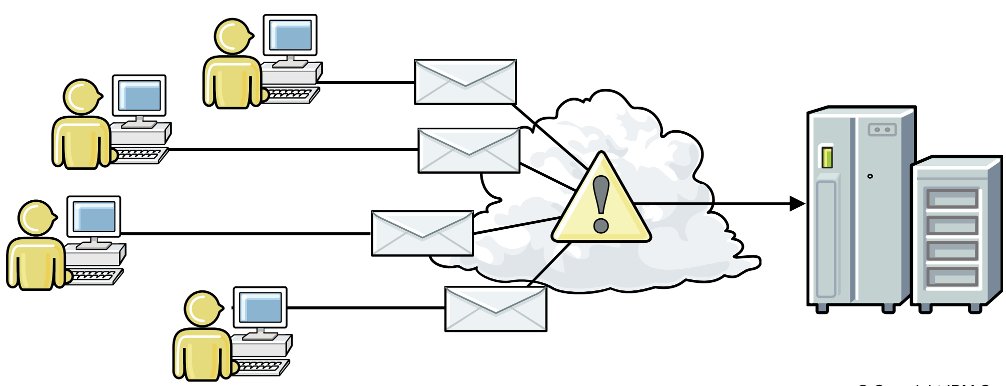


Figure 1-9. Types of problem symptoms WB8691.0

Notes:

Most problems are due to:

- Environment and configuration issues
- Misunderstanding or miscommunication
- Hard-to-diagnose application issues

Most problems are not related to product code defects. Non-defect-oriented problems (NDOP) are typically greater than 90% of PMRs. Rediscoveries (both DOP and NDOP) are typically greater than 50% of PMRs. Some problems take more time to resolve because they are hard to diagnose. Some problems are being seen for the first time. Other problems are critical to production or involve issues in the IBM Business Process Manager internal components. Finally, some problems might involve multiple interdependent issues.

Also, the approach to problem determination can be different, depending on whether the symptoms are seen in a development environment or production environment.

WebSphere Education

Problem prevention

- Tune and monitor the environment regularly
- Keep good system documentation
  - Create system architecture or topology diagram
  - Establish baselines
  - Maintain change log
- Create and maintain the following plans:
  - Diagnostic data collection plan
  - Relief or recovery plan
  - Maintenance plans (scheduled and emergency)
- Apply good practices for module design and connectivity groups
- Thoroughly plan application design, exception types, and fault processing
- Create an error-handling strategy
  - Examples: consistent fault processing for all component types, use of retry logic
- Test, test, and test

Figure 1-10. Problem prevention WB8691.0

Notes:

Good problem determination starts long before anything bad happens. Here are some actions that you can take to help prevent problems from occurring.

Infrastructure monitoring

Infrastructure monitoring and the use of infrastructure monitoring tools is a requirement for a production system. System administrators can use these tools to detect critical system behavior and react appropriately to prevent an outage. IBM provides infrastructure monitoring tools such as IBM Tivoli Composite Application Manager and Tivoli Performance Viewer. Regardless of whether the tool is from the IBM product stack or not, a required basic level of IT monitoring for the production system is essential to meeting availability service level agreements.

Tuning

Tuning exercises are a regular part of the system development lifecycle. A performance evaluation must be scheduled along with each major application deployment. Each solution that is released to the production environment should be evaluated and tested in the preproduction environment to measure the impact to existing applications and the current system parameters and resources. If this activity is missed, the solution suffers a higher-than-necessary risk of having a recovery problem.

System documentation

When investigating a problem, you need 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 several important devices for formalizing this information:

- A topology or flow diagram
- A series of baselines for the normal behavior of the system
- Change logs

Diagnostic data collection plan

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 diagnostic data you collect and how you collect them.

Relief or recovery 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 occurs.

Maintenance plans

Applying regular maintenance is one of the key factors in reducing the probability and effect 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.

Maintenance occurs at all levels: on the operation systems level, and on each of the products that are involved in the system. You can track current maintenance levels in the topology diagram.

Module design and connectivity groups

It is a good practice to create a “connectivity group” to represent the possible request sources for the system. A connectivity group is a specific pattern of behavior that is found in an SCA module. The connectivity group does not contain useful component types such as long-running business processes and business state machines. These connectivity groups provide encapsulation and isolation of the specific endpoint’s integration requirements. WebSphere Enterprise Service Bus mediation modules are commonly used for this purpose as they represent convenient ways to implement infrastructure-related tasks.

The concept of connectivity groups also provides a convenient way to quiesce the system in case there is a need for recovery. Since a connectivity group module is stateless, the module can be temporarily stopped, thus cutting off the inbound flow of new events. After the system is recovered and able to process new work, these modules can be restarted.

Application design, exception types, and fault processing

Good application design takes advantage of the error handling and fault processing capabilities from IBM Business Process Manager and WebSphere Enterprise Service Bus. It is, therefore, necessary for solution architects to understand how IBM Business Process Manager and WebSphere Enterprise Service Bus represent declared and undeclared exceptions before they can create a comprehensive error-handling strategy.

The SCA programming model provides two types of exceptions:

- Service business exceptions: Exceptions that are defined on the service interface
- Service runtime exceptions: Exceptions that are undeclared (an unexpected condition in the runtime)

Create an error handling strategy

The architecture team must understand the error handling and recovery tools and capabilities of the product. This team is responsible for creating the error handling strategy for the project and must account for the following items:

- Appropriate usage of units of work (transactions and activity sessions)
- Declaration and usage of faults and ServiceBusinessExceptions
- Consistent fault processing for all component types, especially BPEL and mediation flow components
- Usage of retry logic and “continue on error” Business Process Choreographer capabilities
- Appropriate settings for completed process instance deletion
- Correct usage of synchronous and asynchronous invocation patterns
- Appropriate usage of import and export types
- Appropriate usage of the retry capability in mediation flows

In addition to this list, the architecture team must create design patterns in which built-in recovery capabilities, such as the IBM Business Process Manager and failed event manager, are used appropriately.

Test, test, test

The best tool for problem prevention in production is the execution of a comprehensive functional and system test plan. In general, tests for deployed solutions can be broken into two groups:

- Functional test: Tests that confirm the functionality of the implementation as compared to the business requirements
- System test: Tests that include cases to verify performance, high availability, and recovery service level agreements

WebSphere Education

IBM Business Process Manager: Handling business faults

- IBM Business Process Manager provides a finite number of locations where business events can be persisted
- The persistence and recovery of business events depend on the proper use and understanding of transactions and units of work
- Follow the application development guidelines to take advantage of the IBM Business Process Manager recovery capabilities
- Know about the important IBM Business Process Manager components, which can increase the possibility of the successful business event recovery
  - Failed Event Manager
  - Business Process Choreographer Hold and Retention queues
  - Administration of service integration bus (SIBus) destinations

Figure 1-11. IBM Business Process Manager: Handling business faults WB8691.0

Notes:

Application development, and programming guides and techniques, can be found in the IBM Business Process Manager documentation:
`http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/ditamaps/ic-homepage-bpm.html`

You can use the Business Process Manager documentation to learn more about:

- Failed events and the failed event manager
- Business Process Choreographer hold and retention queues
- Service integration bus configuration

1.3. Organize the investigation

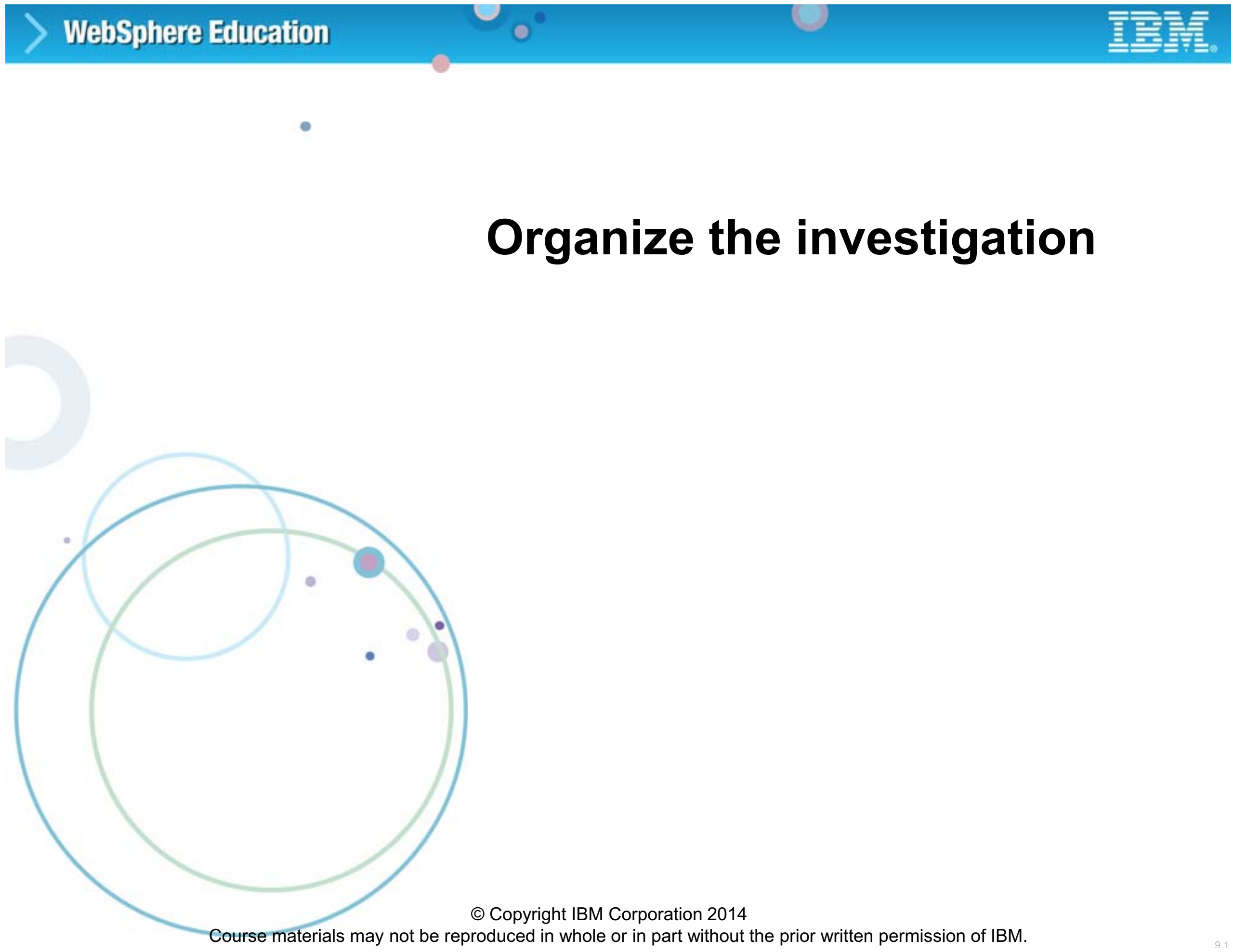


Figure 1-12. Organize the investigation WB8691.0

Notes:

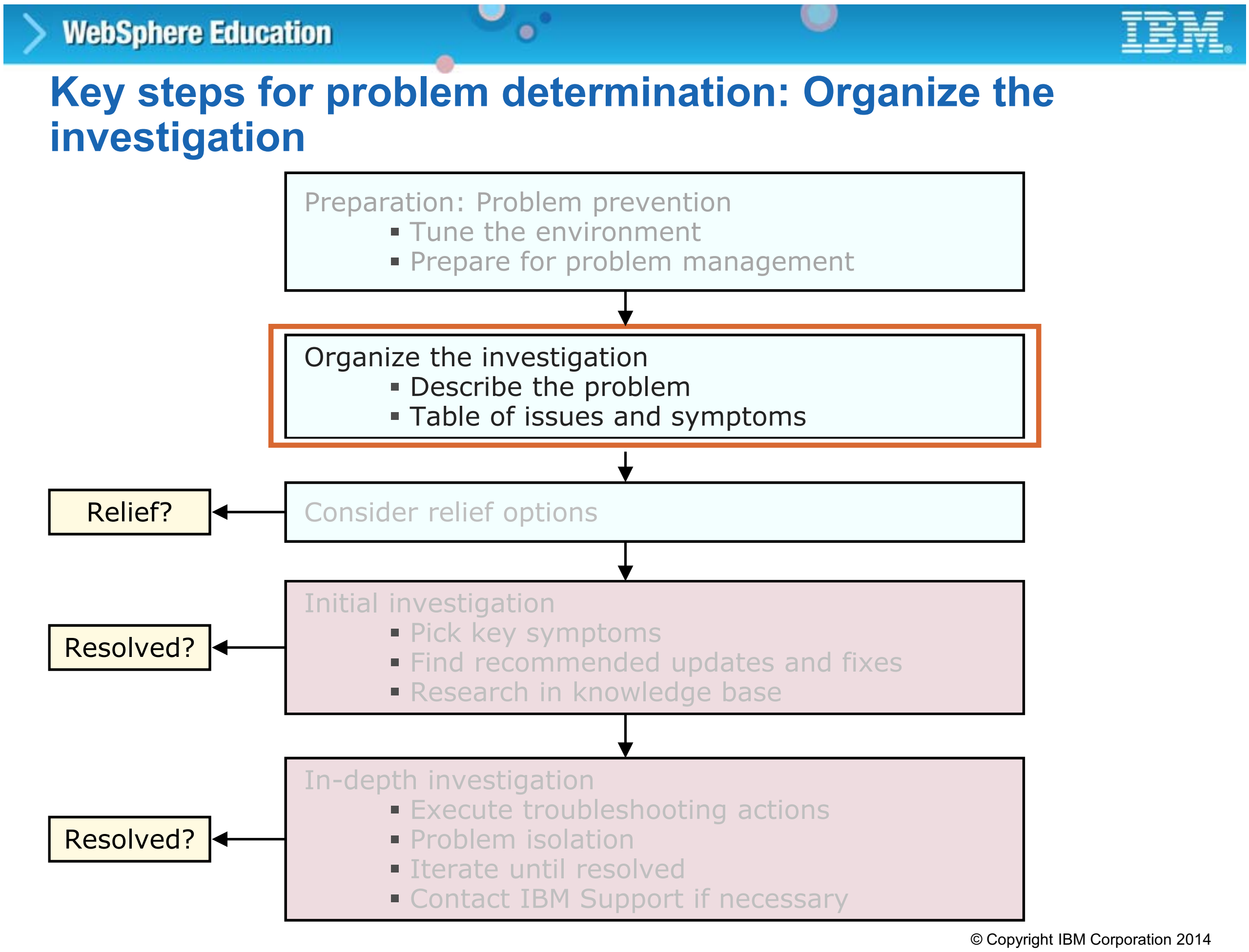


Figure 1-13. Key steps for problem determination: Organize the investigation WB8691.0

Notes:

This topic covers organizing the investigation.

WebSphere Education

Describe the problem completely

- Take time to carefully understand the problem and its context
  - Develop a clear and specific description of what happened
  - Be alert for possibly unrelated symptoms, and consider that several independent problems might occur at the same time
  - Beware of vague terms like crash or fails: a crash is not the same as a hang, which is not the same as an exit
- Listen and ask questions
  - Ask: What? Where? When? Why?
  - In many cases, asking questions is all that it takes to solve simple problems
  - For complex problems, not asking questions can result in considerable delays
- Create a table of symptoms and issues or a high-level timeline to record and describe:
  - Incidents
  - Status
  - Times when occurred
  - Action plans and other information

Figure 1-14. Describe the problem completely WB8691.0

Notes:

Troubleshooting is a systematic approach to solving a problem. The goal is to determine why something does not work as expected and how to resolve the problem.

The first step in the troubleshooting process is to describe the problem completely. Without a problem description, both you and IBM cannot know where to start to find the cause of the problem. Answering the questions that are listed in the slide can lead to a good description of the problem, and it is the best way to start down the path of problem resolution.

There is a good basic explanation of problem-solving methodology in G. Polya, “How to Solve It,” second edition, Princeton University Press, 1957, ISBN 0-691-08097-6. You can view a summary at the following website:

`http://www.math.utah.edu/~alfeld/math/polya.html`

WebSphere Education

Describe the problem completely: What and where

- **What** are the symptoms of the problem?
  - Who or what is reporting the problem?
  - What is the observed abnormal behavior?
  - What are the error codes and messages?
  - How does the system fail? Is it a loop, hang, crash, performance degradation, or incorrect result?
  - What is the business impact of the problem?
  - How would you recognize the same problem if it happens again?
  - What exactly would be different after the problem is solved?

- **Where** does the problem occur?
  - Is the problem specific to one platform or operating system?
  - Is it common to all multiple platforms or operation systems?
  - Are the current environment and configuration supported?

Figure 1-15. Describe the problem completely: What and where WB8691.0

Notes:

What are the symptoms of the problem?

When you start describing a problem, the most obvious question is “what is the problem?” This question might seem straightforward; however, you can break it down into several more focused questions that create a more descriptive picture of the problem.

Where does the problem occur?

Determining where the problem originates is not always easy, but it is one of the most important steps in resolving a problem. Many layers of technology can exist between the reporting and failing components. Networks, disks, and drivers are only a few of the components to be considered when you are investigating problems.

Remember that if one layer reports the problem, the problem does not necessarily originate in that layer. Part of identifying where a problem originates is understanding the environment in which it exists. Take some time to completely describe the problem environment, including the operating system and version, all corresponding software and versions, and hardware information. Confirm that you are running within an environment that is a supported configuration. Many problems can be traced back to incompatible levels of software that are not intended to run together or were not fully tested together.

WebSphere Education

Describe the problem completely: When

- **When** does the problem occur?
  - Does the problem happen only at a certain time of the day or night?
  - Did it happen only one time, or does it happen repeatedly?
- Under what sequence of events or conditions does the problem occur?
  - Does the problem happen after an environment change, such as upgrading or installing software or hardware?
  - Does the problem always occur when the same task is being performed?
  - Does a certain sequence of events need to occur for the problem to surface?
  - Do any other applications fail at the same time?
- If the problem happens repeatedly, characterize the circumstances
  - Apparently random times?
  - What frequency?
  - Every time that you do X, or every time some other external event happens?

Figure 1-16. Describe the problem completely: When WB8691.0

Notes:

When does the problem occur?

Develop a detailed timeline of events that led up to the failure, especially for those cases that are one-time occurrences. You can most easily do this activity by working backward. Start at the time an error was reported as precisely as possible (even down to the millisecond), and work backward through the available logs and information. Typically, you need to look only as far as the first suspicious event that you find in a diagnostic log. However, this activity is not always easy to do and takes practice. Knowing when to stop looking is especially difficult when multiple layers of technology are involved, and when each has its own diagnostic information.

WebSphere Education

Describe the problem completely: Why?

Consider **why** the problem occurs:

- Are you attempting something new for the first time?
  - Is it the first time the product was installed?
- Has anything changed in the environment?
  - Configuration changes in the failing system
  - Changes to dependent systems
- Does this problem occur in similar environments?
- Can the problem be reproduced?
  - Can the problem be re-created on a test machine?
  - Do multiple users or applications encounter the same type of problem?
  - Can the problem be re-created by running a single command, a set of commands, a particular application, or a stand-alone application?

Figure 1-17. Describe the problem completely: Why? WB8691.0

Notes:

Consider **why** this problem occurred here and now, and not in the past. Has anything in the system environment changed recently? Can the problem be reproduced, and if so, what are the conditions that are involved?

WebSphere Education IBM

Table of issues and symptoms and high-level timelines

Both tables of issues and symptoms and high-level timelines list all major events when they occur

- Incidents (occurrences of the same or other problem)
- Data collection actions and experiments
- Relief and remedy actions
- Other maintenance

Useful for:

- Identifying patterns, cause, and effect
- Avoiding confusion about what really happened
- Facilitating communication between all parties (customer, IBM Support, and others)
- Reinforcing change control policies
  - If too many things change at the same time, it might never be possible to understand the problem

Figure 1-18. Table of issues and symptoms and high-level timelines WB8691.0

Notes:

One practice is to build a high-level timeline or a table of issues and symptoms.

In either of these documents, you list all major events and when they occur. Include incidents, data collection actions and experiments, relief and remedy actions, and other maintenance actions. This practice helps to identify patterns, cause and effect, avoiding confusion about what happened, facilitating communication among all parties, tracking all available diagnostic artifacts and what they correspond to, and reinforcing change control policies.

WebSphere Education IBM

Example: Table of issues and symptoms

No	Status, priority	Symptom or issue	Plan of action, disposition
#1	Open/High	AppServer unresponsive to HTTP requests	Awaiting result of investigation on #8
#9	Open/Medium	Large number of HTTP processes: observed during #1	Defer
#8	Open/High	"CONM6026W: Timed out waiting for connection from data source": observed during #1	Enable connection pool diagnostic tracing
#5	Open/Medium	Application error: "Cannot validate credit card"	Awaiting feedback from application developer
#2	Open/High	AppServer crashes: Happens sometimes after #1, but not always	Awaiting result of investigation on #6
#6	Open/High	verboseGC reports allocation failure for large (2BM) object: last entry in log before #2	Capture heap memory dump

Figure 1-19. Example: Table of issues and symptoms WB8691.0

Notes:

Developing a table of issues and symptoms is not necessarily a required step, but it is a good idea.

List all top-level issues reported by users of the system, and all low-level symptoms observed during the investigation.

These tables are useful for:

- Organizing the investigation: finding all symptoms and always identifying what to check on next
- Tracking progress
- Making sure that nothing is overlooked

Prioritize and cluster entries to reflect the current state of the investigation while you constantly update and revise the information as the investigation progresses.

This example is just one way of developing a table of issues and symptoms. This particular example is a complex situation, where suddenly there are many seemingly independent symptoms and several unrelated problems. You do not have to use the format that is shown in the slide. You can customize it to your specific needs.

WebSphere Education

Troubleshooting checklist for IBM Business Process Manager

1. Is the configuration supported?
2. Have you applied the latest fixes?
3. What is the problem?
   - Installing and configuring IBM Business Process Manager
   - Migrating existing applications and configuration information to IBM Business Process Manager
   - Deploying applications on IBM Business Process Manager
   - Administering applications and components on IBM Business Process Manager
   - Using **WebSphere Application Server** capabilities in IBM Business Process Manager
4. Have any error messages been generated?

Figure 1-20. Troubleshooting checklist for IBM Business Process Manager WB8691.0

Notes:

Refer to the requirements for IBM Business Process Manager to ensure that your system meets all hardware, operating system, and software requirements. Check the IBM Business Process Manager Advanced system requirements website:

`http://www.ibm.com/support/docview.wss?uid=swg27023005`

1.4. Consider and implement relief options

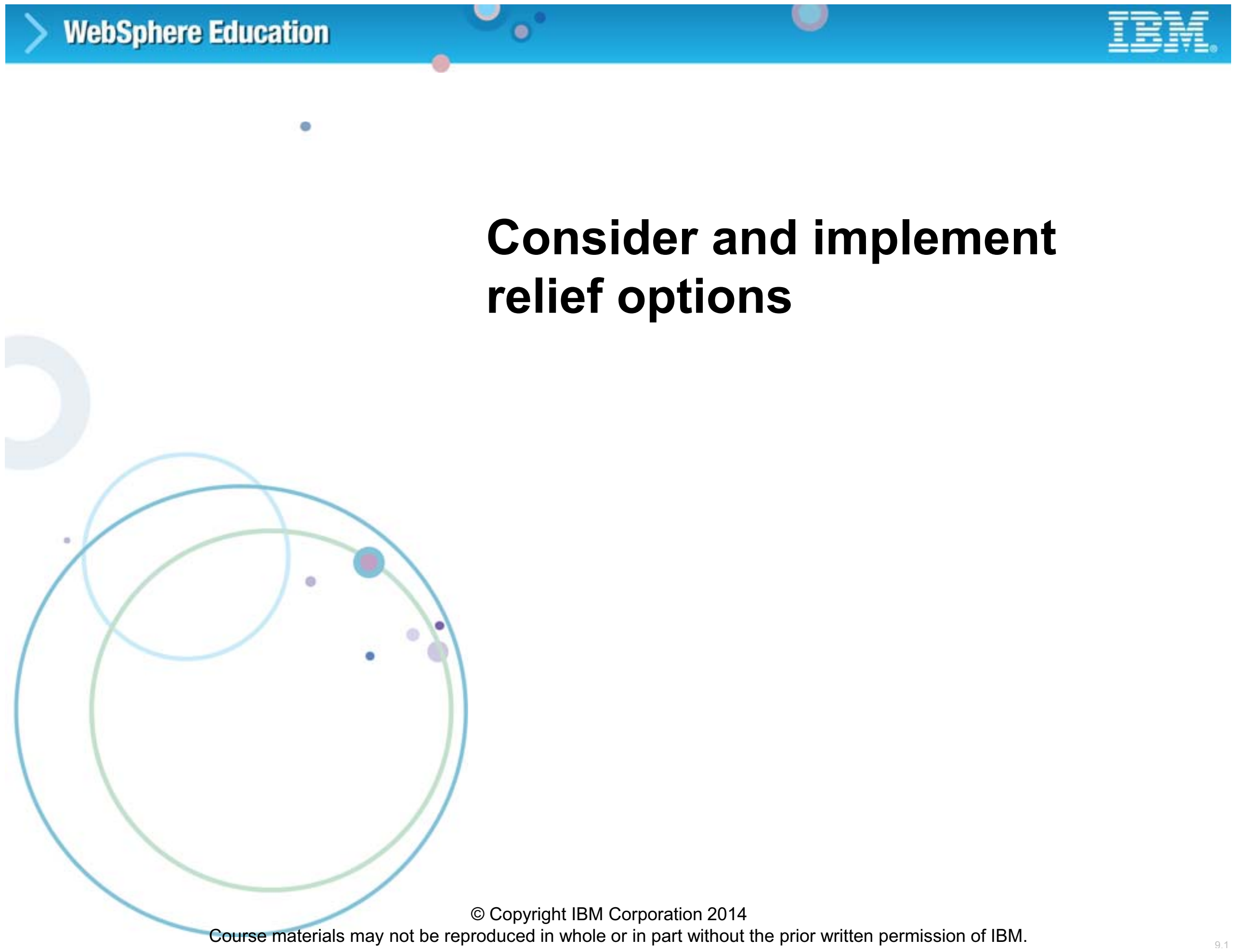


Figure 1-21. Consider and implement relief options WB8691.0

Notes:

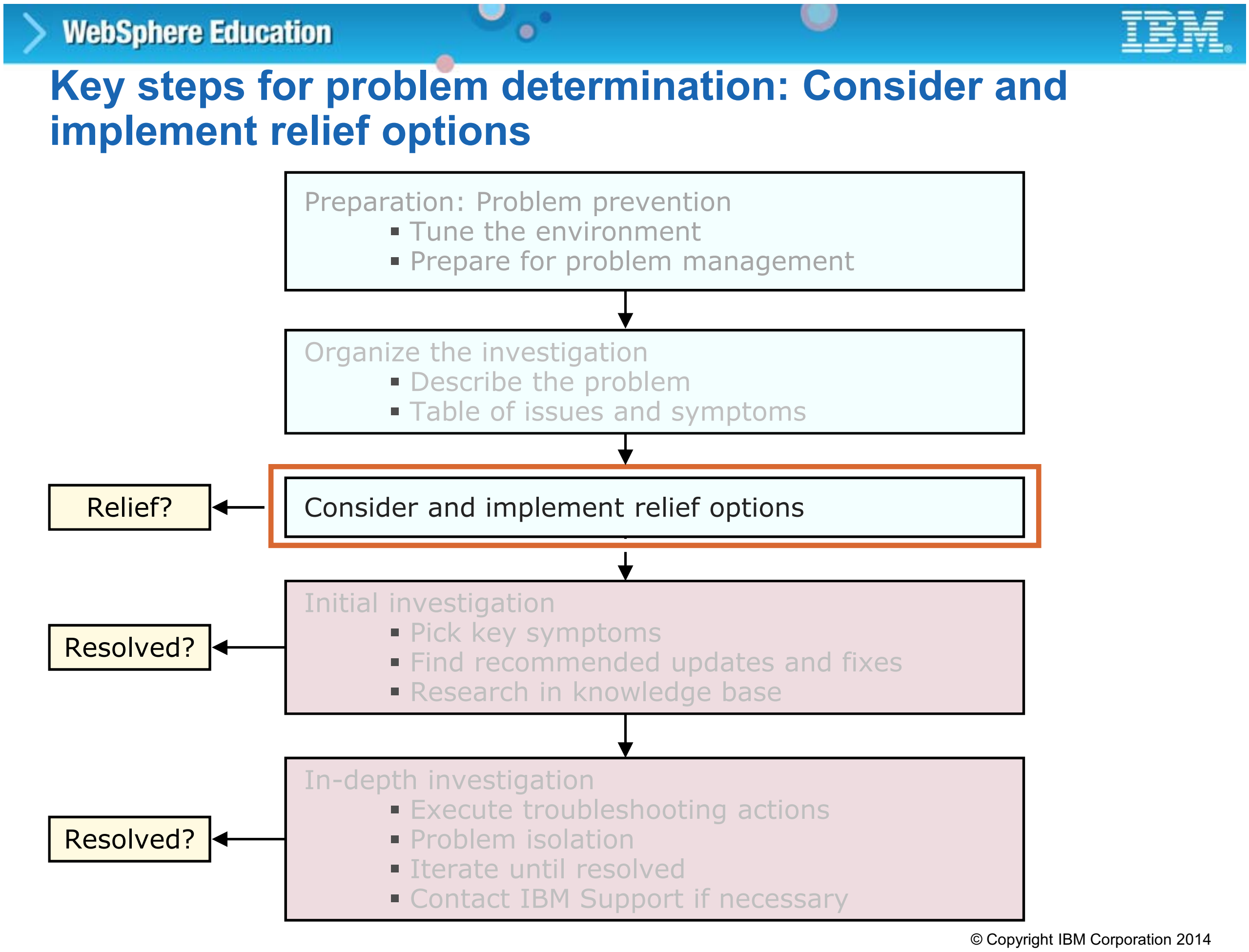


Figure 1-22. Key steps for problem determination: Consider and implement relief options WB8691.0

Notes:

WebSphere Education IBM

Relief options

- If the problem occurs in production or other critical-availability systems, you must provide relief long before you fully resolve the problem
- Consider the relief timeline:
  - How much time do you have to investigate the problem before you need to provide some relief?
  - Can you keep the system in its failed state in case it has information that you want to collect later?
- Identify the relief actions
  - You can usually restart one or more components, but beware of the chain-reaction effects of stopping and starting components in a live system
  - Sometimes, you can change the usage characteristics of the application, such as reducing the load and avoiding some “dangerous operations”

> Dangerous operations refer to any function in the application that is deemed to trigger a failure due to its complexity, or its dependency on other components

- Reevaluate relief options at every step through the investigation

Figure 1-23. Relief options WB8691.0

Notes:

When a problem occurs, one of the main priorities, independent of any investigation, should be to restore function to *users*. It is impossible to predict all possible situations in advance, but confusion, wasted time, and even greater disasters can occur when relief actions are attempted impulsively and without forethought.

The relief or recovery plan

After a problem occurs, consider how much time is available to look into the problem before you must provide relief to affected users. It is a good practice to create a relief or recovery plan to help you restore function to users.

The relief or recovery plan lays out general steps that you undertake to restore functions, and actions to take for specific problems.

For the relief or recovery plan:

- Try to predict the most common types of problems, which are based on knowledge of the system topology and flows:
  - Loss of an application server

  - Loss of database connectivity
  - Loss of the LDAP server
- Identify different regions of the system that can be isolated or restarted independently:
  - Applications (Java EE, BLA, OSGi)
  - Application servers (JVM)
  - Application server clusters
  - Servers
- Document the following processes:
  - Who decides what to do and who does it?
  - Have criteria for making such decisions.
- Practice the most common relief actions in advance.
  - Ensure that you know how to do them.
  - Ensure that they have no unexpected side effects.

1.5. Initial investigation

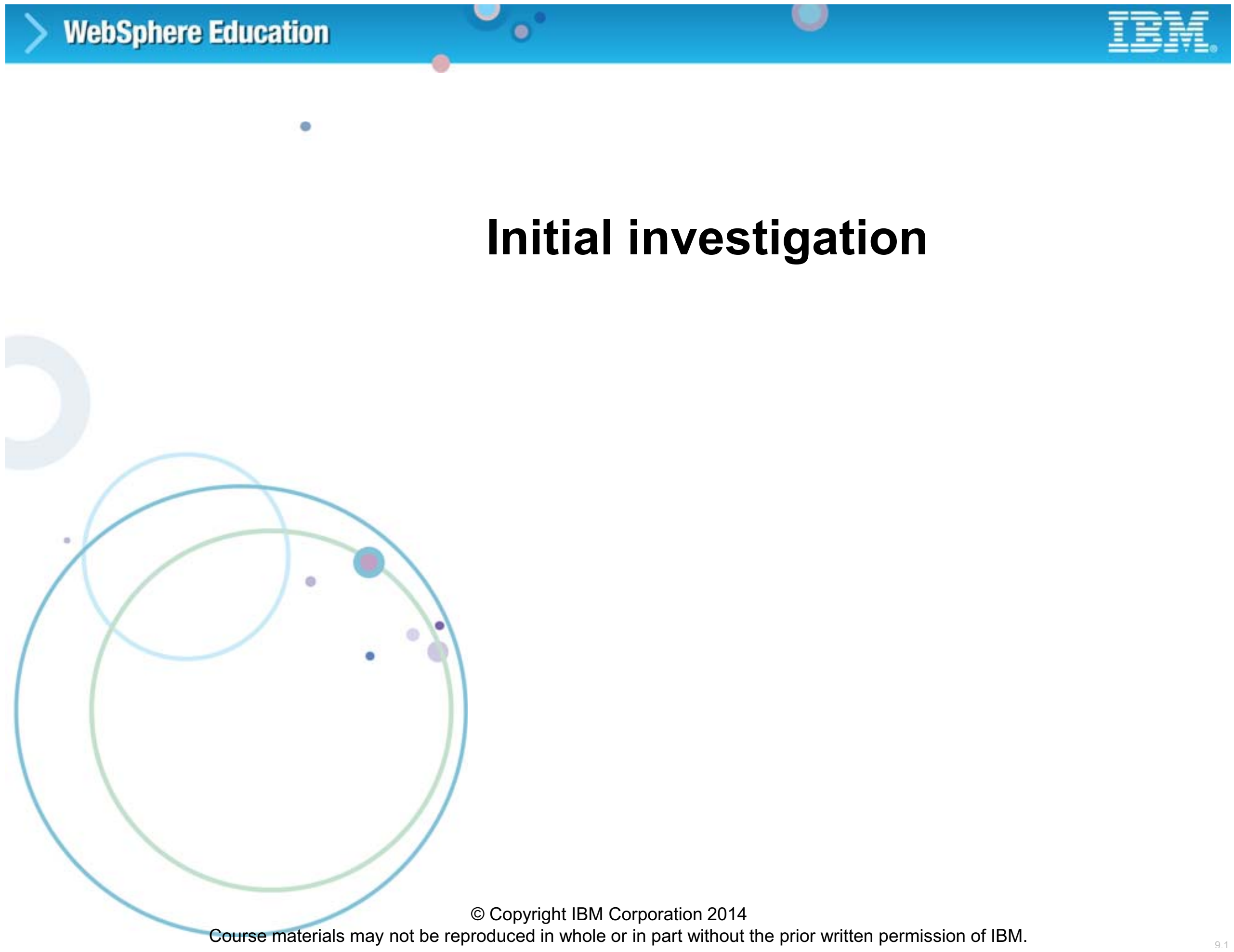


Figure 1-24. Initial investigation WB8691.0

Notes:

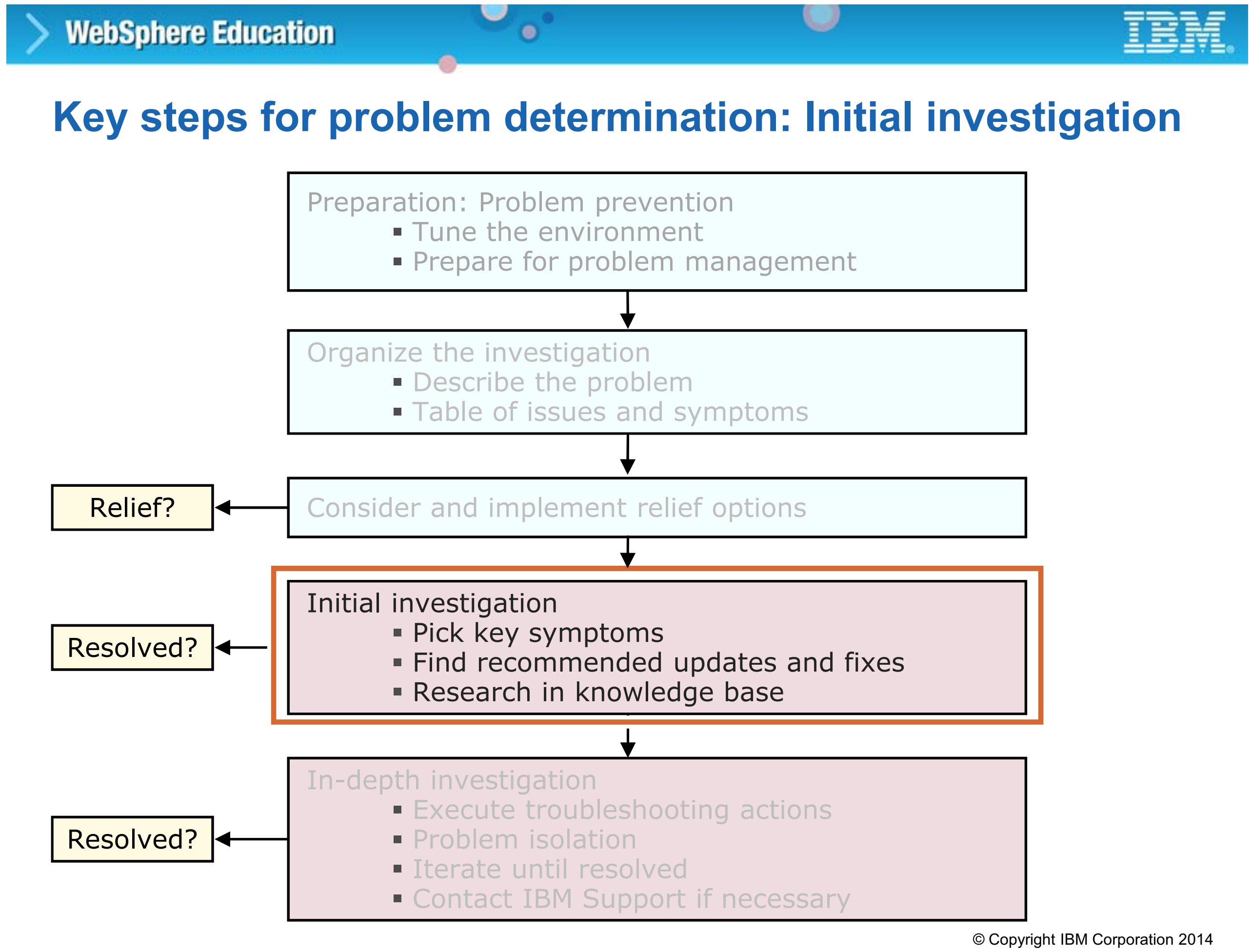


Figure 1-25. Key steps for problem determination: Initial investigation WB8691.0

Notes:

WebSphere Education

Search for known problems and other relevant information

The goal at this stage is to establish a starting point for further investigation, if necessary

- Locate known problems and solutions
  - Research the top symptoms in IBM Support sites and developerWorks
  - Search for APARs
  - Check recommended fixes for IBM Business Process Manager

- Search for other relevant information
  - Search for relevant *Technotes* through the IBM Business Process Manager support site
  - Check the IBM Business Process Manager forums on developerWorks
  - Search for troubleshooting information in the IBM Business Process Manager information center

Figure 1-26. Search for known problems and other relevant information WB8691.0

Notes:

WebSphere Education

Inventory anomalies and symptoms

Make an inventory of all pertinent anomalies and potential symptoms:

- Errors, warnings, exceptions, out-of-range statistics, any other unusual behavior
  - Scan through available logs
- Use the topology diagram, baselines, and change logs that you prepared earlier
  - Use the flows in the topology diagram as a guide for research
- Integrate found information into the table of issues and symptoms or high-level timeline
- Assess and prioritize symptoms

Figure 1-27. Inventory anomalies and symptoms WB8691.0

Notes:

Remember: Troubleshooting is not a perfect science.

It can be hard to tell which symptom is a cause and which symptom is a consequence of the original problem.

WebSphere Education IBM

Document events and communicate

- 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
  - Include anomalies and symptoms
  - Monitor multiple components in parallel and attempt to correlate
- 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
  - Use topology flows, timelines, table of issues and symptoms, and other documents

Figure 1-28. Document events and communicate WB8691.0

Notes:

1.6. In-depth investigation

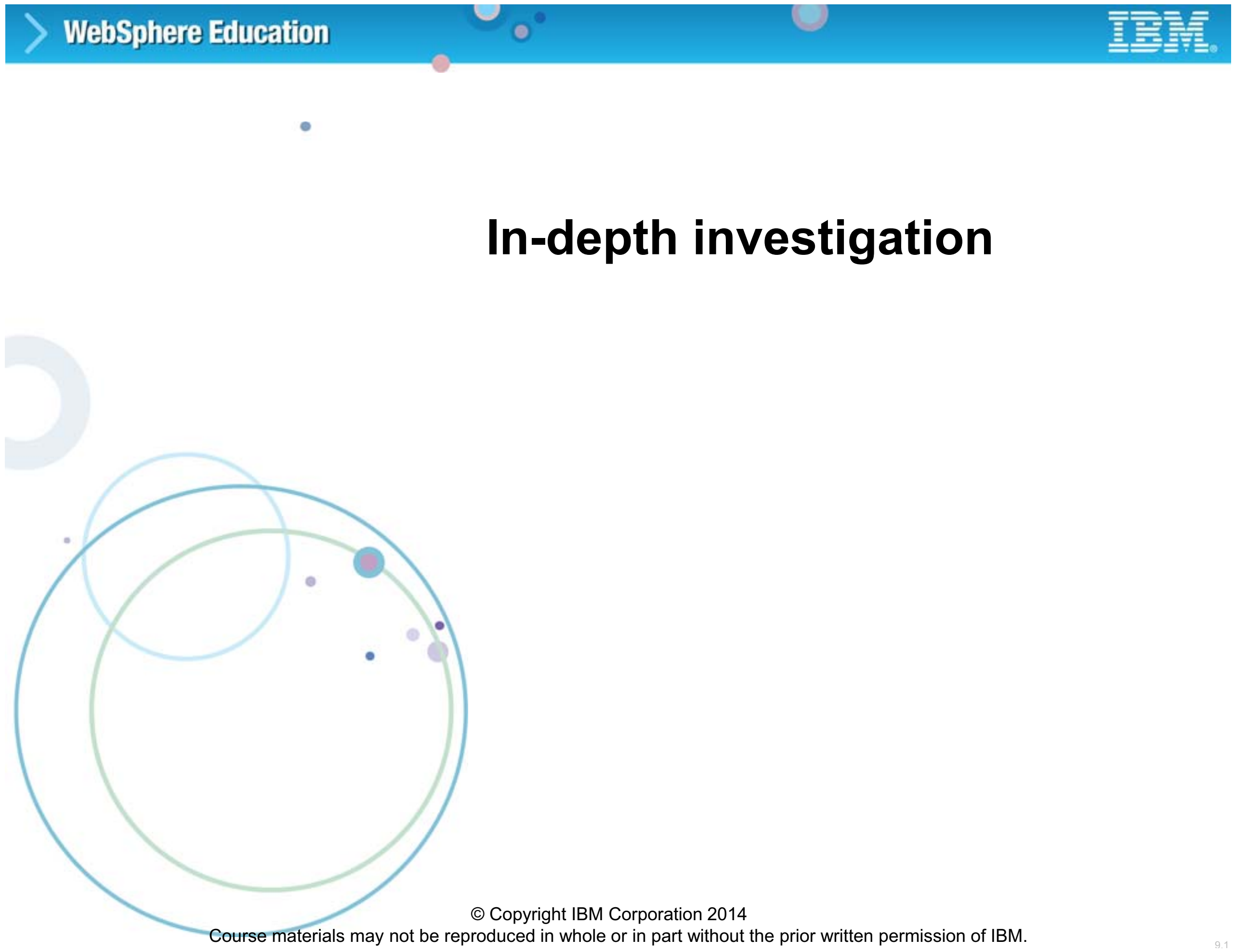


Figure 1-29. In-depth investigation WB8691.0

Notes:

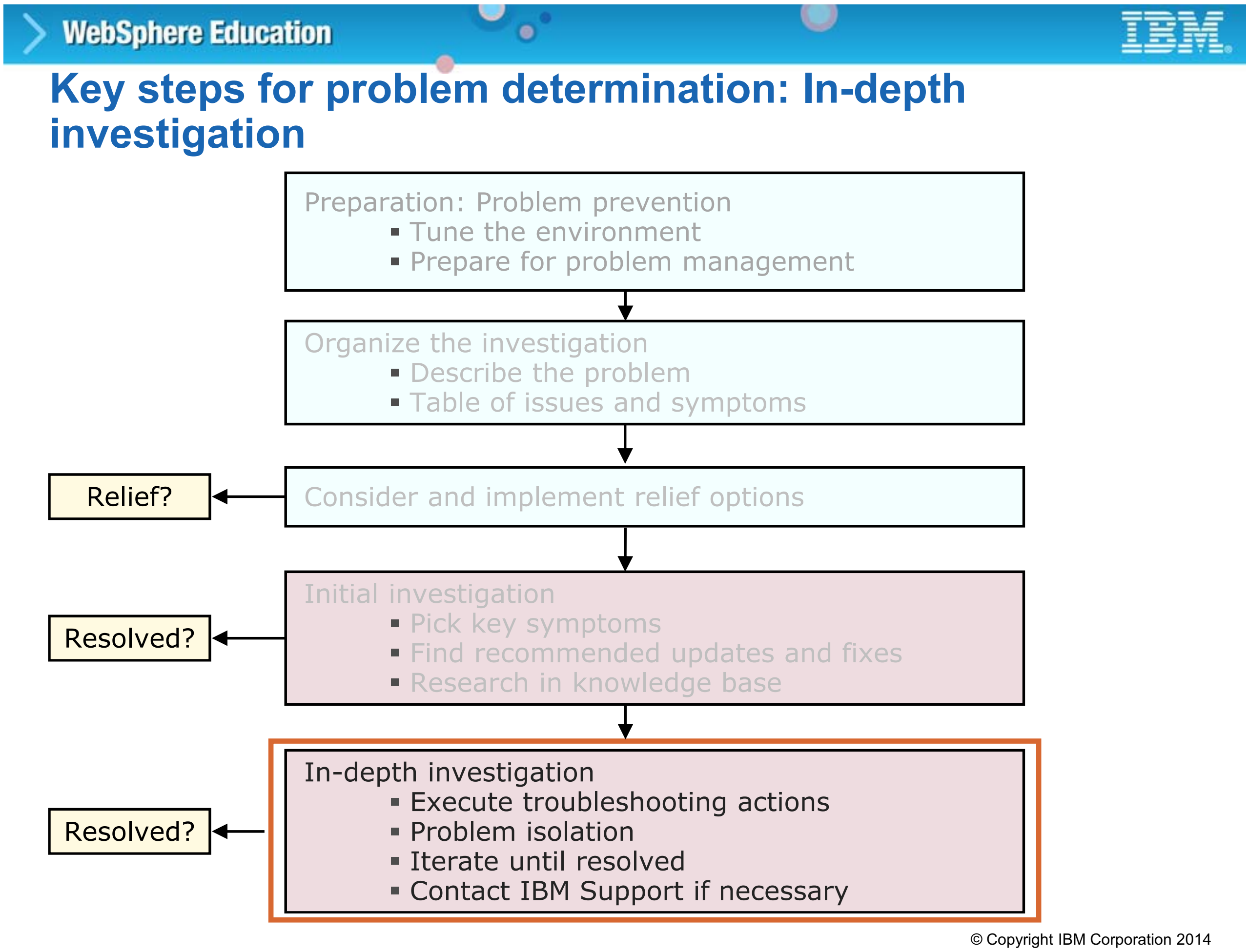


Figure 1-30. Key steps for problem determination: In-depth investigation WB8691.0

Notes:

WebSphere Education IBM

In-depth investigation

- If the initial research did not find a solution, you must undertake a more in-depth investigation
  - Two main approaches: analysis and isolation
  - Each approach uses a specific set of methods and tools for each problem
  - If necessary, contact IBM Support
- There are several sources of information to start with:
  - Troubleshooting section in the information center
  - IBM Problem Determination Redbooks
  - Collect MustGather documents
  - Define initial set of diagnostic data to focus on (either for IBM support or for internal investigation)
- From there, search for other technotes with troubleshooting instructions, tools, and other resources

Figure 1-31. In-depth investigation WB8691.0

Notes:

WebSphere Education

Analysis approach

- Choose one or more specific symptoms; then drill down to gain more detailed information about the items and their causes

- Techniques used:
  - Search knowledge bases, traces, and system dumps
  - Use available diagnostic data in increasing detail until solution is found
  - Leverage the tooling and concentrate on must-gathers
  - Collect and analyze diagnostic data (possibly through several iterations) until the root cause is found

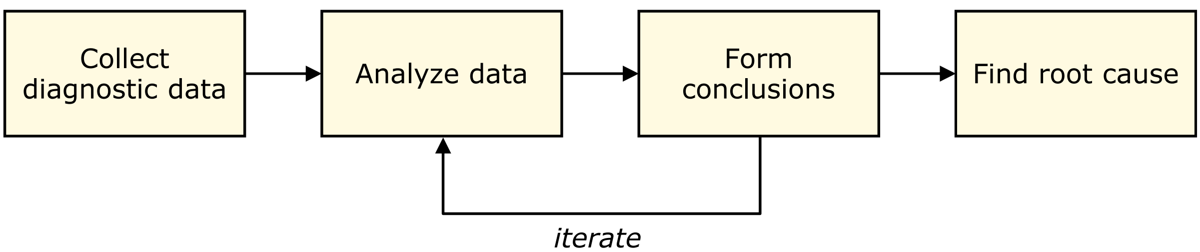


Figure 1-32. Analysis approach WB8691.0

Notes:

Analysis approach example: Consider a problem in which an application server appears to be failing. You analyze the diagnostic material from the crash and determine that the crash originated in a library called `mylib`. Next, you look at the source code for the library. You take the native stack trace information from the gathered diagnostic material and review it. You can see that a bit of code creates a pointer to a memory location, but does not handle it correctly. Running this code results in an invalid operation and a subsequent crash.

WebSphere Education

Isolation approach

- Examine the context in which each symptom occurs within the system and the relation of each symptom to each other, eliminating factors
  - Reproduce the problem, analyze results, and remove variables (possibly through several iterations) until the root cause is found

- Techniques used:
  - Perform specific experiments to observe behavior changes
  - Break down the system or operations into smaller parts

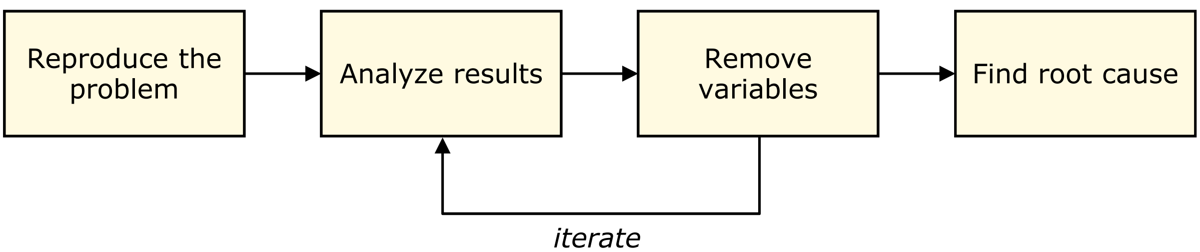


Figure 1-33. Isolation approach WB8691.0

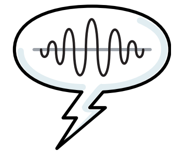
Notes:

Isolation approach example: Consider a large WebSphere Application Server environment, which consists of many nodes across several physical machines. This environment includes an accounting application, which is deployed into two clusters. This accounting application has long response times. Using the isolation approach, you might opt to trace the application servers that are involved along with the network links between the servers. With this method, you can isolate the cause of the slowdown between the network and the application servers. Isolating the cause of the slowdown makes it possible for you to do a more in-depth investigation on the affected component.

WebSphere Education

Communicating with IBM support

- Define the problem
  - Describe the problem and symptoms
  - Be as specific as possible when explaining a problem
- Gather background information
  - What levels of software were you running when the problem occurred?
  - Has the problem happened before, or is this problem an isolated one?
  - What steps led to the failure?
  - Can the problem be re-created?
  - Were any messages or other diagnostic information produced?
  - Define your technical question in specific terms and provide the version and release level of the IBM Business Process Manager
- Gather relevant diagnostic information
- Determine the severity level



Often, the biggest challenge in problem determination is effectively communicating the issue and how to reproduce it

Figure 1-34. Communicating with IBM support WB8691.0

Notes:

The IBM service request (SR) application is used to open and update service requests (formerly called problem management records or PMRs) online. You can access the SR application here: `https://www.ibm.com/support/servicerequest/Home.action?lnk=msdTS-srap-usen`

WebSphere Education

Severity level situations and examples

Severity level	Definition	Examples
1	**Critical impact: system down** • Business-critical software component is inoperable or critical interface failed • This condition requires an immediate solution	• A live IBM Business Process Manager instance fails to complete a request, and generates an error • Live production customer data is directly affected
2	**Significant business impact** A software component is severely restricted in its use, or you are in jeopardy of missing business deadlines because of problems with a new application rollout	Development of a new version of an application is giving incorrect output, and the application is scheduled for rollout in one month
3	**Some business impact** Indicates that the program is usable with fewer significant features (not critical to operations) unavailable	• A client cannot connect to a server, but there is an alternative to do the same task • Does not affect the business
4	**Minimal business impact** A noncritical software component is malfunctioning, causing minimal impact, or a nontechnical request is made	• Documentation is incorrect • More documentation is requested

Figure 1-35. Severity level situations and examples WB8691.0

Notes:

This table describes the definition of severity levels that IBM Support uses, and it also provides an example for each level.

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Describe the key steps in problem determination
- Characterize a problem from its symptoms
- Implement a relief or recovery plan
- Describe the basic troubleshooting checklist for IBM Business Process Manager
- Describe a problem efficiently to IBM support

Figure 1-36. Unit summary WB8691.0

Notes:

WebSphere Education

Checkpoint questions

1. What are the key steps for problem determination?
2. What are the three main goals of problem determination?
3. What are relief options?
4. What are the two main in-depth investigation approaches?

Figure 1-37. Checkpoint questions WB8691.0

Notes:

Write down your answers here:

1.

2.

3.

4.

WebSphere Education

Checkpoint answers

1. The key steps for problem determination are:
   – Problem prevention
   – Organize the investigation
   – Consider and implement relief options
   – Conduct the initial investigation
   – Conduct an in-depth investigation, if needed
2. The three main goals of problem determination are:
   – Quickly provide a temporary solution (relief)
   – Find and implement the correct permanent solution
   – Try to make sure that similar problems do not recur
3. Relief options are actions that you can take to restore function to affected users while you conduct a problem investigation. For example, restart one or more components, or reduce the load.
4. The two main in-depth investigation approaches are analysis and isolation.

Figure 1-38. Checkpoint answers WB8691.0

Notes:

Unit 2. WebSphere Application Server and IBM Business Process Manager systems and components

What this unit is about

This unit provides an overview of WebSphere Application Server and IBM Business Process Manager systems and components.

What you should be able to do

After completing this unit, you should be able to:

- Describe the WebSphere Application Server architecture
- Describe the Network Deployment (ND) cell components
- Identify the components of IBM Business Process Manager and Process Center
- Describe the purpose and business value of IBM Integration Designer

How you will check your progress

- Checkpoint

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Describe the WebSphere Application Server architecture
- Describe the Network Deployment (ND) cell components
- Identify the components of IBM Business Process Manager and Process Center
- Describe the purpose and business value of IBM Integration Designer

Figure 2-1. Unit objectives WB8691.0

Notes:

WebSphere Education

Topics

- WebSphere Application Server architecture
- Network Deployment components
- IBM Business Process Manager and Process Center
- Business value of IBM Integration Designer

Figure 2-2. Topics WB8691.0

Notes:

2.1. WebSphere Application Server architecture

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. It describes the steps for collecting some of the general MustGather data so that you can use some of the log files during troubleshooting.

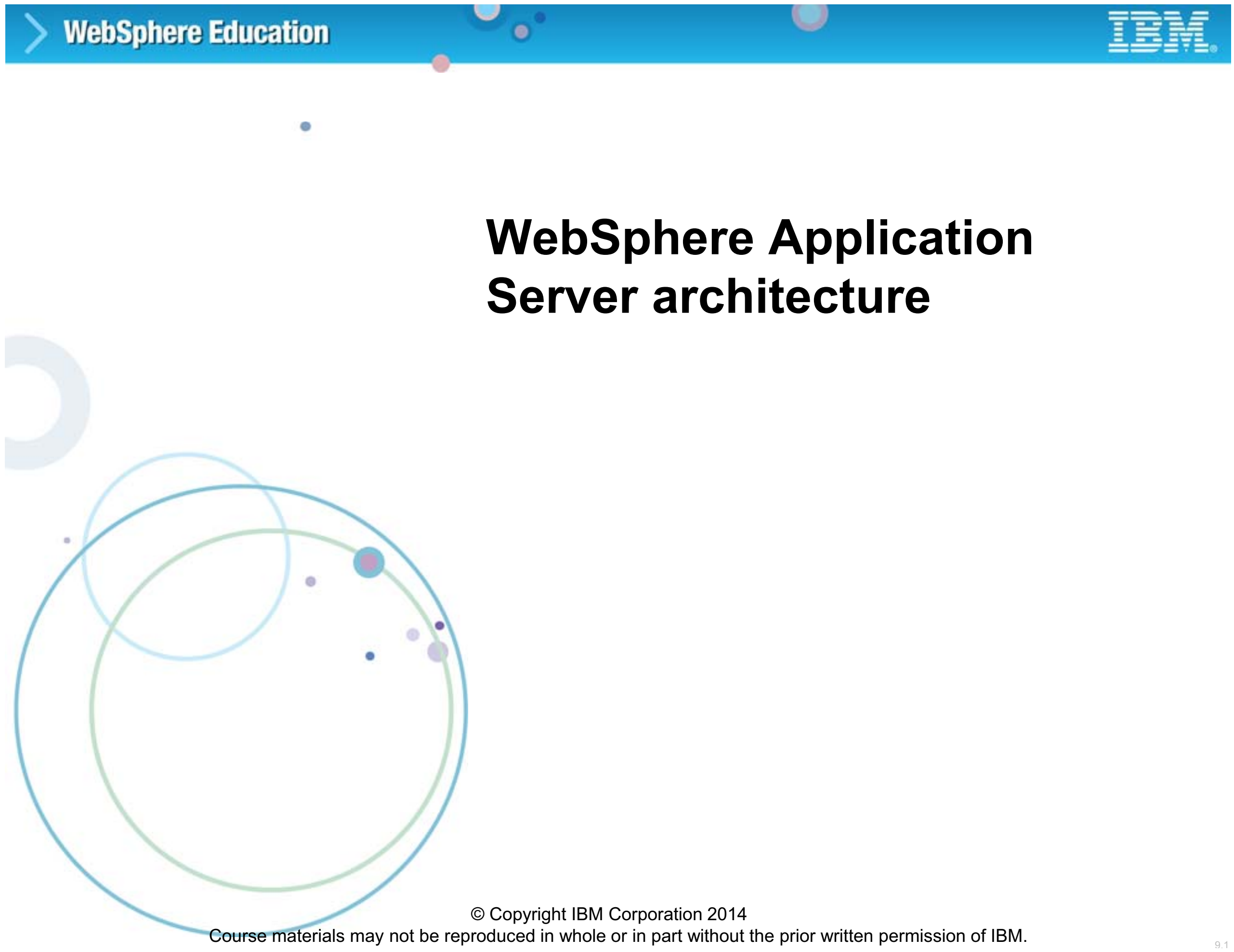


Figure 2-3. WebSphere Application Server architecture WB8691.0

Notes:

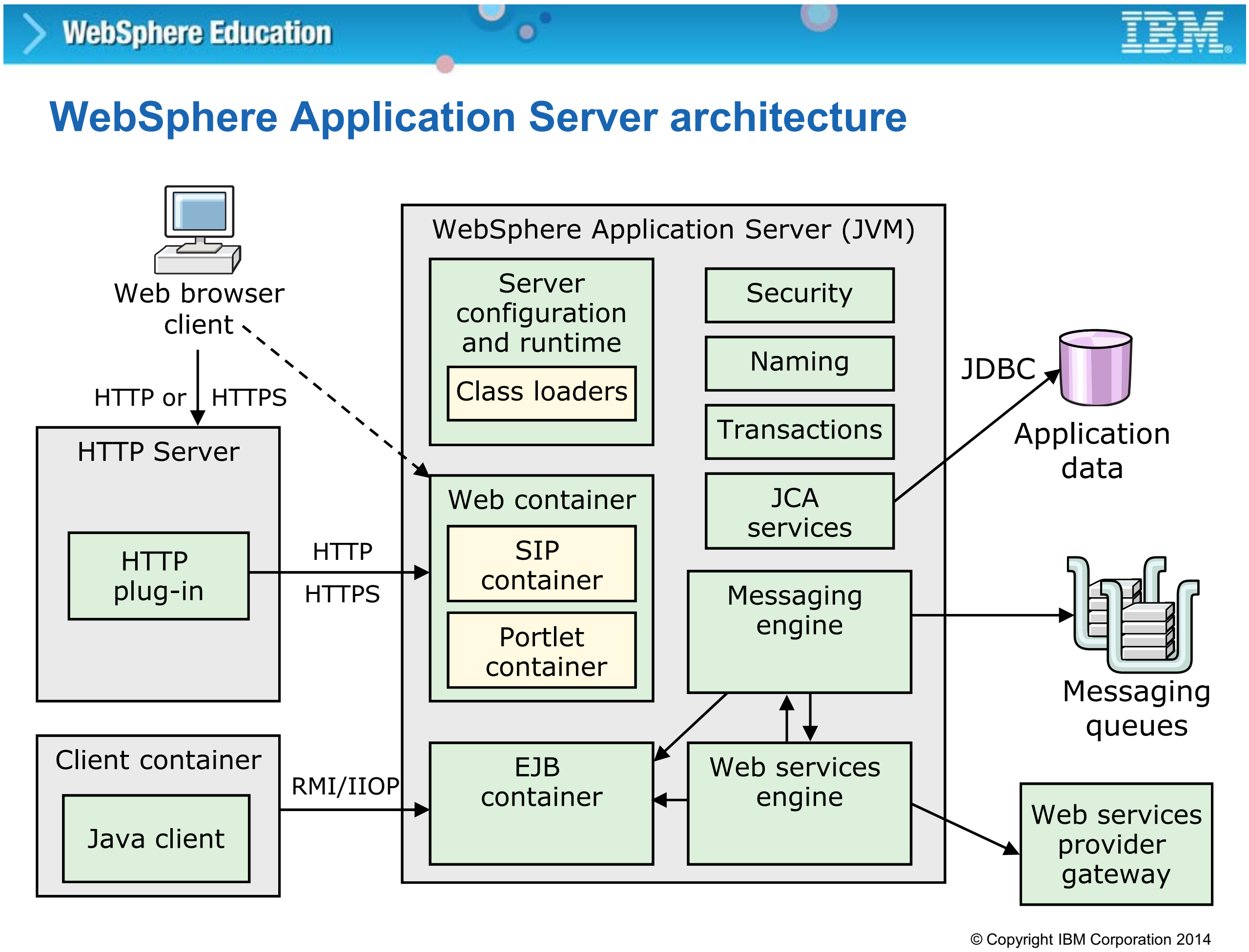


Figure 2-4. WebSphere Application Server architecture WB8691.0

Notes:

This diagram illustrates the basic architecture of WebSphere Application Server, including several of the larger components.

The main element is the application server, a Java process that encapsulates many services, including the containers, where business logic runs. If you are familiar with Java EE, you recognize the web container and the EJB container. The web container runs servlets and JavaServer Pages (JSPs), both of which are Java classes that generate web markup language. Traffic into and out of the web container travels through the embedded HTTP server. While servlets and JSPs can act independently, they most commonly make calls to Enterprise JavaBeans (EJBs) to run business logic or access data. EJBs, which run in the EJB container, are easily reusable Java classes. They most commonly communicate with a relational database or other external source of application data. These EJBs return data to the web container or update the data on behalf of the servlet or JSP.

The JMS messaging engine is built into the application server. This engine is a pure Java messaging engine. JMS destinations, which are known as queues and topics, provide asynchronous messaging services to the code that is running inside the containers. JMS is covered in more depth later in this course.

As you see in more detail later on, the web services engine enables application components to be exposed as web services, which can be accessed with SOAP.

Several other services are run within the application server, including the dynamic cache, data replication, security, and others. These services are covered later in the course.

There are also some important components outside of the application server process.

WebSphere Application Server also provides a plug-in for HTTP servers that determines what HTTP traffic is intended for WebSphere to handle, and routes the requests to the appropriate server. The plug-in is critical to workload management of HTTP requests, as it can distribute the load to multiple application servers, and steer traffic away from unavailable servers. It too reads its configuration from a special XML file.

WebSphere Education

Web container

- The web container processes:
  - Servlets
  - JSP files
  - Other types of server-side includes
- Each application server runtime has one logical web container, which can be modified, but not created or removed
- Each web container provides:
  - Web container transport chains
  - Servlet processing
  - JSP processing
  - HTML and other static content processing
  - Session management
  - Thread pool
  - Portlet container

Figure 2-5. Web container WB8691.0

Notes:

- **Web container transport chains**

  The web container inbound transport chain is used to direct requests to the web container. The chain consists of a TCP inbound channel that provides the connection to the network, and an HTTP inbound channel that serves HTTP 1.0 and 1.1 requests. The chain continues with a web container channel over which requests for servlets and JSPs are sent to the web container for processing.

- **Servlet processing**

  When handling servlets, the web container creates a request object and a response object; it then invokes the servlet service method. The web container invokes the destroy method of the servlet when appropriate and unloads the servlet, after which the JVM does garbage collection.

- **HTML and other static content processing**

  The web container inbound chain serves requests for HTML and other static content that are directed to the web container. However, in most cases, the use of an external web server and web server plug-in as a front end to a web container is more appropriate for a production environment.

- **Session management**

  Support is provided for the `javax.servlet.http.HttpSession` interface as described in the servlet application programming interface (API) specification.

- **Web services engine**

  Web services are provided as a set of APIs in cooperation with the Java EE applications. Web services engines are provided to support SOAP.

The following properties allow you to configure the services that the web container provides:

- **Default virtual host**

  This virtual host is the default to use for applications on the server.

- **Enable servlet caching**

  You can use dynamic cache to improve application performance by caching the output of servlets, commands, and JSPs. This setting allows you to enable dynamic caching for servlets. You must first enable dynamic caching and create the appropriate cache policies so that you can use servlet caching.

- **Session management**

  You can determine how the web container manages HTTP session data. This determination includes settings for the session tracking mechanism (for example, cookies), session timeout, and for the session persistence method.

- **Web container transport chains**

  You can add to or configure the communication channels that are used for accessing applications in the web container. By default, you have four transport chains predefined. These transport chains are for secure and nonsecure administration console access, and for default access to the web container. The transport chains are related to port definitions seen in the communications section. Port numbers must be unique for each application server instance on a machine.

- **Custom properties**

  You can specify name-value pairs for configuring internal system properties. Some components use custom configuration properties, which can be defined here. It is not common to pass information to the web container this way, but the Java EE specification indicates it as a requirement. Most configuration information can be handled programmatically, or through the deployment descriptor.

WebSphere Education

Enterprise JavaBeans (EJB) container

- The Enterprise JavaBeans (EJB) container provides the runtime services that are needed to deploy and manage enterprise beans
- It is a server process that handles requests for:
  - Session beans
  - Entity beans
  - Message-driven beans (MDBs)
- The enterprise beans that are packaged in EJB modules and installed in an application server do not communicate directly with the server
  - The EJB container provides an interface between the enterprise beans and the server
- Together, the EJB container and the application server provide the enterprise bean runtime environment
- The WebSphere runtime provides many low-level services, including:
  - Thread pool support
  - Transaction support
- The EJB container manages data storage and retrieval for the contained EJBs
- A single EJB container can host more than one EJB application module

Figure 2-6. Enterprise JavaBeans (EJB) container WB8691.0

Notes:

The following properties allow you to configure the services that the EJB container provides:

- **Passivation directory**

  This attribute provides the directory that you can use to store the persistent state of passivated, stateful session EJBs. If you are using the EJB container to manage session data, you should give WebSphere the ability to swap data to disk when necessary. This directory tells WebSphere where to hold EJB session data when it passivates and activates beans from the pool.

- **Inactive pool cleanup interval**

  Because WebSphere builds a pool of EJBs to satisfy incoming requests, you must tell it when to remove beans from this pool to preserve resources. This attribute allows you to define the interval at which the container examines the pools of available bean instances to determine whether some instances can be deleted to reduce memory usage.

- **Default data source JNDI name**

  Here you can set a default data source to use for EJBs that have no individual data source defined. This setting is not applicable for EJB-compliant CMP beans.

- **Initial state**

  This attribute allows you to identify the state of the container when WebSphere is started. If you must recycle the application server, this attribute is used to determine whether to start the EJB container at server startup. You would set it to stopped only if you planned on never using the EJB container or EJBs within that specific application server instance.

- **EJB cache settings**

  You can set up two types of cache settings in WebSphere:

  - **Cleanup interval**: This attribute allows you to set the interval at which the container attempts to remove unused items from the cache. It reduces the total number of items in cache to the value you set in the cache size attribute.
  - **Cache size**: This attribute specifies the number of buckets in the active instance list within the EJB container. WebSphere uses this attribute to determine how large the cache is and when to remove components from the cache to reduce its size.

Java EE Connector Architecture service (JCA)

- The JCA Connection Manager administers:
  - Connections that are obtained through resource adapters that the JCA defines
  - Data sources that the JDBC 2.0 Extensions define

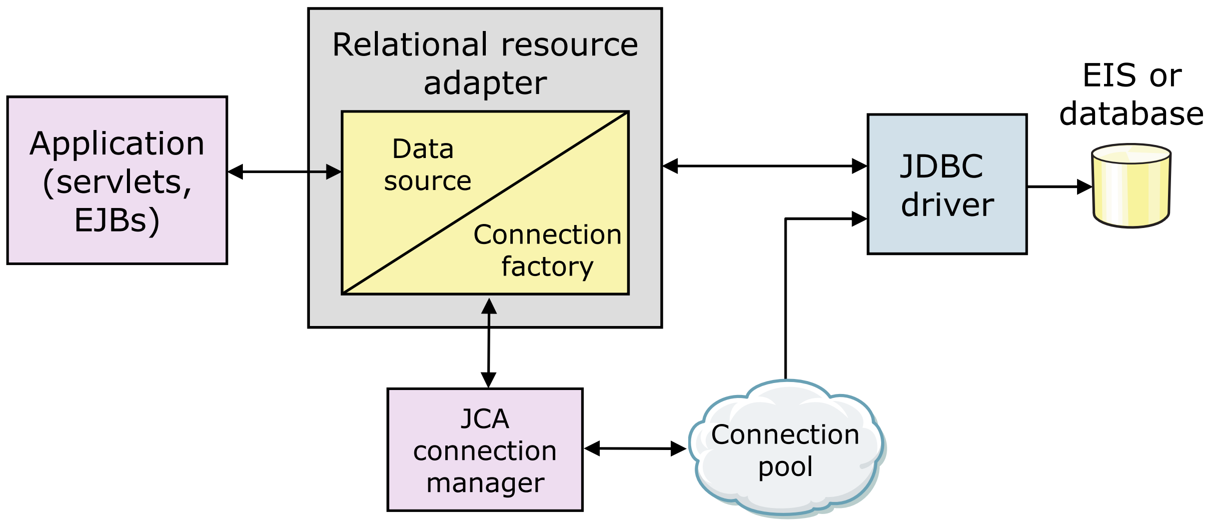


Figure 2-7. Java EE Connector Architecture service (JCA) WB8691.0

Notes:

Connection management for access to enterprise information systems (EIS) in WebSphere Application Server is based on the Java EE Connector Architecture (JCA) specification, also sometimes referred to as J2C. The connection between the enterprise application and the EIS is made by using EIS-provided resource adapters, which are plugged into the application server. The architecture specifies the connection management, transaction management, and security contracts that exist between the application server and the EIS.

Within the application server, the connection manager pools and manages connections. The connection manager administers connections that are obtained through both resource adapters that are defined according to the JCA specification and sources that are defined according to the JDBC 2.0 extensions, and later, specification.

The JCA connection manager provides the connection pooling, local transaction, and security supports. The relational resource adapter provides the JDBC wrappers and JCA CCI implementation that allow applications that use bean-managed persistence, JDBC calls, and container-managed persistence beans to access the database JDBC driver.

The JCA resource adapter is a system-level software driver that EIS vendors or other third-party vendors supply. It provides the connectivity between Java EE components (an application server or an application client) and an EIS.

One resource adapter, the WebSphere Relational Resource adapter, is predefined for handling data access to relational databases. This resource adapter provides data access through JDBC calls to access databases dynamically. It provides connection pooling, local transaction, and security support. The WebSphere persistence manager uses this adapter to access data for container-managed persistence beans.

WebSphere Education

Transaction service

- WebSphere applications use transactions to coordinate multiple updates to resources as one unit of work
- The application or the container starts and ends transactions; hence transactions can be configured as either:
  - Container-managed
  - Bean-managed
- WebSphere Application Server is a transaction manager that supports the coordination of resource managers through the XAResource interface and participates in distributed global transactions
- The transaction service writes information to the **transaction log** for every global transaction that involves two or more resources, or that is distributed across multiple servers
- You can also configure WebSphere applications to interact with:
  - Databases
  - Java Message Service (JMS) queues
  - JCA connectors
  - EJBs in other application servers

Figure 2-8. Transaction service WB8691.0

Notes:

How applications use transactions depends on the type of application component, for example:

- A session bean can use either container-managed transactions where the bean delegates management of transactions to the container, or bean-managed transactions where the bean manages transactions itself.
- Entity beans use container-managed transactions.
- Web components, or servlets, use bean-managed transactions.

WebSphere Education

Name service

- Each application server hosts a name service that provides a Java Naming and Directory Interface (JNDI) namespace
- Registers all EJB and Java EE resources that the application server hosts, including:
  - JDBC providers
  - JMS destinations
  - JCA (J2C) components
  - URL providers
  - JavaMail providers
- The deployment manager and all node agents host a name service
- Configured bindings can map resources to remote locations

Figure 2-9. Name service WB8691.0

Notes:

The naming service is used to register resources that the application server hosts. The JNDI implementation in WebSphere Application Server is built on top of a Common Object Request Broker Architecture (CORBA) naming service (CosNaming).

JNDI provides the client-side access to naming and presents the programming model that application developers use. CosNaming provides the server-side implementation and is where the namespace is stored. JNDI essentially provides a client-side wrapper of the namespace that is stored in CosNaming and interacts with the CosNaming server on behalf of the client.

Clients of WebSphere applications use the naming architecture to obtain references to objects related to those applications. These objects are bound into a mostly hierarchical structure, referred to as a namespace. The namespace structure consists of a set of name bindings, each containing a name relative to a specific context and the object that is bound with that name. The namespace can be accessed and manipulated through a name server.

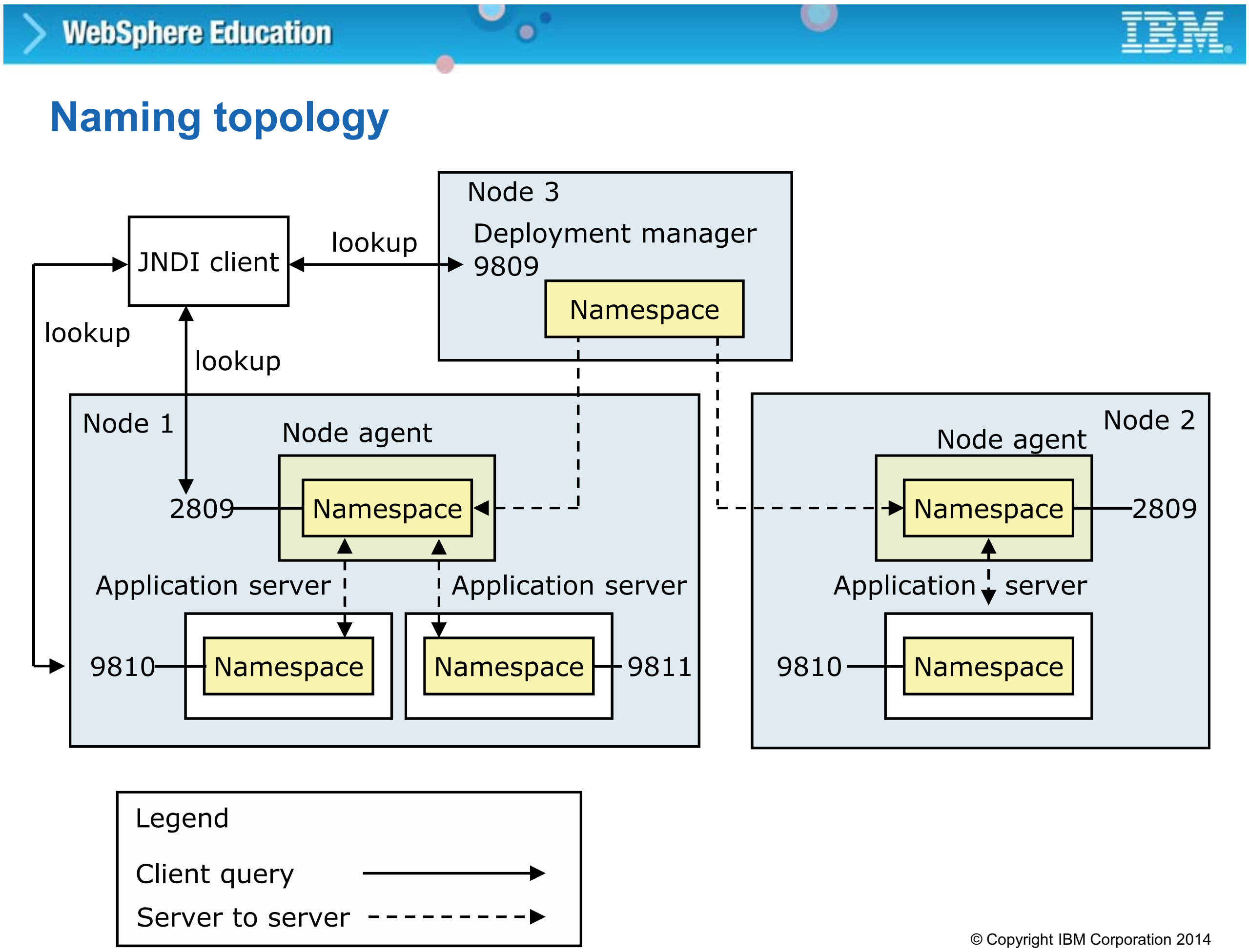


Figure 2-10. Naming topology WB8691.0

Notes:

- **Distributed namespace**

  For more scalability, the namespace for a cell is distributed among the various servers. The deployment manager, node agent, and application server processes all host a name server. The default initial context for a server is its server root. System artifacts, such as EJB homes and resources, are bound to the server root of the server with which they are associated.

- **Transient and persistent partitions**

  The namespace is partitioned into *transient* areas and *persistent* areas.

  Server roots are transient. System-bound artifacts such as EJB homes and resources are bound under server roots. A cell-persistent root is used for cell-scoped persistent bindings, and a node-persistent root is used to bind objects with a node scope.

- **Federated namespace structure**

  A *namespace* is a collection of all names that are bound to a particular name server. A namespace can contain naming context bindings to contexts in other servers. If it does, then the namespace is said to be a *federated* namespace, because it is a collection of namespaces from multiple servers. The namespaces link together to cooperatively form a single logical namespace. In a federated namespace, the real location of each context is not apparent to client applications. Clients have no knowledge that multiple name servers are handling resolution requests for a particular requested object.

In a Network Deployment distributed server configuration, the namespace for the cell is federated among the deployment manager, node agents, and application servers of the cell. Each such server hosts a name server. All name servers provide the same logical view of the cell namespace, and the various server roots and persistent partitions of the namespace are interconnected through the single logical namespace.

- **Configured bindings**

  You can use the configuration graphical interface and script interfaces to configure bindings in various root contexts within the namespace. These bindings are read-only and are bound according to the system at server startup.

- **Support for CORBA Interoperable Naming Service (INS) object Uniform Resource Locator (URL)**

  WebSphere Application Server contains support for CORBA object URLs (**corbaloc** and **corbaname**) as JNDI provider URLs and lookup names.

- **dumpNameSpace tool**

  You can use the dumpNameSpace tool to view the namespace for a particular server. The dumpNameSpace tool does not dump all of the objects in the distributed namespace. It dumps only the objects that are in the local namespace of the process against which the command was run.

WebSphere default messaging

- Integrated asynchronous capabilities for WebSphere
  - Integral JMS messaging service for WebSphere Application Server
  - Fully compliant JMS 1.1 provider
  - JMS provider is the default messaging provider
- Based on service integration bus (SIBus) technology
  - Intelligent infrastructure for service-oriented integration
  - Unifies SOA, messaging, message brokering, and publish/subscribe
- Complements and extends WebSphere MQ and application server
- Other WebSphere family products use default messaging

Figure 2-11. WebSphere default messaging WB8691.0

Notes:

The service integration functionality within WebSphere Application Server provides a highly flexible messaging fabric that supports a service-oriented architecture with a wide spectrum of quality of service options, supported protocols, and messaging patterns. It supports both message-oriented and service-oriented applications.

SOA is an architectural style whose goal is to achieve loose coupling among interacting software agents. A service is a unit of work that a service provider does to achieve specified results for a service consumer. Both provider and consumer are roles that software agents play on behalf of their owners.

2.2. Network Deployment components

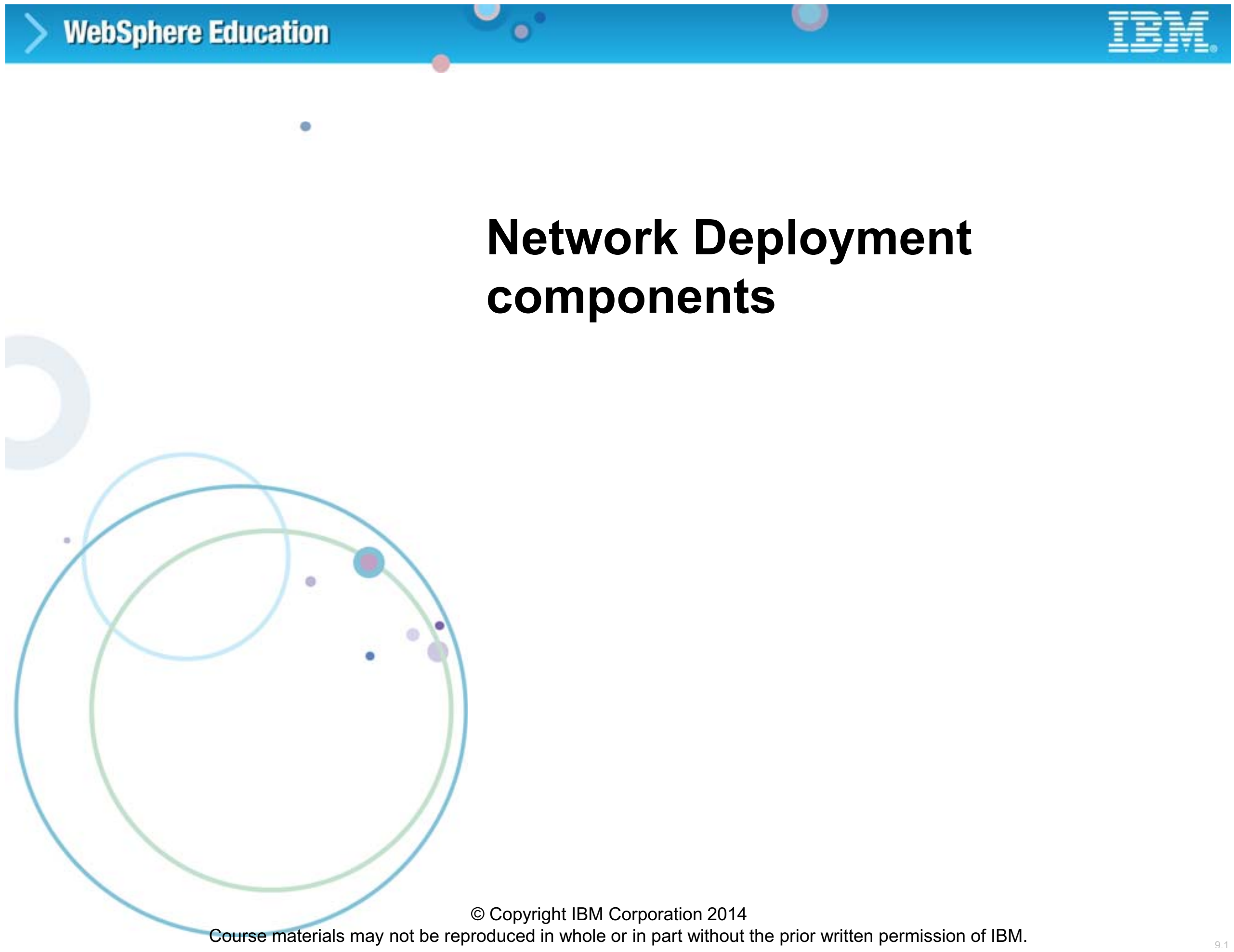


Figure 2-12. Network Deployment components WB8691.0

Notes:

WebSphere Education

Network Deployment concepts

- A **deployment manager** process manages the node agents
  - Holds the configuration repository for the entire management domain, called a **cell**
  - In a cell, the administrative console runs inside the deployment manager
- A **node** is a logical grouping of application servers
  - A single **node agent** process manages each node
  - Through profile configuration, multiple nodes can exist on a single computer

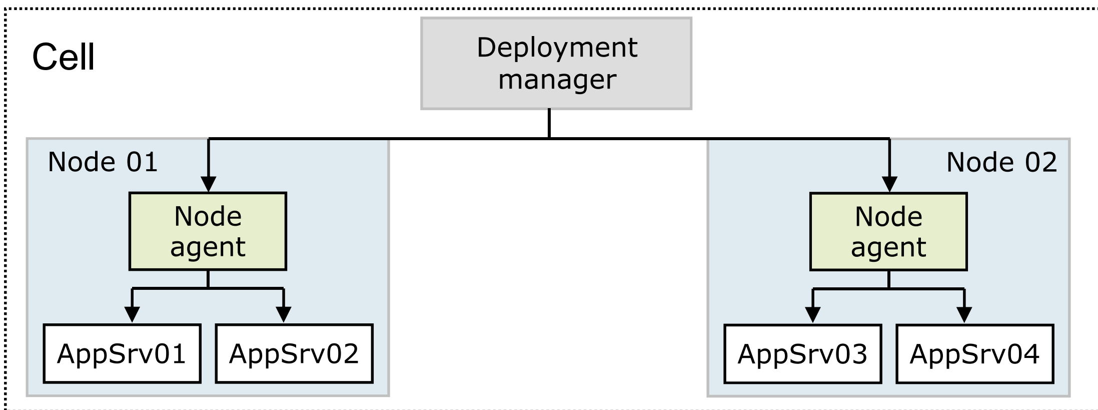


Figure 2-13. Network Deployment concepts WB8691.0

Notes:

The deployment manager here is an application server that manages the administrative environment within a cell. A profile represents a node. Multiple nodes can exist on a single machine by using profiles. The node agent is an important process that allows for communication of administrative information (commands and configuration files) to reach the application servers.

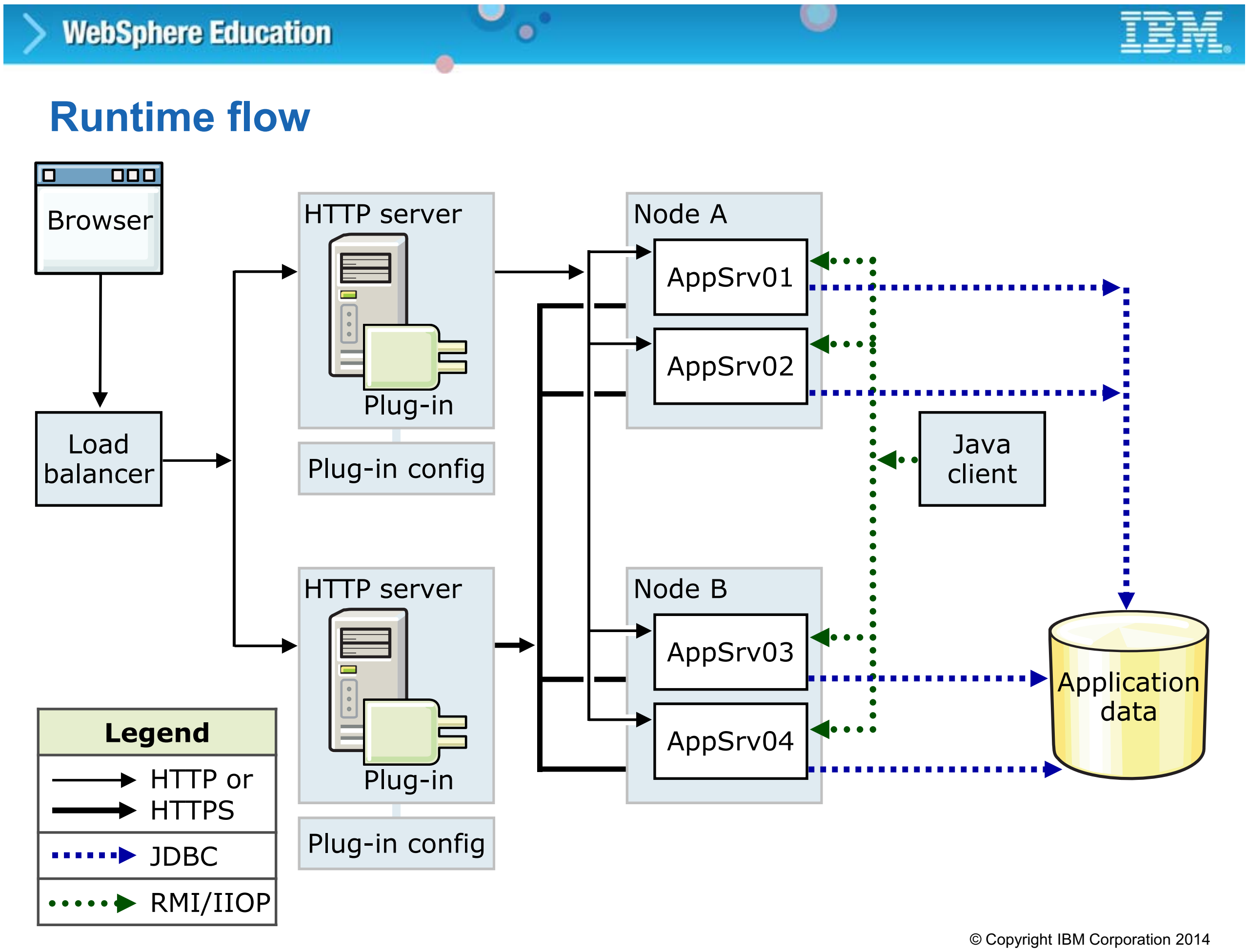


Figure 2-14. Runtime flow WB8691.0

Notes:

The main theme with Network Deployment is distributed applications. While the “flow” of an application remains the same, there are significant additions to the runtime of an application. Note the “load balancer”: it allows for multiple HTTP servers. Users point their browsers to the load balancer, and their requests are workload managed to an HTTP server. As soon as a request reaches one of these HTTP servers, the HTTP server plug-in load balances the request between the application servers that it is configured to serve. When the request enters the application server, the flow is identical to how it was in Express and Base.

The Java client’s requests to EJBs can also be workloads that are managed so that the requests do not all go to one application server.

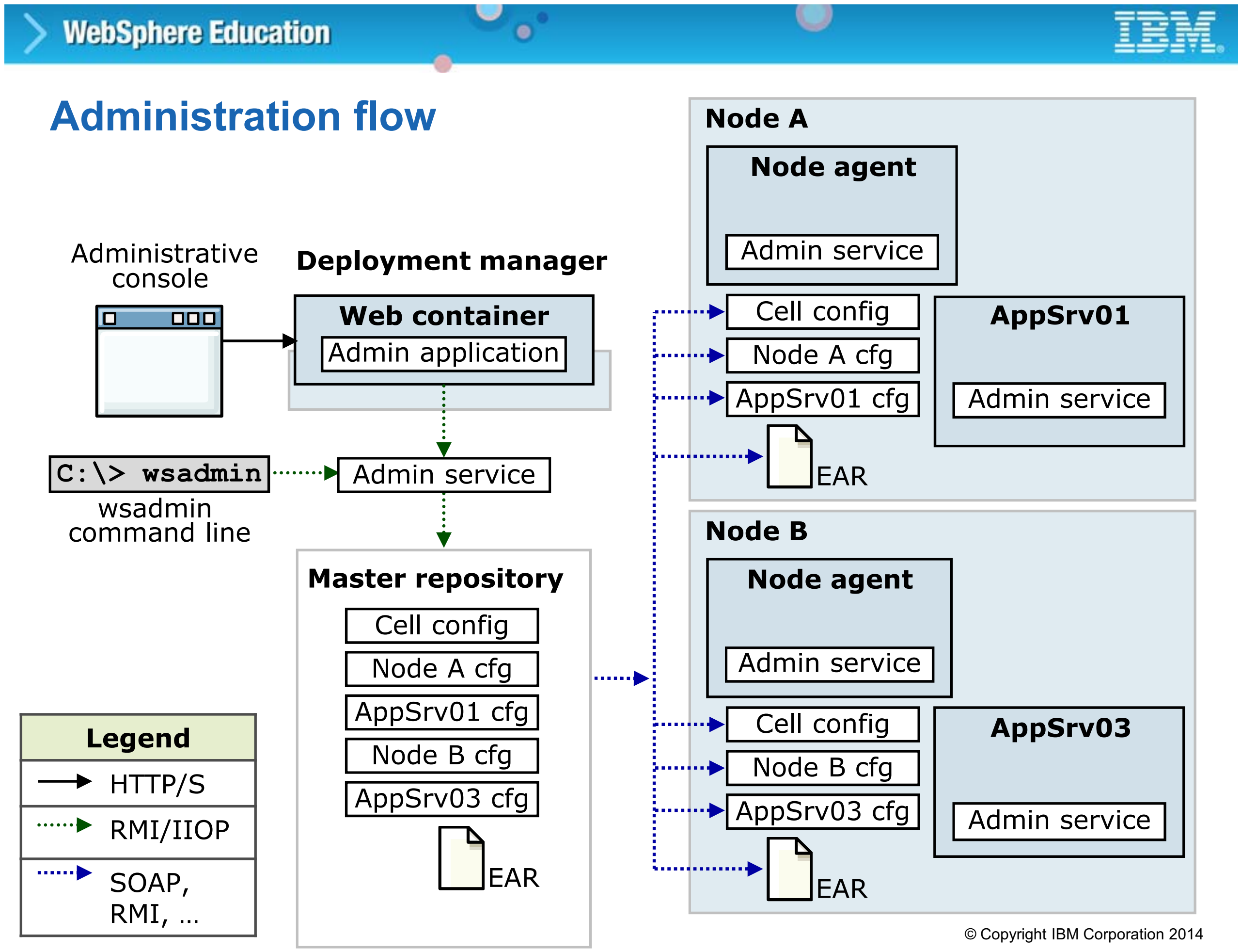


Figure 2-15. Administration flow WB8691.0

Notes:

The administrative console and wsadmin are still the two ways that the environment is administered. However, take note that these tools now communicate to the deployment manager and not to the application servers directly. The communication of these commands flows from the tools to the deployment manager to the node agents, to the application servers. This communication flow allows administration of multiple nodes (each possibly containing multiple application servers) from a single focal point (the deployment manager).

There is one main repository for the configuration files within a cell that are associated with the deployment manager. All updates to the configuration files should go through the deployment manager. You are going to see in a moment how this process works. Be careful in connecting to an application server directly with wsadmin or the administrative console, as any changes that are made to the configuration files are only temporary. The configuration files overwrite them from the master files.

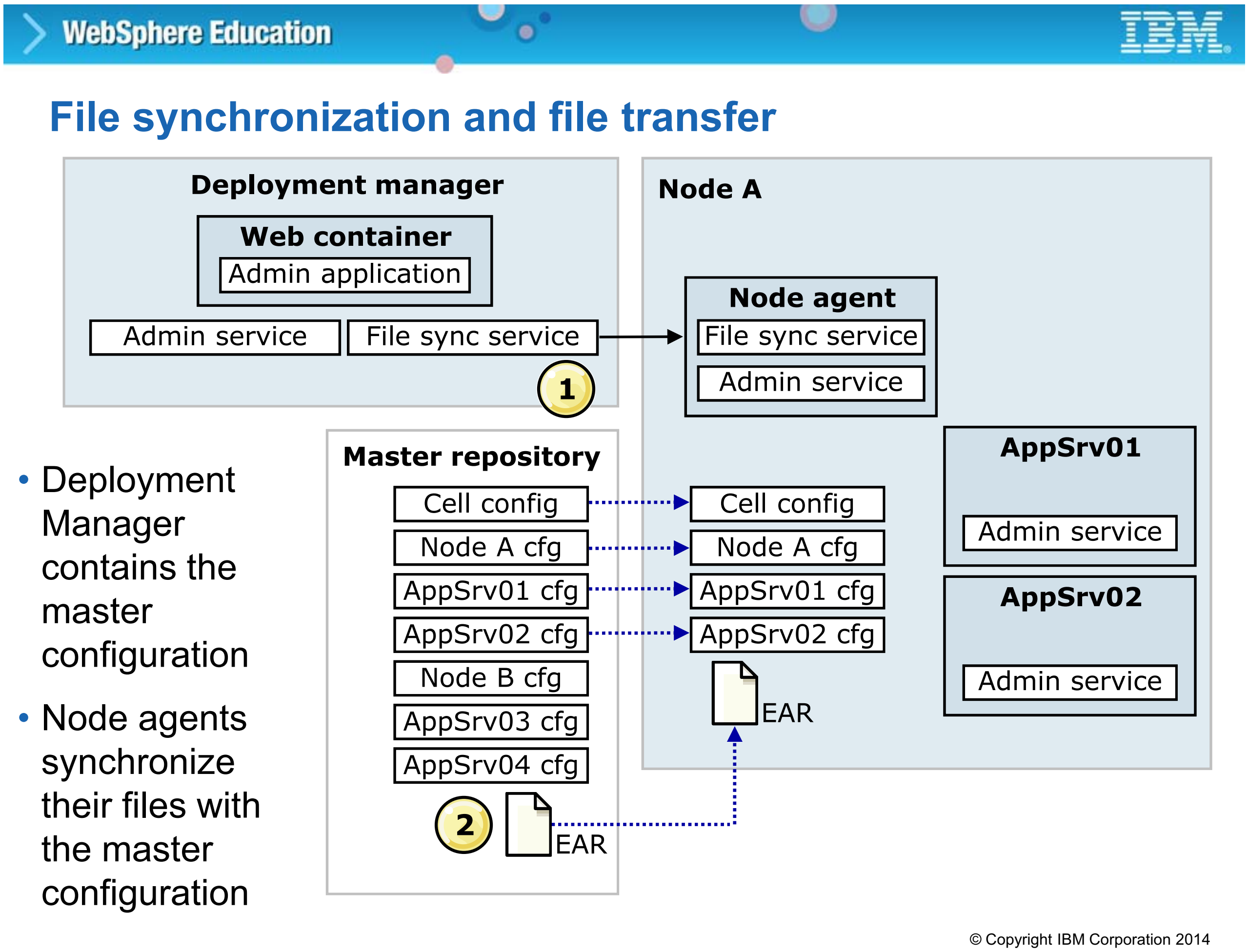


Figure 2-16. File synchronization and file transfer WB8691.0

Notes:

File synchronization service

Node agents synchronize their files with the “master” configuration (repository) as follows:

- Automatically at startup
- Periodically through configuration
- Manually from the administrative console or command line

During synchronization:

- The node agent checks for changes to master configuration.
- New or updated files are copied to the node.

This administrative service is responsible for keeping up-to-date the configuration and application data files that are distributed across the cell. The service runs in the deployment manager and node agents, and ensures that changes made to the master repository are propagated out to the nodes, as necessary. The file transfer system application is used for the synchronization process. File synchronization can be forced from an administration client, or can be scheduled to happen automatically. During the synchronization operation, the node agent checks with the deployment manager to see whether any files that apply to the node are updated in the master repository. New or updated files are sent to the node, while any deleted files are also deleted from the node. Synchronization is one-way. The changes are sent from the deployment manager to the node agent. No changes are sent from the node agent back to the deployment manager.

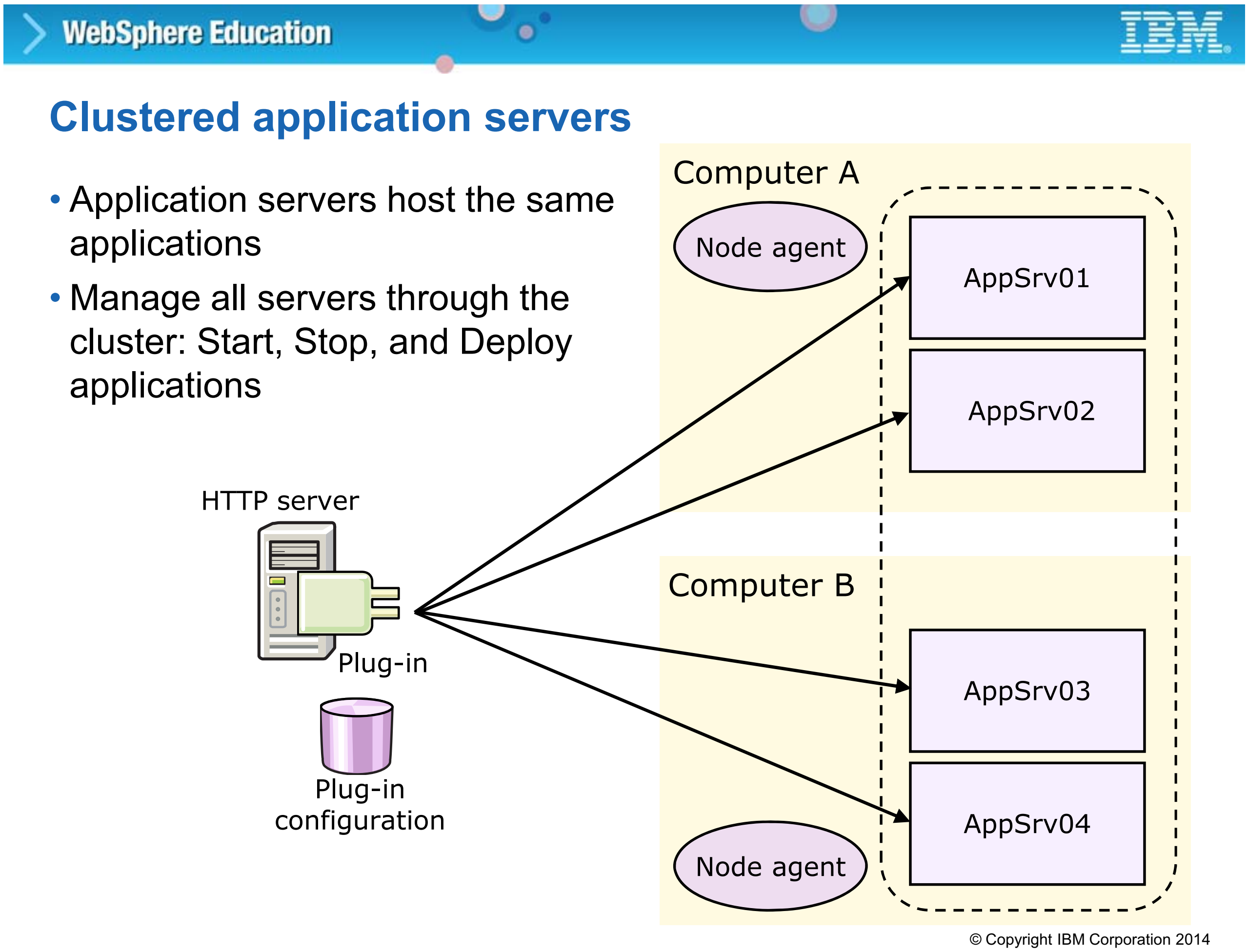


Figure 2-17. Clustered application servers WB8691.0

Notes:

A cluster of application servers provides scalability, throughput, and failover. The plug-in routes web requests for the HTTP server to servers in the cluster by using a weighted or random round-robin algorithm. The plug-in configuration file contains information about clusters, members of the cluster (clones), and the applications that the servers in the cluster host. In addition, each cluster member has a weight that is stored in the configuration file. The plug-in uses this weight in its routing algorithms to determine what percentage of incoming requests are routed to each cluster member.

WebSphere Education

Liberty profile

- The Liberty profile provides a lightweight server with a small memory footprint
- Using WebSphere Application Server Developer Tools for Eclipse, you can create a server configuration quickly
- The simple and flexible configuration is stored in the `server.xml` file and contains the configuration for the runtime instance
- You can edit the `server.xml` file directly by using an XML editor or an Eclipse-based editor
  - The configuration lists the features (capabilities or bundles) that are installed in the server
  - By defining just the features that you need, the Liberty profile provides the smallest runtime footprint for applications
- This dynamic runtime allows the adding of features and updating of configuration parameters without requiring you to restart the server

Figure 2-18. Liberty profile WB8691.0

Notes:

The Liberty profile provides a lightweight server with a small memory footprint. Using WebSphere Application Server Developer Tools for Eclipse, you can create a server configuration quickly with just a couple of mouse clicks. The simple and flexible configuration is stored in the `server.xml` file and contains the configuration for the runtime instance. You can edit the `server.xml` file directly by using an XML editor or an Eclipse-based editor. The configuration lists the features (capabilities or bundles) that are installed in the server. By defining just the features that you need, the Liberty profile provides the smallest runtime footprint for applications. This dynamic runtime allows the adding of features and updating of configuration parameters without requiring you to restart the server.

WebSphere Education

Liberty profile: Composable runtime

- Full profile includes everything
- Liberty profile includes only those features you add
  - Improved performance
  - Faster server starts
- **Note:** Problem determination for Liberty is not covered in this course

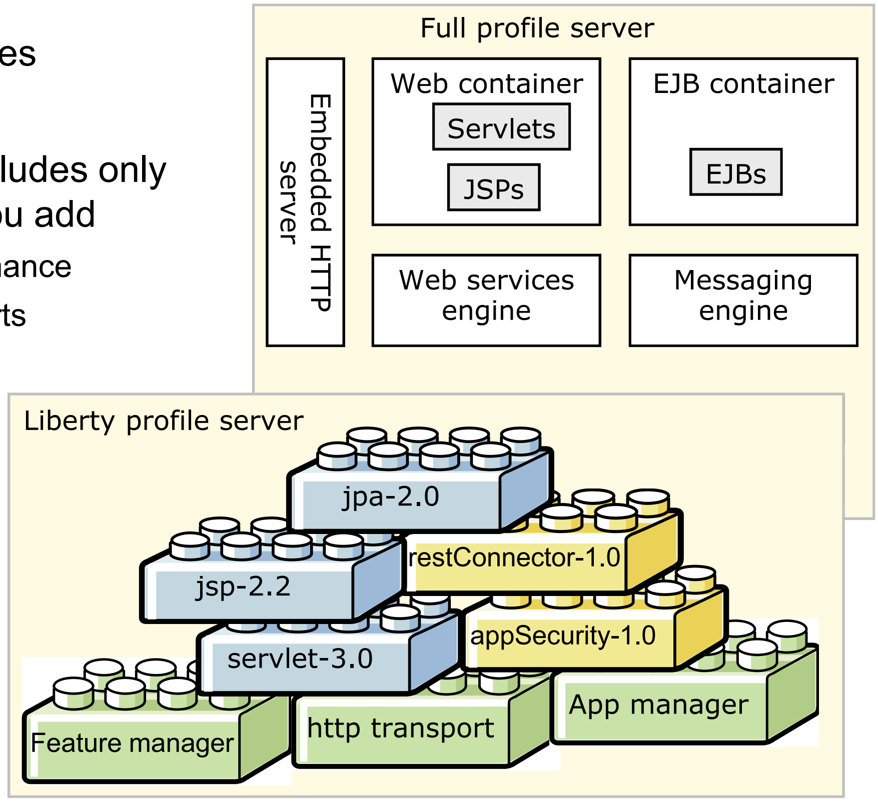


Figure 2-19. Liberty profile: Composable runtime WB8691.0

Notes:

Application servers in the full profile have a JVM shown on the left that includes all services (features) whether the applications require them or not.

The Liberty profile server JVM starts only those features that you add to its `server.xml` file. By default, a server contains only the jsp-2.2 feature to support servlet and JSP applications. You use the feature manager to add the features that you need.

The building blocks shown in the graphic on this slide show some of the Liberty profile features that can be defined for the feature manager. These features include JPA, JSP, servlet, application security, and a remote JMX connector. However, several other features can be configured for a particular server. In addition to the features, you can add HTTP port definitions for the HTTP transport, and application definitions for the application manager.

2.3. IBM Business Process Manager and Process Center

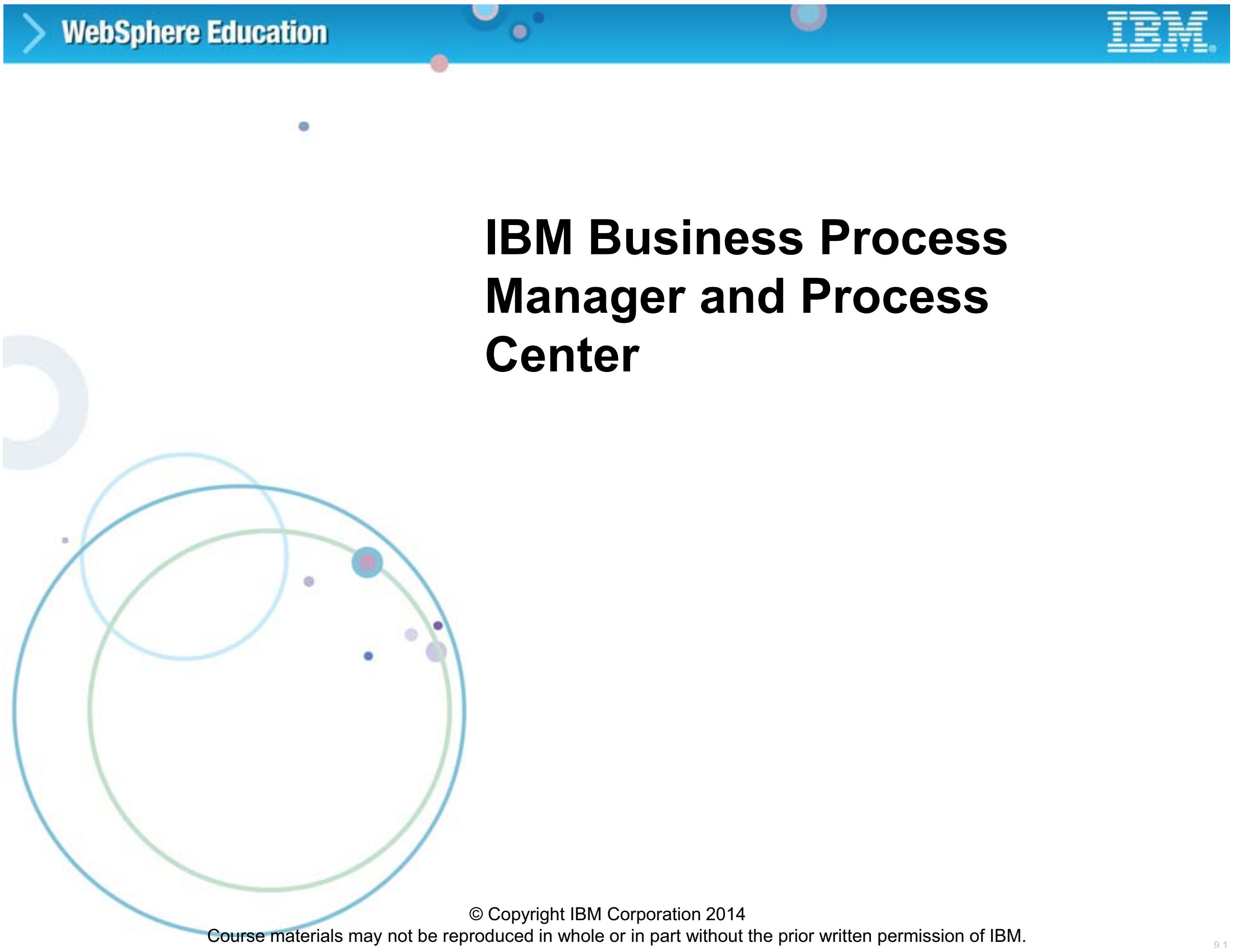


Figure 2-20. IBM Business Process Manager and Process Center WB8691.0

Notes:

WebSphere Education IBM

IBM Business Process Manager V8.5 (1 of 2)

- 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

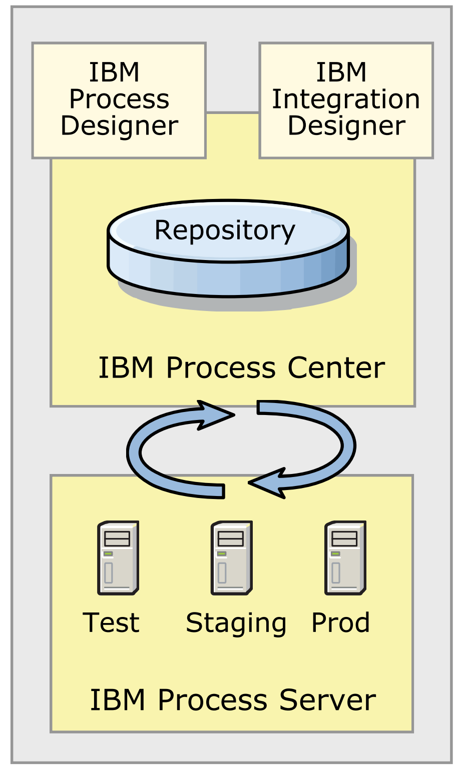


Figure 2-21. IBM Business Process Manager V8.5 (1 of 2) WB8691.0

Notes:

IBM Business Process Manager is a suite of products that includes a number of tools for working with business processes. These tools are useful for modeling, designing, implementing, and deploying business processes. They include IBM Process Designer, IBM Integration Designer, IBM Process Center, and IBM Process Server. To another degree, Blueworks Live is a utility in the cloud that can be used for elementary modeling of business processes, which can then be imported and designed in IBM Process Designer.

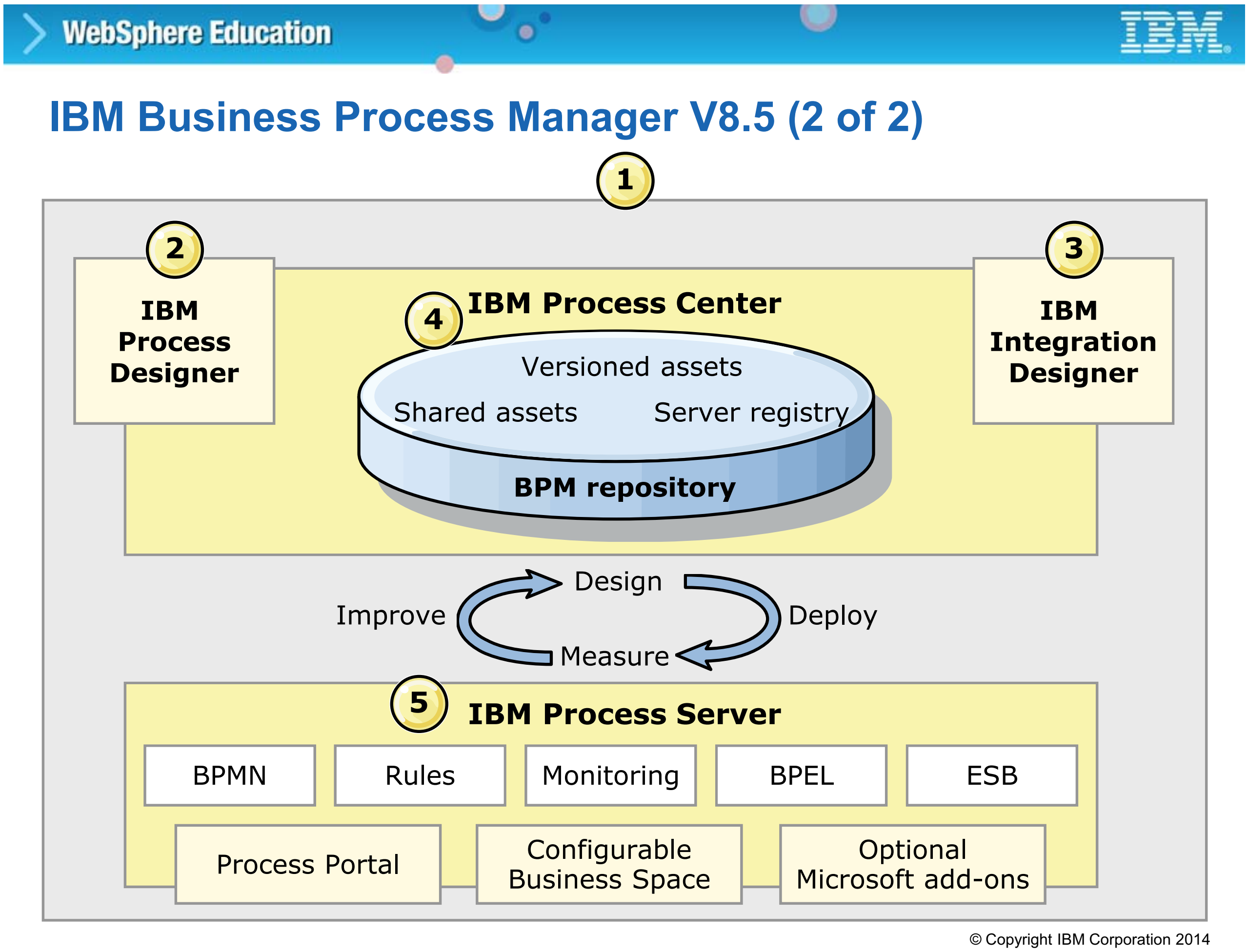


Figure 2-22. IBM Business Process Manager V8.5 (2 of 2) WB8691.0

Notes:

The tools of IBM Business Process Manager (1) advanced include IBM Process Designer, IBM Integration Designer, IBM Process Center, and IBM Process Server. IBM Process Designer (2) and IBM Integration Designer (3) are authoring environments for creating process applications, services, and implementations of activities in business process applications. IBM Process Center (4) offers a repository wherein development teams might store version-controlled or shared assets, and keep a record of them in a server registry. Eventually, these process applications can be deployed to the IBM Process Server (5), a runtime environment that supports several different implementation options (such as BPMN, business rules, BPEL, and ESB). IBM Process Server also offers the ability to integrate with other systems, with process portals, business spaces, or other add-ons.

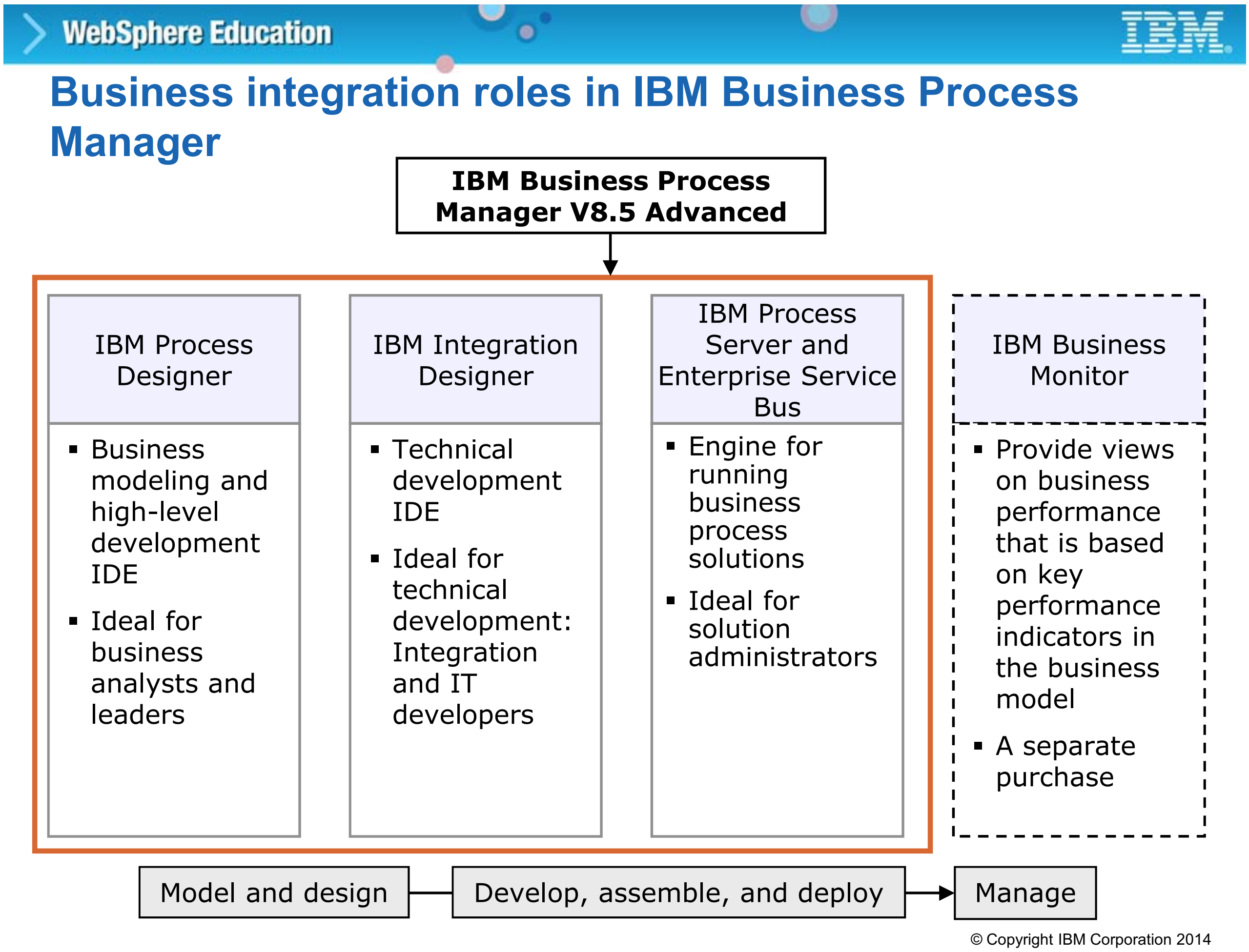


Figure 2-23. Business integration roles in IBM Business Process Manager WB8691.0

Notes:

IBM Business Process Manager consists of a number of tools, such as:

- IBM Process Designer
- IBM Integration Designer
- IBM Process Server with WebSphere Enterprise Service Bus

A specific set of roles, or a "team", uses each of these tools in each stage of the development process. Business analysts and project team leaders who are part of the "modeling team" can use IBM Process Designer to model and compose high-level implementations of business process definitions. Integration and IT developers, part of the "development team", can use IBM Integration Designer for the low-level, highly technical implementation and code that are required in the development and assembly phase.

IBM Process Center (which is not pictured here) can be used as a repository to store the artifacts that the modeling and development teams create. The teams share these artifacts to quickly produce solutions. Solution administrators and integration developers, part of the "deployment team", can use IBM Process Server and WebSphere Enterprise Service Bus to deploy those solutions to the production server.

IBM Business Monitor, which is purchased separately, can be used to measure performance and to produce optimization reports, which feed back into the development process.

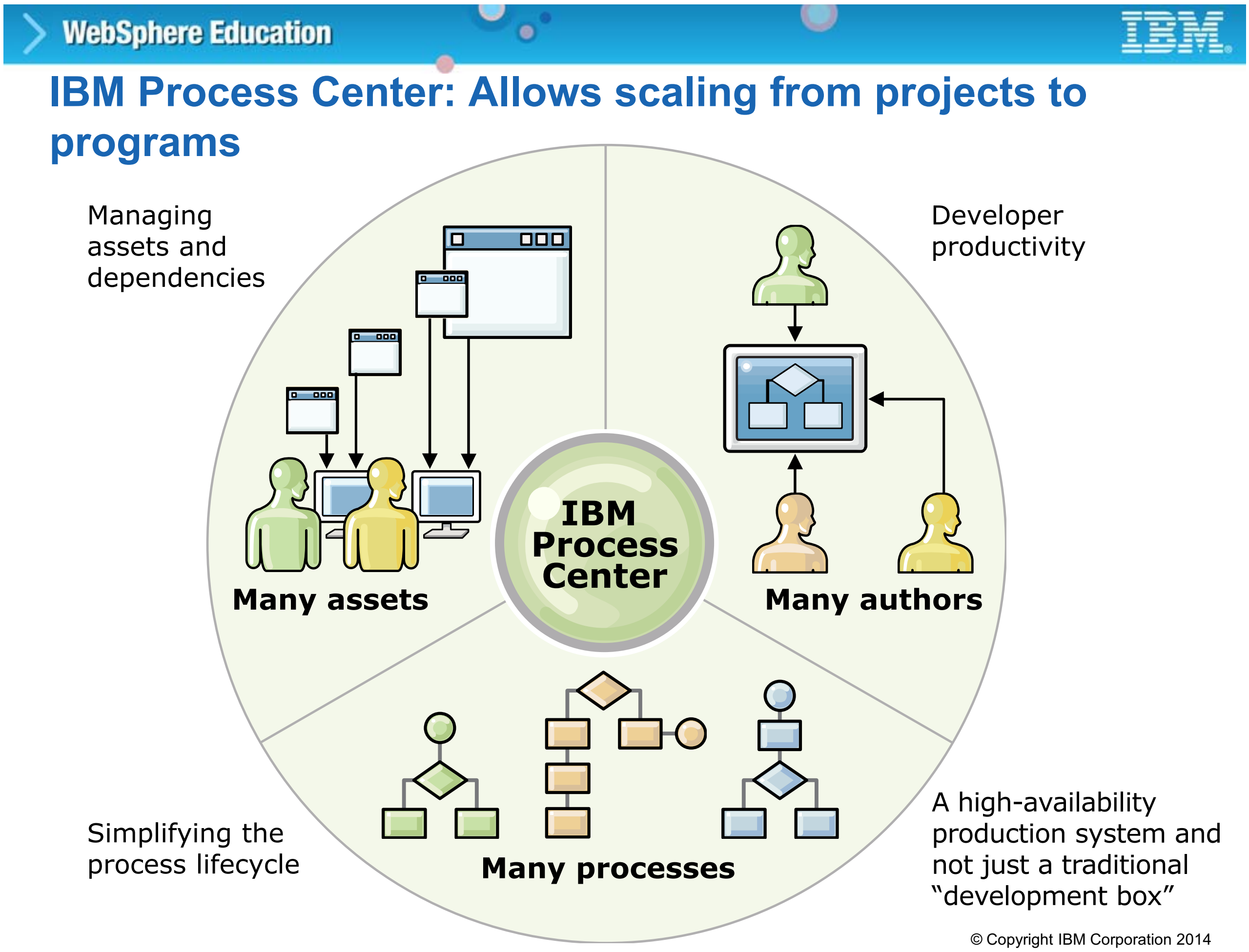


Figure 2-24. IBM Process Center: Allows scaling from projects to programs WB8691.0

Notes:

IBM Process Center includes a repository for all processes, services, and other assets that are created in the IBM Business Process Manager authoring environments (IBM Process Designer and Integration Designer).

IBM Process Center is a runtime environment where assets are shared, which in effect allows developers to build business processes cooperatively in a highly interactive manner. These business processes can use monitoring points that are created with the IBM Business Monitor development toolkit. The result is a business process that can be examined at run time for effectiveness under real working conditions. IBM Business Monitor provides a dashboard view with gauges and scorecards. You can add alerts and notifications that inform you from moment to moment on how the business process is doing. Bottlenecks, inefficiencies, and errors in resource allocation in a running business process can be spotted and corrected, resulting in improving the performance of your business processes.

The IBM Process Center console provides the tools that are needed to maintain the repository. From that console, you can create process applications and toolkits, and grant other users access to those process applications and toolkits. In the authoring environments, you can create process models, services, and other assets within process applications. IBM Process Center includes a server and a performance data warehouse, allowing users to work in the authoring environments to run processes and store performance data for testing and playback purposes.

The console also provides the following features:

- Administrators 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.
- The console provides a convenient location in which to create and maintain high-level containers, such as process applications and toolkits. Administrators who do not actively work in the Designer view can use the console to provide a framework in which BPM analysts and developers can build their processes and underlying implementations. Another primary task for administrators is managing access to the repository by setting up the appropriate authorization for users and groups.

Those users with appropriate authorization can do some administrative tasks directly in IBM Process Designer and IBM Integration Designer. For example, developers with write access to the process application might want to capture the state of all project assets at a particular milestone. They can create a snapshot while working in the Designer view of the console.

WebSphere Education IBM

Development environments

- IBM Process Designer and IBM Integration Designer are two IDEs provided with IBM Business Process Manager V8.5 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
- IBM Integration Designer is intended for **technical development** that is used for low-level business process solutions, which offer:
  - Full SOA support
  - Scalability, development power, and reliability
  - Integration with other systems and services
  - Mediation module support for deployment to the WebSphere Enterprise Service Bus

Figure 2-25. Development environments WB8691.0

Notes:

The Express or the Standard configuration of IBM Business Process Manager does not include IBM Integration Designer. All modeling and development must be done in IBM Process Designer or another tool.

IBM Business Process Manager Advanced provides two separate development environments, each of which can be used in the development of a business process solution. It is important to use both of these development environments correctly to maximize the efficiency of a project.

IBM Process Designer is intended for modeling. Modeling teams, consisting of business analysts and project team managers, use IBM Process Designer. It provides high-level business process definitions, but does not offer support for mediation or services through an enterprise service bus.

IBM Integration Designer is intended for technical development. Development teams, consisting of IT and integration developers, use IBM Integration Designer. It provides full SOA support, scalability, power, reliability, and integration with enterprise service bus.

Development and modeling teams can share the assets that they produce through IBM Process Center.

WebSphere Education

IBM Process Center Console

- The IBM Process Center Console provides a web-based interface for managing the Process Center maintained projects
- It provides the tools that are needed to maintain the repository

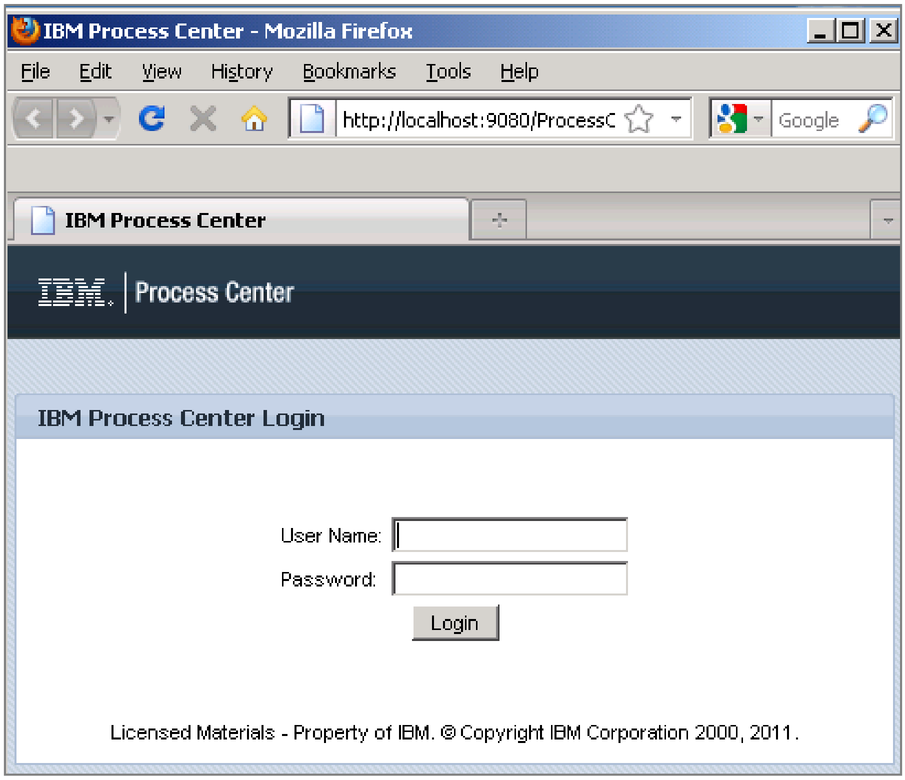


Figure 2-26. IBM Process Center Console WB8691.0

Notes:

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

The IBM Process Center includes a repository for all processes, services, and other assets. The IBM 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 install process applications that are ready for testing or production on the IBM Process Servers in those environments.
- Administrators manage running instances of process applications in configured environments.

WebSphere Education

Accessing IBM Process Center

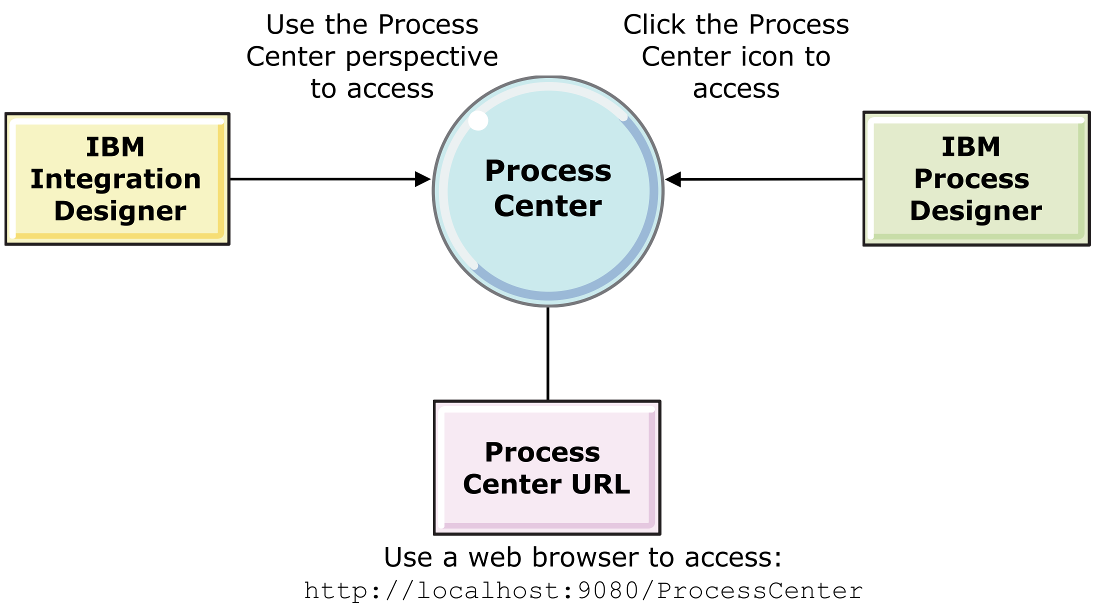


Figure 2-27. Accessing IBM Process Center WB8691.0

Notes:

IBM Process Center can be accessed in several ways:

- Switching to the Process Center perspective in the IBM Integration Designer
- Clicking the IBM Process Center icon at the upper right in the IBM Process Designer
- Using a web browser at the default URL: `http://localhost:9080/ProcessCenter`

The Process Center view and capability can vary slightly, depending which tool you are using. The tool dictates the capability that depends on its function.

IBM Process Center: Process Apps

- A process application is the container for a solution
- Initially created through the Process Center console
- It is given a name and a tag that is called an *acronym*
- The process application and its artifact contents are stored within a repository that the IBM Process Center hosts and manages

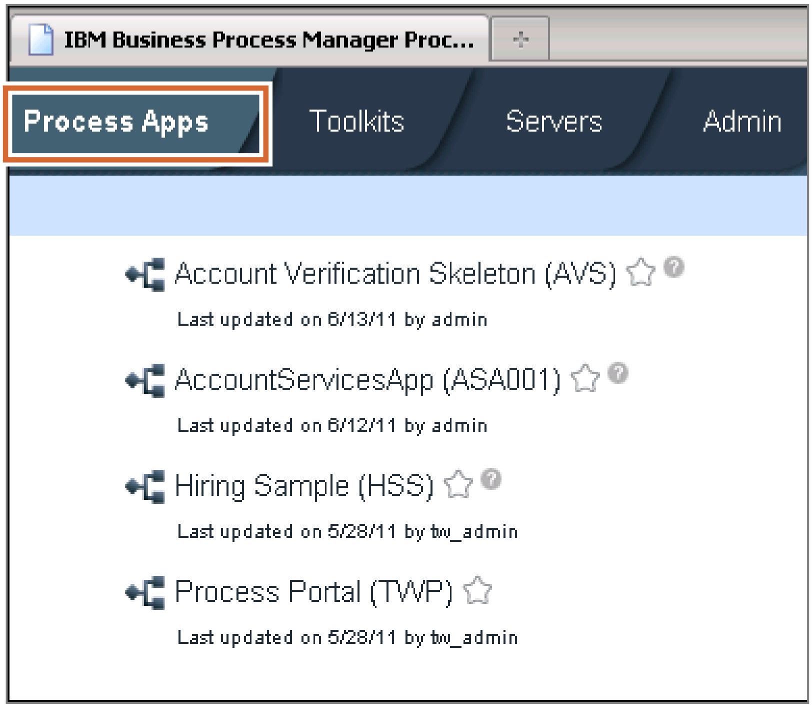


Figure 2-28. IBM Process Center: Process Apps WB8691.0

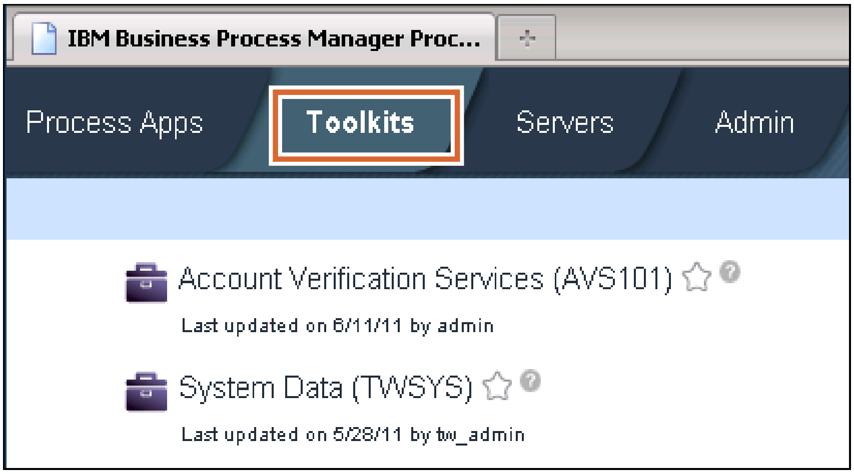
Notes:

A process application is the container for a solution. You can loosely think of it as a project. The process application is initially created through the Process Center console. It is given a name and a tag that is called an *acronym*. The acronym must be unique and can be no more than 7 characters in length. When the process application container is created, artifacts can then be further created within it using the IBM Process Designer.

The process application and its artifact contents are stored within a repository that the IBM Process Center hosts and manages. The main Process Apps page has a button to create a process application.

WebSphere Education

IBM Process Center: Toolkits



- Container for artifacts that are used in solutions
  - Does not result in a deployable application
- Can be “included” or “used” by one or more process applications
  - Similar to a library with artifacts
- Can be added as a dependency to a process application

Figure 2-29. IBM Process Center: Toolkits WB8691.0

Notes:

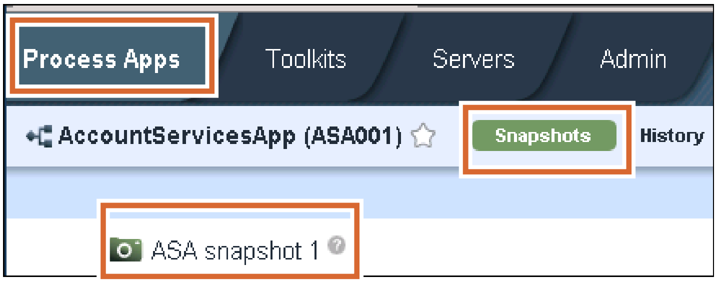
Similar to process applications, a toolkit can also be thought of as a container for artifacts that are used in solutions. Unlike a process application, a toolkit does not result in a deployable application. Instead, the contents of the toolkit can be “included” or “used” by one or more process applications.

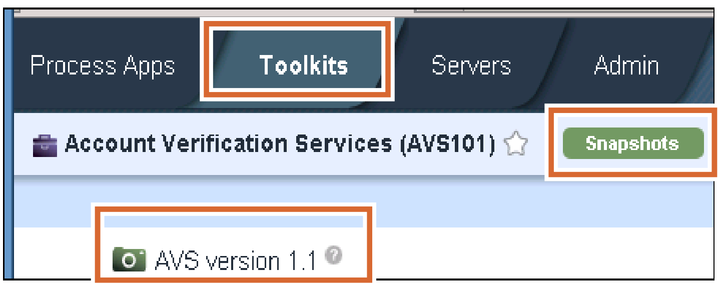
When Process Center is installed and configured, an IBM supplied toolkit that is called “System Data” is automatically imported into the repository. This toolkit is marked as read-only and is implicitly dependent on all other process applications and toolkits. It is the System Data toolkit that contains the core definitions for data structures and other items common across all process applications.

Toolkits have their own tabs in the Process Center consoles. From that point, new toolkits can be created or exported, and otherwise managed in a similar fashion to the ones of the process applications.

WebSphere Education

IBM Process Center: Snapshots





- A snapshot is a copy of the state of all the artifacts in a process application or toolkit at the point in time when the snapshot was made
- Allows users to revert in time to the state of the snapshot
- Allows for creating a version of toolkits and process applications

Figure 2-30. IBM Process Center: Snapshots WB8691.0

Notes:

A snapshot is a copy of the state of all the artifacts in a process application or toolkit at the point in time when the snapshot was made. The purpose of taking a snapshot is to allow you to revert in time to the state of the snapshot that is needed. A snapshot can be captured by clicking the Snapshot icon in IBM Process Designer.

A snapshot is required in some circumstances, for example:

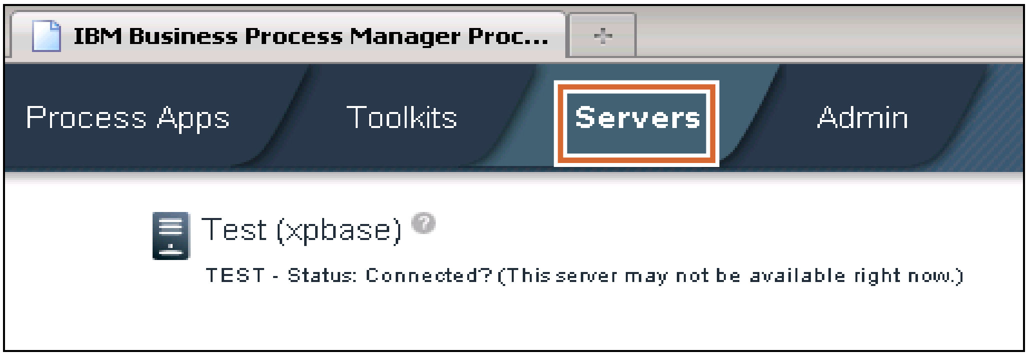
- A snapshot of a toolkit is required before it can be added as a dependency on other toolkits or process applications.
- A snapshot of a process application is required before that application can be installed on IBM Process Server.
- A snapshot is required before a new “workspace” can be created.

Just like process applications, toolkits can have snapshots that are taken of them, allowing all the artifacts in a toolkit to be version-controlled.

To add a toolkit as a dependency to a process application, the toolkit must first have a snapshot that is associated with it. This requirement is because the dependency added to the process application is **not** just the name of the toolkit, but is instead a specific snapshot of that toolkit.

WebSphere Education

IBM Process Center: Servers



- The Servers tab lists the IBM Process Servers that are connected to the IBM Process Center
  - IBM Process Server can be a stand-alone server
  - IBM Process Server can be a server that is running inside the IBM Integration Designer test environment
  - Multiple servers can be connected
  - Multiple environments can be connected: development, testing, staging, and production

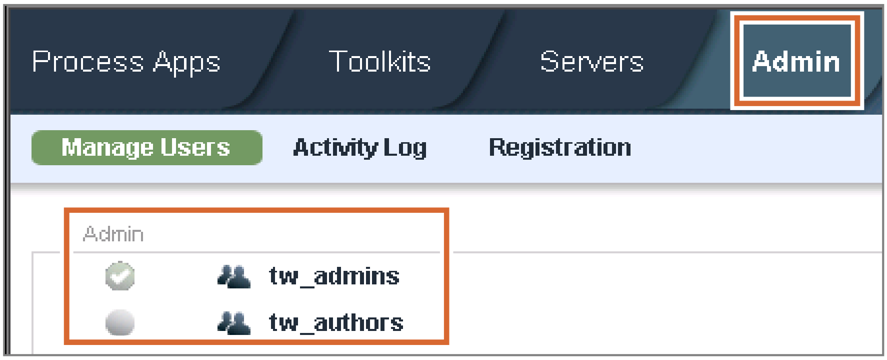
Figure 2-31. IBM Process Center: Servers WB8691.0

Notes:

The servers that are shown are the IBM Process Servers that are connected to the IBM Process Center. Authorized users can install snapshots of process applications on connected IBM Process Servers. For each server, you can view the snapshots that are currently installed.

WebSphere Education

IBM Process Center: Admin



- Add new users and groups to the list of authorized users
  - Granting users authority to access the repository allows them to log in
- Process Apps and Toolkits are individually controlled with their own access control

Figure 2-32. IBM Process Center: Admin WB8691.0

Notes:

Granting users authority to access the repository allows them to log in to the IBM Process Center console. However, this access does not give privileges to work on or even see all the process applications in the environment. Process applications and toolkits are individually controlled with their own access control lists. From within the Process Apps section of the IBM Process Designer or the IBM Process Center console, an application can be selected and the **Manage** tab clicked. In that tab, there is a section with which users and groups can be associated. These groups define the permissions for those entities.

There are three roles a user or group can have:

- Read: This role allows a user or group to see the project and see the artifacts within it. The read role cannot be removed without removing the user or group association completely. If a user or group is **not** associated with a process application, then the user or group has no authorities on that application. If an artifact is opened and the user has read authority only, the artifact is flagged as read-only in the editor.
- Write: This role allows the user or group to update or add artifacts into the process application.
- Admin: This role allows the user or group to administer the process application.

2.4. Business value of IBM Integration Designer

WebSphere Education

Business value of IBM Integration Designer

Figure 2-33. Business value of IBM Integration Designer WB8691.0

Notes:

WebSphere Education

IBM Integration Designer: Create, assemble, and test applications

- IBM Integration Designer is the unified development tool for building SOA-based integration applications for IBM Process Server, IBM Process Center, and WebSphere Enterprise Service Bus
  - Visual development environment that requires minimal programming skill
  - Provides prebuilt mediation functions and BPEL activities
  - A comprehensive environment for developing, assembling, testing, deploying, and managing integration modules and mediation modules for run time

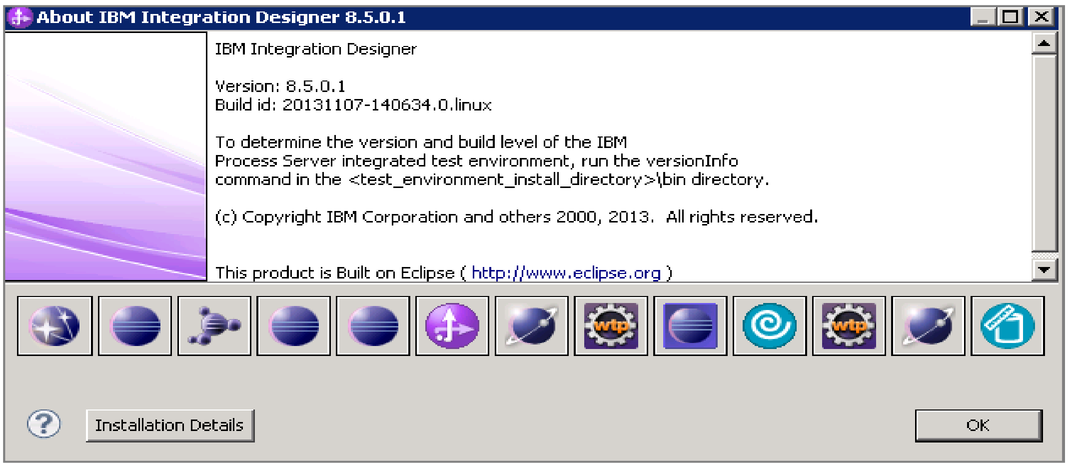


Figure 2-34. IBM Integration Designer: Create, assemble, and test applications WB8691.0

Notes:

IBM Integration Designer is the common tool for building SOA-based integration solutions across many of the IBM Business Process Manager runtime environments: IBM Process Server, IBM Process Center, WebSphere Enterprise Service Bus, and IBM Business Monitor. It simplifies integration with rich features that accelerate the adoption of SOA by rendering existing IT assets as service components, encouraging reuse, and efficiency.

IBM Integration Designer allows integration developers to assemble complex business solutions that require minimal skills, whether they involve processes, mediations, adapters, or code components. Users can construct process and integration solutions by using “drag-and-drop” technology without having a working knowledge of Java.

In addition to providing the tools necessary to build and assemble these artifacts, the product includes a full test framework. This framework allows the tests to be completed in a seamless fashion in an environment identical to a production environment, but without having to do the steps to administer and configure a production environment.

WebSphere Education

IBM Integration Designer: Platform architecture

- IBM Integration Designer is based on Rational Application Developer
  - Not all Rational Application Developer features are included
- Rational Software Development Platform provides the foundation for IBM Integration Designer and Rational Application Developer
  - Based on Eclipse V3
  - Contains the common components for Eclipse-based products
  - Installed one time per system with the first product

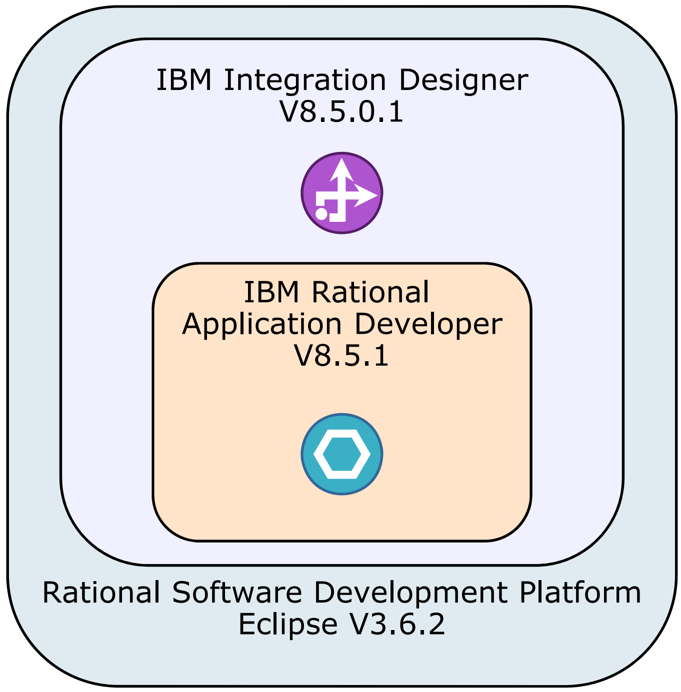


Figure 2-35. IBM Integration Designer: Platform architecture WB8691.0

Notes:

IBM Integration Designer V8.5 is built on top of Rational Application Developer V8.5.1 Eclipse-based technology. IBM Integration Designer is based on the core IBM Rational Application Developer product. However, it is important to note that there are several features not included in the core IBM Rational Application Developer product on which IBM Integration Designer is built.

Some of the features that are available in Rational Application Developer but not included in IBM Integration Designer are:

- Crystal Reports tools
- Tools for WebSphere Application Server
- Code review
- Rational Unified Process (RUP)
- Rational RequisitePro integration
- Rational Team Concert
- Rational Team Concert debug extensions

Because IBM Integration Designer is based on a subset of the full IBM Rational Application Developer tools, advanced users who need all the capabilities require a full license for IBM Rational Application Developer.

As the diagram on this slide shows, IBM Integration Designer is built upon the Rational Software Development Platform. The Rational Software Development Platform is based on Eclipse V3 technology, and each IBM product that is built upon this platform coexists and shares plug-ins with other Rational Software Development Platform-based products. The Rational Software Development Platform is installed once per system with the first product that is installed. As other products built on this platform are installed on the system, only the necessary plug-ins are installed.

The current version of Rational Software Architect and IBM Integration Designer are all based on Eclipse V3.6.2 technology; therefore shell sharing is supported. Users can switch between different perspectives to develop UML, SCA, or business models.

WebSphere Education

IBM Integration Designer roles

Role	Responsibilities
Integration developer	• 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 – Loops, conditions, string manipulation, and other programming concepts • Understands business process choreography, workflow (including human interaction), WSDL, and BPEL • Creates mediation modules to implement connectivity logic • Works with the IBM Process Center repository • Manages and deploys snapshots in the IBM Process Center
Application (IT) developer	• Is knowledgeable in one or more application development platforms (Java EE) • Understands SOA, process choreography, workflow, WSDL, and BPEL • Implements application-specific business logic for integrated solutions such as EJBs • Exposes application logic as a service

Figure 2-36. IBM Integration Designer roles WB8691.0

Notes:

The primary focal group for IBM Integration Designer is integration developers who are not required to be fluent Java programmers. Therefore, IBM Integration Designer **hides** advanced implementation details as much as possible. However, the base technology is still Rational Application Developer. Application developers can still do EJB development by using IBM Integration Designer if it is necessary (it is necessary to turn on some more capabilities).

Two primary user roles are associated with IBM Integration Designer, and a description of each is provided here:

- The **integration developer** is the primary user role that is associated with IBM Integration Designer. This user type is focused on building service-oriented solutions, and expects the authoring tools to simplify many of the advanced implementation details that are associated with building complex business applications. Although the integration developer is typically familiar with basic programming concepts (such as loops, conditions, and string manipulations), this user might not be an expert in a particular implementation platform or programming language. The purpose of the tools that are provided in IBM Integration Designer is to provide the integration developer with a development environment for building complex business applications without extensive knowledge of the underlying implementation technologies. When those business process applications are built and are ready for deployment, the integration developer is responsible to manage the process application on the IBM Process Center repository. This responsibility includes managing snapshots for the process application, deploying those snapshots, and working with toolkits.

- Another important user role for IBM Integration Designer is the **application developer**. Unlike the integration developer, the application developer is typically knowledgeable in one or more programming languages or application development platforms. The application developer also has a basic understanding of one or more technologies that are associated with building integrated business applications such as SOA, process choreography, workflow, WSDL, or BPEL. Application developers have more extensive knowledge of the implementation details that are associated with building a business application. For this reason, they are typically responsible for implementing the application-specific business logic for the actual business application that is being built. In addition, with the introduction of the SCA programming model, it is also the application developer who is responsible for exposing specific application logic as an SCA service component.

WebSphere Education

Application composition and deployment

- Use the Business Integration perspective to develop business integration projects
- Default views: Business Integration, Editor, Task Flows, Build Activities, Properties, Problems, Server Logs, and Servers

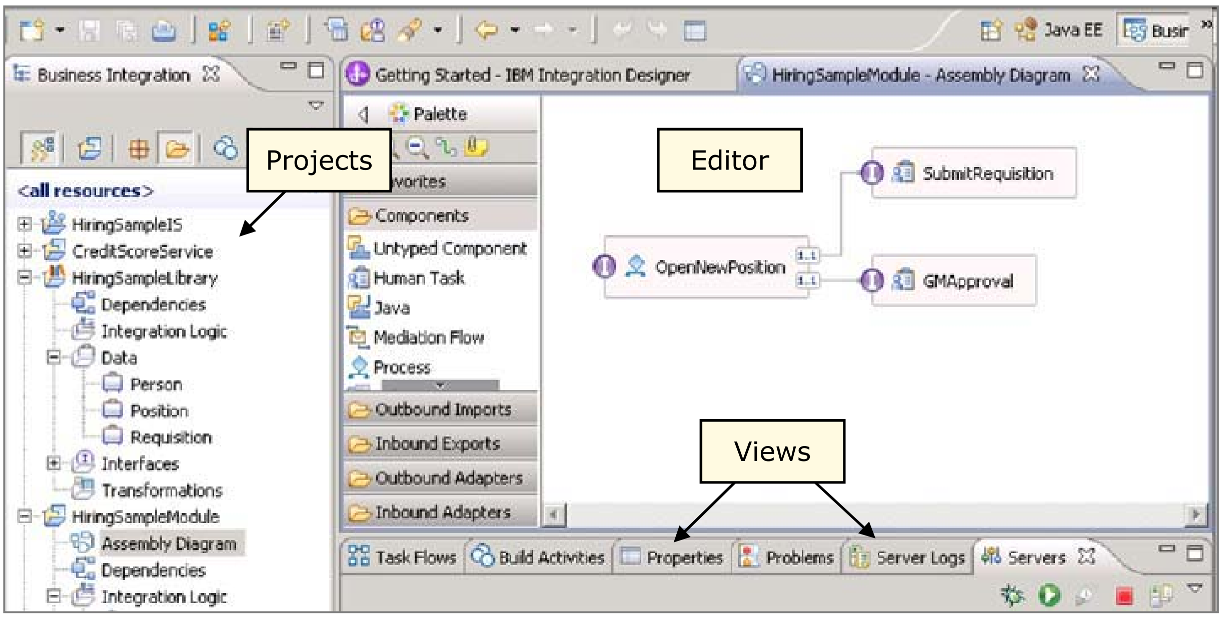


Figure 2-37. Application composition and deployment WB8691.0

Notes:

The user interface consists of several perspectives, which are groups of views for various tasks. Use the Business Integration perspective to develop business integration projects.

The Business Integration perspective is composed of the following default views:

- **Business Integration view**: The Business Integration view provides a logical view of the key resources in each module, mediation module, and library. Non-SCA projects (such as Java EE projects) are also shown in the Business Integration view in their natural form: the content and labels of the tree match the ones that are seen in the perspective. Showing non-SCA projects in the Business Integration view allows you to edit non-SCA projects without switching perspectives.
- **Editor pane**: When you open a resource from the Business Integration view with an editor, the resource is displayed in the editor pane. Each of the components that are presented in the IBM Process Server or WebSphere Enterprise Service Bus architectural diagram has an equivalent editor in IBM Integration Designer. These editors are used to create the components that form your integration and mediation modules.

- **Task Flows view**: Task flows are provided as a way to learn related concepts and tasks in an interactive manner. Learning task flows briefly describe key concepts to quickly get you up to speed with IBM Integration Designer. Creating task flows show you how to do key tasks.
- **Build Activities view**: The Build Activities view helps you manage builds and allows you to view the build and server status of projects. The view also allows you to view the operational state of supported servers.
- **Properties view**: The Properties view allows you to see detailed information about a selected artifact. When using the Properties view with editors, you are able to modify properties of elements you select.
- **Problems view**: The Problems view helps you debug errors by providing message text. You can see further help for the problem message by selecting a message and pressing F1.
- **Servers view**: Use the Servers view to manage server profiles. This view is also used for deploying applications, for starting and stopping servers (including starting in debugging mode), for creating tables and data sources, and for starting the administrative console and various clients. These clients include the BPEL Process Choreographer Explorer, Business Space, and business rules manager.
- **Server Logs view**: Use the Server Logs view to display the contents of the server console and server log files (the view automatically displays output from the test server JVM log: `SystemOut.log`). The view can also be used to load logs from other servers and to filter log messages by type.

WebSphere Education

Creating modules and libraries

- In the Business Integration perspective, you create an SCA library, a module (called a business integration module in this course), or a mediation module
- Modules and libraries contain multiple SCA artifacts that are grouped according to type
- Libraries are projects that are used to store shared resources
  - To access libraries, add them to module dependencies
- Integration modules provide the business services, and mediation modules provide the connectivity logic
  - Mediation flows and business services are modeled as SCA components
  - SCA components are wired together in the assembly diagram to form applications

Figure 2-38. Creating modules and libraries WB8691.0

Notes:

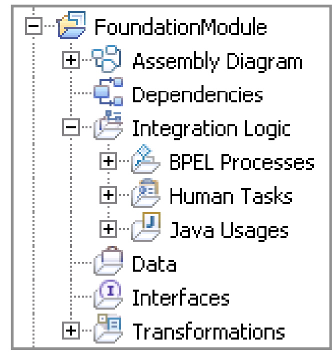
When you are working in the Java development perspective, the first thing you create is a “project.” In the Business Integration perspective you create a module, mediation module, or library.

WebSphere Education

Module components

- Business integration modules include:
  - **Assembly diagrams**: Wire SCA components together to form applications
  - **Dependencies**: Include other modules, libraries, Java EE projects, and predefined resources
  - **Integration logic**: Artifacts that do specific tasks (business processes, state machines, human tasks, business rules and rule groups, or mediation flows)
  - **Data**: Business objects
  - **Interfaces**: Service interfaces and their operations
  - **Transformations**: Data (XML) maps and relationships
- Mediation modules include:
  - **Integration logic**: *Only* mediation flows and subflows for processing messages that are passed between services
  - Artifacts present in integration modules: Assembly diagram, dependencies, data types, interfaces, and transformations

Integration module



Mediation module

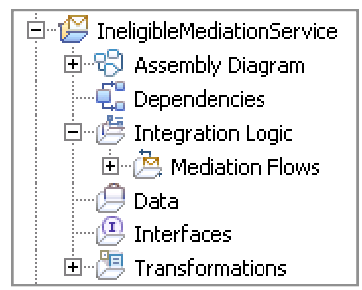


Figure 2-39. Module components WB8691.0

Notes:

In IBM Integration Designer, you can filter on the integration solution, allowing you to quickly search and find a specific module, mediation module, or library. This option requires that your modules and libraries are built into an integration solution. Similarly, you can use a filter inside a module, mediation module, or library to filter for a specific artifact.

Library components

- Libraries are project types for storing artifacts that are shared between several modules
- Libraries contain the following artifacts:
  - **Dependencies**: Are used to include other libraries and predefined resources
  - **Integration logic**: Contains artifacts that do specific tasks (mediation subflow and business calendar)
  - **Data**: Business objects and business vocabulary
  - **Interfaces**: Service interfaces and operations
  - **Transformations**: Contains data (XML) maps and relationships
- Libraries are not runnable applications
  - No assembly diagram

Library

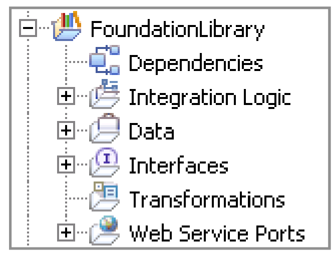


Figure 2-40. Library components WB8691.0

Notes:

A library project is a business integration project type that is used for storing artifacts that are shared between multiple modules. Unlike a module project, a library project is not a deployable unit to the IBM Process Server runtime environment. Another important difference between a module and a library project is the type of artifacts that can be contained by each project type. Specifically, library projects contain interfaces, business objects, graphs, Extensible Markup Language (XML) maps, business calendars, and relationships. Library projects do not include other types of business integration resources such as SCA components.

A library project is created to store artifacts that are shared between module projects. If a module is dependent upon a particular library project, then add that project to the dependency list for the appropriate module by using the dependency editor.

WebSphere Education IBM

Unit summary

Having completed this unit, you should be able to:

- Describe the WebSphere Application Server architecture
- Describe the Network Deployment (ND) cell components
- Identify the components of IBM Business Process Manager and Process Center
- Describe the purpose and business value of IBM Integration Designer

Figure 2-41. Unit summary WB8691.0

Notes:

WebSphere Education

Checkpoint questions

1. True or False: There are only two ways of accessing the IBM Process Center console; the first is through IBM Process Designer, and the second is through a web browser by using the Process Center URL.
2. How many logical web containers does each application server runtime have?
   a) One
   b) Two
   c) Many
   d) Zero
3. For which tool does it provide a comprehensive environment for developing, assembling, testing, deploying, and managing integration modules and mediation modules for run time?
   a) IBM Process Designer
   b) IBM Integration Designer
   c) IBM Process Center
   d) IBM Business Monitor

Figure 2-42. Checkpoint questions WB8691.0

Notes:

Write down your answers here:

1.

2.

3.

WebSphere Education

Checkpoint answers

1. False. There is a third way, by using the IBM Integration Designer perspective.

2. (a) One. Each application server runtime has one logical web container, which can be modified, but not created or removed.

3. (b) IBM Integration Designer.

Figure 2-43. Checkpoint answers WB8691.0

Notes:

Unit 3. Gathering diagnostic data

What this unit is about

This unit covers how to gather diagnostic data. It is important to identify resources for problem investigation. You learn how to locate and configure server log files. This unit also provides an overview of High Performance Extensible Logging (HPEL) for a server and how to use log viewer tools. Finally, this unit covers the usage of cross-component trace (XCT), and how to use the administrative console to examine Java virtual machine (JVM) diagnostic data.

What you should be able to do

After completing this unit, you should be able to:

- Identify and access several resources for problem investigation
- Identify, locate, and configure server log files
- Enable HPEL logging for a server and use log viewer tools
- Describe and use cross-component trace (XCT)
- Generate JVM diagnostic data by using the administrative console and other tools

How you will check your progress

- Checkpoint

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Identify and access several resources for problem investigation
- Identify, locate, and configure server log files
- Enable HPEL logging for a server and use log viewer tools
- Describe and use cross-component trace (XCT)
- Generate JVM diagnostic data by using the administrative console and other tools

Figure 3-1. Unit objectives WB8691.0

Notes:

WebSphere Education

Topics

- Resources for problem investigation
- Logs and tracing: Basic mode
- High Performance Extensible Logging (HPEL)
- Gathering JVM-related data

Figure 3-2. Topics WB8691.0

Notes:

3.1. Resources for problem investigation

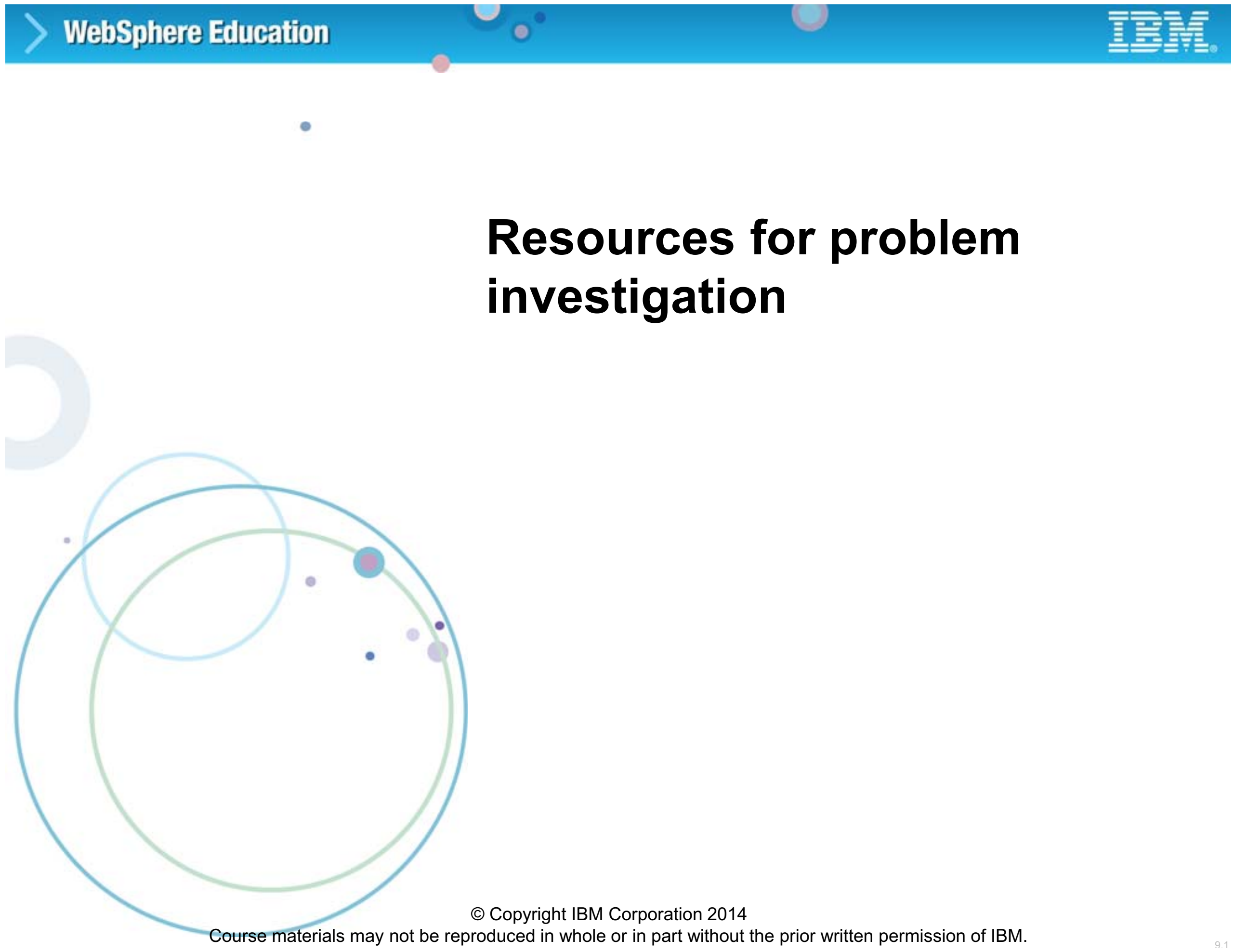


Figure 3-3. Resources for problem investigation WB8691.0

Notes:

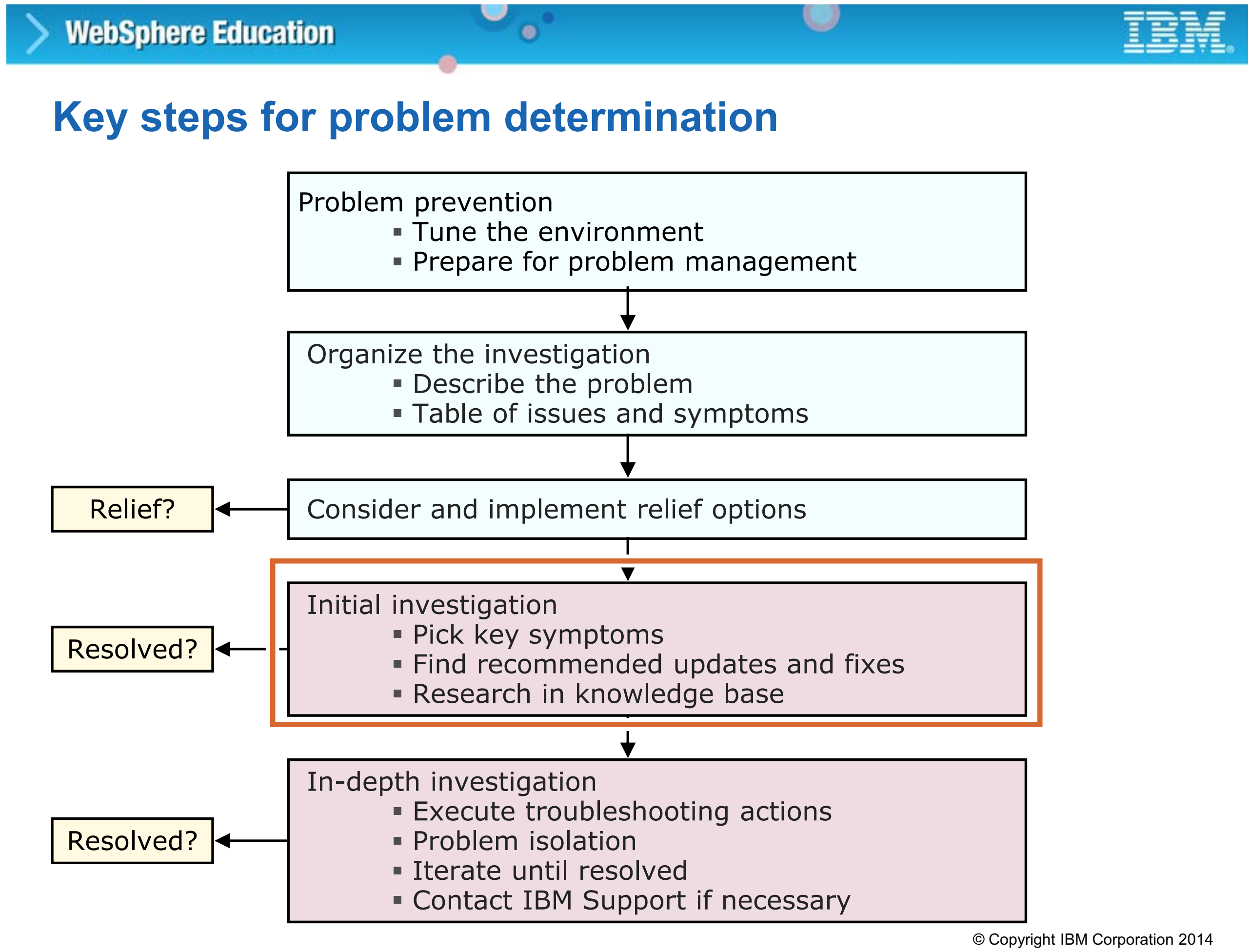


Figure 3-4. Key steps for problem determination WB8691.0

Notes:

WebSphere Education

Basic instruments for problem determination

- Enable logging and tracing to examine general runtime messages
- Enable **`verbose:gc`** trace to collect garbage collection data
- Analyze heap dumps to detect possible memory leaks
- Analyze **`Javacore`** logs to detect any hung threads
- Use **IBM Support Assistant** to facilitate program maintenance request (PMR) resolution:
  - Which logs need to be collected for opening PMRs
  - How to gather software version information along with the maintenance package information
  - Dump the contents of the Business Process Choreographer system destinations
  - Dump the contents of the Service Component Architecture (SCA) system destinations

Figure 3-5. Basic instruments for problem determination WB8691.0

Notes:

WebSphere Education

Types of tools and where to find them (1 of 4)

- Logging and tracing (Basic and HPEL modes)
- Troubleshooting panels in the administrative console
- Specialized tracing and runtime checks:
  - Connection leak detection
  - Memory leak detection
  - Enabled by tracing a specific component or setting a specialized custom property
- First-failure data capture (FFDC)
  - Always enabled
  - Captures key information when a potentially abnormal situation occurs
  - Data is collected in the <profile_root>/logs/ffdc directory
  - If an FFDC record is written, that does not necessarily mean that a serious problem occurred

Figure 3-6. Types of tools and where to find them (1 of 4) WB8691.0

Notes:

The next few slides provide an overview of some of the types of tools that are used for troubleshooting and where they can be found. It is not an exhaustive list. Some of these tools are described in more detail later in the course. The following article provides a more detailed explanation of this topic:
`http://www.ibm.com/developerworks/websphere/techjournal/0702_supauth/0702_supauth.html`

The first-failure data capture (FFDC) log file saves information that is generated from a processing failure (for example, a Java exception):

- Captured data, which is saved in log files for use in analysis
- An index file that references all of the exceptions that FFDC logs
- An exception file for each exception type from each probe

Capturing FFDC data does not affect performance.

You can configure the number of days this information is saved (afterward, it is deleted). Retrieve these log files with an FTP client from any other environment. Because the index and exception logs are text files, they can be viewed in any ASCII-capable text editor or viewer.

The FFDC configuration properties files are in the properties directory under the WebSphere Application Server product installation. There are three properties files, but only the `ffdcRun.properties` file should be modified. You can set the exceptionFileMaximumAge property to configure the number of days between purging the FFDC log files. The value of the ExceptionFileMaximumAge property must be a positive number.

The following Redbooks publication contains some good documentation on using the FFDC for problem determination: SG246880: *WebSphere for z/OS V5 Problem Determination*.

WebSphere Education IBM

Types of tools and where to find them (2 of 4)

- JVM diagnostic data can be generated per server from the administrative console
  - Verbose garbage collection
  - Heap memory dump, javacore, system memory dump
- Performance-related tools:
  - Performance Monitoring Infrastructure (PMI) is a facility specific to WebSphere
  - Tivoli Performance Viewer, available from the administrative console, is the primary tool for viewing PMI data
  - Java Health Center: GC and memory monitoring
  - Request metrics can be accessed by using Application Response Measurement (ARM) infrastructure
  - IBM Application Performance Diagnostics Lite is a powerful, lightweight tool that can be used to optimize the performance of Java EE applications

Figure 3-7. Types of tools and where to find them (2 of 4) WB8691.0

Notes:

The Java Diagnostics Guides also provide details about some of these tools and how to generate various types of memory dumps. You can find the V6 guide here:
`http://publib.boulder.ibm.com/infocenter/javasdk/v6r0/index.jsp`

Request metrics are available from many products, making it possible to follow a request from end to end in a complex system. Both PMI and request metrics are exported through public APIs, making it possible to write specialized or third-party tools to use this information.

The Tivoli Composite Application Manager family of tools is the comprehensive platform for working with performance data, including PMI, request metrics, and other techniques.

For more information about IBM Application Performance Diagnostics Lite, see the developerWorks website: `http://www.ibm.com/developerworks/servicemanagement/apm/apd/`

WebSphere Education

Types of tools and where to find them (3 of 4)

Monitoring and detection:

- **Hung thread detection facility** is directly connected into the WebSphere Application Server run time
- **Performance and diagnostic advisors** are accessible from the administrative console
- Performance advisor is available in the Tivoli Performance Viewer

Specific component investigation:

- **System Management Configuration Validation** can detect errors in the XML configuration files
  - It is accessible from the administrative console
- **DumpNameSpace** dumps the contents of the JNDI name space at a particular server
  - It is a stand-alone product with WebSphere Application Server (`<install_root>/bin`)
- **Class loader viewer** helps resolve class loading issues and is accessible through the troubleshooting menu of the administrative console

Figure 3-8. Types of tools and where to find them (3 of 4) WB8691.0

Notes:

You can specify a hang threshold, or timeout period, for threads, and receive alerts about potentially hung threads.

The System Management Configuration Validation facility can do automated checks to detect inconsistencies and errors in the complex set of XML files that contain the entire WebSphere Application Server system configuration. Such errors, though relatively rare in recent versions of the product because of many runtime safety checks, can still crop up. These errors can be because of as-yet-undiscovered product defects, unexpected events that occur during configuration operations (like crashes), or operator mistakes during configuration. This facility is embedded inside the WebSphere Application Server runtime itself, and can be started from the administration console (in the troubleshooting pane).

WebSphere Education

Types of tools and where to find them (4 of 4)

- Installation issues:
  - **Installation Manager**
  - **Installation verification tool (IVT)** verifies each profile from its First Steps console with the IVT tool (`<profile_root>/bin`)
  - **Installation verification utility (installver)**: Replaced by the verification capabilities of the Installation Manager in version 8
  - **VersionInfo, HistoryInfo, GenHistoryReport** is used to track changes and apply fix packs; all bundled with WebSphere Application Server
- Debuggers and profilers are valuable to application support, but are outside the scope of this course
  - Rational Application Developer
  - IBM Assembly and Deploy Tools
  - Java Health Center can be used for profiling

Figure 3-9. Types of tools and where to find them (4 of 4) WB8691.0

Notes:

WebSphere Application Server Version 7 and earlier had an installation verification utility, the installer command, that would verify checksums of installed files against a bill of materials that was included with the product. In WebSphere Application Server Version 8.0 and later, where the installation is based on the Installation Manager rather than on InstallShield MultiPlatform (ISMP), the verification capabilities of the Installation Manager replace the installer command.

Debuggers and profilers:

- The ability to debug and profile is also often valuable to the application support process. WebSphere Application Server provides these facilities through the JVM, either through the JVMPI or JVMTI interfaces of that JVM. WebSphere Application Server system management makes it easy to set up the appropriate JVM parameters to enable these facilities, either through the WebSphere Application Server administration console, or through a wsadmin script.
- Rational and other Eclipse-based development tools, including Rational Application Developer and IBM Assembly and Deploy Tools, include a powerful debugger and profiler tool that connects to these facilities. For profiling-related work, you might also consider the Performance Inspector family of tools, which provides various tools to extract and analyze runtime

performance information from a JVM, by using these same basic interfaces. These tools are available on alphaWorks (for Windows) or SourceForge (for Linux).

3.2. Logs and tracing: Basic mode

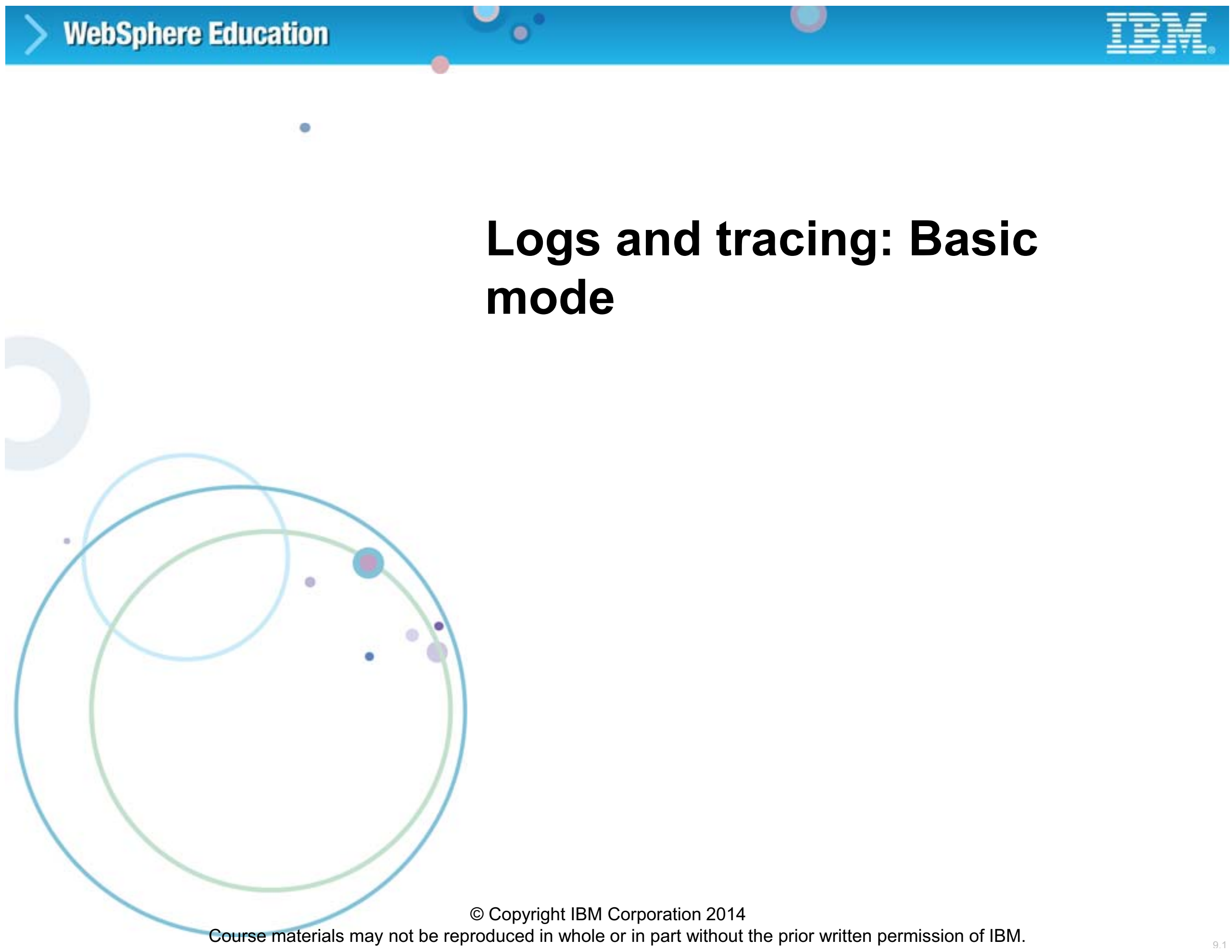


Figure 3-10. Logs and tracing: Basic mode WB8691.0

Notes:

WebSphere Education

Basic mode logs and tracing

- Logs:
  - Report key events in the system and enabled by default
  - Incur minimal performance usage
  - Look for key messages like warnings, errors, and exceptions
- Tracing:
  - Application code-level events; level of detail can be configured
  - Detailed tracing can be enabled by specifying a trace string
  - Trace helps you understand the flow of the application
  - HPEL consolidates key messages on a particular node and contains extended service information
- JMX-based monitoring:
  - Most key events are exported as JMX events
  - Tools can be built for remotely monitoring and capturing events

Figure 3-11. Basic mode logs and tracing WB8691.0

Notes:

The Tivoli Monitoring for Web Infrastructure tool uses JMX-based monitoring.

The default trace string is `*=info`. This string controls the logging level of the `trace.log` file. If you want only errors to be logged, you can specify `*=fatal` or `*=severe`. For more messages, you can specify `*=detail`.

Tracing is explained later in this unit.

General troubleshooting tool: Logs

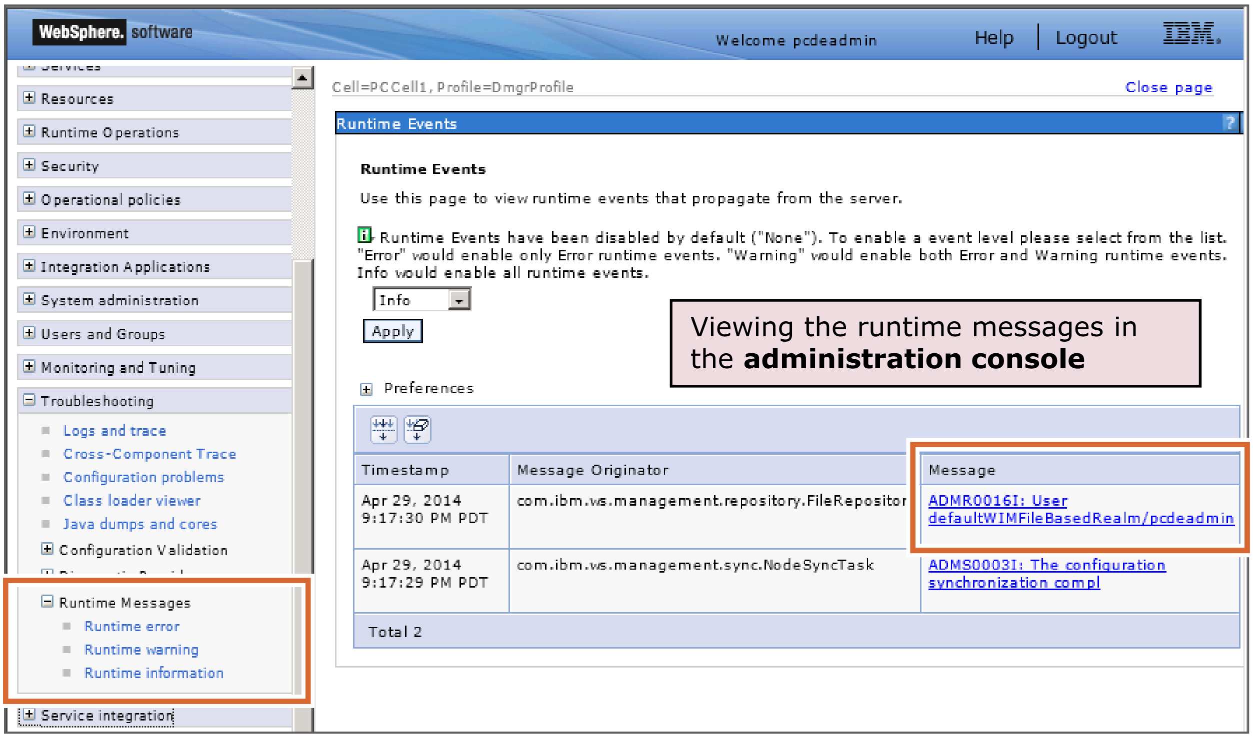


Figure 3-12. General troubleshooting tool: Logs WB8691.0

Notes:

WebSphere Education

Trace configuration

- **Troubleshooting > Logs and Trace >** <***server_name***> **> Diagnostic Trace**
- On **Configuration** tab, server restart is required for settings to take effect
- Configurable output
  - Memory buffer or file

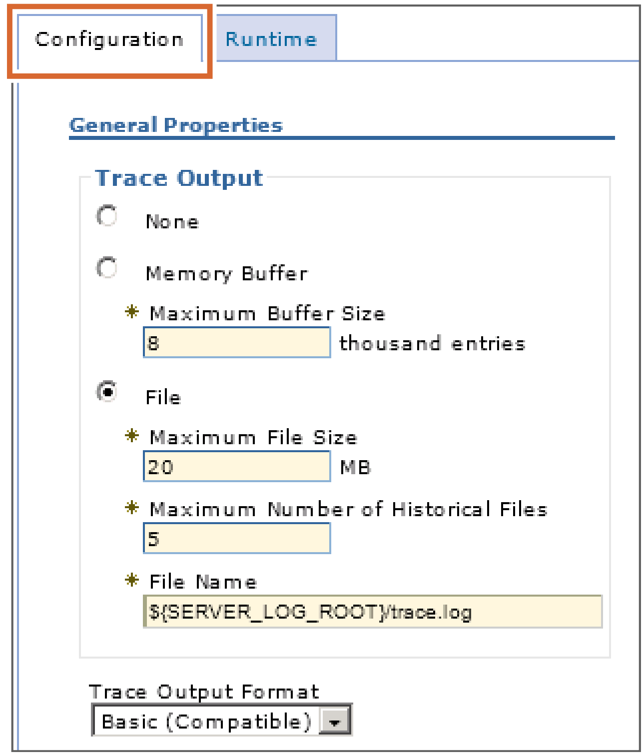


Figure 3-13. Trace configuration WB8691.0

Notes:

Trace dump and the Runtime tab

- On the **Runtime** tab, settings take effect immediately (useful in production environments)
- View and dump are available in the **Runtime** tab of the diagnostic trace
- Before you can view or dump trace, you need to specify the log detail level

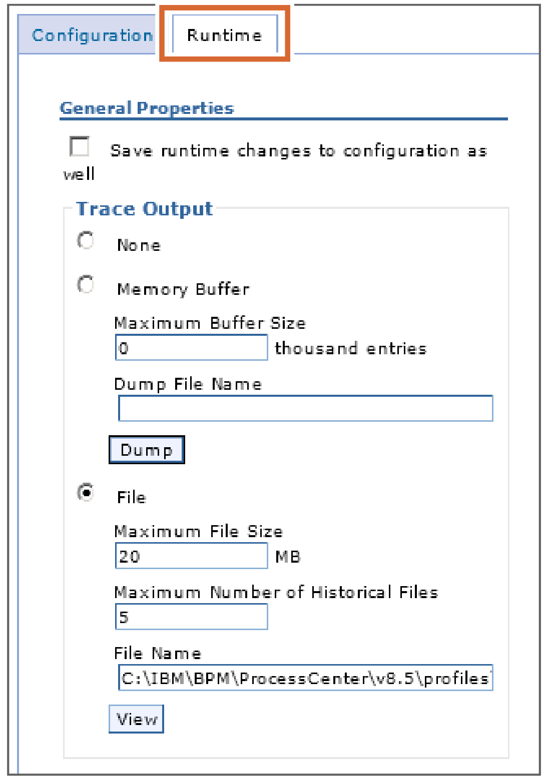


Figure 3-14. Trace dump and the Runtime tab WB8691.0

Notes:

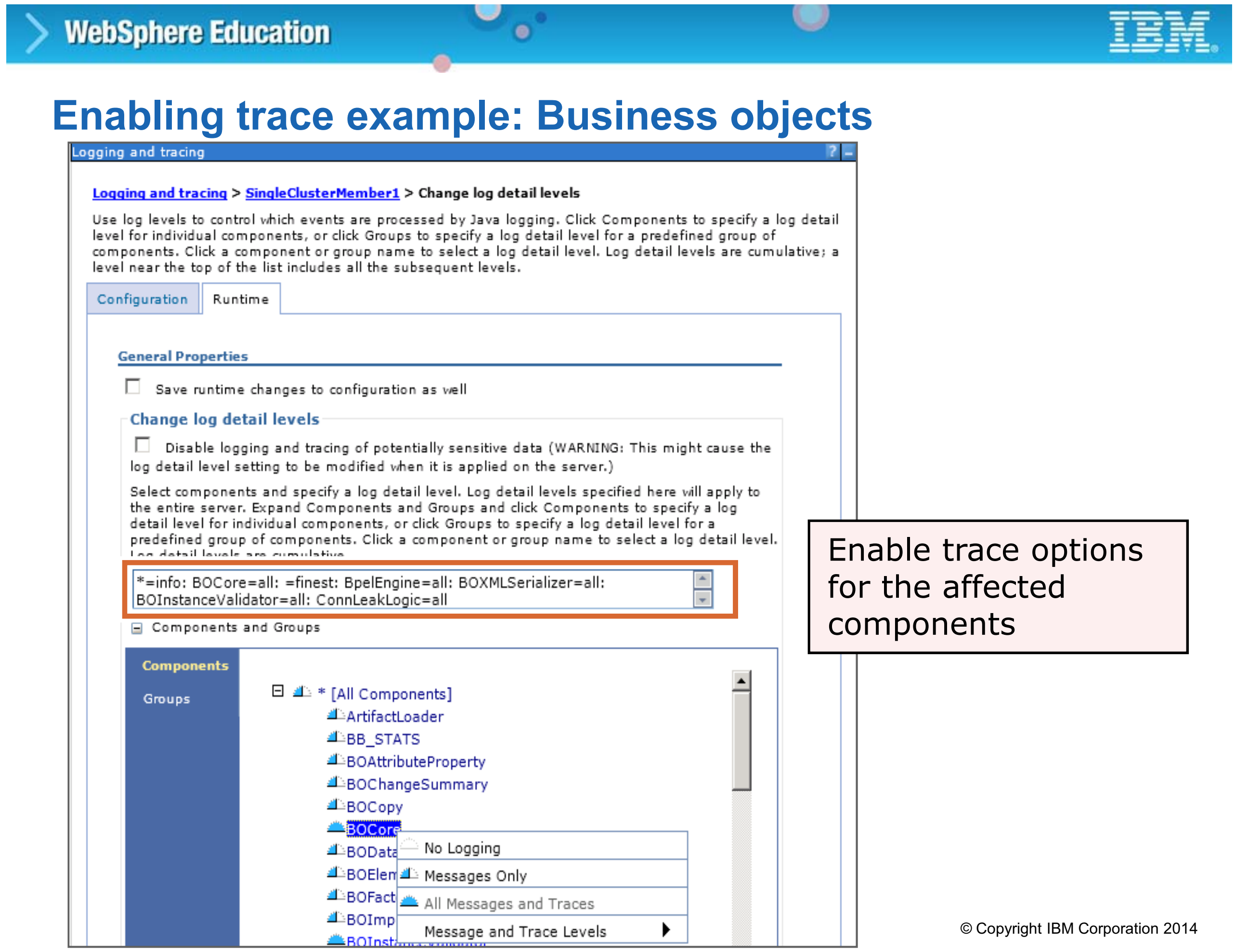


Figure 3-15. Enabling trace example: Business objects WB8691.0

Notes:

WebSphere Education IBM

Log files and locations

Log files	Location
`SystemOut.log`	<*was_root*>/profiles/<*profile_name*>/logs/<*server_name*>
`SystemError.log`	<*was_root*>/profiles/<*profile_name*>/logs/<*server_name*>
`startServer.log`	<*was_root*>/profiles/<*profile_name*>/logs/<*server_name*>
`stopServer.log`	<*was_root*>/profiles/<*profile_name*>/logs/<*server_name*>
`activity.log`	<*was_root*>/profiles/<*profile_name*>/logs/<*server_name*>
`trace.log`	<*was_root*>/profiles/<*profile_name*>/logs/<*server_name*>
`http_plugin.log`	<*plugin_root*>/logs/<*webserver_name*>
`native_stdout.log`	<*was_root*>/profiles/<*profile_name*>/logs/<*server_name*>
`native_stderr.log`	<*was_root*>/profiles/<*profile_name*>/logs/<*server_name*>
`btrace.1`	<*was_root*>/profiles/<*profile_name*>/logs/<*server_name*>

Note: `btrace.1` is used for the binary trace facility of Intelligent Management, which is covered in a later unit

Figure 3-16. Log files and locations WB8691.0

Notes:

All WebSphere Application Server log files are stored in the <*was_root*>\profiles\<*profile_name*>\logs directory, by default.

`SystemOut.log` and `SystemErr.log` are the default names for the JVM logs. They contain server and also user-program information (sent by a `System.out.xxx` code in the program).

The `startServer.log` and `stopServer.log` files can also be found under the <*was_root*>\logs\<*servername*> directory; they contain information that the server logs as it starts up and shuts down. The IBM service log (`activity.log`) file size can be set by using the administrative console. You can also disable the `activity.log`.

There are several types of logs:

- JVM logs: Created by redirecting the `System.out` and `System.err` streams of the JVM to independent log files.
  - One set of JVM logs for each application server and all of its applications that are located by default in the <*install_root*>/profiles/<*profile_name*>/logs/<*server_name*> directory

- Process logs: Contain two output streams (stdout and stderr) that are accessible to native code that runs in the process.
  - One set for each application server
- IBM service log: Contains both the WebSphere Application Server messages that are written to the `System.out` stream and some special messages that contain extended service information that is normally not of interest, but can be important when analyzing problems.
- The HTTP server plug-in maintains a special log.
- Java virtual machine (JVM) logs: The JVM logs are created by redirecting the `System.out` and `System.err` streams of the JVM to independent log files. WebSphere Application Server writes formatted messages to the `System.out` stream. In addition, applications and other code can write to these streams by using the `print()` and `println()` methods that the streams define. If there is a WebSphere Application Server Network Deployment configuration, JVM logs are also created for the deployment manager and each node agent because they also represent JVMs.
- Process logs: WebSphere Application Server processes contain two output streams that are accessible to native code that runs in the process. These streams are the stdout and stderr streams. Native code, including Java virtual machines (JVM), might write data to these process streams. In addition, JVM-provided `System.out` and `System.err` streams can be configured to write their data to these streams. As with JVM logs, there is a set of process logs for each application server since each JVM is an operating system process. For a WebSphere Application Server Network Deployment configuration, there is a set of process logs for the deployment manager and each node agent.
- IBM service logs: The IBM service logs contain the WebSphere Application Server messages that are written to the `System.out` stream. Some special messages that contain extended service information that is normally not of interest, but can be important when analyzing problems, are also written. There is one service log for all WebSphere Application Server JVMs on a node, including all application servers. The IBM Service log is maintained in a binary format and requires a special tool to view. This viewer, the Log Analyzer, provides more diagnostic capabilities. In addition, the binary format provides capabilities that IBM Support organizations use. The HTTP server plug-in log is covered later in this presentation.

WebSphere Education

Configuring JVM logs

- Click:
  Troubleshooting > Logs and trace > *server_name* > JVM Logs
- `System.out` and `System.err` logs can be configured from this page
- Logs are self-managing
  - Rollover based on time or file size
  - Number of historical log files is configurable
- Use the **Runtime** tab to view the logs

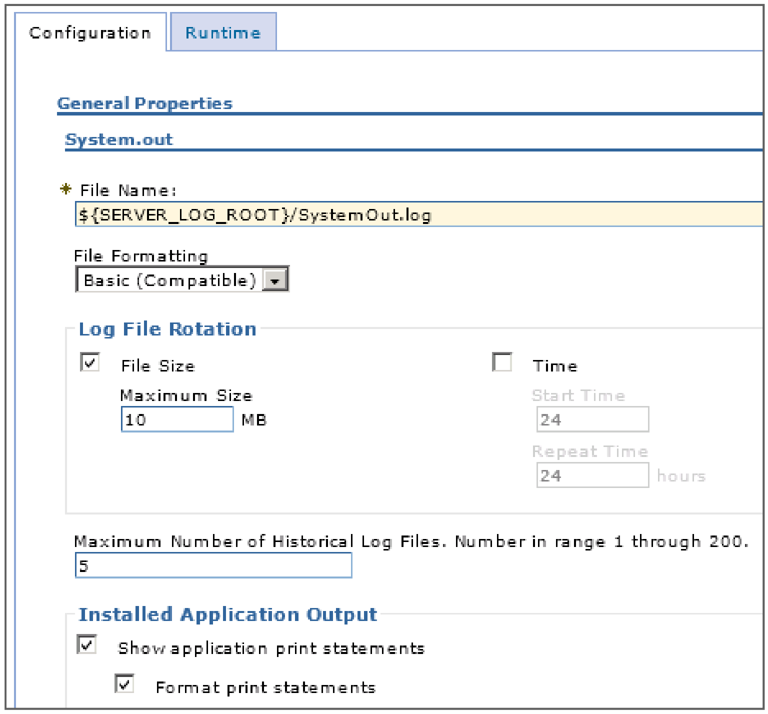


Figure 3-17. Configuring JVM logs WB8691.0

Notes:

You can also configure the logs from this alternative method: **Servers > Application** servers > ***server_name*** **> Logging and Tracing > JVM Logs**.

WebSphere Education IBM

Viewing runtime messages in the console

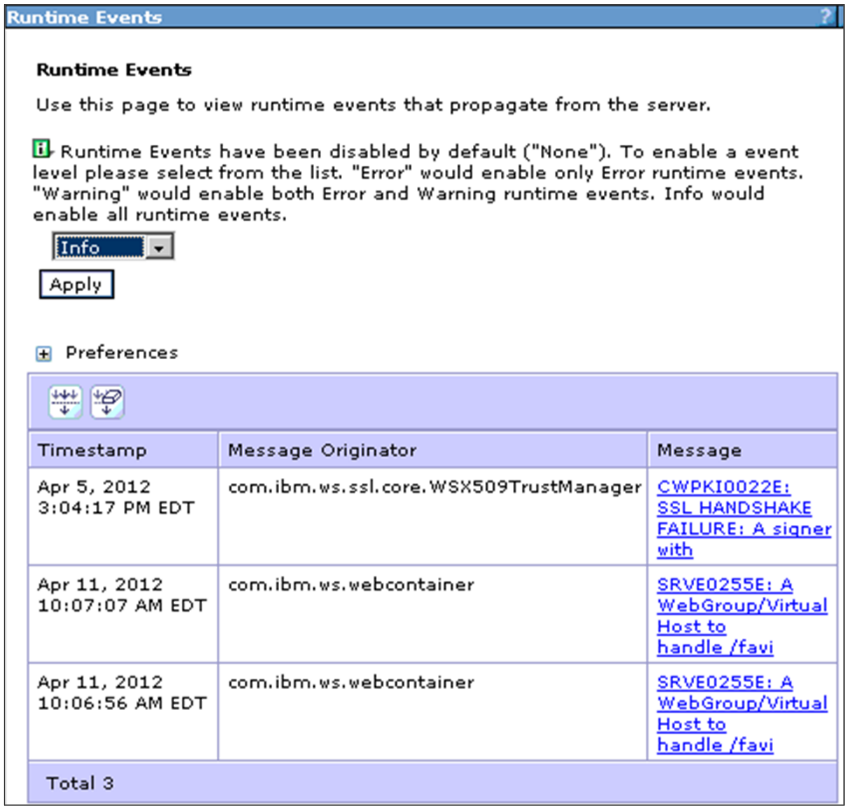


Figure 3-18. Viewing runtime messages in the console WB8691.0

Notes:

Navigate to **Troubleshooting > Runtime Messages > Runtime error.**

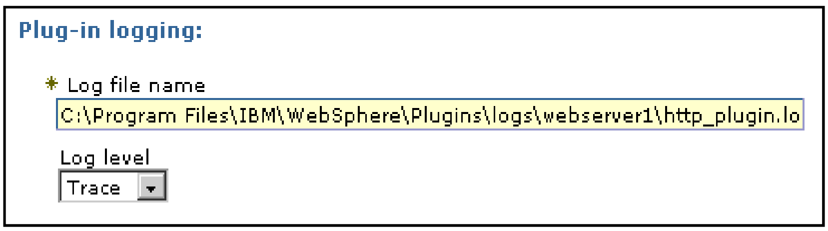
While viewing runtime messages, first select the error, warning, or information category links (a count of zero means nothing is available). The details for the selected category are shown. Selecting one of these links gives you more detailed information.

Note: you might have multiple pages of messages, and the button on the bottom of the page allows you to see them all. Information is displayed on the detail screen for the event so you can resolve the problem with user action.

WebSphere Education

HTTP plug-in logs and tracing

- Click **Servers > Server Types > Web servers >** ***web_server_name*** **> Plug-in properties > Configuration tab > Plug-in logging** to configure plug-in logs and tracing
- Default location:
  `<plugins_root>/logs/<web_server_name>/http_plugin.log`
- Set the Log level to **Trace** to trace all the steps in the request process (caution: produces much output)



- Use the tool **IBM Web Server Plug-in Analyzer for WebSphere Application Server** to analyze the plug-in trace
  - Available in the IBM Support Assistant

Figure 3-19. HTTP plug-in logs and tracing WB8691.0

Notes:

IBM Web Server Plug-in Analyzer for WebSphere Application Server helps discover potential problems with trace and configuration files during use of WebSphere Application Server. The tool parses both plug-in configuration and corresponding trace files and then applies pattern recognition algorithms to alert users to possible inconsistencies.

The tool provides a list of HTTP return codes, URIs, graphical presentations of available clusters, and server topologies from the configuration and trace files. IBM Web Server Plug-in Analyzer for WebSphere Application Server is available as a tool add-on in the IBM Support Assistant.

WebSphere Education

Embedded HTTP server logs

- Administrative console panels for configuring embedded HTTP server logs (access and error)
- From the main application server panel, click HTTP Error and NCSA Access Logging
- Access and error logs can be controlled separately
- When maximum file size is reached, the oldest entries are pruned

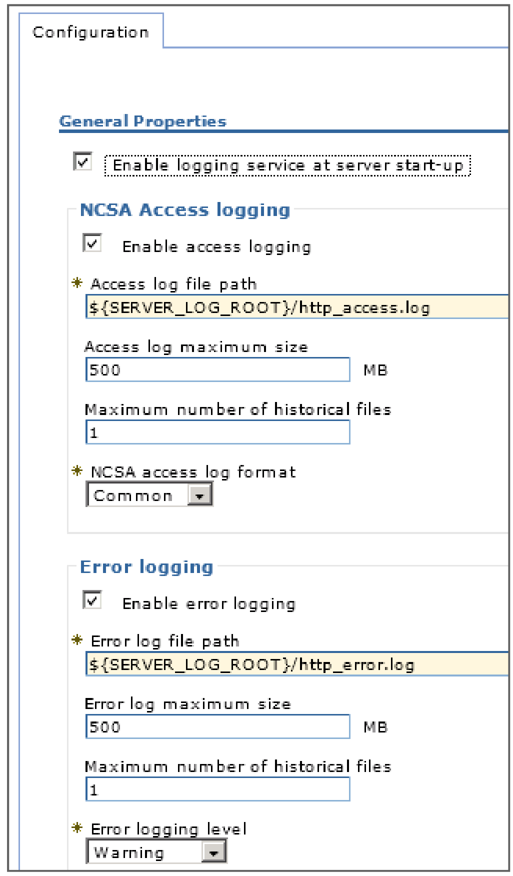


Figure 3-20. Embedded HTTP server logs WB8691.0

Notes:

Embedded HTTP server logs:

- Administrative console panes for configuring embedded HTTP server logs (access and error).
- From main application server pane, navigate to **Troubleshooting > Logs and trace >** ***server_name*** **>** HTTP Error and NCSA Access Logging.
- Access and error logs can be controlled separately.
- When maximum file size is reached, the oldest entries are removed.

To enable the access or error log, you must check the “Enable service at server startup” check box, and also the check box for the specific log you want to enable. There is not a runtime tab for these logs. Logging begins after changes are saved to your configuration and the application server restarts.

WebSphere Education IBM

Diagnostic traces

- Trace files show the time and sequence of methods that are called from WebSphere base classes
  - Use trace data to help pinpoint the failure
- Trace can be started:
  - While the server is running, use Runtime Diagnostic Trace
  - When the server is started, use Configuration Diagnostic Trace
- Trace output can be directed to:
  - Memory ring buffer that is dumped after trace stops
  - File
- Trace has a significant impact on performance
  - Enable temporarily for problem determination
  - Trace to file is slower than trace to memory ring buffer in the **Runtime** tab

Figure 3-21. Diagnostic traces WB8691.0

Notes:

Trace output allows administrators to examine processes in the application server and diagnose various issues.

On an application server, trace output can be directed either to a file or to an in-memory circular buffer. If trace output is directed to the in-memory circular buffer, it must be dumped to a file before it can be viewed.

On an application client or stand-alone process, trace output can be directed either to a file or to the process console window.

In all cases, trace output is generated as plain text in either basic, advanced, or log analyzer format as specified by the user. The basic and advanced formats for trace output are similar to the basic and advanced formats that are available for the JVM message logs.

The procedure for using trace is as follows:

1. Configure an output destination to which trace data is sent.
2. Enable trace for the appropriate WebSphere Application Server or application components.
3. Run the application or operation to generate the trace data.

4. Analyze the trace data or forward it to the appropriate organization for analysis.
5. Click **Troubleshooting > Logs and Trace > *server_name*** in the administrative console.

WebSphere Education

Enabling diagnostic trace

- **Troubleshooting > Logs and Trace > *server_name* > Diagnostic Trace**
- Configure Trace Output
  - None
  - Memory buffer
  - File (default)
- Configure Trace Output Format
  - Basic (recommended by IBM support)
  - Advanced
- **Note**: Configure Log Detail Level to get trace output

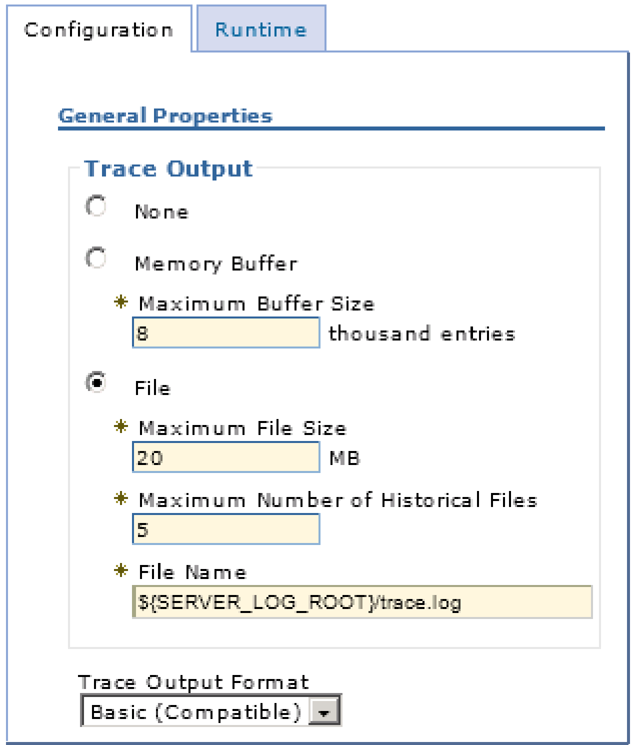


Figure 3-22. Enabling diagnostic trace WB8691.0

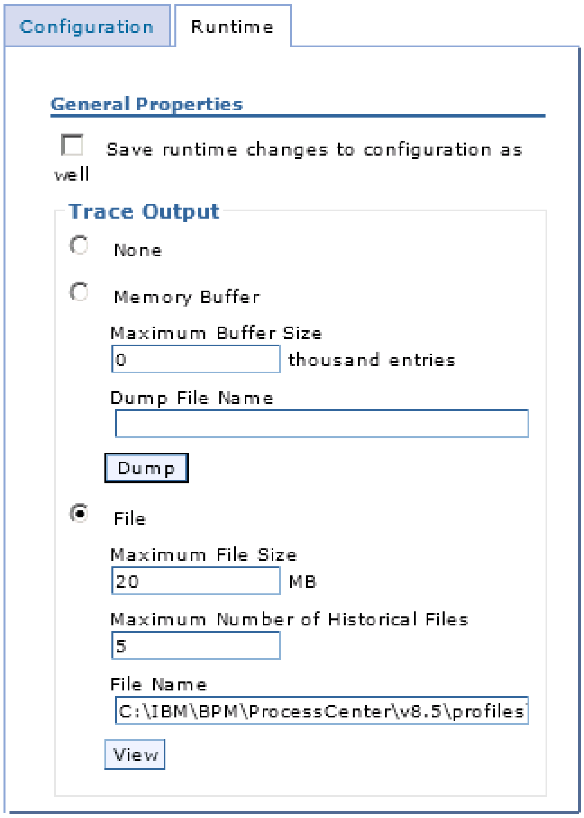
Notes:

Specifying the trace settings under the Configuration tab requires a server restart for the tracing to take effect. However, enabling the tracing under the Runtime tab takes effect immediately (which can be useful for tracing environments such as a production environment, which can rarely afford a server restart). Nonetheless, it is important to know that after a server restart, the runtime settings are lost unless the “Save runtime changes to configuration as well” box is checked. Also, runtime traces report only the current system information; if the server is already having issues, enabling runtime traces does not report what caused the problem to begin with, only why current actions fail. Therefore, IBM Support suggests that the trace settings be made under the Configuration tab so that server startup can be reviewed and the root cause of the problem can be determined.

The Maximum File Size and Maximum Number of Historical Files are important to ensure the trace information that is captured does not get wrapped and lost. By default, these values are set to 20 MB for the file size, and one backup file. As a result, when the `trace.log` file reaches 20 MB in size, a new `trace.log` gets created. The previous `trace.log` file is renamed to `trace_`*<date>*`.log` where *<date>* is the date and time of the last entry in the trace. However, as soon as the new `trace.log` file reaches 20 MB in size again, it then wraps into a new trace file as before, but the previous `trace_<date>.log` file is deleted. Therefore, it is important to ensure that the Number of Historical Files and Maximum File Size are large enough when re-creating a complex or time-consuming problem. The largest value that can be inserted for the Historical Files is 20. The WebSphere Application Server support team prefers to keep the Maximum File Size no larger than 50 MB, as the traces become more difficult to load and read if the file size is too large.

WebSphere Education

Diagnostic trace dump and the Runtime tab



- On the **Runtime** tab, settings take effect immediately (useful in production environments)
- File and dump are available in the **Runtime** tab of diagnostic trace
- A trace analyzer can be used to analyze trace output, but you might prefer your favorite editor
- Before you can view or dump trace, specify the log detail level

Figure 3-23. Diagnostic trace dump and the Runtime tab WB8691.0

Notes:

The trace analyzer for WebSphere Application Server enables you to view a complex text-based trace file in a more easy-to-use GUI interface.

WebSphere Education IBM

Setting the log detail level

- **Logs and trace > *server_*name > Change Log Detail Level**
- Log detail level affects tracing ***and*** regular logging
  - Setting levels below **`*=info`** reduces the amount of data in logs
  - **`*=off`** disables logging altogether
- Trace levels (**fine**, **finer**, **finest**) do not appear in the trace file unless logging is enabled
- Use the graphical menu to type in or set the log string
- Default is **`*=info`**
- User-created applications can be instrumented too, and be included in the trace output

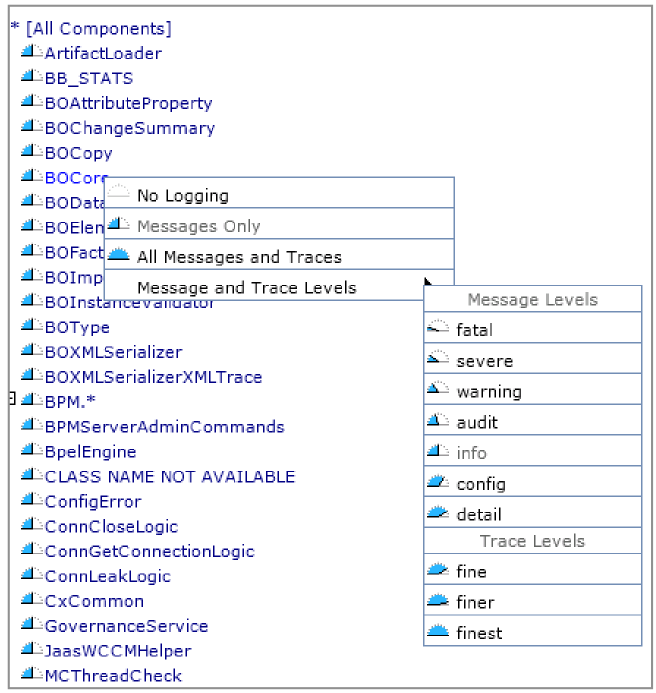


Figure 3-24. Setting the log detail level WB8691.0

Notes:

Log levels control the events that Java logging processes.

WebSphere Application Server controls the levels of all loggers in the system. The level value is set from configuration data when the logger is created and can be changed at run time from the administrative console.

Note: Trace information, which consists of events at levels fine, finer, and finest, can be written only to the trace log. Therefore, if you do not enable diagnostic trace, setting the log detail level to fine, finer, or finest does not affect the data that is logged.

Log string syntax: `<`*`component / group`*`> = <`*`log level`*`>`

Examples include:

- `com.ibm.ws.classloader.ClassGraph=finest`

  Enables the finest trace level for `com.ibm.ws.classloader.ClassGraph`

- `EJBContainer=fine`

  Enables the least verbose trace level for all components in the EJBContainer group

- `com.ibm.ws.classloader.*=finer`

  Enables detailed trace for all classes in the `com.ibm.ws.classloader` package

- `*=info Sets`

  Sets the log level for all components to `*=info` (default: no trace output)

WebSphere Education

Enabling and disabling sensitive log and trace guard

- Administrators can prevent sensitive information, such as data provided from users in HTTP requests, from being written in log and trace files
- In some cases, if access to private data can help with debugging, you can disable sensitive log and trace guard
  - For example, you might see that a credit card number that was entered in a web form did not have the required number of digits

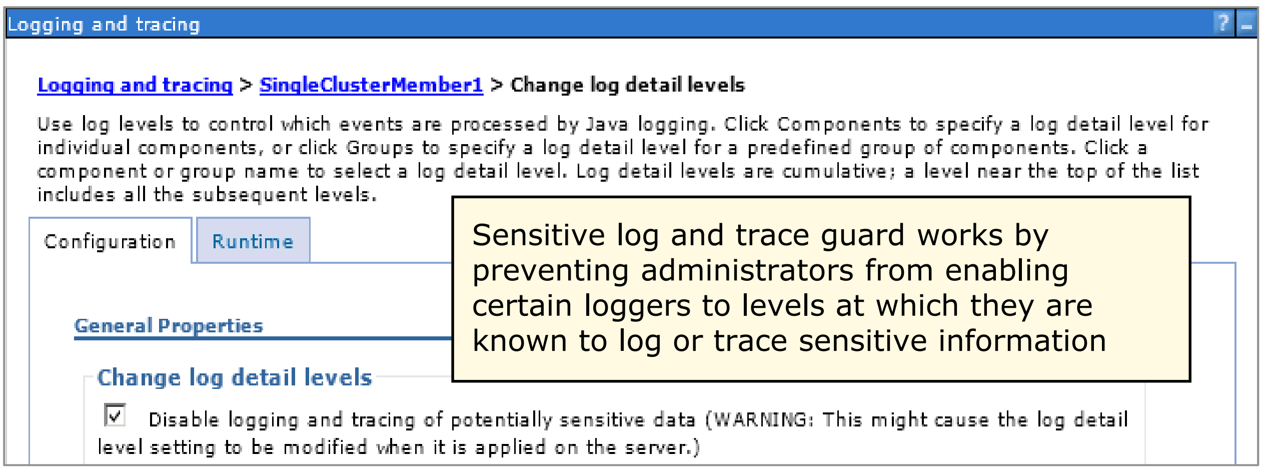


Figure 3-25. Enabling and disabling sensitive log and trace guard WB8691.0

Notes:

Administrators who use WebSphere Application Server can prevent sensitive information, such as data that users provide in HTTP requests, from being written in log and trace files. In some cases, when having access to private data can help with debugging, you might want to disable sensitive log and trace guard. For example, you might see that a credit card number that was entered in a web form did not have the required number of digits.

Sensitive log and trace guard works by preventing administrators from enabling certain loggers to levels at which they are known to log or trace sensitive information.

After you enable sensitive log and trace guard, the server is now configured to prevent known sensitive loggers from writing sensitive content to the log and trace files. After you disable sensitive log and trace guard, the server is now configured to allow known sensitive loggers to write sensitive content to the log and trace files. If you used the deployment manager to complete these steps, you might be required to synchronize the node agent on the target node before restarting the server.

See the information center topic “Enabling and disabling sensitive log and trace guard” at:

`http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/`
`com.ibm.websphere.express.doc/ae/ttrb_enablesensitivelogtrace.html`

WebSphere Education

Enabling trace by using Jython scripts (1 of 2)

- Use these commands to enable tracing on the configuration
- This example sets the trace string `com.ibm.ws.*=all=enabled`

```
cellName = 'myCell'
nodeName = 'myNode'
serverName = 'myServer'

server = AdminConfig.getid(
    '/Cell:%(cellName)s/Node:%(nodeName)s/Server:%(serverName)s' %
    locals() )
ts = AdminConfig.list( 'TraceService', server )

# print AdminConfig.showAttribute( ts, 'startupTraceSpecification' )

AdminConfig.modify( ts, [ [ 'startupTraceSpecification',
    'com.ibm.ws.*=all=enabled' ] ] )
AdminConfig.save()
```


Figure 3-26. Enabling trace by using Jython scripts (1 of 2) WB8691.0

Notes:

If access to the administrative console is not possible, tracing can also be enabled by using the wsadmin utility. The information center contains more detailed instructions on enabling tracing by using the wsadmin tool.

WebSphere Education

Enabling trace by using Jython scripts (2 of 2)

- To enable runtime traces with wsadmin, use these commands:

```
ts = AdminControl.queryNames(
    'type=TraceService,node=%(nodeName)s,process=%(serverName)s,*' %
    locals() )
# print AdminControl.getAttribute( ts, 'traceSpecification' )
AdminControl.setAttribute( ts, 'traceSpecification',
    'com.ibm.ws.*=all=enabled' )
```

- You can find trace strings for WebSphere components that are associated with a specific problem in the MustGather documents

Figure 3-27. Enabling trace by using Jython scripts (2 of 2) WB8691.0

Notes:

WebSphere Education

JVM log and trace log headers

```
************ Start Display Current Environment ************
1 WebSphere [BPMPS 8.5.0.1 20131107-140634]Platform 8.5.5.1 [ND 8.5.5.1
    cf011341.03] running with process name PSCell1\Node1\server1 and
    process id 5496
Host Operating System is Windows Server 2008 R2, version 6.1
2 Java version = 1.6.0, Java Compiler = j9jit26, Java VM name = IBM J9 VM
was.install.root = C:\IBM\IID\PS\v8.5
user.install.root = C:\IBM\IID\PS\v8.5\profiles\qbpma
Java Home = C:\IBM\IID\PS\v8.5\java\jre
ws.ext.dirs = C:\IBM\IID\PS\v8.5/java/lib;
…<truncated>
3 Classpath = C:\IBM\IID\PS\v8.5\profiles\qbpmaps/properties;
…<truncated>
Java Library path = C:\IBM\IID\PS\v8.5/lib/native/win/x86_64/;
…<truncated>
Orb Version = IBM Java ORB build orb626ifx-20130530.00
    (SR6+IX90121+IX90123)
4 Current trace specification = *=info:com.ibm.ws.*=all
************* End Display Current Environment *************
```

1. WebSphere level
2. Java version
3. Classpath
4. Trace spec

Figure 3-28. JVM log and trace log headers WB8691.0

Notes:

This header is also produced each time that the server is started. From this header, you can see the exact:

1. WebSphere Application Server level (cell name, node name, and server name, along with the process ID)
2. Java version
3. Class path
4. Current trace specification

Additionally, the contents of the class path and Java lib path are visible, and also the WebSphere Application Server variables `was.install.root` and `user.install.root`. During startup, the trace specification in use is printed right after the header.

WebSphere Education IBM

Default trace format

- **Basic** format: the following format is used to record trace events:

```
[timestamp] <threadId> <className> <eventType> <methodName> <textmessage>
```

- Example:

```
[4/3/14 16:41:28:041 EDT] 00000020 CacheHook     3   Request Type:GET
[4/3/14 16:41:28:041 EDT] 00000020 CacheHook     3   dncResult: true
    skipCache: false
[4/3/14 16:41:28:041 EDT] 00000020 FragmentCompo 3   starting fragment
    composer for /secure/layouts/contentLayout.jsp
    parent=/com.ibm.ws.console.events.forwardCmd.do
[4/3/14 16:41:28:041 EDT] 00000020 ESISupport    >
    handleESIPreProcessing() Entry
```


Figure 3-29. Default trace format WB8691.0

Notes:

The “basic” trace format is the preferred trace format of the WebSphere Application Server support team, and it is also the default trace format.

A full listing of each of the Diagnostic Trace Service settings and their descriptions is documented in the WebSphere Application Server Information Center.

From the sample trace above, you can see:

- The time frame is on Apr. 3, 2014 at 16:41:28:041 EDT time.
- The trace output shows a single threaded process with ID 000000a.
- The class name that is executing is “CacheHook.”

  If the class name is over 13 characters, only the first 13 characters are displayed.

Here is the breakdown of each trace entry line and the information it provides:

- `[4/3/14 16:41:28:041 EDT] 00000020 CacheHook    3 Request Type:GET`
  - Debug point in the CacheHook class.
- `[4/3/14 16:41:28:041 EDT] 00000020 ESISupport > handleESIPreProcessing() Entry`

- The `handleESIPreProcessing()` method on ESISupport is executing.

Not all WebSphere Application Server code outputs the method entry points, exit points, and debug messages.

WebSphere Education IBM

Reading a log or trace file (1 of 2)

- Example log record format

```
[4/21/14 10:10:45:134 EDT] 00000001 WSKeyStore    W
CWPKI0041W: One or more key stores are using the default
password.
```

- Time stamp = `[4/21/14 10:10:45:134 EDT]`
- Thread ID = `00000001`
- Logger = `WSKeyStore`
- Message type = `W`
- Message code = `CWPKI0041W`
- Message = `One or more key stores are using the default password.`

Msg type	Description
1,2,3	Trace info: fine, finer, finest
A	Audit
W	Warning
Z	Type was not recognized
E	Error
D	Detail
C	Configuration
F	Fatal (exits process)
I	Information
O	Program output (`system.out`)
R	Program output (`system.err`)

Figure 3-30. Reading a log or trace file (1 of 2) WB8691.0

Notes:

The following list defines the sections of a trace entry:

- **TimeStamp**: The time stamp is formatted to agree with the locale of the process where it is formatted. It includes a fully qualified date (YYMMDD), 24-hour time with millisecond precision, and the time zone.
- **ThreadId**: An eight-character hexadecimal value that is generated from the hash code of the thread that issued the trace event.
- **ThreadName**: The name of the Java thread that issued the message or trace event.
- **ShortName**: The abbreviated name of the logging component that issued the trace event. This abbreviated name is typically the class name for WebSphere Application Server internal components, but might be some other identifier for user applications.
- **LongName**: The full name of the logging component that issued the trace event. This name is typically the fully qualified class name for WebSphere Application Server internal components, but might be some other identifier for user applications.
- **EventType**: A one-character field that indicates the type of the trace event. Trace types are in lowercase.

- **ClassName**: The class that issued the message or trace event.
- **MethodName**: The method that issued the message or trace event.
- **CWPKI0041W**: One or more keystores are using the default password.
  - **Explanation**: When the Application Server starts for the first time as a stand-alone application server or in a Network Deployment configuration, each server creates a keystore and truststore for the default Secure Sockets Layer (SSL) configuration. When the application server creates these files, by default, it uses `WebAS` for the password. Do not use the default password in production. The warning message suggests that you change the password.
  - **Action**: To eliminate this warning message, use the administrative console to change the default password for the keystore and the truststore, and also change these passwords by editing the `ssl.client.props` file. When you change the passwords in the `ssl.client.props` file, you must use the PropFilePasswordEncoder utility to re-encode the new passwords.

WebSphere Education

Reading a log or trace file (2 of 2)

- Timestamps provide good clues:
  - Timestamps are real system time values
  - Useful when comparing traces from different processes and correlating events of different servers
- Look for exceptions (search for exception from top of stack trace)
  - Events before exception are probable causes
  - Events after exception are recovery attempts
- Useful to follow a single thread
  - Use the Thread ID to gather related messages

Figure 3-31. Reading a log or trace file (2 of 2) WB8691.0

Notes:

While it is possible to read logs and trace files by hand by using only a text editor, it is suggested that you use a tool such as Log Analyzer, which is available in the IBM Support Assistant.

The following article gives a good introduction to interpreting trace data: `http://www.ibm.com/developerworks/websphere/techjournal/0704_supauth/0704_supauth.html`

3.3. High Performance Extensible Logging (HPEL)

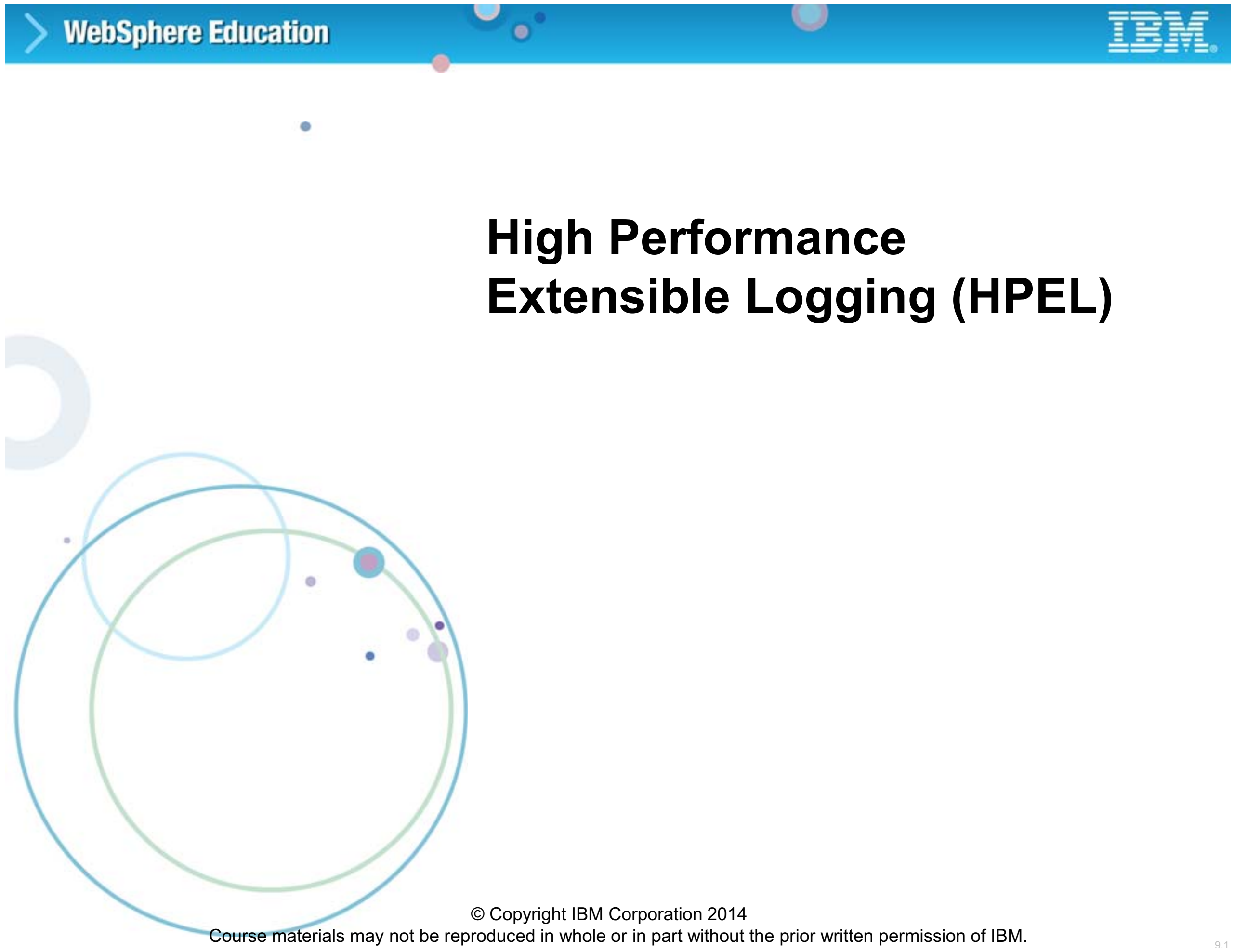


Figure 3-32. High Performance Extensible Logging (HPEL) WB8691.0

Notes:

WebSphere Education IBM

High Performance Extensible Logging (HPEL) (1 of 2)

- Performance
  - Log and trace events are stored as binary data so performance is substantially faster than default log and trace framework (basic mode)
  - Less impact to systems when trace is enabled
- Extensions
  - HPEL includes information about which application each log or trace record is from
  - When used with cross-component trace, HPEL includes information about which request each log or trace record is from
  - Developers can add their own log or trace record extensions
  - HPEL `logViewer` command enables filtering logs or trace data by extension name and value
- Filtering
  - HPEL `logViewer` command helps you filter logs and trace data by date, time, level, and thread ID

Figure 3-33. High Performance Extensible Logging (HPEL) (1 of 2) WB8691.0

Notes:

A number of factors contribute to the overall performance of HPEL logging and tracing.

Log and trace events are each stored in only one place.

Log events, `System.out`, and `System.err` are stored in the log data repository. Trace events are stored in the trace data repository. If the text log file is disabled, HPEL might write log and trace content only to these repositories. Storing each type of event in one place ensures that performance is not wasted on redundant data storage.

Log events, and optionally trace events, are written to the text log file when it is enabled. Since this data is always also stored in the log data and trace data repositories, the text log file content is redundant. The text log is convenient for users who do not want to run the LogViewer command-line tool to see their logs and trace; but you can disable the text log if this convenience is not needed.

Data is not formatted unless it is needed.

Formatting data for a user to read uses processor time. Rather than format log event and trace event data at run time, HPEL log and trace data is stored more rapidly in a proprietary binary representation. This fast binary storage improves the performance of the log and trace facility. By deferring log and trace formatting until the LogViewer is run, sections of the log or trace that are never viewed are never formatted.

Log and trace data is buffered before being written to disk.

WebSphere Education

High Performance Extensible Logging (HPEL) (2 of 2)

- Enable HPEL on any server in the cell
  - Deployment manager
  - Node agent
  - Application server
- Click **Troubleshooting > Logs and trace > *server_name***
  - Click **Switch to HPEL Mode**
- Is also available in stand-alone administrative console

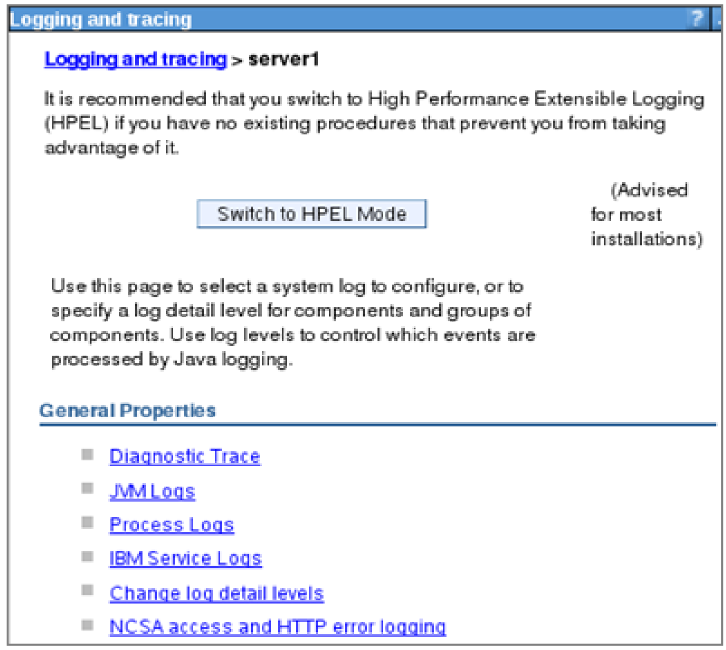


Figure 3-34. High Performance Extensible Logging (HPEL) (2 of 2) WB8691.0

Notes:

As soon as the log level is switched to HPEL mode for a server, new links appear in the General Properties section for configuring HPEL logging and tracing. These links are shown on the next slide. A new link appears to change log and trace mode, which allows you to switch back to basic logging.

WebSphere Education

HPEL logging and tracing configuration

- Use this page to configure HPEL logging and tracing

Logging and tracing

Logging and tracing > **server1**

General Properties

Configure HPEL logging

Directory	C:\WebSphere\AppServer\profiles\profile1/logs/server1
For cleanup, delete records older than	Disabled
For cleanup, maximum size of logs	50 Megabytes

Configure HPEL trace

Directory	C:\WebSphere\AppServer\profiles\profile1/logs/server1
For cleanup, delete records older than	Disabled
For cleanup, maximum size of trace	50 Megabytes

Configure HPEL text log

Current status:	Enabled
Directory	C:\WebSphere\AppServer\profiles\profile1/logs/server1
For cleanup, delete records older than	Disabled
For cleanup, maximum size of text log	50 Megabytes

Figure 3-35. HPEL logging and tracing configuration WB8691.0

Notes:

This screen capture shows the default configuration for HPEL logging and tracing. To modify the configuration, click any of the links on this page.

After HPEL is enabled for a server, it is good to disable the HPEL text logging. Disabling text logging improves server performance.

WebSphere Education IBM

HPEL log and trace storage

- HPEL provides a binary log data repository, a binary trace data repository, and a text log file
- When HPEL is enabled and configured, the user is no longer required to understand the details of the individual files to do analysis

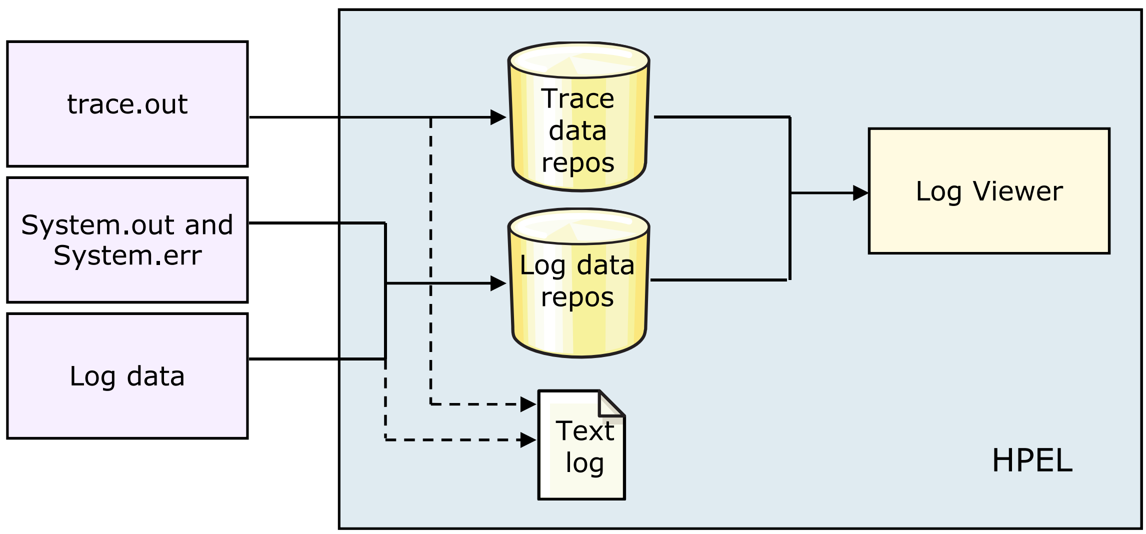


Figure 3-36. HPEL log and trace storage WB8691.0

Notes:

The user is no longer required to be aware of whether data is written to `SystemOut.log`, `SystemErr.log`, or `trace.log`. With HPEL, the logs are consolidated and the log viewer can be used to view all the data, or it can be filtered according to what subset of messages are needed.

The system still uses process logs, `native_stdout.log` and `native_stderr.log`, and they are found in the default server logs directory.

They are not viewable with the HPEL Log Viewer.

WebSphere Education

Log files and locations for HPEL

Log files	Location
`Logdata` (directory)	*<was_root>*/profiles/*<profile_name>*/logs/*<server_name>*
`Tracedata` (directory)	*<was_root>*/profiles/*<profile_name>*/logs/*<server_name>*
`TextLog.log`	*<was_root>*/profiles/*<profile_name>*/logs/*<server_name>*
`startServer.log`	*<was_root>*/profiles/*<profile_name>*/logs/*<server_name>*
`stopServer.log`	*<was_root>*/profiles/*<profile_name>*/logs/*<server_name>*
`native_stdout.log`	*<was_root>*/profiles/*<profile_name>*/logs/*<server_name>*
`native_stderr.log`	*<was_root>*/profiles/*<profile_name>*/logs/*<server_name>*
`btrace.1`	*<was_root>*/profiles/*<profile_name>*/logs/*<server_name>*

Note: `btrace.1` is used for the binary trace facility of Intelligent Management, which is covered in a later unit

Figure 3-37. Log files and locations for HPEL WB8691.0

Notes:

HPEL log data repository:

The log data repository is a storage facility for log records. Log data is typically intended for review by administrators. This data includes any information that the server or applications write to `System.out`, `System.err`, or `java.util.logging` at level Detail or higher. (It includes Detail, Config, Info, Audit, Warning, Severe, Fatal, and any custom levels at level Detail or higher.)

HPEL trace data repository:

The trace data repository is a storage facility for trace records. Trace data is typically intended for use by application programmers or by the WebSphere Application Server support team. This data includes any information that the server or applications write to `java.util.logging` at levels below level Detail (including Fine, Finer, Finest, and any custom levels below level Detail).

HPEL text log:

The text log file is a plain text file for log and trace records. The text log file is provided for convenience, primarily so that log content can be read without having to run the LogViewer command-line tool to convert the log data repository content to plain text.

The text log file does not contain any content that is not also stored in either the log data repository or trace data repository. You can disable the text log to enhance server performance. The text log can be configured to record trace content for debugging convenience.

WebSphere Education

Contents of the logs directory

- The `logdata` directory contains the `standard.out`, `standard.err`, and log messages for each instance of the server
- The `tradedata` directory contains trace data for each instance of the server

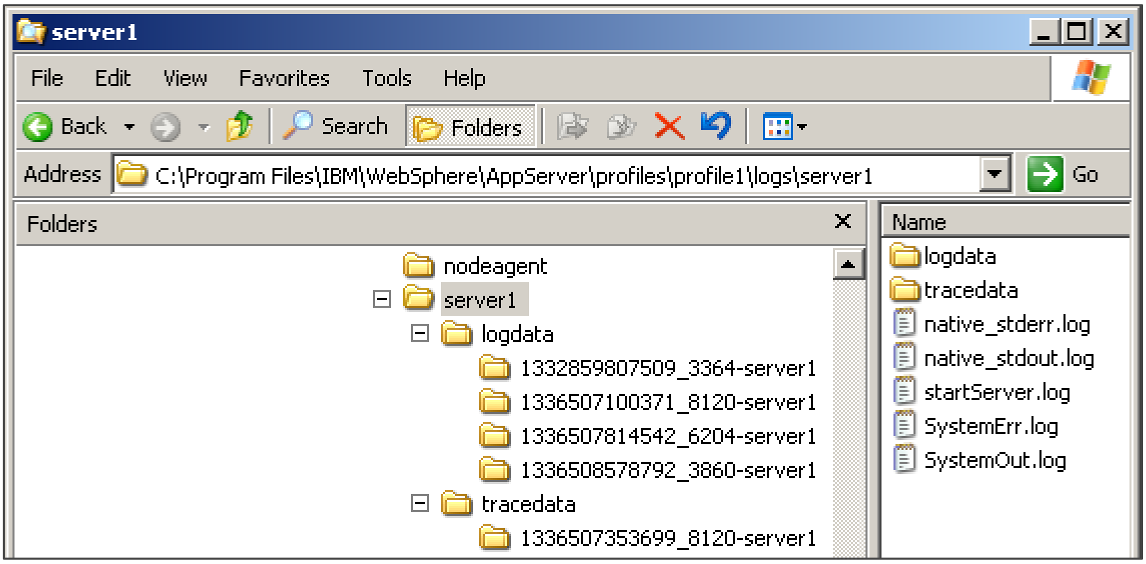


Figure 3-38. Contents of the logs directory WB8691.0

Notes:

In the `logdata` directory, there is a subdirectory for each instance of the server. The subdirectory name contains the PID of the server instance.

The directory for server1 contains the `pid` file, `native_stderr`, `native_stdout`, `startServer`, and `stopServer` log files, as in previous versions of WebSphere Application Server.

WebSphere Education

Configure HPEL logging

- Clicking the **Configure HPEL logging** link starts the configuration page
- Changes that are made on the **Configuration** tab require a server restart
- Changes that are made on the **Runtime** tab take effect immediately

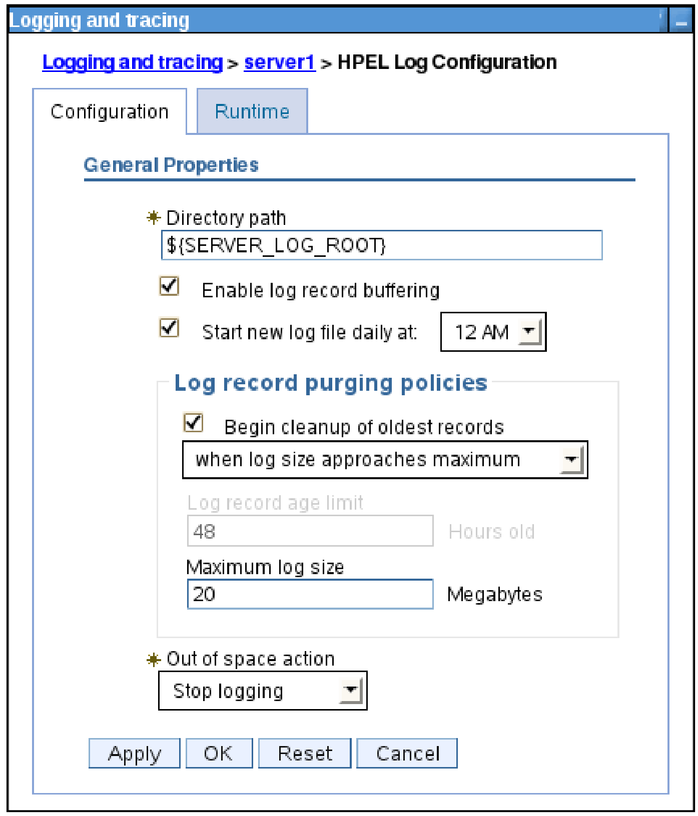


Figure 3-39. Configure HPEL logging WB8691.0

Notes:

Enable log record buffering: Specifies that the logging system avoids writing to disk each time a log record is created. The logging system creates a buffer that can hold a number of log records, and writes the buffered events when the buffer is full. The logging system also writes the buffered events after a few seconds pass, even if the buffer is not full. Selecting this setting significantly improves logging performance; however, if the server stops unexpectedly, the contents might not be written to the log repository.

Begin cleanup of oldest records: Specifies the log cleanup settings to be used to automatically purge the oldest log records, or log records that no longer fit in the configured space, from the log repository.

Out of space action: Specifies how the server reacts to an inability to add content to the log repository.

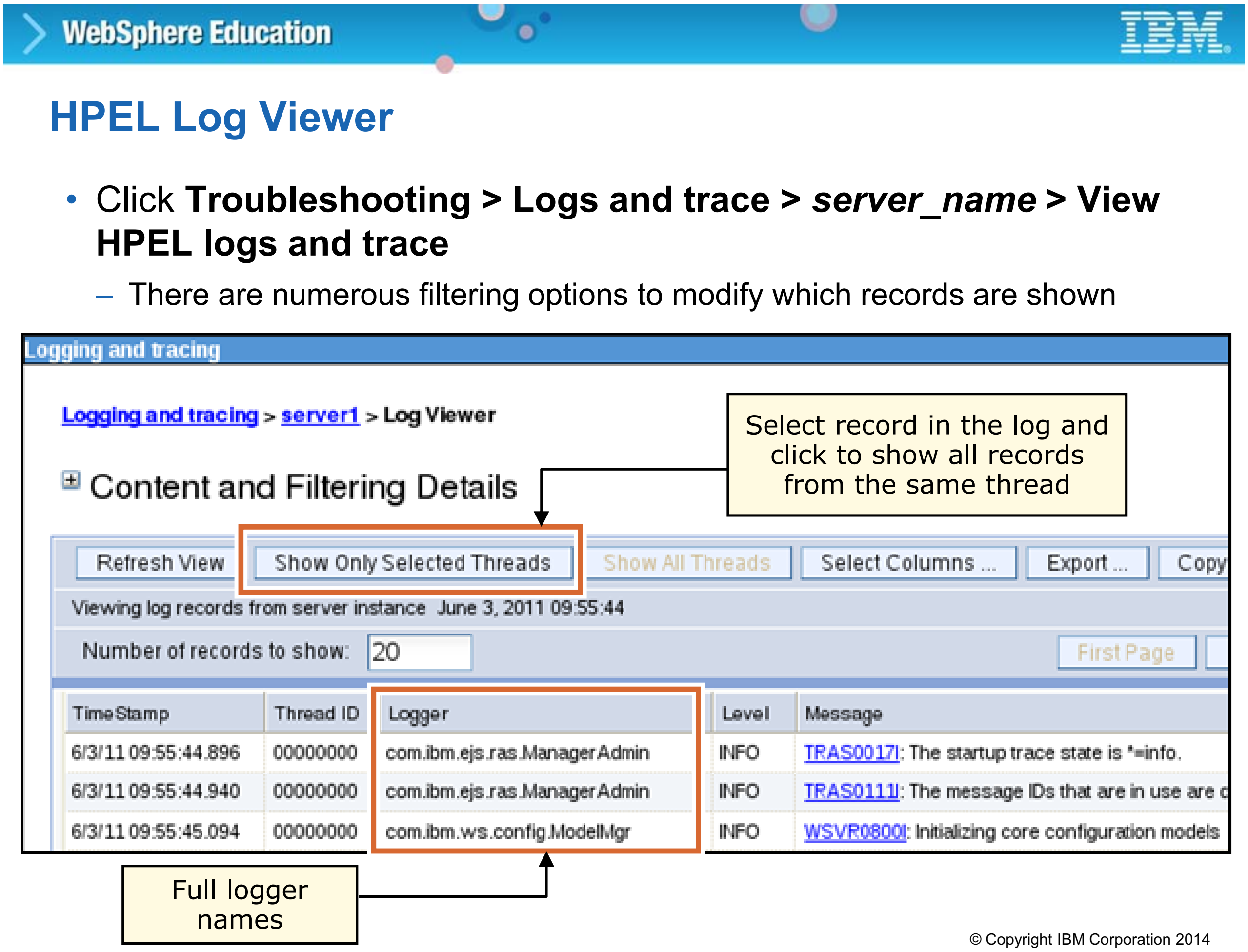


Figure 3-40. HPEL Log Viewer WB8691.0

Notes:

The log view section displays the records. Use the First Page, Previous Page, Next Page, and Last Page buttons to move through the list of records. You can also specify filter criteria in the Content and Filtering Details section to limit the rows that are displayed. Records are always displayed in the order that the server recorded them. By default the log view has five columns:

- **Time Stamp**: The time when the event was recorded.
- **Thread ID**: The identity of the thread that recorded the event in hexadecimal notation.
- **Logger**: The logger that recorded the event.
- **Level**: The type of event that was recorded.
- **Message**: The message from the recorded event. If the message has a message ID, the message ID is underlined. Click the message ID to get an explanation and suggested user action for the message.

To manipulate the log view, you can use available buttons to complete the following actions:

- **Refresh View**: Clears the contents of the viewer and reinitializes the view by using records from the server. Use this button to retrieve information about any additional rows that are created since the log viewer was started.
- **Show Only Selected Threads**: Filters any records that are created by any thread other than the one selected in the selection area. Clicking this button enables the Show All Threads button.
- **Show All Threads**: Displays any records that were filtered when you clicked Show Only Selected Threads. This button is enabled only when you use the Show Only Selected Treads button to restrict the view.
- **Select Columns**: Enables you to select the columns in the viewer that you want to view.
- **Export**: Exports logs to a local workstation in any of basic, advanced, or binary (HPEL) formats.
- **Copy to Clipboard**: Copies the records that are highlighted in the selection area into the operating system clipboard.
- **Server Instance Information**: Displays attributes for the selected server instance process. Use this table to find attributes and corresponding values for the server instance process environment. These properties are similar to the ones found in the header of basic mode logs.

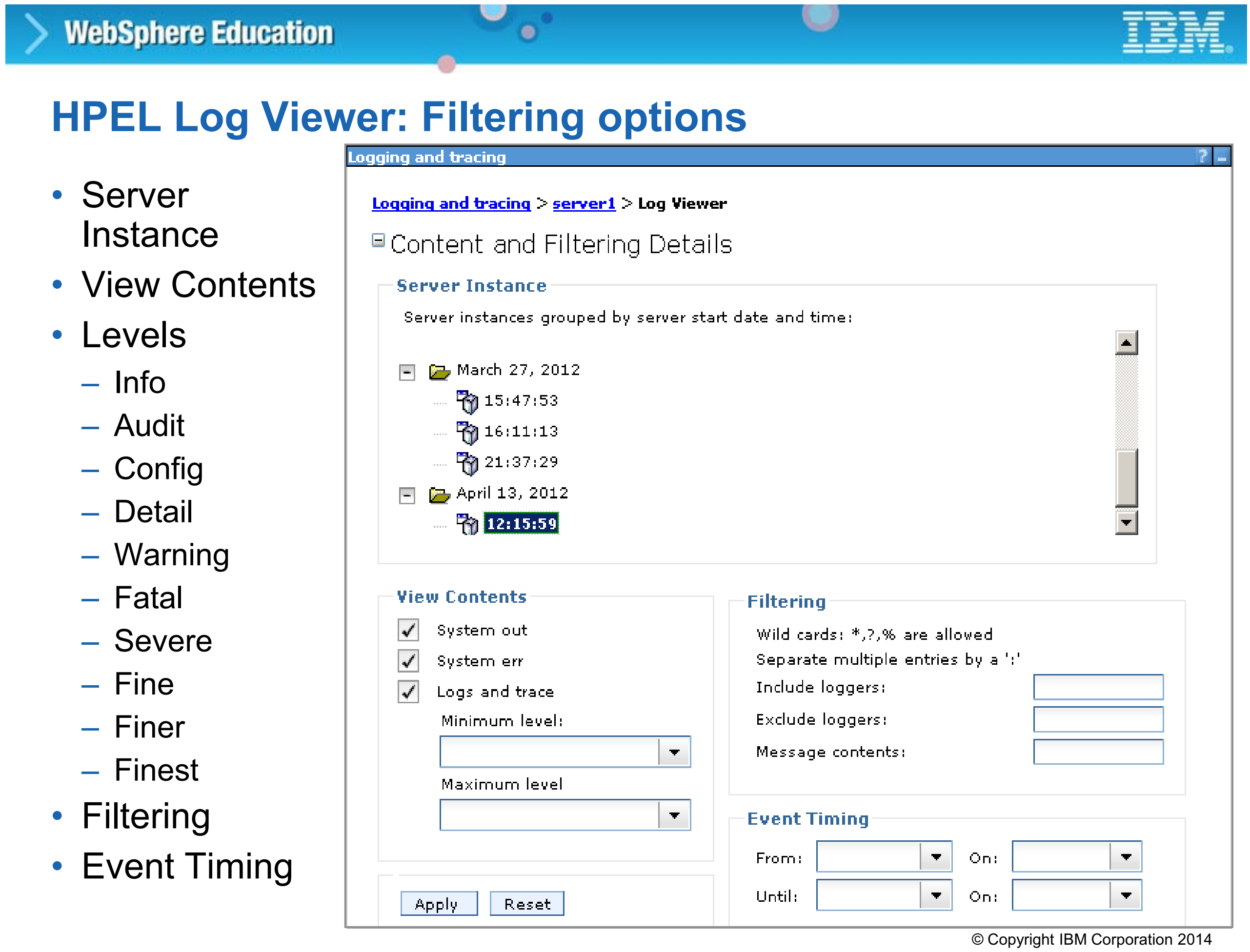


Figure 3-41. HPEL Log Viewer: Filtering options WB8691.0

Notes:

- **Server instance**: A server instance represents a run of a server process. Each time the server is restarted, a new server instance is created. By default, the log view table shows log records that are generated in the most recent server instance. To select a different server start time, choose a server instance with the appropriate start time stamp. The time stamps that are shown represent the time stamp of the first record that is written to each server instance. Select a particular server instance to change the server instance from which log records are retrieved.
- **View Contents**: Controls what content sources are displayed in the log view.
- **System Out**: Specifies that content that is logged to the `System.out` output stream is included in the log view.
- **System Error**: Specifies that content that is logged to the `System.err` output stream is included in the log view.
- **Logs and trace**: Specifies that log and trace records are included in the log view. Log and trace entries can be further specified to include a minimum or maximum level. Minimum and maximum can be specified together, for example to display only a certain level of trace. If you do not select Logs and trace, then no log or trace records of any severity can be displayed.

- **Filtering**: Controls which records are included in and excluded from the log view. For all filters in this section, a colon (:) can be used as a separator character to specify multiple entries. A limited set of regular expression characters can be used. Refer to console documentation for more details. If multiple filter settings are specified, the filter conditions must all be true for a record to be displayed in the log view.
- **Include loggers**: Specifies the list of loggers whose records are included in the log view.
- **Exclude loggers**: Specifies the list of loggers whose records are excluded from the log view.
- **Message contents**: Specifies the message content that each record must contain to be included in the log view.
- **Event timing**: Controls what records are displayed in the log view as based on a start and end date and time.

WebSphere Education

HPEL LogViewer: Command-line tool (1 of 2)

- Use the `LogViewer` command to query the contents of the HPEL log and trace repositories
  - `logViewer.sh/bat`
- You can also use the `LogViewer` command to view new log and trace repository entries as the server writes content to them
  - `logViewer.sh/bat -monitor [interval]`
- The are many command-line options for filtering the log and trace data
  - `-format` Specifies the output format
  - `-thread` Shows log entries from a specific thread
  - `-latestInstance` Gets log and trace data from the recent server instance
  - `-level` Extracts log entries from the specified level
  - `-minLevel` Shows log entries that are below the specified level
  - `-maxLevel` Shows log entries that are above the specified level
  - Others

Figure 3-42. HPEL LogViewer: Command-line tool (1 of 2) WB8691.0

Notes:

The first tool for analyzing HPEL logs is the command-line log viewer. This tool is simple, intuitive, and fast for doing analysis on the logs in problem determination efforts. The user can be unaware of whether data is written to `SystemOut.log`, `SystemErr.log`, or `trace.log`. With HPEL, the logs are consolidated and the log viewer can be used to view all the data, or it can be filtered according to what subset of messages are needed.

- `-monitor [interval]`

  Specifies that you want the LogViewer to continuously monitor the repository and output new log record entries as they are created. You can provide an optional integer argument after this parameter to specify how often you want the LogViewer tool to query the repository for new records. By default the LogViewer queries the repository for new records every 5 seconds. When used with other filtering options, only those new records that match the filter criteria are displayed.

- `-level FINEST | FINER | FINE | DETAIL | CONFIG | INFO | AUDIT | WARNING | SEVERE | FATAL`

Specifies that LogViewer must extract log entries only from the specified level. If combined with `-minLevel` or `-maxLevel`, the last option or options are used.

- `-minLevel FINEST | FINER | FINE | DETAIL | CONFIG | INFO | AUDIT | WARNING | SEVERE | FATAL`

  Specifies that LogViewer must not show log entries that are below the specified level. Specifying a level extracts all messages at that level and the ones above it.

- `-maxLevel FINEST | FINER | FINE | DETAIL | CONFIG | INFO | AUDIT | WARNING | SEVERE | FATAL`

  Specifies that LogViewer must not show log entries that are above the specified level. Specifying a level extracts all messages at that level and the ones below it.

- `-format <basic | advanced | CBE-1.0.1>`

  Specifies the output format. Supported formats include basic, advanced, and the `CBE-1.0.1` format. If you do not include this parameter, the output is in basic format.

- `-thread <thread_id>`

  Displays log entries from a specific thread. This option filters out any log messages that were not created according to the thread ID that you specified. Note: Specify the thread ID in hexadecimal format.

- `-latestInstance`

  Retrieves the log and trace data from the most recent server instance.

WebSphere Education

HPEL LogViewer: Command-line tool (2 of 2)

- To view log and trace data in ADVANCED format for the latest server instance and a specific thread ID
  - `logViewer.sh -lastestInstance -format Advanced -Thread 88`

```
[root@washost bin]# ./logViewer.sh -latestInstance -format Advanced -Thread 88

Using /opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1 as repository directory.

. ************ Start Display Current Environment ************
.    < Environment data removed>
.
************ End Display Current Environment ************
[8/1/13 12:49:25:878 EDT] 00000088  I UOW= source=com.ibm.ws.wsgroup.p2p.P2P
    Group org=IBM prod=WebSphere component=Application Server thread=[Thread-63]
       ODCF8040I: Detected process washostCell\washostNode01\nodeagent started.

[8/1/13 12:49:25:879 EDT] 00000088  I UOW= source=com.ibm.ws.wsgroup.p2p.P2P
    Group org=IBM prod=WebSphere component=Application Server thread=[Thread-63]
      ODCF8040I: Detected process washostCell\washostCellManager\dmgr started.
Operation Complete
Processed 2 records in 0.052 seconds (38.462 records per second).
[root@washost bin]#
```


Figure 3-43. HPEL LogViewer: Command-line tool (2 of 2) WB8691.0

Notes:

With the advanced format, complete logger names can be seen as `UOW=source`.

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
  - 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

Figure 3-44. What is Cross Component Trace (XCT)? WB8691.0

Notes:

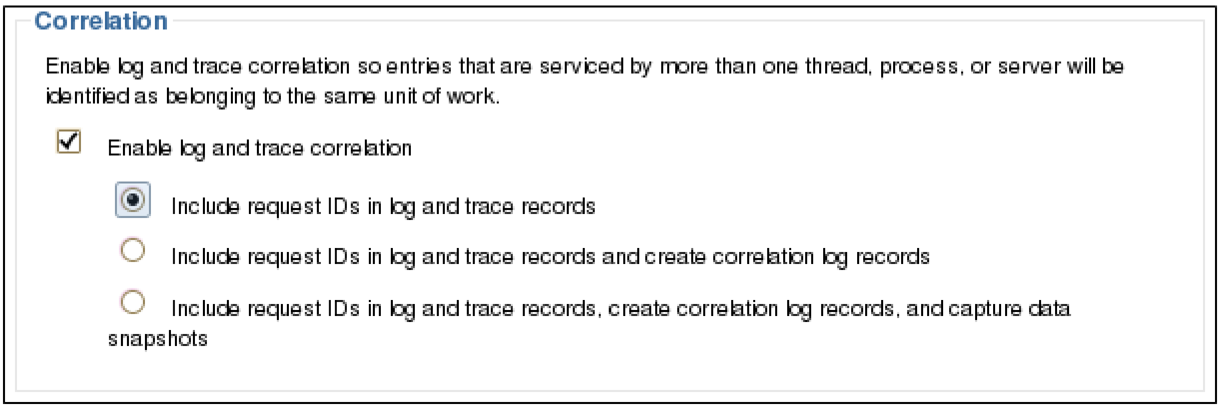
Depending on the nature of your applications, multiple threads within an application server can be used to handle requests, such as HTTP requests or JMS requests. More than one application server might handle some requests; for example, when one application server makes a request to another application 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

Administering XCT

- XCT can be enabled for a server that uses HPEL or Basic mode
- Click **Troubleshooting > Logs and trace >** ***server_name*** **> Change log detail levels**
- Check **Enable log and trace correlation**



- Select option for including request IDs, creating correlation logs, and capturing data snapshots

Figure 3-45. Administering XCT WB8691.0

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 it 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 it is best suited to test and development environments. XCT captures data snapshots for message requests and responses that the SIBus handles.

WebSphere Education

XCT request IDs

- XCT request IDs are identifiers added to log and trace records that the server produces
- XCT adds the same request ID to every log or trace record while the record is a part of the same request, regardless of which thread or JVM produces the log or trace entry
- When XCT is used with the HPEL log and trace infrastructure, you can view request IDs with the logViewer tool when logs are output in advanced format
  - `logViewer.sh -minLevel WARNING -format advanced`

```
[Time_stamp] 00000094  W UOW= source=com.ibm.ws.webcontainer.srt
class=com.ibm.ws.webcontainer.srt.SRTServletResponse method=setIntHeader
org= prod= component= thread=[WebContainer : 4]
requestID=[AAAsirk1Njr-AAAAAAAAAA+] appName=[PlantsByWebSphere]
```


Figure 3-46. XCT request IDs WB8691.0

Notes:

You can use XCT to augment your log and trace files with correlation information. This correlation information clarifies which threads and which application server processes participated in the handling of each request.

WebSphere Education IBM

Use XCT request ID information to track requests

- Use the HPEL LogViewer command-line tool to filter your logs by request ID
  - `logViewer.sh -includeExtensions requestID=AAAsirk1Njr-AAAAAAAAAA+`

```
[Time_stamp] 00000094 XCT           I   BEGIN AAAsirk1Njr-AAAAAAAAAA+
00000000000-cccccccccc2 HTTPCF(InboundRequest
/PlantsByWebSphere/javax.faces.resource/jsf.js.jsf
RemoteAddress(127.0.0.1) RequestContext(-957274864))

[Time_stamp] 00000094 srt           W
com.ibm.ws.webcontainer.srt.SRTServletResponse setIntHeader SRVE8094W:
WARNING: Cannot set header. Response already committed.

[Time_stamp] 00000094 XCT           I   END   AAAsirk1Njr-AAAAAAAAAA+
00000000000-cccccccccc2 HTTPCF(Request AsyncWrite RequestContext(-
957274864))
```


Figure 3-47. Use XCT request ID information to track requests WB8691.0

Notes:

XCT log records:

XCT log records are typically added to the logs to:

- Demarcate the beginning and end of work for a particular request on a particular thread
- Demarcate when work is about to transfer to another thread or process, or to indicate when work returned from another thread or process
- Demarcate when work moves from major component to major component, even if work continues on the same thread; for example, to show transfer of control from application server code to application code

XCT log records are composed of:

- XCT type (BEGIN / END)
- XCT parent correlator ID (for example, `00000000000-cccccccccc2`)
- XCT current correlator ID (for example, `AAAsirk1Njr-AAAAAAAAAA+`)

- XCT annotations

  For example, `HTTPCF(InboundRequest/PlantsByWebSphere/javax.faces.resource/jsf.js.jsf RemoteAddress(127.0.0.1) RequestContext(-957274864))`

XCT tools:

The HPEL logViewer tool is able to filter log and trace records by request ID.

Tools such as the XCT Log Viewer can also take advantage of XCT log records or XCT request IDs, or both, when rendering log and trace content. The XCT Log Viewer is available as a tool add-on for the IBM Support Assistant.

3.4. Gathering JVM-related data

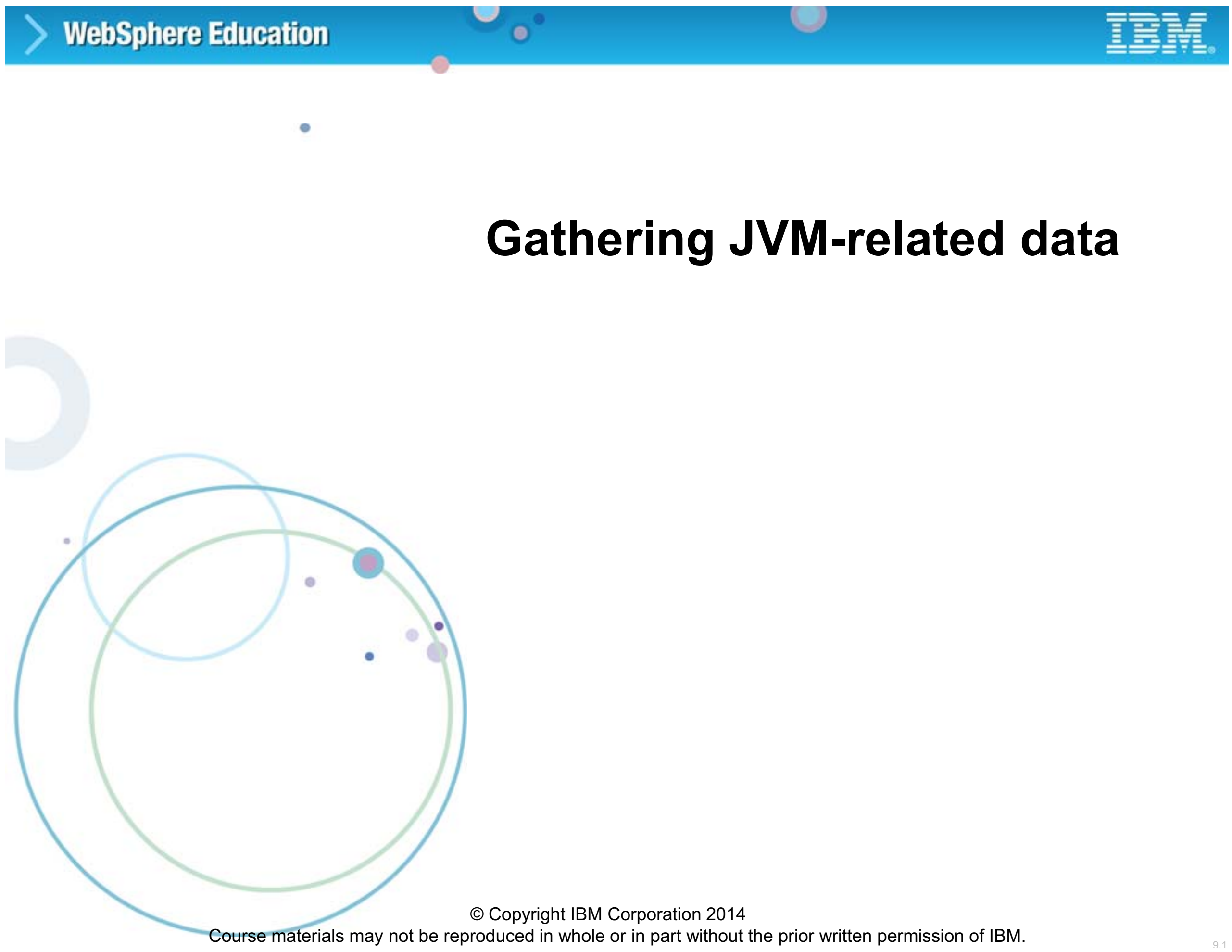


Figure 3-48. Gathering JVM-related data WB8691.0

Notes:

WebSphere Education

Enable verbose garbage collection

- The JVM runtime provides verbose GC as an option
- Enables a garbage collection log that provides data such as:
  – Interval between collections
  – Duration of collection
  – Whether compaction was required
  – Memory size, memory that is freed, memory available
- Enable the verbose GC by using the administrative console
  – **Servers > *server_name* > Process Definition > Java Virtual Machine**
  – Select **Verbose Garbage Collection** check box on configuration or runtime tab

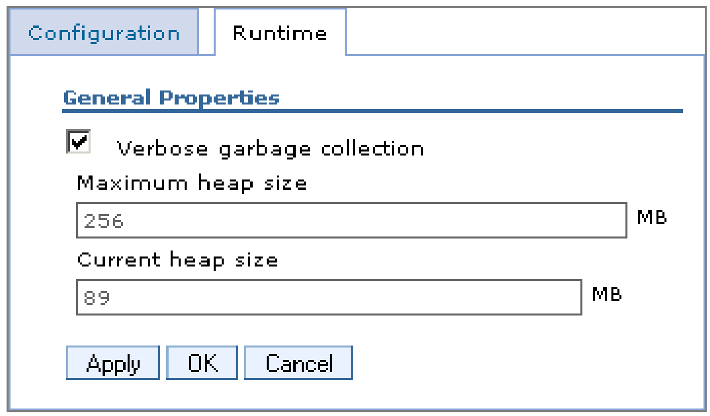


Figure 3-49. Enable verbose garbage collection WB8691.0

Notes:

It is often suggested that you have verbose GC enabled permanently in production. The performance cost on a reasonably well-tuned JVM is small. The benefits of having it on the first time something happens are considerable (no need to reproduce the problem a second time after enabling). It is also good to keep an eye on the verbose GC regularly as a way to monitor the health of the system, even when nothing bad is noticed.

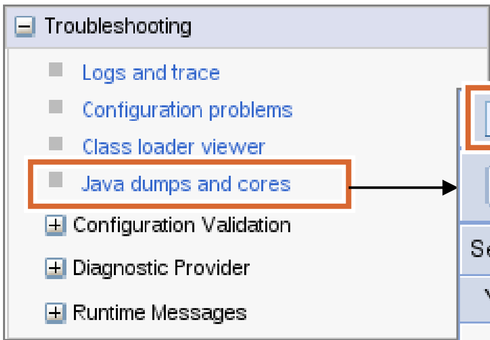
Enabling verbose GC is a decision that each system administrator must make conscientiously. But it is no longer plainly “not recommended as a normal production setting.”

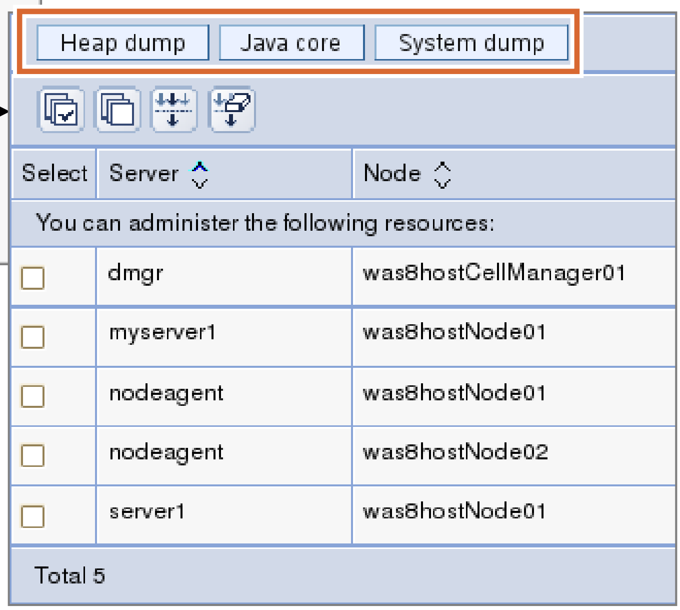
It usually writes to the `native_stderr.log` file. It varies depending on platform and WebSphere version. There is some processor cost because of disk I/O, but it is usually minimal unless thrashing.

WebSphere Education

Java dumps and cores

- New feature in the Troubleshooting section is **Java dumps and cores**





- Select servers and click
  - Heap dump
  - Javacore
  - System dump

Figure 3-50. Java dumps and cores WB8691.0

Notes:

Clicking the Java dumps and cores link starts the pane that is shown in the screen capture. Use this pane to generate heap dumps, javacores, or system dumps for a running process. Select the server and click the appropriate button for Heap dump, javacore (thread dump), or System dump (JVM core). The files that result from these operations are placed on the local file system; by default they are written to the profile root directory, `was_root/profiles/profile-name`.

- **Heap dump**: A heap dump is a snapshot of JVM memory. It shows live objects in the memory and references between them.
- **Javacore**: Use this button to investigate why a server is hanging or investigate messages in the logs that indicate a thread did not complete its work in the expected amount of time.
- **System dump**: Use this button to generate system native dumps of the server process. These memory dumps can be large.
- **A note on verbose garbage collection data**: As in previous versions, verbose GC is not enabled by default. When you enable verbose GC for a server in V8, the default garbage collection policy is generational-concurrent (gencon). The data is written to the

native_stderr.log or native_stdout.log file, depending on the operating system of the server.

WebSphere Education

Dumping the JNDI name space

- dumpNameSpace utility shows JNDI directory content
- Useful to ensure correct association of named objects:
  - JDBC resources
  - EJBs
  - JMS resources
  - Other resources
- Syntax and some of the options:

```
<was_root>/bin/dumpNameSpace
   [-host bootstrap_host_name (defaults to localhost)]
   [-port bootstrap_port_number (defaults to 2809)]
   [-startAt subcontext/in/the/tree]
```

- Output can be redirected to a file and inspected

Figure 3-51. Dumping the JNDI name space WB8691.0

Notes:

- Usage: `dumpNameSpace [-keyword value]`
  - If a keyword occurs more than once with different values, the value that is associated with the last occurrence is used.
  - The keywords and associated values are: `-host myhost.austin.ibm.com`
    - Bootstrap host is the WebSphere host whose namespace you want to dump. It defaults to “localhost.”
    - `-port nnn`
      - Bootstrap port, defaults to 2809.
    - `-factory com.ibm.websphere.naming.WsnInitialContextFactory`
      - The initial context factory to be used to get the JNDI initial context. It defaults as shown and normally does not need to be changed.
- `-root [ cell | server | node | host | legacy | tree | default ]`

WebSphere Education IBM

Dumping the JNDI name space example

- Enter command: **`dumpNameSpace -root server -port 9810`**

```
==============================================================================
Name Space Dump
  Context factory: com.ibm.websphere.naming.WsnInitialContextFactory
  Provider URL: corbaloc:iiop:localhost:9810
  Requested root context: server
  Starting context: (top)=was8host01Cell01/nodes/was8host01Node01/servers/server1
  Formatting rules: jndi
  Time of dump: Fri Apr 13 12:27:55 EDT 2012
==============================================================================

==============================================================================
Beginning of Name Space Dump
==============================================================================
   1 (top)
   2 (top)/com                                   javax.naming.Context
   3 (top)/com/ibm                               javax.naming.Context
   4 (top)/com/ibm/websphere                     javax.naming.Context
  ...
  28 (top)/jdbc                                  javax.naming.Context
  29 (top)/jdbc/lrsched                          javax.resource.cci.ConnectionFactory
  30 (top)/jdbc/pgc                  j           javax.resource.cci.ConnectionFactory
  31 (top)/jdbc/DefaultEJBTimerDataSource        javax.resource.cci.ConnectionFactory
  32 (top)/jdbc/PlantsByWebSphereDataSource      javax.resource.cci.ConnectionFactory
  ...
  41 (top)/jta                                   javax.naming.Context
  42 (top)/jta/usertransaction                   java.lang.Object
  43 (top)/ejb                                   javax.naming.Context
  44 (top)/ejb/ivtEJBObject                      com.ibm.websphere.ivt.ivtEJB.ivtEJBHome
  45 (top)/eis                                   javax.naming.Context
  46 (top)/eis/DefaultDatasource_CMP             javax.resource.cci.ConnectionFactory
```


Figure 3-52. Dumping the JNDI name space example WB8691.0

Notes:

The figure shows an example of the JNDI namespace dump. Note: the entire memory dump is not shown on the slide because of space limitations.

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Identify and access several resources for problem investigation
- Identify, locate, and configure server log files
- Enable HPEL logging for a server and use log viewer tools
- Describe and use cross-component trace (XCT)
- Generate JVM diagnostic data by using the administrative console and other tools

Figure 3-53. Unit summary WB8691.0

Notes:

WebSphere Education

Checkpoint questions

1. Trace output can be directed to ___________.
   a) File or web page
   b) Web page or dashboard
   c) File or ring buffer
   d) Dashboard or ring buffer

2. True or false: Tracing cannot be started while the server is running.

Figure 3-54. Checkpoint questions WB8691.0

Notes:

Write down your answers here:

1.

2.

WebSphere Education

Checkpoint answers

1. (c). File or ring buffer
2. False: Tracing *can* be started while the server is running.

Figure 3-55. Checkpoint answers WB8691.0

Notes:

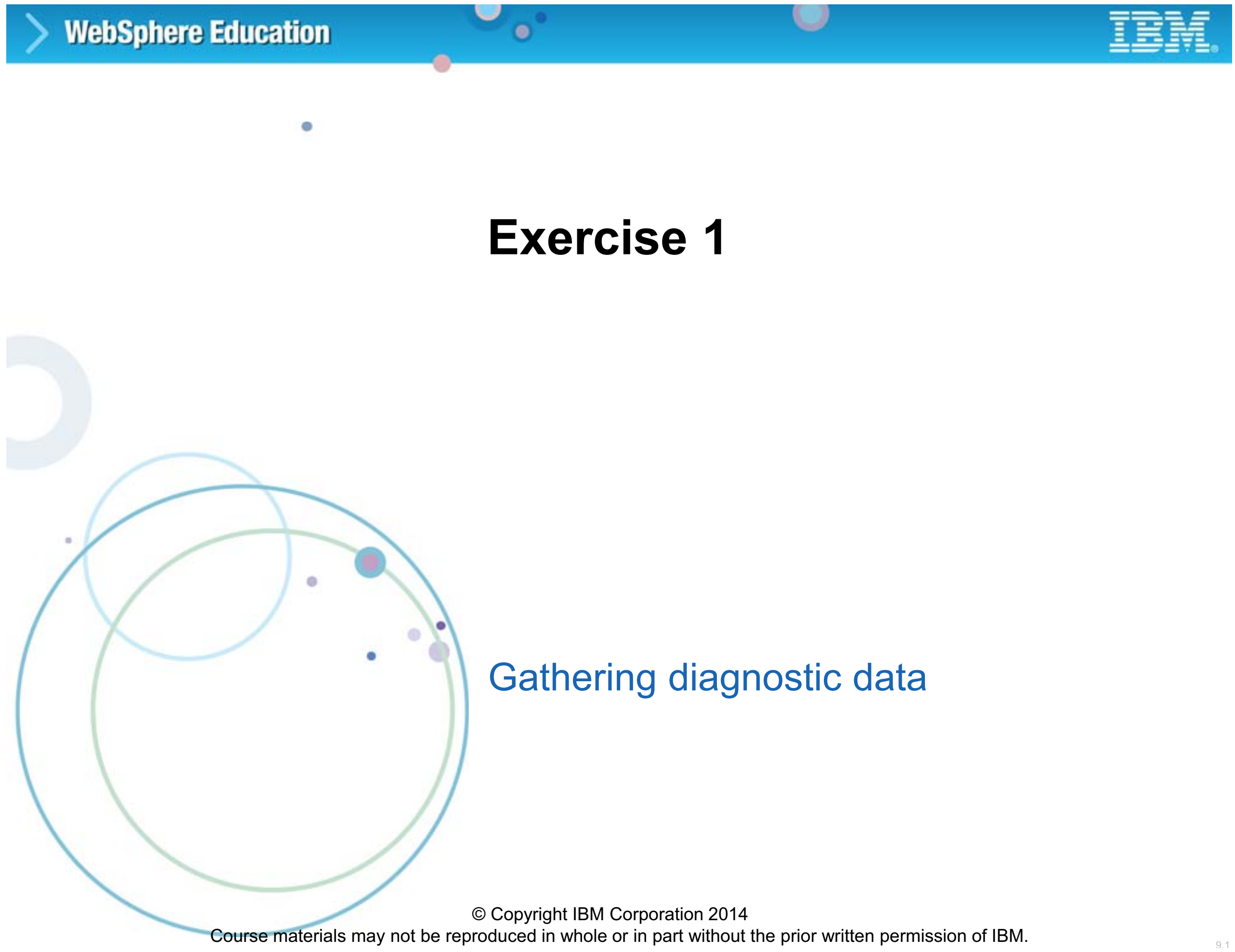


Figure 3-56. Exercise 1 WB8691.0

Notes:

WebSphere Education IBM

Exercise objectives

After completing this exercise, you should be able to:

- Enable tracing on a WebSphere Application Server component and read a trace output log
- Use the administrative console troubleshooting section to view runtime messages and configuration problems
- Enable and configure High Performance Extensible Logging (HPEL)
- Use the class loader viewer in the administrative console
- Enable and use diagnostic providers in the administrative console
- Examine runtime messages and first-failure data capture (FFDC) logs
- Use the dumpNameSpace tool to dump the contents of a namespace

Figure 3-57. Exercise objectives WB8691.0

Notes:

Unit 4. MustGather data and IBM Support Assistant

What this unit is about

This unit covers the IBM Support Assistant tool and MustGather data. IBM Support Assistant is a free tool that IBM Support provides. It includes different troubleshooting tools that can help you collect and analyze data for troubleshooting. The term “MustGather data” refers to diagnostic data that IBM Support uses to resolve issues. You can collect MustGather data for Business Process Manager by using the IBM Support Assistant Data Collector.

What you should be able to do

After completing this unit, you should be able to:

- Describe MustGather data
- Describe IBM Support Assistant
- Describe various IBM Support Assistant tools for troubleshooting
- Describe the Data Collector for IBM Business Process Manager

How you will check your progress

- Checkpoint

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Describe MustGather data
- Describe IBM Support Assistant
- Describe various IBM Support Assistant tools for troubleshooting
- Describe the Data Collector for IBM Business Process Manager

Figure 4-1. Unit objectives WB8691.0

Notes:

WebSphere Education IBM

Topics

- MustGather data for troubleshooting
- IBM Support Assistant
- IBM Support Assistant Data Collector for IBM Business Process Manager

Figure 4-2. Topics WB8691.0

Notes:

4.1. MustGather data for troubleshooting

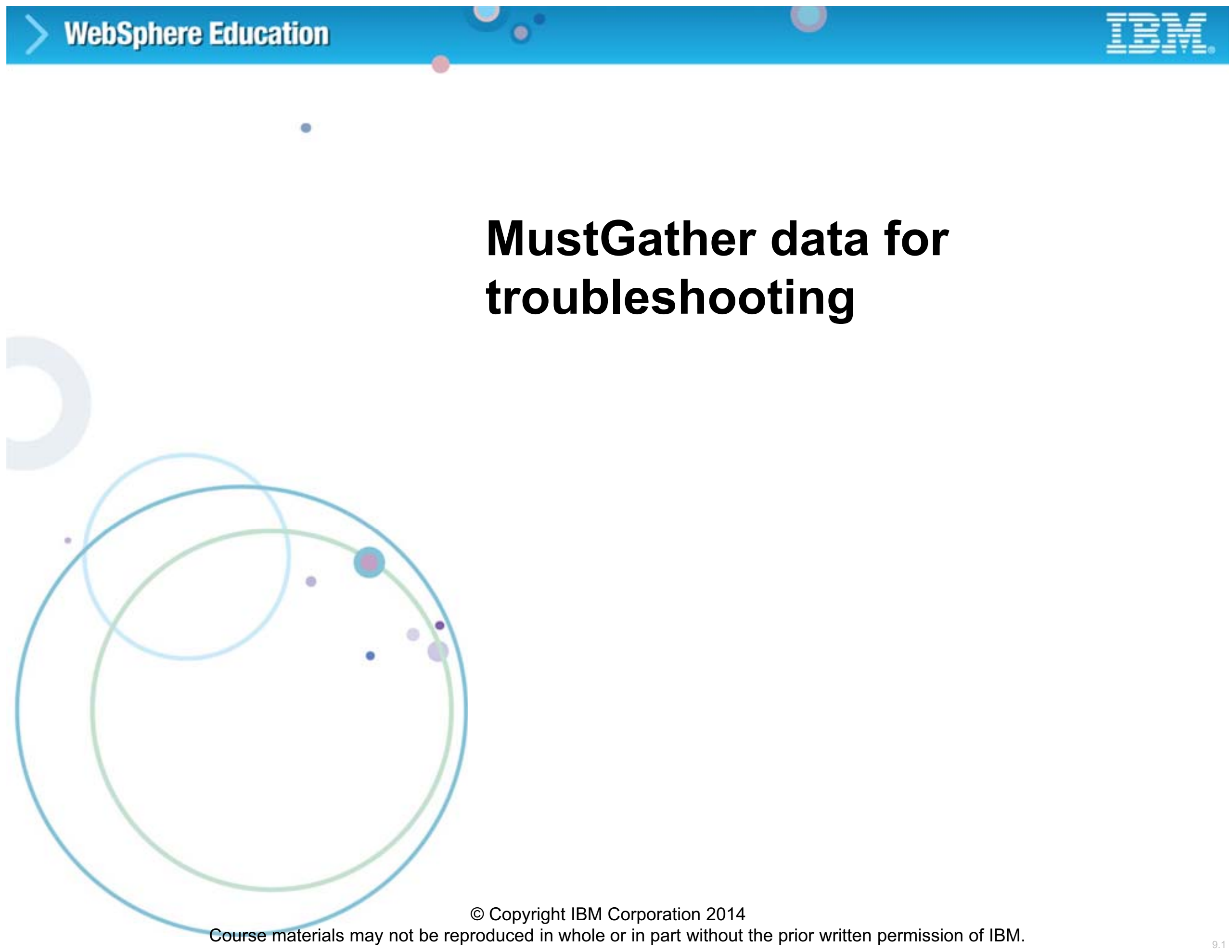


Figure 4-3. MustGather data for troubleshooting WB8691.0

Notes:

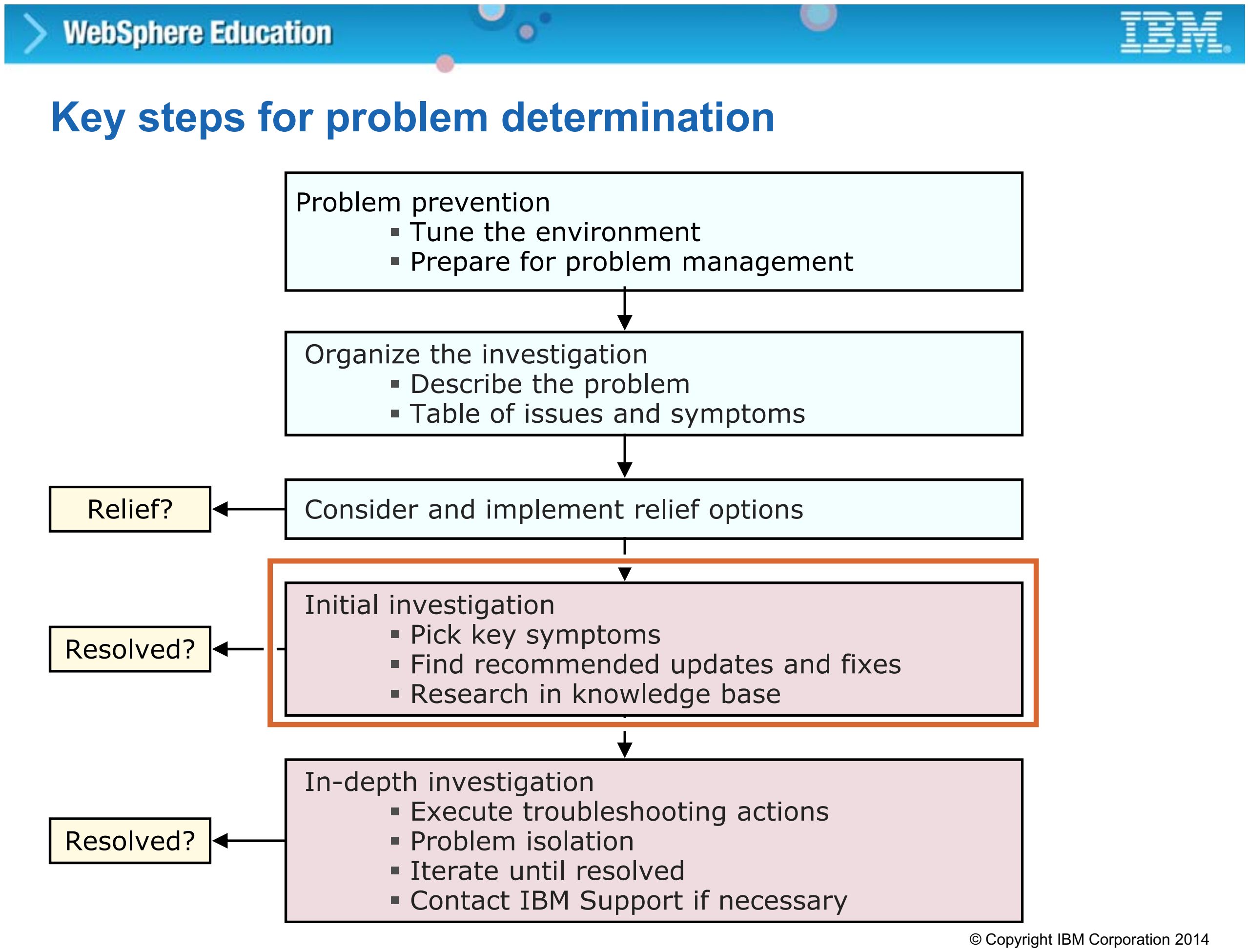


Figure 4-4. Key steps for problem determination WB8691.0

Notes:

WebSphere Education

MustGather data

- Before calling IBM support, gather troubleshooting data (***MustGather*** data) for problems with IBM Business Process Manager
  - The term “MustGather” represents the diagnostic data that is required to resolve a problem such as system information, problem symptoms, and log and trace files
- By collecting MustGather data early, you help IBM support to quickly determine the following information:
  - Whether symptoms match known problems
  - Is the issue a nondefect problem that can be identified and resolved
  - Identify a workaround for a defect to reduce severity
  - Whether locating the root cause can speed development of a code fix
- Collecting diagnostic data (***MustGather***) aids in problem determination and saves time when resolving Problem Management Records (PMRs) for IBM Business Process Manager

Figure 4-5. MustGather data WB8691.0

Notes:

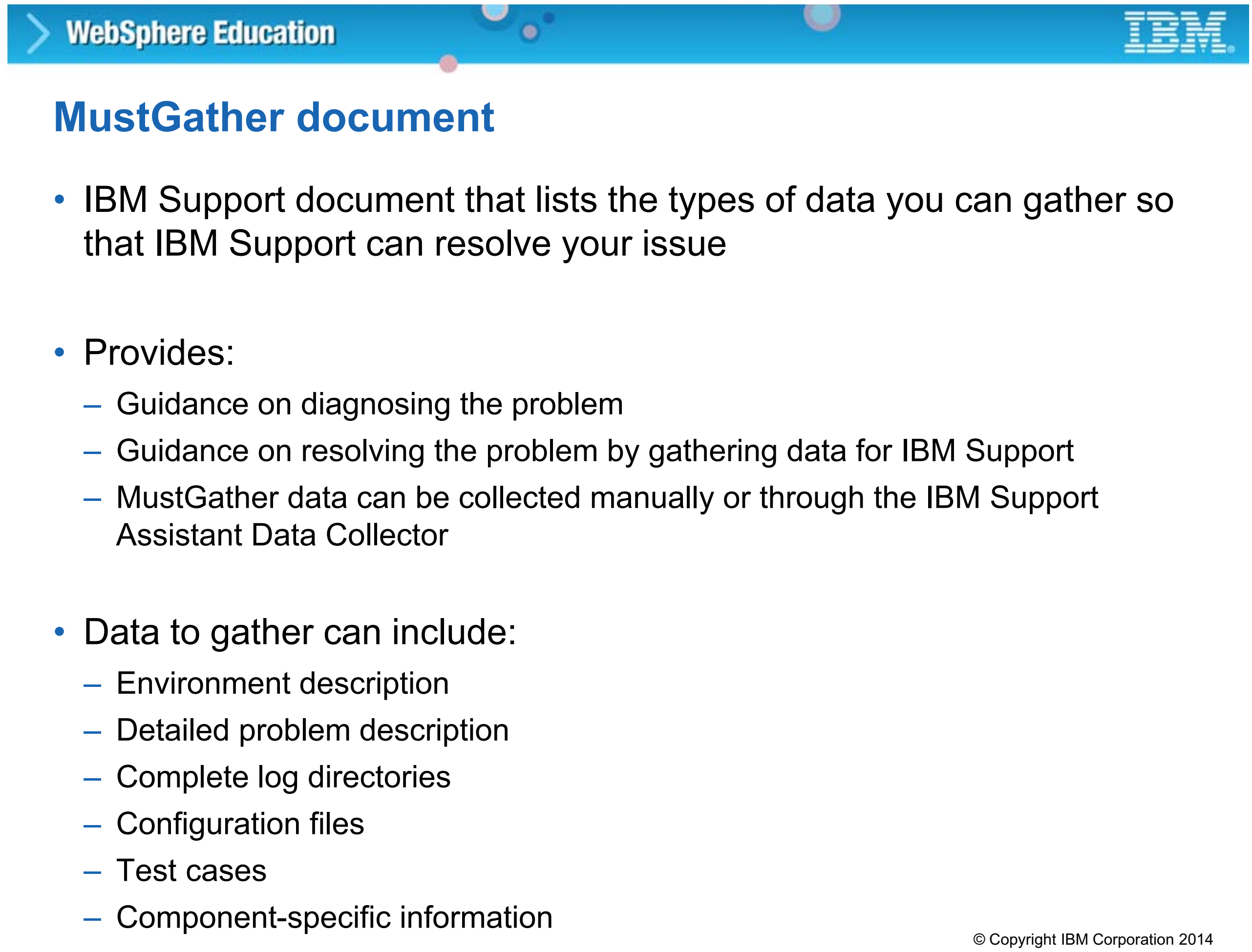


Figure 4-6. MustGather document WB8691.0

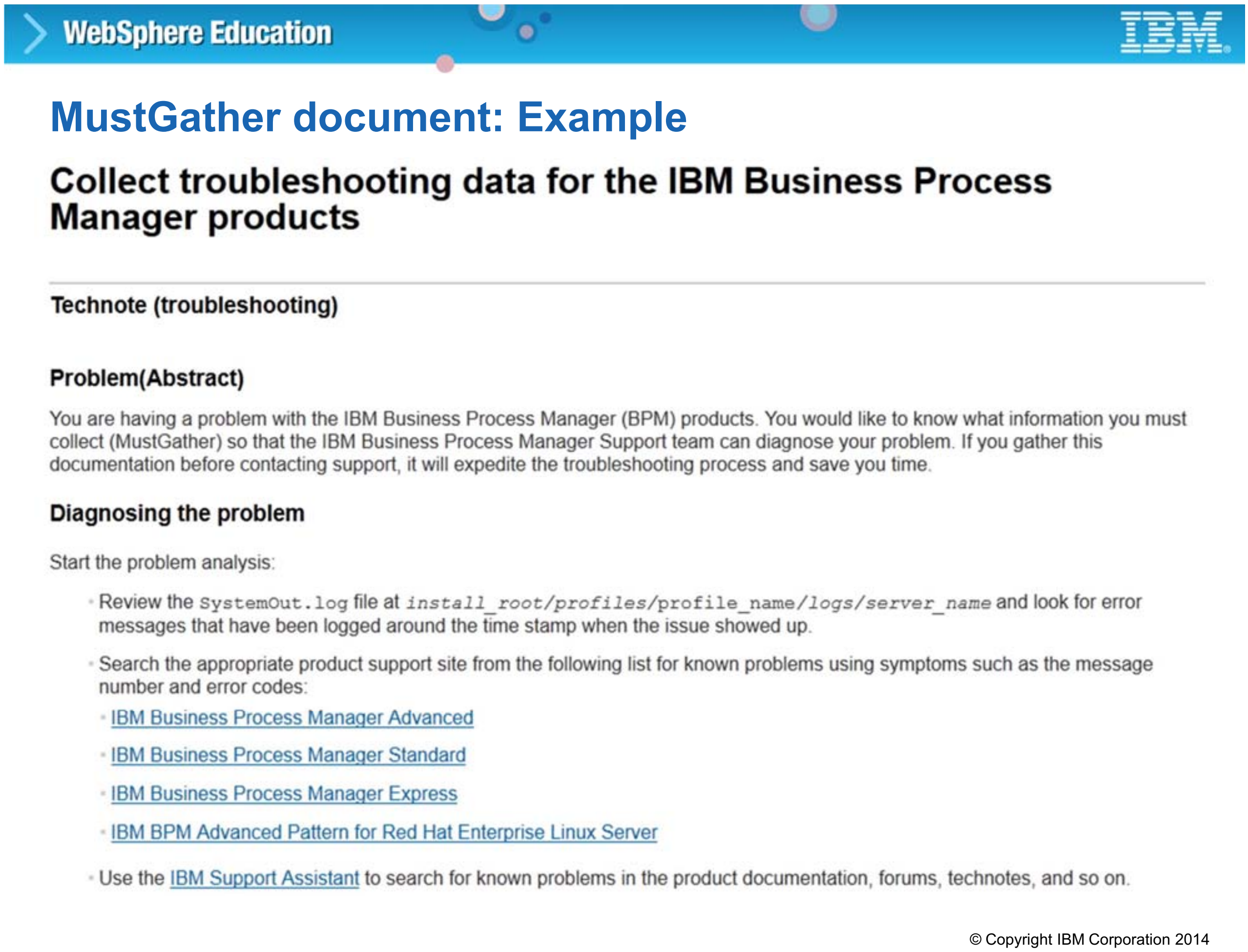
Notes:

MustGather documents can be accessed from within IBM Support Assistant or on the IBM Support website.

The screen capture displays a MustGather document for the IBM Business Process Manager family of products. This page shows how to collect data for Business Process Manager.

You can also click the links for the specific editions of the product, such as IBM Business Process Manager Advanced or IBM Business Process Manager Standard. These links provide troubleshooting and support information for the product version that you use.

You can find this page at the following web address:
`www.ibm.com/support/docview.wss?uid=swg21569731`



WebSphere Education

MustGather document: Example

Collect troubleshooting data for the IBM Business Process Manager products

Technote (troubleshooting)

Problem(Abstract)

You are having a problem with the IBM Business Process Manager (BPM) products. You would like to know what information you must collect (MustGather) so that the IBM Business Process Manager Support team can diagnose your problem. If you gather this documentation before contacting support, it will expedite the troubleshooting process and save you time.

Diagnosing the problem

Start the problem analysis:

- Review the `SystemOut.log` file at *`install_root/profiles/profile_name/logs/server_name`* and look for error messages that have been logged around the time stamp when the issue showed up.
- Search the appropriate product support site from the following list for known problems using symptoms such as the message number and error codes:
  - IBM Business Process Manager Advanced
  - IBM Business Process Manager Standard
  - IBM Business Process Manager Express
  - IBM BPM Advanced Pattern for Red Hat Enterprise Linux Server
- Use the IBM Support Assistant to search for known problems in the product documentation, forums, technotes, and so on.

Figure 4-7. MustGather document: Example WB8691.0

Notes:

MustGather documents can be accessed from within IBM Support Assistant or on the IBM Support website.

The screen capture displays a MustGather document for the IBM Business Process Manager family of products. This page how to collect data for Business Process Manager.

You can also click the links for the specific editions of the product, such as IBM Business Process Manager Advanced or IBM Business Process Manager Standard. These links provide troubleshooting and support information for the product version that you use.

You can find this page at the following web address:
`www.ibm.com/support/docview.wss?uid=swg21569731`

WebSphere Education

MustGather data: Environment information

- IBM Business Process Manager product and version
  - Run the following command from the `bin` directory of your IBM Business Process Manager product:

    `versionInfo -fixpacks -ifixes`

- Operating system and version
- Database software and version
- Information about other involved software
- Environment description: Production/Development/Test/QA
- If the problem is occurring in a cluster, describe the cluster topology

Figure 4-8. MustGather data: Environment information WB8691.0

Notes:

The purpose of covering how to manually collect the MustGather data is to help you understand which data is important to the IBM support team to troubleshoot your environment.

This slide describes environment information to collect for the MustGather data.

•

•

WebSphere Education

MustGather data: Detailed problem description

- Can create project interchange file for specific test scenarios
  - Include detailed instructions for how to run your test scenario
- Provide detailed description of the issue and the steps to reproduce it
  - Include sample input values and provide screen captures, if applicable
  - What are the symptoms that you see? Which components or applications are involved? Can you reproduce the issue?
  - Do you see any error messages?
  - Did the current scenario work previously? If yes, what was changed?
  - How often does this problem occur? Do you see any indicators on why this issue happens?
- Describe how this issue affects your company
  - Upcoming deadlines
  - Impact on production, development, or testing
  - Any realized or potential effect on customers

Figure 4-9. MustGather data: Detailed problem description WB8691.0

Notes:

This slide describes detailed problem description information to collect for the MustGather data.

WebSphere Education

MustGather data: Log and configuration files

- Provide **SystemOut** and **SystemErr** logs
  - Can also include first failure data capture (FFDC) logs, if applicable
  - ***<install_root>/profiles/<profile name>/logs/<server name>***
- Provide configuration files: TeamWorksConfiguration.running.xml file, which is in the following locations:
  - Process Center profile: *install_root*/profiles/*profile_name*/config/cells/*cell_name*/nodes/*node_name*/servers/*server_name*/process-center
  - Process Server profile: *install_root*/profiles/*profile_name*/config/cells/*cell_name*/nodes/*node_name*/servers/*server_name*/process-server
  - Can include the complete *install_root*/profiles/*profile_name*/config directory, if the support team requests it

Figure 4-10. MustGather data: Log and configuration files WB8691.0

Notes:

This slide describes log and configuration information to collect for the MustGather data.

WebSphere Education

Generate versionReport.html

- Gather the installation information
  - Execute `<PS home>\bin\genVersionReport.bat` or `genVersionReport.sh` to generate `versionReport.html`

IBM WebSphere Product Installation Status Report

Report at date and time April 29, 2014 9:49:00 PM PDT

Installation

Product Directory	C:\IBM\BPM\ProcessCenter\v8.5
Version Directory	C:\IBM\BPM\ProcessCenter\v8.5\properties\version
DTD Directory	C:\IBM\BPM\ProcessCenter\v8.5\properties\version\dtd
Log Directory	C:\ProgramData\IBM\Installation Manager\logs

Product List

BPMPC	installed
ND	installed

Figure 4-11. Generate versionReport.html WB8691.0

Notes:

4.2. IBM Support Assistant

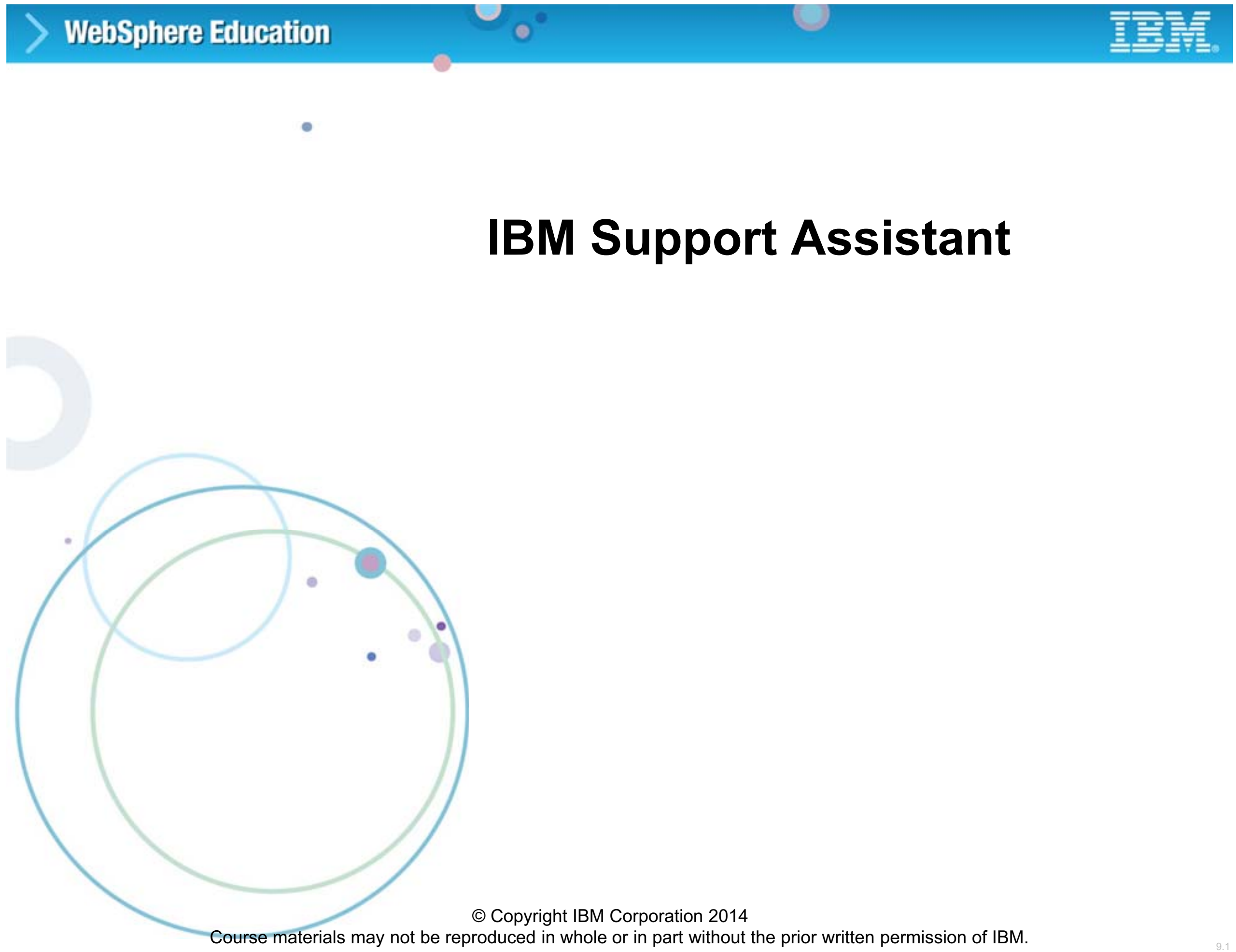


Figure 4-12. IBM Support Assistant WB8691.0

Notes:

WebSphere Education

IBM Support Assistant Team Server

- A group of users can all access a single, shared instance of IBM Support Assistant
- Installed as a single-user desktop application and can be accessed through a browser

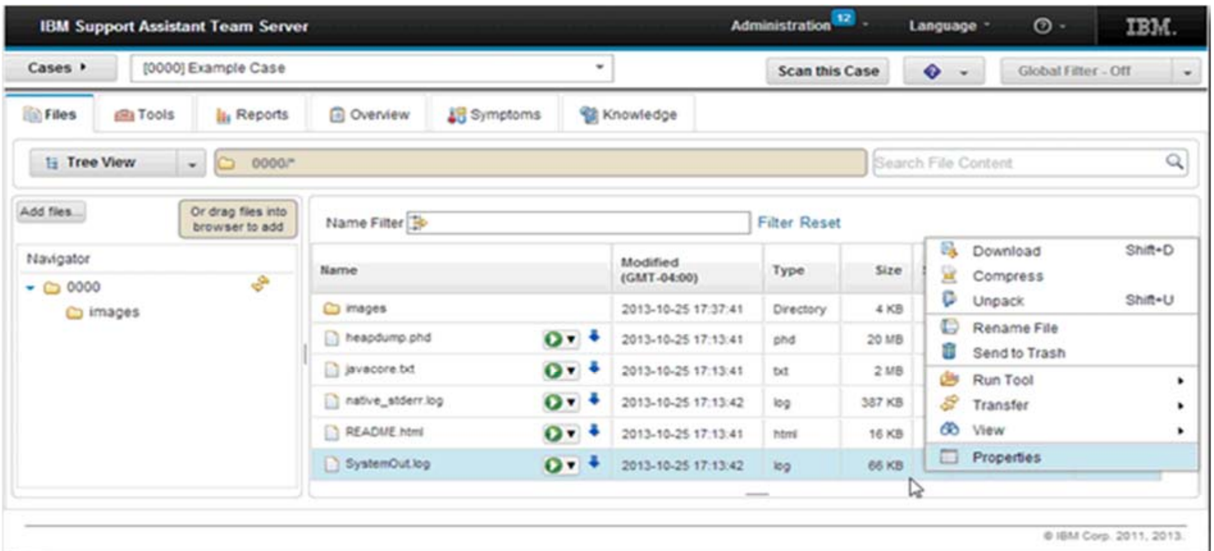


Figure 4-13. IBM Support Assistant Team Server WB8691.0

Notes:

IBM Support Assistant is a web-based application to assist with organizing, analyzing, and diagnosing issues with software. It applies the concept of cases to grouping problem diagnostic files together regarding an overlying issue. Using an intuitive interface, it provides easy case management, file management, problem determination capabilities, and automated data collection capabilities.

IBM Support Assistant V5 includes its own server that is easily configured and maintained. IBM Support Assistant can be installed as a team server so that a group of users can all access a single, shared instance of IBM Support Assistant by pointing their browsers to it. Alternatively, it can also be installed as a single-user desktop application and can be accessed through a browser by pointing to the local server.

The IBM Support Assistant Data Collector tool is included with the installation of Business Process Manager V8.5. Data Collector is a command-line tool that you can use to either interactively or automatically collect data. This data can help IBM support troubleshoot your issue.

WebSphere Education

IBM Support Assistant Team Server: Deployment options

- Embedded server
  - Install and update IBM Support Assistant and problem determination tools with IBM Installation Manager
- EAR deployment
  - Deploy an EAR file into your existing WebSphere Application Server
- Compressed file
  - Install IBM Support Assistant and problem determination tools in one package
  - Automatic updates to the application and tools are not available with this package

Figure 4-14. IBM Support Assistant Team Server: Deployment options WB8691.0

Notes:

WebSphere Education

IBM Support Assistant Team Server: Features and benefits (1 of 2)

- Allows you to easily open a browser to access the application and problem determination tools
- Install once and collaborate with multiple team members
- Analysis processing can be offloaded from your desktop
- Single user mode to run the IBM Support Assistant application on your desktop
- Case management to organize diagnostic files by problem incident or other meaningful organization approaches
- File management helps quickly navigate through files that are stored in cases
- Reports tab provides a quick, concise view into all reports that are generated from server-based problem determination tools
- Automated data analysis scans your case files to find common problem symptoms

Figure 4-15. IBM Support Assistant Team Server: Features and benefits (1 of 2) WB8691.0

Notes:

IBM Support Assistant Team Server: Features and benefits (2 of 2)

- Overview tab shows a summary of key version and configuration information that is found in each file
- Symptoms tab shows a list of anomalies or symptoms that are discovered during the scan and the details for each symptom
- Knowledge tab shows known potential solutions for each symptom
- Global filter to filter what is shown in all tabs that are based on criteria you specify
- Secure communication allows all transfers that are encrypted to protect your data
- Administration functions can be restricted to a subset of users
- Information Center provides expanded documentation for IBM Support Assistant and problem determination tools

Figure 4-16. IBM Support Assistant Team Server: Features and benefits (2 of 2) WB8691.0

Notes:

WebSphere Education

Accessing IBM Support Assistant Team Server (1 of 2)

- Start the IBM Support Assistant server
  - **Start > All Programs > IBM Support Assistant 5 > Start IBM Support Assistant 5**

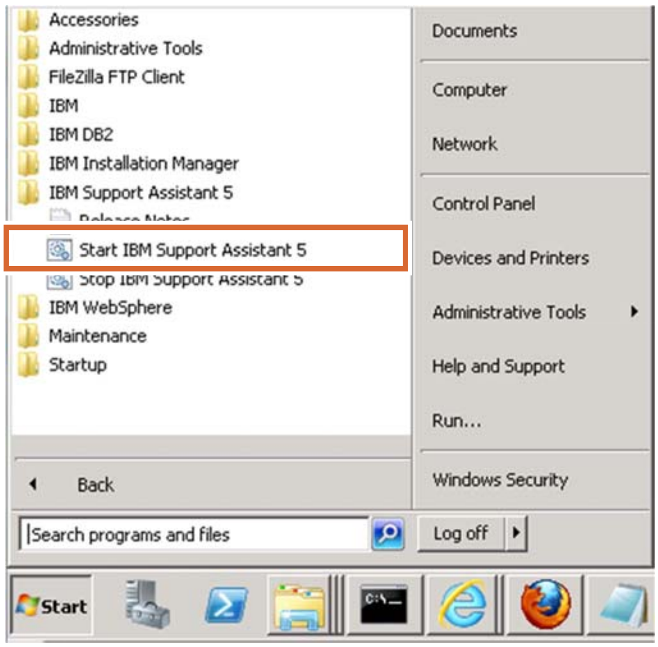


Figure 4-17. Accessing IBM Support Assistant Team Server (1 of 2) WB8691.0

Notes:

WebSphere Education

Accessing IBM Support Assistant Team Server (2 of 2)

- Open a web browser window and enter the following web address:
  `http://localhost:10911/isa5`
  - If you get a certificate security warning, you can ignore it
- The first time that you open IBM Support Assistant, you can select whether to enable anonymous usage statistics
  - You can change this option at any time by going to **Administration > Application settings**
  - **The Enable usage statistics** check box is selected by default

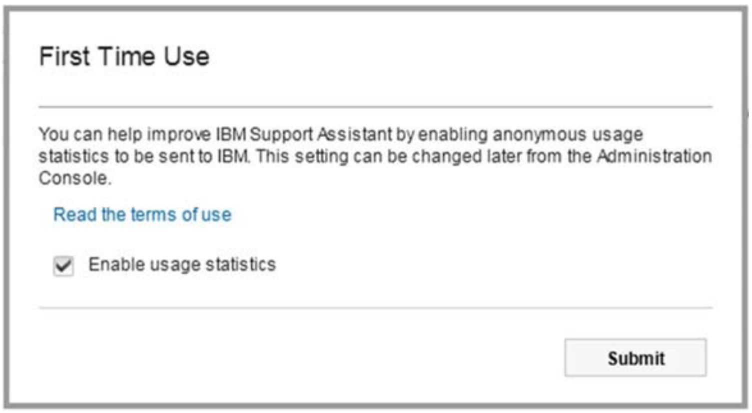


Figure 4-18. Accessing IBM Support Assistant Team Server (2 of 2) WB8691.0

Notes:

If you go to **Administration > Application settings**, and are asked to log in, the default login user name and password are `admin` and `admin`.

WebSphere Education

IBM Support Assistant Team Server main page

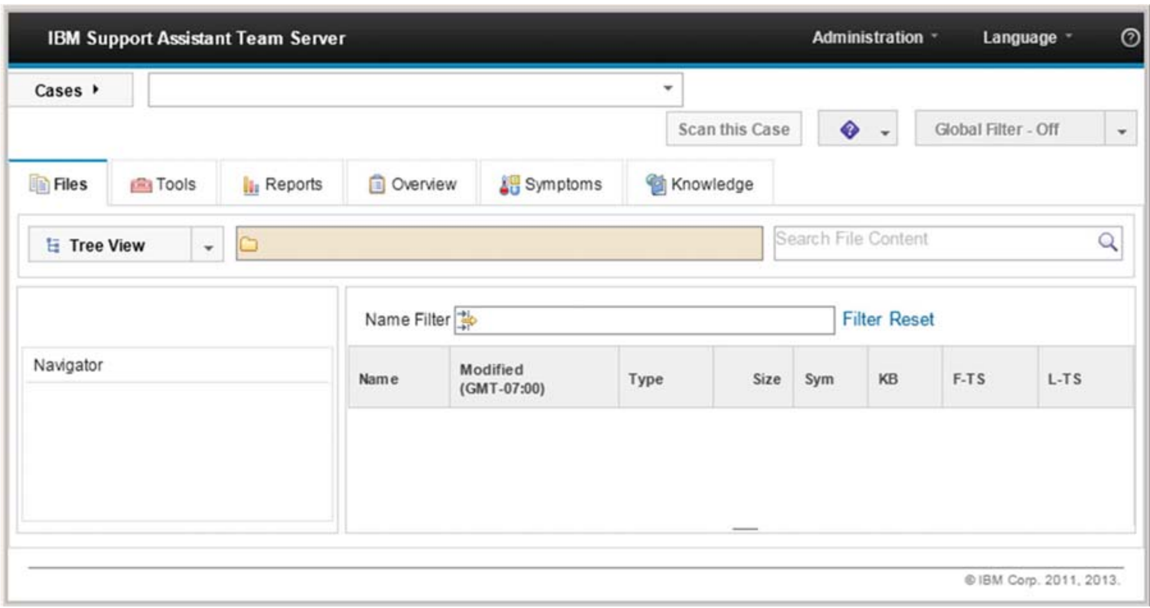


Figure 4-19. IBM Support Assistant Team Server main page WB8691.0

Notes:

This screen capture shows the main page for IBM Support Assistant. It opens on the **Files** view.

Cases

In IBM Support Assistant, you can organize your issue data by using cases. A case is a container for a logical grouping of files and information that are relevant to one singular issue. Cases help organize related files and run tools against them to diagnose and fix issues.

All cases have a unique Case ID that IBM Support Assistant assigns, a short summary that explains the issue that is tracked, and an optional full length description.

You can manage cases by using the case management pane that appears when you click **Cases** in the upper-left corner of the screen. You can use the Case Manager to create, delete, and edit cases. After you select a case, all file changes and tool executions are associated with that case.

The **Files** view and **Tools** view automatically refresh to display information on the current case.

A sample case is included with IBM Support Assistant Team Server. You can use the sample case to try different features in IBM Support Assistant Team Server.

Views

The main views are **Files**, **Tools**, and **Reports**.

Files view

With the **Files** view in IBM Support Assistant, you can manage and interact with diagnostic data files, which are important to any troubleshooting process.

In the **Files** view, you can:

- Add and organize files in your case by using the move, rename, delete, and copy file actions
- Manage data collections with the upload, download, unpack, and compress file actions
- Run analysis tools on files to generate reports or interactively work with the diagnostic data
- Use the Quick Tool Launcher to quickly identify tools that can run against a specific file and launch the associated tool
- Search file content by using keywords, and then view that file content by double-clicking it or clicking the direct download icon
- Filter the files list to show only the files that you are interested in analyzing

Tools view

The Tools view contains information on each of the Problem Determination Tools that are provided in IBM Support Assistant. Through the tools view, you learn about the analysis that each tool can perform and other information that you should know before you run the tool.

From the **Tools** view, you can:

- Read about the tools
- Launch tools and see their status

Note: You can also launch tools from the context menu in the Files view.

Reports view

The **Reports** view provides a concise view into all reports that were generated from the execution of report generator analysis tools within a case.

With the **Reports** tab, you can:

- Interact with a report
- Navigate to the location in a case where the report’s input and output are stored
- Open the report in a separate browser tab

The **Overview**, **Symptoms**, and **Knowledge** views display information after you scan the case files. You can scan case files by clicking the **Scan this Case** button.

Overview: This view displays a consolidated overview of information that was found in all the files that were encountered during the scan in the form of a list of properties. These properties can include information such as product version, operating system, server names and instances, and other information.

Symptoms: This view shows a list of all the potential anomalies that were found in the files within in the case, ranked in order of probable relevance. In this view, you can search for symptoms that are related to the problem that you are investigating. You can view detailed information by selecting the individual symptoms.

Knowledge: This view shows a list of potential solutions that were found for all the symptoms that were identified during the scan of this case, such as Technotes or APARs, which might apply to each symptom. You can view detailed information by selecting each individual entry in the list.

WebSphere Education

IBM Support Assistant Team Server: Problem Determination tools

- Classloader Analyzer
- FileNet Optical Storage and Retrieval (OSAR) Cable Tool
- Garbage Collection and Memory Visualizer (GCMV)
- Health Center
- Heap Analyzer
- Memory Analyzer
- Pattern Modeling and Analysis Tool for Java Garbage Collector
- Portal Log Analyzer
- Profile Port Checker
- Thread and Monitor Dump Analyzer for Java
- Trace and Request Analyzer for WebSphere Application Server
- Web Server Plug-in Analyzer for WebSphere Application Server
- WebSphere Application Server Configuration Visualizer

Figure 4-20. IBM Support Assistant Team Server: Problem Determination Tools WB8691.0

Notes:

This link provides a complete list of Problem Determination Tools available in IBM Support Assistant: `www.ibm.com/support/docview.wss?uid=swg27036217`

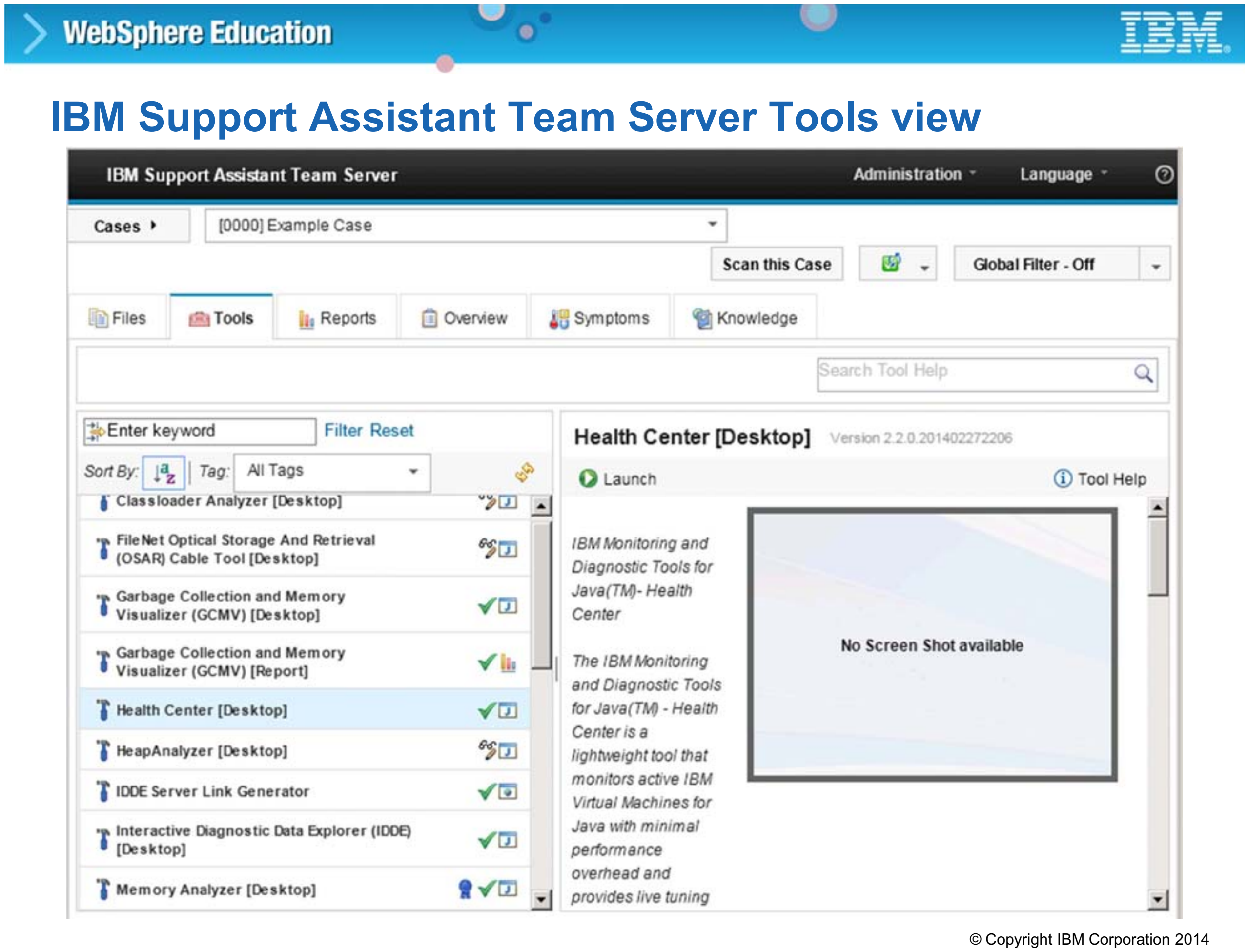


Figure 4-21. IBM Support Assistant Team Server Tools view WB8691.0

Notes:

This screen capture shows the **Tools** view in IBM Support Assistant Team Server. To view more information about a specific tool, select the tool on the left. Detailed information about the tool loads in the right side of the window.

4.3. IBM Support Assistant Data Collector for IBM Business Process Manager

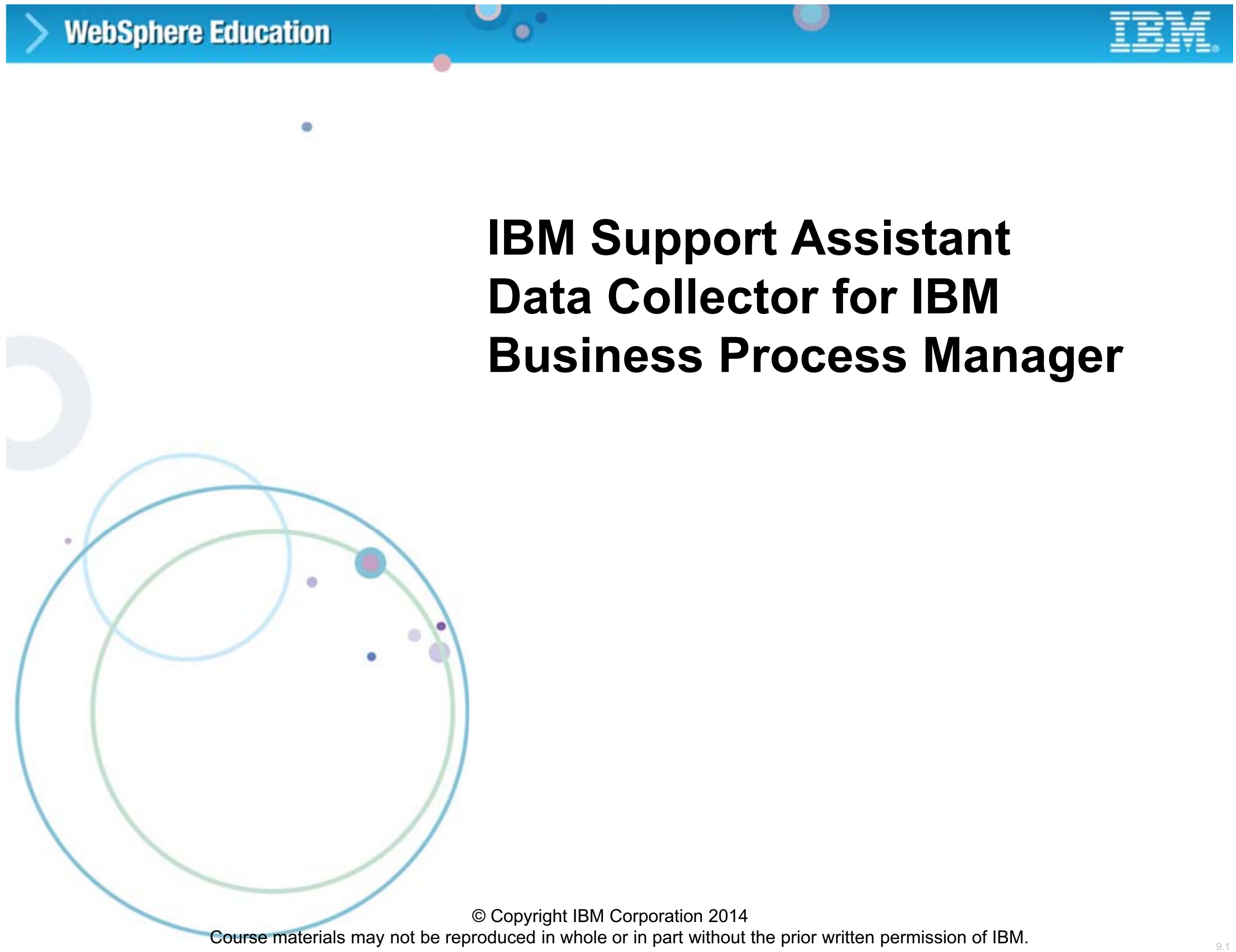


Figure 4-22. IBM Support Assistant Data Collector for IBM Business Process Manager WB8691.0

Notes:

WebSphere Education IBM

Using the IBM Support Assistant Data Collector

- The IBM Support Assistant Data Collector for Business Process Manager is a tool for gathering diagnostic data from an application server
  - Designed for ease of use
  - Gathers some of the standard information that is needed to debug IBM Business Process Manager issues
  - Can gather log file or configuration information and upload it to a Problem Management Record (PMR)
- The tool runs in console mode by starting the launch script from the command line
- In a Windows environment, run the `<profile_root>/bin/bpmdc.bat` command
  - Can use various parameters to specify output directory, whether to upload the file to IBM, and other actions

Figure 4-23. Using the IBM Support Assistant Data Collector WB8691.0

Notes:

The IBM Support Assistant Data Collector for Business Process Manager focuses on automatic collection of problem data. The script gathers various product information, compresses it into a `.zip` file, and can optionally send the compressed file to IBM for a PMR. You can also use this tool to compress the data without sending it to IBM.

For more information about the Data Collector tool and the various parameters you can use, see the following website: `www.ibm.com/support/docview.wss?uid=swg21641516`

WebSphere Education

IBM Support Assistant Data Collector for Business Process Manager: Example screen

Administrator: Command Prompt - bpmdc.bat

```
C:\IBM\BPM\ProcessServer\v8.5\profiles\DmgrProfile\bin>bpmdc.bat -outputDirector
y=C:\labfiles\ex2\ISADC\MustGather
Running the Collector command with the following options
/-----------------------------------------------------------------------------

type=General
config=FALSE
uploadToIBM=FALSE
PMR=
outputDirectory=C:\labfiles\ex2\ISADC\MustGather
\-----------------------------------------------------------------------------


The collection is beginning and could take a few minutes depending on the size.
/-----------------------------------------------------------------------------

| Your collection file is located at :
| C:\labfiles\ex2\ISADC\MustGather/General_Mon05-12-2014_14-21-08.89.zip
\-----------------------------------------------------------------------------


Collection is complete

C:\IBM\BPM\ProcessServer\v8.5\profiles\DmgrProfile\bin>
```


Figure 4-24. IBM Support Assistant Data Collector for Business Process Manager: Example screen WB8691.0

Notes:

In this example screen, the output is saved in the `C:\labfiles\ex2\ISADC\MustGather` directory.

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Describe MustGather data
- Describe IBM Support Assistant
- Describe various IBM Support Assistant tools for troubleshooting
- Describe the Data Collector for IBM Business Process Manager

Figure 4-25. Unit summary WB8691.0

Notes:

WebSphere Education

Checkpoint questions

1. Which of the following tools you can use to gather information about how to troubleshoot a problem?
   a) IBM Process Center
   b) IBM Process Designer
   c) IBM Support Assistant
   d) IBM Business Monitor

2. True or False. A MustGather document tells you the specific information that you must collect for a particular type of problem.

Figure 4-26. Checkpoint questions WB8691.0

Notes:

Write down your answers here:

1.

2.

WebSphere Education

Checkpoint answers

1. (c) IBM Support Assistant

2. True.

Figure 4-27. Checkpoint answers

WB8691.0

Notes:

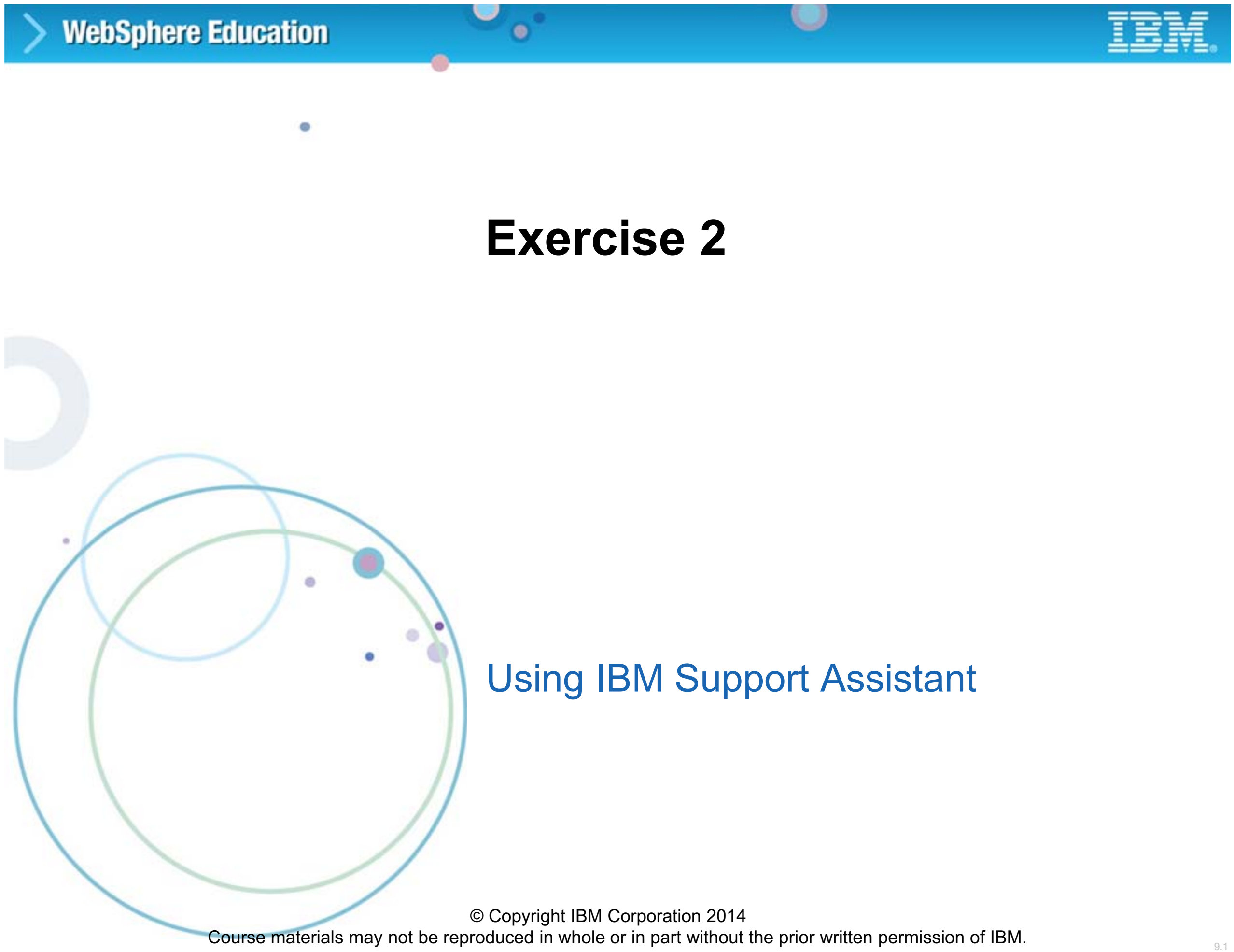


Figure 4-28. Exercise 2 WB8691.0

Notes:

WebSphere Education

Exercise objectives

After completing this exercise, you should be able to:

- Start and stop IBM Support Assistant Team Server
- Administer IBM Support Assistant Team Server
- Run report generator tools and examine the reports
- Use the Case Manager to create a case and add diagnostic data
- Run interactive desktop tools
- Use Automated Analysis to scan a case and examine the results
- Run the Data Collector tool
- Collect MustGather data for IBM Business Process Manager by using the automated Data Collector tool

Figure 4-29. Exercise objectives WB8691.0

Notes:

Unit 5. WebSphere Application Server problem determination refresher

What this unit is about

This unit provides an overview of WebSphere Application Server problem determination techniques.

What you should be able to do

After completing this unit, you should be able to:

- Describe components of JVM and overall architecture
- Describe garbage collection (GC) and GC tuning policies
- Identify a sluggish JVM and detect bottleneck problems
- Explain how to tune the heap size
- Define out-of-memory conditions
- Describe tools for analyzing out-of-memory problems
- Detect a hang condition
- Describe tools for analyzing hangs
- Define and detect a crash
- Describe the tools that are available for troubleshooting a crash

How you will check your progress

- Checkpoint

WebSphere Education IBM

Unit objectives

After completing this unit, you should be able to:

- Describe components of JVM and overall architecture
- Describe garbage collection (GC) and GC tuning policies
- Identify a sluggish JVM and detect bottleneck problems
- Explain how to tune the heap size
- Define out-of-memory conditions
- Describe tools for analyzing out-of-memory problems
- Detect a hang condition
- Describe tools for analyzing hangs
- Define and detect a crash
- Describe the tools that are available for troubleshooting a crash

Figure 5-1. Unit objectives WB8691.0

Notes:

WebSphere Education IBM

Topics

- Troubleshooting JVM problems
- Troubleshooting WebSphere out-of-memory problems
- Troubleshooting hangs
- Troubleshooting crashes

Figure 5-2. Topics WB8691.0

Notes:

5.1. Troubleshooting JVM problems

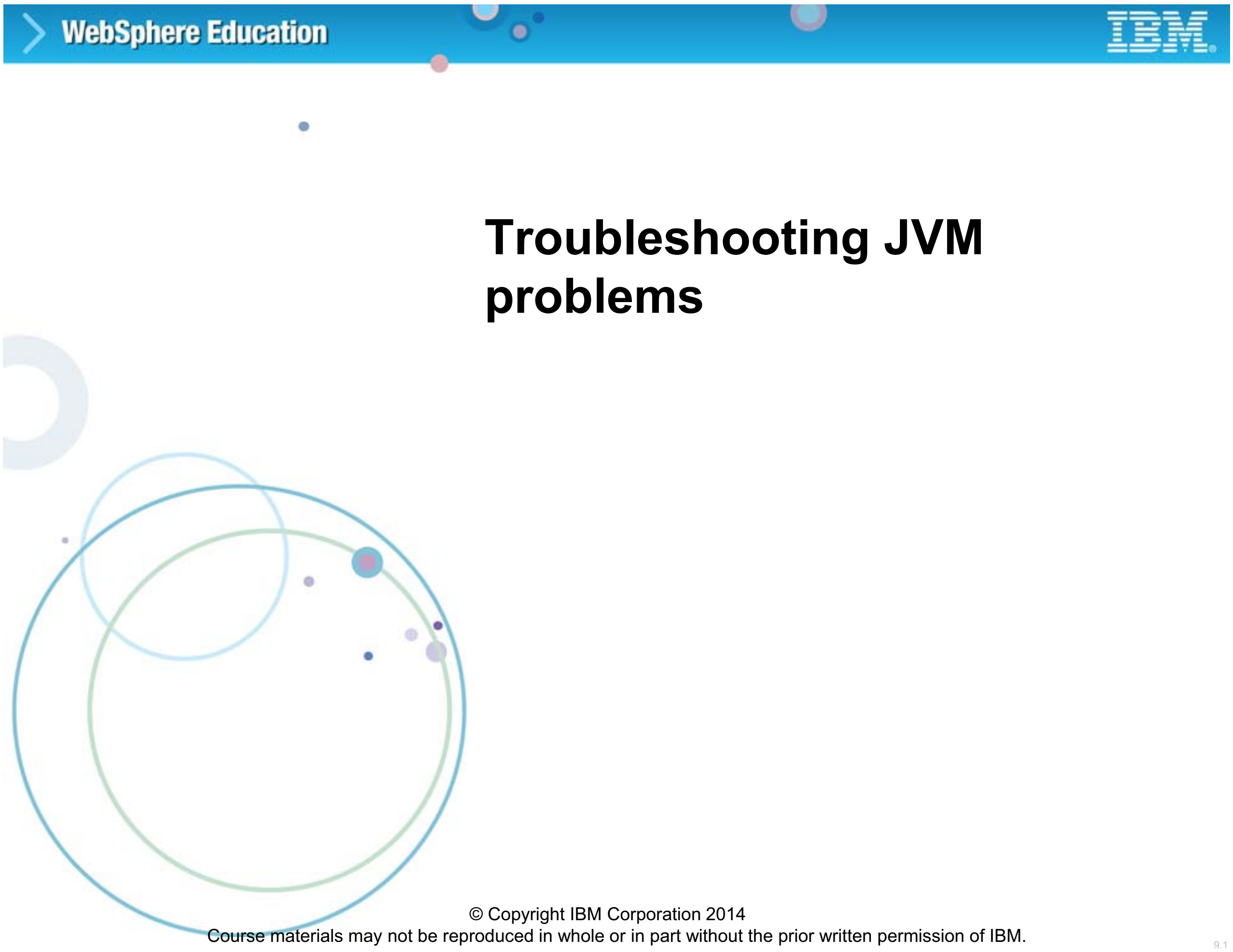


Figure 5-3. Troubleshooting JVM problems WB8691.0

Notes:

WebSphere Education IBM

Java virtual machine (JVM) features

Almost every WebSphere process runs in a JVM

The JVM provides:

- Class loading
  - A class loader verifies and loads classes into memory
  - Multiple class loaders are involved in loading the required libraries for an application to run
  - Each class loader loads its own classes
- Garbage collection (GC)
  - Garbage collection takes care of memory management for the entire application server
  - The GC process searches memory to reclaim space from program segments or inactive Java objects
- Execution management
  - Manages the bookkeeping work for all the Java threads
- Execution engine (interpreter)
  - Interprets the Java methods

Figure 5-4. Java virtual machine (JVM) features WB8691.0

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.

It 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

JVM version (1 of 2)

- WebSphere supports several JVMs based on version and operating system type
  - Windows, AIX, and Linux: IBM supplied
  - Can use only IBM SDK that WebSphere Application Server 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:

  `http://www.ibm.com/support/docview.wss?uid=swg27038218`

- To determine the JVM version in use:
  - Look in the `SystemOut.log` file of one of the profile instances

    <profile_home>`/logs/server1/SystemOut.log`

- JVM version can be found in the server job logon z/OS

Figure 5-5. JVM version (1 of 2) WB8691.0

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.

WebSphere Application Server V7.0 supports the Java Development Kit (JDK) Version 6.0.

WebSphere Application Server V6.1 supports the Java Development Kit (JDK) Version 5.0. This version can be different from JDK version 1.4.2, which is WebSphere Application Server V6.0 and earlier versions support. This new JDK introduces a new virtual machine implementation, a new garbage collection scheme, and a new just-in-time (JIT) compiler.

The JVM version matches the JDK level. The JVM is the runtime component of the JDK. The prerequisite page identifies the supported JDKs. The WebSphere supplied Java 2 SDK is required for both the runtime and any remote Java clients for Windows. For AIX and Linux, the supported version of the JDK is IBM 32-bit SDK (or 64-bit SDK where appropriate), Java 2 Technology Edition, V5 SR2. The actual JVM build number is provided, which is helpful in determining the exact JVM being used.

IBM does not supply a software developer kit or runtime environment for HP-UX and Solaris platforms. However, IBM does make strategic products, such as the WebSphere Application Server, for these platforms.

The JVM is installed as part of the WebSphere Application Server in the Java directory tree.

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.

IBM does service JVMs, but only when it is an embedded part of IBM middleware (for example, WebSphere Application Server).

WebSphere Education IBM.

JVM version (2 of 2)

- To determine the JVM version in use:
  - Run `java -fullversion` (IBM JVMs only) from the command line

```
[root@washost bin]# ./java -fullversion
java full version "JRE 1.6.0 IBM Linux build pxa6460_26sr5fp1ifx-
20130408_02 (SR5 FP1)"
```

  - Run `java -version` from the command line

```
[root@washost bin]# ./java -version
java version "1.6.0"
Java(TM) SE Runtime Environment (build pxa6460_26sr5fp1ifix-
20130408_02(SR5 FP1+IV38399+IV38578))
IBM J9 VM (build 2.6, JRE 1.6.0 Linux amd64-64 Compressed References
20130301_140166 (JIT enabled, AOT enabled)
J9VM - R26_Java626_SR5_FP1_20130301_0937_B140166
JIT  - r11.b03_20130131_32403
GC   - R26_Java626_SR5_FP1_20130301_0937_B140166_CMPRSS
J9CL - 20130301_140166)
JCL  - 20130408_01
```


Figure 5-6. JVM version (2 of 2) WB8691.0

Notes:

The Java virtual machine (JVM) used by the IBM SDK for Java is the IBM J9 virtual machine (J9 VM).

- JIT is just-in-time compiler.
- GC is garbage collector.
- J9CL is IBM J9 class library.
- JCL is Java class library.

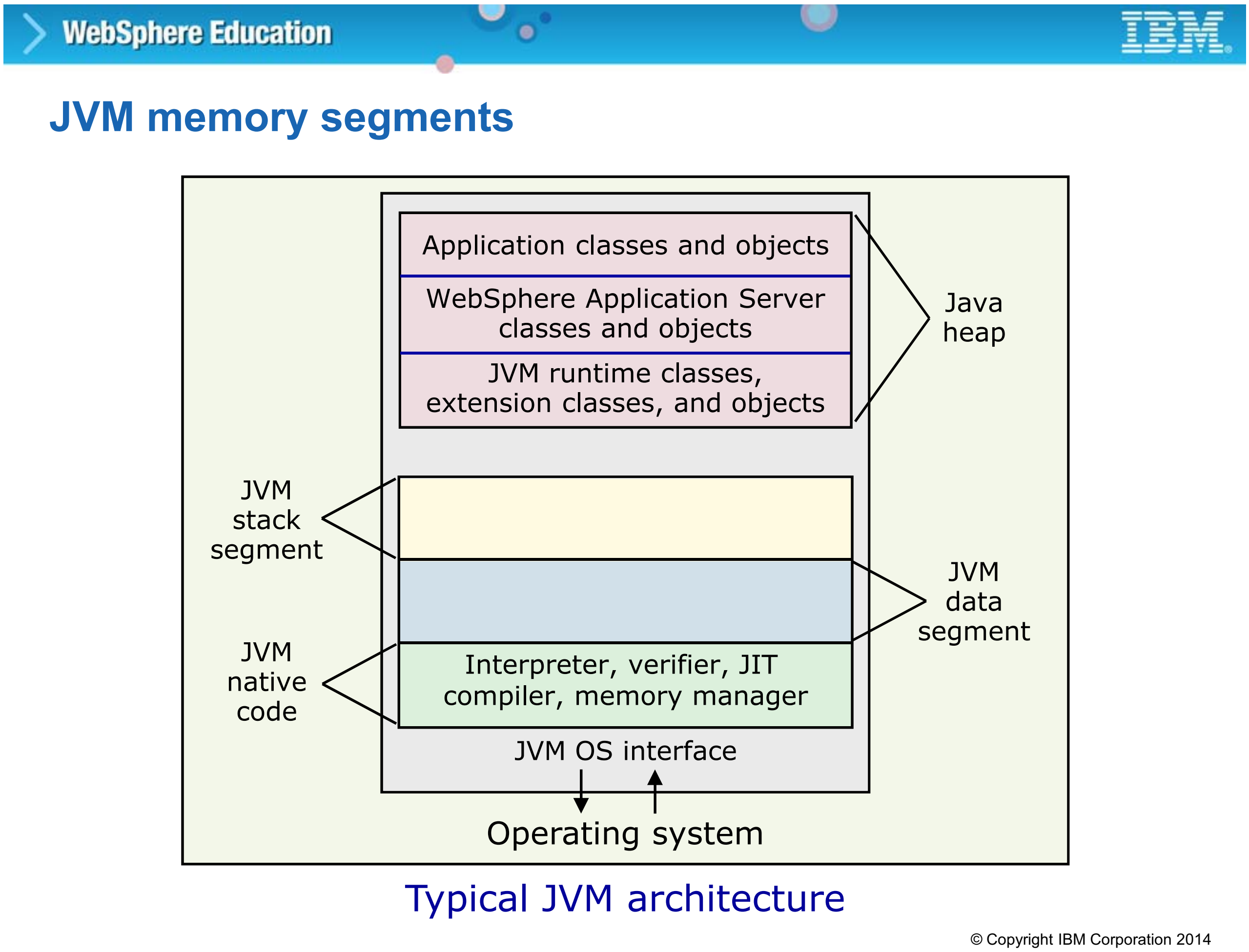


Figure 5-7. JVM memory segments WB8691.0

Notes:

Here are the important points to remember on this slide:

- The JVM is a process whose memory layout depends on the OS platform. The diagram that is shown is a typical abstraction of process memory layout.
- As a process, the JVM itself needs its own heap (JVM native heap) from which it gets its own dynamic memory needs.
- A portion of that heap is allocated for the Java heap where classes and objects are stored. This memory is the heap that Java programmers are familiar with.
- Garbage collection occurs only in the Java heap.
- The Java heap can grow and shrink. When there is enough available memory in the JVM heap, the Java heap can be extended easily without acquiring memory from the OS. Otherwise, a request for more memory from the OS is made, and the JVM process size increases as necessary.

The data segment contains many native memory buffers used for the network I/O facility and a database buffer if using a type 2 JDBC driver.

Memory that is not allocated to the Java heap is available to the native heap.

Available process memory space minus the Java heap memory equals the native heap.

WebSphere Education

Understanding garbage collection (GC)

- JVM manages the Java heap and does garbage collection
- Object is eligible for GC when there are no more references from root to that object
- Root set: References in the stack and registers
- Reference is dropped when variable goes out of scope

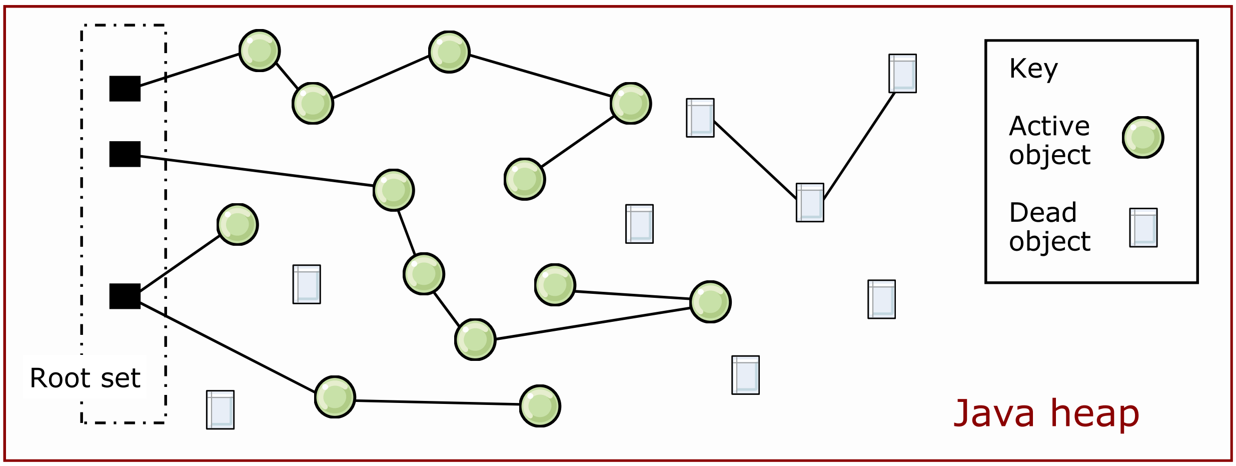


Figure 5-8. Understanding garbage collection (GC) WB8691.0

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 these 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 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

IBM

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 5-9. IBM JDK GC process WB8691.0

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. It 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.

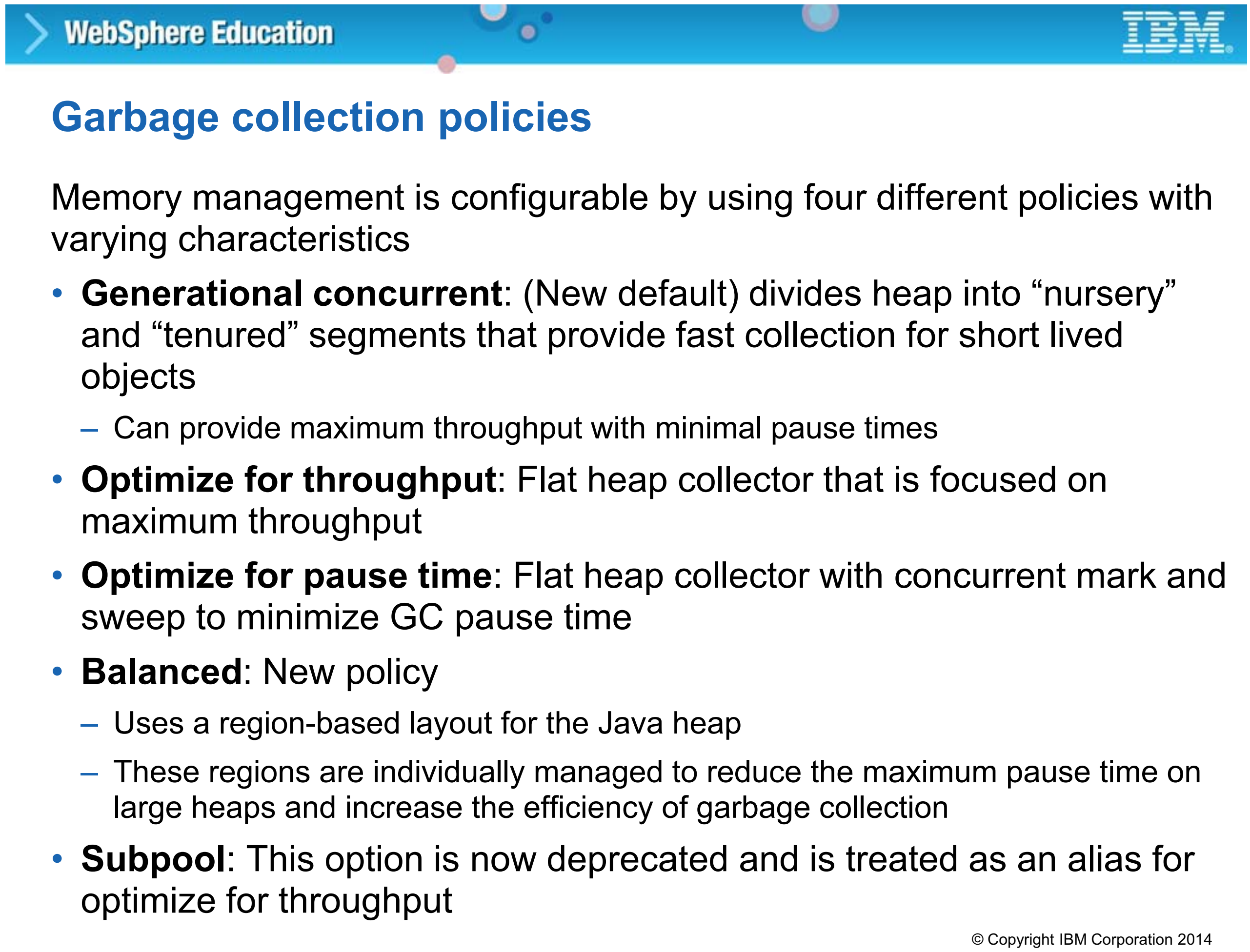


Figure 5-10. Garbage collection policies WB8691.0

Notes:

The default policy is “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 marks:

-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 marks with their 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. A flat heap technique to help increase performance on large SMP systems with 16 or more processors by optimizing the object allocation. It 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

JVM problem determination: Symptom analysis

- Application server stops responding under the following conditions:
  - Server crash
  - Hung process
  - Out-of-memory condition
- Server crash
  - Application server stops or exits unexpectedly
  - Tools are available to determine the cause of crash
- Hung process
  - Verify that application server process is still running
  - Threads might be deadlocked
  - Code might be running in a loop
  - Tools available to determine cause of the hang
- Out-of-memory condition
  - Errors and exceptions that are logged without process exit
  - At times, can result in unexpected process exit
- Performance degradation
  - Application server might crash and the node agent restarts it
  - Check to see whether the process ID is continually changing

Figure 5-11. JVM problem determination: Symptom analysis WB8691.0

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.

Server crashes, hung processes, and out of memory are covered in detail in subsequent units.

WebSphere Education

JVM problem determination: Data to collect

- Core files
  - Also known as process memory dumps or system core files
  - Complete memory dump of the virtual memory for the process
  - Can be large
  - Usually required by IBM support
  - Tools available to parse these files
- Javacore files
  - Also known as javadump or thread dump files
  - Text file that is created during an application server failure
  - Can also be generated manually
  - Error condition is given at top of memory dump
  - Format of the file is specific to the IBM JDK
- VerboseGC logs
  - Provides detailed information about garbage collection cycles
- Heap memory dump
  - Shows the objects that use Java heap memory
  - Needed for memory leak determination

Figure 5-12. JVM problem determination: Data to collect WB8691.0

Notes:

These data artifacts are covered in more detail in later units. They are presented here as an overview on the types of data available.

Javacore creation is enabled by default and can be disabled with option “DISABLE_JAVADUMP.” The steps to initiate and control the output of the javacore file vary by JVM type. It always best to look at the documentation that is provided with the specific JVM. On Solaris and HP-UX platforms, the thread dump goes to native_stdout.

WebSphere Education

Javacore overview

What is a javacore?

- A snapshot of the running Java process
- Small diagnostic text file that the JVM produces, contains vital information about the running JVM process
- Lists the JVM command line, environments, and loaded libraries
- Provides a snapshot of all the running threads, their stack traces, and the monitors (locks) held by the threads
- GC history and storage management (memory) information
- Necessary for detecting hung threads and deadlock conditions
- Useful for detecting some categories of native memory leaks
- Also, helpful for detecting performance problems
  - Take at least three snapshots of the JVM (about 2 – 3 minutes apart)
  - Analyze the javacores to see what different threads are doing in each snapshot
  - **Example**: A series of snapshots where container threads are in the same method or waiting on the same monitor or resource might indicate a bottleneck, hang, or deadlock

Figure 5-13. Javacore overview WB8691.0

Notes:

A javacore file is a snapshot of a running Java process. The IBM Java SDK produces a javacore file in response to specific operating system signals. Javacore files show the state of every thread in the Java process, in addition to the monitor information that identifies Java synchronization locks. If deadlocks are detected, it is noted in the file.

It provides various storage management values (in hexadecimal), including the free space in heap, the size of current heap, and details on other internal memory that the JVM is using. It also contains garbage collection history data. This feature is new for JDK 5.

You can use javacore files to resolve the problems that are listed here. More information on capturing data for problem determination can be found in the MustGather documents for these problems:

- 100% CPU usage
- Crash
- Hang or performance degradation

Javacore files can also provide an initial insight for memory problems.

WebSphere Education

Javacore file location and naming

- The javacore file is stored in the first viable location of:
  - Xdump:java:file=<path_name>/<filename.txt> (command-line argument)
  - The setting of the IBM_JAVACOREDIR environment variable (deprecated)
  - *<WAS_install_root>*/profiles/*<profile>*
  - TMPDIR or TEMP environment variable
  - Windows only: If the javacore cannot be stored in any of the above, it is put to STDERR
  - A new option -Xdump:java:defaults:file=<path/filename> can be used to change the default path and name of all javacore dump agents
- Javacore naming:
  - Windows and Linux: javacore.YYYYMMDD.HHMMSS.PID.txt where YYYY = year, MM = month, DD = day, SS = second, PID = processID
  - AIX: javacorePID.TIME.txt where PID = processID, TIME = time since 1/1/1970
  - z/OS: Javadump.YYYYMMDD.HHMMSS.PID.txt where YYYY = year, MM = month, DD = day, SS = second, PID = processID

Figure 5-14. Javacore file location and naming WB8691.0

Notes:

A javacore file can be generated on demand through the wsadmin command-line interface:

1. From the command prompt, enter the command wsadmin.bat to get a wsadmin command prompt.

   If security is enabled or the default SOAP ports are changed, you need to pass extra parameters to the batch file to get a wsadmin prompt. For example: wsadmin.bat [-host *host_name*] [-port *port_number*] [-user *userid*[-password *password*]

   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, you can get thread dumps for any JVM in that cell. If it is attached to a node agent, you can get thread dumps for any JVM in that node. If it is attached to a server, you can get thread dumps only for the server to which it is connected.

2. Get a handle to the problem application server. (Note: the contents in brackets "`[...]`", along with the brackets, are not optional. It must be entered to set the JVM object. Also, notice that 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 process.

3. Generate the thread dump:

   `wsadmin>$AdminControl invoke $jvm dumpThreads`

   The IBM_JAVACOREDIR depends on the JVM version and platform. For detailed information, check the IBM JDK Diagnostic Guide, which explains in detail all of the options for controlling javacores, such as naming and location.

Javacore file subcomponents

- **TITLE**: Shows basic information about the event that caused the generation of the javacore, the time it was taken, and the file name
- **GPINFO**: Shows general information about the OS
  - If the memory dump resulted from a general protection fault (GPF), information about the fault module is provided
- **ENVINFO**: Shows information about the JRE level, details about the command line that started the JVM process, and the JVM environment
- **NATIVEMEMINFO**: Provides information about the native memory that the Java runtime environment (JRE) allocates
  - Requested from the OS by using library functions such as `malloc()` and `mmap()`
- **MEMINFO**: Shows the free space in heap, the size of current heap, details of other internal memory that the JVM is using, and garbage collection history data
- **LOCKS**: Shows the locks that threads hold on resources, including other threads
  - A lock (monitor) prevents more than one entity from accessing a shared resource
- **THREADS**: Identifies the current thread, provides a complete list of Java threads that are alive, and provides their stack traces
- **CLASSES**: Shows information about the class loaders and specific classes loaded

Figure 5-15. Javacore file subcomponents WB8691.0

Notes:

GPF stands for general protection fault.

Use the Java command line to determine whether the javacore is for a WebSphere Application Server Java process. If so, which WebSphere Java process it is for: an Application Server process, an administrative server process, a jmsserver, a node agent, or some other WebSphere Application Server process?

The current thread is the thread that is running when the signal is raised that causes the javacore to be written. The **current thread details** show the Java and native stacks of the current thread. On stacks, the most recent calls are at the top of the stack.

- The Java stack shows the Java method calls that the current thread made.
- If there is a native stack, this javacore contains the most recent events that the thread ran. As the name implies, these events are in native code (libraries) called by Java.

The javacore processing dumps the current stack for every thread in the JVM. It shows the current state of the thread, whether it is *Runnable* or not. It also shows the thread name, which is useful in determining how that thread is used. It also shows the Java stack and native stack for each thread.

It is important to understand what signal caused the javacore. The signal determines how to interpret the information in the javacore file. Does the signal indicate a JVM crash? Does the signal information state it was due to an operator action (noted as a “user dump” event)? If the javacore is written to diagnose a hung JVM process, the steps you take are different than if the javacore is due to a JVM crash.

Each line of information is identified from tags, such as `TISIGINFO`. Different sections contain different tags, which make the file easier to parse for simple analysis.

Normal tags have these characteristics:

- Tags are up to 15 characters long (padded with spaces).
- The first digit is a nesting level (0,1,2,3).
- The second and third characters identify the component that wrote the message. The major components are:
  - **CI**: command-line interpreter
  - **CL**: Classloader
  - **LK**: locking
  - **ST**: storage (memory management)
  - **TI**: title
  - **XE**: execution engine
- The remainder is a unique string.
- Special tags have these characteristics:
  - A tag of `NULL` means that the line is just to aid readability.
  - A tag of `0SECTION` with the section title heads every section.

To verify that the javacore file is complete, you should see an `END OF DUMP` string at the end of the file.

WebSphere Education

Verbose garbage collection (GC)

- Verbose GC is an option that the JVM runtime provides
- Provides a garbage collection log:
  - Interval between collections
  - Duration of collection
  - Whether compaction was required
  - Memory size, memory that was freed, memory available
- Select **Servers > Server Types > WebSphere application servers >** <***serverName***>
- Under Server Infrastructure, click **Java and Process Management > Process Definition > Java Virtual Machine**
  - On the **Configuration** tab, select the **Verbose Garbage Collection** check box, restart server
  - On the **Runtime** tab, select the **Verbose Garbage Collection** check box, effective for the current server instance
- IBM JVM writes to `native_stderr.log`

Figure 5-16. Verbose garbage collection (GC) WB8691.0

Notes:

It is often a good idea to have verbose GC enabled permanently in production. The performance cost on a reasonably well-tuned JVM is small. The benefits of having it enabled the first time that something happens are considerable (no need to reproduce the problem a second time after enabling). It is also good to keep an eye on the verbose GC regularly, as a way to monitor the health of the system, even when nothing bad is noticed.

Enabling verbose GC by default is a decision that each system administrator must make consciously. But it is no longer plainly “not recommended as a normal production setting.”

In version 6.1, it is possible to enable verbose GC output from the **Runtime** tab. This setting allows the initiation of verbose GC output on a running application server.

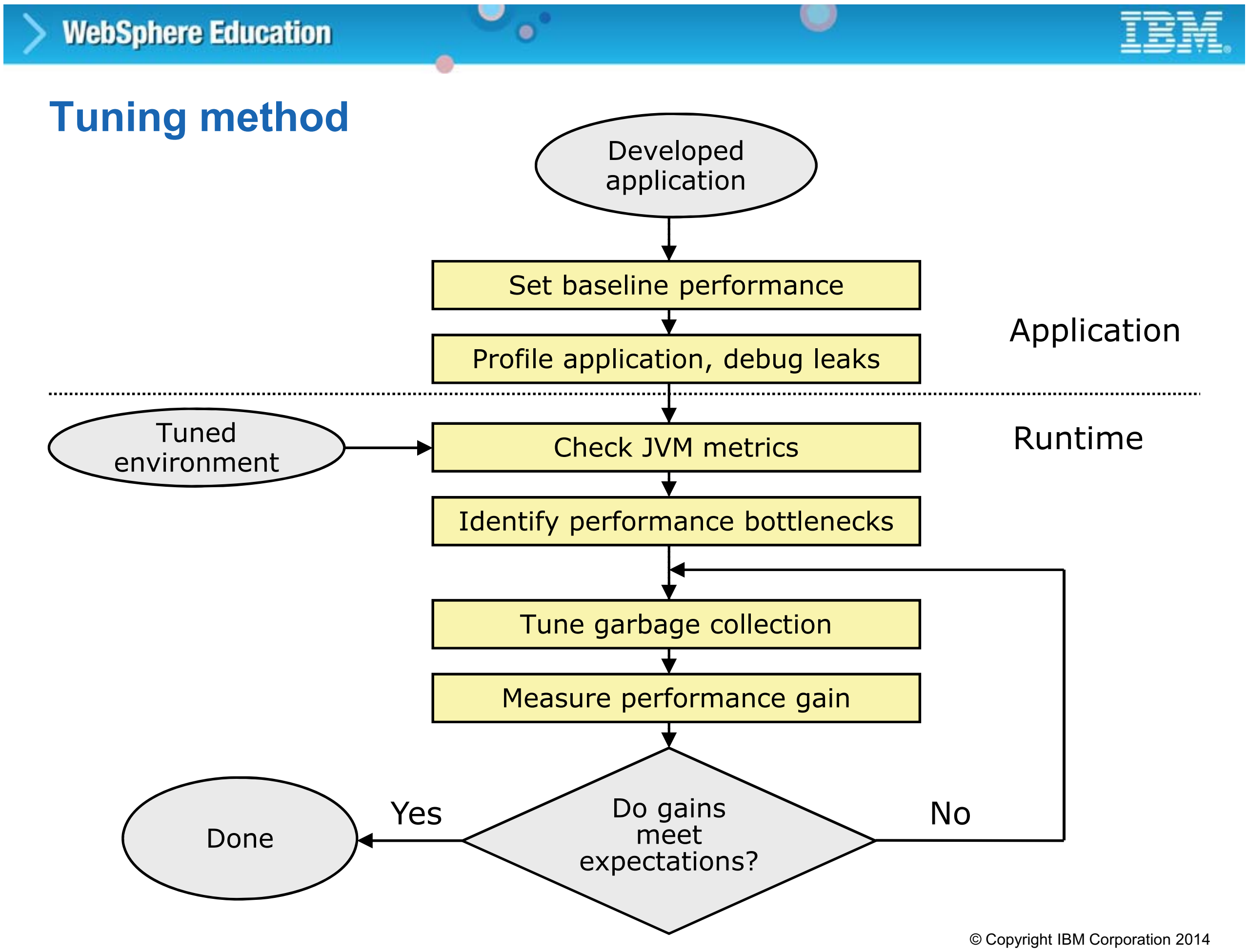


Figure 5-17. Tuning method WB8691.0

Notes:

The tuning process is iterative in nature. To find the optimal configuration, several tests and evaluation should be done.

There are many tuning parameters available to control the behavior of the JVM. These parameters are described detail in the JDK Diagnostic Guide. It is worth noting that most of JVM tuning involves configuring the ideal heap size and to a lesser degree the GC policy. Developers are encouraged to concentrate on these two areas. The other parameters provide fine tuning and are beyond the scope of this class.

WebSphere Education IBM

Maximum heap size

- 32-bit Java processes have maximum heap size
  - Varies according to the OS and hardware that is used
  - Determined by the process memory layout
- 64-bit processes do not have this limit
  - Limit exists, but is so large it can be effectively ignored
  - Addressability usually in the range of 2^{44} – 2^{64}, which is 16+ terabytes

Figure 5-18. Maximum heap size WB8691.0

Notes:

WebSphere Education

Theoretical and advised maximum heap sizes

- The larger the Java heap, the more constrained the native heap
- Limits are advised to prevent native heap from becoming overly restricted, leading to OutOfMemoryErrors
- Exceeding advised limits is possible, but should be done only when native heap usage is understood
- Native heap usage can be measured by using OS tools:
  - `svmon` (AIX)
  - `ps` (Linux) (for example: `ps -o pid,vsz,rss -p <PID>` where `vsz` is total virtual address space size and `rss` is resident set size)
  - `PerfMon` (Windows)
  - `RMF` (z/OS)

Figure 5-19. Theoretical and advised maximum heap sizes WB8691.0

Notes:

The `svmon` command provides a more in-depth analysis of memory usage. It is more informative, but also more intrusive, than the `vmstat` and `ps` commands. The `svmon` command captures a snapshot of the current state of memory. However, it is not a true snapshot because it runs at the user level with interrupts enabled.

The virtual set size (vsz) is the total size that the process occupies in virtual memory. The resident set size (rss) is the actual amount of space the process occupies in physical memory. Some of the memory that the application requested is now paged out to disk. This paged-out memory is part of the vsz number but not the rss number.

For example:

```
ps -o pid,vsz,rss -p 19854
PID    VSZ    RSS
19854 816864 316840
RMF=Resource Measurement Facility
```


WebSphere Education IBM

Tuning considerations: Heap

- Start with a reasonable maximum heap (-Xmx) size
  - 256 MB is the default for an application server
  - Try setting it to 512 MB
- Test different maximum heap sizes to find optimal setting
- Size the maximum heap larger than steady state to allow for peak load
- Consider opposing forces
  - The larger the heap, typically the longer the GC cycle
  - The smaller the heap, the more frequently GC is required
- Setting a larger minimum heap size (-Xms) can improve server startup time
  - If using the 50 MB default, might be resized several times during startup
- Setting the minimum too large can affect runtime performance
  - Larger value means more memory space that requires garbage collection
  - The JVM cannot compensate if you make a poor choice

Figure 5-20. Tuning considerations: Heap WB8691.0

Notes:

It is worth noting that with a minimum heap size different from the maximum heap size, the JVM is allowed to resize the heap over time to adjust to changing conditions. Even among the experts, opinions vary on whether that is a good thing or a bad thing.

The appropriate JVM heap size is usually the only major area of WebSphere Application Server that requires tuning:

- Too little heap:
  - Causes the application to GC frequently
  - Uses too much CPU
  - Extreme cases can lead to error conditions and server failure
- Too much heap:
  - Wasteful
  - Might cause long response times, especially on Solaris

WebSphere Education

The “correct” Java heap size (1 of 2)

- 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
  - Which means longer pause times than necessary
  - Which generally means reduced performance

Figure 5-21. The “correct” Java heap size (1 of 2) WB8691.0

Notes:

Heap occupancy is the amount of live data on the heap.

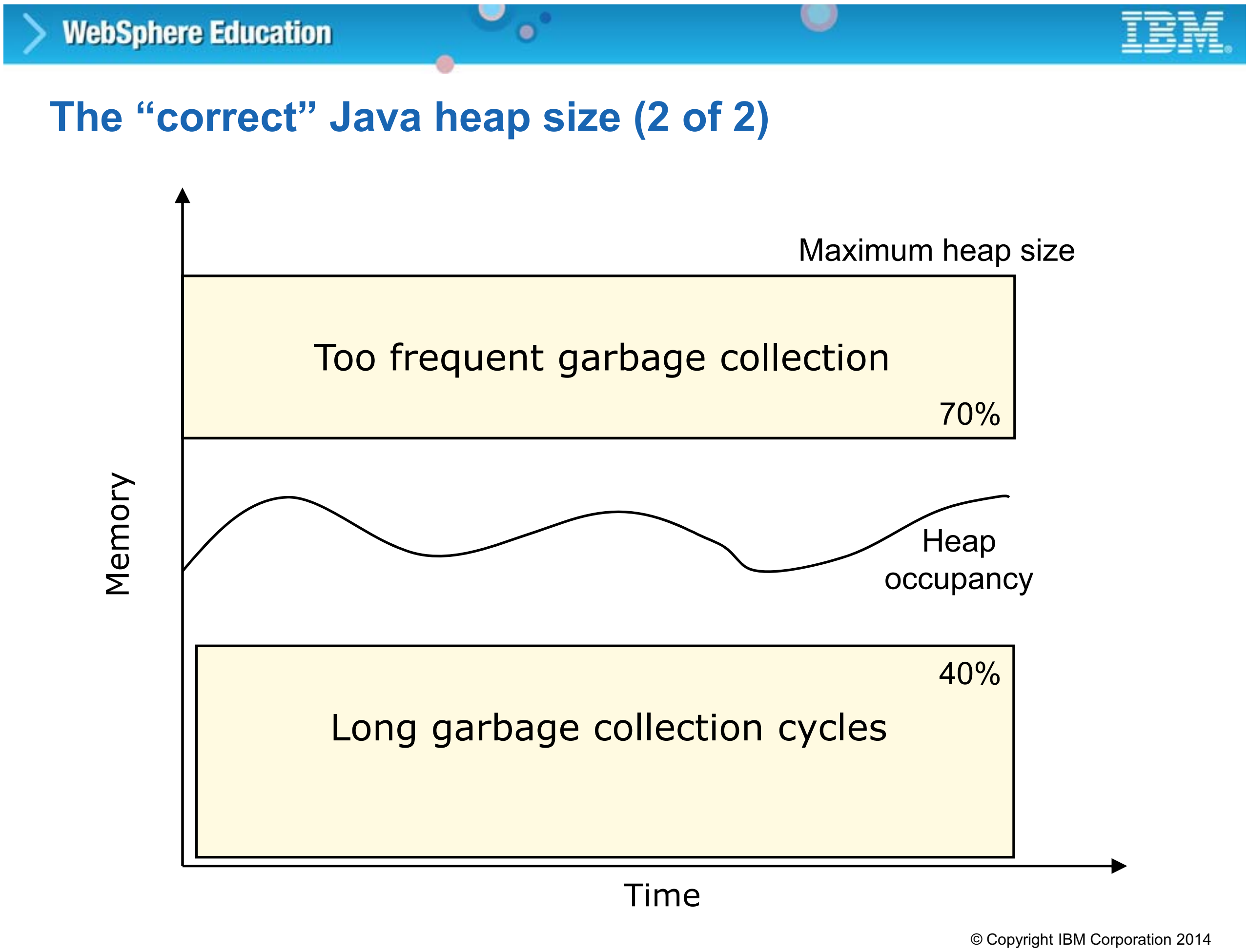


Figure 5-22. The "correct" Java heap size (2 of 2) WB8691.0

Notes:

For most applications, the default settings work well. The heap expands until it reaches a steady state, and then remains in that state, which should give a heap occupancy (the amount of live data on the heap) of 70%. At this level, the frequency and pause time of garbage collection should be acceptable.

WebSphere Education

Fixed heap sizes versus variable heap sizes

- Should the heap size be “fixed”?
  - That is, minimum heap size (-Xms) = maximum heap size (-Xmx)?
  - Does not expand or shrink the Java heap (avoids compactions)
  - Use for “flat” memory usage
- Variable heap sizes
  - GC adapts heap size to keep occupancy between 40% and 70%
  - Expands and shrinks the Java heap
  - Allows for scenario where usage varies over time, where variations would take usage outside of the 40 – 70% window
  - Provides more flexibility and ability to avoid OutOfMemoryErrors
- Each option has advantages and disadvantages
  - As for most performance tuning, you must select which is right for the particular application

Figure 5-23. Fixed heap sizes versus variable heap sizes WB8691.0

Notes:

A common performance-tuning measure is to make the initial heap size (`-Xms`) equal to the maximum heap size (`-Xmx`). Since no heap expansion or contraction occurs, this setting can result in significant performance gains in some situations. Usually, only the applications that need to handle a surge of allocation requests keep a substantial difference between initial and maximum heap sizes. Remember, though, that if `-Xms100m -Xmx100m` is specified, the JVM consumes 100 megabytes of memory for its complete lifetime, even if the heap usage never exceeds 10%.

WebSphere Education IBM

Tuning considerations: Garbage collection

- "Throughput" (the JVM definition)
  - Measure of productive time
  - Time that is spent in GC is not included
- Pauses
  - Measure of time when application execution pauses during GC
- Turn on verbose GC for more information about GC operation
- Allow the JVM to determine optimum time to GC, avoid calling `System.gc()` from the application
  - Causes the least efficient, slowest GC to take place
  - Always triggers a compaction
  - `System.gc()` does not always trigger an immediate GC
  - Can be disabled in IBM JDKs using `-Xdisableexplicitgc`
- `-Xdisableexplicitgc` must not be a permanent setting as the JVM uses `System.gc()` calls in extreme situations

Figure 5-24. Tuning considerations: Garbage collection WB8691.0

Notes:

When analyzing GC cost, two metrics must be investigated: frequency and duration. GC is the main cause of memory-related performance bottlenecks in Java.

There are two primary measures of garbage collection performance. *Throughput* is the percentage of total time that is not spent in garbage collection, which is considered over long periods of time. Throughput includes time that is spent in allocation (but tuning for speed of allocation is generally not needed). *Pauses* are the times when an application appears unresponsive because garbage collection is occurring.

Users have different requirements of garbage collection. For example, some consider the right metric for a web server to be throughput, since pauses during garbage collection can be tolerable, or network latencies can obscure them. However, in an interactive graphics program, even short pauses negatively affect the user experience.

Some users are sensitive to other considerations. *Footprint* is the working set of a process, which is measured in pages and cache lines. On systems with limited physical memory or many processes, footprint can dictate scalability. *Promptness* is the time between when an object becomes dead and when the memory becomes available, an important consideration for distributed systems, including Remote Method Invocation (RMI).

`–Xdisableexplicitgc` must not be a permanent setting as JVM uses `System.gc` calls in extreme situations (for example, DirectByteBuffer native OOM) and this setting inhibits this self-correction.

WebSphere Education

Tuning considerations: GC policy

- "I want my application to run to completion as quickly as possible"
  `-Xgcpolicy:optthruput`
- "My application requires good response time to unpredictable events"
  `-Xgcpolicy:optavgpause`
- "My application has a high allocation and death rate (objects are short-lived)"
  `-Xgcpolicy:gencon`
- "I'm using a 64-bit system and need heap sizes greater than 4 GB"
  `-Xgcpolicy:balanced`

Figure 5-25. Tuning considerations: GC policy WB8691.0

Notes:

The *subpool* option is deprecated in version 8.

The *balanced* option is intended for environments where heap sizes are greater than 4 GB. The policy is available only on 64-bit platforms.

WebSphere Education

Tuning considerations: Native heap

Beware: Garbage collection does not look into the native heap

- Ensure that JVM is not being swapped out of memory to disk
- Total memory that JVMuses > maximum heap size
- Native memory that is allocated in addition to the Java heap
- Native objects include:
  - Database connections for Type 2 JDBC drivers
  - Thread stacks
  - Compiled methods
  - JNI code and more
- Physical memory available > total memory that the JVM uses
  - JVMs do not page effectively due to garbage collection

Figure 5-26. Tuning considerations: Native heap WB8691.0

Notes:

Java Database Connectivity (JDBC) driver types:

- Type 2: Uses native C++ drivers and bridges to JDBC
- Type 4: Uses 100% pure Java drivers that communicate directly to the database with the native protocol of the database

5.2. Troubleshooting WebSphere out-of-memory problems

WebSphere Education

IBM.

Troubleshooting WebSphere out-of-memory problems

9.1

Figure 5-27. Troubleshooting WebSphere out-of-memory problems WB8691.0

Notes:

WebSphere Education

What is a java.lang.OutOfMemoryError?

- Java virtual machine error
- Not enough memory to allocate an object; some of the causes are:
  - The Java heap is too small
  - Memory is available in the heap, but it is fragmented (rare for Java 6)
  - Memory leak in the Java code
  - Not enough space in the native memory
- If available, first analyze the javacore file for memory issues:
  - Verify OutOfMemory as the cause for the javacore
  - Check heap size information
- Generate and analyze garbage collection and memory information:
  - Verbose GC
  - Javacore
  - Heap dump
  - System dumps

Figure 5-28. What is a java.lang.OutOfMemoryError? WB8691.0

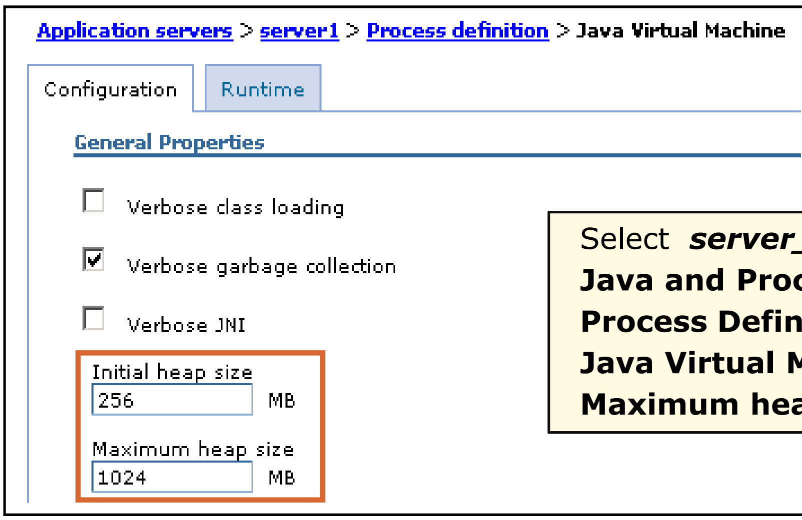
Notes:

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

JVM heap is too small

- How do you know that the heap is too small?
  - If the JVM heap shows constant growth until it reaches the maximum heap size, it is too small
  - The JVM heap never achieves a steady state if it is too small
- Must increase the Java maximum heap size in the administrative console



Select ***server_name*** **>**
Java and Process Management >
Process Definition >
Java Virtual Machine >
Maximum heap size

Figure 5-29. JVM heap is too small WB8691.0

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

Memory fragmentation

- Space is available on the heap, but not in contiguous blocks large enough to allocate most objects
- Some fragmentation always occurs
  - Should be treated as if the heap were too small
  - Tenured size might be too small for generational garbage collection
- The object that is being allocated is excessively large
  - The JVM is required to allocate an object that takes up a significant portion of the heap by itself
  - The application developer should attempt to reduce the size of the object that is being allocated
  - If that is not possible, increase the JVM heap
- Heap fragmentation is not common in recent versions of Java (5 and later) because of improved garbage collection modes (gencon and balanced)

Figure 5-30. Memory fragmentation WB8691.0

Notes:

The second potential reason for an OOM is memory fragmentation. This category was much more common for Java 1.4.2 and earlier. Therefore, it is unlikely to be the reason for an OOM in WebSphere 8.0; however, it is not impossible. Java objects must be allocated into a contiguous space of free memory; therefore it is possible that total free memory is large enough to satisfy the object, but it is not in a contiguous block. If the allocation size is relatively large (over 1 MB is a good indicator of large), then the best solution might be for developers to redesign the application to no longer require such large objects. If that is not possible, then increasing the overall heap size or tenured size for generational collection is the next step.

Symptoms of memory leak in the Java code

- No matter what the JVM maximum heap size is set to, the heap is still going to run out of memory
- Increasing the maximum heap size merely causes the problem to take longer to occur
- One or more objects are taking up a high percentage of the JVM heap:
  – A few large objects
  – Thousands of instances of small objects
- Memory leaks can also occur in native code
  – "**Iceberg objects**": Threads, classes, and direct bytebuffers that have a small Java heap footprint, but a large native heap footprint

Figure 5-31. Symptoms of memory leak in the Java code WB8691.0

Notes:

The third and most common cause for an OOM is a leak in Java memory usage. This problem occurs when some part of the code is allocating objects without ever releasing them, and therefore the heap grows until objects no longer fit. Unlike when the maximum heap size is too small, increasing the heap size when there is a memory leak prolongs the inevitable OOM. Increasing the heap can make matters worse because more data can be lost when the OOM occurs. Memory use that grows over time is called a memory leak. While memory leaks can occur quickly, they often take an extended time to reach maximum heap. Most memory leaks occur because of large collections of objects; however, it is also possible that a few large objects are occupying most of the heap. Memory leaks can occur outside of the Java heap in native memory.

WebSphere Education IBM

Classloader memory leaks

- Many memory leaks manifest themselves as class loader leaks
  - Classes with the same name can be loaded multiple times in a single JVM, each in a different class loader
  - The class loader retains a reference to every class it loads
  - These references often remain after a web application reload
  - With each reload, more classes are pinned, which leads to an out-of-memory error
  - The application code or JRE triggered code can cause class loader memory leaks
  - Beginning in V8.5, you can configure WebSphere Application Server to detect, prevent, and act, if possible, on class loader memory leaks by using the memory leak detection policy
- You can also use the IBM Classloader Analyzer tool, available in IBM Support Assistant, to detect and analyze class loader problems

Figure 5-32. Classloader memory leaks WB8691.0

Notes:

Many memory leaks manifest themselves as Classloader leaks. The name and Classloader uniquely identify a Java class. Classes with the same name can be loaded multiple times in a single JVM, each in a different Classloader. Each web application gets its own Classloader, and this Classloader is what WebSphere Application Server uses for isolating applications.

An object retains a reference to the class it is an instance of. A class retains a reference to the Classloader that loaded it. The Classloader retains a reference to every class it loads. Retaining a reference to a single object from a web application pins every class that the web application loads. These references often remain after a web application reload. With each reload, more classes are pinned, which leads to an out-of-memory error.

The application code or JRE-triggered code typically causes Classloader memory leaks.

See the information center topic, “Configuring the memory leak policy”:

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

To find more information on the IBM Classloader Analyzer, see the developerWorks article, “What is the IBM Classloader Analyzer?”:

`https://www.ibm.com/developerworks/mydeveloperworks/groups/service/html/communityview?communityUuid=a0a94b0d-38fe-4f4e-b2e6-4504b9d3f596`

WebSphere Education

Not enough native memory

- Insufficient memory available in the native memory segment
- There is more than sufficient space in the JVM heap, but the allocation still fails
- The JVM is not necessarily trying to allocate a large object, but rather memory is not available in the native memory space
- The JVM attempts to create an OutOfMemoryError, but might not have the necessary resources

Figure 5-33. Not enough native memory WB8691.0

Notes:

The final cause for an OOM is when there is not enough native memory to satisfy an object allocation. There is rarely a correlation between the size of the Java object allocation request and the amount of space that is required in native memory. Unfortunately, the verbose GC logs do not help identify the cause of a native memory problem. In addition, the JVM tries to create an OOM but at times might not have the resources to even create that object.

WebSphere Education

Java process restrictions

- Not all Java process space is available to the Java application heap
- The Java runtime needs memory for:
  - The Java virtual machine resources
  - Backing resources for some Java objects
- This memory area is part of the native heap
- Most memory that is not allocated to the Java heap is available to the native heap
  - Available memory space minus Java heap ≈ native heap
- Effectively, the Java process maintains two memory pools: Java and native

Figure 5-34. Java process restrictions WB8691.0

Notes:

Java process spaces can be logically broken into two key memory segments: native memory and Java heap memory. The native memory is where the JVM allocates its own resources and interacts with the underlying OS. Some Java objects have native memory requirements (such as Type 2 JDBC drivers and I/O buffers), but the size of the native requirement is often not associated with the size of the Java object. The Java process space also contains the native libraries. An easy way to approximate the size of the native memory available to the process is to subtract the maximum Java heap size from the total memory available to the process space. This approximation is not accurate for Windows because of how native libraries are allocated.

WebSphere Education

The native heap

- Allocated using `malloc()` and `mmap()`
  - Therefore, subject to memory management by the OS
- Used for virtual machine resources:
  - Execution engine
  - Class loader
  - Garbage collector infrastructure
- Used to underpin Java objects:
  - Threads, classes, AWT objects
- Used for allocations by JNI code
- Size cannot be directly controlled
  - To increase this memory space, decrease the Java heap size

Figure 5-35. The native heap WB8691.0

Notes:

The `malloc()` and `mmap()` methods are used to allocate memory in the native heap; therefore it is subject to management by the operating system. There are several reasons why a Java process must allocate space in the native memory, and the best way to determine native memory needs is to monitor the Java process from the operating system. The size of native memory needs cannot be directly controlled; however, decreasing the maximum Java heap effectively increases the space available for native memory.

WebSphere Education

Using Tivoli Performance Viewer to anticipate an OutOfMemoryError

- Tivoli Performance Viewer runs in the administrative console
- Uses Performance Monitoring Infrastructure (PMI) to capture information about the WebSphere runtime
- Provides graphs and charts of the captured PMI data

Figure 5-36. Using Tivoli Performance Viewer to anticipate an OutOfMemoryError WB8691.0

Notes:

One of the built-in tools for monitoring WebSphere for an OOM is Tivoli Performance Viewer. Tivoli Performance Viewer uses the Performance Monitoring Infrastructure (PMI) to display information about the WebSphere runtime and provides a graphical display of the information. Tivoli Performance Viewer runs on the Deployment Manager; however, it must communicate with the application server (often through the node agent) and can have a performance impact if used during a heavy load.

WebSphere Education

Administrative console Tivoli Performance Viewer graph

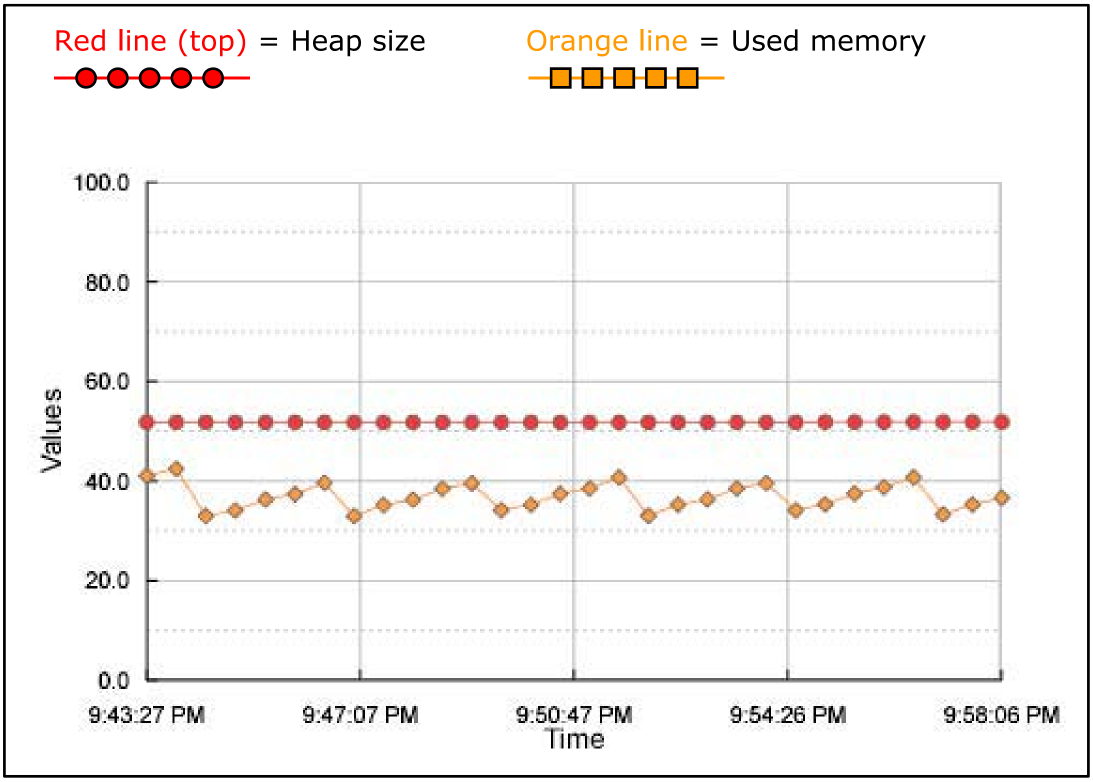


Figure 5-37. Administrative console Tivoli Performance Viewer graph WB8691.0

Notes:

This screen is a Tivoli Performance Viewer graph of heap size versus used memory. The sawtooth pattern of the used memory line is indicative of a correctly working JVM. If the line steadily approaches the heap size (red line), it indicates either a leak or a need for a larger heap size.

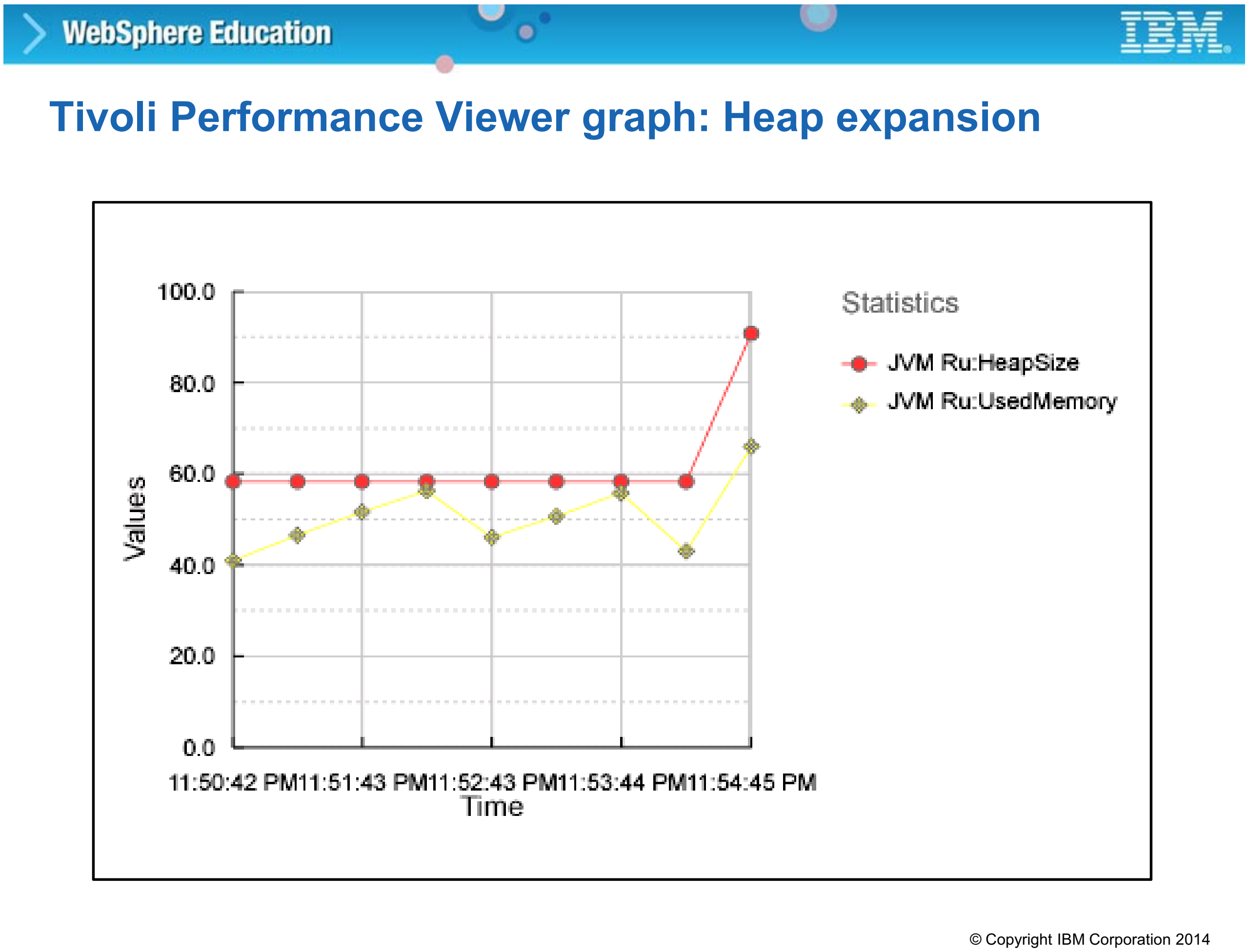


Figure 5-38. Tivoli Performance Viewer graph: Heap expansion WB8691.0

Notes:

Observe that the used heap grows to heap size after the first two garbage collections. Following the garbage collection, the heap expands. A sudden burst of application activity, requests for large objects, or possibly a memory leak can cause the heap to grow rapidly.

WebSphere Education

Tivoli Performance Viewer usage

- Tivoli Performance Viewer
  - Monitors internal thread pools and heap statistics
- Know the expected behavior of your application: I/O intensive, JMS, number of EJBs, expected load requirements
- Use Tivoli Performance Viewer to monitor the heap usage
  - If the used heap continues to grow over time with no corresponding increase in user load, there might be a memory leak
  - Use a profiler to dig into the specific heap to determine where the problem exists
  - The Java Virtual Machine Tool Interface (JVMTI) is a native programming interface that provides tools the ability to inspect the state of the Java virtual machine (JVM)

Figure 5-39. Tivoli Performance Viewer usage WB8691.0

Notes:

Tivoli Performance Viewer can be used to monitor several aspects of both an application and the WebSphere Application Server itself. The information in Tivoli Performance Viewer is best used when combined with existing expectations for how those metrics should behave in production.

JVMTI collects information about the JVM that runs the application server. The Tivoli Performance Viewer uses these interfaces to enable more comprehensive performance analysis.

JVMTI is a two-way function call interface between the JVM and an in-process profiler agent. The JVM notifies the profiler agent of various events, for example, garbage collection and thread starts. The profiler agent activates or deactivates specific event notifications that are based on the needs of the profiler.

JVMTI supports partial profiling by allowing you to choose which types of profiling information to collect and to select certain subsets of the time during which the JVM API is active. JVMTI moderately increases the performance impact. Therefore, it is good to use JVMTI monitoring to help diagnose application problems only.

To learn how to enable the Java virtual machine profiler data, read the information center article:

`http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/tprf_jvmpidata.html`

WebSphere Education IBM

How to obtain a verboseGC log

- The JVM runtime provides the verbose GC option
- Enables a garbage collection log
  - Interval between collections
  - Duration of collection
  - Whether compaction is required
  - Memory size, memory that is freed, memory available
- Use the administration console to enable the verbose GC for the server
  - **Servers > Server Types > WebSphere application servers >** ***server_name***
  - Under Server Infrastructure, click **Java and Process Management > Process Definition > Java Virtual Machine**
  - Select **Verbose Garbage Collection** check box
  - Restart the server or configure on runtime tab
- Usually writes to native_stderr
  - Varies depending on the operating system and WebSphere version
  - Some performance impact because of disk I/O, but usually minimal unless thrashing

Figure 5-40. How to obtain a verboseGC log WB8691.0

Notes:

The recent GC information in the javacore might not be enough to determine the cause of the OOM; therefore, it is a good practice for administrators to enable verbose GC for the application servers. In fact, several support teams encourage customers to always enable verbose GC because the added value of the information on the GC logs far exceeds the administrative cost of having to monitor the disk usage. VerboseGC can be easily enabled through the administrative console and requires a server restart to take effect. By default, the verboseGC output writes to the `native_stderr.log` file. There is some administrative cost due to writing the information to disk; however, this usage is minimal.

WebSphere Education

System dump files

- A system dump is a superset of a PHD heap dump
- A system dump also includes memory contents (strings, primitives, variable names, and others), thread and frame local information, some native memory information, and more
- This added information can solve a larger class of problems, provide more general insight into a running JVM, and ultimately reduce the time that it takes to solve a problem
- In earlier versions of IBM Java, a system dump had to be post-processed using the `jextract` tool
- With recent versions of IBM Java, a DTFJ-capable tool, such as the Memory Analyzer Tool, can directly load a system dump without any post-processing for OOMs
- For crashes, however, `jextract` should still be used

Figure 5-41. System dump files WB8691.0

Notes:

For more information, go to:
`http://www.ibm.com/developerworks/opensource/library/j-memoryanalyzer/index.html`

WebSphere Education

Obtaining system core dumps

- In addition to being automatically triggered the first time an OutOfMemory error occurs, system core dumps can be triggered manually in several ways:
  - Using wsadmin
  - Using the system dump button in the administrative console
  - Using OS commands
- Run the following `wsadmin` commands:

```
jvm=AdminControl.completeObjectName(
  'type=JVM,<server_name>,*')
AdminControl.invoke(jvm,'generateSystemDump')
```

- This function is not supported when using a non-IBM Java virtual machine (JVM)
  - HP-UX and Solaris operating systems

Figure 5-42. Obtaining system core dumps WB8691.0

Notes:

The sample script can trigger a system core dump with wsadmin. This method might not work if the target JVM exhausts the Java heap and is unable to process the message that instructs it to create the system core dump.

WebSphere Education

Java heap dumps

- A heap dump contains all the reachable objects that are on the Java heap
  - Those objects are the ones that the Java application uses
- Heap dumps are created by default for OutOfMemory exceptions
  - By default, heap dumps are in binary (PHD) format
  - Can also be generated in test format
- Used for analyzing the actual contents of the Java heap
  - Shows the objects that are using the heap memory
  - Shows the objects that are using large amounts of memory on the Java heap, and what is preventing the garbage collector from collecting them
  - Useful for debugging memory leaks in Java applications

Figure 5-43. Java heap dumps WB8691.0

Notes:

Heap dump is an IBM JVM facility that generates a dump of all the reachable objects that are on the Java heap; that is, the objects that a Java application uses. This dump is called a heap dump. It shows the objects that are using large amounts of memory on the Java heap, and what is preventing the garbage collector from collecting these objects.

The heap dump contains two lines per object. The first line displays the address of the object, various flags, its size, and type information. The second line contains a list of the memory addresses of other objects that the object references.

WebSphere Education

Heap dumps versus system core dumps

- The Memory Analyzer tool accepts a full system core file or a heap dump file
  - Which is better?
- A system core dump allows you to do everything a heap dump can; however,
  - It is considerably larger
  - It usually takes a little longer to create
- The size and time to create the system core must be balanced against its advantages, such as:
  - Accurate GC root information
  - Native heap memory
  - Ability to see the values of primitive fields and strings (might be a privacy issue)
  - More views such as threads and their related stacks and frame local references
- Unless the write time or size make system cores unmanageable, consider the use of them instead of heap dumps

Figure 5-44. Heap dumps versus system core dumps WB8691.0

Notes:

You should be aware that since the values of some variables and strings are available in a system core dump, you might display confidential data such as credit card numbers and identity information. If you must protect the privacy of such data, system core files must be kept confidential.

WebSphere Education

Obtaining Java heap dumps

- Generated from a running JVM in these ways:
  - Explicit, manual generation, such as using `wsadmin` commands
  - JVM-triggered generation, which uses dump agents to handle exceptions
  - Automated heap dump generation facility
- IBM Java heap dump for IBM JVMs
  - Windows, AIX, and Linux
  - JDK V5.0 and later: Use command-line argument `-Xdump:heap`
  - JDK V1.4.2 and earlier: Use environment variables

Figure 5-45. Obtaining Java heap dumps WB8691.0

Notes:

The other useful artifact for determining the cause of an OOM is heap dumps. Heap dumps provide insight into the contents of the Java heap and are the best way to determine what might be causing a memory leak. Several methods can generate heap dumps, including wsadmin, JVM triggers, or automatically when an OOM or other conditions occur.

WebSphere Education

Garbage Collection and Memory Visualizer overview

- The Garbage Collection and Memory Visualizer (GCMV) is a visualizer for verbose garbage collection output
  - The tool parses and plots verbose GC output and garbage collection traces (`-Xtgc` output)
- Started from the IBM Support Assistant
- The GCMV provides:
  - Raw view of data
  - Line plots to visualize various GC data characteristics
  - Tabulated reports with heap occupancy recommendations
  - View of multiple data sets on a single set of axes
  - Ability to save data as an image (jpeg) or comma-separated file (csv)

Figure 5-46. Garbage Collection and Memory Visualizer overview WB8691.0

Notes:

GCMV is available through the IBM Support Assistant and can parse both verboseGC logs and garbage collection trace. The main value of GCMV is the ability to visualize the variety of GC data and visualize the health of the JVM over time. You can also export information from GCMV in `.jpeg` or `.cvs` format to share the findings with other parties.

WebSphere Education

GCMV usage scenarios

- Investigate performance problems
  - Long periods of pausing or not responding
- Evaluate your heap size
  - Check heap occupancy and adjust heap size if needed
- Garbage collection policy tuning
  - Examine GC characteristics, compare different policies
- Look for memory growth
  - Heap consumption slowly increasing over time
  - Evaluate the general health of an application

Figure 5-47. GCMV usage scenarios WB8691.0

Notes:

OOM is not the only reason to evaluate verboseGC logs. The information in verboseGC and help improves performance by finding a better heap size, more appropriate garbage collection policy, or by monitoring the growth in memory consumption.

WebSphere Education

IBM Pattern Modeling and Analysis Tool (PMAT)

- Predates GCMV but can also be useful when analyzing verbose GC
  - Available from the IBM Support Assistant
  - Works with IBM JVM 1.3 and up
- Provides the following features:
  - Tabulated and graphical views
  - Summary usage analysis
  - Analysis includes tuning recommendations
- Suggestion: GCMV is your first choice; use PMAT only if you have some particular reason to prefer it

Figure 5-48. IBM Pattern Modeling and Analysis Tool (PMAT) WB8691.0

Notes:

Another tool that predates GCMV but can also be useful when analyzing verboseGC is PMAT. PMAT provides similar capabilities to GCMV though it is not Eclipse-based.

WebSphere Education IBM

OutOfMemory: Interpret heap dumps

Heapdump:

- Use tools to parse the object trees
  - Almost impossible to do manually
- Look for root objects that hold significant memory resources
  - Includes memory that is allocated for the root object itself and all the memory that is required for all the objects in the reference tree
- Check reference tree for sudden drop in memory
  - Expand the tree; examine each object as you traverse downwards
  - Compare the memory with the object right above it in the tree
  - A significant drop in memory indicates a potential memory leak
  - A long chain of similarly sized objects also indicates a potential leak

Jmap:

- Contact support to parse output
- Manually inspect the output list

Figure 5-49. OutOfMemory: Interpret heap dumps WB8691.0

Notes:

The heap dump is the final artifact that is associated with an OOM and can be the most useful if you are trying to determine the cause of a memory leak. The heap dump contains the object hierarchy for every object in the Java heap. The normal approach to leak detection is to look for drops in the reach size of an object. The reach size (also referred to by other names) is the size of the current object except for the cumulative size of all the objects beneath it in the hierarchy. A drastic drop in the reach size would indicate that the parent object has several multiple child objects that are not large on their own but add up to a substantial part of the heap. This method of analysis does not detect all memory leaks, but it does find the most common forms.

WebSphere Education IBM

How to analyze a Java heap dump

- Single dump analysis
  - Inspects a single heap dump file for container objects that have large reach size as compared to their largest child object
  - Most commonly used to analyze automatically generated heap dumps after an OutOfMemoryException
- Comparative analysis (rarely necessary)
  - Analyzes heap growth between two dumps that are taken over time
  - Identifies the objects with the largest size difference and their ownership relationships
- Analysis results
  - Summary of analysis results that show heap contents, size, and growth
  - Lists suspected data structures, data types, and packages that contribute to the growth in heap usage
  - Tabular views of all the objects and data types in the memory dump with filters and sorted columns

Figure 5-50. How to analyze a Java heap dump WB8691.0

Notes:

There are a few ways to go about analyzing heap dumps, and the best choice depends on what artifacts are available. If you have a single dump, likely generated on OOM, then single dump analysis is the only choice. However, if you have a dump from the same Java process (meaning before the OOM but after the most recent restart), you can use comparative analysis to better narrow down the leak suspects. In either case, most tools provide an analysis summary that breaks down the heap dump contents and aims to better direct your investigation.

WebSphere Education

HeapAnalyzer

- Analyzes heapdump for leaks and builds object hierarchy
  - Detection method is suited for large collections of objects
  - Simplifies hierarchy to one parent that might cause artifacts
- Available from IBM Support Assistant
- Available on alphaWorks:
  - `http://www.alphaworks.ibm.com/tech/heapanalyzer`
- Suited for quick and simple hierarchy inspection
- Some features include:
  - List of Java heap leak suspects
  - List of gaps among allocated objects, classes, and arrays
  - Java objects, classes, and arrays search engine
  - List of objects, classes, and arrays by type, object, size, size of child, and other parameters
  - List of available heap spaces by size
  - Tree of Java heap dump
- Suggestion: Memory Analyzer is your first choice, use HeapAnalyzer only if you have some particular reason to prefer it

Figure 5-51. HeapAnalyzer WB8691.0

Notes:

The final heap analysis tool that is mentioned is the HeapAnalyzer from alphaWorks. This tool does “in process” heap analysis and is suited for quick and simple hierarchy inspection, and detecting leaks due to large collections of objects. An in-depth tutorial walks through using HeapAnalyzer to detect a leak in a heap dump.

The tutorial is available at: `http://www.ibm.com/support/docview.wss?uid=swg27006624`

5.3. Troubleshooting hangs

Troubleshooting hangs

Figure 5-52. Troubleshooting hangs WB8691.0

Notes:

WebSphere Education

Application server hangs defined

Clarify the nature and extent of the hang

- A “hang” is not the same as a “crash”
- Is the entire process truly hung, or does it still respond to some requests?
  – Test with sample “Snoop” servlet, wsadmin commands, and other commands
- Deadlocks:
  – Often, one process fails to respond to a request because it called another process that is itself hung; sometimes it is hard to find the true culprit
  – Deadlocks also can occur within the same Java process, where one thread is deadlocked on another

Figure 5-53. Application server hangs defined WB8691.0

Notes:

This hung transaction might consume system resources, such as processor time, when threads run unbounded code paths, such as when the code is running in an infinite loop. Alternatively, a system can become unresponsive even though all resources are idle, as in a deadlock scenario. Unless a user or a monitoring tool reports the problem, the system can remain in this degraded state indefinitely.

Notice that some deadlocks or other simpler hangs are contained entirely within one process: all the threads and monitors that are involved in the same JVM process and are visible in a single javacore. Other deadlocks can involve multiple processes. For example, one thread in process A waits for some response from process B, and some thread in process B waits for a response from process A.

In the latter case, it can be difficult to see the full extent of the deadlock or blockage by looking at just one JVM. When you look at a javacore from that type of situation, you can see a bunch of threads that made some remote request and are waiting for responses. You must look at the other remote processes to fully understand what is going on in that case. One typical example is where many threads in the server are all waiting for some response from the database.

WebSphere Education

Thread hangs

- WebSphere Application Server threads can become hung due to running poorly coded Java EE applications
- WebSphere administrators might not know that the application is slow or hung until a customer calls and complains
- WebSphere administrators should be alerted before significant problems arise
- Detecting whether a thread is hung or just taking a long time to respond can be a difficult problem to solve correctly

Figure 5-54. Thread hangs WB8691.0

Notes:

There are tools available, and WebSphere has a built-in monitor to help with 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

Thread hang examples in application code

- Understand root causes
  - Customer code enters an endless loop such as:

```
while (true) {
  i++;
}
```

  - Customer code waits infinitely such as:

```
run() {
  object1.wait();
}
```

  - Customer code creates a deadlock such as:

```
synchronized (object1) {
  synchronized (object2) {
    // Work with objects
  }
}
```

```
synchronized (object2) {
  synchronized (object1) {
    // Work with objects
  }
}
```


Figure 5-55. Thread hang examples in application code WB8691.0

Notes:

Here are some examples of coding practices that lead to hung threads. Such coding errors are typically discovered and fixed during the development and testing phase of an application.

WebSphere Education

WebSphere 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 a 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 numbers of threads that are blocked
   - Look for threads that are waiting after sending a request to some other process, now awaiting a response

Figure 5-56. WebSphere process hangs detection steps WB8691.0

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 diagnostic 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.

Later you examine two tools that can help in analyzing the thread memory dump.

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
  - WebSphere Application Server is not receiving traffic for some reason
  - Check front-end resources, networks, test clients, and so forth
  - Also, check timing of the thread dump
- Hundreds of threads
  - Shared resource is not available
  - Customer must control web container threads better
  - Might need more capacity

Figure 5-57. Thread dump analysis WB8691.0

Notes:

Thread dumps are snapshots 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 5-58. Javacore hang indicators WB8691.0

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 is 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 IBM

Thread state values

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, the thread might be waiting 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 suspends another thread
- Z: Zombie
  - The thread was killed
- P: Parked
  - The new concurrency API (`java.util.concurrent`) parks thread
- B: Blocked
  - The thread is waiting to obtain a lock that something else currently owns

Figure 5-59. Thread state values WB8691.0

Notes:

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.

WebSphere Education IBM

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, it might indicate looping logic
- If threads are in wait states, it 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 5-60. Javacore hang symptoms (1 of 2) WB8691.0

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 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 time (ms)	Minimum time (ms)	Maximum time (ms)	Total time (ms)
12.8	4.88	32.5	372

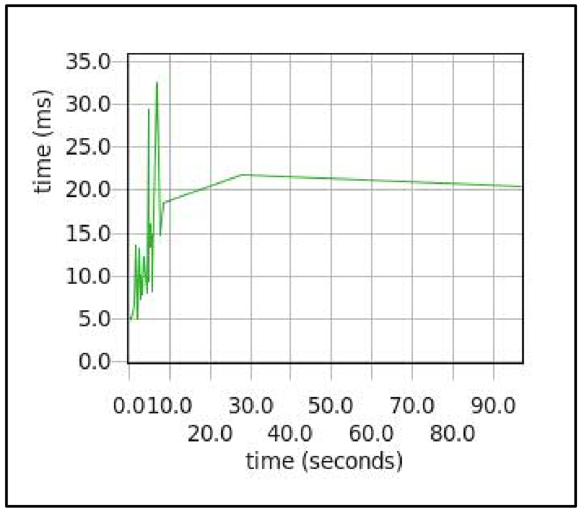


Figure 5-61. Javacore hang symptoms (2 of 2) WB8691.0

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.

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 5-62. Hang detection tools WB8691.0

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%.

These tools are covered in more detail in the following sections.

WebSphere Education

What is Thread and Monitor Dump Analyzer (TMDA)?

- TMDA is a tool to analyze thread dumps
  - Available from the IBM Support Assistant
- Used for:
  - Analyzing threads and monitors in javacore files
  - Comparing multiple javacore files 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:
  - Performance bottlenecks
  - Determining whether deadlocks exist
  - Determining whether threads are being blocked on monitors (might not be a deadlock)

Figure 5-63. What is Thread and Monitor Dump Analyzer (TMDA)? WB8691.0

Notes:

The ThreadAnalyzer tool is still available through IBM Support Assistant. It is deprecated and not actively maintained.

WebSphere Education

IBM Whole-system Analysis of Idle Time (WAIT)

- New web-based tool from IBM Research
- WAIT report identifies bottlenecks such as:
  - Waiting for database data
  - Waiting on hot locks
  - Garbage collection degradation of performance
  - Inefficient code
- The report shows you the full stack context responsible for each bottleneck
- Access at `http://wait.ibm.com`
- Using WAIT
  - Generate javacore files on your system
  - Upload to WAIT, data is encrypted in transit
  - View report interactively in your browser

Figure 5-64. IBM Whole-system Analysis of Idle Time (WAIT) WB8691.0

Notes:

The **IBM Whole-system Analysis of Idle Time (WAIT)** tool helps you identify performance bottlenecks. It works with both stand-alone Java applications and distributed applications with a central Java tier.

Each WAIT report identifies bottlenecks, such as waiting for database data, waiting on hot locks, garbage collection activity, and inefficient code. The report shows you the full stack context responsible for each bottleneck.

WAIT is **zero-install**: it requires no monitoring agents, and the report is an interactive web page. You do not need to restart your application server, and the data collection time should be minimal. Infrequent snapshots usually suffice, and each snapshot pauses the JVM for usually under a second.

WebSphere Education

Example WAIT report

- The report can be viewed interactively by hovering the cursor over sections for more detail

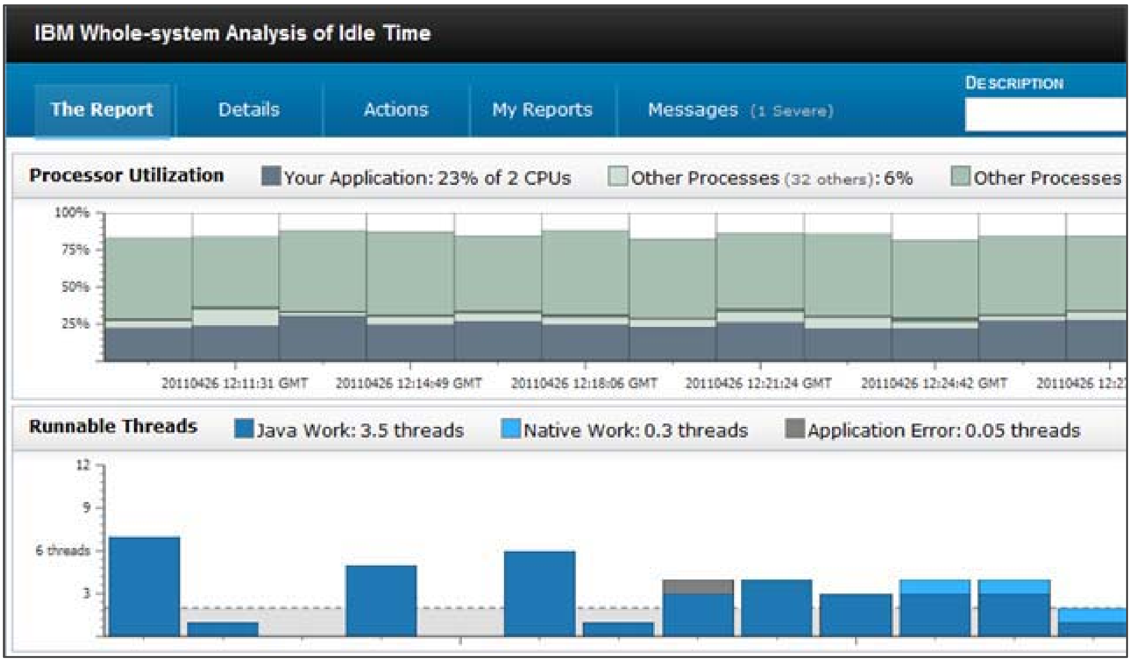


Figure 5-65. Example WAIT report WB8691.0

Notes:

The report has three timelines:

1. The top (gray) timeline shows processor activity and how much your workloads contributed to it. Lots of dark gray is generally good, meaning that your workloads were running smoothly for the entire duration over which data was collected.

2. The middle (blue) timeline shows what Java work is running. This timeline generally mimics the top timeline in its form (although not always). It shows Java threads that are runnable (either running or capable of being run). Again, if there are high blue bars all the way across, it generally means that your workload was running unimpeded. However, there can still be hot code that contributes to poor performance. You can click the blue bars of the timeline or the aggregate summary at right to see the activities that use most of the processor. The middle timeline also shows when there is more work than the processors in the system can accommodate. This situation occurs whenever there is light blue above the dotted line (representing the number of processors in the system). Especially if there are such light blue caps uniformly over the course of execution, it suggests that the number of processors is underprovisioned, and that your workload's performance would increase with the addition of more processors.

3. The third timeline (not shown in the screen capture) shows Java threads that are delayed and not capable of being run at the current time. WAIT has a patented set of expert rules to diagnose these situations and the general reason why threads are delayed. In the example there are many such reasons: “Delayed by Remote Request”, “Blocked on Monitor”, “Delayed by Disk I/O”, and “Waiting for GC Work” (GC = garbage collection). As with the middle timeline, you can click any of the individual bar segments or the legend at left to learn more about the causes of delay; more about that later. However, if your report shows any significant number of delayed threads, it suggests that the performance of your workload is subpar and can be improved with appropriate tuning of software or hardware. This situation is true even if the middle blue bars are exhausting the processor’s capacity. Delays in the third timeline mean that work is taking longer to run and is in transaction-oriented systems, likely to increase user response time, in addition to using resources like memory, thus reducing efficiency.

5.4. Troubleshooting crashes

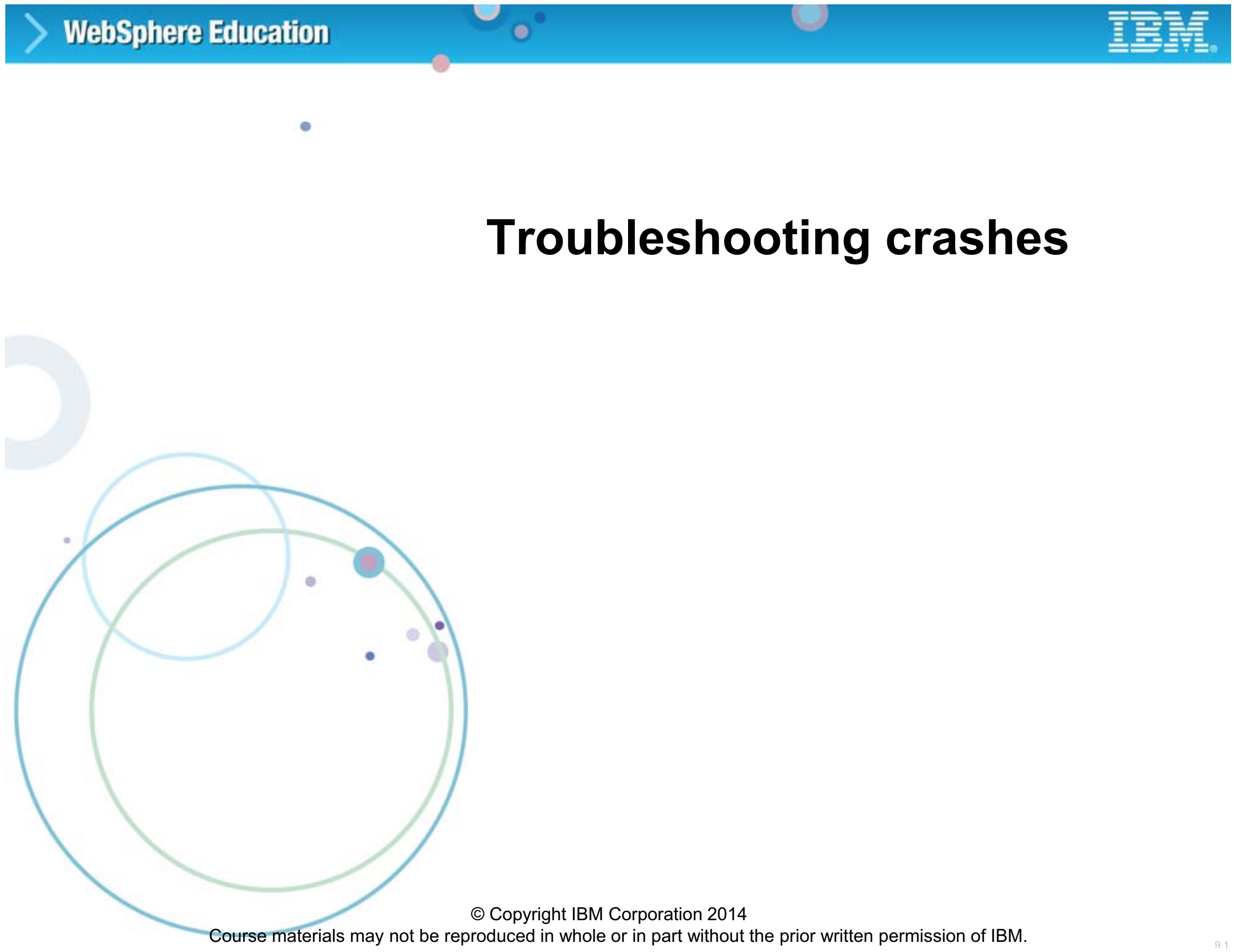


Figure 5-66. Troubleshooting crashes WB8691.0

Notes:

WebSphere Education

JVM process crash defined

- Not the same as thread hangs
- Symptoms
  - Process was stopped because of a Java exception or an OS signal
- Usual causes
  - Bad JNI call or library problem
  - Segmentation violations while running native code
  - Out-of-memory exception
  - Call stack overflow
  - Unexpected exception (for example, out of disk space)
  - Optimizer failure (for example, JIT)

Figure 5-67. JVM process crash defined WB8691.0

Notes:

All JVMs use the just-in-time (JIT) compiler to compile heavily used Java bytecode into native instructions during server runtime to enhance performance.

Crashes can be categorized into three main types:

- Failures that are a result of some type of problem in the environment where the JVM runs, such as a lack of resources: OutOfMemory, StackOverflow, unexpected exception, and more. These problems are solved by fixing the application or the environment in which the problem occurred.
- Failures that are a result of an internal error (a defect) in the JVM implementation (for example, JIT). These problems are fixed by applying a JDK patch.
- Failures that are a result of an internal error in some external third-party JNI library that the customer uses and are loaded in the JVM. These problems are fixed by applying a patch to the third-party library, or avoiding the use of that particular library.

For crashes due to OutOfMemory or StackOverflow, the JVM usually gets a chance to do some cleanup before exiting, and in particular, it usually produces a javacore file. With internal JVM errors and JNI errors, you often get a “hard” crash, with just a core file but no javacore file.

WebSphere Education

Crash problem determination: Data to collect

- Javacore files
  - Also known as javadump or thread dump files
  - Text file that an application server creates during a failure
  - Can also be triggered manually
  - Error condition is given at the top of the javacore file
- Core files
  - Also known as process dumps or system core files
  - Created by underlying operating system
  - Complete dump of the virtual memory for the process
  - Can be large
  - Tools available to parse files into readable formats
- Steps to collect data
  - Look for a javacore file that is automatically created during a crash
  - If no javacore was generated, look for a system core dump

Figure 5-68. Crash problem determination: Data to collect WB8691.0

Notes:

When a process crashes, the underlying operating system often creates a process dump. Process dumps are a snapshot of the contents of the entire process in binary format. Information about what was going on in that process when the memory dump was taken can be extracted and analyzed.

The javacore files are text files that contain only a summary of the information that is most pertinent to the most common problems.

Javacores are enabled by default on IBM JDK; they can be disabled by setting the `DISABLE_JAVADUMP` environment variable to TRUE.

On Solaris and HP-UX, instead of a separate javacore file, a memory dump of all thread stack traces is written to `native_stdout.log`.

Here are some process dump file locations:

- AIX: Output is written to a core file named `core.&processid.&timestamp.txt` that is in the current working directory.
- Windows: Output is written to a file named `core.&processid.&timestamp.dmp` into the same directory that is used for `JAVADUMP`.

- Linux: Output is written to a core file named `core.&timestamp.&processid.dmp` that is in the current working directory, for example: `/opt/IBM/WebSphere/AppServer/profiles/Dmgr01`

WebSphere Education

Make sure that a full core dump is produced

- Underlying operating system might have settings that prevent creation of a full core dump when a process crashes
  - For example, **`ulimit`** on UNIX systems might specify a limit on the size of core file that is too small and the core dump is truncated
  - Some operating systems have a parameter to control the type of core files
  - On AIX, use the command `lsattr -Elsys0|grep full` to check whether fullcore is configured or not

- Make sure that a full core dump can be created before a problem occurs
  - Avoid re-creation of problem, especially in a production environment

Figure 5-69. Make sure that a full core dump is produced WB8691.0

Notes:

The procedure to enable full core dump is different depending on the operating system:

- For Linux: `http://www.ibm.com/support/docview.wss?&uid=swg21115658`
- For AIX: `http://www.ibm.com/support/docview.wss?&uid=swg21052642`
- For Windows: `http://www.ibm.com/support/docview.wss?&uid=swg21053924`

WebSphere Education IBM

Diagnostics Collector

- You can configure the JVM to use the Diagnostics Collector to gather documentation and diagnostic data automatically after detecting a runtime problem such as a crash or out-of-memory condition
  - Enable by using the generic JVM argument `-Xdiagnosticscollector`
- The Diagnostics Collector gathers system memory dumps, javacore files, heap memory dumps, verbose GC log (if present), and JVM trace files that match the time stamp for the Java problem that caused the collector to start
  - Outputs a single compressed file
  - For example, `java.gpf.<time_stamp>.<event_ID>.<pid>.zip`
- Available on IBM JDK 1.6 and 1.7
- For more details, see the topic “The Diagnostics Collector” in the Java Diagnostics Guide 6

Figure 5-70. Diagnostics Collector WB8691.0

Notes:

The JVM automatically starts the Diagnostics Collector after detecting a runtime problem and producing diagnostic data. The Diagnostics Collector gathers diagnostic data files that the JVM produces and stores them in a single archive file.

If a system dump is produced, the Diagnostics Collector runs jextract on the memory dump and collects the jextract output. There is an option to disable this behavior if jextract is not required. All actions are logged in a text file, which is also copied into the output compressed file.

The Diagnostics Collector gathers system dumps, Java dumps, heap dumps, verbose GC log (if present), and JVM trace files that match the time stamp for the Java problem that caused the collector to start. The Diagnostics Collector copies the diagnostic files that it finds into a single output compressed file. The output file is named by using the type of problem that occurred, the time stamp of the problem event, and the process ID of the Java application; for example, `java.gpf.20081219.105104.9221.zip`.

At JVM startup, the Diagnostics Collector runs a diagnostic configuration check. It checks for any settings that disable Java diagnostics. If any are found, a warning is printed on stderr and in the `JavaDiagnosticsCollector.log` file. Fix the settings that are identified from any warning messages before restarting your Java application. Fixing warnings makes it more likely that the right data is available for IBM Support to diagnose a Java problem.

WebSphere Education

UNIX operating system common signals

- `SIGQUIT (kill -3)`
  - Indicates that a command was issued to generate a thread dump
  - Typically, it does not end the JVM process
- `SIGILL (kill -4)`
  - Means that an illegal instruction was issued
  - This signal can mean a corruption of the code segment or a branch that is not valid within the native code
  - This signal often indicates a problem with JIT-compiled code
- `SIGSEGV (kill -11)`
  - Indicates an operation that is not valid in a program
  - Example: Accessing an illegal memory address
  - This signal is typically indicative of a programming problem in one of the native libraries

Figure 5-71. UNIX operating system common signals WB8691.0

Notes:

Keep in mind that non-JVM processes (that is, non-Java-based applications) can end if a quit signal is received.

The javacore file displays the operating system signal that caused the javacore file to be written. Some signals also produce a core file in UNIX operating systems. The JVM signal handler library is `libhpi.a`.

Following are some of the most commonly seen signals in javacore files:

- `SIGQUIT`: This signal is sent when a `kill -3` command is run against the JVM process. This signal typically does not end the JVM process. It generates a thread dump (javacore) for diagnosing a potentially hung JVM process.
- `SIGILL`: This signal is the equivalent of a `kill -4` command. It means that an invalid instruction was run. It can mean a corruption of the code segment or a branch that is not valid within the native code. This signal often indicates a problem that is a result of JIT-compiled code.

- `SIGSEGV`: This signal is equivalent to a `kill -11` command. It indicates an operation that is not valid in a program, such as accessing an invalid memory address. This signal is typically indicative of a programming problem in one of the native libraries.

Signals `-1`, `0`, `OUTOFMEMORY`, and `SIGNONE` are the memory signals. These signals are described in a later unit.

Signals `10`, `11`, `SIGBUS`, and `SIGSEGV` indicate an application server crash.

WebSphere Education

Windows operating system common signals

- **Memory access error**
  – Invalid memory address
  – JVM action: Javacore file and stop the process
- **Illegal access error**
  – JVM action: Javacore file and stop the process

Figure 5-72. Windows operating system common signals WB8691.0

Notes:

The signals that Windows uses are entirely different from the signals that UNIX uses. Here is a list of some of the operating system signals that are produced on Windows. Some signals also produce a `user.dmp` file. The JVM signal handler library is `hpi.dll`.

WebSphere Education IBM

Javacore subcomponents helpful for crash debugging

TITLE	Shows basic information about the event that caused the generation of the javacore file, the time it was taken, and the file name
GPINFO	• Shows some general information about the operating system • If the memory dump resulted from a general protection fault (GPF), information about the failure is provided; namely, the fault module is identified
ENVINFO	Shows information about the JRE level, details about the command line that started the JVM process, and the JVM environment
THREADS	Identifies the current thread and provides a stack trace

Figure 5-73. Javacore subcomponents helpful for crash debugging WB8691.0

Notes:

It is important to understand what signal caused the javacore. This signal determines how to interpret the information in the javacore file. Does the signal indicate a JVM crash? Does the signal information state it was due to an operator action (noted as a “user” dump event)? If the javacore is written to diagnose a hung JVM process, the steps you take are different than if the javacore is due to a JVM crash.

Use the Java command line to determine whether the javacore is for a WebSphere Application Server Java process. If so, which WebSphere Java process it is for: an application server process, an administrative server process, a jmsserver, a node agent, or some other WebSphere Application Server process?

The current thread is the thread that is running when the signal is raised that causes the javacore to be written. The **current thread details** show the Java and native stacks of the current thread. The most recent calls are at the top of the stack.

- The Java stack shows the Java method calls that the current thread made.
- If there is a native stack, this native stack contains the most recent events that the thread ran. As the name implies, these events are in native code (libraries) called by Java.

This output can be used to determine what was running when the crash occurred.

To verify that the javacore file is complete, you should see an "`END OF DUMP`" string at the end of the file.

WebSphere Education

Steps if the cause of the crash is not identified

- Frequently, the javacore file does not clearly identify the cause of the signal
- Often the native stack shows the following message:

```
 ----- Native Stack -----
unable to backtrace through native code - iar 0x3062e73c not
in text area (sp is 0x2ff21748)
```

- Steps that you can take:
  - Upgrading to a more recent JDK can sometimes resolve a problem
  - Use the core file (on UNIX) or `user.dmp` file (on Windows) to see whether this file provides more information
  - Sometimes a bad Java SDK installation can cause problems

Figure 5-74. Steps if the cause of the crash is not identified WB8691.0

Notes:

Frequently when the crash is not identified in the javacore file, it is due to a problem with the JIT compiled code. Disabling JIT compilation can be used to confirm if that is the case. The JIT compiler is a JVM optimization. There can be significant performance degradation when the JIT is disabled, as much as 20% or more.

The problem might be in the JIT compiler. If so, you might be able to skip JIT compiling for a method, if the problem is in the code that is generated for a method or in the code generation itself.

Sometimes a bad Java SDK installation can cause problems. The full version for the javacore file comes from the `libjvm.a`. The full version for the Java `-fullversion` command is from the Java executable file. If there is a discrepancy between these two dates, either the wrong `libjvm` is picked up due to an error in the library path statement, or there is a problem in the JVM installation.

If you did not get a javacore file, check to see whether a process dump (core file) exists. If it does, it is still likely that you have an application server crash.

WebSphere Education IBM

What is DTFJ?

- DTFJ (Diagnostic Tooling Framework for Java) is a new technology within the IBM JDK to analyze and diagnose problems in Java applications
  – Read RAS artifacts from a JVM (for example, a core file) and extract all kinds of useful information from that memory dump
- Not just one tool: an extensible framework for building many different tools
- By providing common framework, the use of specific tools for specific JVM artifacts is avoided

Figure 5-75. What is DTFJ? WB8691.0

Notes:

The Diagnostic Toolkit and Framework for Java (DTFJ) API is a Java-based API for accessing postmortem information from the system dump of a Java process. This framework allows tool writers to access information from the memory dump about the system, process, Java VM, and Java application without having to understand how the relevant structures are laid out in memory. This framework makes the ability to write postmortem tools far more accessible.

There are many different JRas artifacts, such as system dumps (core files, MiniDumps), heap dumps (`.phd` files), javacore files (javacore, `.txt` files), but no common tools. There are many good reasons for different file formats, but often tools are JVM software-specific (jformat, jcore, kca).

The DTFJ-based Dump Analyzer tool was created to convert system dump files into human readable format.

In the past, there were various other ad hoc tools, and sometimes people even used a simple low-level debugger (for example, dbx, gdb) to try to understand a crash. Such methods are not as good as running a DTFJ tool, but sometimes that is all you can do.

WebSphere Education IBM

Components of the DTFJ family

- **`jextract`**: A tool to capture information from a JVM system memory dump (for example, core file) and package it into a platform-independent format
- DTFJ library or core library: A library that parses the contents of the system dump file that jextract packages, and provides access to its contents in a standardized manner, through a standard API
- DTFJ-based tools: A collection of tools that call the DTFJ library through the DTFJ API to present, and analyze information in various ways useful to the users

Figure 5-76. Components of the DTFJ family WB8691.0

Notes:

The jextract tool is in *<was_root>*`\java\jre\bin`.

WebSphere Education

What is the Interactive Diagnostic Data Explorer?

- Interactive Diagnostic Data Explorer (IDDE) is a GUI-based alternative to the dump viewer (`jdmpview` command)
- IDDE provides the same function 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
- The support that is available in the investigation log includes the following items:
  - Command assistance
  - Auto-completion of text, and some parameters such as class names
  - The ability to save commands and output, which you can then send to other people
  - Highlighted text and flagging of issues
  - The ability to add your own comments
  - Support for using the Memory Analyzer from within IDDE

Figure 5-77. What is the Interactive Diagnostic Data Explorer? WB8691.0

Notes:

Interactive Diagnostic Data Explorer (IDDE) is the strategic tool for allowing interactive analysis of JVM problems with post mortem artifacts such as core files or javacores. It is a lightweight tool that allows you to quickly get information from the artifact you are investigating where you are not sure what the problem is and you want to avoid starting resource-intensive analysis.

WebSphere Education

Using Interactive Diagnostic Data Explorer (IDDE)

- IDDE supersedes the Dump Analyzer tool
- Distributed and accessed through IBM Support Assistant
- Based on DTFJ, which provides cross system support
- Supports
  - Javacore files
  - Heap dumps (PHD file format)
  - System dumps (also known as core files)
  - Also supports multiple dump files that are contained in a compressed file, and dumps that were created on the z/OS operating system

Figure 5-78. Using Interactive Diagnostic Data Explorer (IDDE) WB8691.0

Notes:

For information about how to enable system dumps in the IBM JVM and running jextract, see the IBM JDK Diagnostics Guide at: `http://www.ibm.com/developerworks/java/jdk/diagnosis/`

WebSphere Education IBM

IDDE features

Attempts to diagnose common JVM problems

- Deadlock in Java code
  - Report thread names, locations, and other information
- Out-of-memory condition
  - Report populations, large collections, and large objects
  - Summarize the native memory usage
- Internal error (general protection fault, segmentation violation)
  - Is failure in non-IBM native code?
  - Probably use coding error, report location, and other details
  - If using JDK V5 or later, it might suggest running with **`-Xcheck:jni`** command-line option
  - Otherwise, call IBM Support
- Otherwise, it generates a default summary report
  - Suggested action is to call IBM Support and provide the output

Figure 5-79. IDDE features WB8691.0

Notes:

If one of the common JVM problems is not found, the general script generates a default summary report.

Specifically, for crashes, Dump Analyzer can read core files and attempt to determine the cause of a JVM crash. The Dump Analyzer can be useful in cases where a javacore is not available. Obviously the output can be useful on other common JVM problems, such as hangs and OutOfMemory conditions.

The `-Xcheckjni` command-line option causes the JVM to monitor JNI usage. When `-Xcheckjni` is used, the JVM writes a warning message when more than 16 local references are required at run time. (The default root set is large enough to contain 16 local references per J2N transition.) This output might indicate that you should manage local references more explicitly, with the JNI functions available.

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Describe components of JVM and overall architecture
- Describe garbage collection (GC) and GC tuning policies
- Identify a sluggish JVM and detect bottleneck problems
- Explain how to tune the heap size
- Define out-of-memory conditions
- Describe tools for analyzing out-of-memory problems
- Detect a hang condition
- Describe tools for analyzing hangs
- Define and detect a crash
- Describe the tools that are available for troubleshooting a crash

Figure 5-80. Unit summary WB8691.0

Notes:

WebSphere Education

Checkpoint questions

1. True or false: In addition to a mark and sweep, a compaction occurs for every garbage collection cycle.
2. True or false: In general, the larger the heap size, the longer the pause time during garbage collection.
3. List three causes of Java heap OutOfMemory error.
4. True or false: An application can seem to be in a hung state if two threads are deadlocked.

Figure 5-81. Checkpoint questions WB8691.0

Notes:

Write down your answers here:

1.

2.

3.

4.

WebSphere Education

Checkpoint answers

1. False: A compaction occurs only if the mark and sweep phases cannot recover enough heap space.

2. True.

3. The following are three causes of Java heap OutOfMemory error:
   - Insufficient maximum heap size for user load
   - Memory leak in the Java code
   - Memory leak in native code

4. True.

Figure 5-82. Checkpoint answers WB8691.0

Notes:

Unit 6. Configuration problems

What this unit is about

This exercise covers how to troubleshoot some of the major configuration problems when working with IBM Business Process Manager. It also shows how to take advantage of the information that is stored in some of the key database tables.

What you should be able to do

After completing this unit, you should be able to:

- Describe BPMConfig content
- Describe troubleshooting options when working with online Process Server
- Describe some key database tables for troubleshooting

How you will check your progress

- Checkpoint

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Describe BPMConfig content
- Describe troubleshooting options when working with online Process Server
- Describe some key database tables for troubleshooting

Figure 6-1. Unit objectives WB8691.0

Notes:

WebSphere Education IBM

Topics

- BPMConfig properties
- Online Process Server
- Key database tables

Figure 6-2. Topics WB8691.0

Notes:

6.1. BPMConfig properties

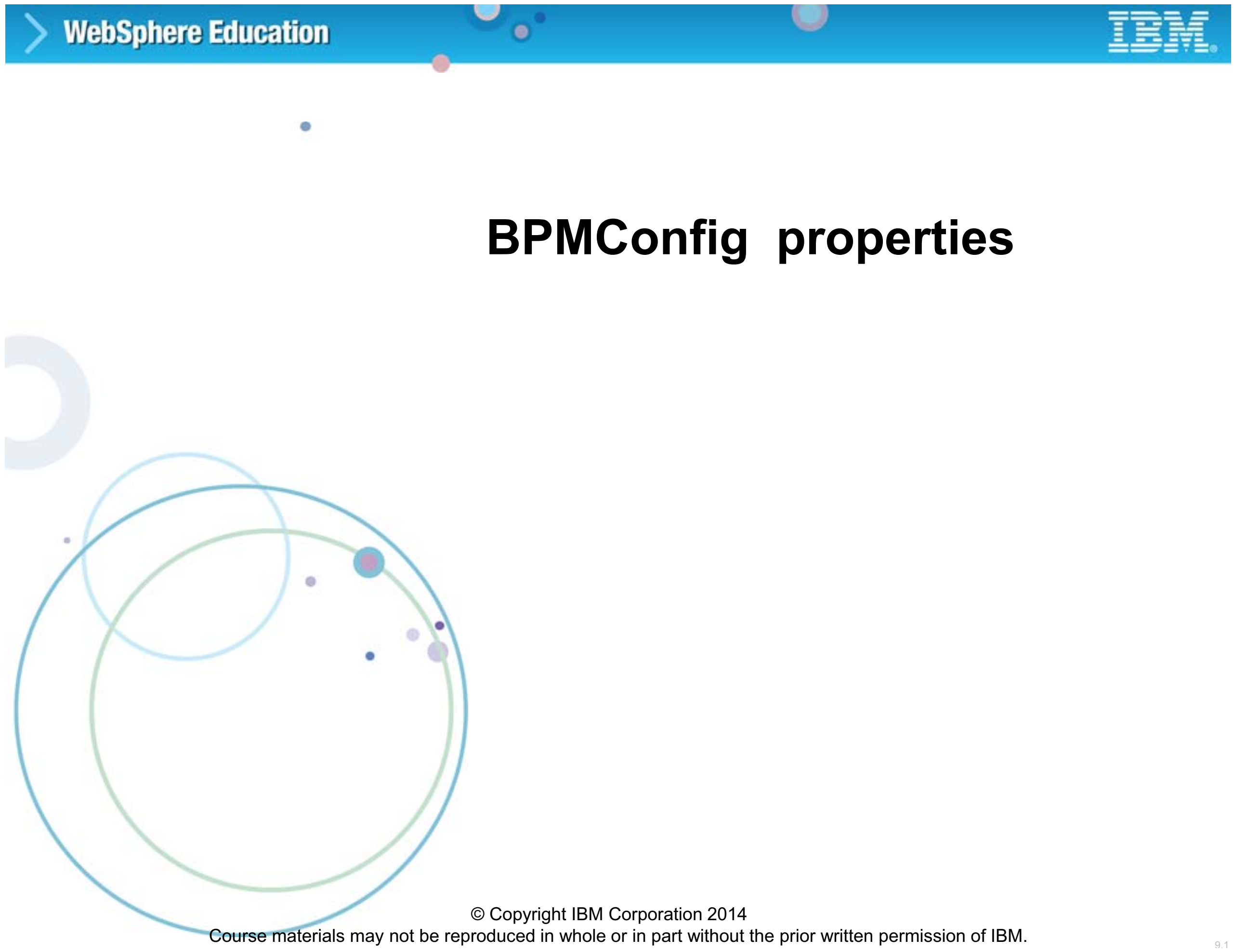


Figure 6-3. BPMConfig properties WB8691.0

Notes:

WebSphere Education

BPMConfig usage

```
Usage: BPMConfig <action> <type> <propertiesFile>
<action>: -create, -validate, -start
<type>: -profile, -de, -sqlfiles
<propertiesFile>: Properties file representing DE config
 -profile and -sqlfiles valid only for -create action
```

- **`-create`** creates the profiles and DE (deployment environment) configuration for profiles that match installPath and host name with the current machine
- **`-validate`** validates that the specified properties file matches the existing DE configuration
- **`-start`** starts the DE within a deployment manager profile

Figure 6-4. BPMConfig usage WB8691.0

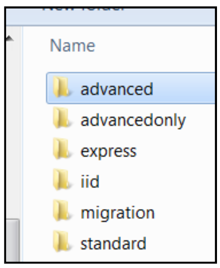
Notes:

The `BPMConfig` command is used to create or extend a typical network deployment environment. For example, it can also be used to create the database scripts and profiles, start and stop the deployment environment, and validate the deployment environment configuration.

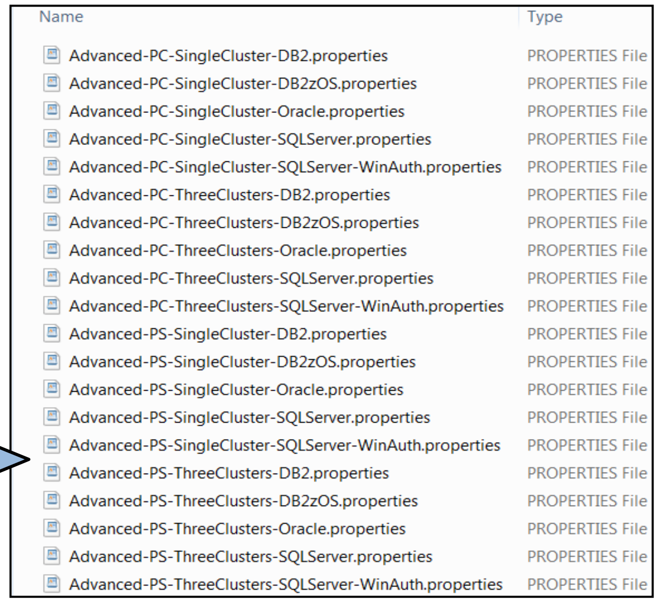
WebSphere Education

BPMConfig properties file

- Sample properties files are provided under `<WAS_HOME>\BPM\samples\config`
- Select the folder for the type



- Find the sample property file for the pattern and database type



Name of file =
***<std or adv> - <PS or PC> - <1 or 3 clusters> - <db type>*.properties**

Figure 6-5. BPMConfig properties file WB8691.0

Notes:

The `BPMConfig` command uses a properties file that contains all of the values that are used in the configuration of your deployment environment. Sample properties files are provided for you to copy and customize to configure your own environments.

WebSphere Education

BPMConfig things to know

- When troubleshooting BPMConfig related activities, check whether the content is correct and consistent

Something about the property	Examples

Some propertied are not expected to change	`bpm.de.roleMapping.1.name=DeAdmin`
	`bpm.cell.roleMapping.1.name=CellAdmin`
	`bpm.de.db.1.roleMapping.1.name=DbUser`
	`bpm.de.db.1.roleMapping.2.name=DbUserXAR`

Must use predefined values	`bpm.de.deferSchemaCreation`
	`bpm.de.type`
	`bpm.de.environment`
	`bpm.de.cluster.1.capabilities`
	`bpm.de.db.1.dbCapabilities`
	`bpm.de.db.1.type`

Some properties depend on others	`bpm.cell.db` and `bpm.de.cluster.XXX.db` depending on `bpm.de.db.XXX`

Figure 6-6. BPMConfig things to know WB8691.0

Notes:

Use the `BPMConfig` command to create a typical network deployment environment, instead of using the Deployment Environment wizard or wsadmin commands. Use the `BPMConfig` command to create the BPMDmgr and BPMNode profiles, instead of using the Profile Management tool or the manageprofiles utility. You can also use BPMConfig to:

- Extend an existing deployment environment that was created with the BPMConfig command
- Create new profiles
- Create a custom context root for the deployment environment
- Create the required database tables or generate the SQL scripts that you can use to create your database tables later
- Validate the configuration settings in an existing configuration properties file
- Start and stop the deployment environment

WebSphere Education

Some properties that are not expected to change

bpm.de.**roleMapping**.1.**name**=DeAdmin

```
bpm.de.authenticationAlias.1.name=MyDepEnv01Alias
bpm.de.authenticationAlias.1.user=MyDepEnv01User
bpm.de.authenticationAlias.1.password=MyDepEnv01PW
…
bpm.de.roleMapping.1.name=DeAdmin
bpm.de.roleMapping.1.alias=MyDepEnv01Alias
```

bpm.cell.**roleMapping**.1.**name**=CellAdmin

```
bpm.cell.authenticationAlias.1.name=MyCell01AdminAlias
bpm.cell.authenticationAlias.1.user=MyCell01AdminUser
bpm.cell.authenticationAlias.1.password=MyCell01AdminPW
…
bpm.cell.roleMapping.1.name=CellAdmin
bpm.cell.roleMapping.1.alias=MyCell01AdminAlias
```

bpm.de.db.1.**roleMapping**.1.**name**=DbUser

bpm.de.db.1.**roleMapping**.2.**name**=DbUserXAR

```
bpm.de.db.1.dbCapabilities=CommonDB
bpm.de.db.1.databaseName=CMNDB1
...
bpm.de.db.1..roleMapping.1.name=DbUser
bpm.de.db.1..roleMapping.1.alias=BPM_DE1_DB_ALIAS
..
bpm.de.db.1..roleMapping.2.name=DbUserXAR
bpm.de.db.1..roleMapping.2.alias=BPM_DE1_DB_ALIAS
```


Figure 6-7. Some properties that are not expected to change WB8691.0

Notes:

WebSphere Education

Some properties must use predefined values

bpm.de.deferSchemaCreation	true, false
bpm.de.type	Express, Standard, Advanced, or AdvancedOnly
bpm.de.environment	Process Center or Process Server
bpm.de.cluster.[n].capabilities	• Application, Support, Messaging • If it is a single cluster environment, specify all three: Application, Support, and Messaging
bpm.de.db.[n].dbCapabilities	• For Advanced (only) configurations the options are: Messaging, BusinessSpace, CommonDB, BPC, and CellScopedDB • For Standard and Express configurations, the options are: Messaging, BusinessSpace, ProcessServer, EmbeddedECM, and Business Performance Data Warehouse • Full Advanced could have all capabilities
bpm.de.db.[n].type	DB2, DB2zOS, Oracle, or SQLServer

Figure 6-8. Some properties must use predefined values WB8691.0

Notes:

WebSphere Education IBM.

Some properties depend on others

- You must be able to connect the dots
  - For example: You defined an AppCluster, and provided some "keys" (can be any words, just has to match later) to say which databases this cluster uses:

```
bpm.de.cluster.1.name=AppTargetCluster01
bpm.de.cluster.1.capabilities=Application
...
bpm.de.cluster.1.db=ProcessServerDB, SharedDb, BusinessProcessChoreographerDb,
BusinessSpaceDb. ContentMgmtDb
```

  - Later in the properties file, the database that is described by each of these "keys" is defined

```
bpm.de.db.[n].name=BusinessProcessChoreographerDb
bpm.de.db.[n].dbCapabilities=BPC
bpm.de.db.[n].databaseName=BPEDB1
bpm.de.db.[n].type=DB2
bpm.de.db.[n].hostname="MyHost3.ibm.com bpm.de.db.6.portNumber=50001"
bpm.de.db.[n].roleMapping.1.name=DbUser
bpm.de.db.[n].roleMapping.1.alias=BPM_DE1_DB_ALIAS
bpm.de.db.[n].roleMapping.2.name=DbUserXAR
bpm.de.db.[n].roleMapping.2.alias=BPM_DE1_DB_ALIAS
bpm.de.db.[n].schema=MYCDE1
```


Figure 6-9. Some properties depend on others WB8691.0

Notes:

Run the `BPMConfig` command with the same properties file on all computers that participate in the deployment environment. You must first run the command on the computer that has the deployment manager profile and then run it on each computer that has a managed node. At any time, only one profile creation can be done on a computer, and only one node federation can be done against a particular deployment manager. For this reason, if you are creating multiple profiles at once on different computers, you must use the `federateLater` option. You then run the command with the `create de` option sequentially on each computer to federate the managed nodes.

6.2. Online Process Server

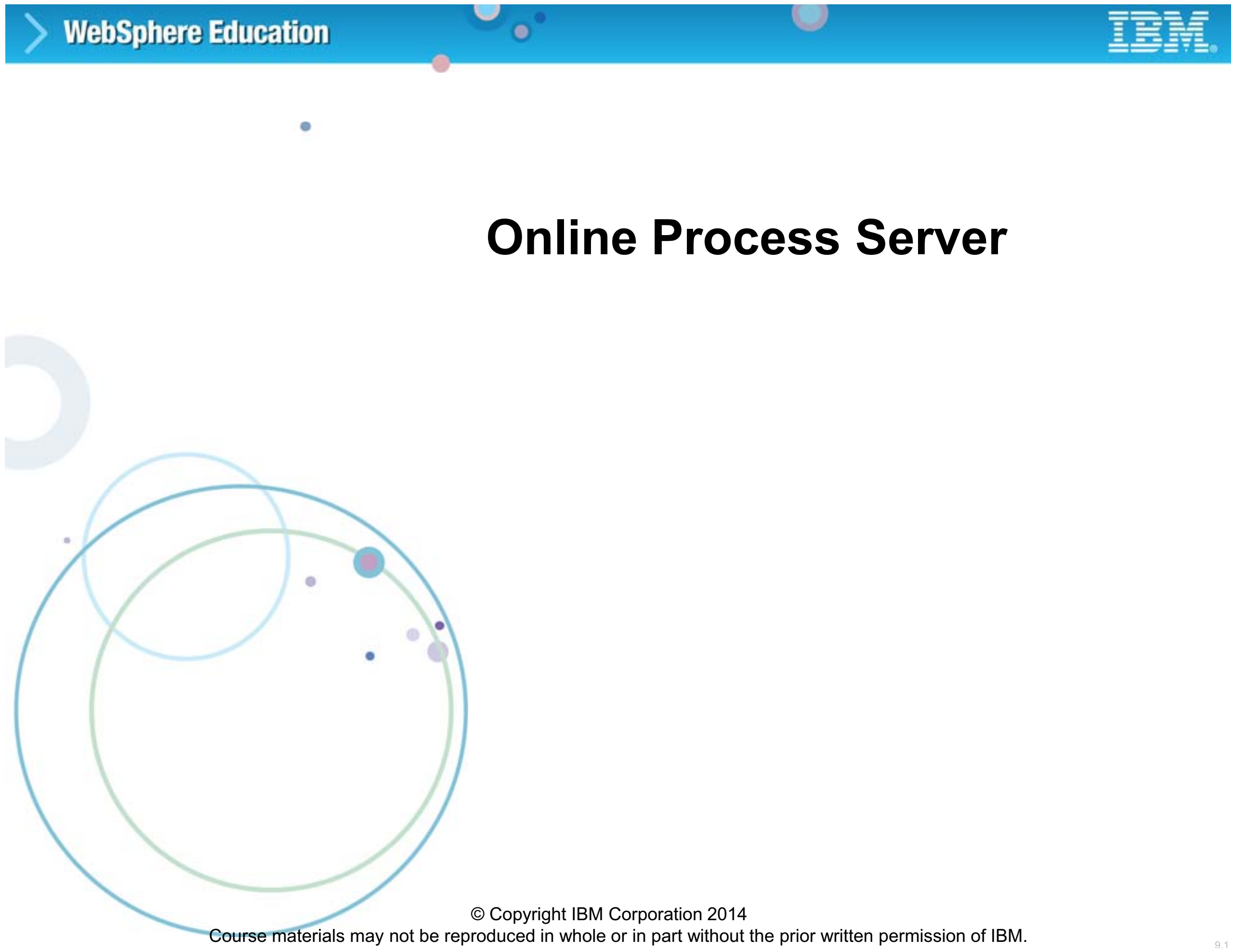


Figure 6-10. Online Process Server WB8691.0

Notes:

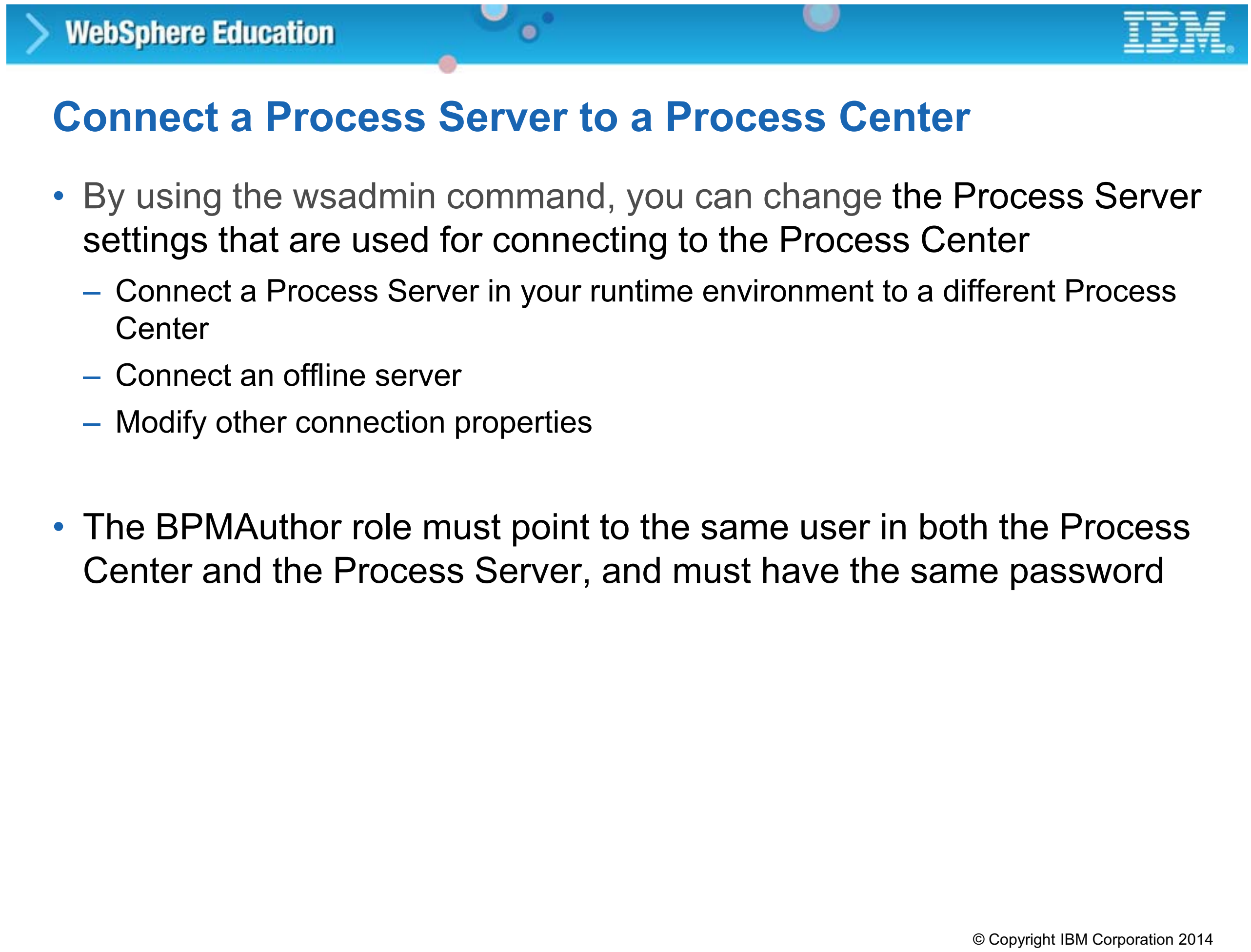


Figure 6-11. Connect a Process Server to a Process Center WB8691.0

Notes:

You can update the server configuration to connect a Process Server in your runtime environment to a different Process Center to connect an offline server, or to modify other connection properties. A few settings can be updated with the Authentication Alias Configuration, and the rest can be updated with the WebSphere command-line administration tool (wsadmin) AdminConfig commands.

Typically, each Process Server in a runtime environment is connected to a Process Center; a single Process Center can be connected to multiple servers. You can install process application snapshots from the Process Center to one or more of these connected Process Servers.

WebSphere Education IBM

Creating the authentication alias (1 of 2)

- IBM Business Process Manager uses an authentication alias that is mapped to the ProcessCenterUser role to connect the Process Center to the Process Server
  - By default, the authentication alias is defaulted to the DeAdmin authentication alias
- Modify authentication aliases with one of the following methods:
  - Administrative console
  - (wsadmin) AdminConfig commands

Figure 6-12. Creating the authentication alias (1 of 2) WB8691.0

Notes:

WebSphere Education

Creating the authentication alias (2 of 2)

- From the administrative console, click **Security > Global Security > Java Authentication and Authorization Service > J2C authentication data**

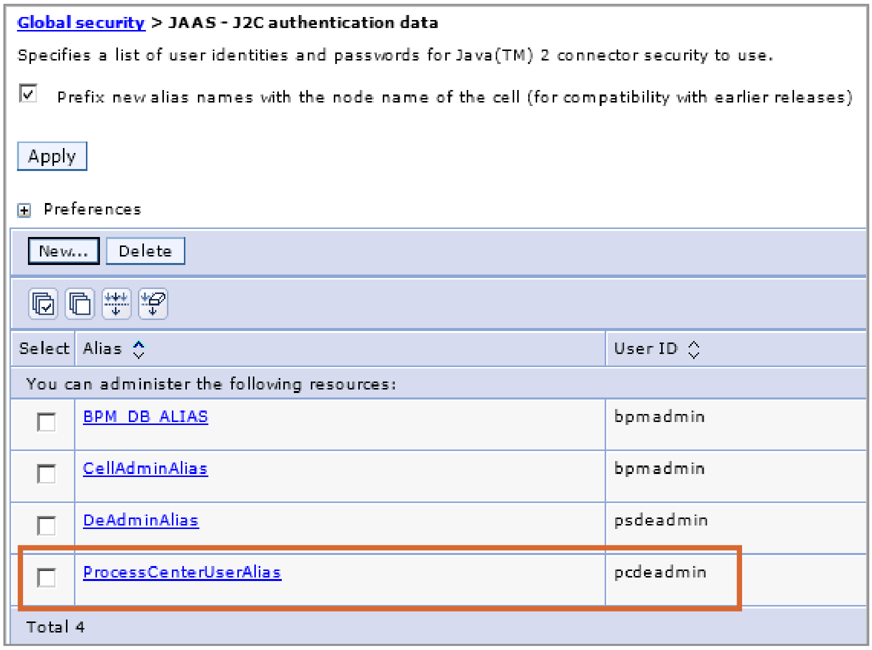


Figure 6-13. Creating the authentication alias (2 of 2) WB8691.0

Notes:

If it does not exist, create the ProcessCenterUserAlias. For ProcessCenterUserAlias, use a valid user name and password from the Process Center environment. The user does not need any special authorization in Process Center.

If you plan to use a user other than DeAdmin (the default) to deploy snapshots from Process Center to the runtime Process Server, create a new BPMAuthorAlias. For BPMAuthorAlias, use a valid user name and password from the Process Server environment that has the authority to access and deploy snapshots to the runtime Process Server. This user is saved in the LSW_SERVER table in the Process Center database and used when the process application is deployed to this Process Server.

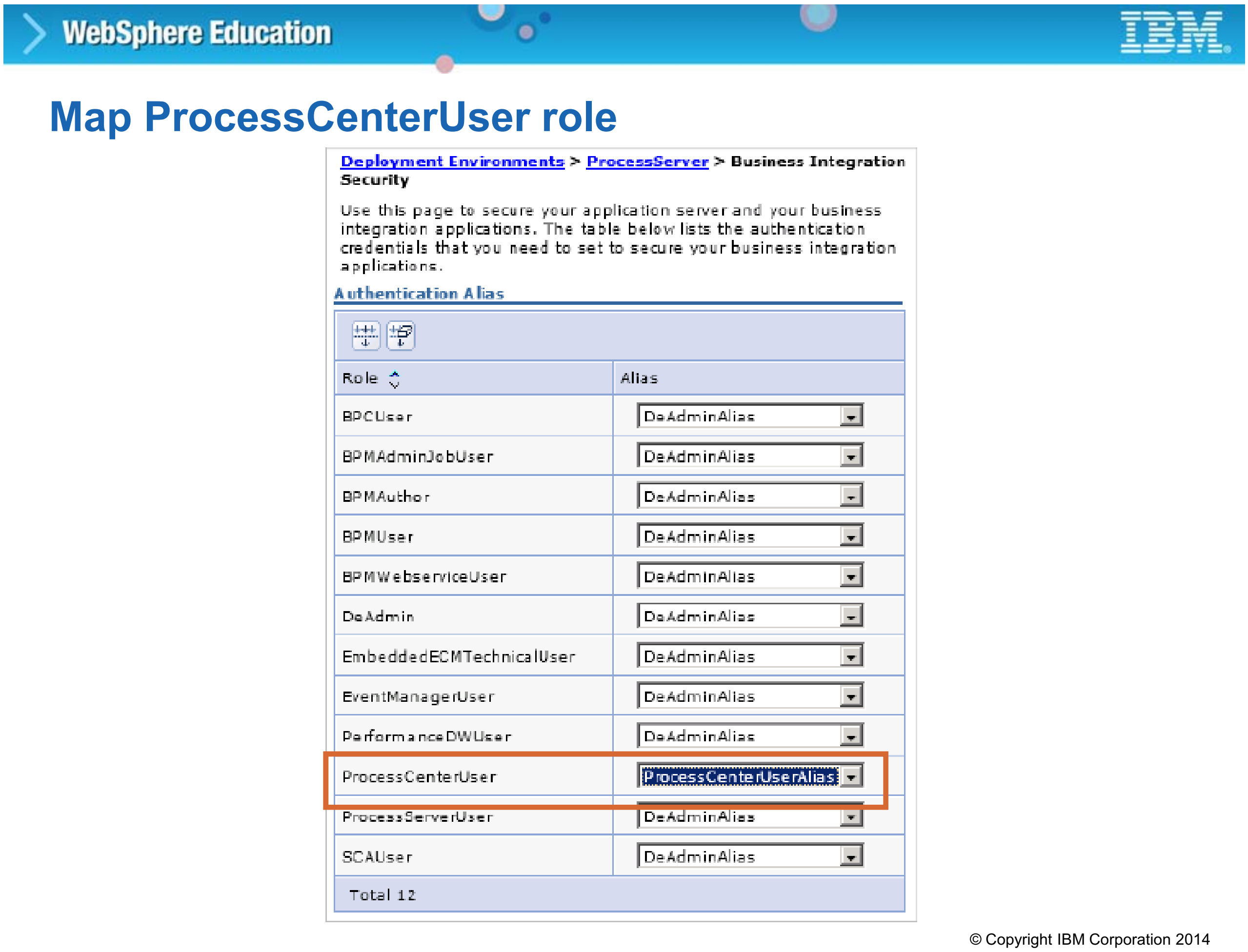


Figure 6-14. Map ProcessCenterUser role WB8691.0

Notes:

In the Business Process Manager security roles screen, map the ProcessCenterUser role to ProcessCenterUserAlias and the appropriate alias to the BPMAuthor role.

WebSphere Education

Process server: Process Center connection failure (1 of 5)

- Check the Servers tab in the Process Center console
- If the status is “Status Disconnected”, or “Status Connected?”, then the Process Server is not connected online

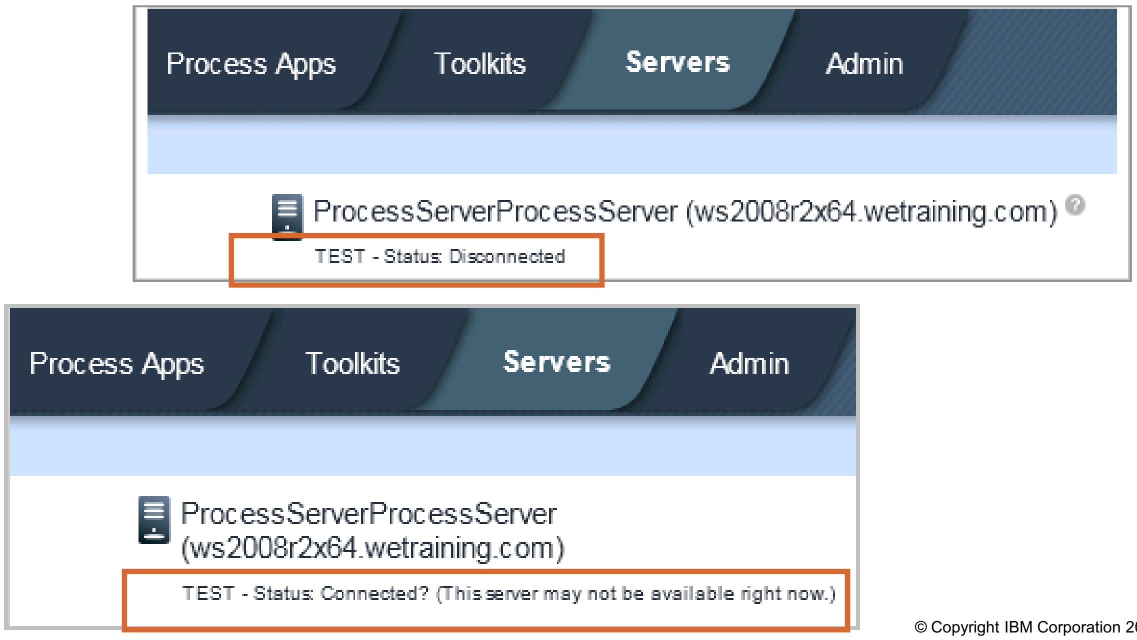


Figure 6-15. Process server: Process Center connection failure (1 of 5) WB8691.0

Notes:

There are many reasons why Process Server does not connect successfully to the Process Center.

WebSphere Education

Process server: Process Center connection failure (2 of 5)

- Check Process Server logs for errors when the Process Server does not successfully connect online
- An incorrectly mapped or missing authentication alias can be the cause

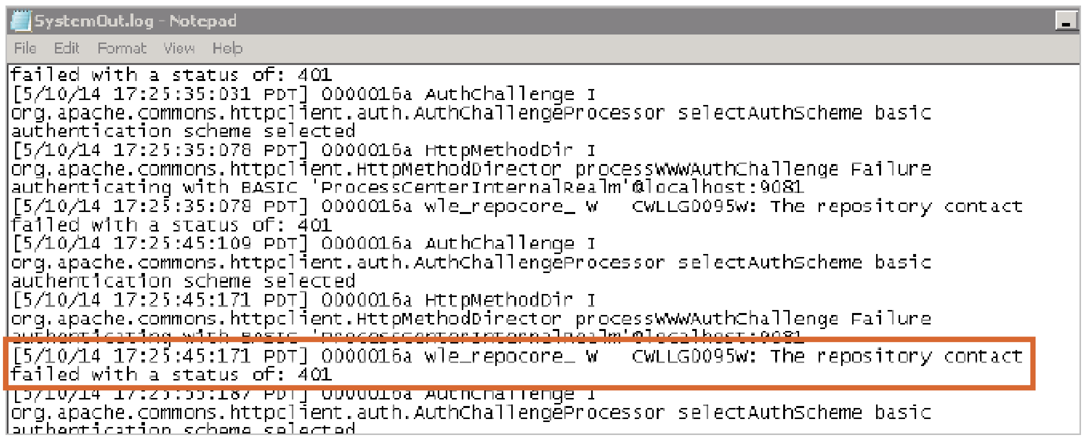


Figure 6-16. Process server: Process Center connection failure (2 of 5) WB8691.0

Notes:

There are many reasons why Process Server does not connect successfully to the Process Center.

WebSphere Education

Process server: Process Center connection failure (3 of 5)

- Check whether a valid user for BPMAuthorAlias exists in the LSW_SERVER table for the Process Center database

NAME	DESCRIPTION	ADDRESS	PROVIDER_URL	PORT
ProcessServerProcessServer	A running process server	http://ws2008r2x64.wetraini...	corbaname:ii...	9082

PORT	IS_OFFLINE	LAST_HEARTBEAT	IS_AVAILABLE	TYPE	USER_NAME
9082	F	2014-05-09 13:...	T	1	psdeadmin

Figure 6-17. Process server: Process Center connection failure (3 of 5) WB8691.0

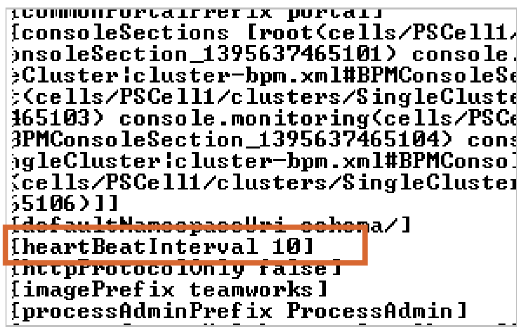
Notes:

For BPMAuthorAlias, use a valid user name and password from the Process Server environment that has the authority to access and deploy snapshots to the runtime Process Server. This user is saved in the LSW_SERVER table in the Process Center database and used when the process application is deployed to this Process Server.

Process server: Process Center connection failure (4 of 5)

- Check the value of the heartbeat interval
  - Value greater than zero good for connection
  - Do not set it too high
  - A negative one (-1) is a special value that indicates an offline server

```
wsadmin> AdminConfig.modify(ps, [['heartBeatInterval', '10']])
```



```
[consoleSections [root(cells/PSCell1/
onsoleSection_1395637465101) console.
eCluster|cluster-bpm.xml#BPMConsoleSe
:(cells/PSCell1/clusters/SingleCluste
465103) console.monitoring(cells/PSCe
3PMConsoleSection_1395637465104) cons
igleCluster|cluster-bpm.xml#BPMConso]
(cells/PSCell1/clusters/SingleCluste
65106)]]
[heartBeatInterval 10]
[imagePrefix teamworks]
[processAdminPrefix ProcessAdmin]
```


Figure 6-18. Process server: Process Center connection failure (4 of 5) WB8691.0

Notes:

To change the state of an offline Process Server to online, update the heartBeatInterval value to a number that is larger than 0 (zero). The heartbeat interval is the polling interval, in second, that the Process Server uses to communicate its location and characteristics to the Process Center. For example, to set the value to 60 seconds, enter the following command:

```
wsadmin> AdminConfig.modify(ps, [['heartBeatInterval', '60']]
```

WebSphere Education

Process server: Process Center connection failure (5 of 5)

- Verify that correct variable is used in the wsadmin command:
  - If working with V8.5.0 then use variable `processCenterUrl`

```
AdminConfig.modify(ps,
[['processCenterUrl','http://localhost:9081/ProcessCenter']])
```

  - If working with V8.5.0.1 then use variable `processCenterInternalUrl`

```
AdminConfig.modify(ps,
[['processCenterInternalUrl',
  'http://localhost:9081/ProcessCenterInternal']])
```

```
[imagePrefix teamworks]
[processAdminPrefix ProcessAdmin]
[processCenterInternalUrl http://localhost:9081/ProcessCenterInternal]
[processCenterUrl http://localhost:9081/ProcessCenter]
[processHelpWikiUrlEdit processhelp/en/Special:Edit?topic=%TITLE%&teamwo
=%TEAMWORKS_TITLE%]
[processHelpWikiUrlView processhelp/en/%TITLE%?teamworksTitle=%TEAMWORKS

[repositoryPrefix ProcessCenter]
[security (cells/PSCell1/clusters/SingleCluster|cluster-bpm.xml#BPMServe
y_1395637465101)]
```


Figure 6-19. Process server: Process Center connection failure (5 of 5) WB8691.0

Notes:

For V8.5.0:

Update the `processCenterUrl` variable:

`wsadmin> AdminConfig.modify(ps, [['processCenterUrl', 'https://`*new_server_name*`/ProcessCenter']])`

For V8.5.0.1:

If you are using IBM BPM V8.5.0.1, update the processCenterInternalUrl variable, which is the same as ProcessCenterUrl but with the literal string `Internal` appended at the end for the context root.

`wsadmin> AdminConfig.modify(ps, [['processCenterInternalUrl', 'https://`*new_server_name*`/ProcessCenterInternal']])`

6.3. Key database tables

WebSphere Education

Key database tables

Figure 6-20. Key database tables WB8691.0

Notes:

WebSphere Education

Why is database for IBM BPM important?

- Nearly all customer data and IBM BPM internal processing are stored in the database
- If you want to troubleshoot a problem that rarely occurs or happens on a high load production system, you might not be in a position to gain information with a trace
- The database information is always there, and you can gain insights that might be not achievable with other means
- The database information is just one part
  - Sometimes a trace can be more useful, depending on the problem scenario

Figure 6-21. Why is database for IBM BPM important? WB8691.0

Notes:

WebSphere Education IBM

Key IBM BPM databases

What databases exist?

- CMNDB
  - BPEL instances
- BPMDB
  - BPD instances
- PDWDB
- The CMNDB and BPMDB are the runtime databases, which store process instances information for BPD and BPEL
- The default configuration when creating the BPE container through the administrative console is CMNDB, but can be changed

Figure 6-22. Key IBM BPM databases WB8691.0

Notes:

WebSphere Education IBM

Some important databases for BPD instances

- Some of the essential database tables for BPD instances:
  - LSW_BPD_INSTANCE (process instances)
  - LSW_TASK (task instances)
  - LSW_SNAPSHOT (snapshot information)
  - LSW_USR_GRP_XREF (user information)

- There are over 200 tables, thus you focus here on only a small selection, which might be useful during normal business

Figure 6-23. Some important databases for BPD instances WB8691.0

Notes:

The LSW_TASK table is also in the PDWDB database, so make sure that you are in the right database when troubleshooting instances.

There are over 200 tables in the system, so use these four tables as a starting point for your investigation. Large volumes of data can become problematic and cause performance problems in your environment. There might be lot of completed tasks and process instances in the database tables, and that can slow down your system. Regular cleanup of completed process instances might be a good idea.

WebSphere Education

LSW_SNAPSHOT table (1 of 2)

- Two important kinds of information for BPD processes are stored in LSW_SNAPSHOT table
  - Snapshots
  - Runtime data
- Snapshots deal with what application design and version are deployed to the system
- Runtime data deals with process instances based on the deployed applications and associated artifacts
- This table can be essential to figure out whether a snapshot is active or not

Figure 6-24. LSW_SNAPSHOT table (1 of 2) WB8691.0

Notes:

There might be too many snapshots and versions in your system so the table LSW_SNAPSHOT comes in handy to investigate which ones are relevant.

WebSphere Education

LSW_SNAPSHOT table (2 of 2)

- Select * output of the LSW_SNAPSHOT table in csv format if needed

	SNAPSHOT_ID	REPOSITORY_BRANCH_ID	CREATED_BY_USER_ID	CRE...	NAME	ACRONYM
1	e985f805-7...	087236bc-693c-4bc4-8...	9	2014-03-2...	8.5...	8.5.0.1
2	09dd7f93-5...	34a0ce6e-631b-465d-b...	9	2014-03-2...	8.5...	8.5.0.1
3	3997ba3c-e...	6d3b84c0-bf15-4878-8...	9	2014-03-2...	Hiri...	HSAV850
4	43f509b2-6...	1a52abd6-b068-4f9e-9...	9	2014-03-2...	Sta...	SHSV850
5	949fba44-5...	087236bc-693c-4bc4-8...	9	2014-03-2...	NULL	949fba...
6	be3f820a-e...	34a0ce6e-631b-465d-b...	9	2014-03-2...	NULL	be3f82...
7	d9f6c031-0...	739cc732-eb97-4b1c-bf...	9	2014-03-2...	8.5...	8.5.0.1
8	814e15d7-5...	4961c615-3524-4196-9...	9	2014-03-2...	8.5...	8.5.0.1
9	5c679113-2...	f757be35-ebca-4e58-a...	9	2014-03-2...	8.5...	8.5.0.1
10	a66f737b-6...	7d491984-182d-45ce-a...	9	2014-03-2...	8.5...	8.5.0.1
11	3ad61f02-4...	50f91713-f1c4-4859-89...	9	2014-03-2...	8.5...	8.5.0.1
12	e9bd3c26-0...	NULL	9	2014-03-2...	NULL	e9bd3c...

Figure 6-25. LSW_SNAPSHOT table (2 of 2) WB8691.0

Notes:

The LSW_SNAPSHOT table is related to the LSW_PROJECT and LSW_BRANCH tables through the Project_ID and Branch_ID.

Is_Default is the default snapshot for a process. There should be exactly one per process in a Process Server environment, and it is not used in Process Center.

WebSphere Education IBM

Cleanup of completed process instances

- With time, the volume of a database system can increase by many completed process instances
  - SQL queries can track these instances, and that might cause performance problems
- What can be done to do a cleanup in Process Center?
  - For BPD instances there exists a stored procedure: LSW_BPD_INSTANCE_DELETE
  - This command can take several hours, depending on the environment
  - For version 8.5.0.1, use the automatic cleanup of unnamed snapshots by editing the `100custom.xml`

```
<unnamed-snapshots-cleanup-config>
    <enabled>true</enabled><cleanup-start-time>23:23:59<
    /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>
```

- For Process Server, use the BPMDeleteSnapshot wsadmin command to delete snapshots

Figure 6-26. Cleanup of completed process instances WB8691.0

Notes:

In version 8.5.0.1, you can configure IBM Process Center to automatically delete unnamed snapshots that you no longer need to keep on the server.

A named snapshot is a snapshot that has a version number or other identifier. Unnamed snapshots exist between the named snapshots. Every time that a Process Designer user saves work, an unnamed snapshot is created on the Process Center server. Hundreds of unnamed snapshots quickly accumulate. If you have many projects under development in the Process Center, the Process Center database is likely to grow rapidly. Reduce database bloat by removing the unnamed snapshots. You might also want to delete unnamed snapshots if your Process Center server performance is slowly degrading.

Keep in mind the following points while you plan how to use this feature:

- Automatic deletion never removes named snapshots.
- Because automatic deletion randomly chooses which process applications and toolkits to work on, specify a duration that is long enough to process all the active projects in your Process Center.

- Automatic deletion removes unnamed snapshots in chunks of 100 to limit database contention. If the duration time expires before all the unnamed snapshots are removed, automatic deletion might not remove all the unnamed snapshots between two named snapshots.
- Automatic deletion runs only when the server is up. If the server is down when the configured start time occurs, automatic deletion will not run until the next time the deletion feature is configured to start.

For more information on deleted snapshots on Process Center, go to `http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r5m0/index.jsp?topic=%2Fcom.ibm.wbpm.admin.doc%2Fmanaginglib%2Ftopic%2Fmanaging_snapshots_j.html`

For deleting snapshots on Process Server, use the `BPMDeleteSnapshot` command.

This command is run against a server instance to delete process application snapshots and their dependencies. For example, you might want to run the command if an excessive number of snapshots is causing your system to slow down.

To find more information about deleted snapshots on Process Server, go to:

`http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r5m0/topic/com.ibm.wbpm.ref.doc/topics/rref_deletesnapshot.html`

WebSphere Education

Some important databases for BPEL processes?

- There exist two kinds of table content relevant to BPEL:
  - Templates
  - Instance
- Templates are filled during the deployment process
  - It is a good place to check whether an application is deployed on the system or whether some artifacts are still in the system after application removal
- Instance tables include all instances of processes, tasks, and activities
  - Useful to see whether a missing entry really exists in the system

Figure 6-27. Some important databases for BPEL processes? WB8691.0

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.

WebSphere Education IBM

Some important databases for BPEL instances

- When you mention BPEL processes, think about business processes, human tasks, activities, and work items

- There exists an equivalent of this information in the database:
  - PROCESS_INSTANCE_B_T (process instances)
  - TASK_INSTANCE_T (task instances)
  - ACTIVITY_INSTANCES_B_T (activity instances)
  - WORK_ITEM_T (work items)
  - SCOPE_INSTANCE_B_T (scope instances)

Figure 6-28. Some important databases for BPEL instances WB8691.0

Notes:

WebSphere Education

Delete BPEL instances

- What can be done to reduce the size of the BPEL database?
- If you see a performance impact and there are many completed process instances:
  - Clean up the database with `deleteCompletedProcessInstances.py` administrative script
  - Archive with `Archive.py` administrative script
  - Process design by setting the property “Automatically delete on completion”

Figure 6-29. Delete BPEL instances WB8691.0

Notes:

Use the `deleteCompletedProcessInstances.py` administrative script to selectively delete from the Business Process Choreographer database or the Business Process Archive database any top-level BPEL process instances that reach an end state of finished, terminated, or failed.

The `archive.py` administrative script moves completed top-level BPEL process instances and human task instances, including their associated data, from a Business Process Choreographer database to an archive database.

Completed process instances are automatically deleted from the Business Process Choreographer database if the corresponding property is set for the process template in the process model.

BPM user and group tables

- LSW_USR: Store default user, BPM internally created user
- LSW_USR_XREF: All available user, historical user
- LSW_USR_GRP_XREF: Where LDAP group is; ad hoc group
- LSW_USR_GRP_MEM_XREF: Defines the relation between user and group
- LSW_GRP_GRP_MEM_XREF: Defines the hierarchy of group
- When does the group table get populated?
  - Startup
  - Manual sync

Figure 6-30. BPM user and group tables WB8691.0

Notes:

WebSphere Education

BPM security tables description (1 of 2)

Table - LSW_USR

Schema : BPMADMIN
Creator : BPMADMIN
Columns : 7

Actions:
- Open
- Query
- Show Related Objects
- Create New Table

Columns

Key	Name	Data type	Length	Nullable
Key	USER_ID	DECIMAL	12	No
	USER_NAME	VARCHAR	200	No
	PASSWD	VARCHAR	1000	No
	FULL_NAME	VARCHAR	200	No
	IS_DISABLED	CHARACTER	1	Yes
	LAST_LOGIN_DATETIME	TIMESTAMP	10	Yes
	OLD_PASSWORDS	CLOB	2147483647	Yes

Table - LSW_USR_XREF

Schema : BPMADMIN
Creator : BPMADMIN
Columns : 4

Actions:
- Open
- Query
- Show Related Objects

Columns

Key	Name	Data type	Length	Nullable
Key	USER_ID	DECIMAL	12	No
	USER_NAME	VARCHAR	256	No
	FULL_NAME	VARCHAR	1020	Yes
	PROVIDER	VARCHAR	512	Yes

Table - LSW_USR_GRP_XREF

Schema : BPMADMIN
Creator : BPMADMIN
Columns : 7

Actions:
- Open
- Query
- Show Related Objects
- Create New Table

Columns

Key	Name	Data type	Length	Nullable
Key	GROUP_ID	DECIMAL	12	No
	GROUP_NAME	VARCHAR	404	No
	DISPLAY_NAME	VARCHAR	512	No
	PARENT_GROUP_ID	DECIMAL	12	Yes
	GROUP_TYPE	DECIMAL	2	No
	DESCRIPTION	VARCHAR	1020	Yes
	GROUP_STATE	DECIMAL	2	No

Figure 6-31. BPM security tables description (1 of 2) WB8691.0

Notes:

WebSphere Education IBM

BPM security tables description (2 of 2)

Table - LSW_USR_GRP_MEM_XREF

Schema : BPMADMIN
Creator : BPMADMIN
Columns : 2

Actions:

Open

Columns

Key	Name	Data type	Length	Nullable
	USER_ID	DECIMAL	12	No
	GROUP_ID	DECIMAL	12	No

Table - LSW_GRP_GRP_MEM_XREF

Schema : BPMADMIN
Creator : BPMADMIN
Columns : 2

Actions:

Columns

Key	Name	Data type	Length	Nullable
	GROUP_ID	DECIMAL	12	No
	CONTAINER_GROUP_ID	DECIMAL	12	No

Figure 6-32. BPM security tables description (2 of 2) WB8691.0

Notes:

The user and group information is stored in the database tables. The XREF tables replicate user and group information from LDAP as well.

WebSphere Education

Snapshot installation tables

- These three tables hold the information that is needed to uniquely identify a snapshot and its process app or toolkit
  - LSW_SNAPSHOT
  - LSW_BRANCH
  - LSW_PROJECT
- LSW_INSTALLATION (Process Server only)
  - Holds the status of installation attempts to a Process Server
- LSW_DEPLOYMENT (Process Center only)
  - Holds snapshot and server information for items that the Process Center deploys
- LSW_DEPLOYMENT_PACKAGE (Process Center only)
  - Holds deployment package exports for offline deployment
- LSW_SERVER (Process Center only)
  - Holds information for online and offline servers that are defined in the Process Center

Figure 6-33. Snapshot installation tables WB8691.0

Notes:

WebSphere Education

LSW_INSTALLATION table

- LSW_INSTALLAION is significant on the process server and indicates the installation status of snapshots

LSW_INSTALLATION

INSTALLATION_ID	SNAPSHOT_ID	STATUS	DETAILS
838a0137-58c0-4a06-bed7-05b0cd8ef288	1e519b1a-c1e5-425c-a34f-2db93e35a260	10	COM.ibm.db2.jdbc.app.DB2Blob@2a0c2a0c
24d197fa-2752-4409-bb55-c20545e9fe53	e13f681c-8ae8-4c74-beca-8e4245d1c9ed	10	COM.ibm.db2.jdbc.app.DB2Blob@2a6f2a6f
e8f1bfea-e6f8-4087-a92a-db44d329f633	12933784-1e9d-4862-8ac3-f7dbb6a2aad7	10	COM.ibm.db2.jdbc.app.DB2Blob@2abf2abf
6e200d81-0541-49ed-9fd6-3177ea8414b1	e372e7f0-95b1-4416-9876-c0f079c647a5	10	COM.ibm.db2.jdbc.app.DB2Blob@2b0f2b0f
4aa7523b-90d6-41ca-bde3-f305a366f456	202df124-ce51-4130-8eb5-e92b5f596d43	10	COM.ibm.db2.jdbc.app.DB2Blob@2b5f2b5f

Figure 6-34. LSW_INSTALLATION table WB8691.0

Notes:

Status: 0 = in progress, 4 = failed, 10 = successful

Status is reset to 0 on an installation attempt where the snapshot does not exist.

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Describe BPMConfig content
- Describe troubleshooting options when working with online Process Server
- Describe some key database tables for troubleshooting

Figure 6-35. Unit summary WB8691.0

Notes:

WebSphere Education IBM

Checkpoint questions

1. When working with BPMconfig, what are the available values for the `bpm.de.environment` property?
2. True or False. A negative one (-1) is a special value that indicates an offline server.
3. True or False. LSW_SERVER table in the Process Server holds information for online and offline servers that are defined in the Process Center.

Figure 6-36. Checkpoint questions WB8691.0

Notes:

Write down your answers here:

1.

2.

3.

WebSphere Education

Checkpoint answers

1. Process Server or Process Center
2. True
3. False. The LSW_Server table in **Process Center** (not Process Server) holds information for online and offline servers that are defined in the Process Center.

Figure 6-37. Checkpoint answers WB8691.0

Notes:

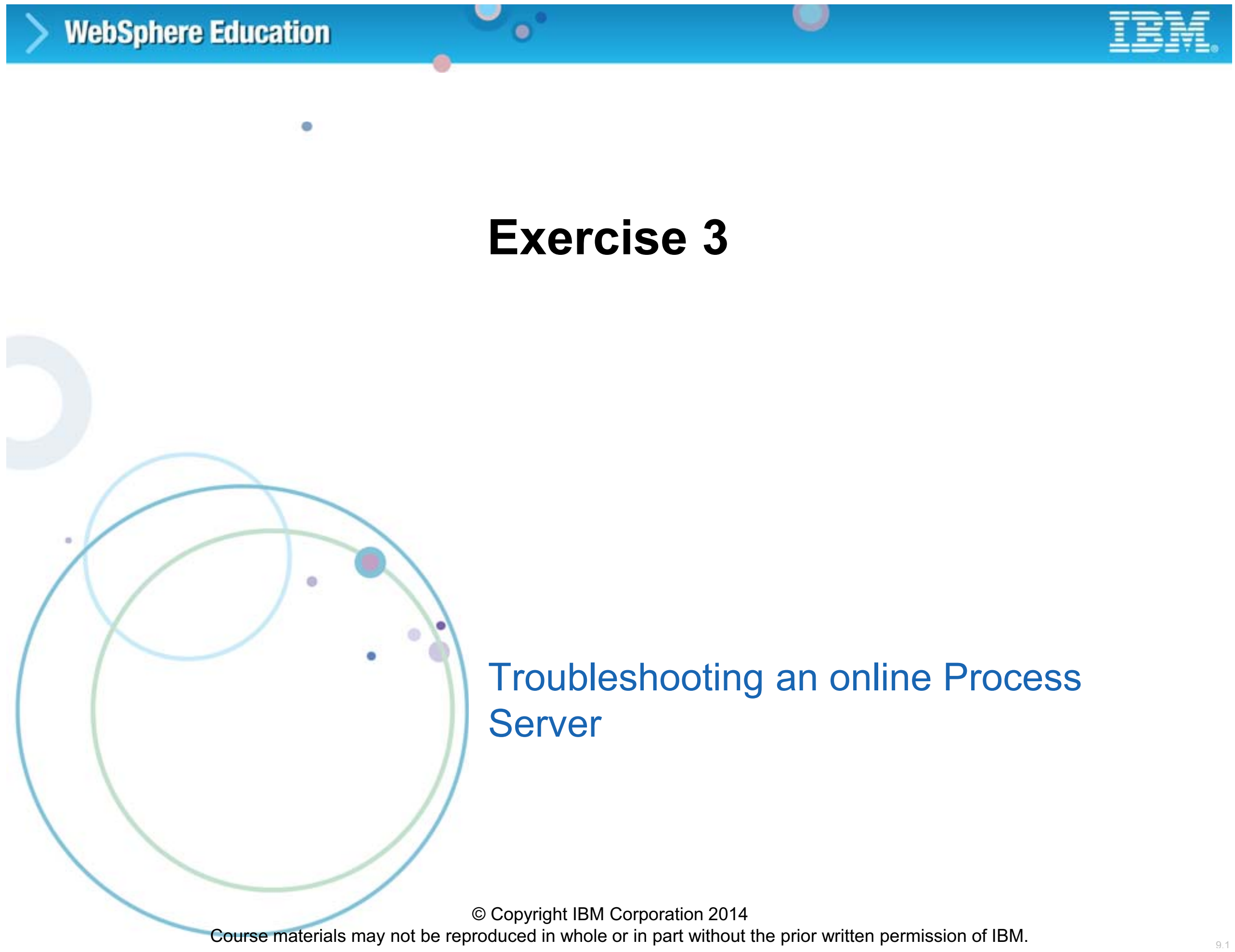


Figure 6-38. Exercise 3 WB8691.0

Notes:

WebSphere Education IBM

Exercise objectives

After completing this exercise, you should be able to:

- Examine the online Process Server connection status in Process Center
- Review the system logs for errors
- Run wsadmin script commands for troubleshooting
- Verify the authentication alias that is used for Process Server and Process Center communication

Figure 6-39. Exercise objectives WB8691.0

Notes:

Unit 7. Troubleshooting process application deployment

What this unit is about

This unit describes troubleshooting and problem prevention techniques for process application deployment.

What you should be able to do

After completing this unit, you should be able to:

- Explain basic process application concepts
- Describe process application management
- Describe user permissions issues with process applications
- Explain troubleshooting actions for process applications

How you will check your progress

- Checkpoint

References

Installing, deploying, and undeploying applications in the runtime environment:
`http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/com.ibm.wbpm.admin.doc/topics/install_runtime.html`

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Explain basic process application concepts
- Describe process application management
- Describe user permissions issues with process applications
- Explain troubleshooting actions for process applications

Figure 7-1. Unit objectives WB8691.0

Notes:

WebSphere Education

Topics

- Process application overview
- Process application management
- User permissions
- Troubleshooting actions for process applications

Figure 7-2. Topics

WB8691.0

Notes:

7.1. Process applications overview

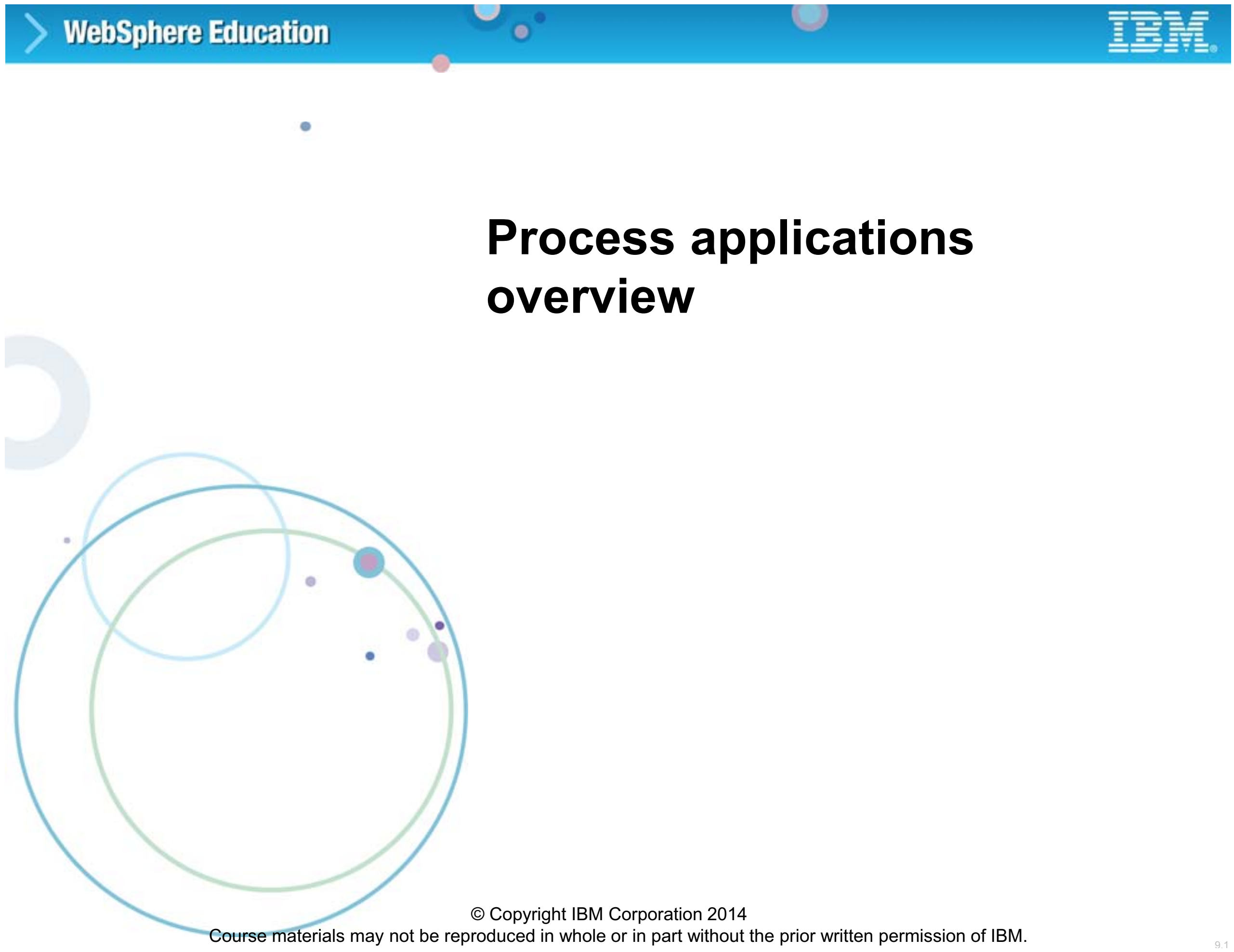


Figure 7-3. Process applications overview WB8691.0

Notes:

WebSphere Education

Process applications

- Process application is a container for process models and their supporting implementations

- Process applications contain the following artifacts:
  - One or more process models, also called Business Process Definitions (BPDs)
  - References to toolkits
  - Services that are required to implement activities or integrate with other systems
  - One or more tracks
  - Service Component Architecture (SCA) modules and libraries
  - Any other items that are required to run the process

- After the relevant artifacts are created, they are assembled into a process application
  - Process applications are stored in the Process Center Repository
  - Access process applications through the Process Center console

Figure 7-4. Process applications WB8691.0

Notes:

WebSphere Education IBM

Process application track, snapshot, and tip

- Track
  - Similar to a branch, is an optional subdivision in the process application
  - By default, each process application has a single track, which is called Main
  - Tracks keep changes isolated; multiple tracks mean that multiple versions of the process application can be edited
- Snapshot
  - Deployable version of a process application
  - Captures state of the process application at the time it is taken
  - Snapshots cannot be edited
- Tip
  - Specialized snapshot that is available on Process Center
  - Current working version of the process application
  - Process application remains at that tip level until you create a snapshot
  - Each track has one tip that is associated with it

Figure 7-5. Process application track, snapshot, and tip WB8691.0

Notes:

Changes to a process application are dynamically saved to the Process Center repository at the tip. The tip is the current working version of the process application.

- You can change the contents of the tip, but it runs only on the Process Center server.
- You cannot install a tip on a process server.

Snapshots record the state of library items within a process application or track at a specific point in time. You typically take a snapshot when:

- You are ready to test the integration.
- You want to install the process application on a Process Center server or a process server for development, testing, staging, or production.

WebSphere Education

Process Center console: Process Apps view

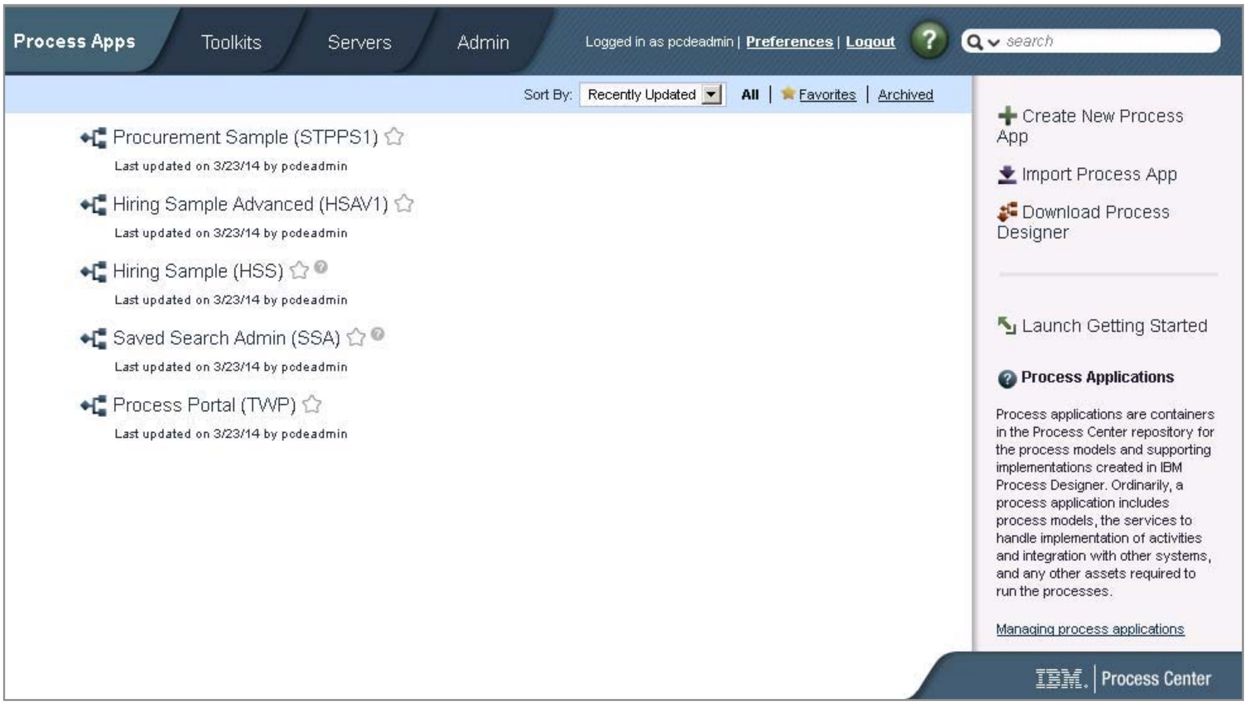


Figure 7-6. Process Center console: Process Apps view WB8691.0

Notes:

The Process Center console provides the tools that you need to maintain the repository. You can use the Process Center console to create, clone, import process applications, and do other maintenance tasks on the process applications.

The **Process Apps** tab lists the process applications that the user can access.

WebSphere Education

Process Center console: Sample process application

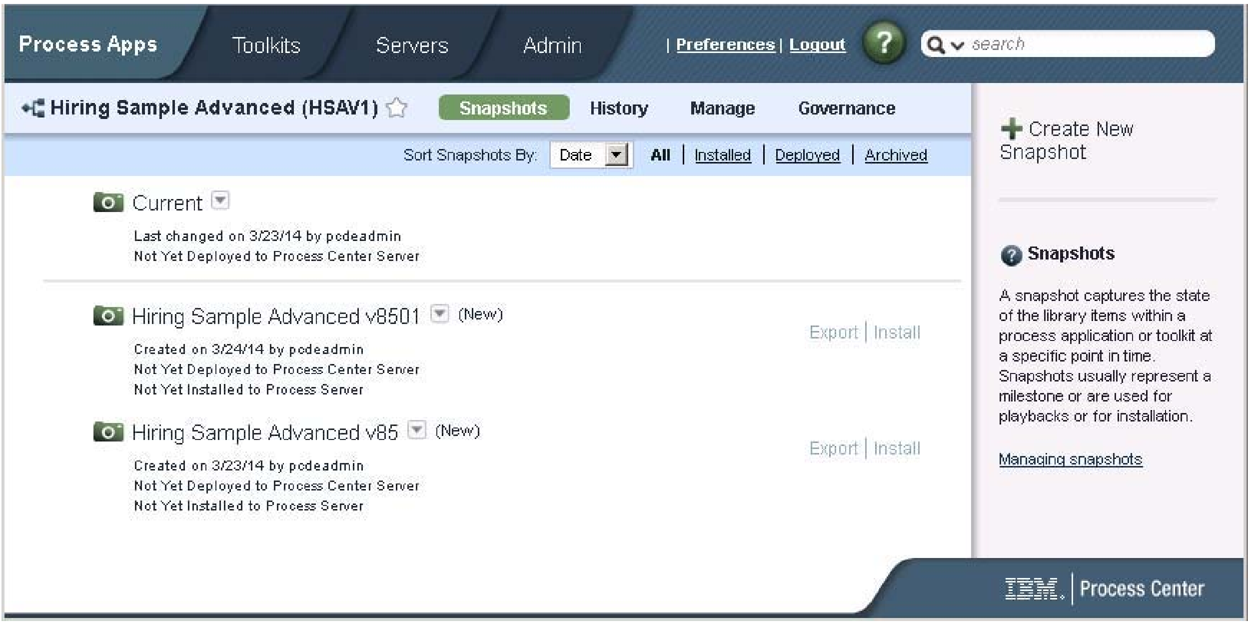


Figure 7-7. Process Center console: Sample process application WB8691.0

Notes:

You can do the following tasks for a process application:

- Create snapshots
- Activate process application snapshots in Process Center
- Install process application snapshots on a process server
- Export process application snapshots
- Archive process application snapshots

Installing process application snapshots to an online process server

Before you install a snapshot on a connected process server, complete the following tasks:

- Create a snapshot of the process application to be installed.
- Ensure that the first 3 digits of the Process Center version match the Process Server version.
- Ensure that the snapshot can be activated successfully in Process Center.

Installing process application snapshots to an offline process server

- Create a snapshot of the process application to be installed.
- Add an offline server, making sure that its version is identical to the version of your Process Center server.
- Before you install a snapshot on a Process Server, make sure that the snapshot can be activated successfully on a Process Center. A snapshot that cannot be activated on Process Center usually fails to activate on a Process Server.
- Ensure that the target Process Server supports the capability that the snapshot uses.

7.2. Process application management

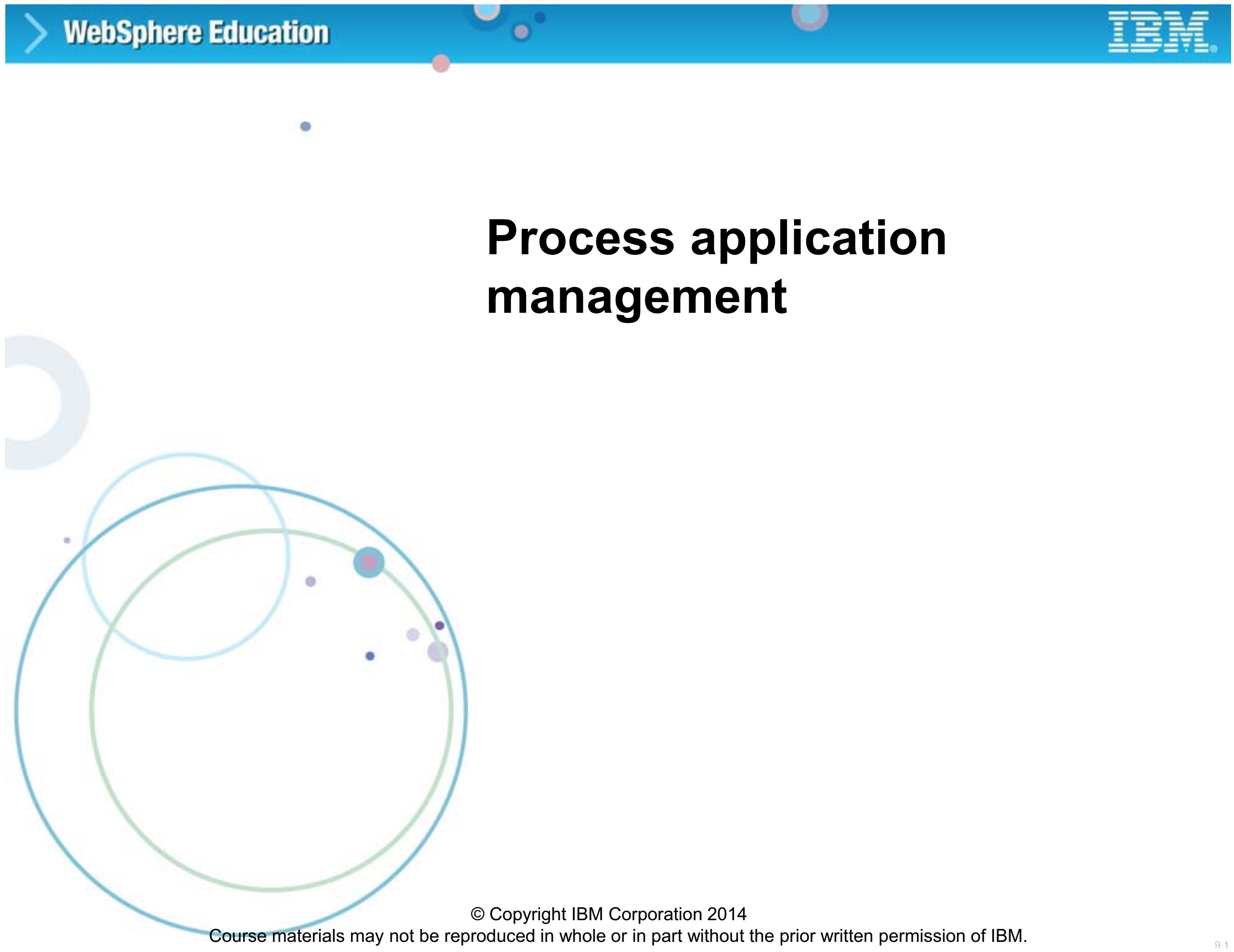


Figure 7-8. Process application management WB8691.0

Notes:

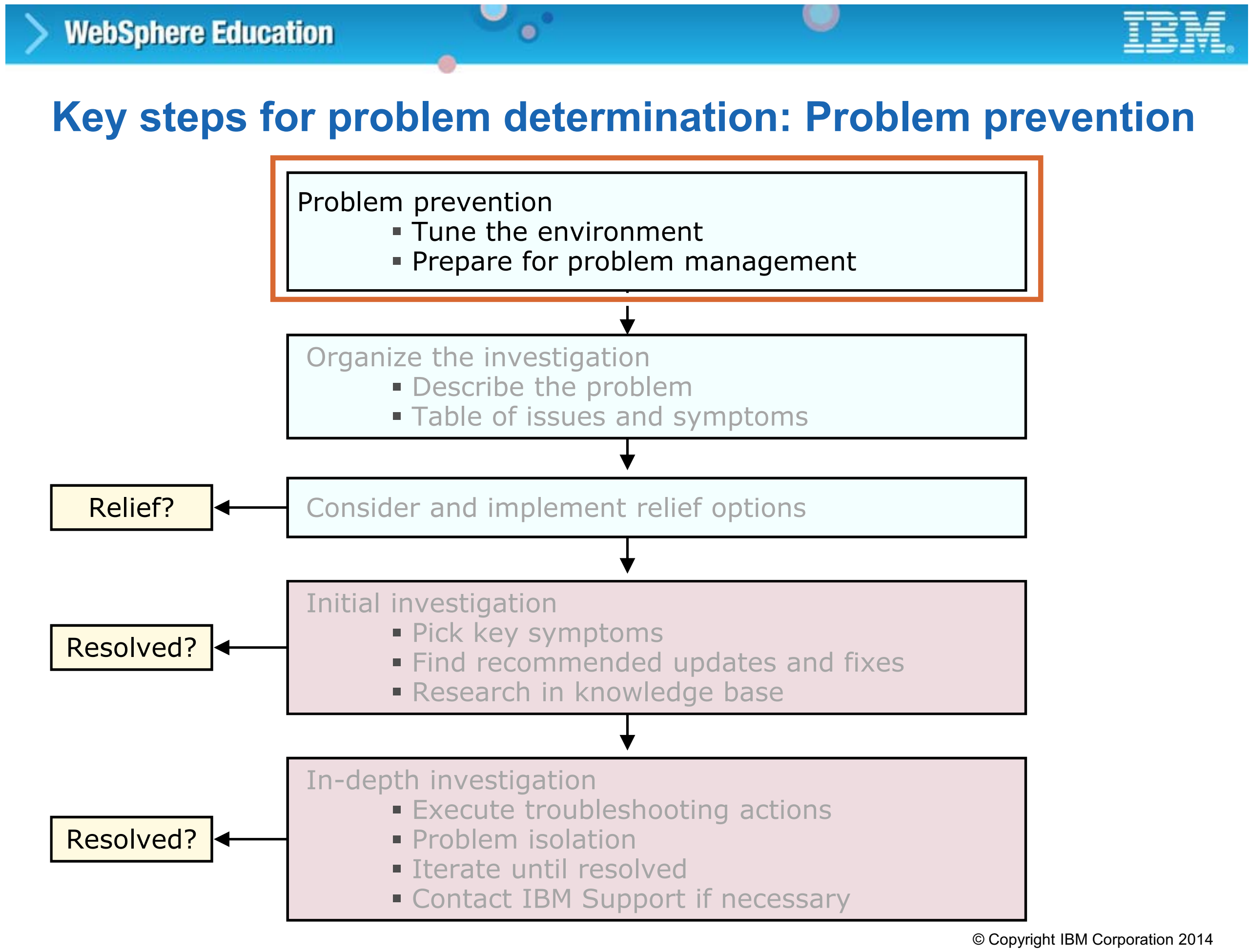


Figure 7-9. Key steps for problem determination: Problem prevention WB8691.0

Notes:

WebSphere Education

Process application management and problem prevention

- Being aware of process application issues can help prevent issues with process application installation
- Issues include:
  - Process application and snapshot names
  - Installing process application snapshots
  - Deleting unused or unneeded process application snapshots

Figure 7-10. Process application management and problem prevention WB8691.0

Notes:

WebSphere Education

Process application and snapshot names

- When you create a process application, you give it a name and an acronym
  - The acronym for a process application must be unique and is limited to 7 characters
  - IBM Business Process Manager uses the acronym as an identifier for this process application and the library items that it contains
- When you create a snapshot, it is associated with a specific track or branch
  - Snapshot names should be unique across tracks
  - Unlike process applications, IBM Business Process Manager does not require unique snapshot names
  - However, you cannot install a snapshot of a process application in a runtime environment if a snapshot with the same name is already installed
  - This restriction also applies for snapshots in different tracks

Figure 7-11. Process application and snapshot names WB8691.0

Notes:

The process application acronym is created when the process application is created. It can be a maximum of 7 characters in length.

The snapshot acronym is created automatically when the snapshot is created. It can be a maximum of 7 characters in length. If the snapshot name meets the criteria for a valid snapshot acronym, the snapshot name and acronym are the same.

WebSphere Education IBM.

Installing process application snapshots

- When you install a process application snapshot to a process server:
  - Library items for that snapshot, including toolkit dependencies, are moved from the repository to the selected process server
  - The process server can be connected or offline
  - You can use the Process Center console or wsadmin commands to install the snapshot
- Important: The Process Center server and the connected or offline process server versions should match exactly **before** you install to the process server
  - Version matching is applicable to the first 3 digits only
  - For example: 8.5.0 or 8.5.1

Figure 7-12. Installing process application snapshots WB8691.0

Notes:

Understanding the individual steps in the installation process can help you identify and resolve any errors that occur during snapshot installation.

The following actions occur on the target server during the process application installation process.

1. Install the necessary library items and assets for the process application and referenced toolkits. The installation process deploys only those referenced toolkits that are not already on the target server. Default values for environment variables and exposed process values (EPVs) are set, and other design-time versioned assets (such as portal searches) are created.
2. Run the installation service for each toolkit. The installation service for each referenced toolkit must be started before the installation service for the referring toolkit.
3. Run the installation service for the process application. The installation service for the process application is the final installation service that is started.
4. Migrate data and process instances if there are running business process definition instances. The specific actions of this step depend on the migration option that you choose. Migrating process instances is not a fully automated process. Migration is handled according to these general principles:

- Consider the business process definition and its variables as the interface, and process instances as the realization of that interface.
- As part of instance migration, completed tasks are migrated into the current process version. It is important that the process can resolve a completed execution context to preserve historical information.
- The new process version must be designed to provide compatibility to earlier versions of instances that you want to migrate.
- If you removed a task, it is sometimes possible to account for the difference by moving the resulting orphaned token, although there are limitations to this capability.

5. Send tracking definitions to the Performance Data Warehouse. The process server updates the Performance Data Warehouse with any new or changed tracking definitions.
6. Activate scheduled undercover agents (UCAs).
7. Deploy advanced content. If the snapshot has advanced content, then the advanced artifacts, such as SCA modules and libraries, are deployed to the process server.
8. On a connected process server, send a message that says that the installation is complete. The user who initiated the installation can see the completion message in the Process Center Console.

WebSphere Education IBM

Deleting unused snapshots

- Over time, as process applications change, the number of snapshots can grow
  – Can take up space in the Process Center database
- In many situations, it is a good idea to purge unneeded or unnamed snapshots on a regular schedule
  – Unnamed snapshots are created every time a Process Designer user saves work
  – You can also accumulate named snapshots that were once active, but are no longer used
- Can use automated or manual methods to delete unnamed snapshots
  – Automated: Configure Process Center to automatically delete unnamed snapshots that you no longer need by changing the `100Custom.xml` file
  – Manual: Use the `BPMSnapshotCleanup` or `BPMDeleteSnapshot` commands in `wsadmin` to manually delete snapshots

Figure 7-13. Deleting unused snapshots WB8691.0

Notes:

WebSphere Education IBM

Deleting unused snapshots automatically

- Configure settings in the `100Custom.xml` file to enable or disable automatic deletion of unnamed snapshots
  - `100Custom.xml` is stored in the profile root for the server
  - Find `<server>` section, then `<unnamed-snapshots-cleanup-config>`
  - Set `<enabled>` property to `true`
  - Configure other settings as needed

```
<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 7-14. Deleting unused snapshots automatically WB8691.0

Notes:

Automated snapshot deletion

Keep in mind the following points while you plan how to use this feature:

- Automatic deletion never removes named snapshots.
- Automatic deletion randomly chooses which process applications and toolkits to work on, so specify a duration that is long enough to process all the active projects in your Process Center.
- Automatic deletion removes unnamed snapshots in chunks of 100 to limit database contention. If the duration time expires before all the unnamed snapshots are removed, automatic deletion might not remove all the unnamed snapshots between two named snapshots.
- Automatic deletion runs only when the server is running. If the server is down when the configured start time occurs, automatic deletion will not run until the next time the deletion feature is configured to start.

The feature to automatically delete unnamed snapshots is controlled with a set of configuration options in the `100Custom.xml` file.

1. Go to the server profile root directory to open the `100Custom.xml` file.

2. Find the `<server>` section of the `100Custom.xml` file. The section for automatic snapshot deletion begins with the heading `<unnamed-snapshots-cleanup-config>`.
3. To turn on the automated deletion feature, set `<enabled> to true`. The default setting is `false`.
4. To set the time of day when you want the automated snapshot deletion to run, provide a value for `<cleanup-start-time>`. The time that is used is the local computer time. The default time is midnight: `23:59:59`. The automatic snapshot deletion process makes intensive use of the database; therefore, run the process when other demands on the system are low. Running it during times of heavy use slows the response time for people who use the system.
5. Set `<cleanup-duration-minutes>` to the number of minutes that you want the process to run. The default duration is **5** minutes.
6. To define which snapshots you want to delete, set `<clean-after-number-named-snapshots>`. Deleting snapshots removes some of the change history for your project. Because of these changes, you might want to keep the most recent snapshots. 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

Deleting unused snapshots manually

- Important: Before running the `BPMDeleteSnapshot` and `BPMSnapshotCleanup` commands, see the following alert and install all interim fixes for your release:
  `http://www.ibm.com/support/docview.wss?uid=swg21669992`
  – “Issues with BPMDeleteSnapshot and BPMSnapshotCleanup commands in IBM Business Process Manager”
  – Interim fixes address potential issues with incomplete artifact deletion
- **`BPMDeleteSnapshots`**
  – Delete snapshots and their dependencies for a specific process application
- **`BPMSnapshotCleanup`**
  – Delete all unnamed and archived snapshots of a process application on a Process Center Server

Figure 7-15. Deleting unused snapshots manually WB8691.0

Notes:

BPMSnapshotDelete

If you want to use **BPMDeleteSnapshot**, the following conditions apply.

- You must have repository administrative privileges.
- The snapshot cannot have any running instances, and it cannot be the default snapshot. Use `BPMShowSnapshot` to determine whether these conditions are true.
- The snapshot cannot be active.
- The snapshot cannot be deployed.
- The snapshot can have no running BPEL instances.
- Business Process Manager Advanced content must be undeployed with the `BPMUndeploy` command.

When you successfully delete a snapshot, notice that any business process definition (BPD) instances for it are deleted with it.

Here is the procedure for running the `wsadmin BPMDeleteSnapshot` command for IBM Business Process Manager Advanced. For a list of parameters, see this web address:

`http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/com.ibm.wbpm.ref.doc/topics/rref_deletesnapshot.html`

1. Run the `BPMShowProcessApplication` command to determine whether the snapshot exists for the process application.
2. Run `BPMShowSnapshot` to determine the status of the snapshot, including whether it is the default snapshot and whether it is active with running instances.
3. Run `BPMDeactivate` to deactivate the snapshot.
4. Run the `BPMStop` command to stop the snapshot and its running instances.
5. Run `BPMUndeploy` to undeploy the snapshot from the server. This command also uninstalls any business-level applications that are related to the snapshot.
6. Run `BPMDeleteSnapshot` to delete the process application snapshot and its dependencies from the server.

BPMSnapshotCleanup

If you want to use `BPMSnapshotCleanup`, the following conditions apply.

- You must be a repository administrator to run this command.
- You cannot delete the first snapshot of a process application, even when it is unnamed or archived. The first snapshot contains original information about the snapshot that is displayed in the history pane in Process Designer.
- You must archive named snapshots before you delete them.
- To avoid conflicts between operations and snapshots on the Process Center, you must run the `BPMSnapshotCleanup` command at specified intervals. Run `BPMSnapshotCleanup` when no operations are running on the Process Center and no connections are open between Process Designer and Process Center.

Here is the procedure for running the `BPMSnapshotCleanup` command on `wsadmin`. For a list of parameters for this command, see the following web address:

`http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/com.ibm.wbpm.ref.doc/topics/rref_bpmsnapshotcleanup.html`

1. Run the `wsadmin BPMListProcessApplication` command on the Process Center server to show all process application snapshots on that server.
2. Archive named snapshots if they are no longer needed and you want to delete them. You can delete named snapshots from the Process Center server only if they are archived. Run `BPMShowSnapshot` to see information about a specific snapshot.
3. Run the `wsadmin BPMShowProcessApplication` command to show details about the process application, including the process application acronym. You need the acronym to run the `BPMSnapshotCleanup` command.
4. Set the `containerAcronym` parameter to identify the process application that contains the snapshots to be deleted.

5. Set optional parameters. You must set at least one optional parameter as a filter for determining which unnamed snapshots are deleted. You can also use one of the optional parameters, `deleteArchivedSnapshot`, to delete archived snapshots in addition to unnamed snapshots.
6. Run the `BPMSnapshotCleanup` command to delete all unnamed snapshots that fit within the parameters that you defined. The command does not delete the first snapshot of each branch, even if it is unnamed or archived. The first snapshot is needed because it contains the information that shows in the Revision History window in Process Designer.

7.3. User permissions

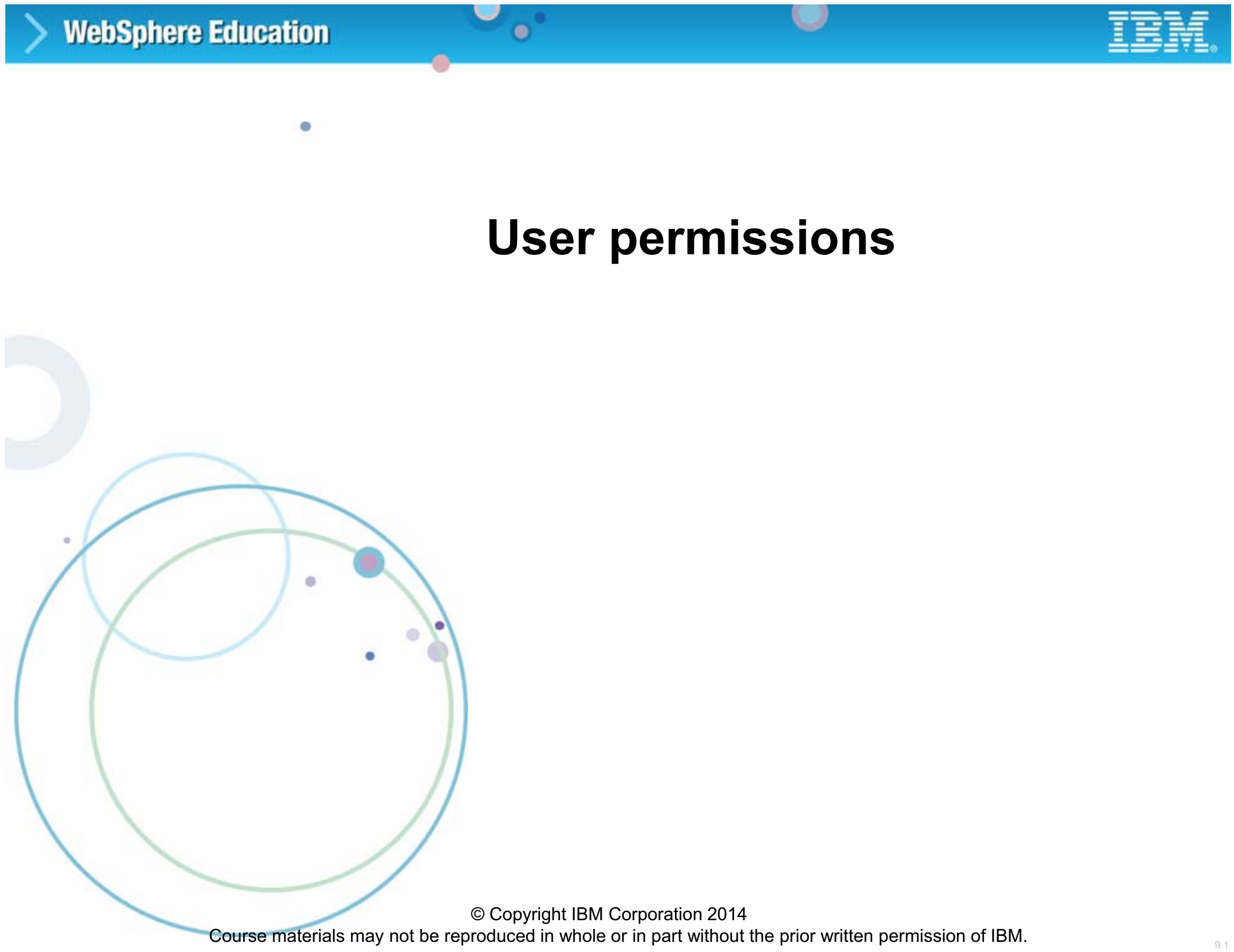


Figure 7-16. User permissions WB8691.0

Notes:

WebSphere Education

Key steps for problem determination: Problem prevention

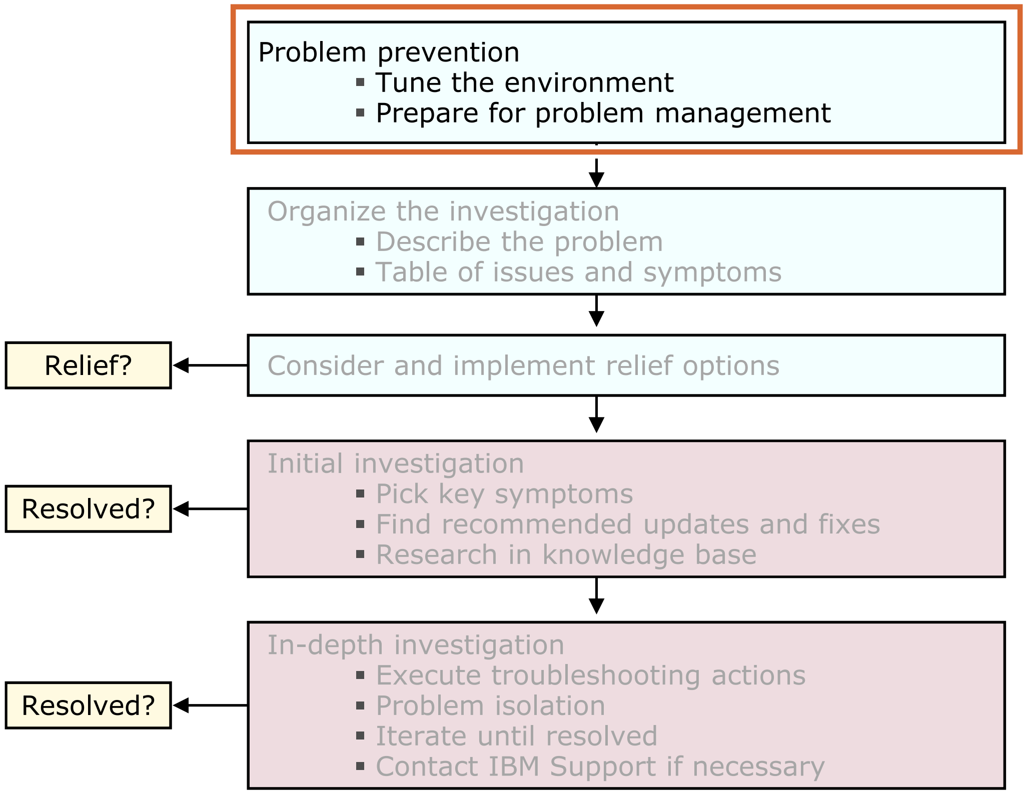


Figure 7-17. Key steps for problem determination: Problem prevention WB8691.0

Notes:

WebSphere Education

Process applications and user permissions

- User permissions can affect whether a user can access or install a process application
- Only certain roles have the authority to install or deploy process applications on the runtime server
- Make sure that the user who does the installation or deployment operation is assigned to the appropriate administrative security role

Figure 7-18. Process applications and user permissions WB8691.0

Notes:

Map a group or groups, rather than specific users, to administrative roles. Configuring group permissions is more flexible and easier to administer than working with individual user permissions. When you map a group to an administrative role, adding or removing users to or from the group occurs outside of IBM Business Process Manager. Thus, you do not need to restart the server for the changes to take effect.

WebSphere Education

Groups, users, and permissions

- Use the Process Admin console to:
  - Create users
  - Create groups
  - Add users or groups to groups
- IBM Business Process Manager has several default administrative user groups
  - Each group has different permissions that are associated with it
  - Example default groups: `tw_admins`, `tw_authors`
- By default, anyone in the `tw_admins` group can install a process application on either an online or an offline Process Server

Figure 7-19. Groups, users, and permissions WB8691.0

Notes:

Here is a complete list of the default user groups for Business Process Manager.

- **tw_admins**: Members of this group have full access to all interfaces, assets, servers, and security. Note: You can rename this group, but there must always be an administrator group defined. Administration of IBM BPM is not possible without this group.
- **tw_allusers**: This group is the default lane assignment for non-system lanes when creating business process definitions (BPDs) in the Process Designer. The dashboards that you create in Process Designer are available to this group by default.
- **tw_allusers_managers**: This group contains the team of managers for the **tw_allusers** group. By default, the **tw_allusers_managers** group includes the **tw_admins** group.
- **tw_authors**: Members of this group have access to the Designer and other interfaces in the Process Designer, including the Process Center console. From the Process Center console, members of this group can create process applications and toolkits and control access to projects. Process Center repository administrators control access to other process applications and toolkits (projects) and the assets they contain.

- **Debug**: You can use this account to restrict access to service debugging in the Inspector in the Process Designer.
- **tw_eventmanager**: Members of this group have full access to historical information about event manager processing.
- **tw_managers**: Members of this group can see the Team Performance dashboard in Process Portal. By default, the **tw_managers** group includes the **tw_allusers** group.
- **tw_portal_admins**: Because of changes in IBM BPM V8, members of this group no longer have any special access rights.
- **tw_process_owners**: Members of this group can see the Process Performance dashboard. By default, the **tw_process_owners** group includes the **tw_admins** group.

WebSphere Education

Process Admin console: User Management

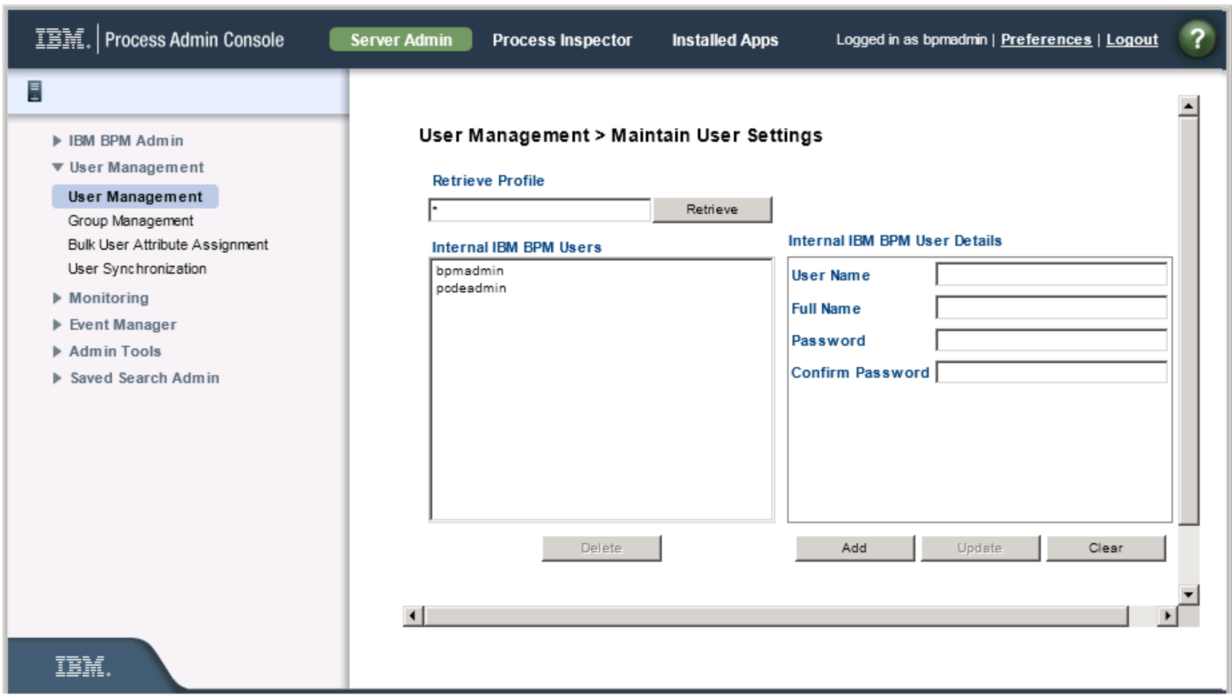


Figure 7-20. Process Admin console: User Management WB8691.0

Notes:

In the Process Admin console, go to **User Management > User Management**.

Use the User Management window to define and manage Process Center users.

To retrieve the list of all users, enter an asterisk (*) in the **Retrieve Profile** field, and click **Retrieve**.

WebSphere Education

Process Admin console: Group Management

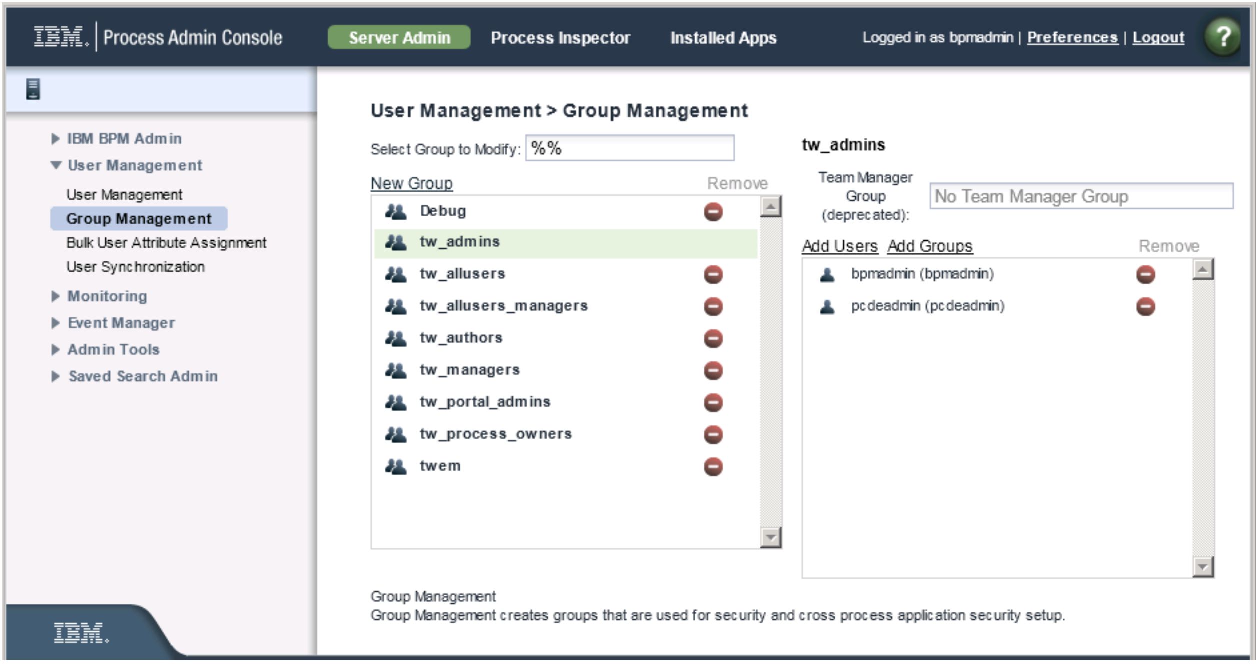


Figure 7-21. Process Admin console: Group Management WB8691.0

Notes:

In the Process Admin console, go to **User Management > Group Management**.

Use the Group Management window to:

- Create a group
- Add users to groups
- Add groups to groups

To retrieve a list of all groups, enter `%%` in the **Select Group to Modify** field.

WebSphere Education

Access to process application installation (1 of 2)

By default, the following access levels are required to install a process application, depending on the Process Server environment

- Production environments: Requires administrative access to Process Servers
- Non-production environments: Requires write or administrative access to Process Servers
- Development environments: Requires read, write, or administrative access to Process Servers

Figure 7-22. Access to process application installation (1 of 2) WB8691.0

Notes:

WebSphere Education IBM

Access to process application installation (2 of 2)

- Installing IBM Business Process Manager Advanced content with a process application snapshot
  - User or group must be assigned the Configurator, Operator, *and* Deployer administrative security roles
  - Assign these roles to users or groups by using the WebSphere Application Server administrative console
- To further restrict installation access for process applications
  - Can define two more subgroups for **`tw_admins`** users through wsadmin: `processCenterInstall` and `offlineInstall` for offline process servers
  - Must belong to subgroup and `tw_admins` to install process applications

Figure 7-23. Access to process application installation (2 of 2) WB8691.0

Notes:

To define `processCenterInstall` and `offlineInstall`:

- Use the wsadmin command-line tool to define the groups
  - If the `processCenterInstall` or `offlineInstall` subgroup is defined, the user must be a member of both **tw_admins** and the subgroup to install a process application.
  - For more information, see the IBM Knowledge Center documentation for Business Process Manager V8.5.

7.4. Troubleshooting actions for process applications

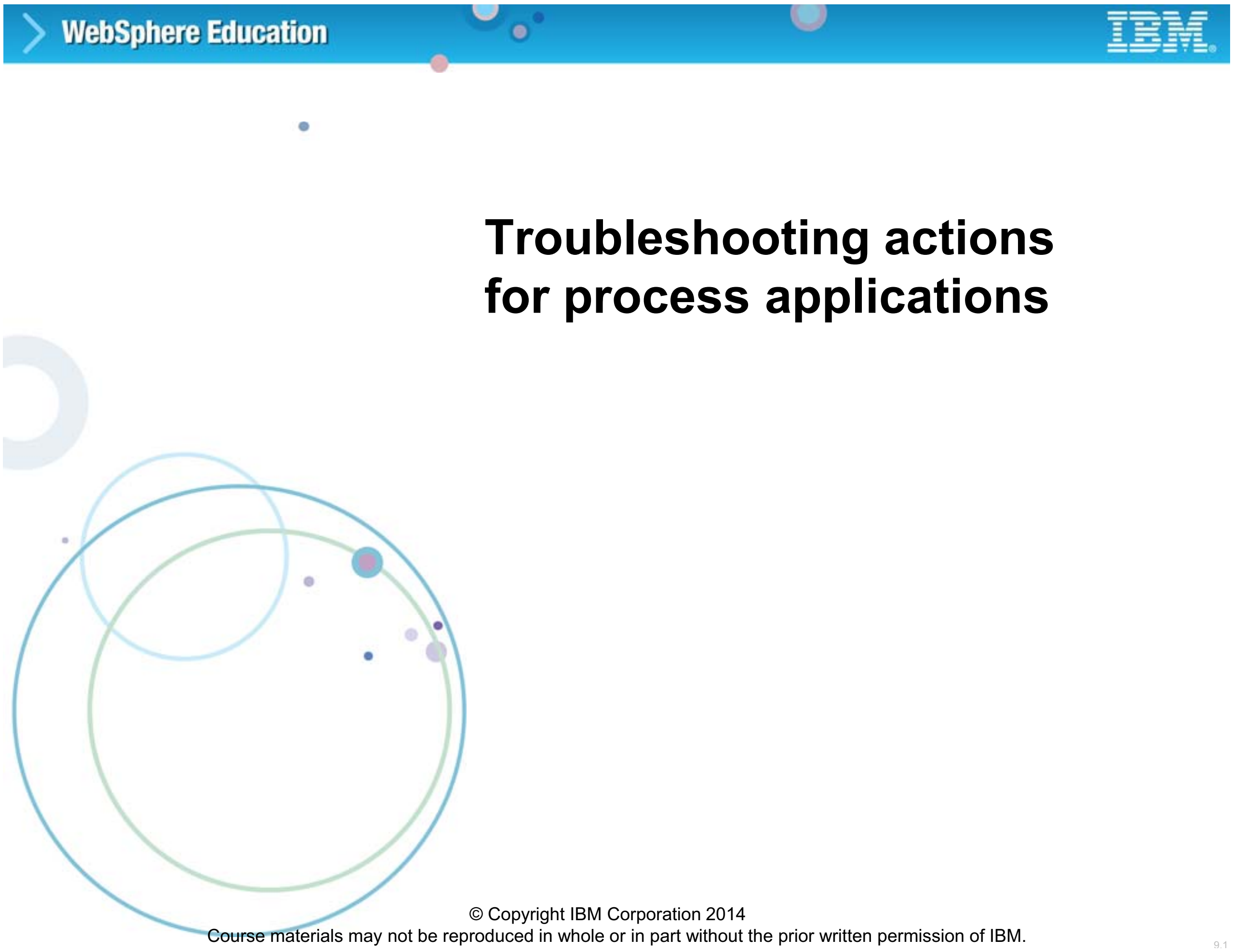


Figure 7-24. Troubleshooting actions for process applications WB8691.0

Notes:

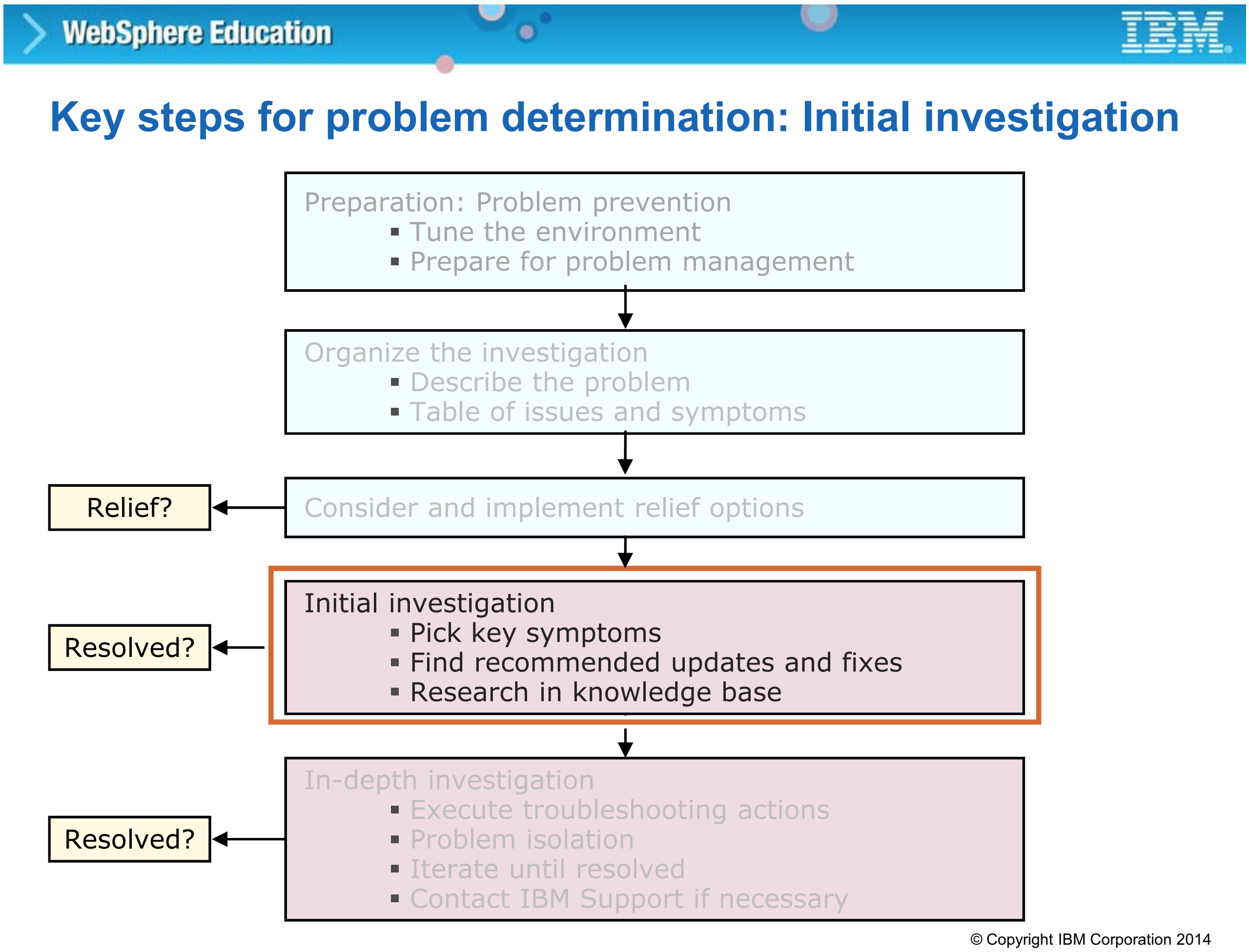


Figure 7-25. Key steps for problem determination: Initial investigation WB8691.0

Notes:

WebSphere Education IBM

Check server logs

- Check various server logs for information about process application issues
  - Log files can be found in the `logs` directory in the server profile directory
- Review the log files and traces at the time of the problem to try to determine the source of the problem
  - Make a note of when the problem occurred
- Logs to review:
  - `SystemOut.log`
  - `SystemErr.log`
  - FFDC logs
  - Trace logs

Figure 7-26. Check server logs WB8691.0

Notes:

If you have problems when installing a snapshot, always check the `SystemOut.log` and `SystemErr.log` files for information. Other log files, such as FFDC and trace logs, can also have useful information.

WebSphere Education IBM

Configuring server log trace settings

- In some cases, you might find it helpful to configure server log trace settings to collect specific information
- Set log trace settings in the WebSphere Application Server administrative console for Process Center or Process Server
- Process application deployment trace string for Process Center and Process Server target cluster servers:

```
*=info:WLE.*=all: WAS.clientinfopluslogging=all:
 com.ibm.bpm.fds.*=all:
ProcessApplicationLifecycle=all:
```


Figure 7-27. Configuring server log trace settings WB8691.0

Notes:

To set server log trace settings:

- Access the WebSphere Application Server administrative console for Process Center or Process Server, and go to **Troubleshooting > Logs and Trace.**
- Select the server that you want to trace, and select **Change log detail levels** to get to trace settings.
- Use the **Configuration** tab to set the string on the master configuration.
- Use the **Runtime** tab to set the string on the running environment.
- Apply your changes.
- If you use the **Configuration** tab, save your changes and restart the servers.

WebSphere Education

Trace settings and log files

- If you are restarting servers, delete the old log files
  - After all servers are stopped, delete the existing files in the profile **`logs`** directory
  - New log files are created when the servers restart
  - Reduces the amount of logged information that you need to review
- Setting trace settings can slow system performance
  - Remember to clear the trace settings after you collect the information you need

Figure 7-28. Trace settings and log files WB8691.0

Notes:

WebSphere Education IBM

Collect other diagnostic troubleshooting data (1 of 2)

- In addition to log files, other data can help you troubleshoot process application deployment
- The troubleshooting data that you collect can help IBM Support
- Test case: Create a simplified and isolated test case
  - Export as a `.twx` file from IBM Process Designer
  - Include Project Interchange (PI) and enterprise archive (EAR) file from the involved Advanced Integration Services, if applicable
  - Document steps to reproduce problem
- If you deploy a snapshot to an offline Process Server:
  - Provide the complete command that is failing
  - Provide the output from the command-line console

Figure 7-29. Collect other diagnostic troubleshooting data (1 of 2) WB8691.0

Notes:

After checking log files, also consider the following list of common errors.

- **An exception is generated during the installation of library items and assets**.
  - The installation fails. Resolve this exception and try to install again.
- **The installation process fails to send tracking definitions to the Business Performance Data Warehouse**.
  - The installation continues, but you need to manually send the definitions after the installation is complete.
- **The installation process fails with the error: Process server does not contain sufficient capabilities to run the Process Application**.
  - Ensure that the target Process Server supports the capability that the snapshot uses.
  - For example, if your process application contains Advanced Integration services (Service Component Architecture modules and dependent libraries), you can install it only on a Process Server that is configured for IBM Business Process Manager Advanced. You must

remove these artifacts before you can install on an IBM BPM Standard or IBM BPM Express Process Server.

- You might receive this error if you previously opened the process application or toolkit in Integration Designer. Opening a process application or toolkit in Integration Designer enables IBM BPM Advanced capabilities in the process application. Therefore, the process application or toolkit requires an IBM BPM Advanced Process Server to deploy it. Restore the process application or toolkit to IBM BPM Standard capabilities by disassociating the modules and libraries that were generated when the process application or toolkit was opened, and then try to install again.

- **The installation service generates an exception when installing the toolkit or process application**.
  - The installation fails. Consider building your installation service to capture exceptions and roll back any changes that are made before the exception is generated. If your installation services do not handle exceptions, you might need to manually roll back changes before attempting to reinstall.
  - Consider this example scenario: all toolkit installation services complete and then the installation service for the process application fails halfway through its execution. You might need to roll back changes that result from partial completion of this step. The toolkit installation is complete and does not need to be run again.
- **The wsadmin connection times out while the BPMInstall command is running**
  - If you are using a SOAP connection, the time that is required for the BPMInstall command to complete often exceeds the default SOAP timeout value. Even though the command continues to run until it completes, you might see the exception `java.net.SocketTimeoutException: Read timed out`. If you want to increase the timeout value to prevent this exception, edit the `com.ibm.SOAP.requestTimeout` property in the `profile_root/properties/soap.client.props` file.

WebSphere Education

Collect other diagnostic troubleshooting data (2 of 2)

- Create output of the contents of the following database tables:
  - LSW_SERVER: on Process Center
  - LSW_DEPLOYMENT: only on Process Center
  - LSW_INSTALLATION: only on Process Server
  - LSW_SNAPSHOT: from both Process Center and Process Server

- CSV output of these three database tables can help identify the snapshot and its associated process application or toolkit
  - LSW_SNAPSHOT
  - LSW_BRANCH
  - LSW_PROJECT

Figure 7-30. Collect other diagnostic troubleshooting data (2 of 2) WB8691.0

Notes:

A database configuration is part of the overall IBM Business Process Manager topology.

For IBM Business Process Manager, three separate databases are required for the Process Server, the Performance Data Warehouse, and the common database components.

You can use the Data Explorer view in the IBM Integration Designer to connect and view database tables.

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Explain basic process application concepts
- Describe process application management
- Describe user permissions issues with process applications
- Explain troubleshooting actions for process applications

Figure 7-31. Unit summary WB8691.0

Notes:

WebSphere Education

Checkpoint questions

1. True or False: IBM Business Process Manager requires unique snapshot acronyms.

2. The `wsadmin` command `BPMDeleteSnapshot` does which of the following actions?
   a. Configure the Process Center `100Custom.xml` file to automatically delete unnamed snapshots.
   b. Delete snapshots and their associated dependencies for a specific process application.
   c. Delete all unnamed and archived process application snapshots.

3. True or False: By default, anyone in the `tw_admins` group can install a process application on either an online or an offline Process Server.

Figure 7-32. Checkpoint questions WB8691.0

Notes:

Write down your answers here:

1.

2.

3.

WebSphere Education

Checkpoint answers

1. False. *IBM Business Process Manager does not require unique snapshot acronyms, but it does require unique process application acronyms. However, it is a good practice to ensure that that snapshot acronyms are also unique, since it can help avoid installation issues.*

2. b. Delete snapshots and their associated dependencies for a specific process application.

3. True.

Figure 7-33. Checkpoint answers WB8691.0

Notes:

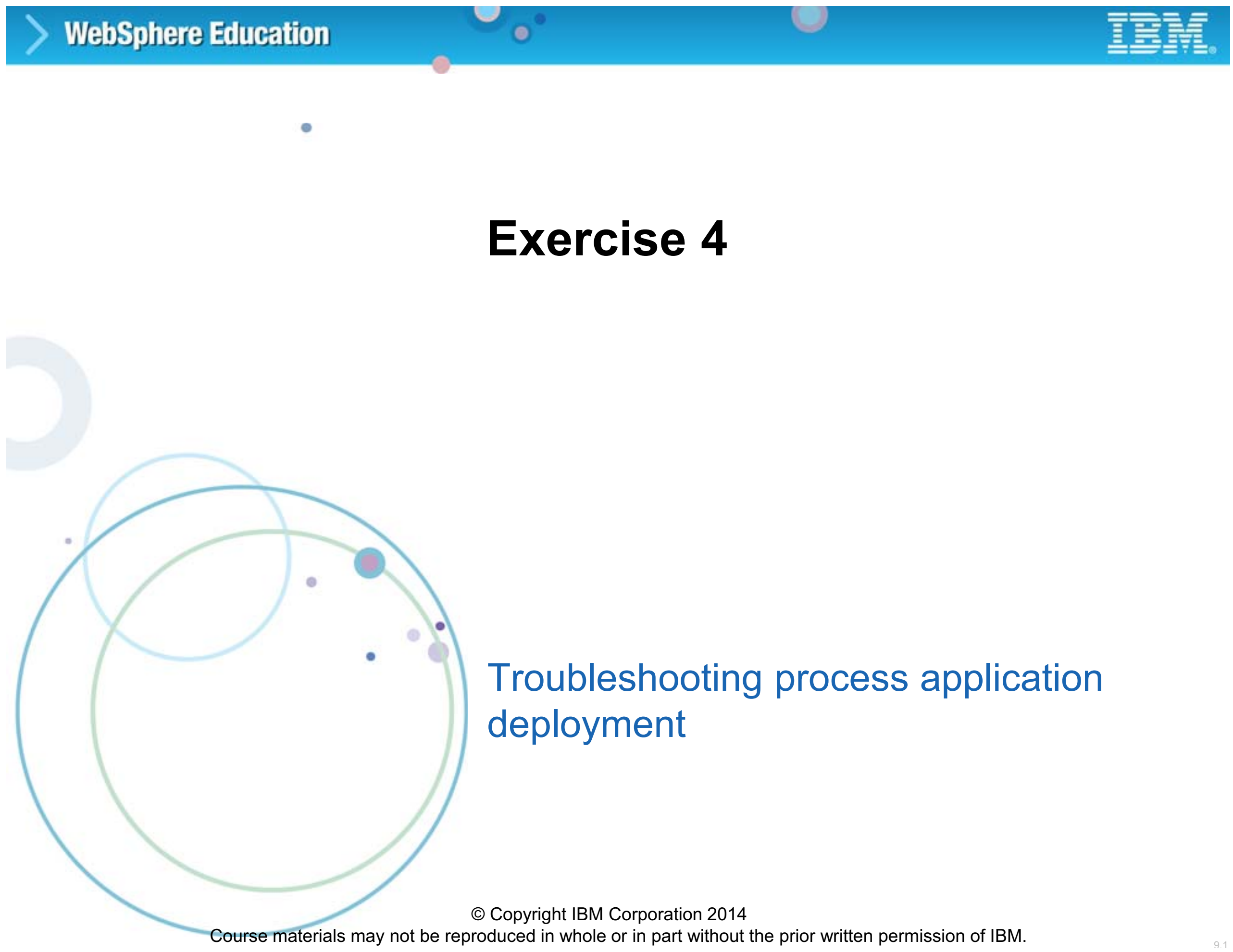


Figure 7-34. Exercise 4 WB8691.0

Notes:

WebSphere Education

Exercise objectives

After completing this exercise, you should be able to:

- Create users in the Process Admin console
- Add users to groups in the Process Admin console
- Check server log files and identify issues
- Search the Business Process Manager product documentation for troubleshooting information

Figure 7-35. Exercise objectives WB8691.0

Notes:

Unit 8. IBM Business Process Manager Standard runtime problems

What this unit is about

This unit covers the IBM Support Assistant tool and MustGather data. IBM Support Assistant is a free tool that IBM Support provides. It includes different troubleshooting tools that can help you collect and analyze data for troubleshooting. The term “MustGather data” refers to diagnostic data that IBM Support uses to resolve issues. You can collect MustGather data for Business Process Manager by using the IBM Support Assistant Data Collector.

What you should be able to do

After completing this unit, you should be able to:

- Describe the REST API tester and use it to make API calls
- Describe how to move tokens in process instances
- Describe Event Manager and related database tables
- Describe how to use Process Monitor and Instrumentation logs
- Describe how to migrate inflight instance data

How you will check your progress

- Checkpoint

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Describe the REST API tester and use it to make API calls
- Describe how to move tokens in process instances
- Describe Event Manager and related database tables
- Describe how to use Process Monitor and instrumentation logs
- Describe how to migrate inflight instance data

Figure 8-1. Unit objectives WB8691.0

Notes:

WebSphere Education

Topics

- REST API
- Event Manager
- Runtime data

Figure 8-2. Topics WB8691.0

Notes:

8.1. REST API

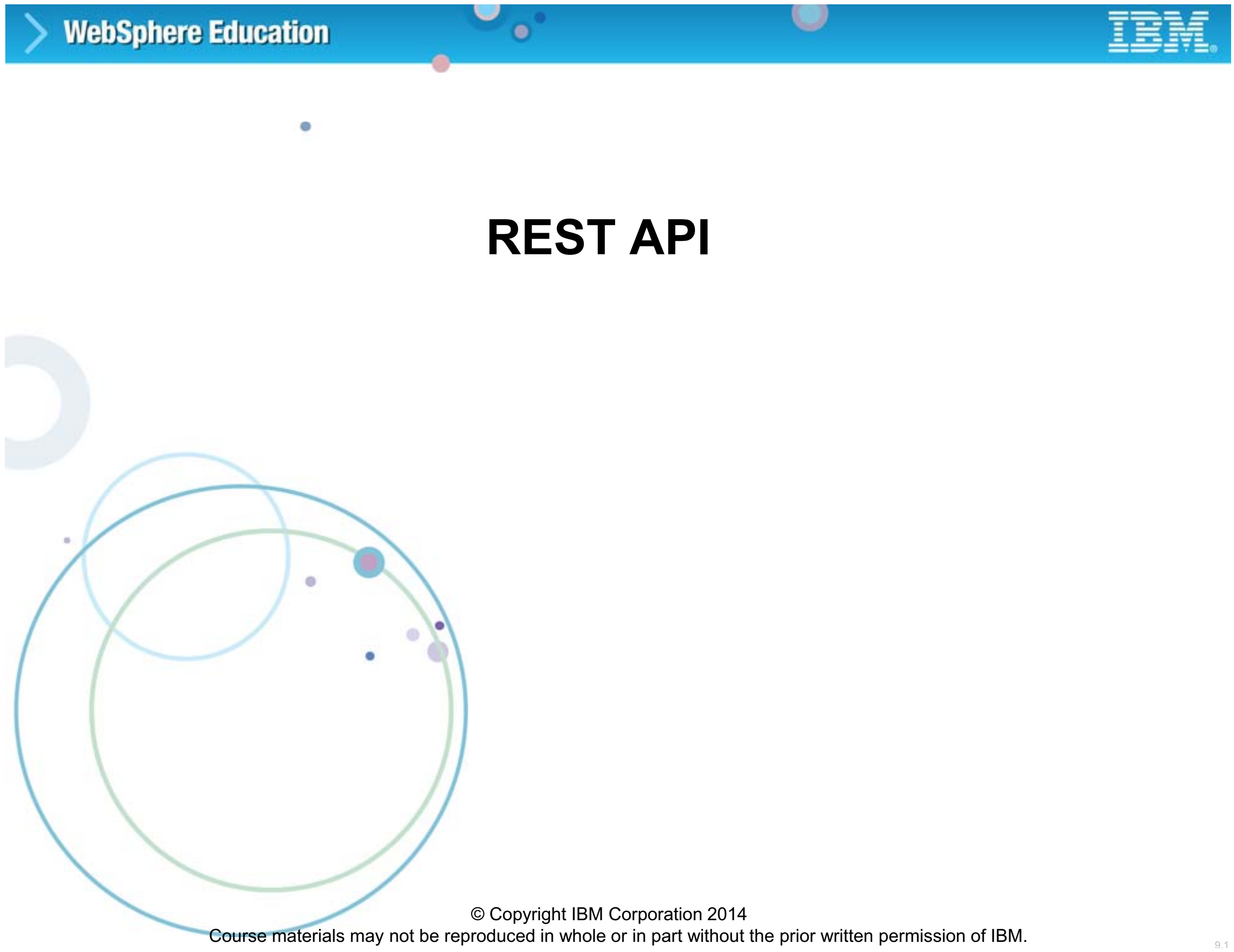


Figure 8-3. REST API WB8691.0

Notes:

WebSphere Education

What is REST API in IBM BPM? (1 of 2)

- IBM Business Process Manager provides a set of APIs that are implemented based on Representational State Transfer (REST) services
  - A set of REST resources is available for accessing IBM BPM artifacts, including business processes and tasks
  - The IBM BPM REST APIs support various Internet standards
  - The IBM Business Process Manager resources are BPD-related, BPEL-related, or federated resources
  - The resource URIs represent the IBM BPM resources
  - The HTTP methods provide the operations, such as create, read, update, and delete, which you can perform on IBM BPM artifacts

Figure 8-4. What is REST API in IBM BPM? (1 of 2) WB8691.0

Notes:

A resource can be any coherent and meaningful concept that might be addressed

A representation of a resource is typically a document that captures the current or intended state of a resource

A uniform interface separates clients from servers. This separation of concerns means that clients are not concerned with data storage, which remains internal to each server, so that the portability of client code is improved. Servers are not concerned with the user interface or user state so that servers can be simpler and more scalable. Servers and clients might also be replaced and developed independently; the interface between them is not altered.

WebSphere Education

What is REST API in IBM BPM? (2 of 2)

- Clients initiate requests to servers; servers process requests and return appropriate responses
  - Requests and responses are built around the transfer of representations of resources
- A resource can be any coherent and meaningful concept that might be addressed
- A representation of a resource is typically a document that captures the current or intended state of a resource
- A uniform interface separates clients from servers

Figure 8-5. What is REST API in IBM BPM? (2 of 2) WB8691.0

Notes:

WebSphere Education

Principles of REST (1 of 3)

- Clients initiate requests to servers; servers process requests and return appropriate responses
  - Requests and responses are built around the transfer of representations of resources
- Individual resources are identified in requests, for example by using URIs in web-based REST systems
  - The resources are conceptually separate from the representations that are returned to the client
  - The server does not send its database but some HTML, XML, or JSON that represents records, depending on the details of the request and the server implementation

Figure 8-6. Principles of REST (1 of 3) WB8691.0

Notes:

WebSphere Education

Principles of REST (2 of 3)

- When a client holds a representation of a resource, including any metadata that is attached, it has enough information to modify or delete the resource on the server, provided it has permission to do so
- If an error occurs during the processing of a REST request, the REST methods return an appropriate HTTP status code to the calling client
  - 200 OK
  - 400 Bad Request
  - 401 Unauthorized
  - 404 Not Found

Figure 8-7. Principles of REST (2 of 3) WB8691.0

Notes:

WebSphere Education IBM

Principles of REST (3 of 3)

- **REST URI**

  http://{*host*}:{*port*}/rest/{*component*}/v1/{*resource*}?{*query*}

  Where:

  "`http://{host}:{port}`" contains the host address and port

  "`/rest/{component}`" is the context root, where component is an IBM REST component

  "`/v1/{anyResource}?{query}`", together with the host address or port and context root, represents the IBM Business Process Manager resource

- **For example:**

  Request:

  `http://localhost:9080/rest/bpm/wle/v1/process/903?parts=all`

Figure 8-8. Principles of REST (3 of 3) WB8691.0

Notes:

The resource URIs represent the IBM BPM resources.

The REST resource URIs have the following format:
http://{*host*}:{*port*}/rest/{*component*}/v1/{*resource*}?{*query*}

Where:

- {*host*}:{*port*} is the host address and port of the REST API endpoint, for example: `MyProcessServer:9080`
- rest/{*component*} is the configurable context root, including one of the following IBM BPM REST component names:
  - `/rest/bpm/htm`
  - `rest/bpm/bfm`
  - `/rest/bpm/federated/htm`
  - `/rest/bpm/federated/bfm`
  - `/rest/bpm/wle`

- `/v1` is the version of the REST resource. REST resources might get updated by IBM; however, the changes are always compatible with existing REST client applications. The following changes to REST resources are considered to be compatible changes and must be expected in client applications:
    - Support for new MIME types or resource representations.
    - The addition of new properties to existing JSON objects that are returned to a client. Clients that are not at the latest level ignore these properties.
- `{`*`resource`*`}` is the hierarchically organized part of the resource identification.
- `{'`*`query`*`'}` is the non-hierarchical part of the resource identification. Typically, this part of the resource identification consists of parameters that are passed to the method implementation.

WebSphere Education

Testing the REST APIs

- A test tool is provided with the IBM BPM REST APIs
  – You can use this tool to help you learn about the REST APIs, and to test those APIs that you are planning to use in your application
- The REST API test tool is a web application that you can use to prototype the following IBM BPM REST resources and their associated parameters
- URL: `http://`*`host_name`*`:`*`port_number`*`/bpmrest-ui`

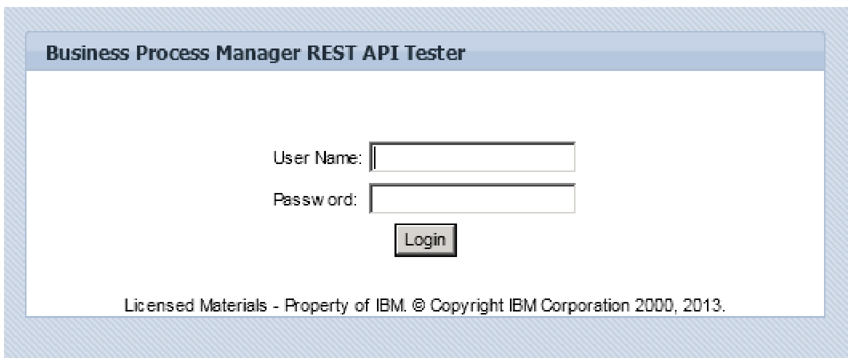


Figure 8-9. Testing the REST APIs WB8691.0

Notes:

- Direct your web browser to the URL for the REST API test tool. The value of the URL depends on how the virtual host and context root were configured for your installation.

The URL takes the following form: `http://`*`host_name`*`:`*`port_number`*`/bpmrest-ui`

Where:

- *host_name* is the network name for the host of the process server.
- *port_number* is the port number that the REST API test tool uses. The port number depends on your system configuration. The default port number is 9080. For example, if you wanted to test the REST APIs available on port 9080 on your server, you would open the `http://myserver1:9080/bpmrest-ui` URL in your browser.
  - In the REST API test tool, select an API.
  - Optional: Enter any associated parameters that you want to include in your test.
  - Click **Execute Call** to start the test.

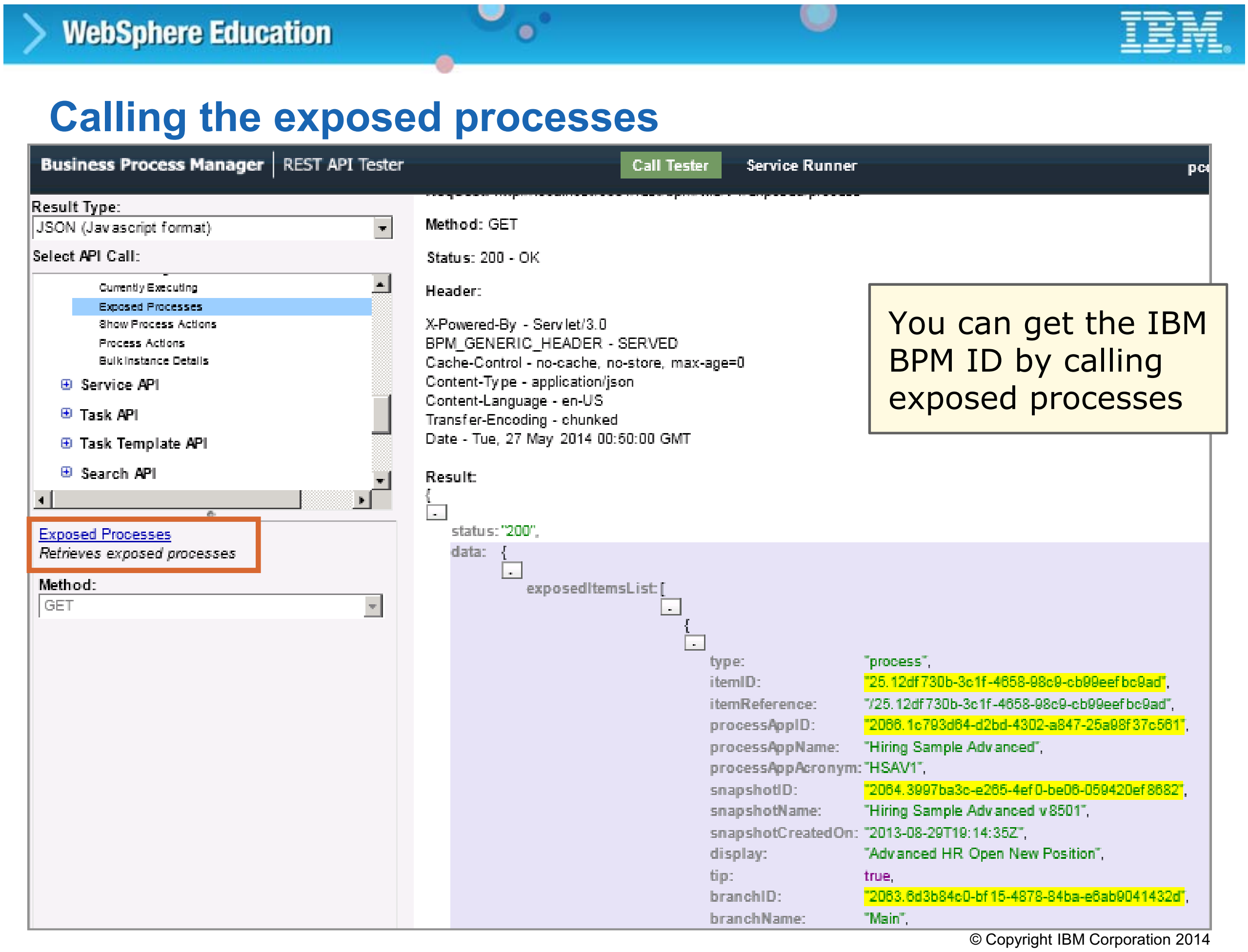


Figure 8-10. Calling the exposed processes WB8691.0

Notes:

To get the IBM BPM ID for exposed processes, you expand the Business Process Manager REST APIs and then under Process API select **Exposed Processes**. When you click **Execute**, it returns a list of all the exposed processes and their item IDs and snapshot details.

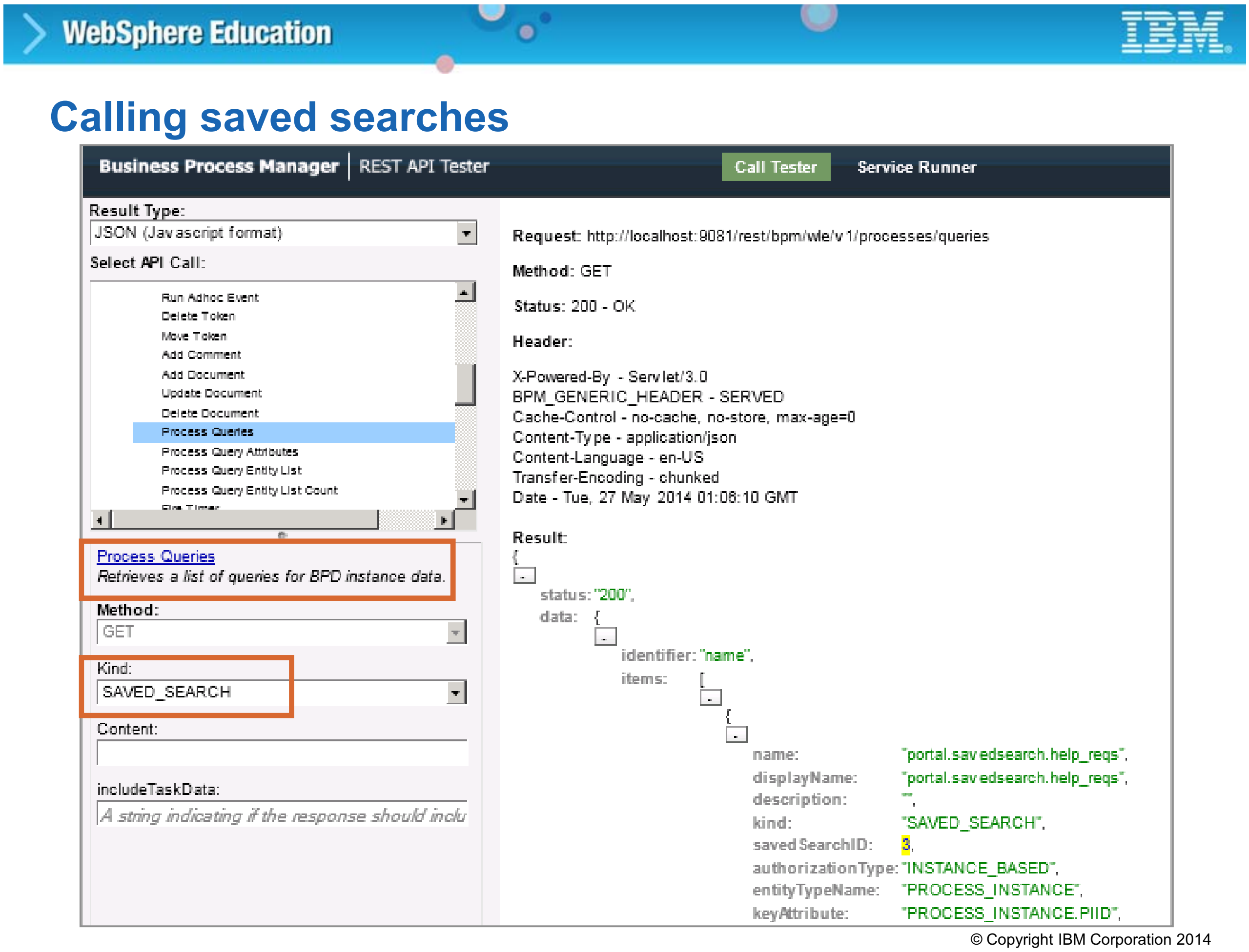


Figure 8-11. Calling saved searches WB8691.0

Notes:

1. Expose the variable to be searched in Process Designer.
2. In Process Admin Console, define the saved search and test to see whether it works.
3. In Process Portal, launch the Process Instance and execute the search.
4. Call the Search from the REST API tester.
   - You can use the Process Queries API call to return a list of all defined saved search queries. As soon as the REST client page is loaded, expand **BPM REST APIs > Process API > Process Queries**.
   - Find out the attributes.
   - Call the saved search: **BPM REST APIs > Process API > Process Query Entity List**.

WebSphere Education

Moving tokens with REST API (1 of 2)

- You can use REST APIs to move tokens
- How do you move a token on an instance?
  - Find the token ID with Inspector
  - Get the target ID to move to using the REST API – Current State
  - Get the Process Instance with Inspector

```
/rest/bpm/wle/v1/process/instance_ID?action=moveToken&
tokenId= token_ID
&target=target_step_ID[&resume=resume_value]
```


Figure 8-12. Moving tokens with REST API (1 of 2) WB8691.0

Notes:

To move a token to a new activity, complete the following steps:

1. Use the Process Inspector in the Process Admin Console to inspect the currently running process instances, and select the instance that contains orphaned tokens that you want to move. Make a note of the process instance ID and also the ID of the orphaned token, which is shown in the execution call stack.
2. Use Process Designer to identify the system ID of the activity where you want to move the orphaned token (the target activity). Record the complete value of that activity, including the `bpdid:` prefix.
3. To move the token to an activity, in the REST API client, enter the following text: `/rest/bpm/wle/v1/process/`*`instance_ID`*`?action=moveToken&tokenId=` *`token_ID`* `&target=`*`target_step_ID`*`[&resume=`*`resume_value`*`]`
   - The following identifies the parameters of the API:
     - `instance_ID`: The instance ID number of the process instance that contains tokens to be moved
     - `action`: States the action to be taken (`moveToken`)

- `token_ID`: The token ID number of the token you need to move
- `target_step_ID`: The ID number of the new process step to which you are moving the token
- `resume_value`: The action that is used to resume the instance after moving the token (set to `true` or `false`); the default value is `true`

4. Press Enter to complete the move.

WebSphere Education

Moving tokens with REST API (2 of 2)

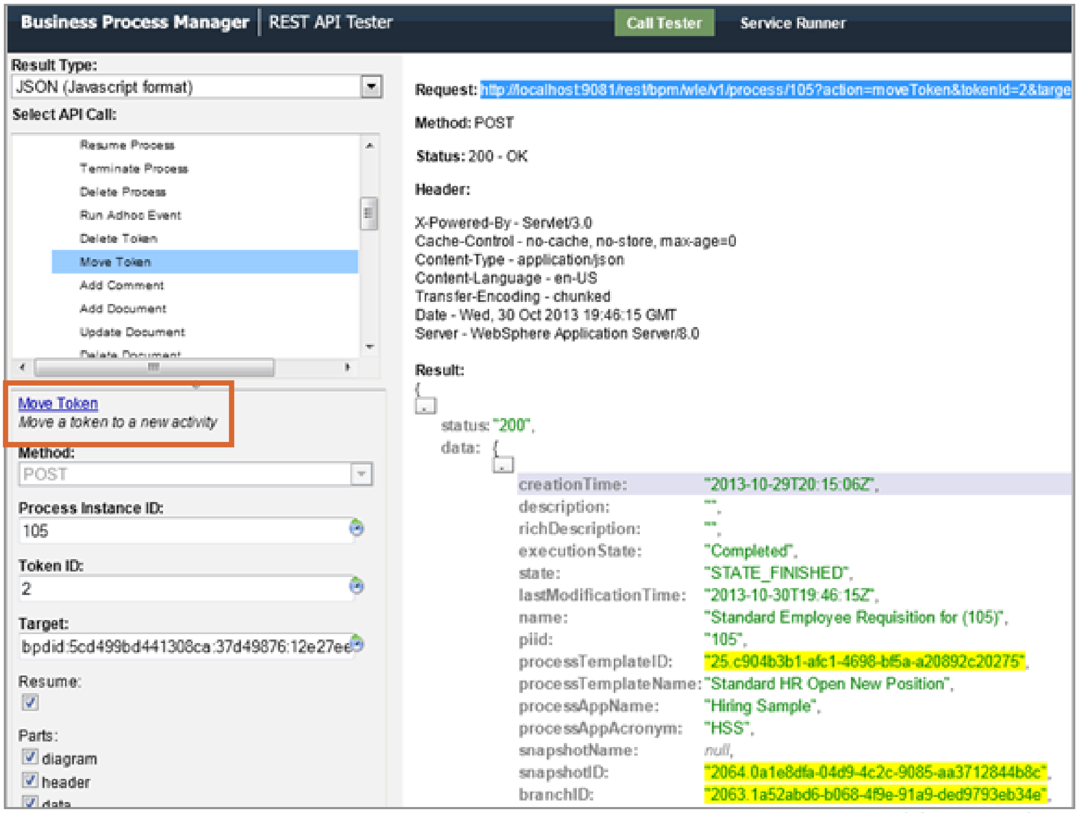


Figure 8-13. Moving tokens with REST API (2 of 2) WB8691.0

Notes:

WebSphere Education

Restricting access to REST Tester and APIs

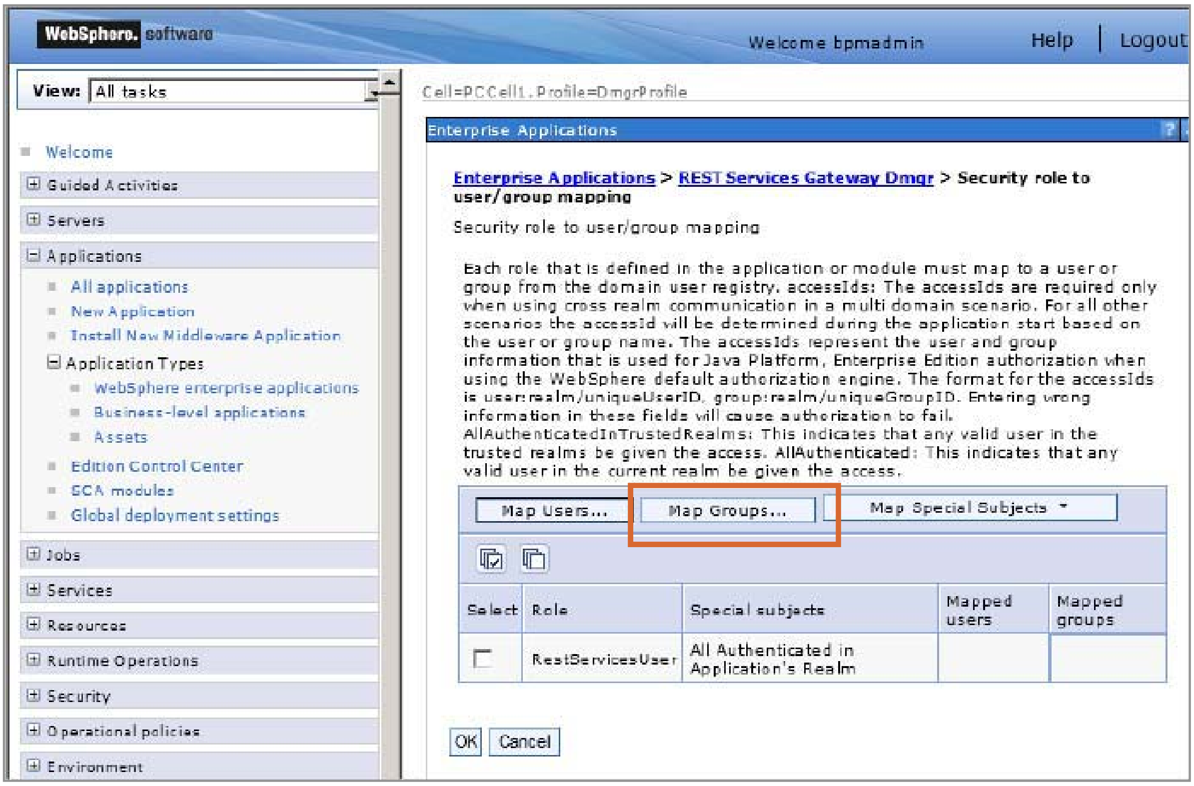


Figure 8-14. Restricting access to REST Tester and APIs WB8691.0

Notes:

You can restrict access to the REST API tester by creating a security group and mapping it to the RestServicesUser role.

WebSphere Education

Trace to enable

- com.ibm.bpm.rest.*=all : com.ibm.bpm.wle.query.*=all:com.ibm.bpm.wle.util.*=all

Figure 8-15. Trace to enable WB8691.0

Notes:

To troubleshoot REST API calls, you can enable the following trace in the administrative console:

```
com.ibm.bpm.rest.*=all : com.ibm.bpm.wle.query.*=all:com.ibm.bpm.wle.util.*=all
```

8.2. Event Manager

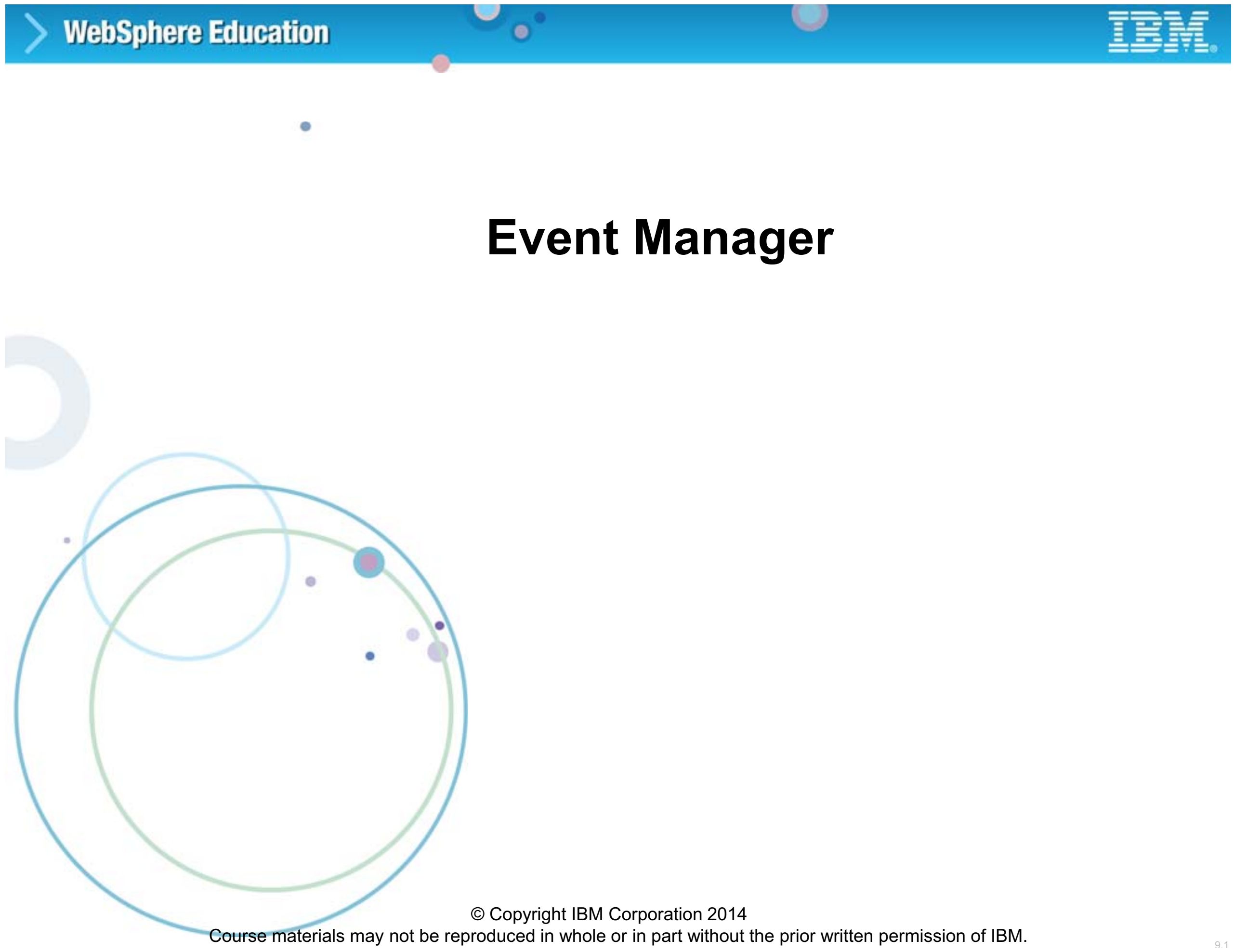


Figure 8-16. Event manager WB8691.0

Notes:

WebSphere Education

What are the engines?

- There are two main work “engines” that allow IBM BPM to work
- These engines work together to move items forward. The engines are:
  - The ***BPD engine***
  - The ***Service engine***
- These two engines drive the processes and services while scheduling through the ***Event Manager***, which ensures no conflicts

Figure 8-17. What are the engines?   WB8691.0

Notes:

WebSphere Education

BPD engine

- Progresses BPDs
- BPDs can run in parallel but can run ***one token per instance at a time only***
- BPDs can have multiple paths and “look” like they are parallel, but they are not
  – Two branches of one BPD instance cannot be run at the same time
- The BPD engine moves BPDs to the next activity (user task, system lane task, join, and other activities)

Figure 8-18. BPD engine WB8691.0

Notes:

WebSphere Education IBM

Service engine

- Items here can be run in parallel
- The engine runs a service to completion or to a coach (which requires human interaction)
- Running a task directly or from the Event Manager kicks off the service engine

Figure 8-19. Service engine WB8691.0

Notes:

WebSphere Education

Event Manager (EM)

- The Event Manager's primary function is to guarantee scheduled execution of code
- It does not execute code, just schedules it
  - UCAs, BPD token movement, system lane activity
- The EM can be running or paused
- The EM has a heartbeat
  - A heartbeat is a dedicated thread of an Event Manager instance that periodically signals that the Event Manager instance is still running
- The EM uses queues to manage its events
- One EM per AppTarget
- Moves one token on a BPD at a time

Figure 8-20. Event manager (EM) WB8691.0

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, Process Server receives an event, and 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 IBM

Event Manager view

- All schedulers (Event Managers) whose heartbeat ever ran are listed in the LSW_EM_INSTANCE table, and are shown on the **EM Monitor page** in the Process Admin Console:

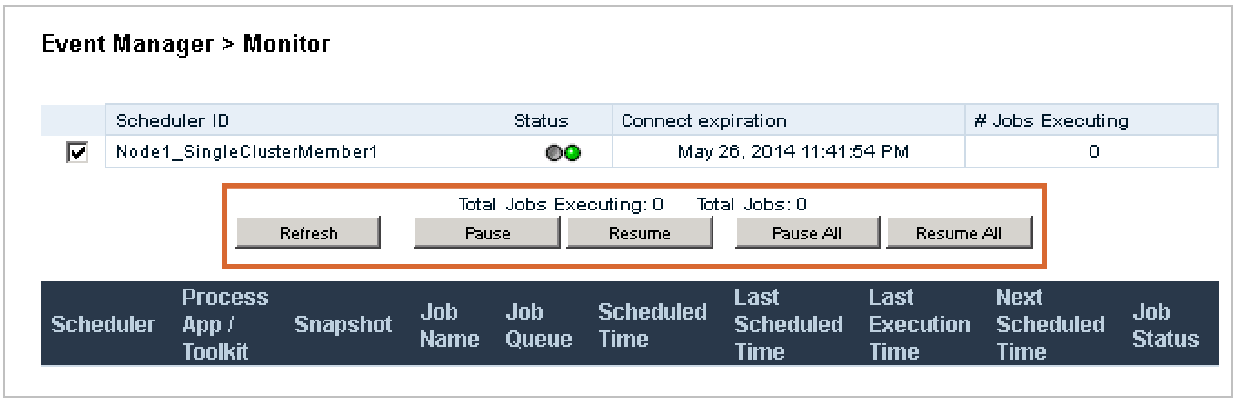


Figure 8-21. Event Manager view WB8691.0

Notes:

When you access the Event Manager monitor, you can see the status for each scheduled job. In a clustered environment, the monitor displays all schedulers in the cluster and the jobs for all Schedulers in the cluster.

Click **Pause** or **Pause All** to pause a selected scheduler or all schedulers. If you pause a Scheduler, any executing jobs are completed before processing is halted. Click **Resume** or **Resume All** to resume processing of the selected scheduler or all schedulers.

WebSphere Education

Event Manager data

The key EM tables are:

- LSW_EM_INSTANCE: A row for each scheduler
- LSW_EM_TASK: A row for each non-completed task, includes a reference to LSW_EM_INSTANCE to indicate which scheduler owns each task
- LSW_EM_TASK _KEYWORD: One-to-many table mapping a task to its associated keywords
- LSW_EM_SYNC_QUEUE: A row for each sync queue, includes a reference to LSW_EM_INSTANCE to indicate which scheduler owns each sync queue
- LSW_UCA_BLACKOUT: A row for each blackout period, including the next transitions into and out of blackout

Figure 8-22. Event manager data WB8691.0

Notes:

Any work that a specific EM schedules is run on the local Process Center Server. The EM scheduler is used anytime a UCA is invoked. It is also used for processing BPD notifications, executing BPD system lane activities, and executing BPD timer events. It is not specific to TWEvents or to UCAs.

In other words, the event manager (EM) is the broker that drives BPD execution and UCA execution. Each BPD token progression is a BPD task in the EM.

By default, every time IBM BPM asks the EM to do work (UCA, user task, BPD notification to advance a token), it forces the EM to reload its queues from the database and act immediately.

WebSphere Education

LSW_EM_TASK table

- Reads from LSW_EM_TASK table
- Loads into queues for work
- Overloaded queues bog down system
  - Solution: Change queue size or add more nodes
- When work is complete, entry is removed from LSW_EM_TASK table
  - Appropriate task or instance data is updated

Figure 8-23. LSW_EM_TASK table WB8691.0

Notes:

WebSphere Education IBM.

Database table LSW_EM_TASK

task_id	integer	Primary key, unique to this table and ***not*** related to LSW_TASK
description	varchar	Name of task that is executing, comes from task data; it is presented in the event manager monitoring page
queue_id	integer	Async or sync queues
scheduled_time	datetime	When the task is going to run; relevant for timers and scheduled UCAs
discriminator	decimal	
task_status	decimal	
task_owner	integer	Who invoked the task; relates back to LSW_USR_XREF table user_id column
blackout_calendar_id	decimal	

Figure 8-24. Database table LSW_EM_TASK WB8691.0

Notes:

WebSphere Education IBM

Description of LSW_EM_TASK

blackout_behavior		
task_execution_class	varchar	Code that is executing this event com.lombardisoftware.bpd.runtime.engine. quartz.DbNotificationBpdTask
task_arguments	image	
repeat_string	varchar	
in_closing_transaction	decimal	
re_execution_count	integer	Related to UCA execution generally
last_scheduled_time	datetime	
last_execution_time	datetime	When the EM processed this item

Figure 8-25. Description of LSW_EM_TASK WB8691.0

Notes:

WebSphere Education

Understanding EM queues

There are two types of EM queues:

- Asynchronous (async) are executed as soon as possible with no guaranteed order
- Synchronous (sync) are executed serially
  - If you have multiple tasks set to run on one sync queue, they will execute one after the other in the order that they are put into they sync queue
- The EM treats sync and async queues differently

Figure 8-26. Understanding EM queues WB8691.0

Notes:

- **There are two types of EM queues**: Asynchronous (async) and synchronous (sync). Async queues are executed as soon as possible with no guaranteed order. Sync queues are executed serially. If you have multiple tasks set to run on one sync queue, they will execute one after the other in the order in which they were put in to they sync queue. The EM treats sync and async queues differently.
- **Sync queue**: Each task in a sync queue must be executed in serial. To prevent problems in a cluster, an EM claims ownership of one or more sync queues when it starts. The ownership is stored in the LSW_UCA_SYNC_QUEUE where QUEUE_OWNER is linked to OWNER_ID in LSW_EM_INSTANCE. It is not a permanent assignment. The LSW_EM_INSTANCE table tracks status of all of the event managers. The status is checked every 15 seconds. If the owner of a sync queue is no longer available, another EM takes ownership of that sync queue.
- **Async queue**: Each EM picks up async tasks when there is room in their async queue for more tasks. Each process server has its own running EM.

WebSphere Education IBM

Event Manager tips

- The Event Manager is quick and efficient
  - Usually it is the tasks that it is executing that slow it down, not the EM itself
- If you want to throttle the Event Manager, do not decrease the thread pool; instead, decrease the queue capacity
- A sync queue can get stuck since it does not advance until the task completes
  - To make it less of a problem, create multiple sync queues
  - You can manage sync queues in the IBM BPM console
- Keeping system clocks in sync is always a good idea
  - All the timestamps that the EM scheduler uses – the heartbeat's expirations and the task's scheduled times – are interpreted relative to the database machine's system clock

Figure 8-27. Event manager tips WB8691.0

Notes:

WebSphere Education

How do tokens fit in?

- Think of a token as a marker as to where in the BPD or service you are
- The tokens can be passed from BPD to an activity (sub-BPD or service)
  - If the activity is a sub-BPD, the token stays in the BPD engine, and a new token is created for the sub-BPD for execution
  - If the activity is a service, the token stays in the BPD engine, and a task is scheduled in the service engine for execution of the activity
  - Note: The service engine does not use tokens; it simply executes services (code)

Figure 8-28. How do tokens fit in? WB8691.0

Notes:

8.3. Runtime data

WebSphere Education

Runtime data

9.1

Figure 8-29. Runtime data WB8691.0

Notes:

WebSphere Education

How do tokens fit in?

- Think of a token as a marker as to where in the BPD or service you currently are
- The tokens can be passed from BPD to an activity (sub-BPD or service)
  - If the activity is a sub-BPD, the token stays in the BPD engine, and a new token is created for the sub-BPD for execution
  - If the activity is a service, the token stays in the BPD engine, and a task is scheduled in the service engine for execution of the activity
  - Note: The service engine does not use tokens by itself; it simply executes services (code)

Figure 8-30. How do tokens fit in? WB8691.0

Notes:

Tokens can move from BPD to activities.

Tokens can encounter an endpoint return to the parent or calling item or end the instance (if already at the top level).

WebSphere Education

Orphaned tokens

- Orphaned tokens are a problem because you can do nothing with them
- If you see them, often you must terminate the instance that you see them in
- Be careful with splits and joins
  - Generally speaking, you must have an equal number of simple splits and joins; otherwise, you might get orphaned tokens
- Another way to see orphaned tokens is code promotion
  - Sometimes people change lines or activities
  - If the old instances expect one path and it is now gone, there can be stranded tokens

Figure 8-31. Orphaned tokens WB8691.0

Notes:

An orphaned token is a pointer that is associated with an activity that was removed from a business process definition (BPD). You can use a policy file, a REST API, or Process Inspector to manage orphaned tokens.

Think of a token as an active execution step within the process. Tokens exist on each active activity and also for timer and message events on an active activity.

A token becomes orphaned if its associated activity is removed from a BPD of a migrated snapshot. You need to decide what to do with potential orphaned tokens or risk that the process instances do not complete. For example, suppose that you installed a new version of a process application. The new version cleans up a number of activities that are no longer used from the earlier version. However, tokens still exist for some of these unused activities. You must either delete or move these orphaned tokens, or the migrated process instances might not be able to complete. When orphaned tokens are deleted or moved, the process instance will try to resume at the next activity that contains tokens. If a next step cannot be determined from the revised BPD, the instance will complete when there are no more active tokens. For example, if you have an activity that contains three tasks (Task A, Task B, and Task C) and Task A is running, it has the token. If you delete the token while Task A is running, Task B and Task C do not run, and the process instance is considered complete.

Consider another example. Again you are installing a new version of a process application. A number of explicit exception events are removed from some of the nested processes. This removal might potentially lead to orphaned tokens when instances are migrated to the new version. It should be possible to delete (that is to ignore) these tokens when instances are migrated from the old to the new version of the process without causing instances to hang.

WebSphere Education IBM.

Managing orphaned token with a policy file

- Use a policy file to proactively compare snapshots before instance migration
  - Helps to identify the potential locations of orphaned tokens and specify whether each orphaned token should be deleted or moved
- Use the `BPMCheckOrphanTokens` command to detect the possibility of orphaned tokens before installing a new snapshot, and identify whether to delete or move each token
  - Compares two snapshots and produces an XML file that lists all the steps in the BPD where orphaned tokens might potentially occur
  - The command generates an XML representation of all possible locations of orphan tokens in a policy file
- Edit the policy file with a text editor
  - Go through the list of potential orphaned tokens and decide whether each token should be moved or deleted

Figure 8-32. Managing orphaned token with a policy file WB8691.0

Notes:

The easiest way to identify and manage orphaned tokens is to generate a policy file and use it to specify whether each potential orphaned token should be moved or deleted during instance migration. If you migrate the snapshot instance without using a policy file, orphaned tokens may be created. In this case, you can use the REST API client to delete or move these orphaned tokens. You can also use the Web Process Inspector to delete orphaned tokens.

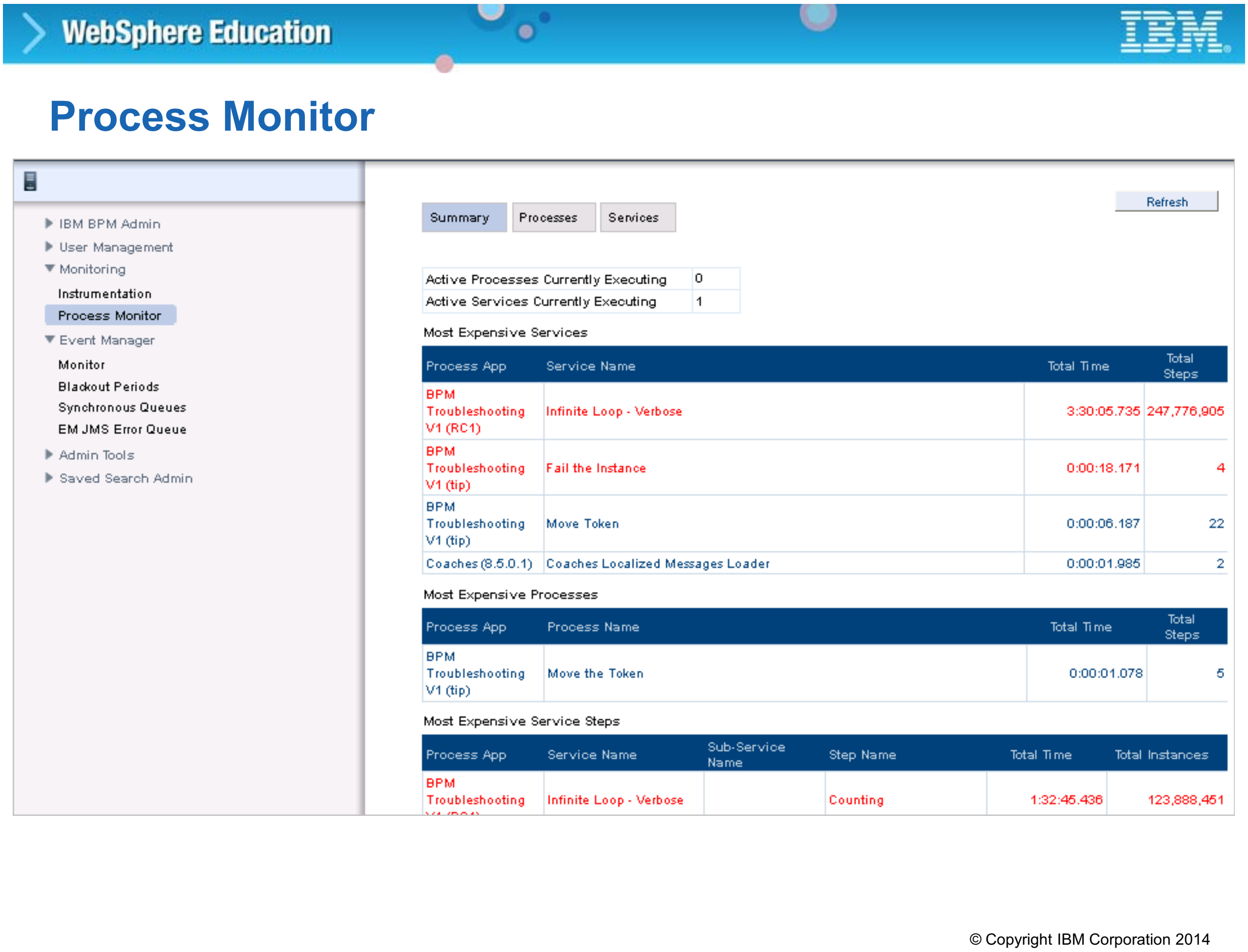


Figure 8-33. Process Monitor WB8691.0

Notes:

The Process Admin Console includes a Process Monitor that enables administrators to view the processes and services that are running on Process Server and also to stop any problematic processes or services. For example, you might need to stop a service that causes an exception or a service that is stuck in a repeating loop.

WebSphere Education IBM

Endless loop

- One of the most common pitfalls in development
- Hanging BPD: This BPD is looping while inside the BPD engine
  - No activities are implemented as a service in this BPD
- Thrashing BPD: This BPD calls a service with an error
  - There is an error handler on the activity that loops back to the same failing service
  - It is VERY hard to stop with the process monitor as the work is quickly hopping between the BPD and the service engine
- Looping error handler: This service has a global error handler
  - Global error handlers are VERY risky
  - If the step that catches the error itself has an error, you go into a tight infinite loop

Figure 8-34. Endless loop WB8691.0

Notes:

The following are some of the most common pitfalls in development:

- Hanging BPD: This BPD is looping while inside the BPD engine. No activities are implemented as a service in this BPD.
- Thrashing BPD: This BPD calls a service with an error. An error handler on the activity loops back to the same failing service. It is very difficult to stop with the process monitor as the work is quickly hopping between the BPD and the service engine.
- Looping error handler: This service has a global error handler. Global error handlers are extremely risky. If the step that catches the error itself has an error, you go into a tight infinite loop.

WebSphere Education

Instrumentation Monitor

Monitoring > Instrumentation

Start Logging

Refresh Reset Automatically refresh every Never

Name	Count/Value	In Process	Average Duration (ms)	Moving Average Duration (ms)	Total (ms)
BPD					
Instances					
BPD Instances Completed	4				
BPD name is Move the Token	4				
BPD Instances Failed	16				
BPD Instances Resumed	8				
BPD Instances Started	12				
BPD name is Move the Token	12				
BPD Instances Terminated	5				
BPD name is Move the Token	5				
Cache					
Connectors					
Webservices					
Inbound					
WebService Call	0	0	0.00	0.00	0.00
Outbound					
WebService Call	0	0	0.00	0.00	0.00
EJB Services					
deploySnapshot	0	0	0.00	0.00	0.00

Figure 8-35. Instrumentation Monitor WB8691.0

Notes:

The Process Admin Console includes an Instrumentation monitor to help identify performance bottlenecks in Process Server and to capture instrumentation data that you can use to further analyze any performance issues.

WebSphere Education IBM

Working with undercover agents (UCA)

- An undercover agent is started by an event
  - The event can be a message event, a content event, or a timer event that is the result of a specific schedule
- When an undercover agent executes, it invokes an IBM Business Process Manager service or a BPD in response to the event
- Time-based UCAs work only on Process Servers, not Process Center
- Building UCAs and having them work is tricky
  - Test a few times to make sure that payload data is working
- Time-elapsed (scheduled) UCAs fire at a set time, essentially a cron job
  - Choose which month, day of week, hour, and minute to run
- Event-based UCAs fire at a specific activity
  - Designed to route activity or wait for activity

Figure 8-36. Working with undercover agents (UCA) WB8691.0

Notes:

When you include a message event or content event in a BPD, you must attach an undercover agent to the event. For example, when a message event is received from an external system, an undercover agent is needed to trigger the message event in the BPD in response to the message.

If you want to run the startBpdWithName application programming interface (API) to start a BPD instance inside an undercover agent, set the `<enable-start-bpd-from-uca>` property to true in the `100Custom.xml` file or another override file. Restart the product, and check the `TeamworksConfiguration.running.xml` file to ensure that the setting has the appropriate value. The property is set to false by default, and if you do not change it, you might have errors that prevent the BPD from starting.

WebSphere Education IBM

Migrating inflight data

- Use Migrate inflight data if you install a new snapshot with running instances and you want to manipulate the data that the running instances use
  - This method enables you to manage potential orphaned tokens
  - It allows you to revert to a previous snapshot while you change the new one
- Only the business process definitions of the process application are migrated
  - If the process application contains BPEL processes, follow the instructions for migrating BPEL processes
- Use the Process Admin Console to migrate inflight data to a snapshot

Figure 8-37. Migrating inflight data WB8691.0

Notes:

After you install a new snapshot that replaces a snapshot with instances that are still running, you might want to migrate some data from the currently running instances to the new snapshot. In that case, use the Migrate Inflight Data window to migrate currently running instances to the new version of the selected snapshot. Wherever the running instances are in the flow of the process or service, the new version is implemented for the next step.

Use Migrate inflight data if you installed a new snapshot with running instances and you want to manipulate the data that the running instances use. This method enables you to manage potential orphaned tokens. It allows you to revert to a previous snapshot while you change to the new one.

Make whatever adjustments you want to make before completing the migration. For example, you can change the value of an environmental variable. The migration program uses the most recent value rather than the original value. The new value is applied to any new instances that are launched, but it is not applied to running instances. You can reduce the risk that an activity in a running instance does not complete by carefully managing tokens that are associated with actions that are deleted. If you are using a policy file to manage potential orphaned tokens, provide the path to that policy file.

WebSphere Education

Deployment phase

- You need to be careful with the Migrate instances option and be sure that you understand what you do
- Make sure that you test send definitions in development first and make sure that there are no problems with this part
- Test your solution by isolating any possible memory leaks, loops, and other resource consumptions; only then promote
- Perform a load test (automatic one): iterative load testing is vital for the success of all projects
  - Without practice and experience, it is possible to write code that works but does not scale
  - Without testing, you have no way to know whether it can handle even moderate load
  - The first phase of testing is performed in the development environment

Figure 8-38. Deployment phase WB8691.0

Notes:

Make sure that you test, test, and then test some more before moving to production.

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Describe the REST API tester and use it to make API calls
- Describe how to move tokens in process instances
- Describe Event Manager and related database tables
- Describe how to use Process Monitor and instrumentation logs
- Describe how to migrate inflight instance data

Figure 8-39. Unit summary WB8691.0

Notes:

WebSphere Education

Checkpoint questions

1. True or False. You can use the REST API tester tool to test the REST APIs that you are planning to use in your application.
2. True or False. Since BPDs can run in parallel, they can run two tokens per instance at a time.
3. True or False. Sync queues are executed as soon as possible with no guaranteed order.

Figure 8-40. Checkpoint questions WB8691.0

Notes:

Write down your answers here:

1.

2.

3.

WebSphere Education IBM

Checkpoint answers

1. True.
2. False. BPDs can run in parallel but can run only ***one token per instance at a time***.
3. False. Async queues are executed as soon as possible with no guaranteed order.

Figure 8-41. Checkpoint answers WB8691.0

Notes:

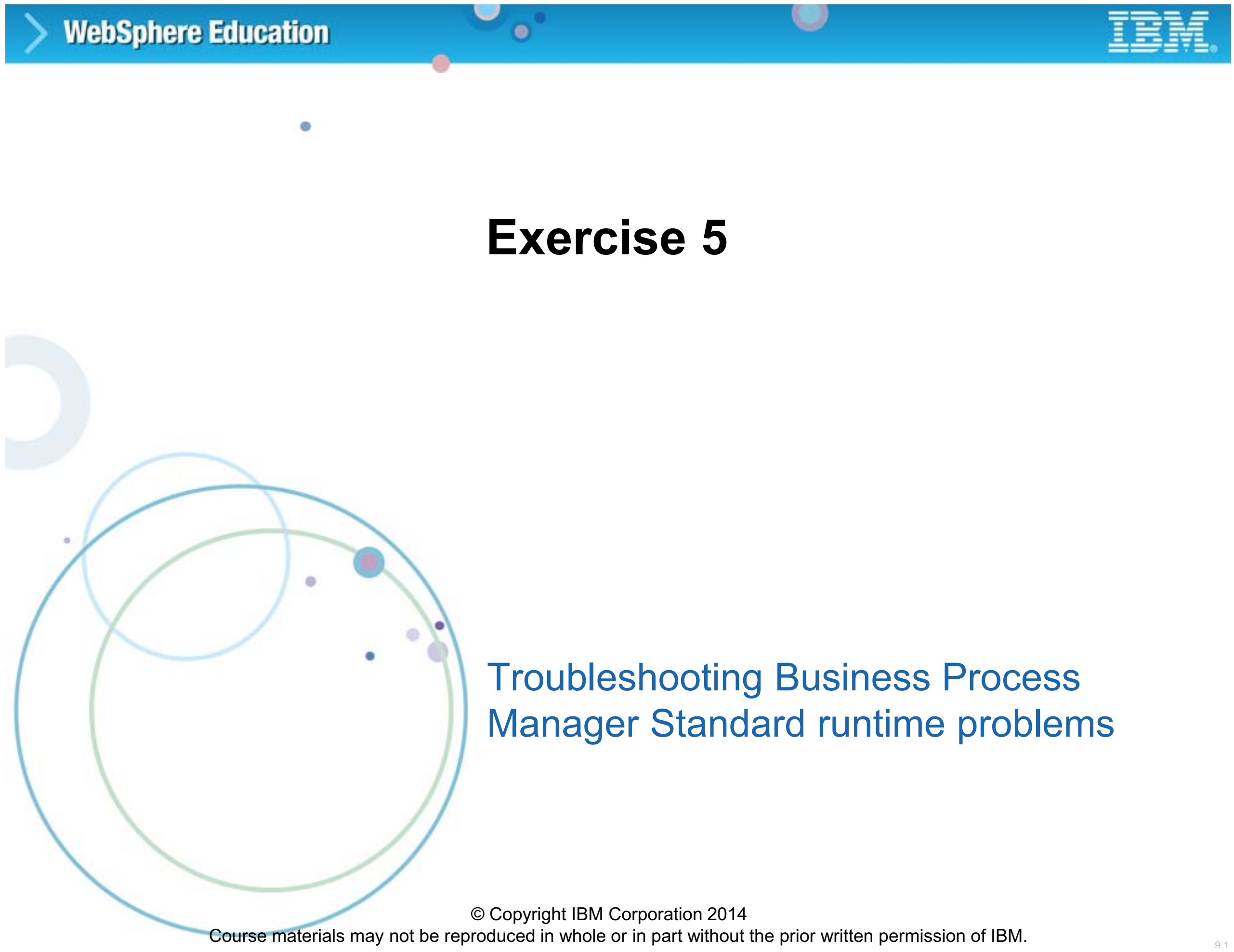


Figure 8-42. Exercise 5 WB8691.0

Notes:

WebSphere Education

Exercise objectives

After completing this exercise, you should be able to:

- Examine a failed instance in Process Inspector
- Restart a failed instance and complete the process
- Explore REST APIs by using the REST API Tester
- Troubleshoot infinite loops
- Examine Process Monitor metrics to identify problem services

Figure 8-43. Exercise objectives WB8691.0

Notes:

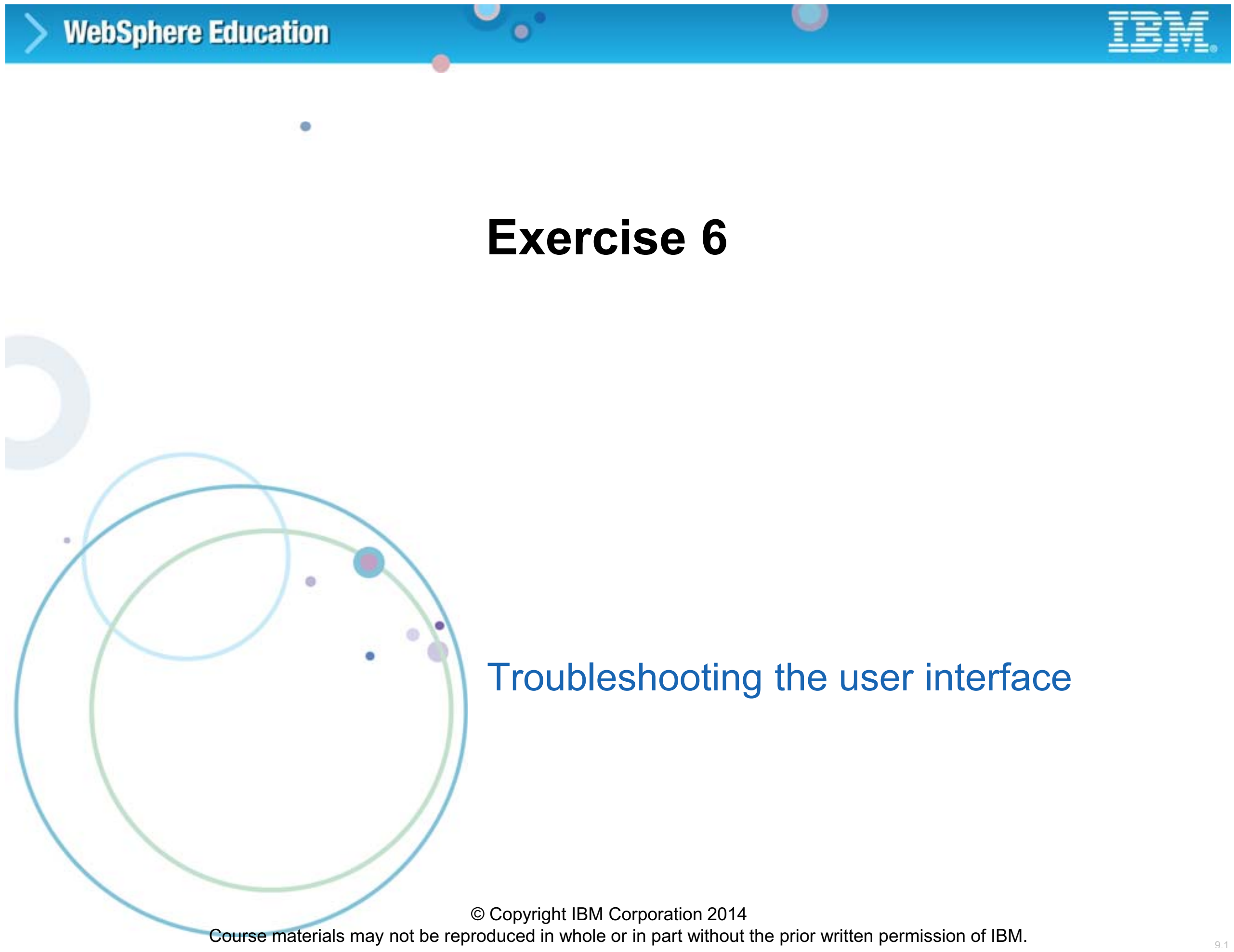


Figure 8-44. Exercise 6 WB8691.0

Notes:

Exercise objectives

After completing this exercise, you should be able to:

- Use Firefox Firebug for troubleshooting a web page
- Create a tabbed coach
- Run several instances of a coach and compare their performance

Figure 8-45. Exercise objectives WB8691.0

Notes:

Unit 9. Troubleshooting SCA runtime problems

What this unit is about

This unit covers how to approach Service Component Architecture (SCA) problems. A problem can be categorized within three types. The first type is a problem with application design; the second type is a problem with the runtime configuration; and the third type includes defects with IBM Business Process Manager. The first step is to narrow down where the problem is located. You can use several tools within IBM Integration Designer to isolate the problem. This unit focuses on troubleshooting techniques that are specific to SCA.

What you should be able to do

After completing this unit, you should be able to:

- Describe the exception types from the SCA programming model: service business exceptions and service runtime exceptions
- Describe how cross-component trace works in IBM Integration Designer
- Describe first-failure data capture (FFDC)
- Describe how to troubleshoot event sequencing

How you will check your progress

- Checkpoint

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Describe the exception types from the SCA programming model: service business exceptions and service runtime exceptions
- Describe how cross-component trace works in IBM Integration Designer
- Describe first-failure data capture (FFDC)
- Describe how to troubleshoot event sequencing

Figure 9-1. Unit objectives WB8691.0

Notes:

WebSphere Education IBM

Topics

- SCA exceptions and failed events
- SCA problem determination tools in IBM Business Process Manager and IBM Integration Designer
- Cross-component trace
- Troubleshooting event sequencing in SCA applications

Figure 9-2. Topics WB8691.0

Notes:

9.1. SCA exceptions and failed events

This topic explains the ServiceRuntimeException and ServiceBusinessException that WebSphere Process Server throws when a failed event gets created. The destinations of failed events and the failed event manager tool are also covered in this topic.

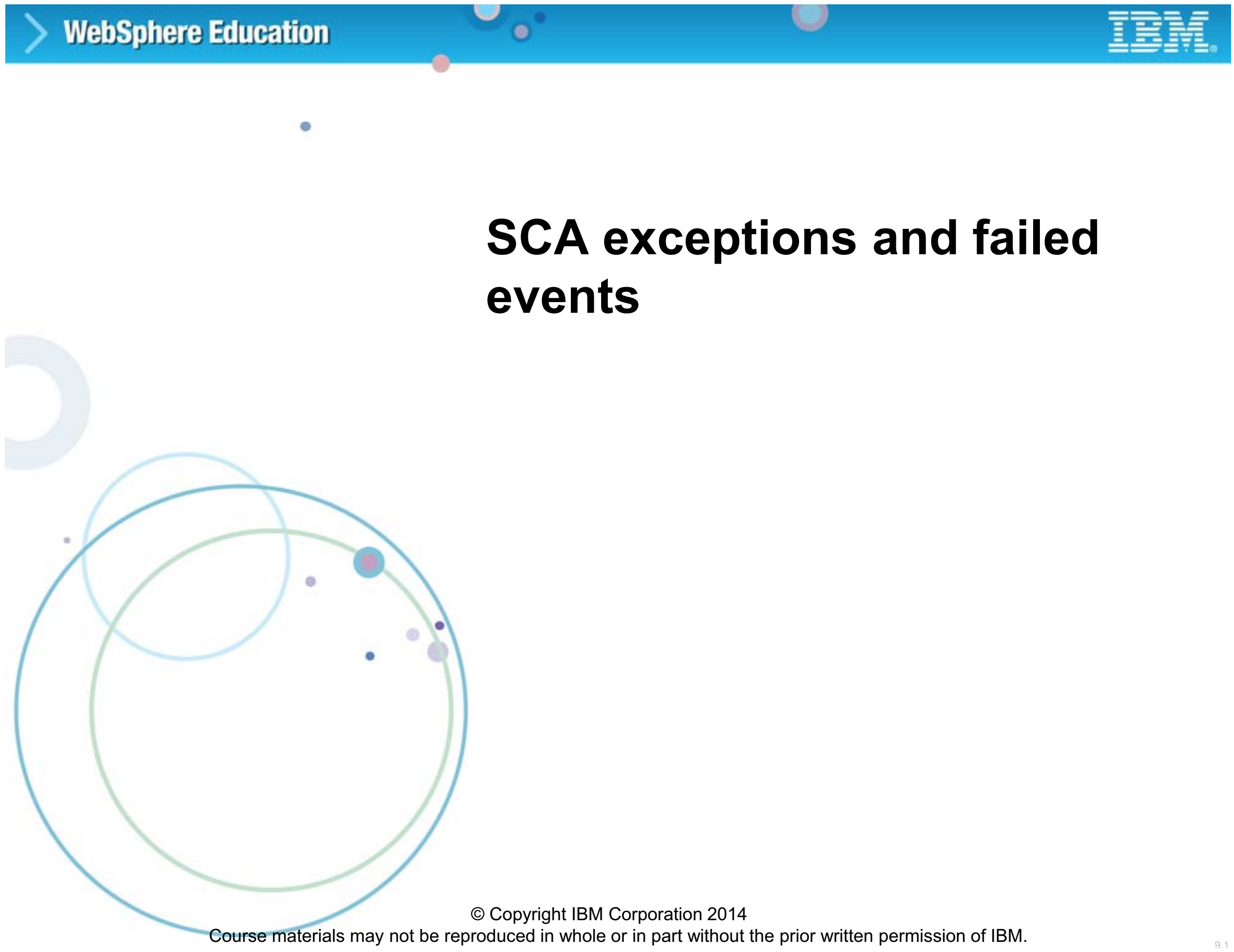


Figure 9-3. SCA exceptions and failed events WB8691.0

Notes:

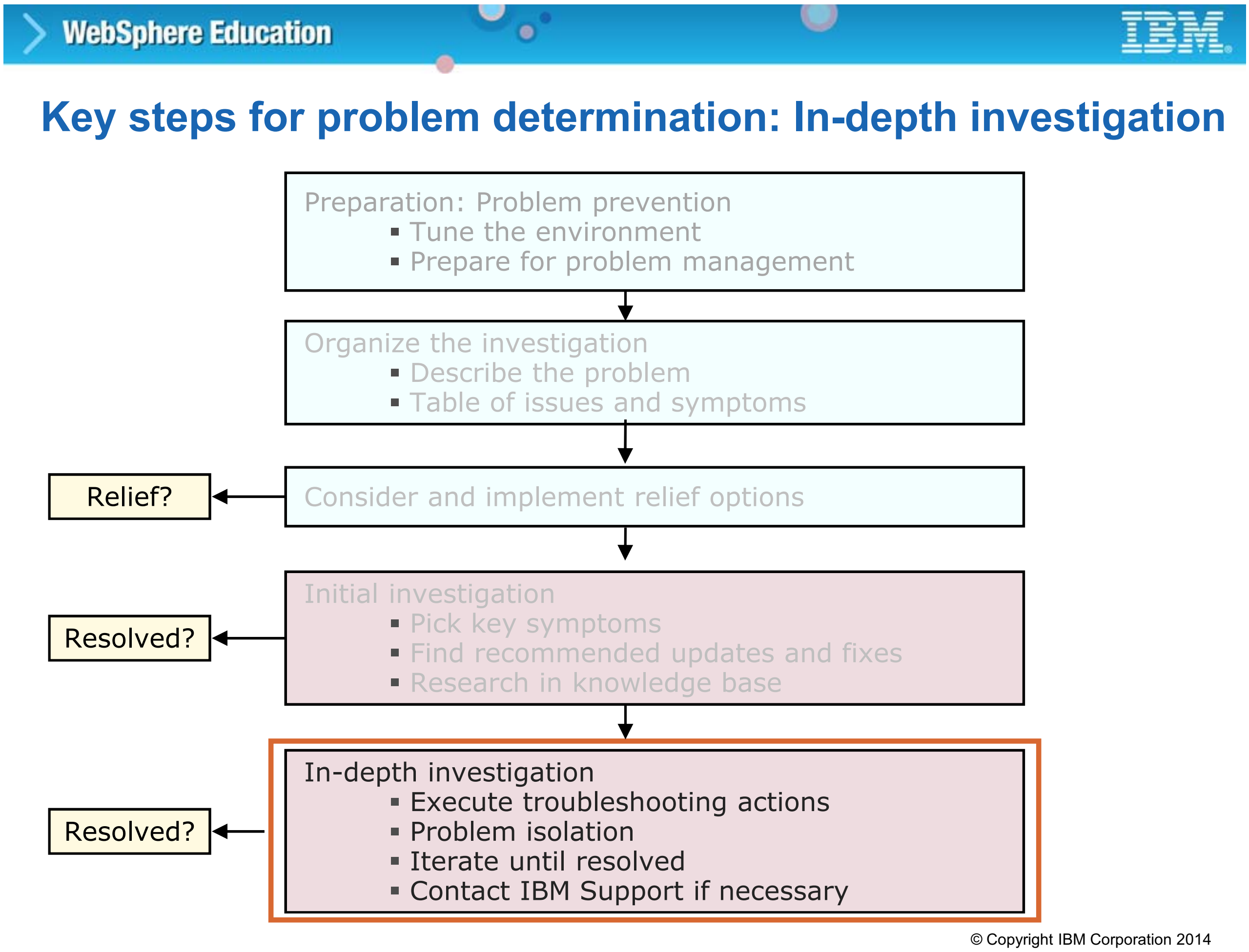


Figure 9-4. Key steps for problem determination: In-depth investigation WB8691.0

Notes:

WebSphere Education

SCA invocation styles

- An SCA component can call a service synchronously or asynchronously
- There are three kinds of asynchronous invocations:
  - **One-way**: No response is expected (“fire and forget”)
  - **Deferred response**: Service requester fetches the response later by using a ticket
  - **Request with callback**: Service provider sends the response back to the service requester when the result becomes available

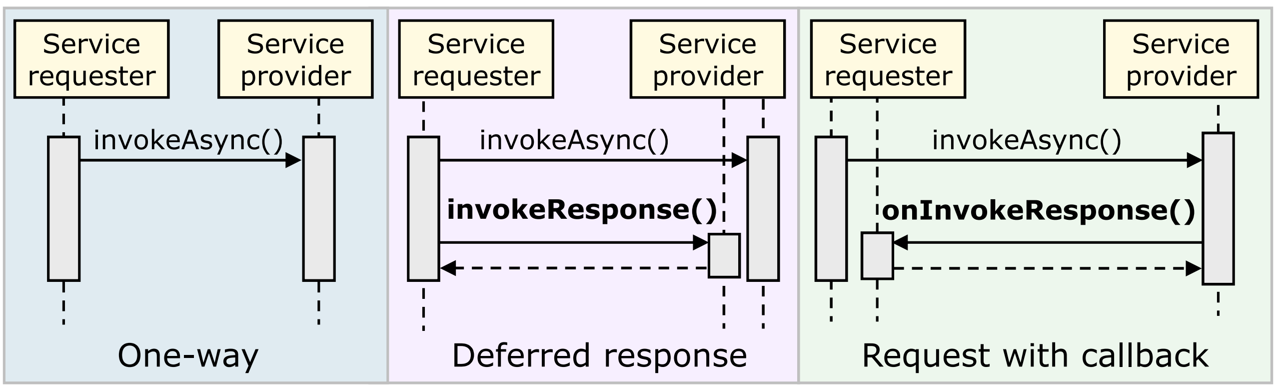


Figure 9-5. SCA invocation styles WB8691.0

Notes:

The **synchronous** method is called the `invoke()` operation, and the **asynchronous** method is called `invokeAsync()`. Determination is normally based on the preferred interaction style in the interface, but it is not a guarantee.

SCA allows services to be called either synchronously or asynchronously. In SCA, there are three types of asynchronous interaction styles available. In all three types of asynchronous invocation, the service requester receives control back immediately from the SCA runtime upon an `invokeAsync()` call.

The service requester can capture the response later in three different ways. First, the service requester can choose to discard the response entirely (or if it is a call to a void method). In this case, the asynchronous invocation is said to be “one-way.” Another option is for the service requester to call `invokeAsync()` and then continue processing until a later time when the service requester makes a request to capture the response. This scenario is termed “deferred response.” Finally, the service requester also has the option of making an asynchronous “request with callback.” To do this option, the service requester needs to first implement the ServiceCallback interface. Then, after calling `invokeAsync()`, the SCA runtime provides a callback to the ServiceCallback handler to provide the response to the service requester.

WebSphere Education

Asynchronous and synchronous invocations in IBM Business Process Manager

- In IBM Business Process Manager, the following components usually have asynchronous implementations:
  - Java POJO components that implement the **`ServiceAsynchImpl`** interface
  - Long-running BPEL and BPMN business processes
  - Human tasks
  - Java Message Service (JMS) imports
  - WebSphere MQ imports
- The following components can take on synchronous or asynchronous behavior that is based on how they are invoked:
  - Interface mediation component
  - Mediation flow component
  - SCA import and export

Figure 9-6. Asynchronous and synchronous invocations in IBM Business Process Manager WB8691.0

Notes:

To ease the service invocation style determination, a preferred interaction style attribute is available at each service interface. This attribute can take on the value of synchronous, asynchronous, or any. Usually, this attribute defaults to a value that matches the corresponding component type implementation. This attribute is configurable for component types that have dual behaviors, or editable for synchronous components to allow asynchronous-over-synchronous invocations.

When invoking a service by using the SCA Java PI, you can query the preferred interaction style of the target service before deciding which invocation style to use for the invocation.

```
public interface Service {
   String getPreferredInteractionStyle(OperationType operationType);
}
```

In most scenarios, a service is not invoked from the user code through Java API, but rather by wiring a component reference to the service component. In this case, how the service gets invoked is determined from the specific implementation of the calling component.

Most of the built-in components in IBM Business Process Manager use the preferred interaction style attributes of the target component. However, the treatment is slightly different among different component type implementations. How various component types interpret this attribute is documented in IBM Business Process Manager invocation styles.

WebSphere Education

Exception types

- You need to understand how IBM Business Process Manager represents declared and undeclared exceptions in order to create a comprehensive error-handling strategy
- The SCA programming model provides two types of exceptions:
  - **Service business exceptions** (business-level exceptions): Represent known exceptions, and exceptions that the application or service anticipates

    Example: An `InvalidSymbolException` for a stock quote service; such exceptions are wrapped in `ServiceBusinessException` and passed back to the client
  - **Service runtime exceptions** (system exceptions): Represent error conditions that the application does not anticipate, such as `NullPointerException` in Java components

Figure 9-7. Exception types WB8691.0

Notes:

WebSphere Education

Exception types: Service business exception

- Declared on the interface
  - Faults for Web Services Description Language (WSDL) interface
  - Throws clause for Java interface
- Service business exceptions that are propagated back to the service client to check and handle
- Unit of work is not set for rollback
- A *failed event* is not created

```
</wsdl:message>
  <wsdl:message name="customerTypeResponseMsg">
  <wsdl:part element="tns:customerTypeResponse" name="customerTypeResult"/>
</wsdl:message>
  <wsdl:message name="customerType_faultMessageMsg">
  <wsdl:part element="tns:customerType_faultMessage" name="faultMessage"/>
</wsdl:message>
  <wsdl:portType name="CustomerTypeInterface">
  <wsdl:operation name="customerType">
    <wsdl:input message="tns:customerTypeRequestMsg" name="customerTypeRequest"/>
    <wsdl:output message="tns:customerTypeResponseMsg" name="customerTypeResponse"/>
    <wsdl:fault message="tns:customerType_faultMessageMsg" name="faultMessage"/>
  </wsdl:operation>
</wsdl:portType>
```


Figure 9-8. Exception types: Service business exception WB8691.0

Notes:

WebSphere Education IBM

Exception types: Service runtime exception (1 of 2)

- Undeclared exceptions that the solution throws whenever a component or runtime encounters something unexpected
- Results in the rollback of the current unit of work
- Service runtime exceptions generally result in creation of a failed event when the client uses an asynchronous invocation pattern
  - **Note:** There are exceptions to this rule when the process is a service client

Figure 9-9. Exception types: Service runtime exception (1 of 2) WB8691.0

Notes:

WebSphere Education

Exception types: Service runtime exception (2 of 2)

Service runtime exceptions have several subclasses:

- **ServiceExpirationRuntimeException**
  – Indicates that an asynchronous SCA message expired
  – Expiration times can be set using the `RequestExpiration` qualifier on a service reference
- **ServiceTimeoutRuntimeException**
  – Indicates that the response to an asynchronous request was not received within the configured period
  – Expiration times can be set using the `ResponseExpiration` qualifier on a service reference
- **ServiceUnavailableException**
  – Indicates that an exception was thrown while invoking an external service through an import
- **ServiceUnwiredReferenceRuntimeException**
  – Indicates that the service reference on the component is not wired correctly

Figure 9-10. Exception types: Service runtime exception (2 of 2) WB8691.0

Notes:

WebSphere Education

When are failed events created? (1 of 2)

Invocation pattern	Exception type that is thrown	Was a failed event created?	Comments
Synchronous	ServiceBusinessException	No	Failed events are not created for a synchronous pattern
Synchronous	ServiceRuntimeException	No	Failed events are not created for a synchronous pattern
Asynchronous one-way	ServiceBusinessException	No	One-way invocation cannot declare faults
Asynchronous one-way	**ServiceRuntimeException**	**Yes**	N/A
Asynchronous deferred response	ServiceBusinessException	No	Failed events are not created for ServiceBusinessException

Figure 9-11. When are failed events created? (1 of 2) WB8691.0

Notes:

WebSphere Education IBM.

When are failed events created? (2 of 2)

Invocation pattern	Exception type that is thrown	Was a failed event created?	Comments
Asynchronous deferred response	**ServiceRuntimeException**	**Yes**	N/A
Asynchronous callback	ServiceBusinessException	No	Failed events are not created for ServiceBusinessException
Asynchronous callback	**ServiceRuntimeException**	**Yes**	N/A
BPEL two-way	ServiceRuntimeException	No	Failed events are not created for BPEL client
BPEL one-way	**ServiceRuntimeException**	**Yes**	N/A

Figure 9-12. When are failed events created? (2 of 2) WB8691.0

Notes:

WebSphere Education

Infrastructure and system considerations

- IBM Business Process Manager provides a solution for resubmitting an event that encountered a **`ServiceRuntimeException`** during asynchronous invocations through the *failed event manager*
- Service integration bus (SIBus) provides a default retry behavior
- SCA bindings serialize and deserialize events that enter and exit the system
  - Events might end up in the system exception destination if an error occurs in the process

Figure 9-13. Infrastructure and system considerations WB8691.0

Notes:

WebSphere Education IBM.

Recovery service at run time

- SCA failed asynchronous interactions are redirected to failed event destinations:
  - Destinations are created on the SCA system bus: **`SCA.SYSTEM.<cellname>.Bus`**
  - One failed event destination per deployment target gets created
  - For single server: **`WBI.FailedEvent.<nodename>.<servername>`** and **`WBI.FailedEvent.<clustername>`** for clustered environment
  - If the client is a BPEL process, a failed event is ***not*** created automatically
- The recovery MDB reads failed events from the failed queue
- Failed events are saved in the failed event database:
  - Default database is **`WPRCSDB`**
  - Three database tables: **`FailedEvents`**, **`FailedEventBOTypes`**, and **`FailedEventMessages`**

Figure 9-14. Recovery service at run time WB8691.0

Notes:

The MDB of the recovery service reads the failed events from this failed event destination and persists them in the failed event schema in the common database (WPRCSDB).

WebSphere Education IBM

System exception destinations (1 of 2)

- The failed event manager uses failed events for end-to-end scenarios, including errors in:
  - SCA asynchronous invocations
  - JMS bindings
  - Business Process Choreographer infrastructure
  - Long-running BPEL processes
  - Failed events in WebSphere Enterprise Service Bus
- JMS and EIS imports and exports leverage the **SCA.Application.<cell name>.Bus** for their integration needs
- Every bus has system exception destination: **SYSTEM.Exception.Destination.<node name>.SCA.APPLICATION.<cell name>.Bus**
- The import and export components place messages into this destination if they encounter an exception

Figure 9-15. System exception destinations (1 of 2) WB8691.0

Notes:

Failed SCA events

In the context of SCA, an event is a request or response that a service application receives. It can come from an external source (such as an inbound application adapter) or an external invocation to a web service. The event comprises a reference to the business logic it wants to operate and its data, which is stored in a Service Data Object (a business object). When an event is received, the appropriate application business logic processes it.

A single thread of execution can branch off into multiple branches (or threads); the individual branches are linked to the main invoking event by the same session context.

If the business logic in one of these branches cannot execute due to system failure, component failure, or component unavailability, the event moves into the failed state. If multiple branches fail, a failed event is created for each. The recovery service handles the following types of failed SCA events:

- Event failures that occur during an asynchronous invocation of an SCA operation
- Event failures that a runtime exception causes (in other words, any exception that is not declared in the methods that the business logic uses)

The recovery service does not handle failures from synchronous invocations.

Failed SCA events typically have source and destination information that is associated with them. The source and destination are based on the failure point (the location where the invocation fails), regardless of the type of interaction. Consider the following example, in which Component A is asynchronously invoking Component B. The request message is sent from A to B, and the response (callback) message is sent from B to A.

- If the exception occurs during the initial request, Component A is the source and Component B is the destination for the purposes of the failed event manager.
- If the exception occurs during the response, Component B is the source and Component A is the destination for the purposes of the failed event manager.

This pattern is true for all asynchronous invocations.

The recovery service sends failed SCA asynchronous interactions to failed event destinations that were created on the deployment environment bus. The data for failed events is stored in the failed event database, which is WPCRSDB by default. The failed event database can be accessed for administrative purposes through the failed event manager interface.

Failed JMS events

The Java Message Service (JMS) binding type and configuration determine whether a failed event is generated and sent to the failed event manager.

JMS bindings

You can use the recovery binding property to enable or disable recovery for each JMS binding at authoring time. The **recoveryMode** property can be set to one of the following options:

- `bindingManaged`: Allows the binding to manage recovery for failed messages
- `unmanaged`: Relies on transport-specific recovery for failed messages

Recovery for JMS bindings is enabled by default. When it is enabled, JMS failed events are created in the following situations:

- The function selector fails.
- The fault selector fails.
- The fault selector returns the `RuntimeException` fault type.
- The fault handler fails.
- The data binding or data handler fails after a single retry attempt in JMS.

In addition, a failed SCA event is created when the `ServiceRuntimeException` exception is thrown in a JMS binding target component after a single retry attempt in JMS.

These failures can occur during inbound or outbound communication. During outbound communication, JMS import sends a request message and receives the response message. A failed event is generated if the JMS import binding detects a problem while processing the service response. During inbound communication, the sequence of events is as follows:

1. JMSExport receives the request message.
2. JMSExport invokes the SCA component.

3. The SCA component returns a response to JMSExport.
4. JMSExport sends a response message.

A failed event is generated if the JMS export binding detects a problem while processing the service request.

The recovery service captures the JMS message and stores it in a recovery table in the common database. It also captures and stores the module name, component name, operation name, failure time, exception detail, and JMS properties of the failed event. You can use the failed event manager to manage failed JMS events, or you can use a custom program.

To disable recovery, you must explicitly disable it by setting the **recoveryMode** property to `unmanaged`.

Note: If the **recoveryMode** property is missing (in earlier versions of applications), the recovery capability is regarded as enabled.

When recovery is disabled, a failed message is rolled back to its original destination and tried again. The system does not create a failed event.

WebSphere MQ JMS bindings and generic JMS bindings

In response to various error conditions, the WebSphere MQ JMS or generic JMS import and export implementation can return one of two types of exceptions:

- Service Business Exception: This exception is returned if the fault specified on the service business interface (WSDL port type) occurs.
- Service Runtime Exception: This exception is raised in all other cases. In most cases, the cause exception contains the original exception (JMSException).

For example, an import expects only one response message for each request message. If more than one response arrives, or if a late response (one for which the SCA response expiration expired) arrives, a Service Runtime Exception is thrown. The transaction is rolled back, and the response message is backed out of the queue or handled by the failed event manager.

If SCA messages that originate through a WebSphere MQ or generic JMS interaction fail, you might expect to find these messages in the failed event manager. If such messages are not appearing in the failed event manager, ensure that the value of the maximum retries property on the underlying listener port is equal to or greater than 1. Setting this value to 1 or more enables interaction with the failed event manager during SCA invocations for the WebSphere MQ or generic JMS bindings.

System exception destinations (2 of 2)

- SCA creates a number of SIBus destinations to support asynchronous communication
  - Required destinations are created during module deployment
- One of the destinations is called "`sca/<module name>`"
- You can adjust the number of retry attempts that happen during a failure of an asynchronous service invocation by changing the value of the **Maximum failed deliveries** property

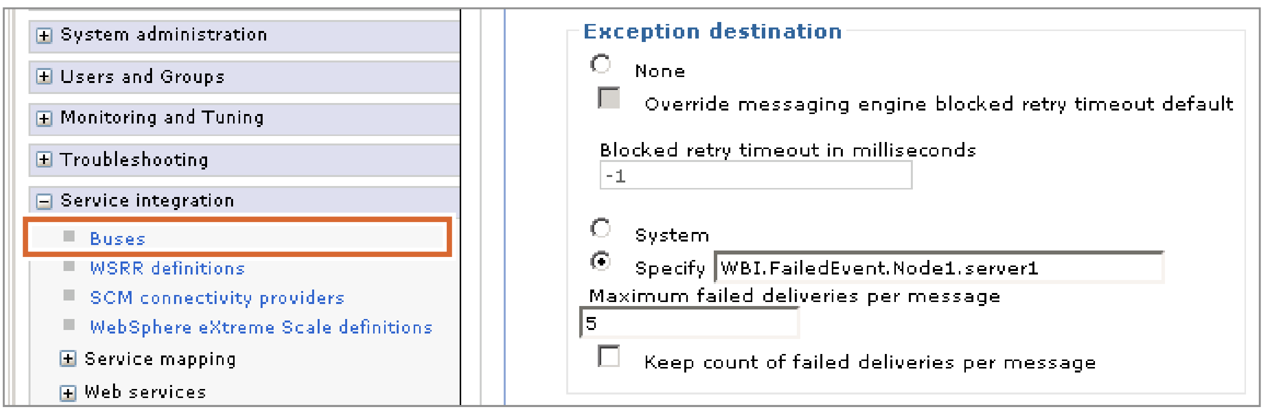


Figure 9-16. System exception destinations (2 of 2) WB8691.0

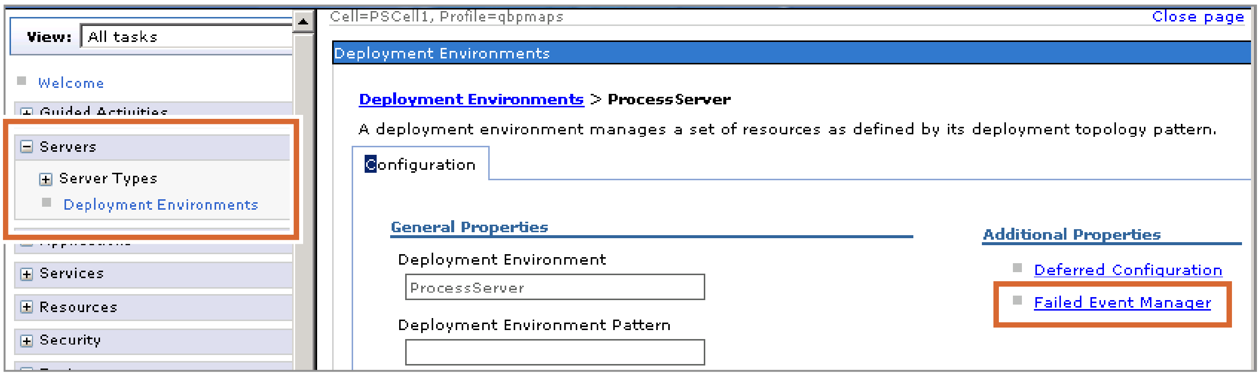
Notes:

Notice that the purpose of this screen capture is to give an example of system exception destination.

WebSphere Education

Failed event manager

- The failed event manager is in the administrative console
  - **Servers > Deployment Environments > *Deployment Environment Name* > Additional Properties > Failed Event Manager**



- The failed event manager handles the following types of events:
  - Service Component Architecture (SCA) failed events
  - Java Message Service (JMS) failed events
  - WebSphere MQ failed events
  - Business Process Choreographer stopped, terminated, and failed events
  - Business Flow Manager hold queue events

Figure 9-17. Failed event manager WB8691.0

Notes:

Here are some descriptions of the events that the failed event manager handles.

- **Service Component Architecture (SCA) failed events**: Failed asynchronous request or response.
- **Java Message Service (JMS) failed events**: Failed data binding or function selector operation.
- **WebSphere MQ failed events**: For WebSphere MQ, a failed event is generated when there is a problem (such as a data-handling exception) in the WebSphere MQ or WebSphere MQ JMS binding export or import that an SCA module uses.
- **Business Process Choreographer stopped, terminated, and failed events**: A failed event is generated when a long-running Business Process Execution Language (BPEL) process fails and one of the following situations occurs:
  - The process instance enters the **Failed** or **Terminated** state.
  - An activity enters the **Stopped** state.
- **Business Flow Manager hold queue events**: A navigation message might be stored in the hold queue if an infrastructure (such as a database) is unavailable, or the message is damaged.

WebSphere Education

Failed event manager tool (1 of 2)

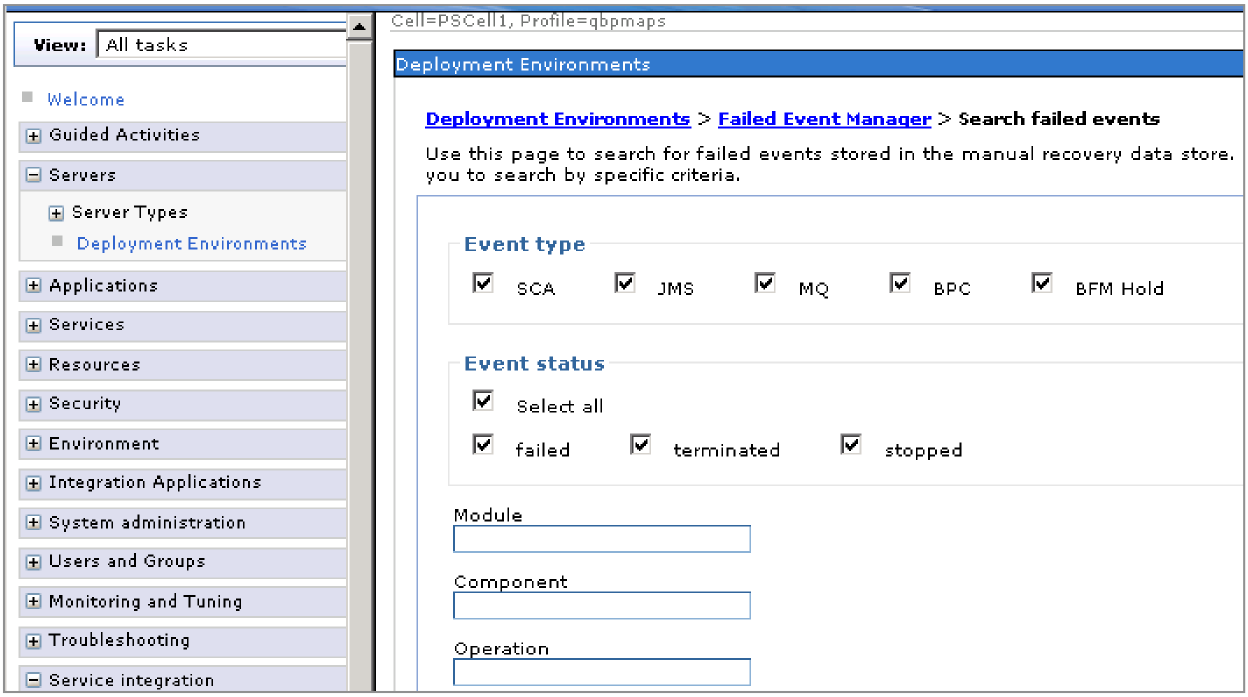


Figure 9-18. Failed event manager tool (1 of 2) WB8691.0

Notes:

Common search criteria:

- Event type
- Event status
- Module name
- Component name
- Operation name
- Failure time

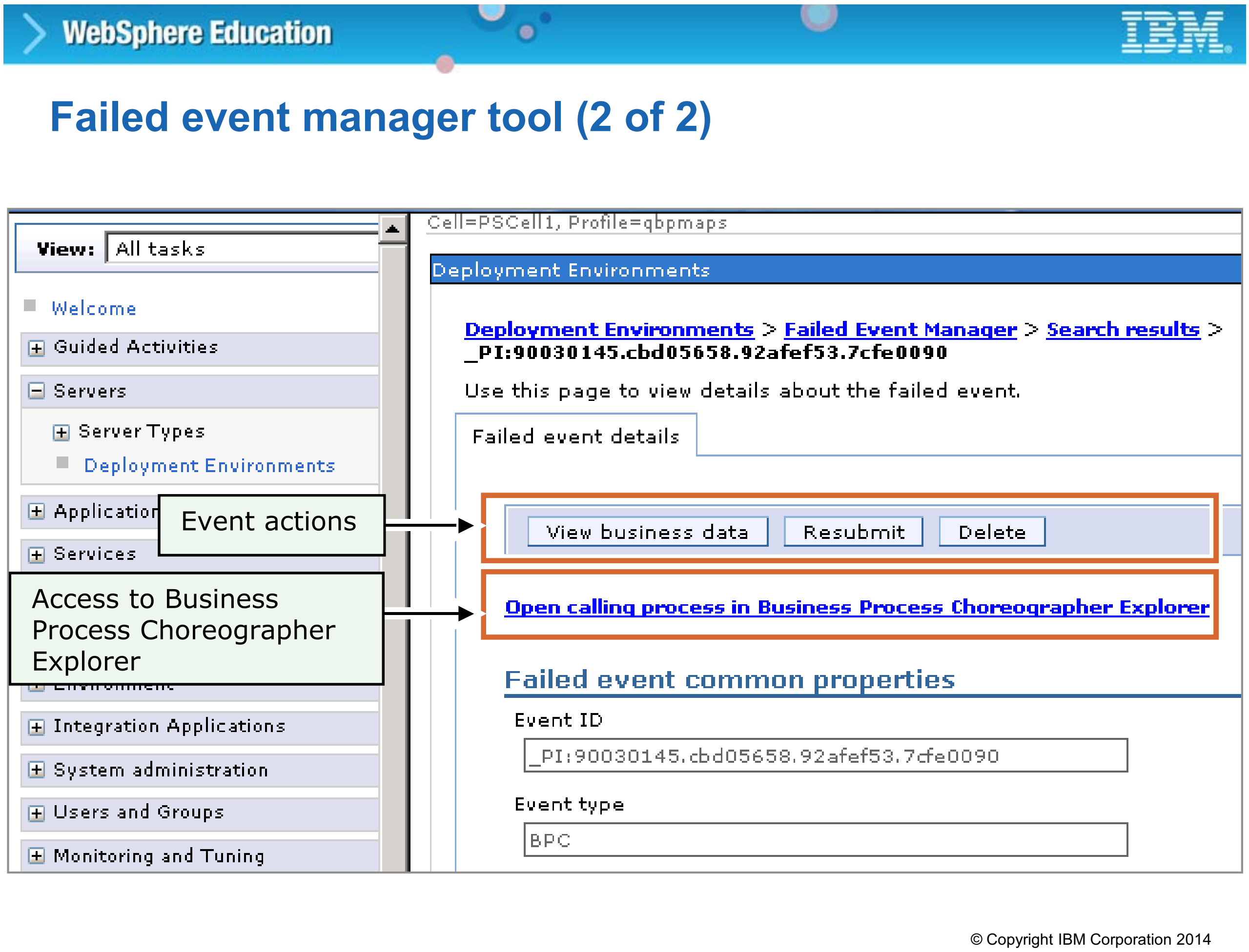


Figure 9-19. Failed event manager tool (2 of 2) WB8691.0

Notes:

Messages in the BFM hold queue are displayed as failed events. You can replay selected messages directly from the failed event manager.

WebSphere Education

Failed event manager: Modifying payload

- Payload data of a failed event can be viewed and modified before resubmission

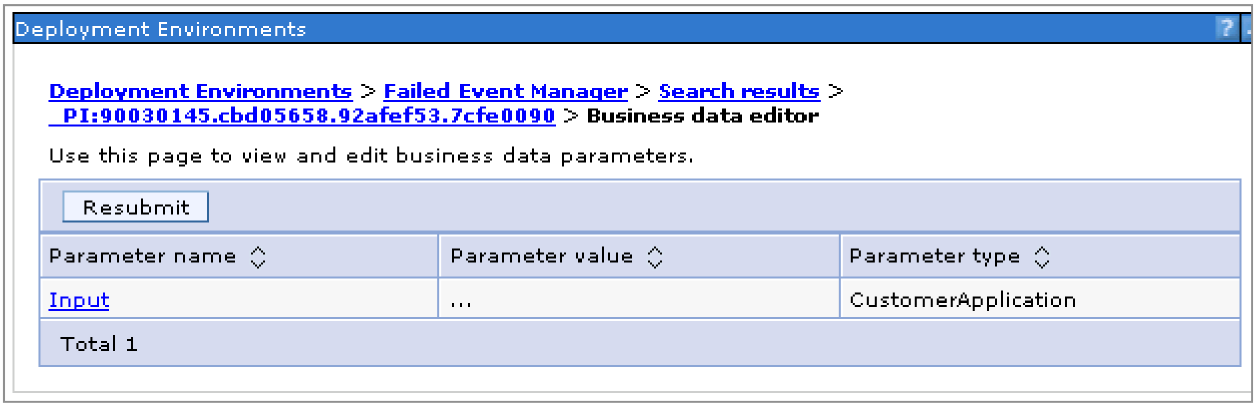


Figure 9-20. Failed event manager: Modifying payload WB8691.0

Notes:

9.2. SCA problem determination tools in IBM Business Process Manager and IBM Integration Designer

WebSphere Process Server and WebSphere Integration Developer provide tools that can be used to troubleshoot runtime exceptions. This topic covers Business Space, the Service Integration Bus browser, the SCA stack trace, the Server Logs view, the `SystemOut.log` file, first-failure data capture (FFDC), and the Debug perspective.

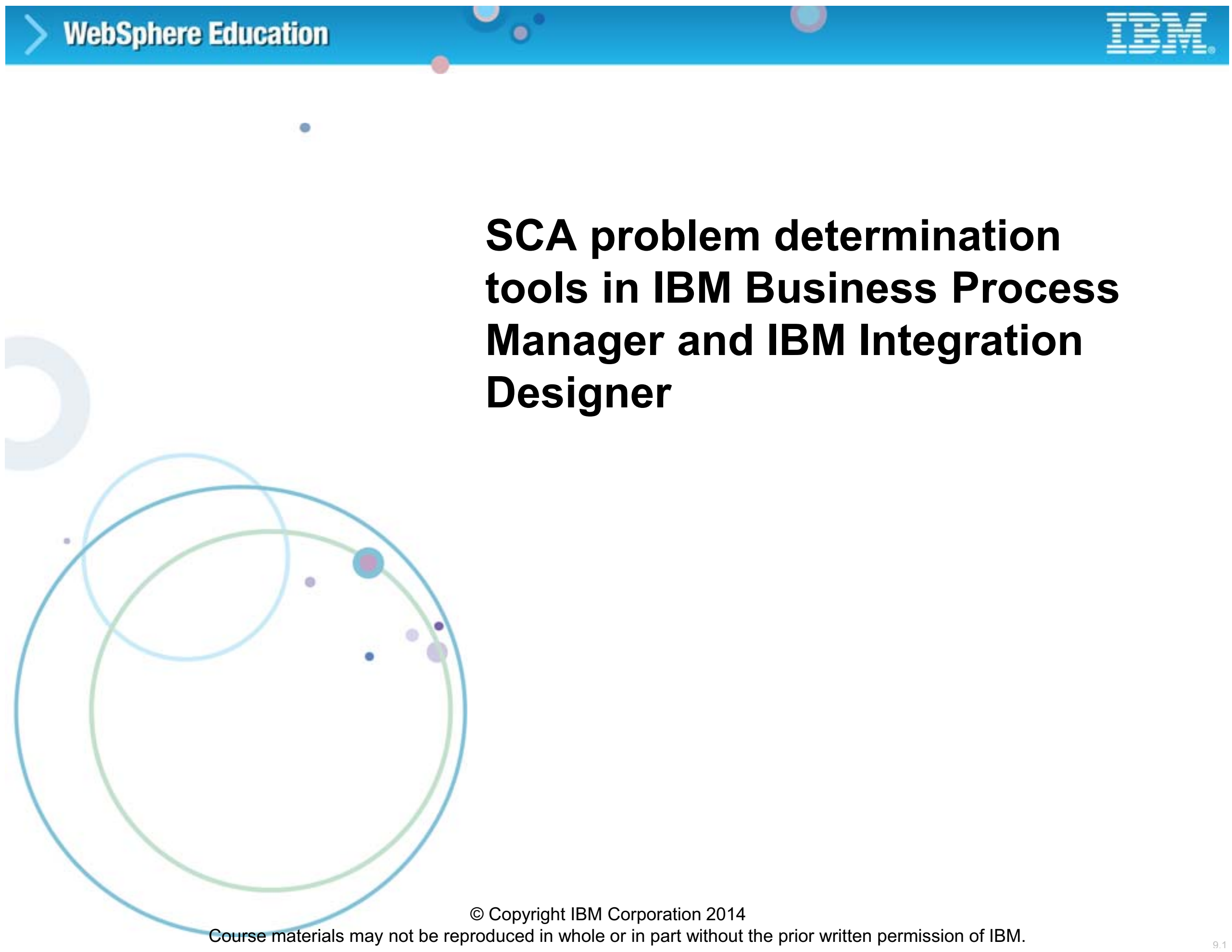


Figure 9-21. SCA problem determination tools in IBM Business Process Manager and IBM Integration Designer WB8691.0

Notes:

Determine where the message is hanging (SIBus)

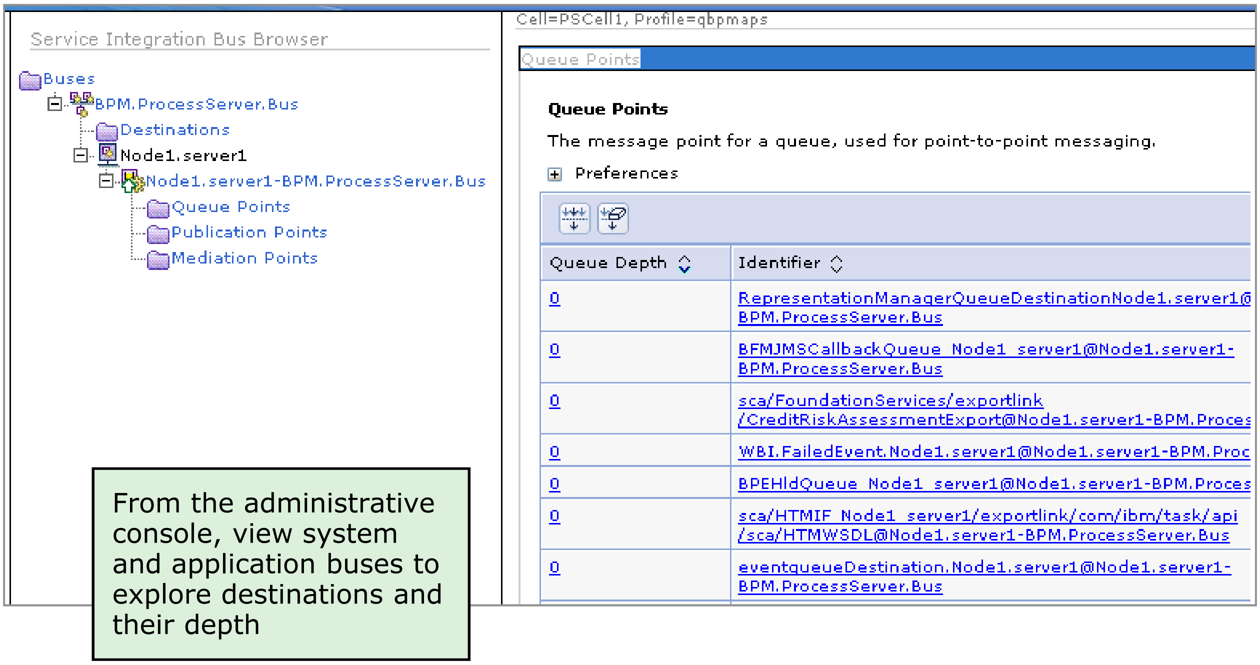


Figure 9-22. Determine where the message is hanging (SIBus) WB8691.0

Notes:

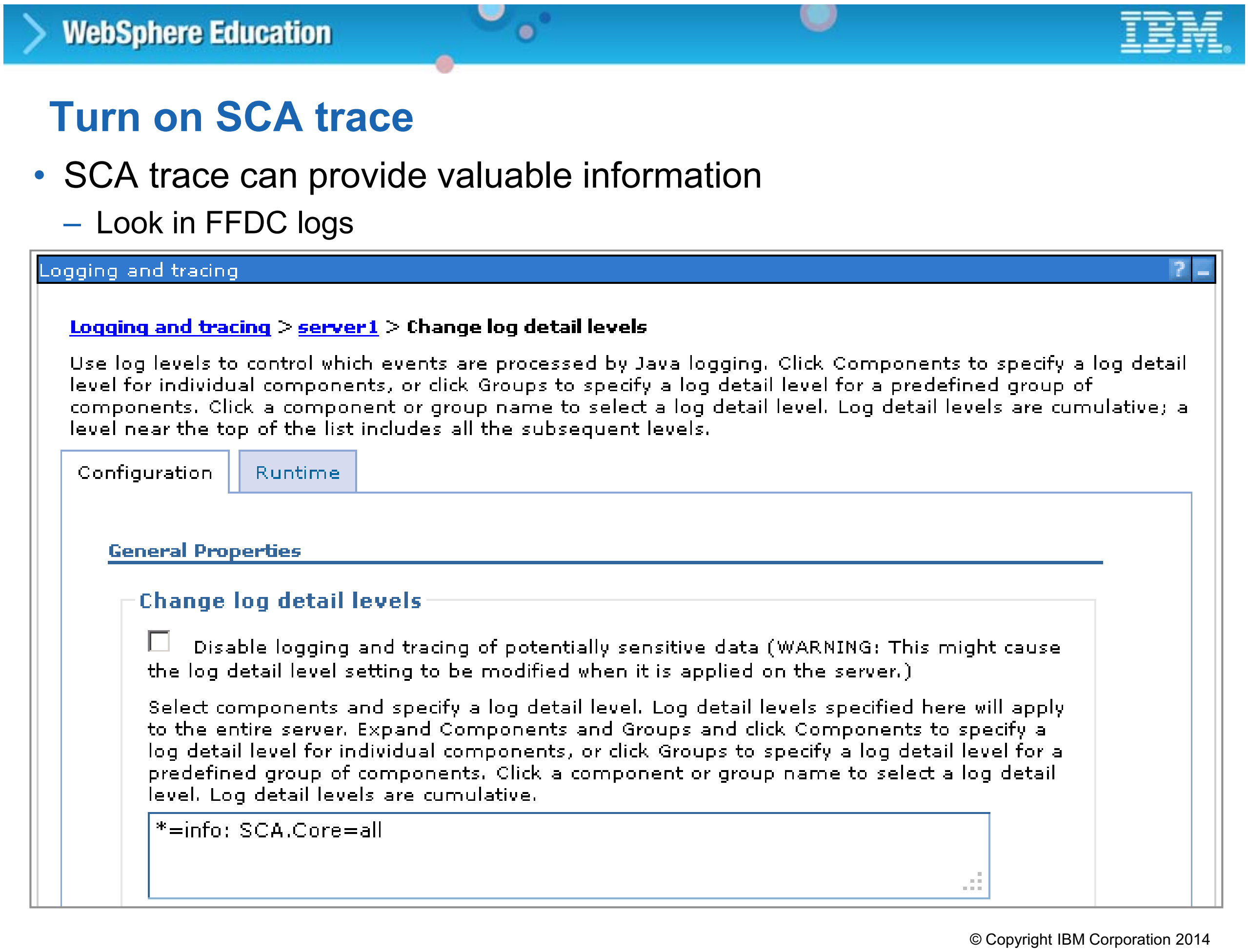


Figure 9-23. Turn on SCA trace WB8691.0

Notes:

You can specify trace settings for all SCA classes.

Examples include: `SCA.*=all`, `SCA.*=fine`

- `SCA.*=all`: This setting is the most verbose setting. It provides all SCA method entries and exits.
- `SCA.*=fine`: This setting provides high-level information that pertains to component-to-component interactions and transaction contexts.

WebSphere Education

Reading the SCA stack trace

- The *SCA context* is provided in the stack trace
  – This context provides important information about what was executing as a part of the SCA invocation path

```
SCA context:
Interaction: [invoke,getCreditScore]
  {CreditAgencySCAModule}CreditAgencyExport.export =>
  {CreditAgencySCAModule}CreditAgency.component
Handler: com/ibm/ws/sca/internal/java/handler/JavaImplementationHandler
  (com.ibm.ws.sca.internal.java.handler.JavaImplementationHandler,
  com/ibm/wsspi/sca/extensions/ImplementationGroup2)
```

Note: Cross-component trace must be enabled

  – The interaction section is used to tell the developer where this exception was encountered in relation to the invocation
  – The exception was encountered in the **`JavaImplementationHandler`**, which is in the **`ImplementationGroup2`**

Figure 9-24. Reading the SCA stack trace WB8691.0

Notes:

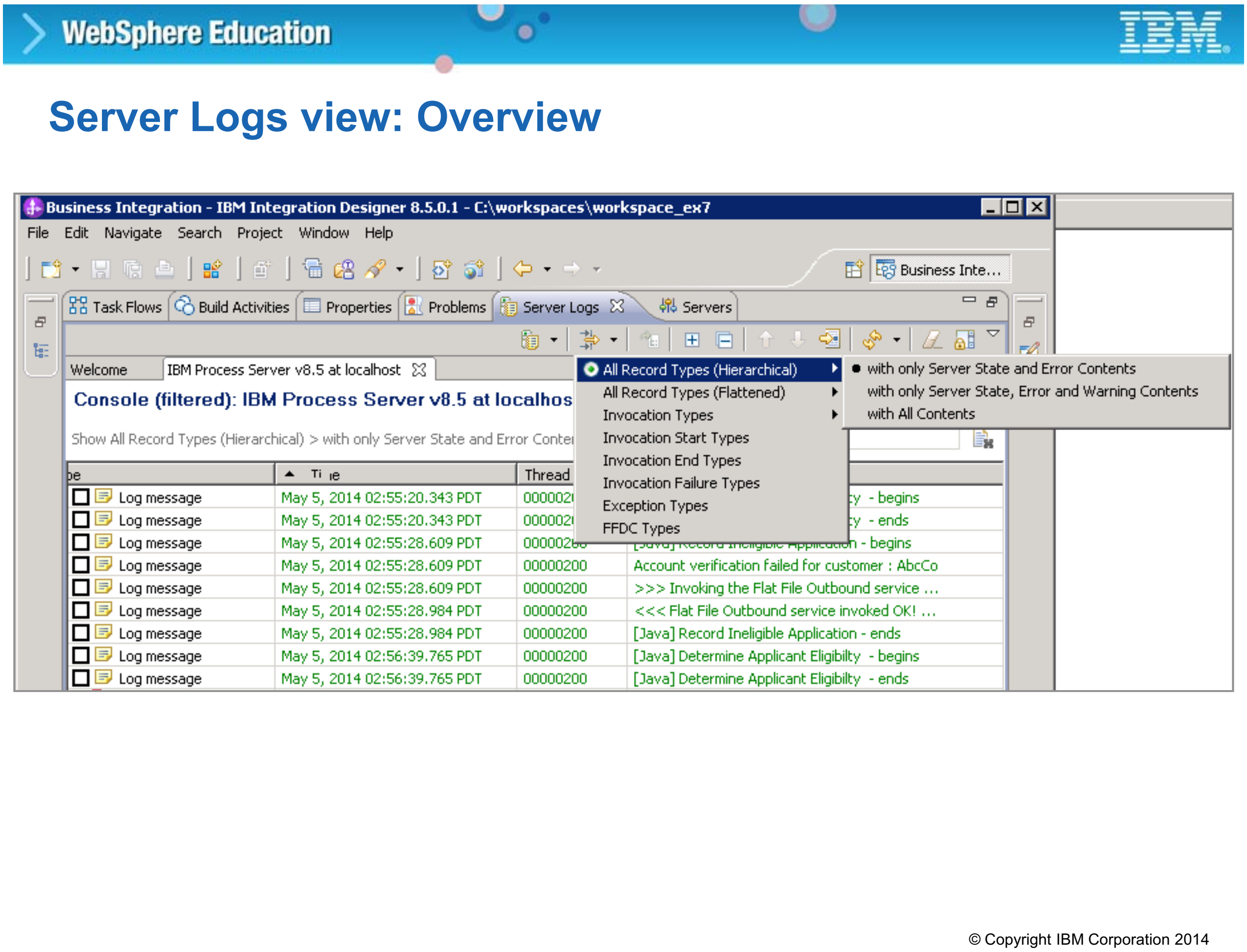


Figure 9-25. Server Logs view: Overview WB8691.0

Notes:

WebSphere Education

What does SystemOut.log provide?

- `SystemOut.log` provides:
  - Useful data such as software levels for the WebSphere Application Server binary files and the IBM Business Process Manager binary files
  - A mix of informational, warning, and error messages
- Data that is collected from SCA processing is added to `SystemOut.log`
- Runtime message identifiers consist of:
  1. A four-character or five-character message prefix
  2. Followed by a four-character or five-character message number
  3. Followed by a single-letter message type code
  - For example: `TRAS0017I`

Figure 9-26. What does SystemOut.log provide? WB8691.0

Notes:

The message type code describes the severity of the error message.

- **C**: Indicates a severe message
- **E**: Indicates an urgent message
- **I**: Indicates an informational message
- **N**: Indicates an error message
- **W**: Indicates a warning message

WebSphere Education IBM

Message codes for SystemOut.log and SystemErr.log

Code	Type
A	Audit
I	Informational
W	Warning
E	Error
F	Fatal
R	`SystemErr` message that is written by an application or by an IBM component
O	`SystemOut` message that is written by an application or by an IBM component
C	Configuration
D	Detail

Figure 9-27. Message codes for SystemOut.log and SystemErr.log WB8691.0

Notes:

WebSphere Education

First-failure data capture (FFDC) log file

- First-failure data capture (FFDC)
  - General term that is applied to the set of diagnostic information that the server automatically captures when errors occur
  - IBM Business Process Manager supports FFDC
- The FFDC log file saves information that is generated from a processing failure (for example, a Java exception)
  - Captured data is saved in log files for use in analysis
  - An index file that references all of the exceptions that FFDC logs
  - An exception file for each exception type from each probe
  - Capturing FFDC data does not affect performance
- FFDC configuration properties files are in the properties directory under the IBM Business Process Manager product installation
  - **`ffdcRun.properties`**
  - **`ffdcStart.properties`**
  - **`ffdcStop.properties`**

Figure 9-28. First-failure data capture (FFDC) log file WB8691.0

Notes:

Set the `ExceptionFileMaximumAge` property to the same value in all three configuration properties files to configure the number of days between purging the FFDC log files.

WebSphere Education

FFDC outputs: Index file

- FFDC **index file** references all of the exceptions that FFDC logs
  - This file provides specific details on one of the exceptions that were logged in the FFDC exception log file. The contents usually includes a stack trace along with other parameters related to the exception.
  - The index file uses a naming convention of `<server name>_exception.log` and has the following columns:

Index	• Number of rows in the table • A plus sign (+) in front of this value signifies that this value was updated since the last persistence of the information to the file system
Occurrences	Number of times that the exception occurred
Time of last occurrence	A time stamp of the Java exception class that the FFDC tool captured
Exception	The name of the Java exception class that the FFDC tool captured
SourceId	The unique identifier for the exception source
ProbeId	The unique identifier for the probe that is used in the data capture

Figure 9-29. FFDC outputs: Index file WB8691.0

Notes:

WebSphere Education

FFDC outputs: Exception file

- FFDC tool generates a single **exception file** for each exception type from each probe
  - The exception file uses a naming convention of **`<`*`server name`*`>_<thread ID>_yy.MM.dd_HH.mm.ss_<`*`unique ID`*`>.txt`** and contains information that is relative to the value of the **`ffdcRun.properties`** **Level** property value

Figure 9-30. FFDC outputs: Exception file WB8691.0

Notes:

WebSphere Education IBM

Using the FFDC for problem determination

- Information might be written to the FFDC files, but **it does not always indicate that there is a problem**
  - Avoid jumping to conclusions when interpreting these files
  - An FFDC entry might not mean that a failure occurred
  - Some FFDC entries are generated for diagnostic purposes only
  - The target audience for FFDC information is IBM support
- Always check **`SystemOut.log`** and **`SystemErr.log`** for information about a problem before searching the FFDC files

Figure 9-31. Using the FFDC for problem determination WB8691.0

Notes:

FFDC logs are a part of the general IBM Business Process Manager ***MustGather*** diagnostic data.

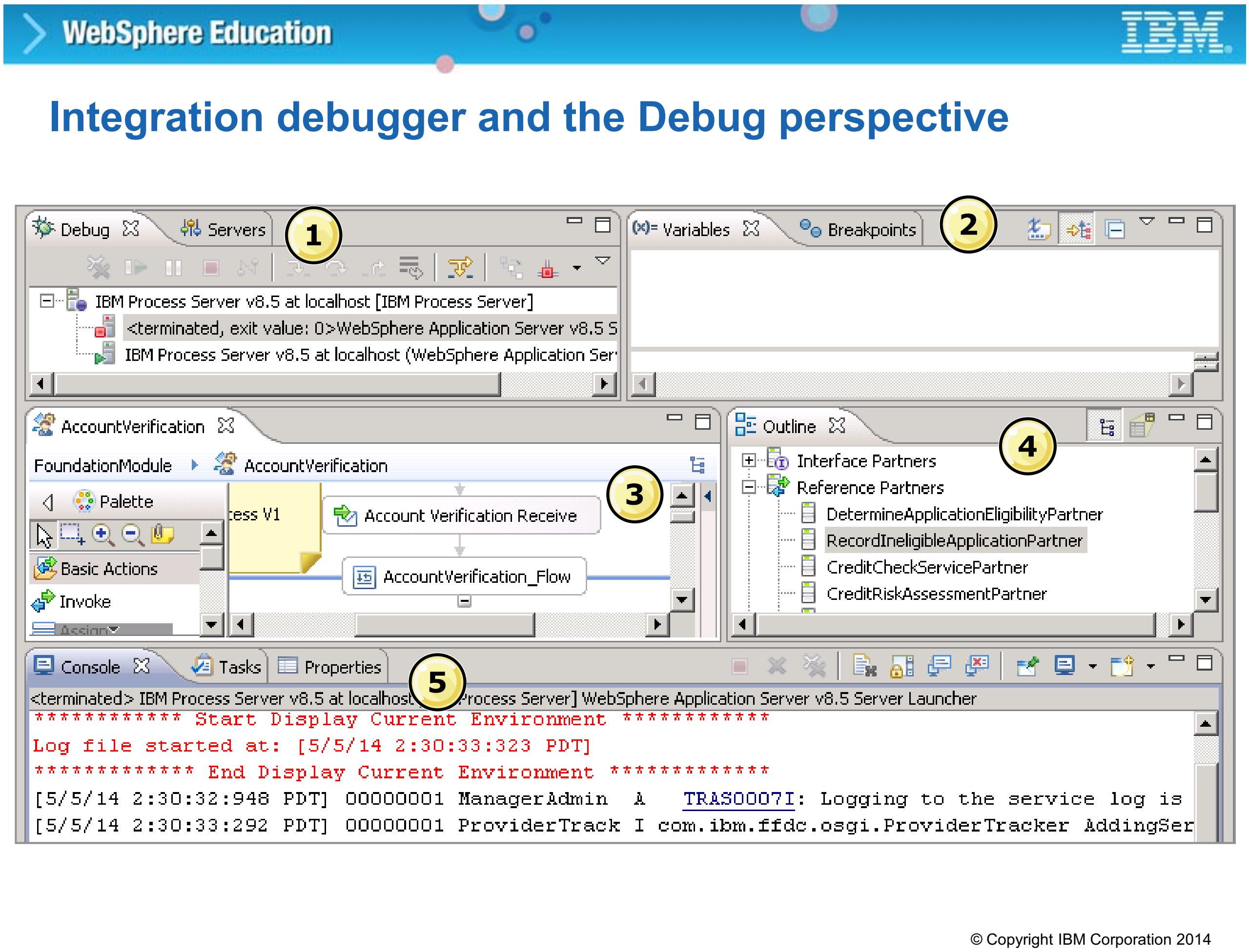


Figure 9-32. Integration debugger and the Debug perspective WB8691.0

Notes:

The integration debugger is primarily controlled from the Debug perspective, and it consists of the following views:

1. Debug, Servers
2. Variables, Breakpoints
3. Component editors
4. Outline
5. Console, Tasks, Properties

WebSphere Education

Using the integration debugger for problem determination

- In IBM Integration Designer, the integration debugger is the designated tool for debugging business integration components
- No special deployment:
  - The integration debugger is installed with IBM Integration Designer
  - Debugging is always available
  - Start your embedded Process Server or WebSphere Enterprise Service Bus runtime in debug mode
- The test client can automatically:
  - Start the debugger and create a component instance for debugging
  - Deploy modules and components to the server, start the server in Debug mode, and create a component instance for debugging
- You can open multiple windows of the test client and use each one to debug a separate component instance

Figure 9-33. Using the integration debugger for problem determination WB8691.0

Notes:

9.3. Cross-component trace

This topic discusses how cross-component trace helps troubleshoot an application error. The steps for enabling the cross-component trace and for viewing cross-component trace output are covered.

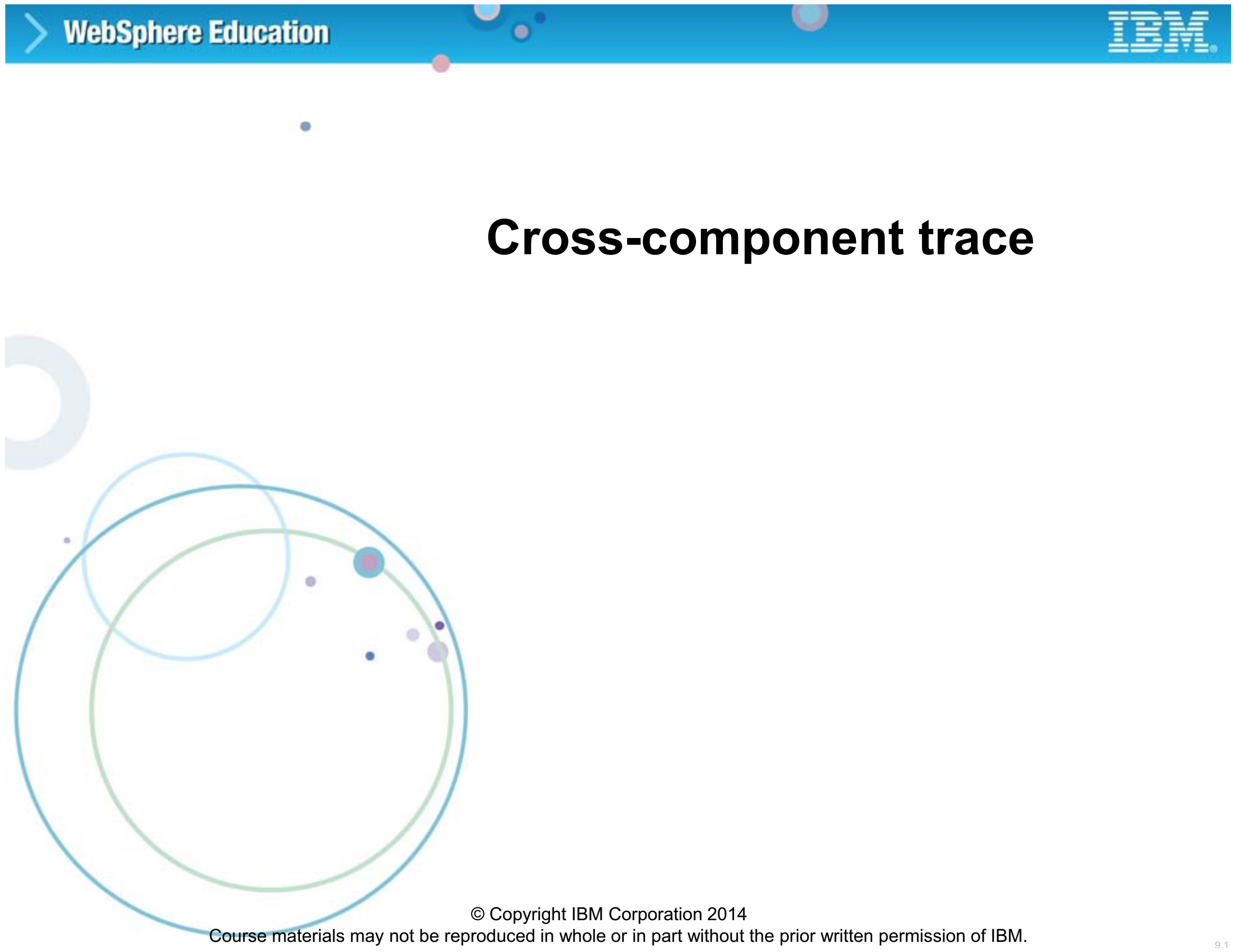


Figure 9-34. Cross-component trace WB8691.0

Notes:

WebSphere Education IBM

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 “**receive**” and **“receive choice”** 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 9-35. Use cross-component trace for problem determination WB8691.0

Notes:

Enabling cross-component trace in testing

- To enable cross-component trace when testing in IBM Integration Designer, start the server.
- Then:
  1. Open the **Server Logs** view
  2. Select the **View menu**
  3. Select **Cross-Component Trace State > *[set trace state]***

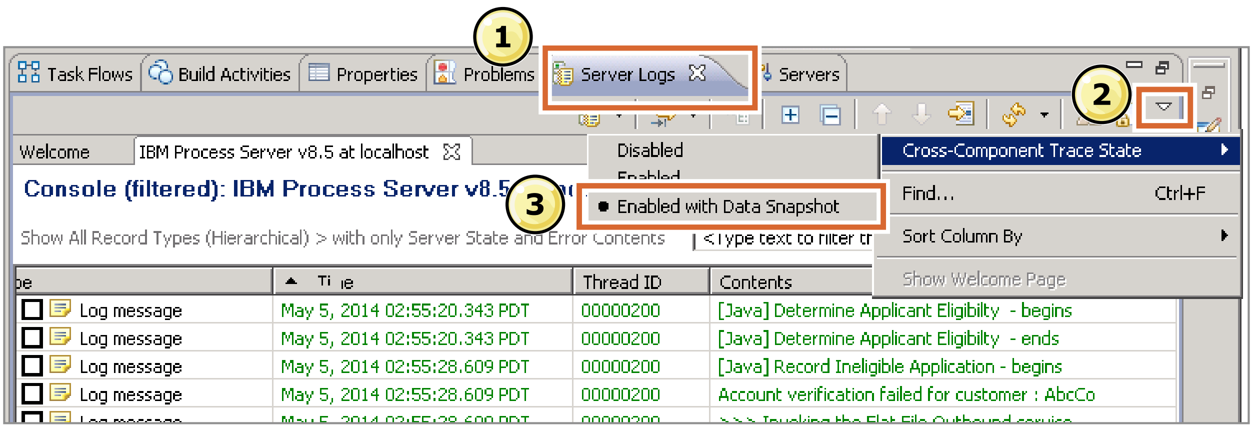


Figure 9-36. Enabling cross-component trace in testing WB8691.0

Notes:

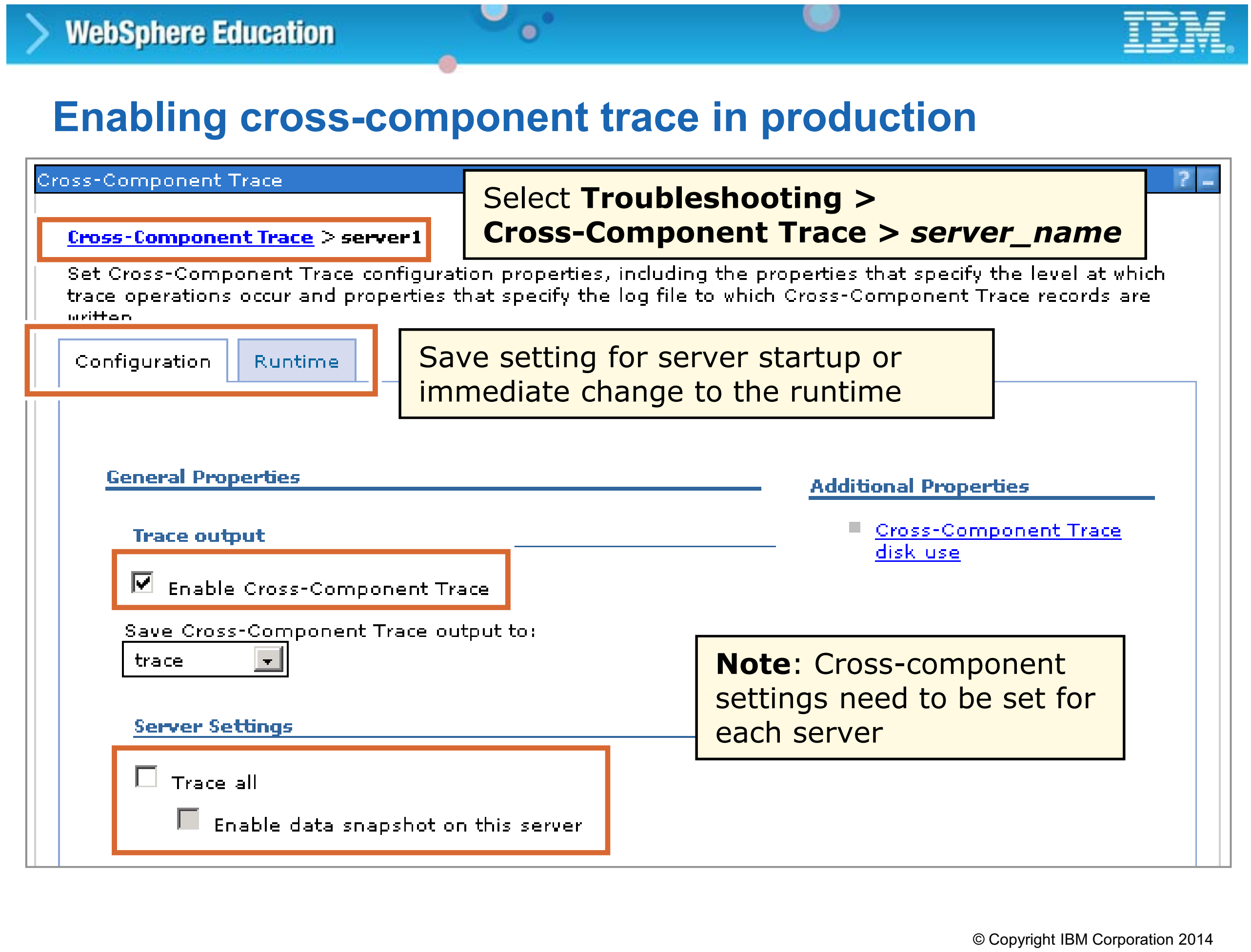


Figure 9-37. Enabling cross-component trace in production WB8691.0

Notes:

WebSphere Education

Trace records in detail

- Cross-component trace (XCT) table displays four pieces of information on trace element record
  - Type of trace, time-stamp, thread ID, and the trace or log message
- Properties dialog shows the trace entry in detail

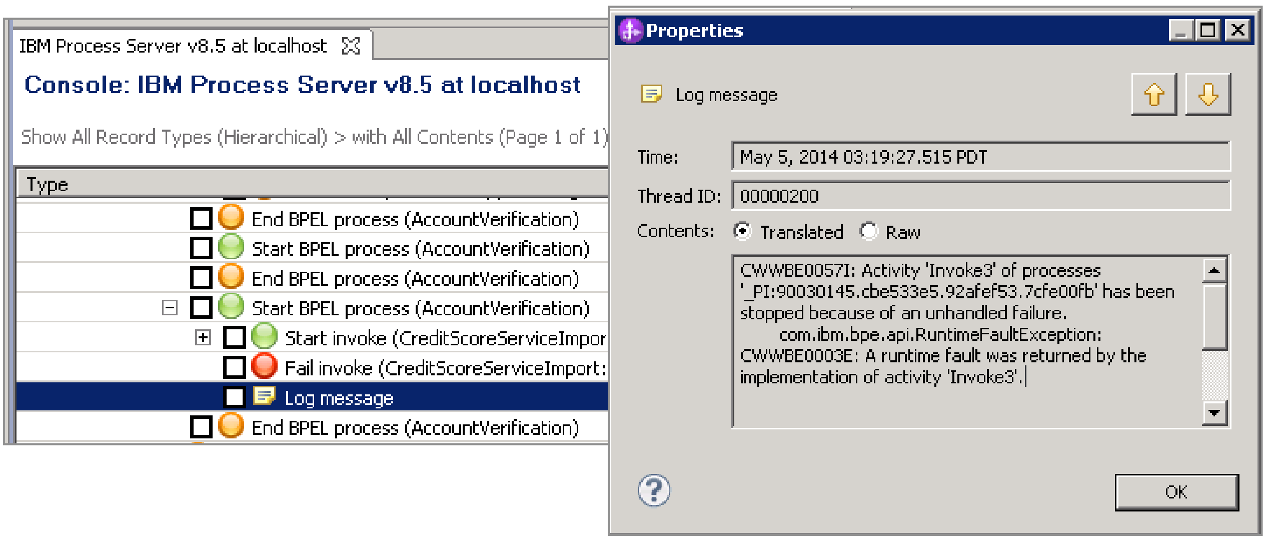


Figure 9-38. Trace records in detail WB8691.0

Notes:

9.4. Troubleshooting event sequencing in SCA applications

This topic covers typical event sequencing problems and how to resolve them.

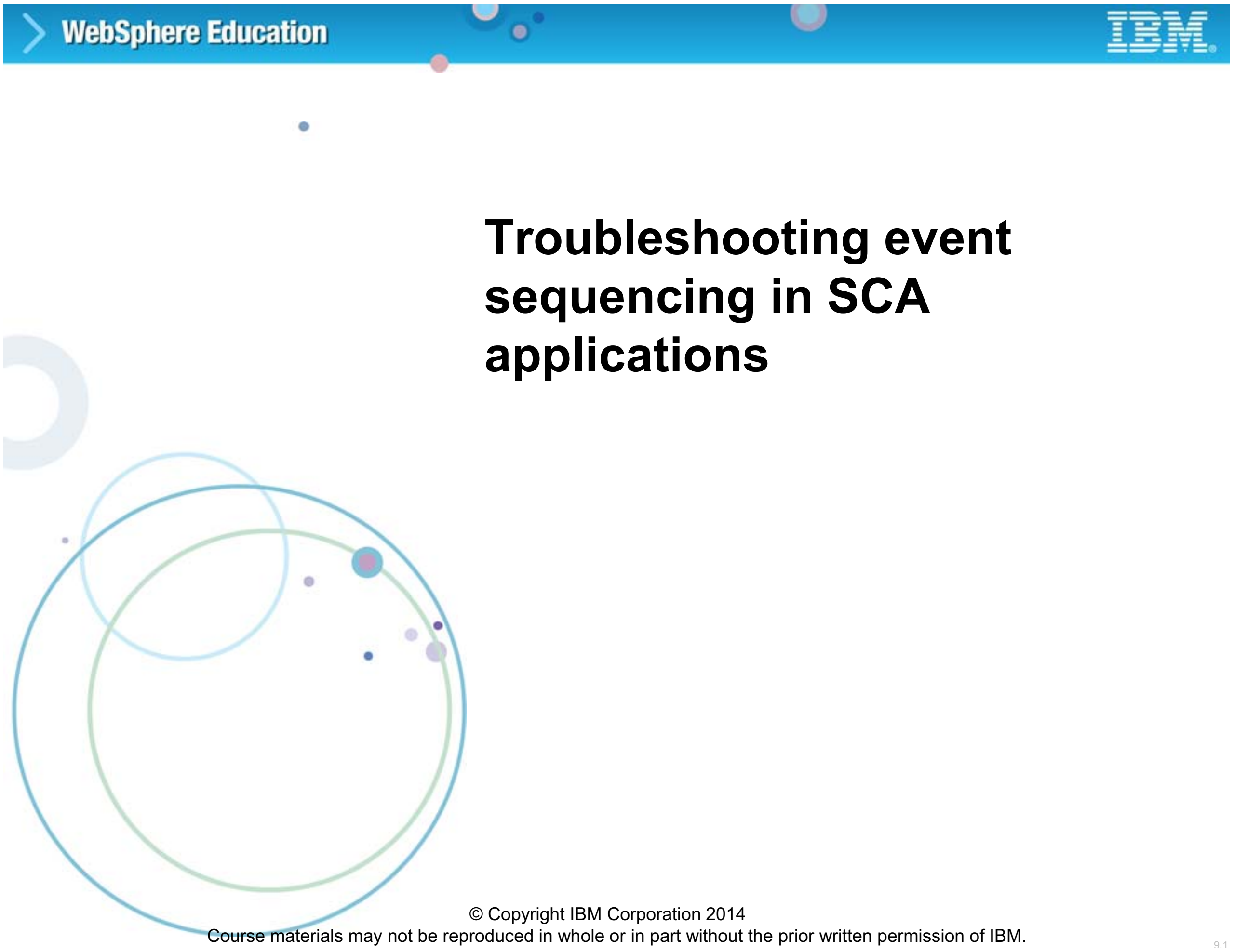


Figure 9-39. Troubleshooting event sequencing in SCA applications WB8691.0

Notes:

WebSphere Education IBM

Event sequencing overview

- Inbound events can be sequenced
  - Ensures that events are processed in the correct order: for example, **withdraw** money can happen only after **deposit**
  - For components that are invoked by using the SCA asynchronous invocation style
- User can choose keys for events to be sequenced
- Supports event sequencing generically for all components
- JMS, MQ/JMS, and WebSphere MQ bindings support the `eventSequencingRequired` property to simplify configuration
- Event sequencing qualifier annotation on a component's interface

Figure 9-40. Event sequencing overview WB8691.0

Notes:

WebSphere Education

Event sequencing (ES) at run time

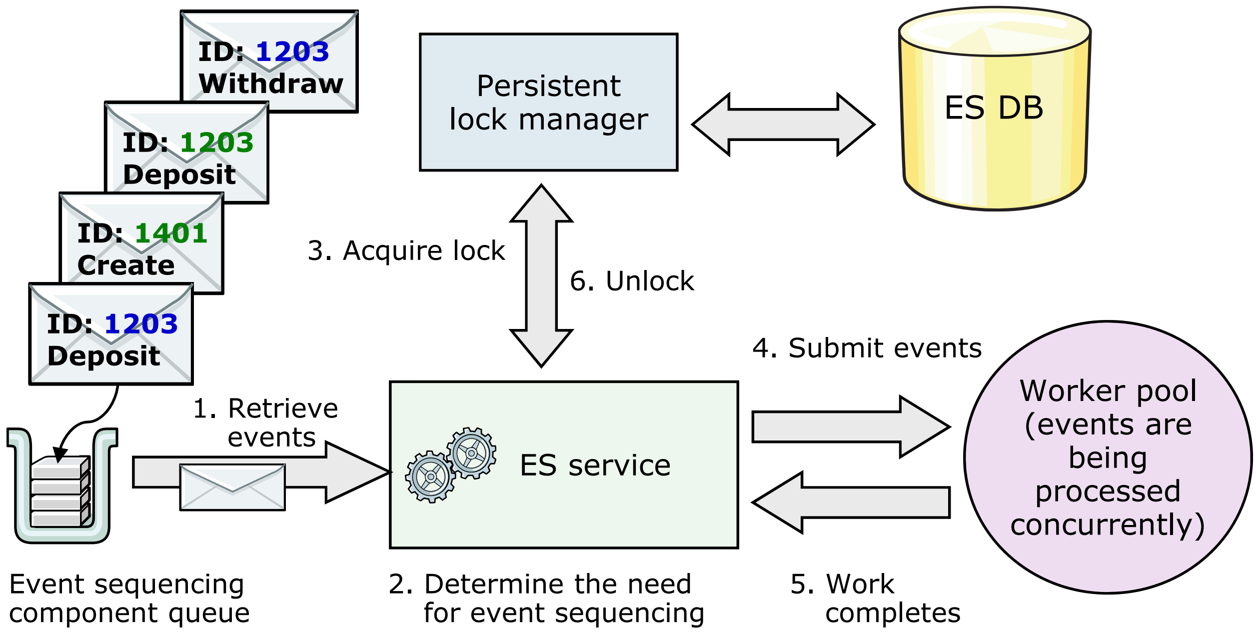


Figure 9-41. Event sequencing (ES) at run time WB8691.0

Notes:

WebSphere Education

Failed sequenced events

- Processing errors or unavailable resources can cause a sequenced event to fail
- How any remaining events in the sequence are handled is determined by whether the **Process requests when error encountered** attribute of the event sequencing qualifier is selected

Selected

– Select this option if you want the processing of the sequence of events to ignore the failure and proceed to process the next event in the sequence

Not selected

– Do not select this option if you want to halt the processing of dependent events in the sequence until the failure is resolved

Figure 9-42. Failed sequenced events WB8691.0

Notes:

WebSphere Education IBM

Troubleshooting event sequencing

Ensure that your component definition is correct

- Is the event sequencing qualifier set on the method?
  - Event sequencing validation fails if the qualifier is erroneously set on the interface
- Is the parameter name valid?
- Is the XPath element valid, and does it correctly resolve to a primitive?
- Is there a single **`eventSequencing`** element for the method?
  - Each method supports only one **`eventSequencing`** element
- Is there a single **`keySpecification`** element for the method?
  - Each method supports only one **`keySpecification`** element

Figure 9-43. Troubleshooting event sequencing WB8691.0

Notes:

WebSphere Education

Troubleshooting event sequencing deadlock

- Deadlocks can occur when you use event sequencing with Business Process Execution Language (BPEL) processes
- Caused by setting event sequencing qualifiers on operations that correspond to both of the following activities:
  - Multiple instantiating **receive** or **receive choice** activities, where the `createInstance` attribute is set to **yes**
  - Correlation set specifications with an initiation attribute set to **join**
- **Problem**: When you try to delete a failed sequenced event in the recovery subsystem, the event sequencing callback can fail to release the event's lock
  - **Reason**: Typically occurs when a target application was removed or when other components of the system such as the database are unavailable
  - **Resolution**: Use the `esAdmin` command to manually delete the lock that is associated with the failed event

Figure 9-44. Troubleshooting event sequencing deadlock WB8691.0

Notes:

WebSphere Education IBM

Command-line utility: esAdmin

- The **`esAdmin`** command-line utility can be used to list locks, delete locks, and break deadlocks
  - List command: Lists lock table entries for given criteria
  - Delete command: Deletes entries from the lock table; server should not be running
  - Unlock command: Deletes the current entry; grants lock to the next queued event

```
$WAS/bin/esAdmin.sh[bat] -help
     Usage: esAdmin [-h hostname] [-p soapPortNumber]
     method [parameters]
     Available methods are :
      listAll
      listLocks moduleName
      listLocks moduleName componentName
      listLocks moduleName componentName methodName
      deleteLocks moduleName
      unlock lockId
```


Figure 9-45. Command-line utility: esAdmin WB8691.0

Notes:

If security is enabled, you must supply a user ID and its associated password with sufficient authority to perform the changes. Enter the user ID and password by using the `-username` and `-password` parameters.

Required parameters

- `hostName`: This parameter specifies the name of the server where the lock manager is running. The value must be a string. If no value is given, the default value `localhost` is used.
- `soapPortNumber`: This parameter specifies the port that is used for the connection to the server. The value must be an integer. If no value is given, the default value `8880` is used.
- `username`: This parameter is required if administrative security is enabled, and it specifies the user ID of a user who has sufficient authority to process the changes. If no value is given and security is enabled, you are prompted to provide a user ID and password.
- `password`: This parameter is required if administrative security is enabled, and it specifies the password that is associated with the user ID that is used in the `username` variable. If no value is given and security is enabled, you are prompted to provide a user ID and password.

- `moduleName`: This parameter specifies the name of the module that contains the component that uses event sequencing. Note: The `esAdmin` command searches for the module under `"Application_Name" + "App"`. This naming convention is a restriction that SCA imposes.
- `componentName`: This parameter specifies the name of the component that uses event sequencing.
- `methodName`: This parameter specifies the name of the method on which event sequencing qualifiers are set.
- `lockId`: This parameter specifies the numeric ID of the lock you want to release. The value for this parameter must be an integer.

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Describe the exception types from the SCA programming model: service business exceptions and service runtime exceptions
- Describe how cross-component trace works in IBM Integration Designer
- Describe first-failure data capture (FFDC)
- Describe how to troubleshoot event sequencing

Figure 9-46. Unit summary WB8691.0

Notes:

WebSphere Education

Checkpoint questions

1. True of False: If a deadlock problem occurs with event sequencing, you can resolve the deadlock by using the **`esAdmin`** command to list and release the current lock.

2. What does cross-component trace do?
   a) Logs records of JMS, HTTP, and web services binding call chains
   b) Provides more tracing information on non-SCA applications
   c) Maps **`SystemOut.log`** and **`trace.log`** records back to the SCA programming model

3. Which statement is true about FFDC?
   a) Enabling the FFDC requires the server to be rebooted.
   b) FFDC is always enabled, and automatically captures key information when a potentially abnormal situation occurs.
   c) You should always check FFDC first, and then check **`SystemOut.log`** and **`SystemErr.log`**.

Figure 9-47. Checkpoint questions WB8691.0

Notes:

Write down your answers here:

1.

2.

3.

WebSphere Education

Checkpoint answers

1. True.

2. (c) Maps **`SystemOut.log`** and **`trace.log`** records back to the SCA programming model.

3. (b) FFDC is always enabled, and automatically captures key information when a potentially abnormal situation occurs.

Figure 9-48. Checkpoint answers WB8691.0

Notes:

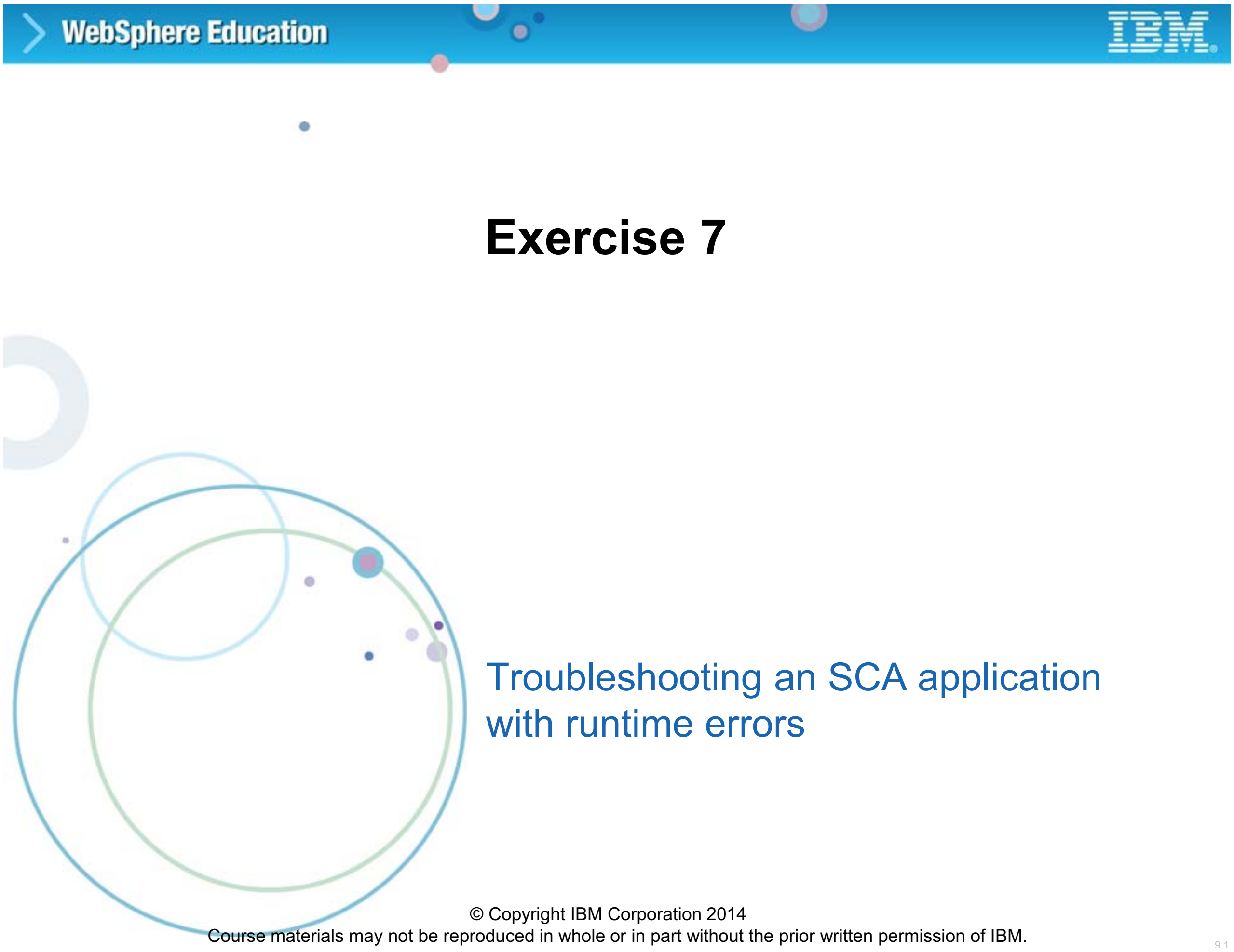


Figure 9-49. Exercise 7 WB8691.0

Notes:

WebSphere Education

Exercise objectives

After completing this exercise, you should be able to:

- View server console log messages in IBM Integration Designer
- Deploy SCA applications onto Business Process Manager
- Use the cross-component trace to follow the call sequences between SCA applications
- Query failed events through the Failed Event Manager and resubmit the events after the problem is resolved

Figure 9-50. Exercise objectives WB8691.0

Notes:

Unit 10. Business Process Choreographer problems

What this unit is about

Business Process Choreographer is the major component of the IBM Process Server, and its container must be installed and configured correctly. Since the BPEL runs on the business process engine, there are tools to help you troubleshoot business process problems. In a way that is similar to SCA troubleshooting, you must be able to differentiate runtime problems from design problems by gathering necessary information (log and trace). Business Process Choreographer Explorer provides monitoring capabilities for running processes, while IBM Integration Designer offers tools for examining the BPEL design. Since the focus of this course is troubleshooting production environments, troubleshooting the BPEL design is kept to a minimum.

What you should be able to do

After completing this unit, you should be able to:

- Collect MustGather data for the Business Process Choreographer
- Enable logging and tracing to gather relevant data
- Examine the trace log to follow the navigation steps inside the business process engine
- Describe the Business Process Choreographer application installation architecture
- Describe some of the known pitfalls that are associated with Business Process Choreographer application installation
- Explain when business process and human task component validation takes place
- Determine the status of process instances by using Business Process Choreographer Explorer
- Schedule the deletion of completed process instances

How you will check your progress

- Checkpoint

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Collect MustGather data for the Business Process Choreographer
- Enable logging and tracing to gather relevant data
- Examine the trace log to follow the navigation steps inside the business process engine
- Describe the Business Process Choreographer application installation architecture
- Describe some of the known pitfalls that are associated with Business Process Choreographer application installation
- Explain when business process and human task component validation takes place
- Determine the status of process instances by using Business Process Choreographer Explorer
- Schedule the deletion of completed process instances

Figure 10-1. Unit objectives WB8691.0

Notes:

WebSphere Education

IBM

Topics

- Gather data
- Application installation
- Validation
- Business process runtime management
- Troubleshooting tips

Figure 10-2. Topics WB8691.0

Notes:

10.1.Gather data

This topic introduces the MustGather data that is specific to the Business Process Choreographer. A sample Business Flow Manager (BFM) trace log is provided to help you analyze the event flow.

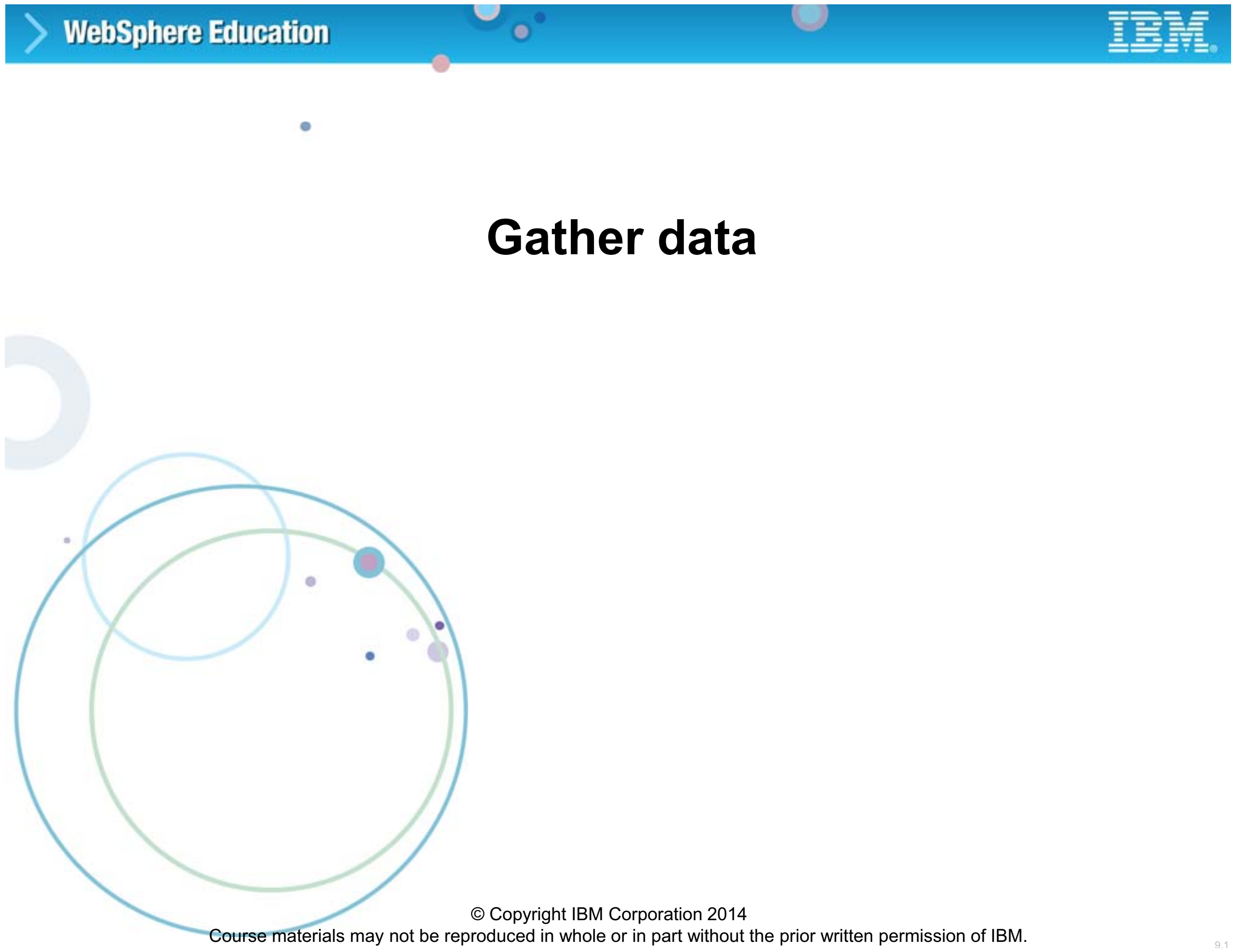


Figure 10-3. Gather data WB8691.0

Notes:

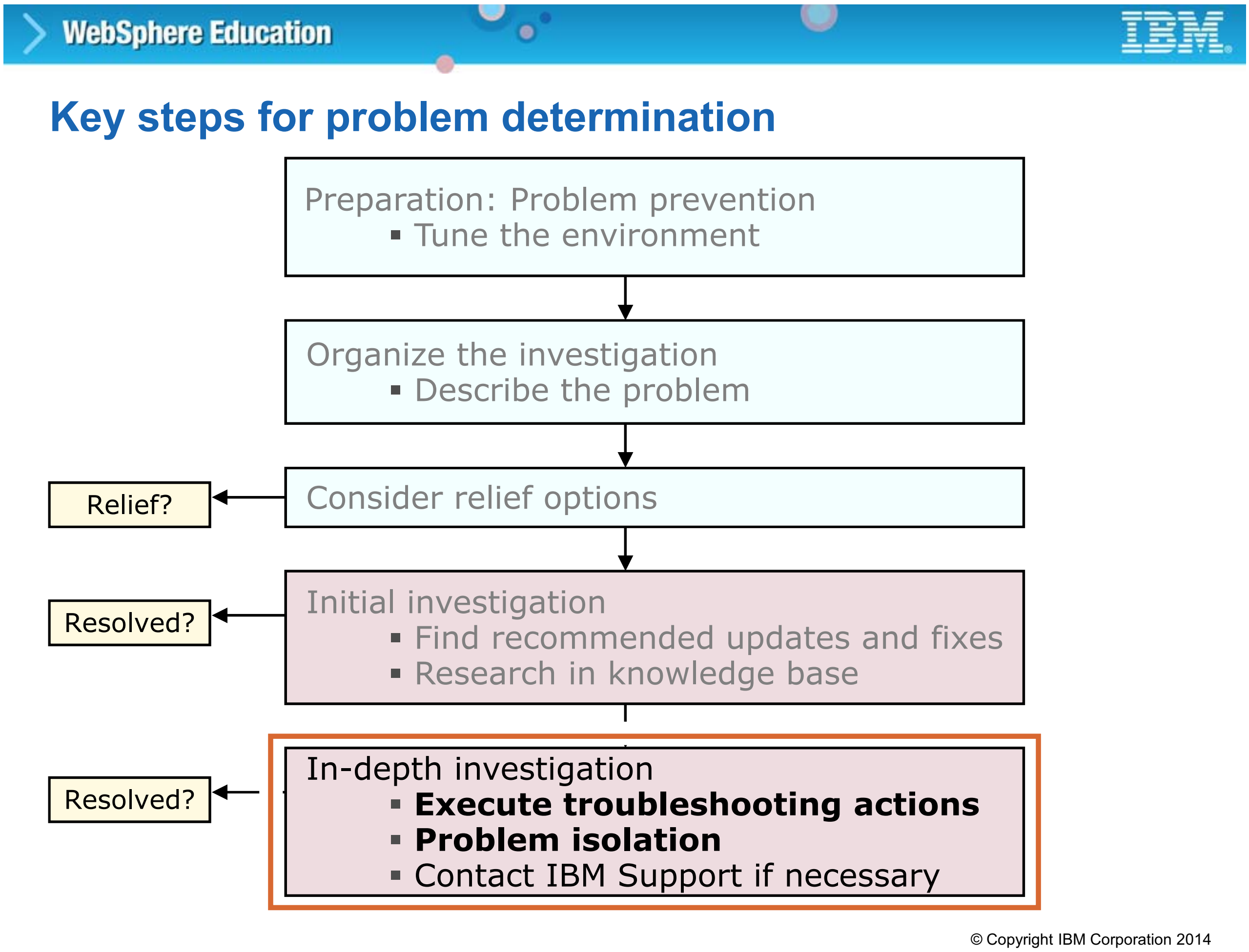


Figure 10-4. Key steps for problem determination WB8691.0

Notes:

WebSphere Education

MustGather data: General information for Business Process Choreographer

- IBM Process Server version and fix pack version
- Database product and product version that is used for Business Process Choreographer repository (BPEDB)
- Environment topology information
  - Stand-alone environment or a clustered environment
- Information about the cluster
  - Number and names of clusters
  - Messaging engines
  - Business Flow Manager and Human Task Manager container
- Information about the environment with problem: production, test, or development environment
- Generate and upload a **`versionReport.html`** file:

```
<install_root>/bin/genVersionReport.bat -maintenancePackages
```


Figure 10-5. MustGather data: General information for Business Process Choreographer WB8691.0

Notes:

WebSphere Education

Business Process Choreographer log files (1 of 2)

- Profile creation
  - The profile actions for Business Process Choreographer write to the `bpcaugment.log` file in the `install_root/logs/manageprofiles/profileName/logs` directory
  - You can find more detailed traces in the `createBPCObjects.traceout` file in the same directory
- Administrative scripts
  - All of the Business Process Choreographer scripts that are run using wsadmin are logged in the `wsadmin.traceout` file in the logs directory of the profile tool

Note: This file is overwritten each time **wsadmin** is invoked

- Administrative utility scripts
  - The administrative scripts in the admin subdirectory of the `ProcessChoreographer` directory do not write their own log files
  - Check the `wsadmin.traceout` file and the application server log file

Figure 10-6. Business Process Choreographer log files (1 of 2) WB8691.0

Notes:

WebSphere Education IBM

Business Process Choreographer log files (2 of 2)

- Configuration-related scripts
  - The script files write their log files in the logs directory

    `bpeconfig.jacl` → **`bpeconfig.log`**

    `bpeupgrade.jacl` → **`bpeupgrade.log`**

    `clientconfig.jacl` → **`clientconfig.log`**

    `bpeunconfig.jacl` → **`bpeunconfig.log`**

  - The **`setUpEventCollector.bat`** configuration scripts write **`setupEventCollector.log`** in the logs directory

Figure 10-7. Business Process Choreographer log files (2 of 2) WB8691.0

Notes:

WebSphere Education

Enable tracing to gather relevant runtime data

- In the administration console, expand **Troubleshooting > Logs and Trace >** ***server_name*** **> Change Log Detail Level**
  - Business Flow Manager (BFM): `com.ibm.bpe.*=all`
  - Business Process Choreographer Explorer and web client: `com.ibm.bpe.client.*=all`
  - Compensation: `compensation=all`
  - For database problems: `RRA=all`
  - Human Task Manager (HTM): `com.ibm.task.*=all`
  - Staff resolution, staff support, and LDAP problems: `com.ibm.task.*all:com.ibm.ws.security.*=all:com.ibm.ws.staffsupport.*=all`
- Messages that belong to Business Process Choreographer are prefixed with:
  - **CWWB** for process-related messages
  - **CWTK** for task-related messages

Figure 10-8. Enable tracing to gather relevant runtime data WB8691.0

Notes:

WebSphere Education

Business Flow Manager (BFM) trace analysis

- Navigation steps inside the engine:
  - Search for the word **`observed`**
  - This action gives a fast overview over the navigation steps
- API transactions:
  - Search for **`BusinessProcessServiceImpl`** method entries and exits or **`TaskManagerServiceImpl`** method entries or exits
- Transactions that are initiated by incoming SCA requests:
  - Search for **`ProcessImplementationHandler.processMessage()`** entry and exit records or **`TaskImplementationHandler.processMessage()`** entry and exit records
- Internal navigation transactions in long-running processes:
  - Search for **`ProcessMDB.onMessage()`** entry and exit

Figure 10-9. Business Flow Manager (BFM) trace analysis WB8691.0

Notes:

WebSphere Education IBM

BFM trace analysis example: Engine trace

```
[9/2/13 15:16:55:921 PDT] 00000070 TraceBPE 1
com.ibm.bpe.engine.observer.BpelStateObserver.observeOrDelay(BpelStateObse
rver.java:1215) Observed process event: processStarted AccountVerification
_PI:90030123.7cd4fb23.6d74573f.78bd0010
...
[9/2/13 15:16:56:328 PDT] 00000070 TraceBPE 1
com.ibm.bpe.engine.observer.BpelStateObserver.observeOrDelay(BpelStateObse
rver.java:1565) Observed scope event: scopeStarted null
AccountVerification _PI:90030123.7cd4fb23.6d74573f.78bd0010
...
[9/2/13 15:16:56:468 PDT] 00000070 TraceBPE 1
com.ibm.bpe.engine.observer.BpelStateObserver.observeOrDelay(BpelStateObse
rver.java:1333) Observed activity event: activityStarted
AccountVerificationReceive AccountVerification
_AI:90040123.7cd4fd94.6d74573f.78bd0028
_PI:90030123.7cd4fb23.6d74573f.78bd0010
...
[9/2/13 15:16:56:500 PDT] 00000070 TraceBPE 1
com.ibm.bpe.engine.observer.BpelStateObserver.processInstanceCreationCompl
eted(BpelStateObserver.java:1094) Observed process instance creation
completion: AccountVerification _PI:90030123.7cd4fb23.6d74573f.78bd0010
```

A new process instance ID

Engine starts a scope, which is used for process navigation

A receive activity, "**AccountVerificationReceive**" was started

Figure 10-10. BFM trace analysis example: Engine trace WB8691.0

Notes:

WebSphere Education

BFM trace analysis example: Navigation transaction

```
[9/2/13 15:16:57:750 PDT] 00000070 TraceBPE >
com.ibm.bpe.framework.ProcessMDB.onMessage(ProcessMDB.java:248) ENTRY
...
[9/2/13 15:16:58:015 PDT] 00000070 TraceBPE >
com.ibm.bpe.framework.sca.ProcessSessionBean.bpc_onMessage(ProcessSession
Bean.java:2281) ENTRY
[9/2/13 15:16:58:015 PDT] 00000070 TraceBPE >
com.ibm.bpe.framework.sca.ProcessSessionBean.runInComponentContext(Proces
sSessionBean.java:1467) ENTRY
...
[9/2/13 15:16:58:015 PDT] 00000070 TraceBPE >
com.ibm.bpe.engine.BpelEngine.onMessage(BpelEngine.java:1322) ENTRY
...
[9/2/13 15:17:01:593 PDT] 00000070 TraceBPE 1
com.ibm.bpe.engine.observer.BpelStateObserver.observeOrDelay(BpelStateObs
erver.java:1702) Observed variable event: VariableUpdated
CustomerApplicationVariable AccountVerification
_PI:90030123.7cd4fb23.6d74573f.78bd0010
...
[9/2/13 15:17:01:156 PDT] 00000070 TraceBPE <
com.ibm.bpe.framework.ProcessMDB.onMessage(ProcessMDB.java:555) RETURN
```

Transaction starts with entry to ProcessMDB.onMessage()

The control goes through SCA infrastructure to establish SCA context

BPEL engine is called; observed events show internal navigation steps

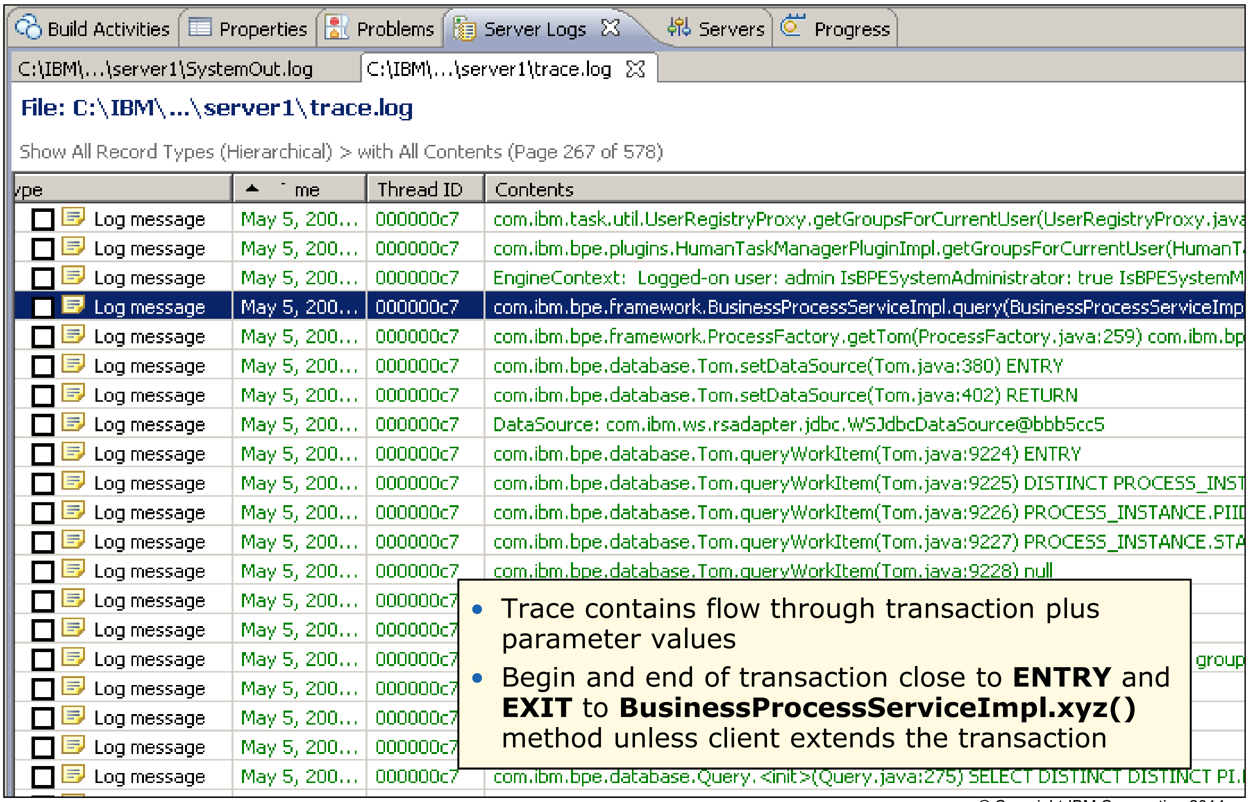
Transaction ends with RETURN

Figure 10-11. BFM trace analysis example: Navigation transaction WB8691.0

Notes:

WebSphere Education

BFM trace analysis example: API transaction



© Copyright IBM Corporation 2014

Figure 10-12. BFM trace analysis example: API transaction WB8691.0

Notes:

The **Server Logs** view in IBM Integration Designer makes it easier to view the trace log. For example, you can sort the entries by the thread ID.

WebSphere Education

Trace activity chains with the cross-component trace view

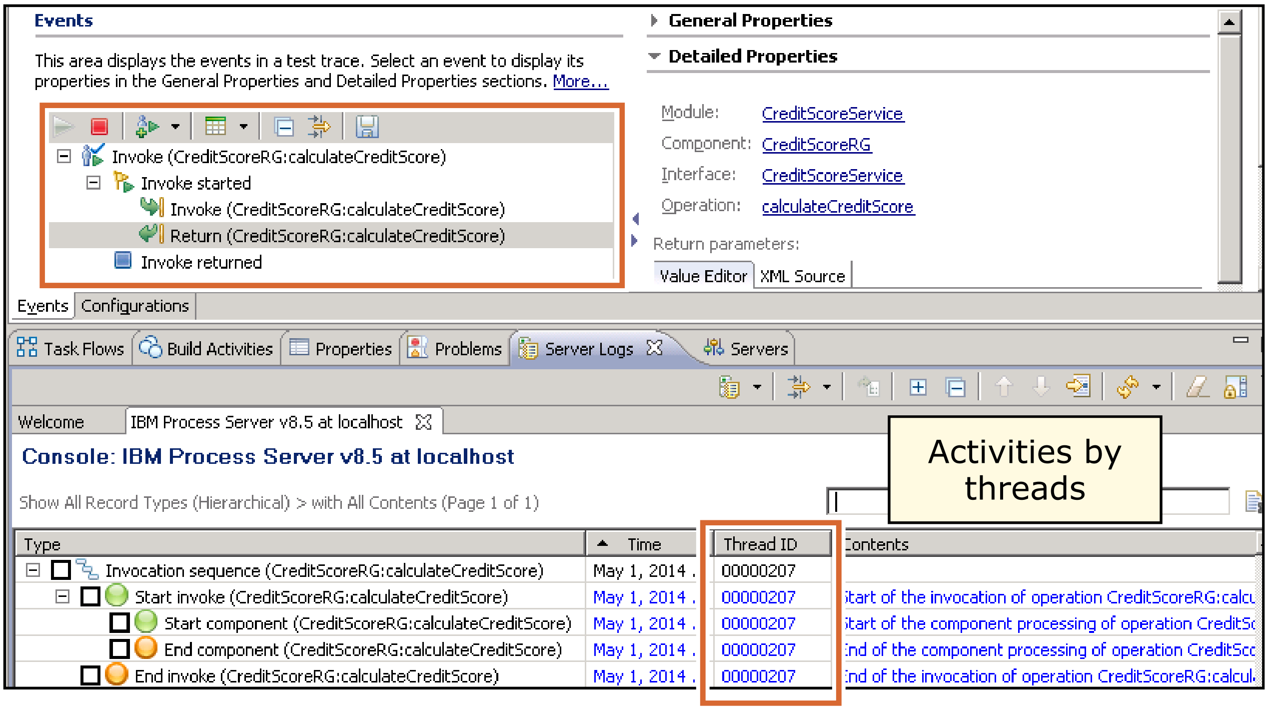


Figure 10-13. Trace activity chains with the cross-component trace view WB8691.0

Notes:

10.2.Application installation

This topic describes how an enterprise application that contains business process templates gets installed onto the server. Some of the common problems and pitfalls are introduced as well. An application that contains business process templates must be uninstalled in a certain order. Therefore, this topic explains the conditions that must be ensured before the application uninstallation.

WebSphere Education IBM

Application installation

Figure 10-14. Application installation WB8691.0

Notes:

WebSphere Education

Business Process Choreographer: Application installation (1 of 2)

- Stand-alone server
  - All application installation steps are executed on the stand-alone server
- Network deployment (ND) environment
  - **Deployment manager**: Performs validation, code generation, and the configuration repository updates
  - **Node agent**: Notifies affected deployment target servers during synchronization of the configuration repository
  - Application server within the deployment target (managed server or cluster): performs deployment of business processes and human tasks into BPEDB

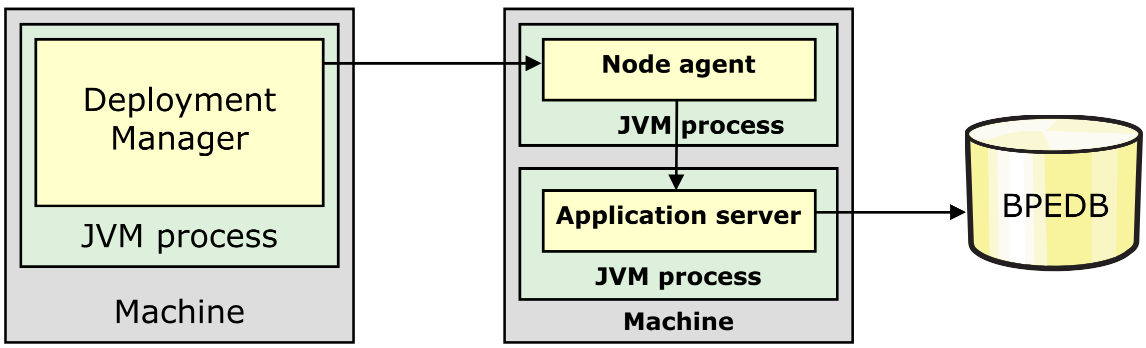


Figure 10-15. Business Process Choreographer: Application installation (1 of 2) WB8691.0

Notes:

WebSphere Education

Business Process Choreographer: Application installation (2 of 2)

Business Process Choreographer application installation consists of the following elements:

- Application configuration
  - Updates the WebSphere configuration repository
- Validation
  - Validates business processes (`.bpel`) and human tasks (`.tel`) for correctness
- Code generation
  - Generates and compiles custom code for JavaSnippets
  - Takes place during the application installation
- Deployment
  - Stores business process and human task templates into Business Process Choreographer database (BPEDB)

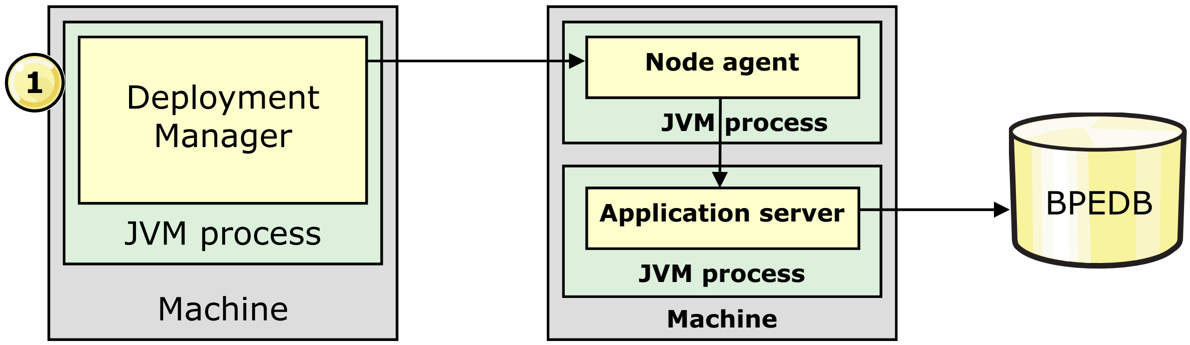
Figure 10-16. Business Process Choreographer: Application installation (2 of 2) WB8691.0

Notes:

WebSphere Education

Typical installation steps: Stage 1 (1 of 2)

- **Stage 1:** Validation, code generation, and then configuration repository update



- Messages can be found in the **`SystemOut.log`**:

```
[2/6/14 13:06:16:593 CET] 00000048 BPELValidation I CWWB001I:
  Validated process model 'BVTAssignP' successfully: 0
  warnings, 0 information
```


Figure 10-17. Typical installation steps: Stage 1 (1 of 2) WB8691.0

Notes:

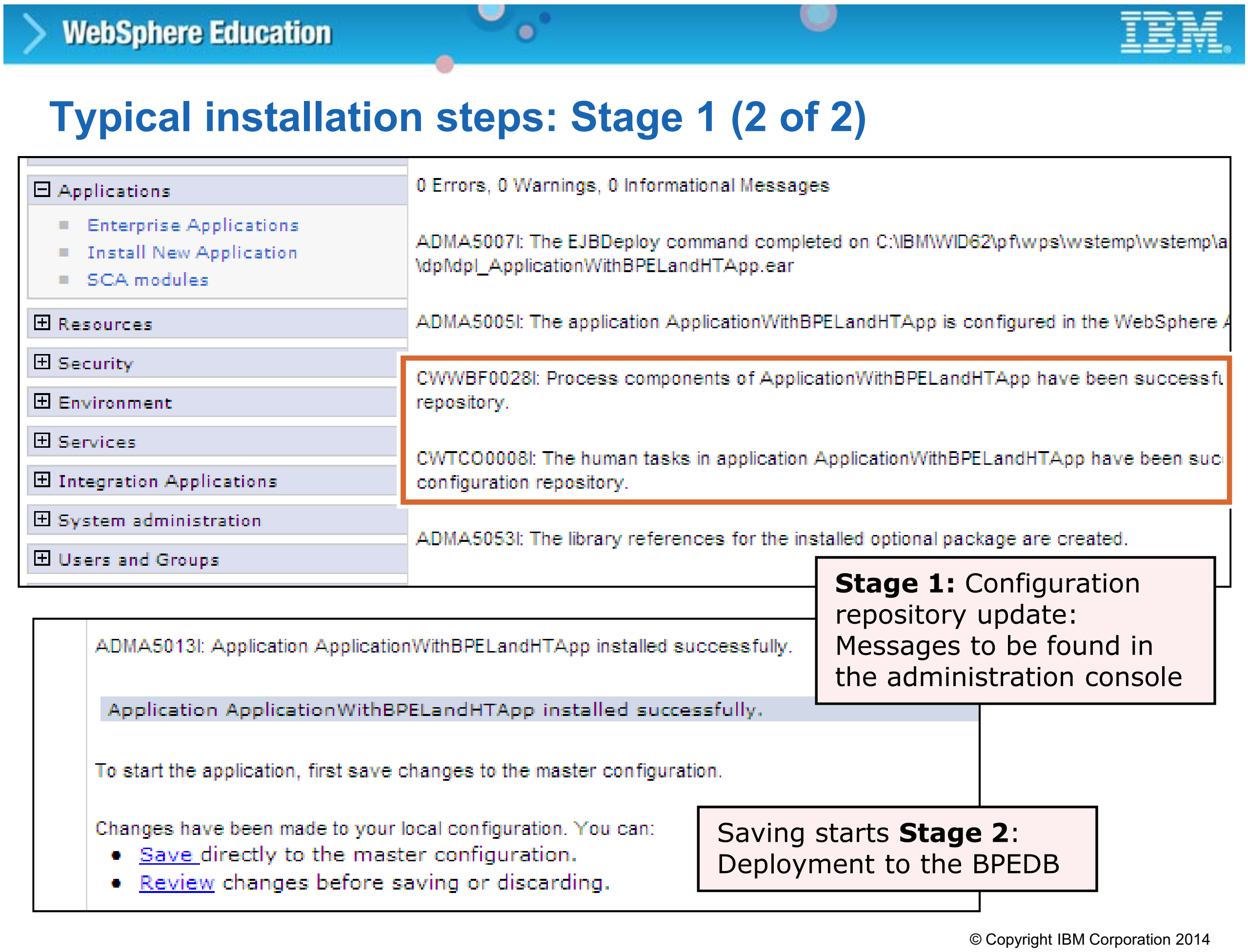


Figure 10-18. Typical installation steps: Stage 1 (2 of 2) WB8691.0

Notes:

Upon a successful configuration, the following messages should be displayed:

```
CWWBF0028I: Process components of <application name> have been successfully
configured in the WebSphere configuration repository.
CWWTCO0008I: The human tasks in application <application name> have been
successfully configured in the WebSphere configuration repository.
```

WebSphere Education

Troubleshooting stage 1 (1 of 2)

- If the configuration repository update fails, then do ***not*** save
  - Click **Manage Applications > Review > Discard**

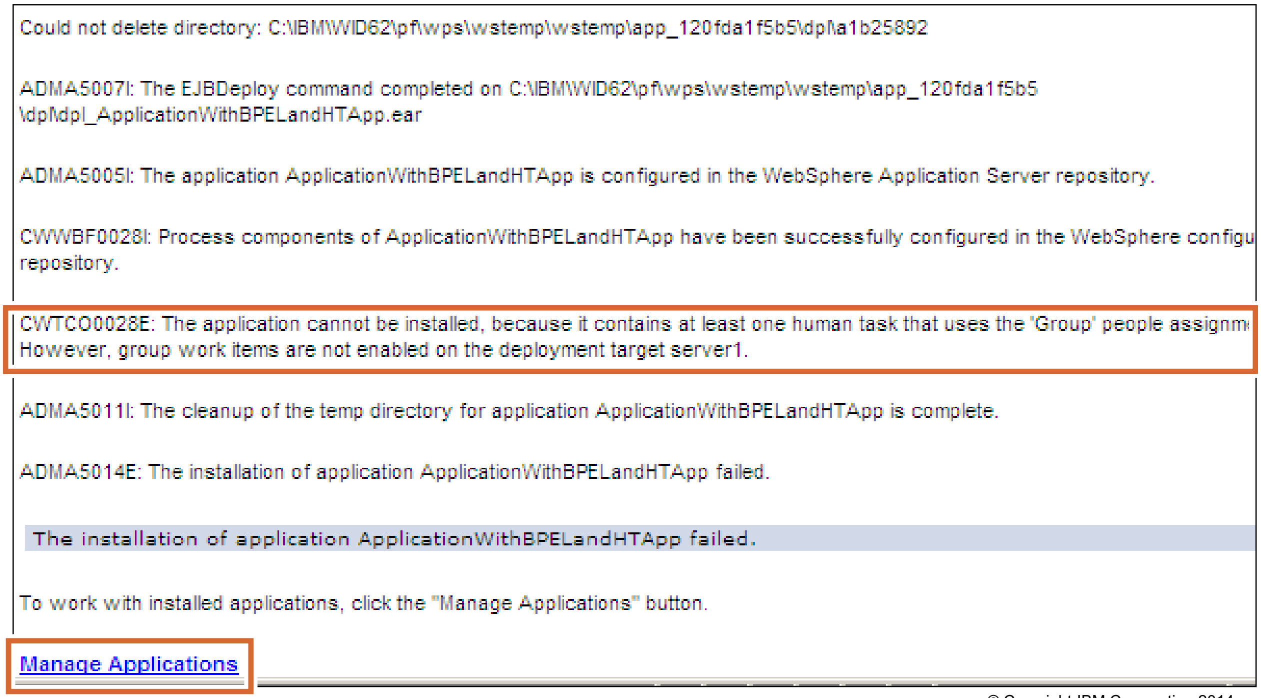


Figure 10-19. Troubleshooting stage 1 (1 of 2) WB8691.0

Notes:

Troubleshooting stage 1 (2 of 2)

- Application installation by using administration console versus **wsadmin** command in connected mode:
  - Logs, traces and FFDC files on deployment manager (or stand-alone server) are required

    ```
    *=info: com.ibm.bpe.*=all: com.ibm.task.*=all:
    com.ibm.ws.staffsupport.*=all
    ```

- Application installation by using the **wsadmin** command in disconnected mode:
  - Logs, traces and FFDC files on deployment manager (or stand-alone server) are required
  - Enable **wsadmin.traceout** by adding the following option in the <profilename>/properties/**wsadmin.properties** file:

    ```
    com.ibm.ws.scripting.traceString=com.ibm.bpe.
    *=all=enabled:com.ibm.task.*=all=enabled:com.ibm.ws.
    staffsupport.*=all=enabled
    ```


Figure 10-20. Troubleshooting stage 1 (2 of 2) WB8691.0

Notes:

WebSphere Education

Typical installation steps: Stage 2

- **Stage 2:** Deployment to the BPEDB
  - Triggered by the configuration repository update
- Execution environment for deployment code depends on deployment environment
  - Stand-alone server: The deployment code is running on the stand-alone server
  - Network deployment (ND) environment: The deployment code is running on one arbitrary server, which is part of the deployment target (managed server or cluster) and not on the deployment manager

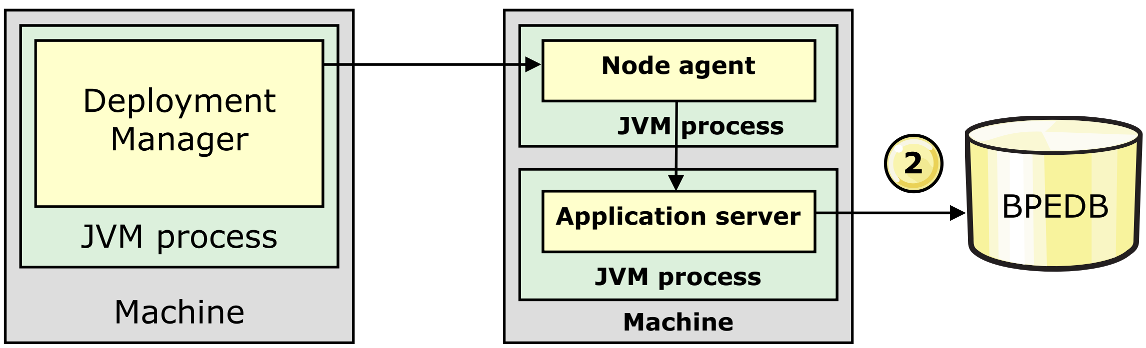


Figure 10-21. Typical installation steps: Stage 2 WB8691.0

Notes:

WebSphere Education

Problem during stage 2

Symptoms:

- Application does not start after the configuration repository was saved

Actions to take:

- Gather logs, traces, and FFDC files

```
*=info: com.ibm.bpe.*=all: com.ibm.task.*=all:
com.ibm.ws.staffsupport.*=all
```

- If the deployment environment is a stand-alone server, tracing needs to be enabled on the stand-alone server
- If the deployment environment is ND:
  - Tracing needs to be enabled on all servers that are part of the deployment target
  - Tracing needs to be enabled on all node agents that are managing these servers

Figure 10-22. Problem during stage 2 WB8691.0

Notes:

WebSphere Education

Application installation facts (1 of 2)

Scenario 1:

- The application that was created on IBM Integration Designer V7 does not install on IBM Process Server V8.5
  - The runtime version has to be equal or higher than the EAR file version you are attempting to install

Scenario 2:

- You have a cluster with cluster members with different versions of IBM Process Server and you migrated half of the members, but you were not able to update your business process applications
  - A mixed version environment does not support the installation, update, or uninstallation of applications, which contain business processes or human tasks
  - Therefore, finish the migration before attempting to update your business process applications

Figure 10-23. Application installation facts (1 of 2) WB8691.0

Notes:

For more information about the IBM Business Process Manager V8.5, see the following website:
`http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r5m0/index.jsp`

WebSphere Education

Application installation facts (2 of 2)

Uninstalling enterprise applications that contain business processes or human tasks

- All the business process and human task instances must ***not*** be in running, finished, or failed states (the instances must be completed before uninstalling the application)
  - **Exception:** If the application server is running in development mode, applications can be uninstalled regardless of the process instance state

- Using the **`bpcTemplates.jacl`** script provides an alternative to the administrative console for uninstalling applications that contain business processes or human tasks
  - **Note:** If there are instances of business process or human task templates present in any state, you must use the **`-force`** option

```
bpcTemplates.jacl ( ( -start | -stop ) <app_name> )|( -uninstall
<app_name> [-force] )
```


Figure 10-24. Application installation facts (2 of 2) WB8691.0

Notes:

Known pitfalls: Deployment

- Deployment for business processes and human tasks is not triggered if the process or task component does not exist
  - Ensure that there is a process or task component as part of the assembly diagram to be deployed
  - If the business process or human task was not dropped on the assembly diagram at modeling time, no component is created and finally no deployment of these artifacts takes place

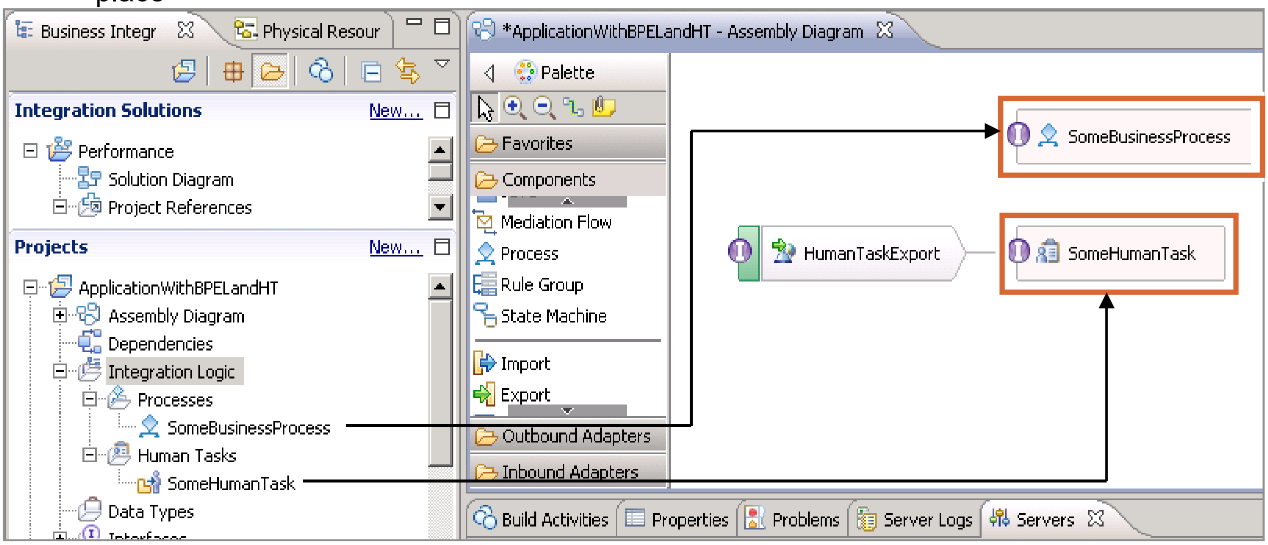


Figure 10-25. Known pitfalls: Deployment WB8691.0

Notes:

WebSphere Education

Known pitfalls: Late binding calls the wrong version of a subprocess

Symptom:

- A parent process invokes a subprocess with late binding
- Both processes are in the same module
- A new version of the subprocess is created by copying the module and changing the valid-from timestamp
- After the module is deployed, the running instances of the parent process continue to invoke the old version of the subprocess instead of the new version

Reason:

- Most likely, the SCA export is missing for the subprocess
- Without an SCA export, processes in other modules are not visible to the parent process
- Thus, it always invokes the version of the subprocess that is within the same module

Resolution:

- In the assembly editor in IBM Integration Designer, generate an SCA export with SCA native binding for the new version of the subprocess

Figure 10-26. Known pitfalls: Late binding calls the wrong version of a subprocess WB8691.0

Notes:

WebSphere Education IBM

Known pitfalls: Uninstalling an application that contains business processes

To uninstall an application that contains business processes or human tasks, the following conditions must apply:

- There are ***no instances*** of business process or human task templates present in any state
- If the application is installed on a stand-alone server, the server must be running and have access to the Business Process Choreographer database
- If the application is installed on a cluster, the deployment manager and at least one cluster member must be running
  - The cluster member must have access to the Business Process Choreographer database
- If the application is installed on a managed server, the deployment manager and the managed server must be running
  - The server must have access to the Business Process Choreographer database

Figure 10-27. Known pitfalls: Uninstalling an application that contains business processes WB8691.0

Notes:

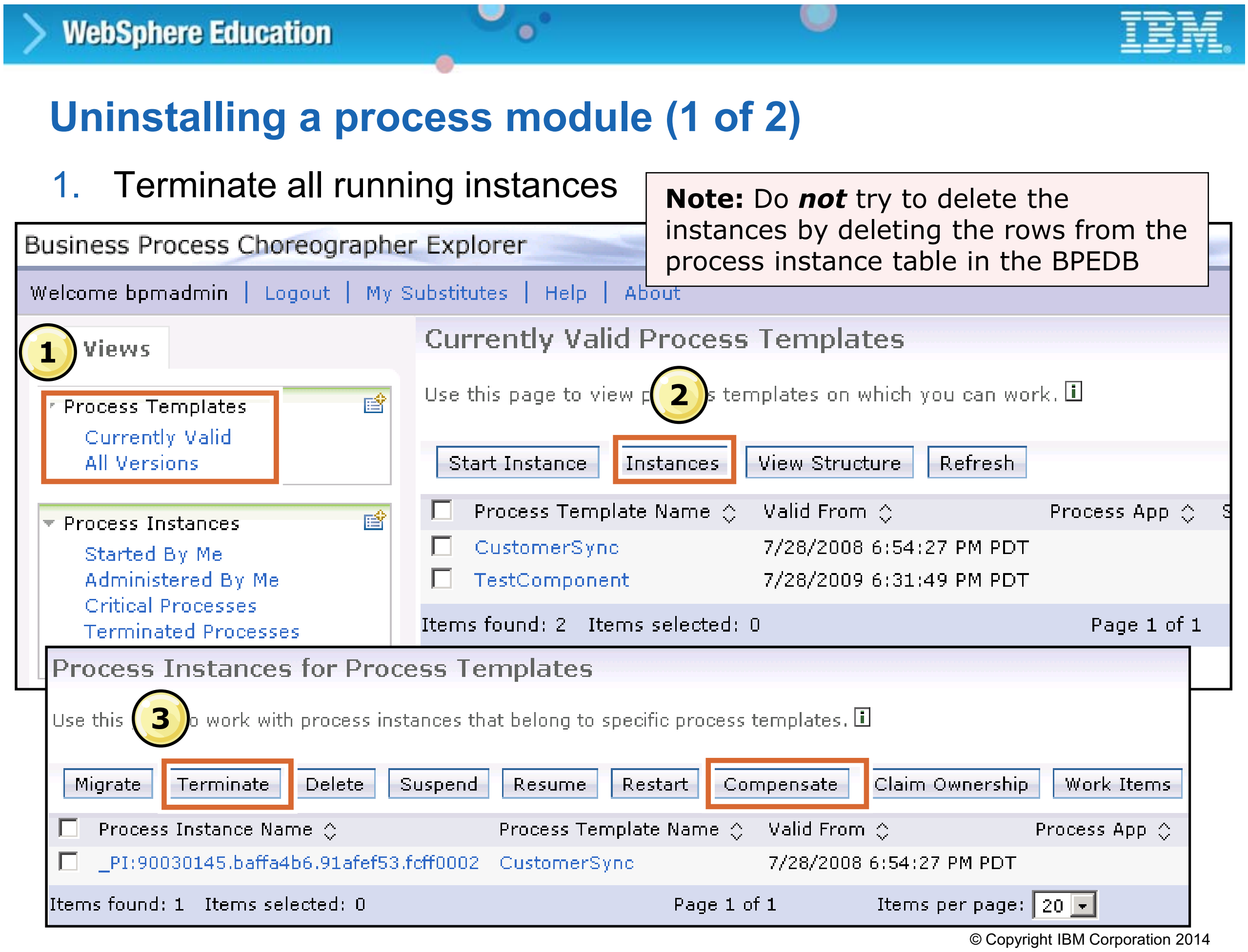


Figure 10-28. Uninstalling a process module (1 of 2) WB8691.0

Notes:

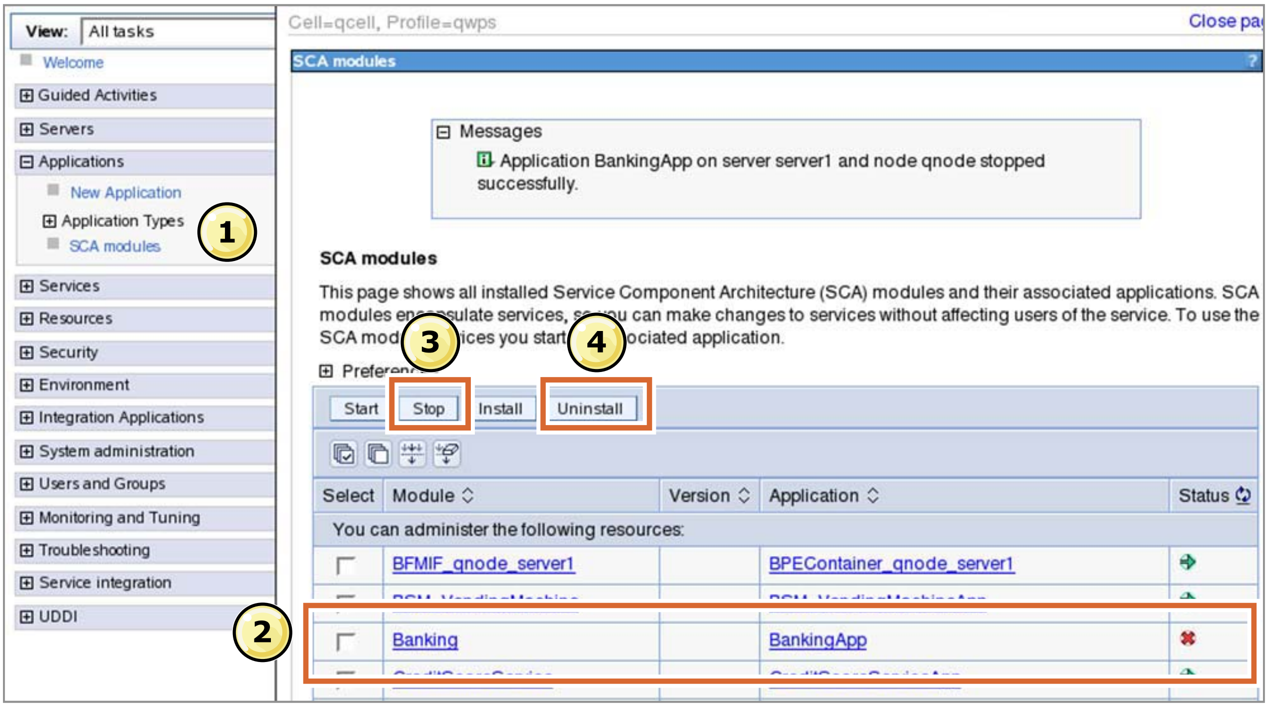
1. In the Business Process Choreographer Explorer, select the **Process Templates** link.
2. Click **Instances** to retrieve the list of active process instances.
3. If there are any active instances, click **Terminate** to terminate the process. If the instance must be compensated, click **Compensate**, which triggers the compensation logic and then terminate the instance.

Note: Do *not* try to delete the instances by deleting the rows from the PROCESS_INSTANCE_B_T in the BPEDB.

WebSphere Education IBM

Uninstalling a process module (2 of 2)

2. Stop the enterprise application; then uninstall



© Copyright IBM Corporation 2014

Figure 10-29. Uninstalling a process module (2 of 2) WB8691.0

Notes:

1. In the administration console, select **Applications > SCA modules**.
2. Select the application that you want to uninstall.
3. Click **Stop** to stop the application. When the application was successfully stopped, the status changes to red X.
4. Finally, click **Uninstall** to uninstall the SCA application.

10.3.Validation

This topic covers process template validation, and also task template validation. A sample log message in the SystemOut.log file shows which message indicates a validation error. Some common mistakes are also discussed.

WebSphere Education

IBM

Validation

Figure 10-30. Validation WB8691.0

Notes:

WebSphere Education IBM

Validation: Business processes and human tasks (1 of 2)

- Static analysis of business processes and human tasks
- Reports errors if the business process or the human task is invalid or incomplete
- Validation runs in IBM Integration Designer and **`serviceDeploy`** if a **`*.bpel`**, **`*.tel`**, or **`*.itel`** file changes
- Validation runs while a business process or human task is installed
  - The installation of an application (**`*.ear`**) that comprises a business process or human task
  - Validation runs on the **`deploymentManager`**
- Syntactical validation against an XSD
- Semantic validation that is based on the **`*.bpel`** or **`*.tel`** Eclipse Modeling Framework (EMF) model

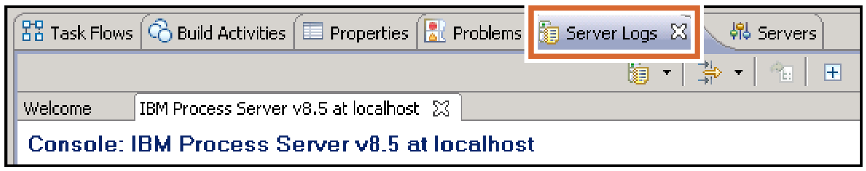
Figure 10-31. Validation: Business processes and human tasks (1 of 2) WB8691.0

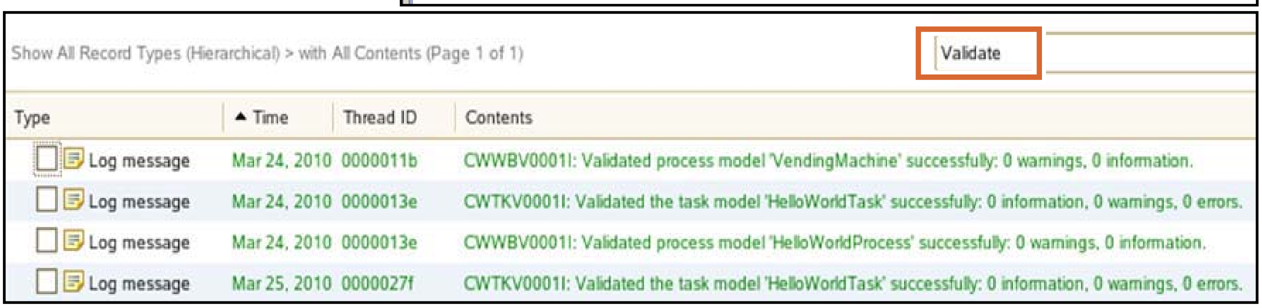
Notes:

WebSphere Education IBM

Validation: Business processes and human tasks (2 of 2)

- Validates human tasks (`*.tel`) and inline human tasks (`*.itel`)
- Message error code: **CWWBV** (BPEL standard terms) and **CWWBW** (simplified terms that are used in IBM Integration Designer only)
- Error code for human task: **CWTKV**
- All messages with explanation and user action that is associated with the validation output can be found in the information center
- A success message in `SystemOut.log` of the deployment manager looks as follows:




Show All Record Types (Hierarchical) > with All Contents (Page 1 of 1)   Validate

Type	Time	Thread ID	Contents
Log message	Mar 24, 2010	0000011b	CWWBV0001I: Validated process model 'VendingMachine' successfully: 0 warnings, 0 information.
Log message	Mar 24, 2010	0000013e	CWTKV0001I: Validated the task model 'HelloWorldTask' successfully: 0 information, 0 warnings, 0 errors.
Log message	Mar 24, 2010	0000013e	CWWBV0001I: Validated process model 'HelloWorldProcess' successfully: 0 warnings, 0 information.
Log message	Mar 25, 2010	0000027f	CWTKV0001I: Validated the task model 'HelloWorldTask' successfully: 0 information, 0 warnings, 0 errors.

Figure 10-32. Validation: Business processes and human tasks (2 of 2) WB8691.0

Notes:

WebSphere Education

Validation: BPEL

- Error message (**`BPELValidationException`**) is logged in the **`SystemOut.log`** file

```
[1/14/13 11:17:56:484 CET] 0000006b BPELValidatio E CWWBV0002E: Validated
process model 'TestProcess' with findings: 1 errors, 0 warnings, 0 information.
[1/14/13 11:17:56:504 CET] 0000006b BPELValidatio E CWWBV3142E: The variable is
not set (activity 'Receive2').
[1/14/13 11:17:56:584 CET] 0000006b BpelEngine E CWWBV0003E: Validated process
model 'TestProcess' with findings: 1 errors, 0 warnings, 0 information:
CWWBV3142E: The variable is not set (activity 'Receive2').
              com.ibm.bpe.plugins.BPELValidationException: CWWBV0003E: Validated
process model       'TestProcess, with findings: 1 errors, 0 warnings, 0 information:
CWWBV3142E: The variable is not set (activity 'Receive2').
 at
com.ibm.bpe.validation.BPELValidation.writeProblemsToSystemOutWithException(BPEL
Validation.java:800)
 at
com.ibm.bpe.validation.BPELValidation.validateExecutableBPELExtensionsProcess(BPEL
Validation.java:317)
 at com.ibm.bpe.deployment.BPELDeployer.installBPEL(BPELDeployer.java:220)
...
```

- Syntactical error (such as **`*.bpel`** contains unknown extension) message in IBM Integration Designer and **`SystemOut.log`**

```
A syntactical error was found (row: 14, column: 12): cvc-complex-type.2.4.a: Invalid
content was found starting with element '...'
```


Figure 10-33. Validation: BPEL WB8691.0

Notes:

WebSphere Education

Validation: SCA component

- Static analysis of the process or human task extensions of **`*.component`** files
- Validation runs only in IBM Process Server and **`serviceDeploy`**
- Validation performs only semantic validation
  - BPEL and human task-specific rules for **`*.component`** files (SCDL)
  - Validates qualifiers on the references and interfaces
- The errors and warnings that these validations find, which are associated with a **`*.component`** file, can be viewed in the Problems view in IBM Integration Designer

Figure 10-34. Validation: SCA component WB8691.0

Notes:

WebSphere Education IBM

Validation: Troubleshooting

Common mistakes

- IBM Process Server supports most of the WS-BPEL V2.0 standard
  - A well-defined set of the BPEL languages is implemented
  - `*.bpel` files from other sources might be based on a different set of the BPEL language such as BPEL4WS V1.1
  - Importing these business processes can lead to validation errors

- Error message: **`CWWBV3055E: The process does not implement the operation '...' of the port type '...'`**
  - All operations of all port types that are used for the inbound interface of a process need to have an implementation of the process (using **`Receive`** or **`Receive Choice`** activities or **`EventHandlers`**)
  - A BPEL process can invoke operations on other partners (outbound interfaces)
  - The operations of inbound interfaces of the process and the operations of its outbound interfaces must not be specified in the same port type

Figure 10-35. Validation: Troubleshooting WB8691.0

Notes:

10.4.Business process runtime management

This topic covers how to BPC Explorer to manage, monitor, and manipulate running process instances.

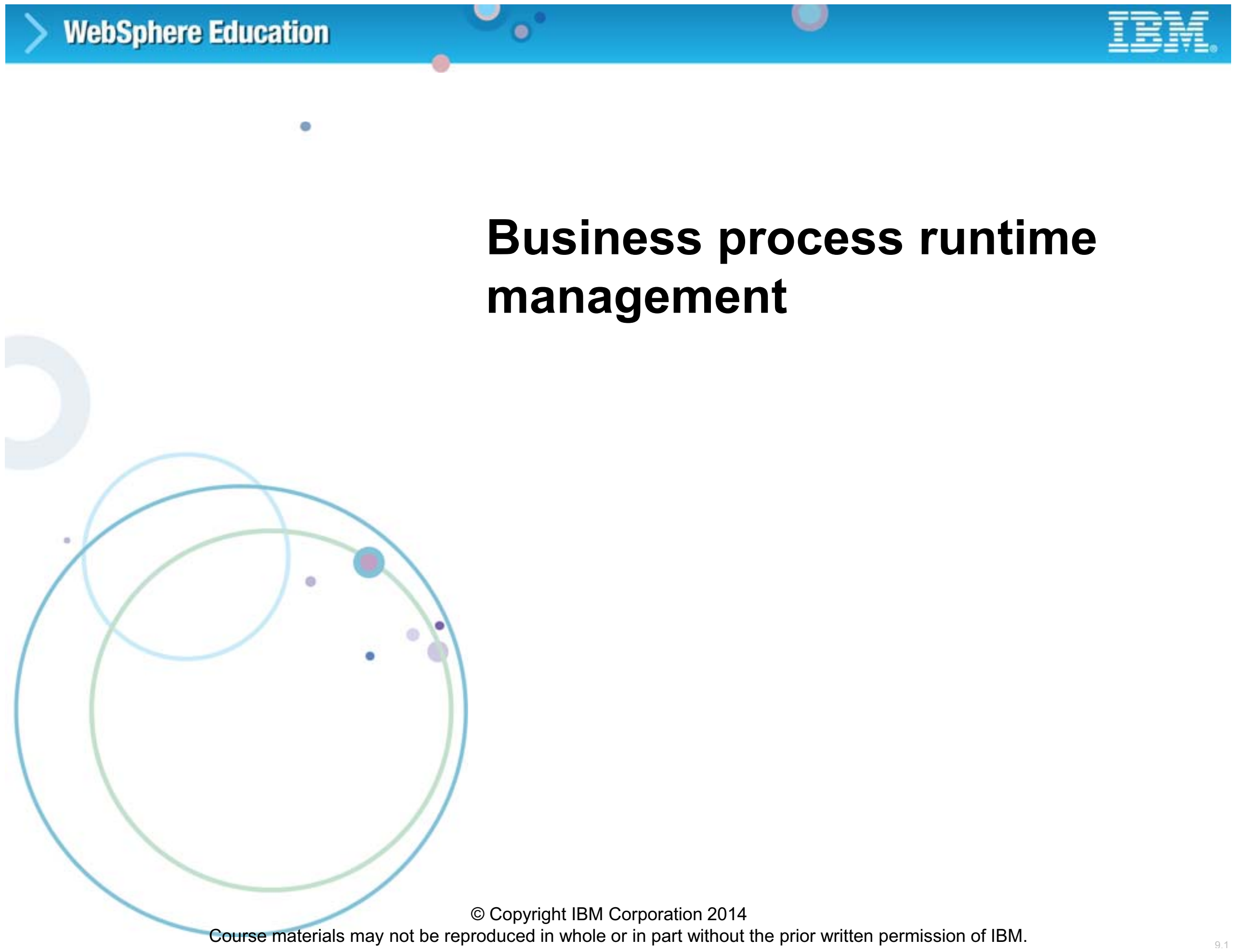


Figure 10-36. Business process runtime management WB8691.0

Notes:

WebSphere Education

Finding business process instances that are related to a failed event (1 of 2)

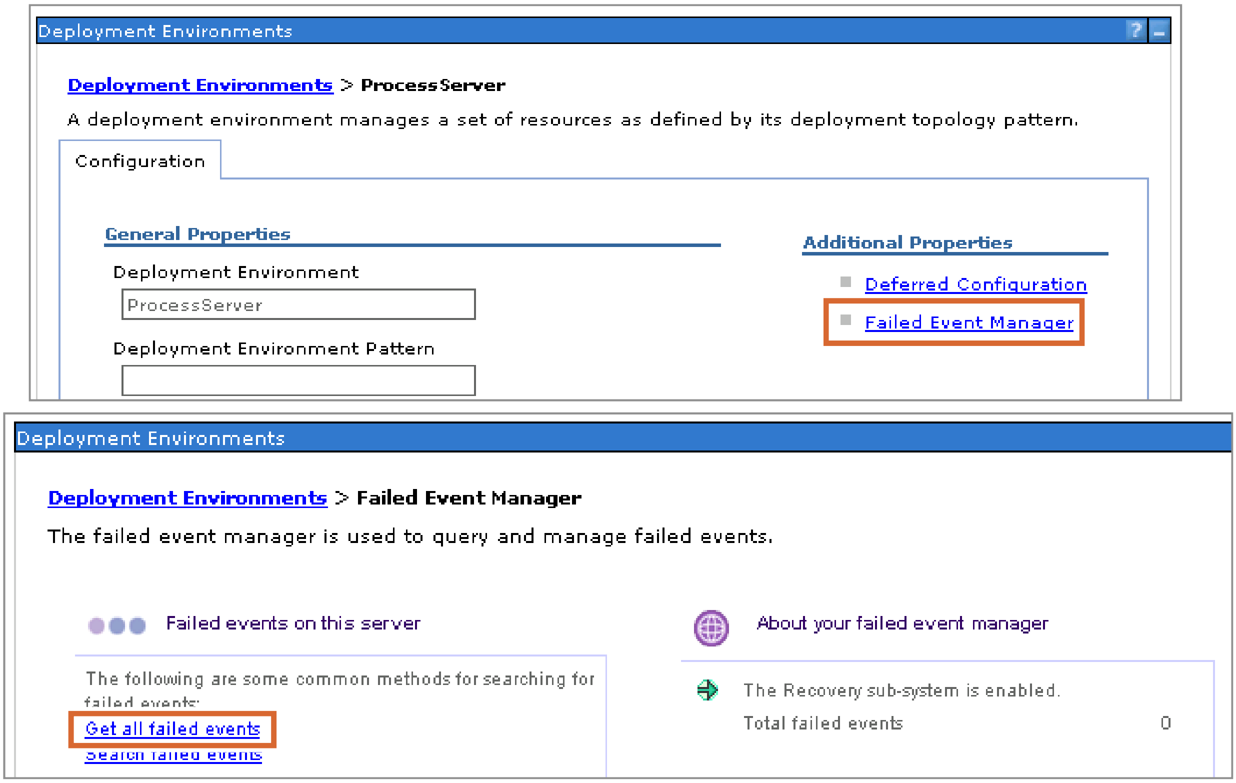


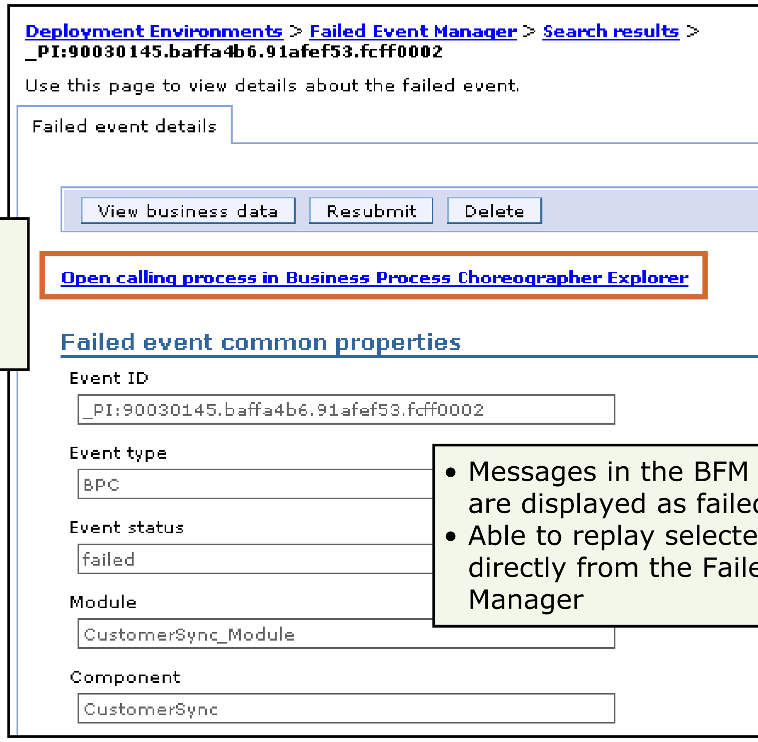
Figure 10-37. Finding business process instances that are related to a failed event (1 of 2) WB8691.0

Notes:

WebSphere Education

Finding business process instances that are related to a failed event (2 of 2)

Access to the Business Process Choreographer Explorer



- Messages in the BFM hold queue are displayed as failed events
- Able to replay selected messages directly from the Failed Event Manager

Figure 10-38. Finding business process instances that are related to a failed event (2 of 2) WB8691.0

Notes:

WebSphere Education

Business Process Choreographer Explorer: Variable repair functions

You can access all variables that are visible to an activity and repair the process by modifying the variable values:

1. Select a process instance.
2. Click the **Variables** button to get a list of all the variables.
3. After selecting and modifying a variable, click **Save** to update the value.
4. Then, you can continue navigating processes.

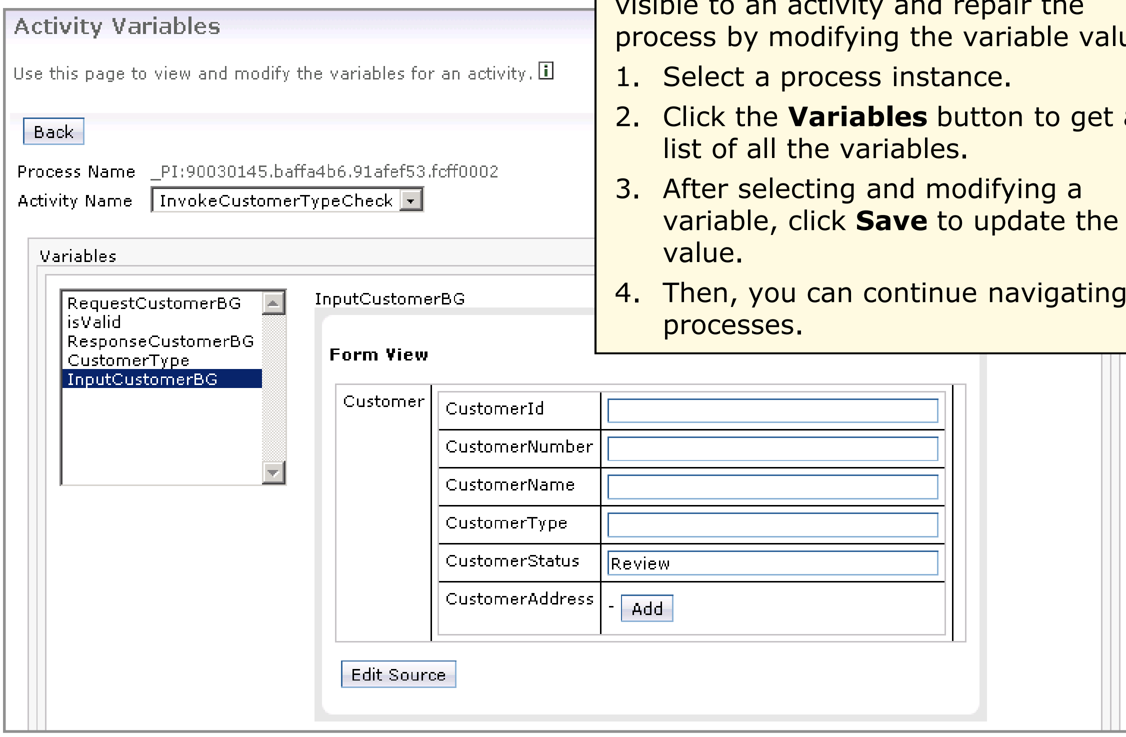


Figure 10-39. Business Process Choreographer Explorer: Variable repair functions WB8691.0

Notes:

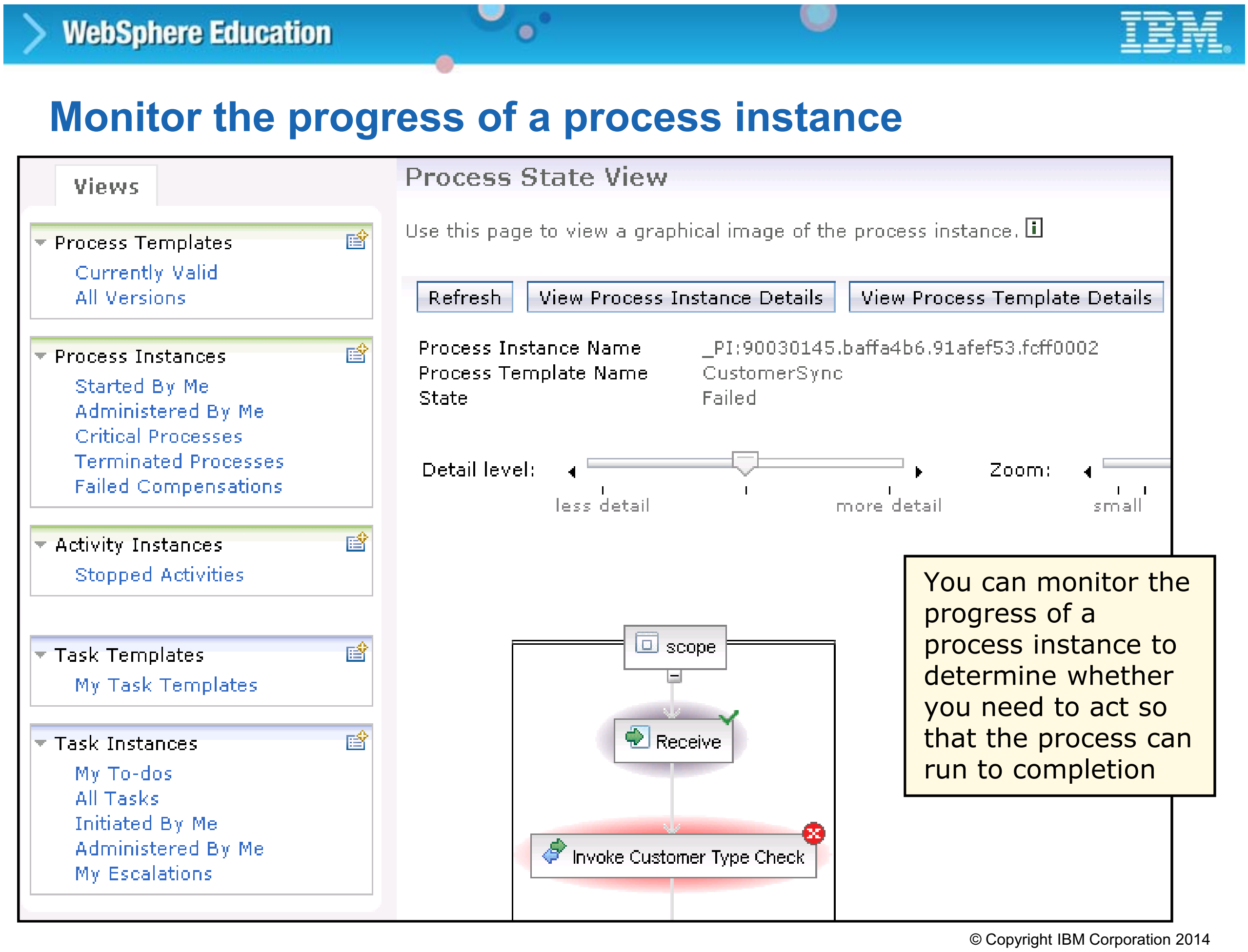


Figure 10-40. Monitor the progress of a process instance WB8691.0

Notes:

1. Display a list of process instances. For example, click **Administered By Me** under Process Instances in the **Views** tab navigation pane.
2. Select the check box next to the process instance and click **View Process State**. The Process State page is displayed. This page shows the activities, the links (including the transition and join conditions for the links), the fault handlers, the compensation handlers, and the event handlers that are defined for the process. Activities are color-coded in the diagram, depending on their state. All states have an associated icon. For example, completed activities have a check mark. For more information, see the online help for the page.
3. To act on an activity, click the activity, and select **Show Activity Details**. Click an activity in the Process State View to open a menu. In this menu, you can view the activity details, skip the activity (mark an activity for skipping), or select it as the source for a jump to a different activity in the process. You can also repair switch activities that failed due to problems with the evaluation of a case condition.

The available actions are displayed. Select the action of your choice. The purpose of this screen capture is to give an example of a process state view in Business Process Choreographer Explorer.

WebSphere Education IBM

Scheduled deletion of completed instances

- ***Cleanup service*** allows scheduled deletion of instances; this feature is included with the product
  - It simplifies maintenance of business processes and human tasks
- ***Cleanup service*** configuration specifies:
  - When the cleanup service should run
  - How long the cleanup service should run (maximum duration in minutes)
  - How many instances it should delete within one technical transaction
  - Which instances to be deleted, which is specified by the *cleanup job*
- A single ***cleanup job*** configuration specifies:
  - Template name and namespace for human task templates of the instances for deletion
  - End states the instances need to be in for the cleanup job to be applied
  - How long to keep the instances before becoming eligible for deletion by the cleanup service

Figure 10-41. Scheduled deletion of completed instances WB8691.0

Notes:

You want completed instances to be deleted 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. You can then 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 only stand-alone human tasks. However, when the Business Flow Manager cleanup service deletes a business process, it also deletes all 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.

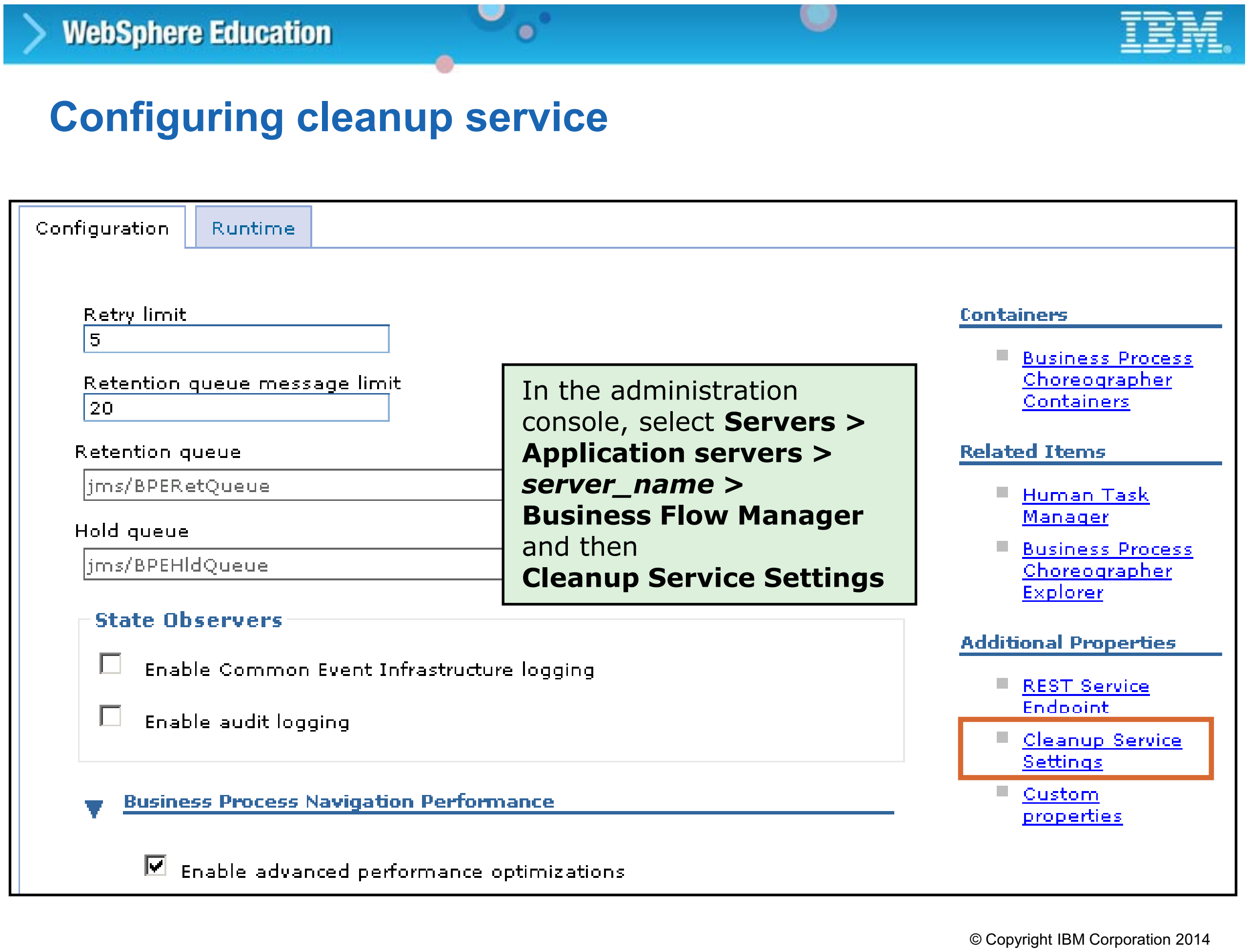


Figure 10-42. Configuring cleanup service WB8691.0

Notes:

There are separate cleanup services for business processes and human tasks.

The configuration is specified in the administration console on the Business Flow Manager configuration pane for process instances, and on the Human Task Manager configuration pane for human tasks.

WebSphere Education

Configuring cleanup jobs

© Copyright IBM Corporation 2014

Figure 10-43. Configuring cleanup jobs WB8691.0

Notes:

10.5. Troubleshooting tips

This topic covers some of the common problems that are associated with Business Process Choreographer and how to resolve them.

Troubleshooting tips

Figure 10-44. Troubleshooting tips WB8691.0

Notes:

WebSphere Education

Hung threads when a long-running process is invoked synchronously (Message: WSVR0605W)

Symptoms:

- A long-running process invokes another long-running process synchronously
- Under heavy workload conditions, the thread monitor reports hung threads in the `SystemOut.log` file (message `WSVR0605W`)

Reason:

- A long-running process that is called synchronously can often cause hung threads
- A long-running process usually spans several transactions and needs a free thread to continue with its navigation
- If all of the available threads are involved in the navigation step of the parent process that invokes the subprocess, the system becomes unresponsive
- Because of the lack of free threads, the subprocess cannot complete

Resolution:

- A long-running process should always invoke another long-running process ***asynchronously***, even if another component separates the processes
- For example, if a long-running process invokes a mediation and this mediation invokes another long-running process, then ensure that the preferred interaction style of the mediation is **asynchronous**

Figure 10-45. Hung threads when a long-running process is invoked synchronously (Message: WSVR0605W) WB8691.0

Notes:

WebSphere Education

NullPointerException in a Java snippet

Symptoms:

- A `NullPointerException` appears in a Java snippet activity when accessing a BPEL variable

Reason:

- According to the BPEL specification, BPEL variables are not implicitly initialized
- Therefore, the value of an uninitialized BPEL variable is just `null` in a Java snippet

Resolution:

- Make sure that the variable is initialized before it is accessed
- In a Java snippet, you might use the `getVariableType()` method and the BOFactory to initialize a BPEL variable

```
BOFactory factory =
     (BOFactory)ServiceManager.INSTANCE.locateService(
     "com/ibm/websphere/bo/BOFactory");
Type type = getVariableType("customerInfo");
customerInfo = factory.createByType(type);
```


Figure 10-46. NullPointerException in a Java snippet WB8691.0

Notes:

WebSphere Education

repeatUntil activity loops forever

Symptoms:

- A `repeatUntil` activity that loops forever (the loop condition never evaluates to **true**)

Reason:

- A possible reason is that a Java expression is used to test the `repeatUntil` loop condition by using a loop counter variable that is updated in this Java expression
- However, by default, access to a BPEL variable inside a Java condition expression is ***read-only***
- That is, any update to the loop counter variable is lost and the BPEL variable stays at its initial value

Resolution:

- Use a separate Java snippet activity inside the `repeatUntil` activity to update the loop counter variable or change the default behavior for the loop counter variable from read-only to read/write
- To do so, place the following Java comment everywhere in your Java condition:

```
//@bpe.readWriteVariables names="..."
```

In the `names="..."` construct you can list a single variable or a list of variables that are separated with blanks

Figure 10-47. repeatUntil activity loops forever WB8691.0

Notes:

WebSphere Education

Data seems to be taken away

Symptoms:

- A DataObject, **`parent1`**, contains DataObject **`child`** per value
- After getting **`child`** from **`parent1`** and setting **`child`** as a property of DataObject **`parent2`**, it disappears from **`parent1`**

Reason:

- The child is moved from **`parent1`** to **`parent2`** according to the SDO programming model

Resolution:

- Use the **`BOCopy.copy()`** API to make a deep copy of **`child`** before setting it as a property of **`parent2`**

```
BOCopy copyService = (BOCopy)
    ServiceManager.INSTANCE.locateService(
    "com/ibm/websphere/bo/BOCopy");
  DataObject child = parent1.get("child");
  DataObject childCopy = copyService.copy(child);
  parent2.set("child", childCopy);
```


Figure 10-48. Data seems to be taken away WB8691.0

Notes:

WebSphere Education

ClassCastException when stopping an application with a microflow

Symptoms:

- The `SystemOut.log` file contains `ClassCastException` exceptions around the time when an application that contains a microflow is stopped

Reason:

- When an application is stopped, the classes that are contained in the EAR file are removed from the class path
- However, microflow instances that need these classes can still be executing

Resolution:

- Perform the following actions:
  1. Stop the microflow process template first. From now on, it is not possible to start new microflow instances from that template.
  2. Wait for at least the maximum duration of the microflow execution so that any running instances can complete.
  3. Stop the application.

Figure 10-49. ClassCastException when stopping an application with a microflow WB8691.0

Notes:

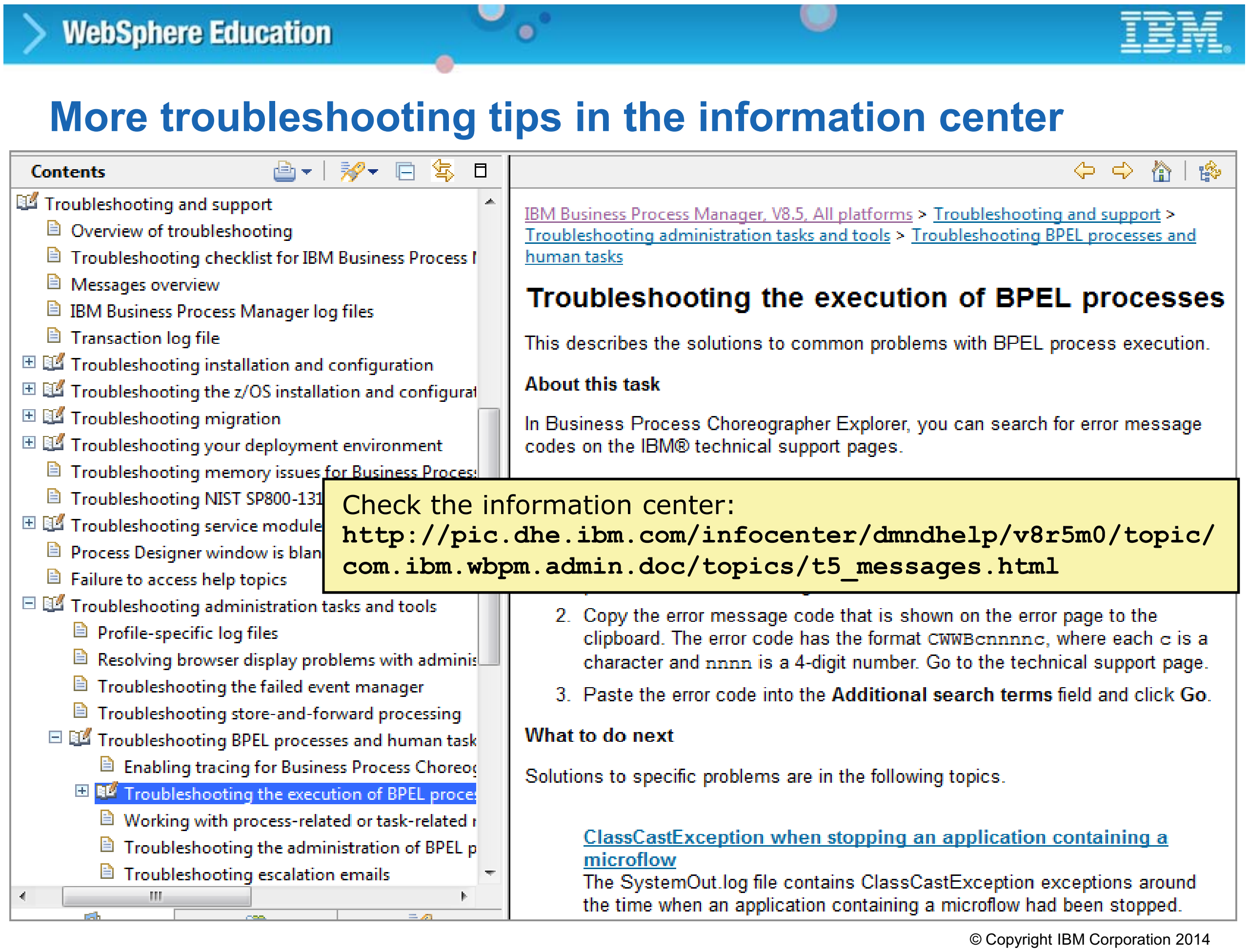


Figure 10-50. More troubleshooting tips in the information center WB8691.0

Notes:

For more information, see the information center:

`http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r5m0/topic/com.ibm.wbpm.admin.doc/topics/t5_messages.html`

WebSphere Education

Troubleshooting tips: Business Process Choreographer database on DB2 (1 of 3)

Symptoms:

- For DB2 Universal JDBC driver type 4, you get DB2 internal errors such as `"com.ibm.db2.jcc.a.re: XAER_RMERR : The DDM parameter value is not supported. DDM parameter code point having unsupported value : 0x113f DB2ConnectionCorrelator: NF000001.PA0C.051117223022"` when you test the connection on the Business Process Choreographer data source or when the server starts up

Actions to take:

- Check the class path settings for the data source
- In a default setup, the WebSphere variable **`${DB2UNIVERSAL_JDBC_DRIVER_PATH}`** can point to the IBM Process Server embedded DB2 Universal JDBC driver, which is found in the **`universalDriver_wbi`** directory
- The version of the driver might not be compatible with your DB2 server version
- Make sure that you use the original **`db2jcc.jar`** files from your database installation, and not the IBM Process Server embedded DB2 Universal JDBC driver
  - If required, changed the value of the WebSphere variable **`${DB2UNIVERSAL_JDBC_DRIVER_PATH}`** to point to your original **`db2jcc.jar`** file
- Restart the server

Figure 10-51. Troubleshooting tips: Business Process Choreographer database on DB2 (1 of 3) WB8691.0

Notes:

WebSphere Education IBM

Troubleshooting tips: Business Process Choreographer database on DB2 (2 of 3)

Symptoms:

- The **`db2diag.log`** file of your DB2 instance contains messages like **`ADM5503E`**:

```
2013-06-25-15.53.42.078000 Instance:DB2 Node:000
PID:2352(db2syscs.exe) TID:4360
Appid:*LOCAL.DB2.027785142343 data management
sqldEscalateLocks Probe:4 Database:BPEDB

ADM5503E The escalation of "10" locks on table "GRAALFS
.ACTIVITY_INSTANCE_T" to lock intent "X" has failed. The
SQLCODE is "-911"
```

Action to take:

- Increase the **`LOCKLIST`** value
  - For example, to set the value to 500, enter the following DB2 command:
    `db2 UPDATE DB CFG FOR BPEDB USING LOCKLIST 500`

Figure 10-52. Troubleshooting tips: Business Process Choreographer database on DB2 (2 of 3) WB8691.0

Notes:

WebSphere Education

Troubleshooting tips: Business Process Choreographer database on DB2 (3 of 3)

- To avoid deadlocks, make sure that your database system is configured to use sufficient memory, especially for the buffer pool
  – For DB2, use the *DB2 Configuration Advisor* to determine reasonable values for your configuration
- If you get errors with the data source implementation class **`COM.ibm.db2.jdbc.DB2XADataSource`**:
  – Check that the class path definition for your JDBC provider is correct
  – If Business Process Choreographer is configured on a server, check that the component-managed authentication alias is set to:
    `BPCDB_`*`nodeName.serverName`*`_Auth_Alias`
  – If Business Process Choreographer is configured on a cluster, check that the component-managed authentication alias is set to:
    `BPCDB_`*`clusterName`*`_Auth_Alias`

Figure 10-53. Troubleshooting tips: Business Process Choreographer database on DB2 (3 of 3) WB8691.0

Notes:

WebSphere Education IBM

Unit summary

Having completed this unit, you should be able to:

- Collect MustGather data for the Business Process Choreographer
- Enable logging and tracing to gather relevant data
- Examine the trace log to follow the navigation steps inside the business process engine
- Describe the Business Process Choreographer application installation architecture
- Describe some of the known pitfalls that are associated with Business Process Choreographer application installation
- Explain when business process and human task component validation takes place
- Determine the status of process instances by using Business Process Choreographer Explorer
- Schedule the deletion of completed process instances

Figure 10-54. Unit summary WB8691.0

Notes:

WebSphere Education IBM

Checkpoint questions

1. True or false: The IBM Process Server version has to be equal or higher than the EAR file version (IBM Integration Designer version) that you are attempting to install.

2. True or false: All the business process and human task instances must be completed before uninstalling the application.

3. Which one of the statements is true about long-running business processes?
   a) A long-running process that is called synchronously can often cause hung threads.
   b) A long-running process should always invoke another long-running process asynchronously.
   c) A long-running process usually spans several transactions and needs a free thread to continue with its navigation.
   d) All of the above.

Figure 10-55. Checkpoint questions WB8691.0

Notes:

Write down your answers here:

1.

2.

3.

WebSphere Education

Checkpoint answers

1. True

2. True

3. (d) All of the above

Figure 10-56. Checkpoint answers WB8691.0

Notes:

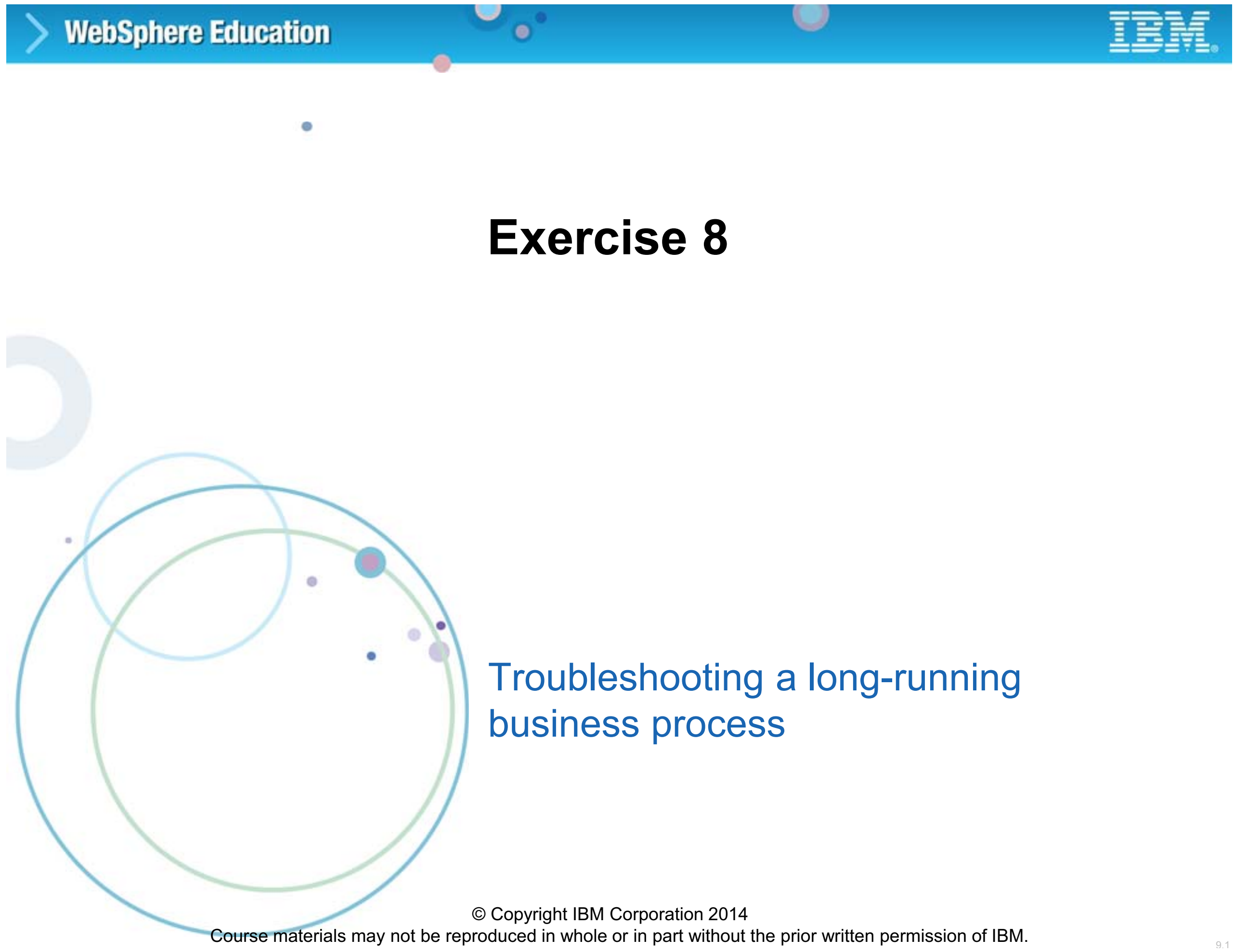


Figure 10-57. Exercise 8 WB8691.0

Notes:

WebSphere Education

Exercise objectives

After completing this exercise, you should be able to:

- Start the Business Process Choreographer Explorer to examine problems with running business process instances
- Check configuration properties
- Find business process instances that are related to a failed event
- Check the activity log and find business process-related messages to get additional information

Figure 10-58. Exercise objectives WB8691.0

Notes:

Unit 11. Troubleshooting the advanced deployment environment

What this unit is about

This unit covers troubleshooting the advanced deployment environment and Advanced Integration services (AIS).

What you should be able to do

After completing this unit, you should be able to:

- Describe the procedure to troubleshoot a deployment environment
- Explain the problem and resolution of troubleshooting Advanced Integration services (AIS)
- Learn some fundamental good practices when using IBM Integration Designer, IBM Process Designer, and IBM Process Center together to create applications
  - Describe a facade pattern
  - Explain the feature of library mirroring
  - Explain how to check for breakages

How you will check your progress

- Checkpoint

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Describe the procedure to troubleshoot a deployment environment
- Explain the problem and resolution of troubleshooting Advanced Integration services (AIS)
- Learn some fundamental good practices when using IBM Integration Designer, IBM Process Designer, and IBM Process Center together to create applications
  - Describe a facade pattern
  - Explain the feature of library mirroring
  - Explain how to check for breakages

Figure 11-1. Unit objectives WB8691.0

Notes:

WebSphere Education

Topics

- Troubleshoot the deployment environment
- Problem and resolution of troubleshooting AIS
- Fundamental good practices when using IBM Integration Designer, IBM Process Designer, and IBM Process Center together

Figure 11-2. Topics WB8691.0

Notes:

11.1. Troubleshoot the deployment environment

This topic introduces the MustGather data that is specific to the Business Process Choreographer. A sample Business Flow Manager (BFM) trace log is provided to help you analyze the event flow.

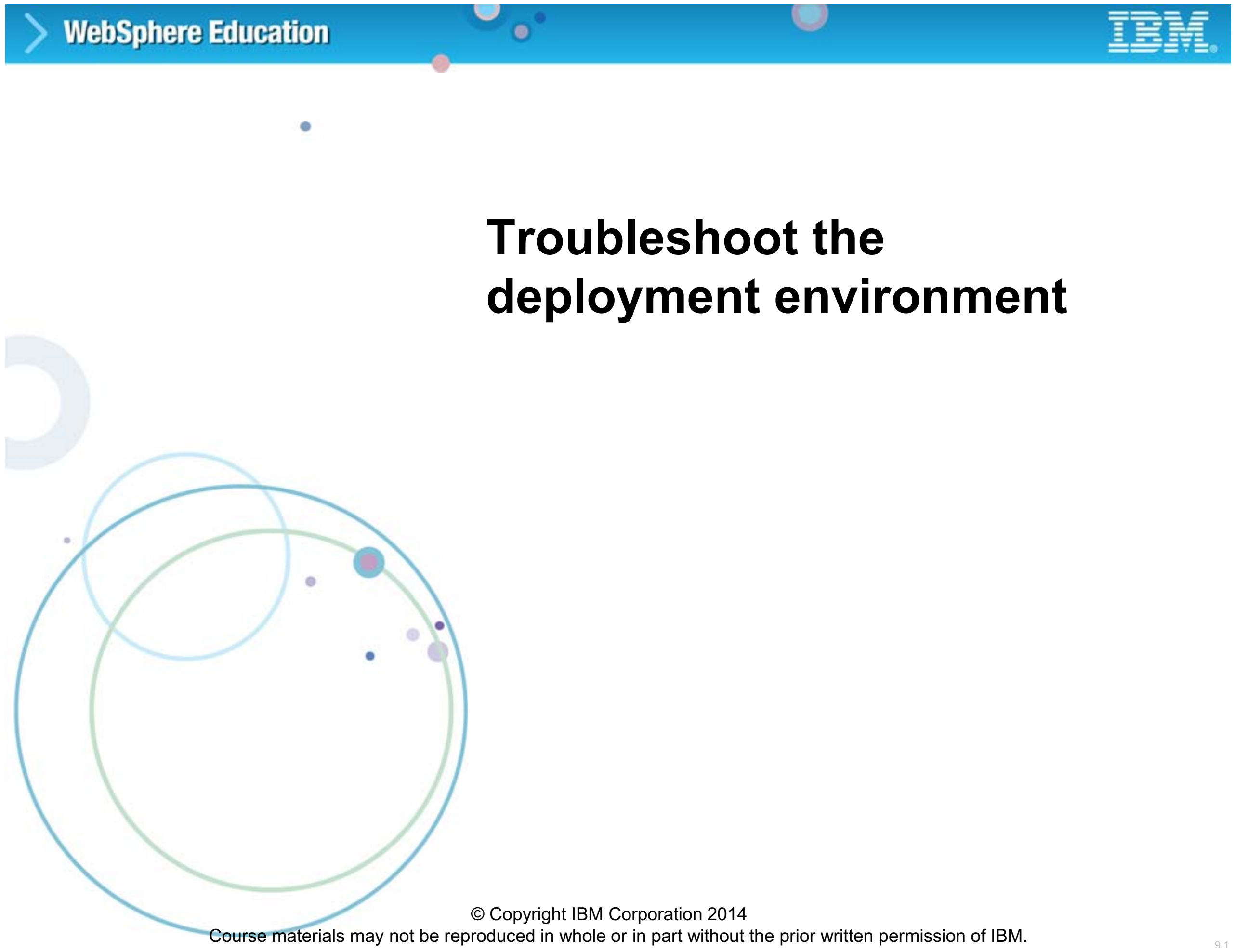


Figure 11-3. Troubleshoot the deployment environment WB8691.0

Notes:

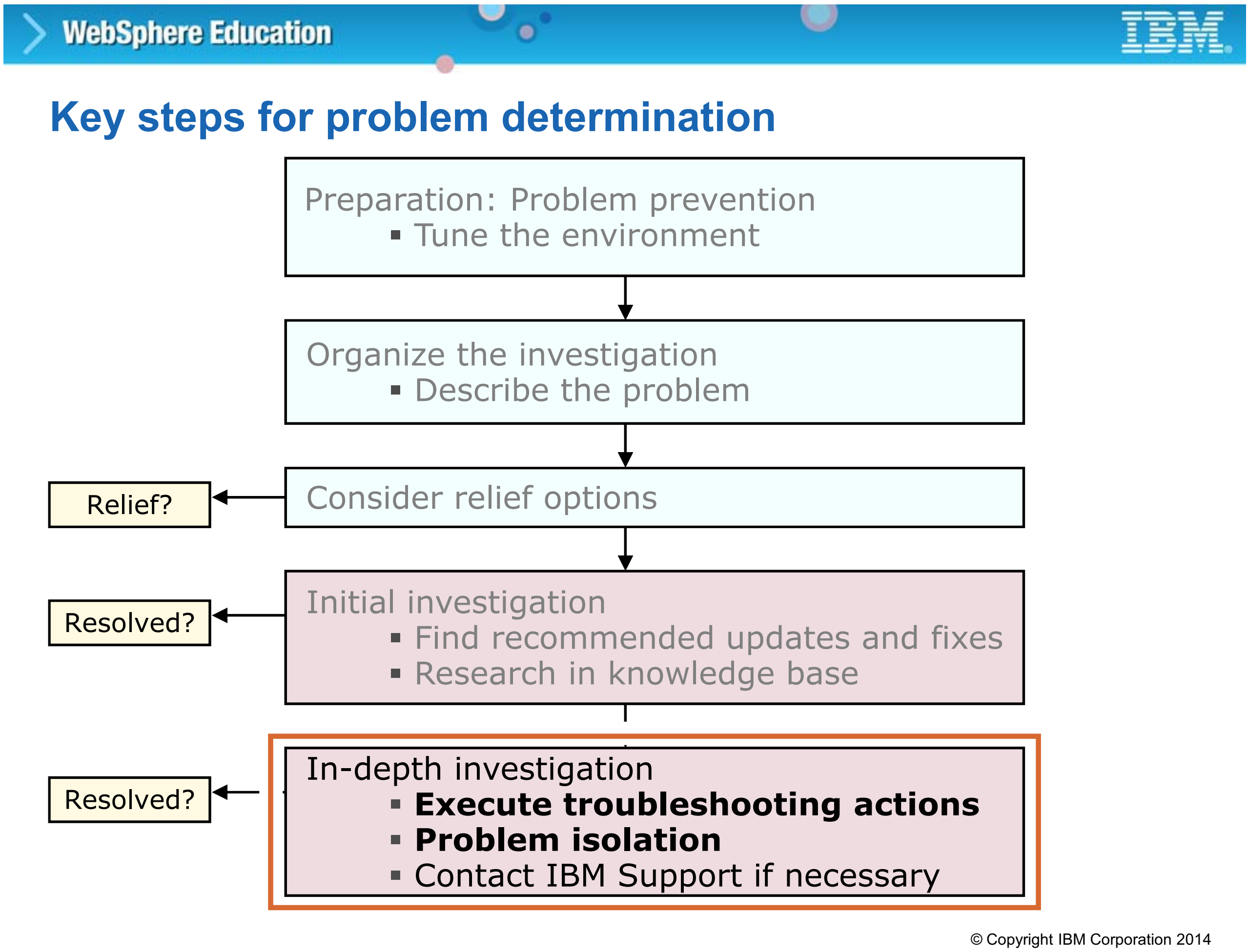


Figure 11-4. Key steps for problem determination WB8691.0

Notes:

WebSphere Education

Symptoms of the deployment environment

- Investigate the state of your deployment environment if you notice any of the following symptoms:
  - Unavailable applications
  - Sluggish applications
  - Stopped applications
  - Decreased throughput
  - Sluggish performance

Figure 11-5. Symptoms of the deployment environment WB8691.0

Notes:

WebSphere Education

Troubleshooting procedure (1 of 2)

This approach is for non-standalone server environments

1. Display the topology layout that describes the deployment environment to determine the status of the topology
2. Display the topology to determine the state of the various roles in the topology
3. Note the roles with unexpected states or warning for further investigation
4. Locate the nodes that are causing the error state for each role
5. Make sure that all nodes are synchronized
6. On the Nodes page in the administrative console, select any unsynchronized nodes and click **Synchronize**
7. Make sure that the messaging engines that are associated with all the buses are running
8. If they are not running, stop and start the messaging engines

Figure 11-6. Troubleshooting procedure (1 of 2) WB8691.0

Notes:

When the processing appears to be sluggish or the requests fail, you can use a focused approach to determine the source of the problem in the environment. This approach is for non-standalone server environments.

To do this task, you must log in to the administrative console of the deployment manager, and you must log in as an administrator or configurator.

WebSphere Education

Troubleshooting procedure (2 of 2)

9. Locate the logs that are associated with the nodes in error, and view the logs for error messages
10. Perform any actions that the error messages prescribe to correct the errors
11. Correct any errors and restart the affected nodes
12. The nodes are started successfully and the status of the topology is “running”
13. Restart any applications that are affected

Figure 11-7. Troubleshooting procedure (2 of 2) WB8691.0

Notes:

11.2. Problem and resolution of troubleshooting AIS

This topic introduces the MustGather data that is specific to the Business Process Choreographer. A sample Business Flow Manager (BFM) trace log is provided to help you analyze the event flow.

WebSphere Education IBM

Problem and resolution of troubleshooting AIS

9.1

Figure 11-8. Problem and resolution of troubleshooting AIS WB8691.0

Notes:

WebSphere Education IBM

Advanced Integration service

- An Advanced Integration service (AIS) is used to call a service that is implemented in IBM Integration Designer from one of the following methods:
  - BPD with a system task
  - Another service with a nested service
- A collaboration between a business user who works with IBM Process Designer and an integration developer who works with IBM Integration Designer
- Available only with IBM Business Process Manager Advanced
- Implementation of a user task or a system task
  - User task: Assign as specified with the assignments of the user task
  - System task: Run by the system user

Figure 11-9. Advanced Integration service WB8691.0

Notes:

AIS is a collaborative arrangement. If you move your Advanced Integration service to another toolkit, you need to notify the integration developer who implemented your service. Your service and its implementation by Integration Designer are decoupled. It means that even though you might move a service in Process Designer, there is no automatic corresponding movement in the implementation by Integration Designer.

IBM Business Process Manager Advanced offers two authoring environments. IBM Process Designer is used to model and execute high-level business processes, which often have human interactions. IBM Integration Designer is used to build and implement services that are automated or that invoke other services such as web services, enterprise resource applications, or CICS applications and IMS, which exist in the enterprise. These authoring environments both interact with the Process Center, which is a shared repository and runtime environment.

There are two roles and skill sets to consider when developing business process management (BPM) applications with these environments:

- The **business author** is responsible for authoring all business processes. The business author is able to use services, but is not interested in the implementation details or how they work. The business author uses Process Designer to create business process definitions (BPDs), and Advanced Integration services (AISs) to collaborate with the integration programmer.

- The **integration programmer** is responsible for doing all of the integration work necessary to support the processes the business author creates. For example, integration programmer implements all the AISs, and produces mappings between backend formats and the requirements of current applications. The integration programmer uses Integration Designer.

WebSphere Education

Example of Advanced Integration service

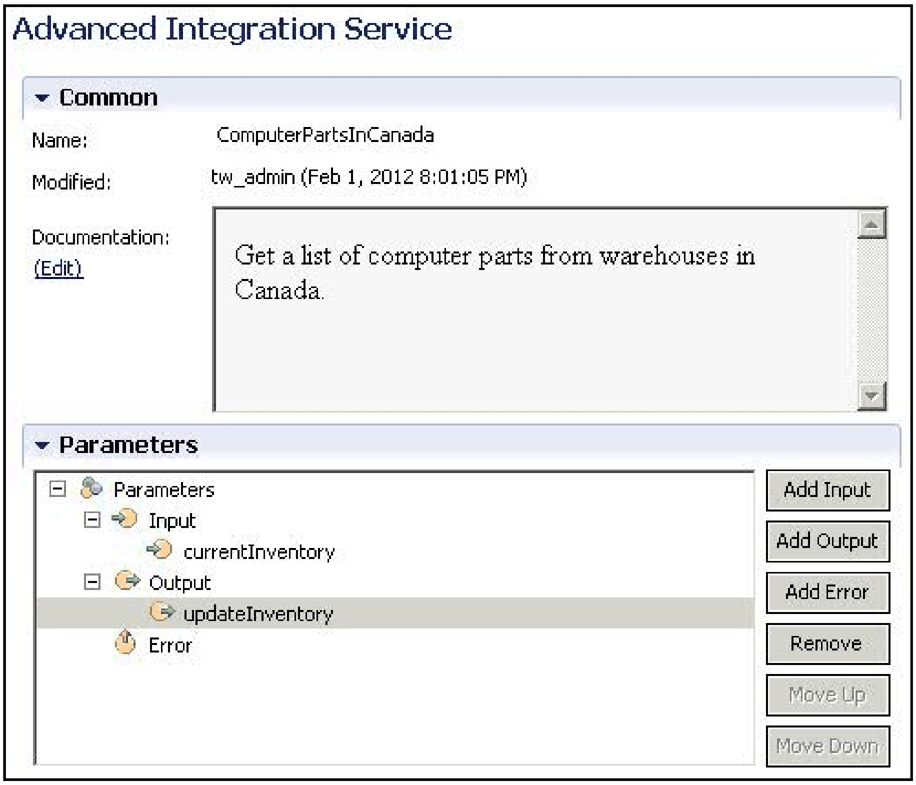


Figure 11-10. Example of Advanced Integration service WB8691.0

Notes:

For example, your business process might need a list of hotels in Canada. Checking with an integration developer, you realize that a service is being built in Integration Designer to query the Canadian hotel and return an inventory list of the Canadian hotels available. You can create an AIS that uses this Integration Designer service as an activity in your business process.

WebSphere Education

Troubleshooting: AIS (1 of 2)

- **Problem: AIS does not refresh automatically in the Inspector view**
  – When you run a business process diagram (BPD) in IBM Business Process Manager Advanced, the process status does not automatically update in the Inspector view
  – An Advanced Integration service (AIS) can take some time to run, and it depends on the service implementation
  – Even fast request and response services can seem to take a long time for the first time that they are called, and the service is initializing
- **Resolution:**
  – If the process state in the Inspector view does not seem to progress, click the **Refresh** icon and wait for the AIS call to complete and the BPD to resume
  – Click the **Refresh** icon again if necessary
  – If the AIS still does not complete, check with its implementation owner
  – The AIS might be waiting for something, such as a human task or an error to be fixed

Figure 11-11. Troubleshooting: AIS (1 of 2) WB8691.0

Notes:

Troubleshooting: AIS (2 of 2)

- **Problem: AIS does not participate in the same transaction as business process**
  - In IBM Business Process Manager Advanced, process navigation of business process definitions (BPDs) does not participate in the same transaction context as an Advanced Integration service (AIS)
  - A runtime failure in the BPD navigation that causes the BPD transaction to roll back does not roll back the transaction under which the currently executing AIS might be running
  - As a result, the AIS might be executed a second time
- **Resolution:**
  - BPD process navigation is not part of the same transaction as an invoked AIS
  - If it is a runtime failure (not a logic error), the process navigation becomes unavailable while the AIS is running
  - Place checks within the AIS application logic to ensure that a second invocation does not corrupt the state of an application

Figure 11-12. Troubleshooting: AIS (2 of 2) WB8691.0

Notes:

BPD process navigation is not part of the same transaction as an invoked AIS. If a BPD process navigation invokes an AIS, and then experiences a runtime failure (not an error in the business process logic), the process navigation becomes unavailable while the AIS is running.

When the AIS commits its transaction, it cannot indicate to the process navigation that the AIS concluded. When the process navigation resumes later, the AIS is invoked again because the process navigation is unaware of the previously successful invocation.

11.3. Fundamental good practices when using IBM Integration Designer, IBM Process Designer, and IBM Process Center together

This topic introduces the MustGather data that is specific to the Business Process Choreographer. A sample Business Flow Manager (BFM) trace log is provided to help you analyze the event flow.

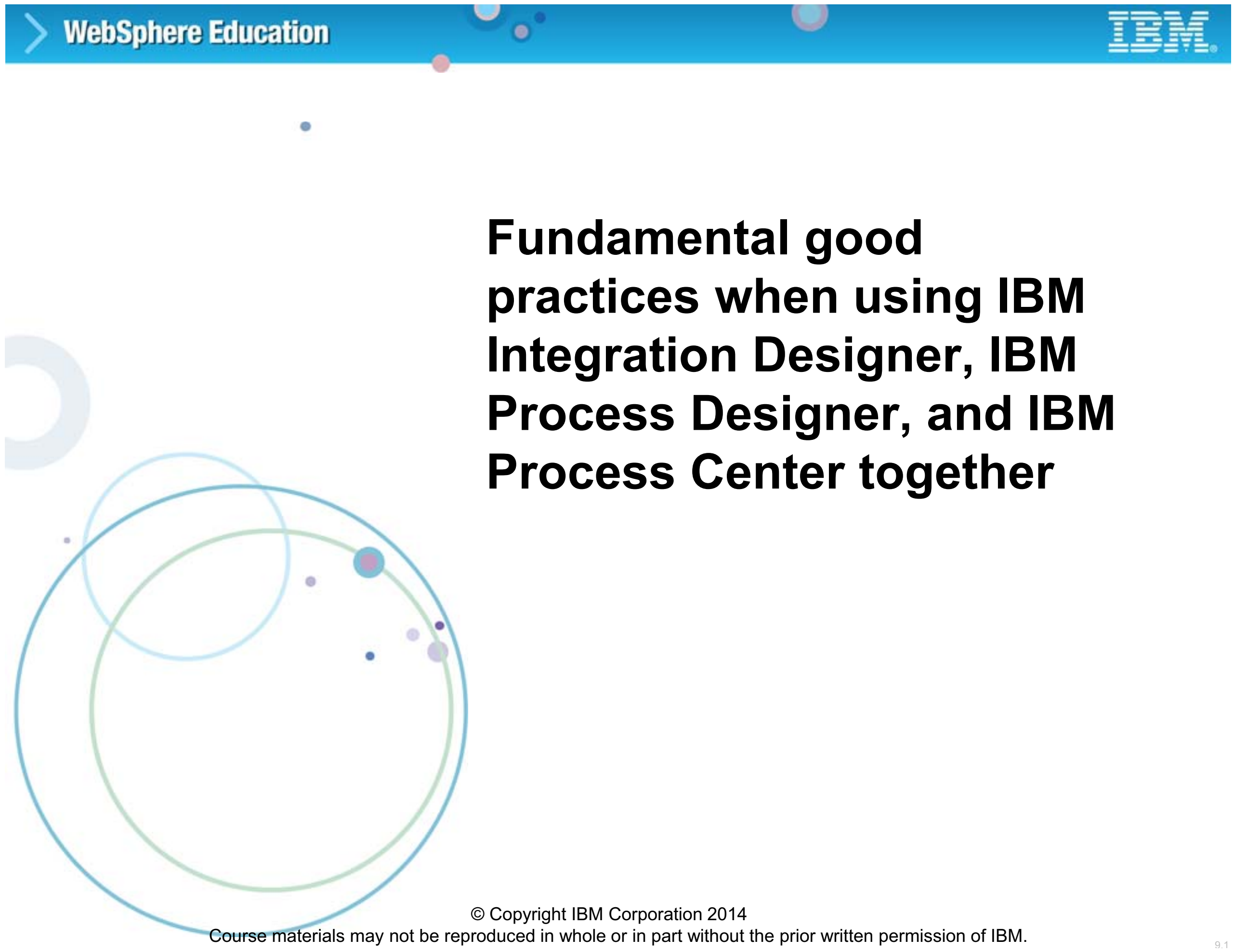


Figure 11-13. Fundamental good practices when using IBM Integration Designer, IBM Process Designer, and IBM Process Center together WB8691.0

Notes:

WebSphere Education IBM

Repository details and canonical models

- Process Designer works live from the Process Center repository artifacts
- Integration Designer has local workspaces with physical artifacts and files with the ability to work offline
  - Integration Designer contributes modules, libraries, and other projects to process applications or toolkits
  - Contains business process definitions (BPDs), AISs, and other artifacts that the business author provides
- Working with a canonical data model in the enterprise
  - Canonical model: A design pattern to communicate between different data formats
  - Reduce the amount of rework later in the iterative development process

Figure 11-14. Repository details and canonical models WB8691.0

Notes:

A major difference between the two products is that Process Designer works live from the Process Center repository artifacts, whereas Integration Designer has local workspaces with physical artifacts and files, allowing you to work offline. Integration Designer contributes modules, libraries, and other projects to process applications or toolkits, which contain BPDs, AISs, and other artifacts that the business author provides.

Sometimes the integration programmer needs to deal with a canonical data model in the enterprise. In this case, the business author must take this issue into consideration to reduce the likelihood of rework later in the iterative development process.

WebSphere Education

Collaborate before defining an AIS

- Advanced Integration services (AISs) might exist in both process applications and toolkits
- Implementation of an AIS must coexist in the same process application or toolkit as its definition
- Important for reuse and organization considerations
- Contact the integration programmer to confirm where the AIS is defined, and implemented
- AIS is shared among many process applications that the business author is best to define the AIS in a toolkit
- Group similar AISs in a specific toolkit
- Consult on the structure before the business author creates and defines a new AIS

Figure 11-15. Collaborate before defining an AIS WB8691.0

Notes:

For example, consider this scenario. The business author needs a particular service that an integration programmer implements. The business author can define the AIS in the process application or toolkit in which the business author is working. However, more care can be taken. The business author needs to contact the integration programmer to confirm where the AIS should be defined, and thus implemented. It is likely that the integration programmer wants to have the AIS shared among many process applications, in which case the business author is best to define the AIS in a toolkit. The integration programmer is best to group similar AISs in a specific toolkit. For these reasons, the integration programmer needs to consult about the structure before the business author creates and defines a new AIS.

WebSphere Education

Connect to only one Process Center from an Integration Designer workspace

- Multiple Process Centers connect to Integration Designer workspace for the following reasons:
  - Import project interchange files
  - Import projects from a source control repository, such as CVS or Rational Team Concert
  - Change the Process Center URL and credentials in the Process Center preference page after opening a process application or toolkit
- Connecting to many Process Centers results in cumbersome user interactions
  - Go to the preference page to change the URL and credentials
  - Leads to confusion as dependencies cannot be made across Process Center servers
- Conclusion: Connect to only one Process Center from an Integration Designer workspace

Figure 11-16. Connect to only one Process Center from an Integration Designer workspace WB8691.0

Notes:

Connections to Process Center servers from Integration Designer are automatically established according to the projects contained in the workspace. It is possible to have process applications and toolkits from different Process Centers in the same Integration Designer workspace for the following reasons:

- Import project interchange files
- Import projects from a source control repository, such as CVS or Rational Team Concert
- Change the Process Center URL and credentials in the Process Center preference page after opening a process application or toolkit

Integration Designer continues to operate normally when the workspace is connected to multiple Process Centers. However, you need to avoid this action because there is only one primary Process Center for an Integration Designer workspace. The primary Process Center is the one captured in the Process Center preference page. This primary Process Center is also the one that is in the Process Center perspective.

Actions on process applications and toolkits are possible only in the Process Center perspective, and they include:

- Opening process applications and toolkits in Integration Designer workspace
- Creating new tracks
- Managing snapshots
- Exporting process applications and toolkits as `.twx` files
- Managing permissions

You use one workspace to connect to many Process Centers, and it results in cumbersome user interactions. This result is because the integration programmer needs to go to the preference page to change the URL and credentials to use the previously stated actions in the Process Center perspective. If you have process applications and toolkits in the workspace from different Process Centers, it can lead to confusion because dependencies cannot be made across Process Center servers. For example, a process application in one server cannot depend on a toolkit in a different server.

WebSphere Education IBM

Business Integration view mode

- Set the level of detail for a process application or toolkit in the Business Integration view
  - Simple mode and detailed mode
- Simple mode
  - The complexities of the SCA model, modules, and libraries are hidden
  - All the data types and interfaces are aggregated to provide a similar look to the Process Designer
  - Drawback: Only one component can implement AIS
- Detailed mode
  - The Integration Designer projects are shown explicitly
  - The artifacts are located under their respective containers
  - Use the Detailed mode if you need to work with the assembly diagram to create more complex services

Figure 11-17. Business Integration view mode WB8691.0

Notes:

WebSphere Education

Have only what you need in Integration Designer

- Do not initialize the dependent toolkits unless you plan to have Integration Designer contributions
  - Call a BPD
  - Provide implementation for an AIS
  - Attach a monitor model
- Do not initialize unnecessary toolkits
  - Publish a newly initialized toolkit, and the toolkit is considered changed
  - Necessary to create a snapshot
  - Dependencies from process applications and toolkits must be updated to the new snapshot level
  - After you open the process application or toolkit in the workspace, do not select the toolkits to avoid initialization
  - Disassociating all projects can uninitialize a toolkit
- Select a subset of modules and libraries that are relevant to your current work to bring into the workspace
  - Reduce build time and prevent unnecessary network operations

Figure 11-18. Have only what you need in Integration Designer WB8691.0

Notes:

Integration Designer can open a process application or toolkit, which needs to be initialized with a default module and a default library. The process application or toolkit is initialized automatically only one time before it first opens in the Integration Designer workspace. However, it is important not to initialize the dependent toolkits unless you plan to have some Integration Designer contributions; for example, call a BPD, provide implementation for an AIS, or attach a monitor model. It is important not to initialize unnecessary toolkits because after you publish a newly initialized toolkit, the toolkit is considered changed. You need to create a snapshot, which means that dependencies on this toolkit, from process applications and toolkits, must be updated to the new snapshot level. You can avoid initializing a toolkit by not selecting it from the dialog that appears after opening the process application or toolkit in the workspace. You can disassociate all projects to uninitialize a toolkit.

For more information about disassociation of projects, see “Disassociating a module or library from a process application or toolkit” in the IBM Business Process Manager V8.5 Information Center: `http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r5m0/topic/com.ibm.wbpm.auth.stp.doc/processcenter/topics/tremoveasscmodlib.html`

Another consideration is that as a process application or toolkit grows over time, it contains multiple modules and libraries. When you open a process application or toolkit in Integration Designer, you can select a subset of modules and libraries that are relevant to your current work to your workspace. This action reduces build time and prevents unnecessary network operations between the Integration Designer and the Process Center.

WebSphere Education

Associate with process application or toolkit

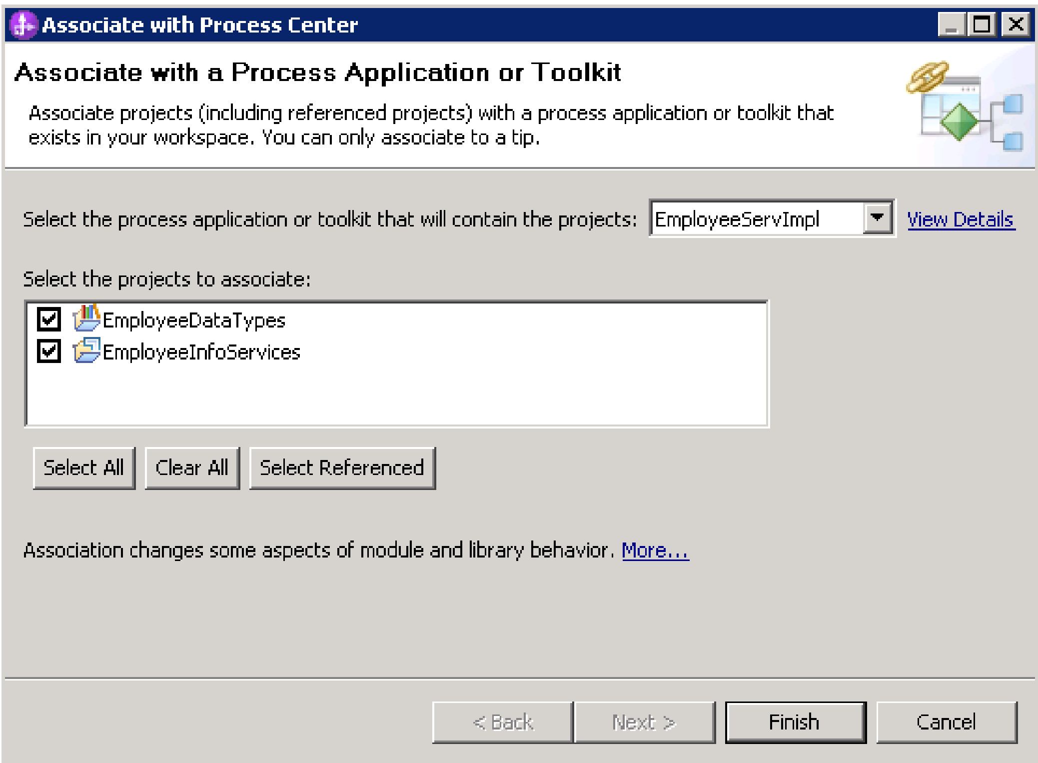


Figure 11-19. Associate with process application or toolkit WB8691.0

Notes:

After you associate the projects with the process application or toolkit, you create a snapshot of the toolkit in Process Center. You also change the permissions on the toolkit, so only a business author who has appropriate access can view and edit the toolkit.

WebSphere Education

Protect mirrored artifacts in toolkits

- Data types are shared artifacts between Process Designer and Integration Designer
- Data types are stored in two forms in Process Center:
  - Process Designer format (proprietary)
  - Integration Designer format (XSD files)
- Translation mechanism keeps data types that are synchronized
- Integration Designer format (XSD) contains structures that cannot be represented in the Process Designer format
- Data types that are authored in Integration Designer and published to Process Center, and modified in Process Designer
  - Original data type format in Integration Designer significantly different
  - Possible loss of modeled information
- Protect them from the loss of modeled information

Figure 11-20. Protect mirrored artifacts in toolkits WB8691.0

Notes:

Data types are artifacts that are shared between Process Designer and Integration Designer. Business authors and integration programmers can change and update these artifacts, and those changes are visible to both of them. To support this capability, data types are stored in two forms in Process Center. These formats are Process Designer format (proprietary) and Integration Designer format (XSD files). After you update a data type, the artifact is updated in both formats with a translation mechanism that keeps them synchronized. When an integration programmer publishes resources to Process Center, the translation occurs and modifies, creates, and deletes the artifacts in the Process Designer format. The business author sees the changes after the publish operation is complete. Conversely, if a business author changes a data type, the translation occurs immediately after the creation, deletion, or editor is saved. An integration programmer sees these changes the next time that the programmer refreshes the workspace.

It is important that these formats have different characteristics. They have different fidelity for the details that might be present in the other format. The Integration Designer format (XSD) contains structures that cannot be represented in the Process Designer format. If you author a data type in Integration Designer, publish to Process Center, and modify in Process Designer, the original data type format in Integration Designer might be significantly different. It is possible to lose the modeled information.

WebSphere Education

Protect from the loss of modeled information

- Set user permissions on process applications and toolkits
- To protect the Integration Designer data types
  - Place them inside a library
  - Associate the library with a toolkit with permissions
  - Prevent business authors to change
- In the Process Center perspective
  - Set the permissions with only read access for that business authors
- Integration programmers are free to modify the data types, and business authors see the updated data types in Process Designer
- Business authors cannot change the data types in Process Designer
  - Avoiding any translation that might result in loss of modeled information

Figure 11-21. Protect from the loss of modeled information WB8691.0

Notes:

It is desirable to protect data types that originate in Integration Designer, whether they are authored or imported from the loss of modeled information. You can use Process Center to set user permissions on process applications and toolkits. Therefore, to protect the Integration Designer data types, you can place them inside a library, and associate the library with a toolkit with permissions that prevent business authors from changes. In the Process Center perspective, the permissions can be changed so that business authors are given only read access, instead of write access. With this technique, integration programmers are free to modify the data types, and business authors see the updated data types in Process Designer. However, the business authors are unable to change the data types in Process Designer, avoiding any translation that might result in loss of modeled information.

WebSphere Education

Protect data types with library mirroring

- Protect data types with the library mirroring preference
- Data types are not referenced directly or indirectly by an AIS implementation
  - Leave the library that is associated with the process application or toolkit as unmarked for mirroring
- Default library is always mirrored
  - Artifacts are translated to the Process Designer artifact domain
- All other libraries are not marked as mirrored by default
  - Artifacts are not visible in Process Designer
- A library is marked as mirrored only when it is necessary
- Business author cannot see the data types at all
  - Do not use this method if the business author needs to see the data types
- Using either approach greatly reduces the likelihood of data types that a business author modifies in Process Designer, which can cause a lower-fidelity model in Integration Designer

Figure 11-22. Protect data types with library mirroring WB8691.0

Notes:

You can protect data types when you set to not mirror them at all. If the data types are not referenced directly or indirectly by an AIS implementation, then the library that is associated with the process application or toolkit is left as unmarked for mirroring. The default library is always mirrored, and it means that the artifacts are translated to the Process Designer artifact domain. All other libraries are not marked as mirrored by default. You can see them in the properties dialog for the library, which means that the artifacts are not visible in Process Designer. A library can be marked as mirrored only when it is necessary. With this technique, the business author cannot see the data types at all, so you cannot use this method if the business author needs to see the data types.

Using either approach greatly reduces the likelihood of data types that a business author modifies in Process Designer, causing a lower fidelity model in Integration Designer.

WebSphere Education

Facade from edge interfaces and data types

- To avoid excessive or accidental breakages
  - "Facade" the data types and the interfaces
  - Isolate the models from changes that are introduced through the other tools
- An input variable has reference to a data type to a business process definition in Process Designer
- A BPEL process uses the same data type with Java snippets and in a state machine in Integration Designer
- If the business author decides to delete the type, or move it to another toolkit, then the BPEL process, Java snippets, and state machine in Integration Designer are all broken
- A facade pattern prevents these artifacts from being broken
  - Only the facade logic, such as a data map, is broken
  - The breakage does not occur within the business logic of the BPEL, the snippets, or the state machine
  - Easier and less error prone to adjust the facade

Figure 11-23. Facade from edge interfaces and data types WB8691.0

Notes:

Data types are artifacts that are shared between Process Designer and Integration Designer. After one tool changes a data type, the definition of the data type in the other format is updated with the changes.

Data types are likely to be used in many other artifacts in both Integration Design and Process Designer. Changes made to the data types through one tool are likely to break artifacts that are defined in the other tool. A good pattern to avoid excessive or accidental breakages is to "facade" the data types and the interfaces, isolating the models from changes that are introduced through the other tool. For example, suppose that a data type is referenced in an input variable to a business process definition in Process Designer. Further, suppose that the same data type is used in a BPEL process, some Java snippets, and in a state machine in Integration Designer. If the business author decides to delete the type, or move it to another toolkit, then the BPEL process, Java snippets, and state machine in Integration Designer are all broken. However, if a facade pattern is used, these artifacts would not be broken at all. Instead, only the facade logic, such as a data map, would be broken. Since the breakage does not occur within the business logic of the BPEL, the snippets, or the state machine, it is easier and less error prone to adjust the facade. The facade pattern is shown in detail in part four of the sample.

WebSphere Education

Facade pattern

- To avoid excessive or accidental breakages
  - "Facade" the data types and the interfaces
  - Isolate the models from changes that are introduced through the other tools
- An input variable has reference to a data type to a business process definition in Process Designer
- A BPEL process uses the same data type with Java snippets and in a state machine in Integration Designer
- If the business author decides to delete the type, or move it to another toolkit, then the BPEL process, Java snippets, and state machine in Integration Designer are all broken

Figure 11-24. Facade pattern WB8691.0

Notes:

For more reference about facade pattern, go to: `http://www.ibm.com/developerworks/websphere/bpmjournal/1112_pacholski/1112_pacholski.html`

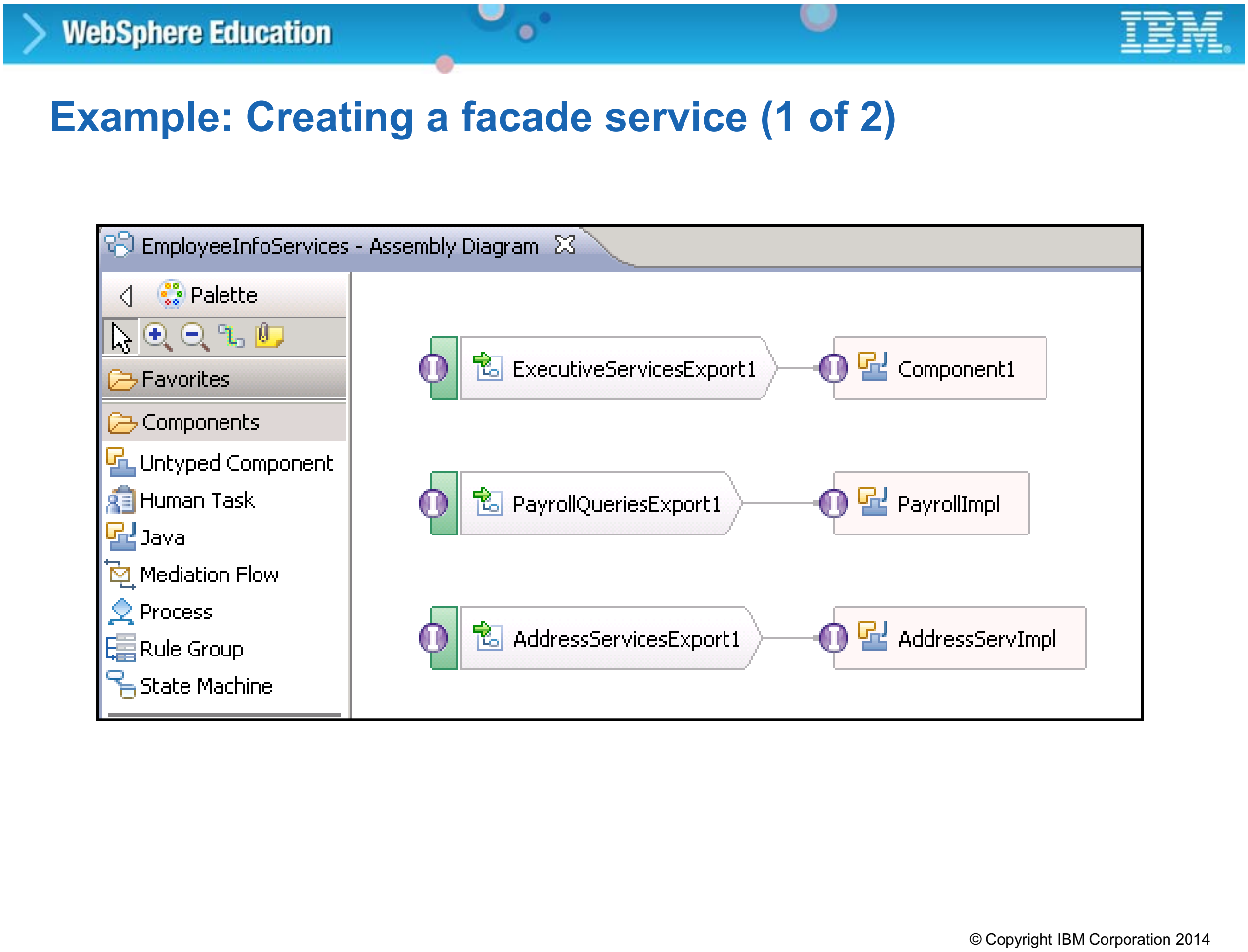


Figure 11-25. Example: Creating a facade service (1 of 2) WB8691.0

Notes:

In this example, there are three operations on the **ExecutiveServices** interface, yet the integration programmer wants to make only one of them visible to business authors. The integration programmer does not want business authors to see the **AddressServices** or **PayrollQueries** services.

To complete the facade pattern, the integration programmer implements a simple mediation flow in EmployeeServices to mediate from the public interfaces and data types to the existing service. The figures show the assembly diagram and mediation flow implementation.

WebSphere Education IBM

Example: Creating a facade service (2 of 2)

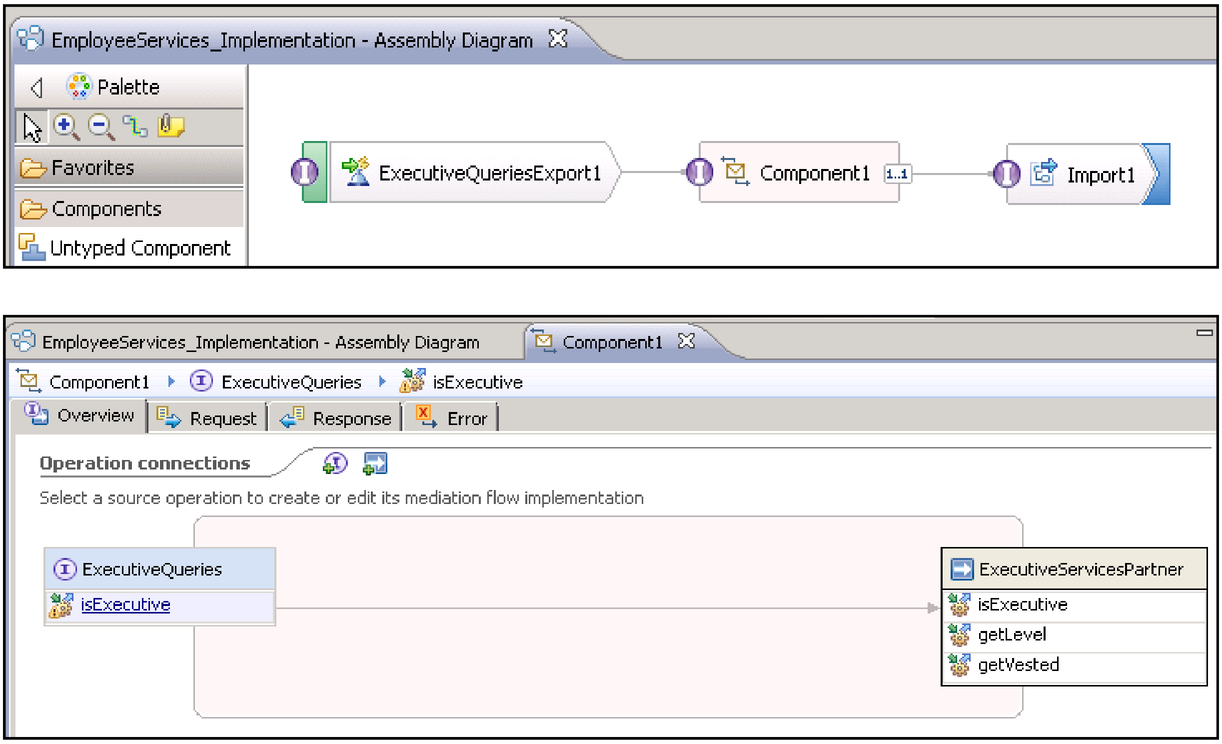


Figure 11-26. Example: Creating a facade service (2 of 2) WB8691.0

Notes:

To complete the facade pattern, the integration programmer implements a simple mediation flow in **EmployeeServices** to mediate from the public interfaces and data types to the existing service. Then, the integration programmer creates a snapshot of **EmployeeServices** and creates a dependency from the main process application to **EmployeeServices**. The integration programmer implements the AIS with the facade service. In this example, a BPEL process is used for the AIS implementation. The business author is free to modify the artifacts in **EmployeeServices,** and the underlying service in **EmployeeServImpl** is not affected.

WebSphere Education

Work on a process application and its dependent toolkit at the same time

- Tips, tracks, and snapshots
  - A track in Process Center is similar to branches or streams in other repository systems
  - One or more tracks can exist for a specific process application or toolkit
  - The latest content of a track is called the *tip*, and all changes must be made on the tip
  - At any point in time, a snapshot can capture the current state of a track
- Private changes
  - Process applications and toolkits with read-only access are disabled in the business integration view and they are immutable in Process Center
  - Snapshots are also immutable in Process Center
  - You can change the projects in Integration Designer; however, you cannot publish the changes to Process Center

Figure 11-27. Work on a process application and its dependent toolkit at the same time WB8691.0

Notes:

Unlike Process Designer, Integration Designer allows you to work on multiple process applications and toolkits at the same time. You can open multiple process applications or toolkits in the workspace. You can bring the dependent toolkits into a workspace automatically when a process application is opened. A process application can depend on a particular snapshot of a toolkit, and the snapshots are read-only. When you want to change both process application and toolkits, you must make sure to work with the “tip” of each track in the workspace.

Integration Designer monitors any changes to the snapshot of a process application or toolkit. If a process application or toolkit is changed, but the tip of the track is not in the workspace, a dialog is presented that the changes cannot be saved to the Process Center. You must bring in the tip (the most recent content of a track) if the changes are intended to be eventually committed.

When you change a process application and its dependent toolkits, you must open the tips of the tracks in the workspace. If a snapshot of a dependent toolkit is in the workspace, it can be replaced with the newest from the tip by using the menu. Only toolkits that are modified must have the track tip that is loaded in the workspace. When changes are published to the Process Center, Integration Designer recognizes that a dependent toolkit is in the workspace and that the tip of its track is present. As a result, a new snapshot of the toolkit is created and all dependencies are updated with the new snapshot value. Even if there is no change to the tip, a dialog still appears. This step is necessary to ensure that consistency of the dependency model exists across Process Center and the Integration Designer workspace. In Process Center, a dependency on a toolkit is always specified with a snapshot of the toolkit. It is not possible to have a dependency on the tip of a toolkit track.

WebSphere Education

Check for breakages in both tools

- Causes for unintended or unnoticed breakages
  - Set of artifacts visible to a business author are different from the artifacts that the integration programmer sees
  - A mediation flow that is authored in Integration Designer is not visible to business author
  - Some artifacts are visible to both roles, such as data types, business process definitions, and Advanced Integration services
- Dependencies between process applications and toolkits that make artifact visible across project dependencies
  - A project dependency in Integration Designer must be resolvable in the scope of the process application or toolkit
  - If a module in a process application depends on a library, the library must be in the same process application or in a dependent toolkit
- If both Integration programmers and business authors change these artifacts or dependencies
  - It causes an adverse impact on an artifact that is visible only to the other role
- Open and analyze in both tools and check any errors in the Problems view

Figure 11-28. Check for breakages in both tools WB8691.0

Notes:

If a process application or toolkit contains content that is authored in both Process Designer and Integration Designer, you must be cautious to avoid unintended or unnoticed breakages.

The set of artifacts that are visible to a business author are different from the artifacts that the integration programmer sees. For example, a mediation flow that is authored in Integration Designer is never visible to business authors. However, some artifacts are visible to both roles, such as data types, business process definitions, and Advanced Integration services.

In addition, there are dependencies between process applications and toolkits that make artifacts visible across project dependencies. For example, a project dependency in Integration Designer must be resolvable in the scope of the process application or toolkit. If a module in a process application depends on a library, the library must be in the same process application or in a dependent toolkit. Both Integration programmers and business authors can change these artifacts or dependencies, and it causes an adverse impact to an artifact that is visible only to the other role.

To ensure that an entire process application or toolkit has no errors, you must open and analyze it in both tools. Using a new workspace in Integration Designer, you select all modules and libraries when opening the process application or toolkit. After the build is complete, you can check any errors in the Problems view.

WebSphere Education IBM

Unit summary

Having completed this unit, you should be able to:

- Describe the procedure to troubleshoot a deployment environment
- Explain the problem and resolution of troubleshooting Advanced Integration services (AIS)
- Learn some fundamental good practices when using IBM Integration Designer, IBM Process Designer, and IBM Process Center together to create applications
  - Describe a facade pattern
  - Explain the feature of library mirroring
  - Explain how to check for breakages

Figure 11-29. Unit summary WB8691.0

Notes:

WebSphere Education

Checkpoint questions

1. True or false: An Advanced Integration service (AIS) is available with both IBM Business Process Manager Standard and Advanced editions.

2. True or false: A facade pattern is to isolate the models from changes that are introduced through the other tools.

3. Data types are shared artifacts and are stored in two forms in Process Center. They are Process Designer format (proprietary) and which of the following formats?
   a) Integration Designer format (XSD files)
   b) XML format
   c) SQL format
   d) Process Center format

Figure 11-30. Checkpoint questions WB8691.0

Notes:

Write down your answers here:

1.

2.

3.

WebSphere Education IBM

Checkpoint answers

1. False. It is available only with IBM Business Manager Advanced.

2. True.

3. (a) Integration Designer format (XSD files).

Figure 11-31. Checkpoint answers WB8691.0

Notes:

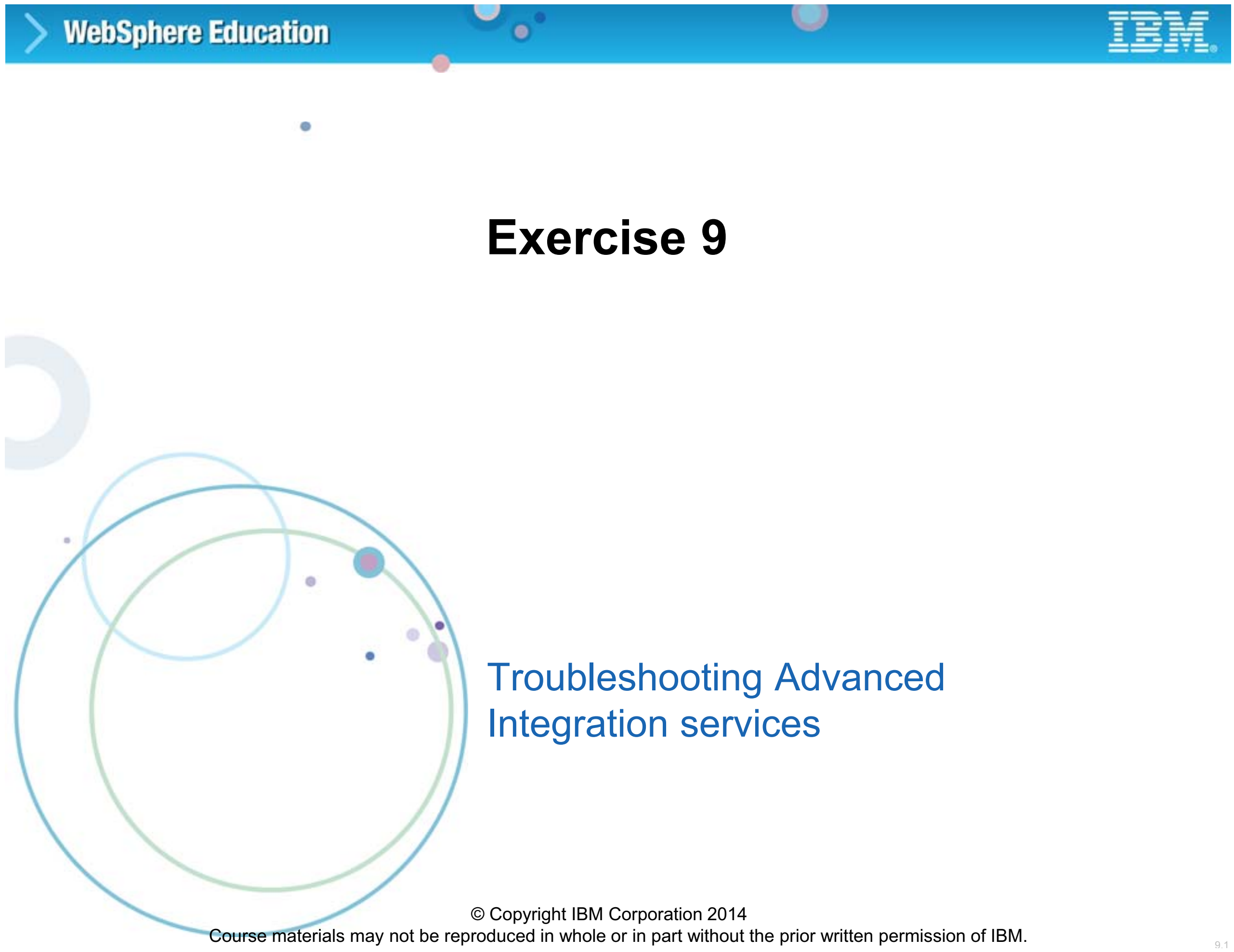


Figure 11-32. Exercise 9 WB8691.0

Notes:

WebSphere Education IBM

Exercise objectives

After completing this exercise, you should be able to:

- Change the version of dependency after migration from the previous version
- Collaborate before defining an Advanced Integration service for missing data objects
- Create a facade service
- Troubleshoot a broken map
- Protect mirrored artifacts in toolkits

Figure 11-33. Exercise objectives WB8691.0

Notes:

Unit 12. Business Process Manager performance problems

What this unit is about

This unit covers the high level of IBM BPM performance-related troubleshooting.

What you should be able to do

After completing this unit, you should be able to:

- Describe the nature of the performance problem
- Describe the types of data that is needed to troubleshoot performance problems

How you will check your progress

- Checkpoint

Unit objectives

After completing this unit, you should be able to:

- Describe the nature of the performance problem
- Describe the types of data that is needed to troubleshoot performance problems

Figure 12-1. Unit objectives WB8691.0

Notes:

For an in-depth education on BPM performance and tuning, see course WB868: *IBM Business Process Manager Advanced V8.5 Performance and Tuning*.

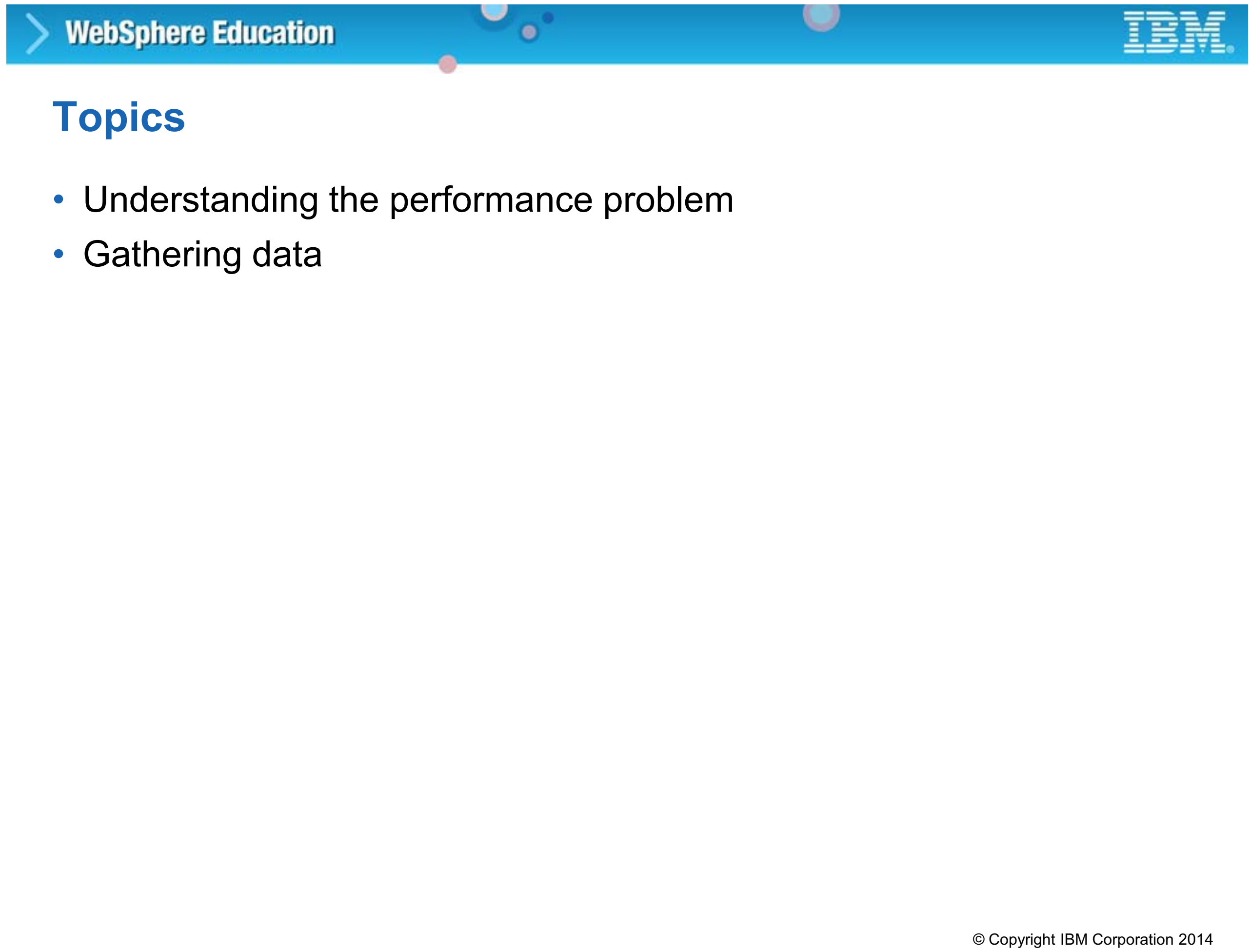


Figure 12-2. Topics WB8691.0

Notes:

12.1.Understanding the performance problem

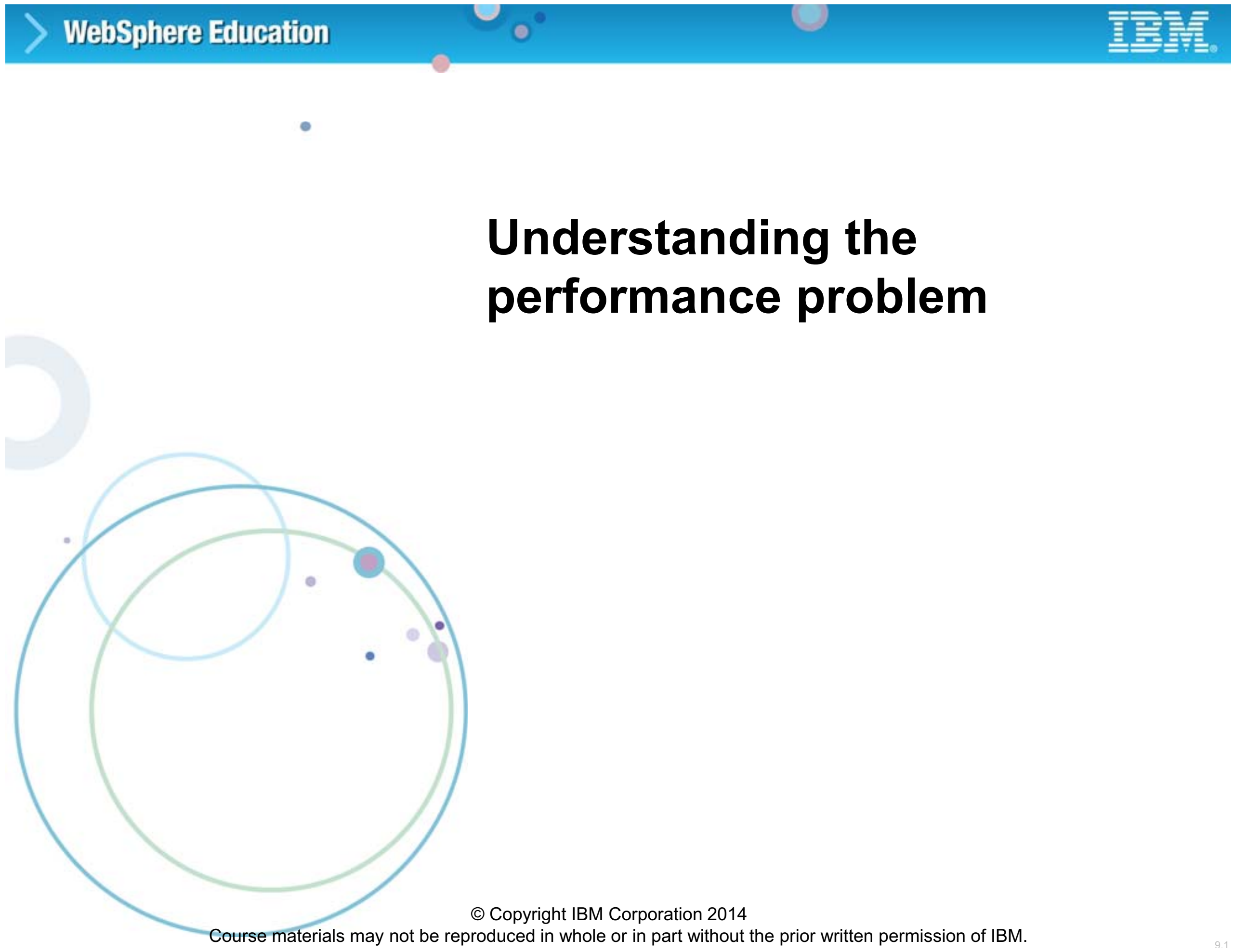


Figure 12-3. Understanding the performance problem WB8691.0

Notes:

WebSphere Education IBM

Understanding the nature of the performance problem

- Performance means different things to different customers
- **Time related**
  - Login
  - Open coach
  - Activity to complete or advance to next activity
  - Open application
  - Save application
- **Throughput related**
  - Concurrent users
  - X tasks or instances per Y time

Figure 12-4. Understanding the nature of the performance problem WB8691.0

Notes:

Get an overall nature of the performance problem

- 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

Figure 12-5. Get an overall nature of the performance problem WB8691.0

Notes:

Before you start troubleshooting a performance problem, you must assess whether it impacts a specific component or the entire environment.

WebSphere Education

Can you 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 worse at certain times of day?
- Is performance bad under load?
- Does performance improve after a restart of the application server?

Figure 12-6. Can you quantify the performance issue? WB8691.0

Notes:

12.2.Gathering data

Gathering data

Figure 12-7. Gathering data WB8691.0

Notes:

WebSphere Education

Gathering information about the environment

- Hardware specifications of IBM BPM servers (# of CPU, RAM)
- Clustered, nonclustered?
- Hardware specifications of the database server
- VM or not? (IBM BPM server and database server)
- Other hardware that is being used

Figure 12-8. Gathering information about the environment WB8691.0

Notes:

WebSphere Education

Gathering information about the software

- IBM BPM version, any interim fixes, fix packs that are installed
- Version of OS
- Database server, what is the RDBMS, what is the version
- VMware or any other VM, what is the version, what type

Figure 12-9. Gathering information about the software WB8691.0

Notes:

WebSphere Education

Gathering resource consumption information

- Performance monitor logs from OS level
- Database server resource consumption logs (AWR report on Oracle, Dashboard report on MSSQL)
- Is database a shared resource or not?

Figure 12-10. Gathering resource consumption information WB8691.0

Notes:

WebSphere Education

Types of data to gather for performance-related problems

Six types of logging can be helpful in isolating performance problems:

- Thread dumps: Gives 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 OOM
- 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)
- Event Manager and Process Monitor: It is always better to look at the process monitor logging
- Solution code logging: Custom logging in the solution code helps to determine where the performance problems lie
- Instrumentation logging: 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 12-11. Types of data to gather for performance-related problems WB8691.0

Notes:

Six types of logging can be helpful in isolating performance 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 OOM.
- Java GC logging: It 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)
- Event manager and process monitor: It is always better to look at the process monitor logging.
- 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: 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, and 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 first?

- 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
- 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
- Thread dumps are the right next step

Figure 12-12. What should you do first? WB8691.0

Notes:

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. It should generally execute in a small amount of time. SystemTask and UCATask are generally services. If these 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 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
  - It is because Java garbage collection is expensive when you are running low on memory
  - It can block the other threads in the JVM and prevent them from moving
- Garbage collection logging can also be useful
  - GC logs give you the ability to graph memory utilization over time

Figure 12-13. What if you are running low on heap? WB8691.0

Notes:

At a high level, when you run low on Java memory (heap space) your CPU suddenly spikes and your application stops responding. It is 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 already ruled that out in the process monitor step above, 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 use over time. This information 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

Other log analysis

- Instrumentation log analysis
- IBM Java cores generation and analysis
- Heap dump analysis
- 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) with the instrumentation logs

Figure 12-14. Other log analysis WB8691.0

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 javacores for a short time; but again, instrumentation logs do not affect performance of the server as traces would. By enabling “wle all” on a server with poor performance, you would likely terminate it; whereas with instrumentation logs, there would be no administrative cost as with verboseGC in WebSphere Application Server.

WebSphere Education IBM.

Unit summary

Having completed this unit, you should be able to:

- Describe the nature of the performance problem
- Describe the types of data that is needed to troubleshoot performance problems

Figure 12-15. Unit summary WB8691.0

Notes:

WebSphere Education

Checkpoint questions

1. True or False. Heap dumps are needed only in case there is an OOM
2. True or False. GC logs give you the ability to graph memory utilization over time

Figure 12-16. Checkpoint questions WB8691.0

Notes:

Write down your answers here:

1.

2.

WebSphere Education

Checkpoint answers

1. True
2. True

Figure 12-17. Checkpoint answers WB8691.0

Notes:

Unit 13. WebSphere Adapter problems

What this unit is about

One of the challenges in troubleshooting IBM Process Server is to isolate adapter problems from enterprise information system (EIS) issues or connection issues. Thus, the first step is to isolate the problem to be sure that the problem was not triggered from the third-party application with which the adapter was communicating. The second step is to gather enough information to differentiate runtime problems from design problems. Technology adapters and application adapters have different sets of configuration properties. This unit covers general adapter troubleshooting techniques.

What you should be able to do

After completing this unit, you should be able to:

- Collect WebSphere Adapter MustGather data
- Enable confidential tracing for added security
- Specify the log and trace file name for a specific adapter
- Explain what to do when the XAResourceNotAvailableException appears in the SystemOut.log file
- Describe possible causes of the OutOfMemoryError and how to resolve the problem
- Explain some of the encoding configuration problems that you might encounter when using the WebSphere Adapter for Flat Files

How you will check your progress

- Checkpoint

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Collect WebSphere Adapter MustGather data
- Enable confidential tracing for added security
- Specify the log and trace file name for a specific adapter
- Explain what to do when the XAResourceNotAvailableException appears in the SystemOut.log file
- Describe possible causes of the OutOfMemoryError and how to resolve the problem
- Explain some of the encoding configuration problems that you might encounter when using the WebSphere Adapter for Flat Files

Figure 13-1. Unit objectives WB8691.0

Notes:

WebSphere Education

Topics

- Troubleshooting WebSphere Adapters
- Troubleshooting tips

Figure 13-2. Topics WB8691.0

Notes:

13.1.Troubleshooting WebSphere Adapters

This topic covers the basics of adapter troubleshooting methods. The MustGather data that is specific to WebSphere Adapter is also discussed. Some of the self-help resources that are associated with WebSphere Adapter for Flat Files are covered.

Troubleshooting WebSphere Adapters

Figure 13-3. Troubleshooting WebSphere Adapters WB8691.0

Notes:

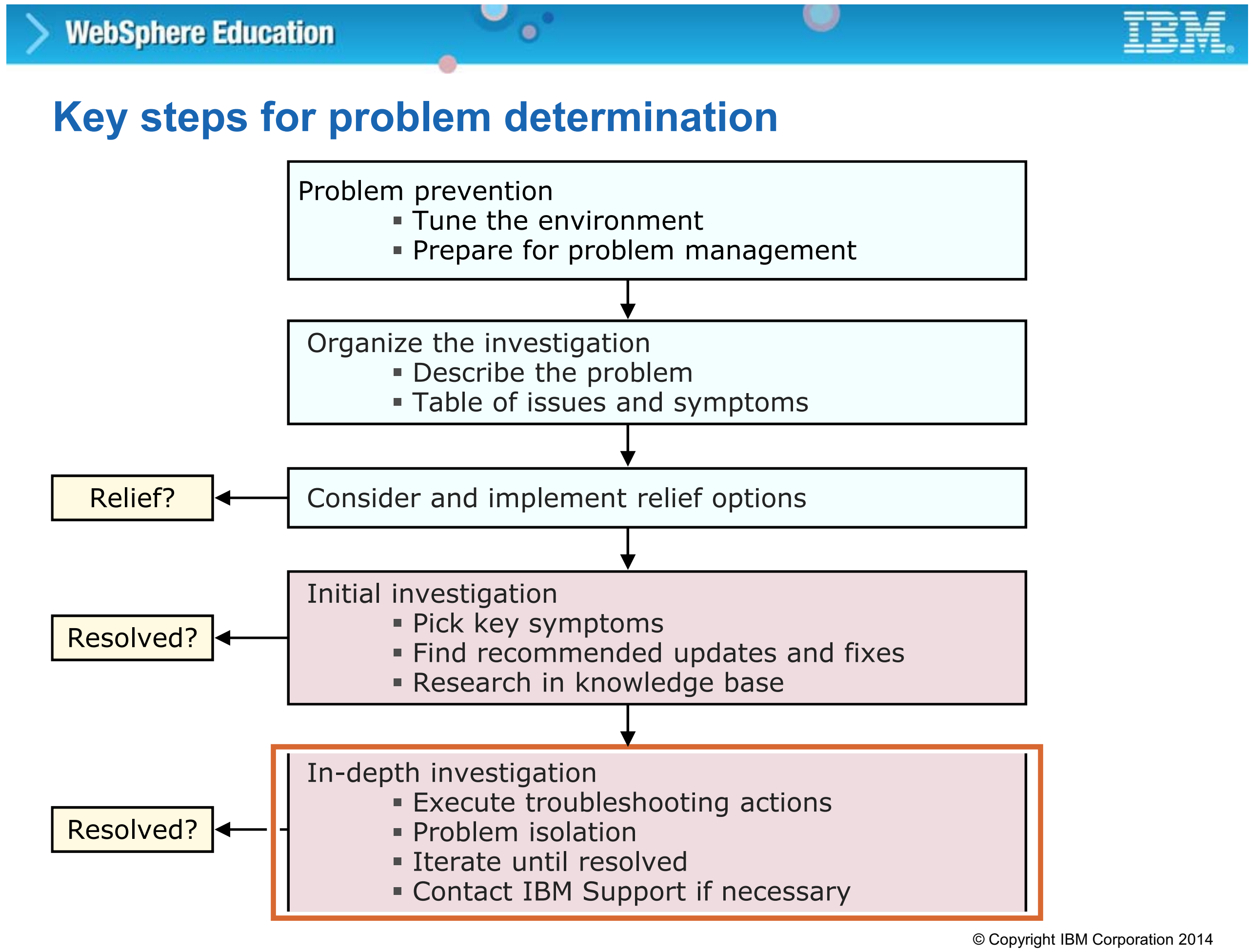


Figure 13-4. Key steps for problem determination WB8691.0

Notes:

WebSphere Education IBM.

WebSphere Adapter support

- WebSphere Process Server supports the following adapters:
  - IBM WebSphere Adapter for PeopleSoft Enterprise
  - IBM WebSphere Adapter for Siebel Business Applications
  - IBM WebSphere Adapter for SAP applications
  - IBM WebSphere Adapter for JD Edwards EnterpriseOne
  - IBM WebSphere Adapter for Oracle E-Business Suite
  - Control System External Call Interface (CICS ECI) resource adapter
  - Information Management System (IMS) Transaction Manager (TM) resource adapter
  - IBM WebSphere Adapter for Flat Files
  - IBM WebSphere Adapter for JDBC
  - IBM WebSphere Adapter for FTP
  - IBM WebSphere Adapter for Email
  - IBM WebSphere Adapter for IBM System i (i5/OS)
  - IBM WebSphere Adapter for Lotus Domino

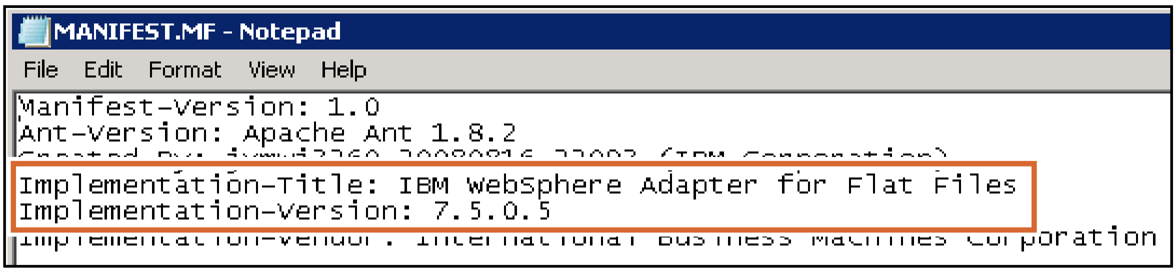
Figure 13-5. WebSphere Adapter support WB8691.0

Notes:

WebSphere Education

MustGather data: WebSphere Adapters

- Adapter version (including the fix pack level)
- To determine the version of the adapter, choose one of these approaches:
  - In IBM Integration Designer, navigate to **File > New > External Service** to open the Enterprise Service Discovery (ESD) wizard that displays the adapter version
  - Extract the **`Adapter.jar`** file within the **`Adapter.rar`** file to find the **`Manifest.mf`** file that contains the adapter version



- IBM Process Server version (including fix pack level)
- IBM Integration Designer version (including fix pack level)

Figure 13-6. MustGather data: WebSphere Adapters WB8691.0

Notes:

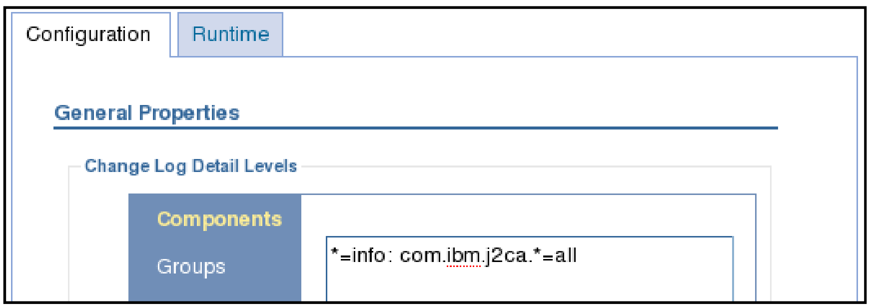
Refer to the MustGather website for WebSphere Adapters:

`http://www.ibm.com/support/docview.wss?rs=695&uid=swg21240322`

WebSphere Education IBM

MustGather data: Adapter development issues

- Collect Enterprise Metadata Discovery (EMD) logs
- For runtime trace:
  1. Enable the adapter to trace from the IBM Process Server administrative console
  2. Navigate to **Troubleshooting > Logs and traces > server > Change Log level details**
  3. Set the tracing either in the **Configuration** tab or the **Runtime** tab to include: `com.ibm.j2ca.*`, or `com.ibm.j2ca.base.*`



  4. Reproduce the problem
  5. Collect the `SystemOut.log`, `trace.log`, and `FFDC` log

Figure 13-7. MustGather data: Adapter development issues WB8691.0

Notes:

To enable the EMD logs, turn on logging in the Enterprise Metadata Discovery (EMD) tool. Selecting the **Change Logging Properties for the Wizard** option enables two fields that allow you to set the file location and logging level. The EMD logs get stored under the `.metadata` folder of your workspace.

For logging and tracing, all WebSphere Adapters start with: `com.ibm.j2ca.*`

The adapter base component starts with: `com.ibm.j2ca.base.*`

To trace specific adapters, set the log detail levels to the following settings:

- WebSphere Adapter for Email (`com.ibm.j2ca.email.*`)
- WebSphere Adapter for FTP (`com.ibm.j2ca.ftp.*`)
- WebSphere Adapter for Flat Files (`com.ibm.j2ca.flatfile.*`)
- WebSphere Adapter for IBM i (`com.ibm.j2ca.i5.*`)
- WebSphere Adapter for JDBC (`com.ibm.j2ca.jdbc.*`)
- WebSphere Adapter for JD Edwards Enterprise (`com.ibm.j2ca.jde.*`)

- WebSphere Adapter for Oracle E-Business Suite:
    - For the core component that is common between the WebSphere Adapter for JDBC and the WebSphere Adapter for Oracle E-Business Suite (`com.ibm.j2ca.dbadapter.core.*`)
    - For the Adapter for Oracle E-Business Suite only (`com.ibm.j2ca.oracleebs.*`)
- WebSphere Adapter for PeopleSoft Enterprise (`com.ibm.j2ca.peoplesoft.*`)
- WebSphere Adapter for SAP Software (`com.ibm.j2ca.sap.*`)
- WebSphere Adapter for Siebel Business Applications (`com.ibm.j2ca.siebel.*`)

WebSphere Education

Enabling confidential tracing (1 of 2)

Enabling confidential tracing

- Displays X values to protect sensitive user data
- Traced as confidential
  - The contents of a business object
  - The contents of the object key of the event record
  - User names
  - The URLs used to connect to the database

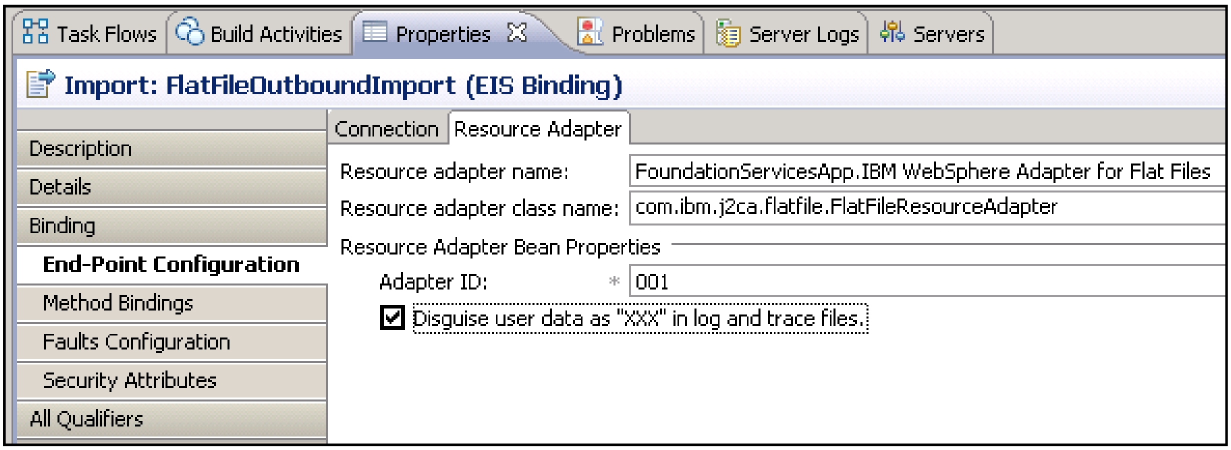


Figure 13-8. Enabling confidential tracing (1 of 2) WB8691.0

Notes:

WebSphere Education

Enabling confidential tracing (2 of 2)

Enterprise Applications > FoundationServicesApp > Manage Modules > CWYFF_FlatFile.rar > FoundationServicesApp.IBM WebSphere Adapter for Flat Files > J2C connection factories > FoundationServices.jca.FlatFileOutboundImport_CF > **Custom properties**

Use this page to specify custom properties that your enterprise information system (EIS) requires for the resource providers and resource factories that you configure. For example, most database vendors require additional custom properties for data sources that access the database.

Preferences

Name	Value	Description	Required
You can administer the following resources:			
adapterID	001	adapterID	false
biDiContextEIS		biDiContextEIS	false
connectionRetryInterval	60000	connectionRetryInterval	false
connectionRetryLimit	0	connectionRetryLimit	false
detectXMLCharacter		detectXMLCharacter	false
generateUniqueFile	false	generateUniqueFile	false
hideConfidentialTrace	true	hideConfidentialTrace	false
logFileSize	0	logFileSize	false

Figure 13-9. Enabling confidential tracing (2 of 2) WB8691.0

Notes:

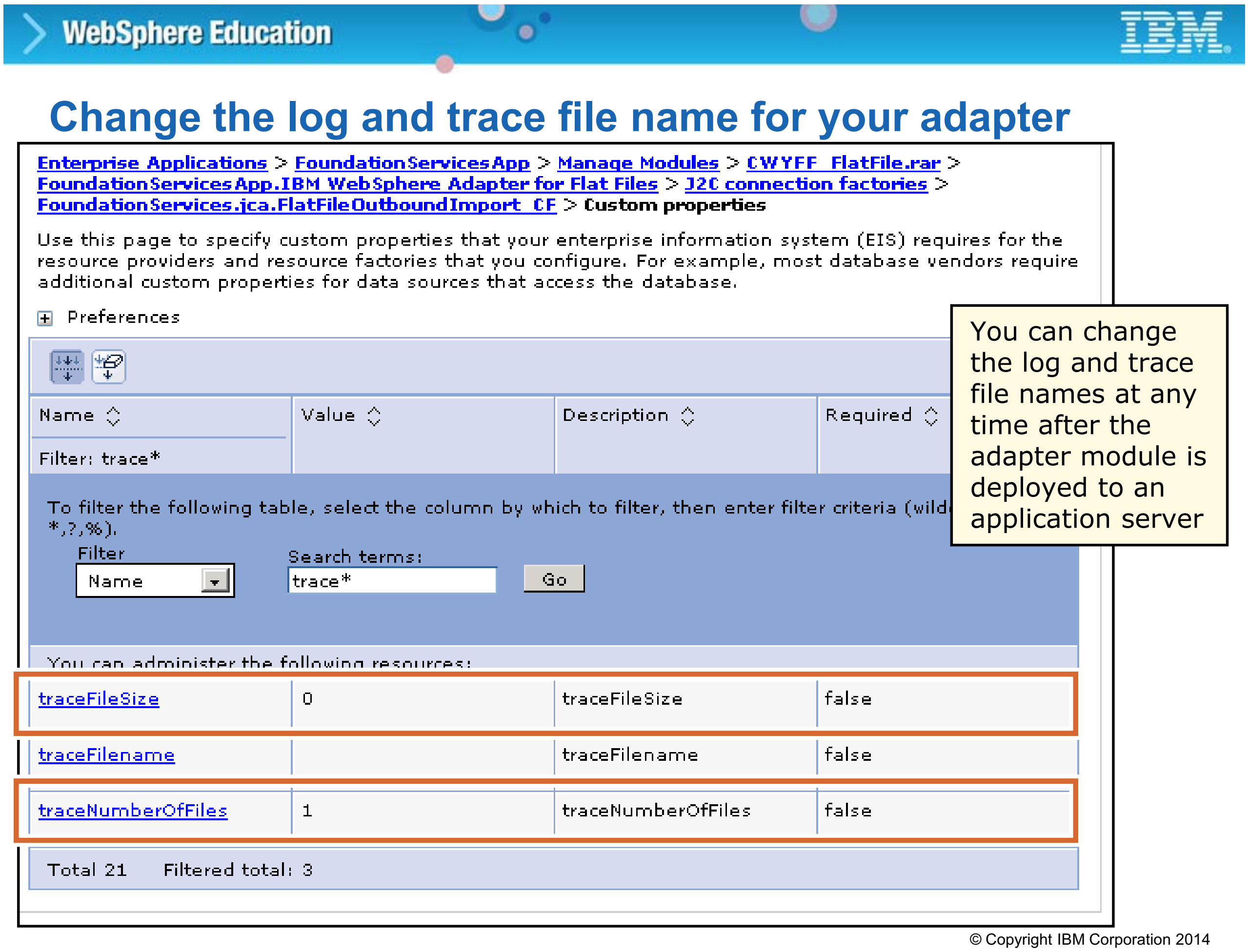


Figure 13-10. Change the log and trace file name for your adapter WB8691.0

Notes:

1. In the navigation pane of the administrative console, select **Applications > Enterprise** Applications.
2. In the **Enterprise Applications** list, click the name of the adapter application. It is the name of the EAR file for the adapter, but without the `.ear` file extension. For example, if the EAR file is named `Accounting_OutboundApp.ear`, then click **Accounting_OutboundApp**.
3. In the **Configuration** tab under the **Modules** list, click **Manage Modules**.
4. In the list of modules, click **IBM WebSphere Adapter for XXX**.
5. In the **Configuration** tab under **Additional Properties**, click **Resource Adapter**.
6. In the **Configuration** tab under **Additional Properties**, click **Custom properties**.
7. In the Custom Properties table, change the file names.
   a. Click either **logFilename** to change the name of the log file or **traceFilename** to change the name of the trace file.
   b. In the **Configuration** tab, type the new name in the **Value** field. By default, the log file is called `SystemOut.log` and the trace file is called `trace.log`.

c. Click **Apply** or **OK** to save the changes on your local machine.

d. To save your changes to the master configuration on the server, use one of the following procedures:

- **Static change**: Stop and restart the server. This method allows you to make changes, but the changes do not take effect until you stop and start the server.
- **Dynamic change**: Click the **Save** link in the Messages box above the Custom properties table. Click **Save** again when prompted. This method allows you to change, and it takes effect right away.

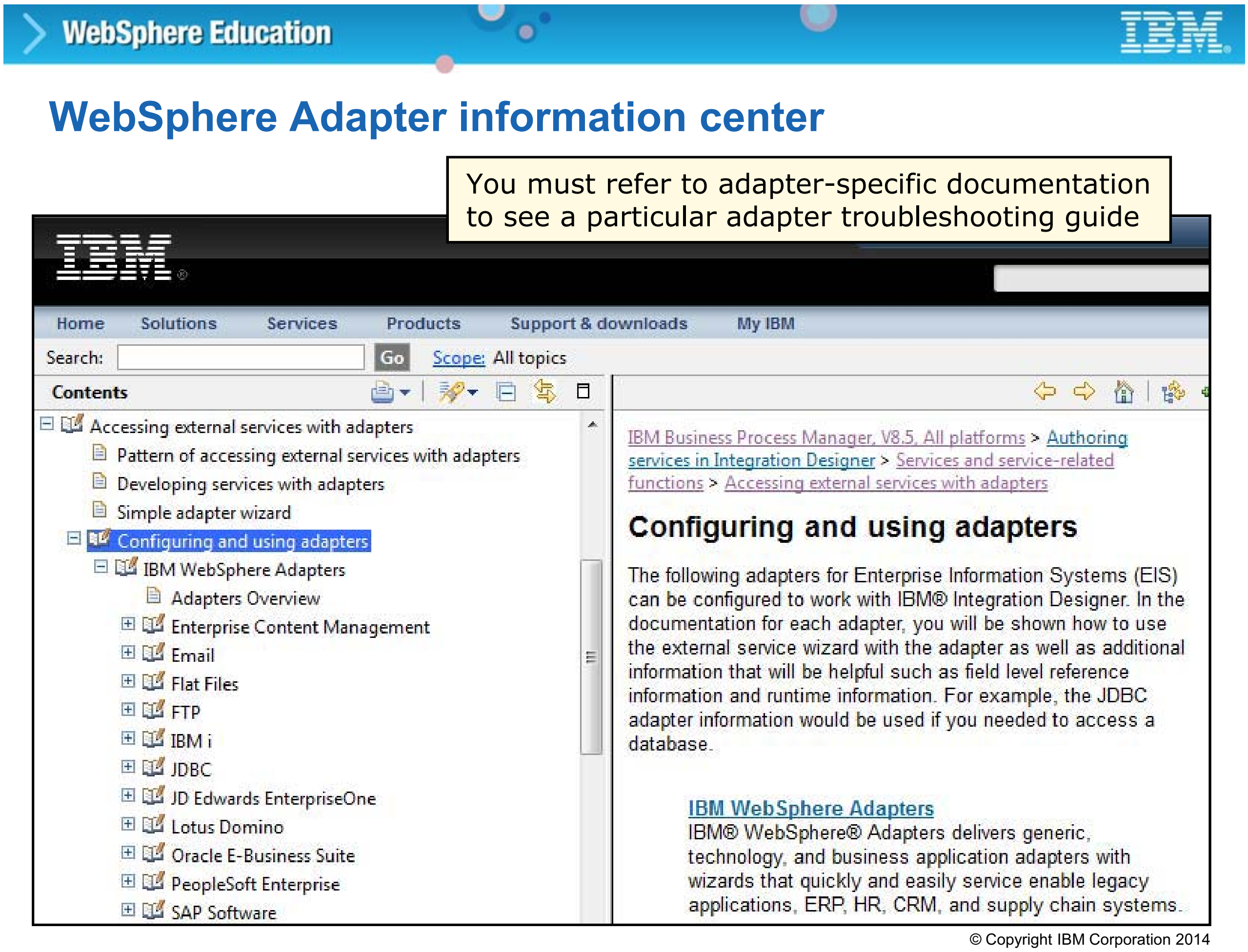


Figure 13-11. WebSphere Adapter information center WB8691.0

Notes:

The web address for the information center page that is shown in the slide is:

`http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r5m0/topic/com.ibm.wbpm.wid.integ.doc/topics/tcreatecmps.html`

WebSphere Education

Troubleshooting support (1 of 2)

- Server Logs view
  - Server Logs in IBM Integration Designer provide detailed information with optional data snapshots
  - Most preferred method for troubleshooting
- Support for Log Analyzer
  - The adapter creates log and trace files that can be viewed with the Log Analyzer
  - The Log Analyzer can filter log and trace files to isolate the messages and trace information for the adapter
  - Highlights the adapter's messages and traces information in the log viewer
- Business faults support
  - The adapter supports business faults, which are exceptions that are anticipated and declared in the outbound service description, or import
  - A business rule violation or a constraint violation can cause business faults to occur at predictable points in a business process

Figure 13-12. Troubleshooting support (1 of 2) WB8691.0

Notes:

Business faults are specific to a WebSphere Adapter. For example, the WebSphere Adapter for Flat Files enables faults such as **DuplicateRecordFault**, **RecordNotFoundFault**, and **MissingDataFault**. You must refer to the information center for details on each WebSphere Adapter in which you are interested.

WebSphere Education

Troubleshooting support (2 of 2)

- First-failure data capture (FFDC) support
  - The adapter supports FFDC, which provides persistent records of failures and significant software incidents that occur during run time
  - The FFDC feature runs in the background and collects events and errors that occur at run time
  - FFDC provides a means for associating failures to one another, allowing software to link the effects of a failure to their causes, and facilitate the quick location of the root cause of a failure

Figure 13-13. Troubleshooting support (2 of 2) WB8691.0

Notes:

WebSphere Education

Self-help resources: IBM Support site

- The WebSphere Adapters software support website provides links to many resources to help you learn about, use, and troubleshoot WebSphere Adapters:
  `http://www.ibm.com/support/entry/portal/product/websphere/websphere_adapters_family`
  - Flashes (alerts about the product)
  - Technical information includes product information center, manuals, IBM Redbooks, and white papers
  - Educational offerings
  - Technotes: Technotes provide the most current documentation about WebSphere Adapters
  - Recommended fixes: List of recommended fixes is available at:
    `http://www.ibm.com/support/docview.wss?uid=swg27010397`

Figure 13-14. Self-help resources: IBM Support site WB8691.0

Notes:

WebSphere Education

Self-help resources: IBM Support Assistant

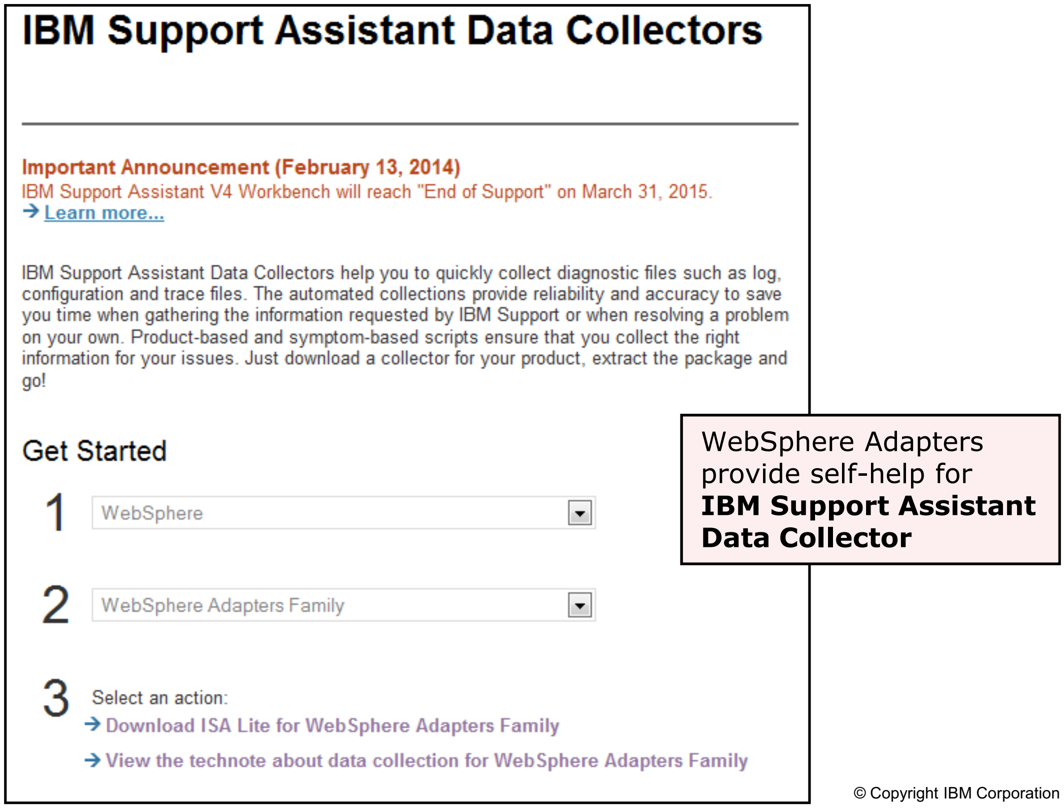


Figure 13-15. Self-help resources: IBM Support Assistant WB8691.0

Notes:

See `http://www.ibm.com/software/support/isa/dc.html` for further reference.

WebSphere Education

How to isolate adapter issues from IBM Process Server

The first step for troubleshooting adapter problems is to ask the right questions to isolate the adapter from its edge components

- Check the connectivity between the adapter and its connecting application
  - For example, can a JDBC adapter connect to DB2?
  - If not, verify that the adapter configuration is correct
- Check that the connected application is running
- Verify that the authentication alias is set properly

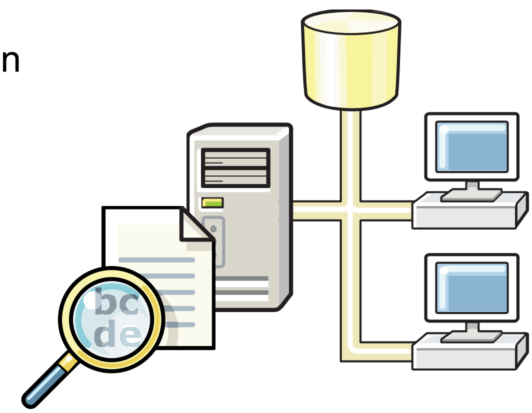


Figure 13-16. How to isolate adapter issues from IBM Process Server WB8691.0

Notes:

Adapter-related messages get logged in `SystemOut.log` and `SystemErr.log`, as part of the IBM Process Server, unless the logs are configured to be in separate files.

Inside the log file, search for messages that start with a prefix related to each of the adapters.

- For JDBC adapter problems, look for messages with the `CWYBC` prefix
- For the Flat File adapter, look for `CWYFF`
- For the email adapter, look for `CWYEM`
- For the FTP adapter, look for `CWYFT`
- For the IBM i adapter, look for `CWYIS`
- For the Siebel adapter, look for `CWYEB`
- For the SAP Software adapter, look for `CWYAP`
- For the PeopleSoft adapter, look for `CWYES`
- For the common component shared by the JDBC adapter and the Oracle E-Business Suite, look for `CWYDB`

- The adapter foundation classes, which all adapters use, issue messages with the prefix `CWYBS`

13.2.Troubleshooting tips

This topic covers the basics of adapter troubleshooting methods. The MustGather data that is specific to the WebSphere Adapter is also discussed. Some of the self-help resources that are associated with WebSphere Adapter for Flat Files are covered.

WebSphere Education

IBM

Troubleshooting tips

Figure 13-17. Troubleshooting tips WB8691.0

Notes:

WebSphere Education

Invalid directory error with WebSphere Adapter for FTP

- **Symptom:** The specified input and output directories throw an invalid directory error because the case of the specified directory does not match the directory case on the FTP server, even if the FTP server is not case-sensitive

```
com.ibm.websphere.sca.ServiceRuntimeException: ResourceException
thrown in J2CMethodBindingImpl.invoke()
javax.resource.spi.InvalidPropertyException: The DirectoryPath
does not have the necessary permissions for out-bound operation
```

- **Reason**: The case-sensitivity of the directory names depends on the specific FTP server that is used, and sometimes on the platform on which the server is running
  - Some FTP servers support case insensitivity, whereas many others do not support it
- **Solution**: Make sure that the case of the directory that is specified matches the case of what is displayed when you log in to a console-based FTP client
  - You can check the case of directory name using the `ls -l` command

Figure 13-18. Invalid directory error with WebSphere Adapter for FTP WB8691.0

Notes:

WebSphere Education

SAXParseException with WebSphere Adapter for Flat Files

- **Symptom:** When the adapter is configured with the XML data handler, the following exception is thrown:
  `org.xml.sax.SAXParseException: Content is not allowed in trailing section`
- **Reason**: The content of the file is not in the specified business object format
- **Possible solution**: Perform the following tasks:
  1. Make sure that the file content matches the business object structure
  2. If the content file contains multiple business objects, make sure that the delimiter is specified correctly

Figure 13-19. SAXParseException with WebSphere Adapter for Flat Files WB8691.0

Notes:

When the adapter is configured with the XML data handler, an `org.xml.sax.SAXParseException` exception is generated if the content is not in the specified business object format. To correct the problem, make sure that the file content matches the business object structure. If the file contains multiple business objects, make sure that the delimiter is specified correctly.

WebSphere Education

OutOfMemoryError with WebSphere Adapter for Flat Files

- **Symptom**: Observe `OutOfMemoryError` in the `trace.log`
- **Reason**: Lack of resources in the system is most likely the issue
- **Possible solution**: Perform the following tasks:
  - Configure the **poll period** and **poll quantity** of the inbound adapter parameters
  - **Example**: If you have a throughput of 20 events per second, and events are continuously arriving to the adapter for processing, set the poll period to a small value and poll quantity to 20

> A good rule of thumb is to use a lower poll quantity and longer poll period when you are dealing with a large object

  - Java heap size should be three times that of the object
  - However, if the heap size is too large, it could result in paging issues, thus reducing performance

Figure 13-20. OutOfMemoryError with WebSphere Adapter for Flat Files WB8691.0

Notes:

See the following for reference:

- Adapter support page: `http://www.ibm.com/support/entry/portal/product/websphere/websphere_adapters_family`
- Open service request online: `http://www.ibm.com/support/servicerequest`

WebSphere Education

Debugging tips for WebSphere Adapter for Flat Files (1 of 3)

- **Symptom**: File content encoding issues are in the form of exceptions and can be observed in the trace log

```
Caused by: commonj.connector.runtime.DataHandlerException:
  java.io.IOException: Invalid byte 2 of 3-byte UTF-8 sequence.
  at
  com.ibm.wbiserver.datahandler.xml.XMLDataHandler.transformInputStreamToD
  ataObject(XMLDataHandler.java:372)
  at
  com.ibm.wbiserver.datahandler.xml.XMLDataHandler.transform(XMLDataHandle
  r.java:81)
  at
  com.ibm.j2ca.extension.emd.runtime.ChildDataHandlerDataBindingInvoker.tr
  ansformToDataObject(ChildDataHandlerDataBindingInvoker.java:101)
  at
  com.ibm.j2ca.extension.emd.runtime.EmbeddedNameFunctionSelector.generate
  EISFunctionName(EmbeddedNameFunctionSelector.java:98)
  ... 22 more
  Caused by: java.io.IOException: Invalid byte 2 of 3-byte UTF-8 sequence.
  at ...
```


Figure 13-21. Debugging tips for WebSphere Adapter for Flat Files (1 of 3) WB8691.0

Notes:

WebSphere Education

Debugging tips for WebSphere Adapter for Flat Files (2 of 3)

- **Symptom**: File content encoding issues are in the form of exceptions and can be observed in the trace log (continued)

```
Caused by: java.lang.RuntimeException: An invalid XML character
 (Unicode: 0x3) was
 found in the element content:PK
 at
 org.eclipse.emf.ecore.xmi.impl.XMLSaveImpl$Escape.convertText(XMLSaveIm
 pl.java:3101)
 at
 org.eclipse.emf.ecore.xmi.impl.XMLSaveImpl.getDatatypeValue(XMLSaveImpl
 .java:2758)
 at
 com.ibm.ws.bo.bomodel.util.BOXMLSaveImpl.getTrueType(BOXMLSaveImpl.java
 :550)
```

- **Reason**: It is most likely caused by using the wrong file content encoding configuration settings while using WebSphere Adapter for Flat Files

Figure 13-22. Debugging tips for WebSphere Adapter for Flat Files (2 of 3) WB8691.0

Notes:

WebSphere Education

Debugging tips for WebSphere Adapter for Flat Files (3 of 3)

Solution: Change the file content encoding configuration settings

- The `Invalid byte 2 of 3-byte UTF-8 sequence` is thrown when the event file contains special characters and the event file is not created with the UTF-8 encoding
- Perform one of the following actions: (1) the event file has special characters that should be created by using the UTF-8 encoding, or (2) use the file content encoding as `iso-8859-1`
- The `An invalid XML character (Unicode: 0x3)` is thrown when the event file that is processed is of binary form and the file content encoding is configured as **UTF-8**
  - To resolve this issue, configure the file content encoding as **BINARY** instead of **UTF-8** to process the binary files

Figure 13-23. Debugging tips for WebSphere Adapter for Flat Files (3 of 3) WB8691.0

Notes:

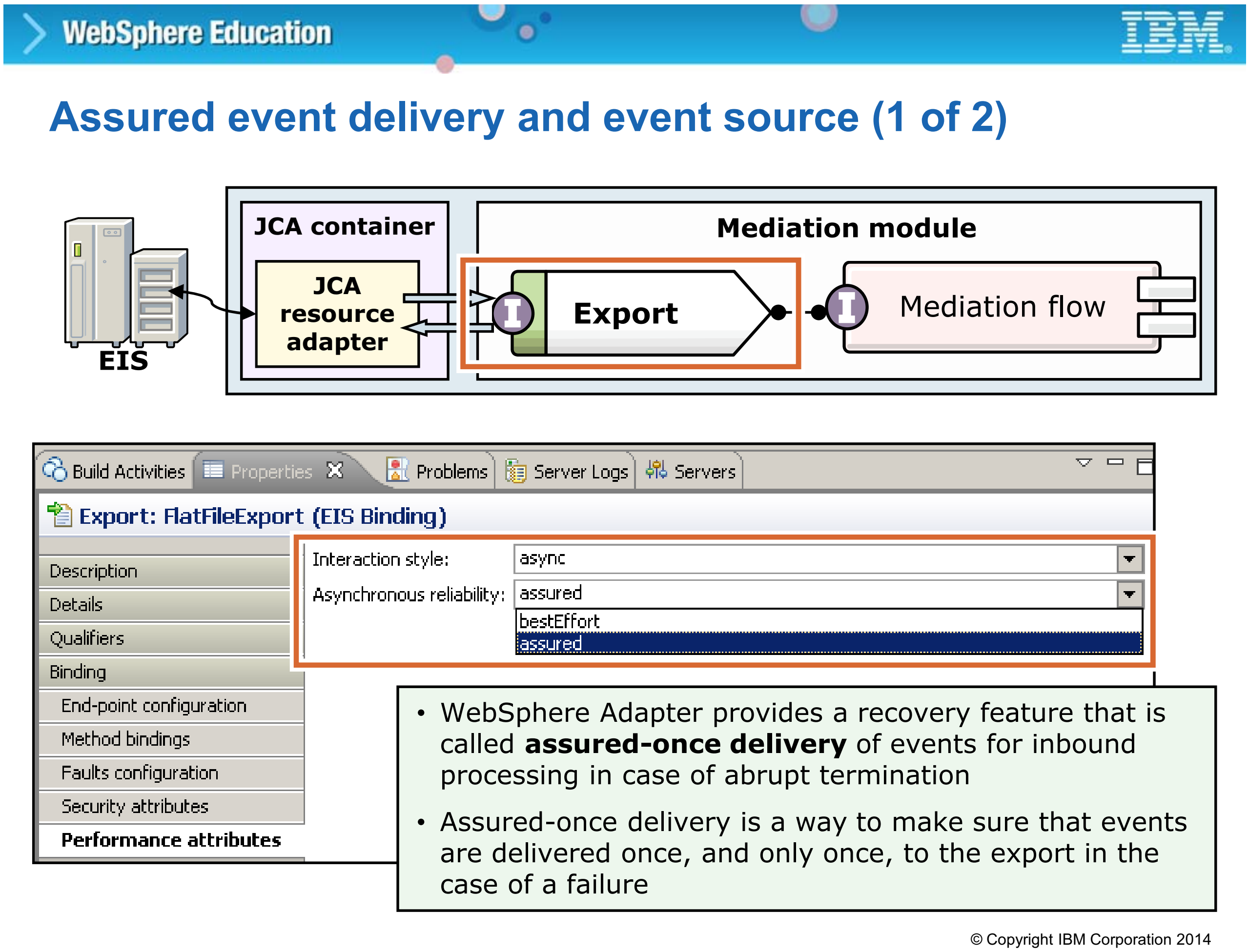


Figure 13-24. Assured event delivery and event source (1 of 2) WB8691.0

Notes:

WebSphere Education IBM

Assured event delivery and event source (2 of 2)

- If you set the **`AssuredOnceDelivery`** activation specification property to **true**, the event store must be created and configured
- The adapter stores an XID (transaction ID) value for each event in the event store when an event is obtained for processing:
  - The XID value for the event is updated in the event store
  - The event is delivered to its corresponding export
  - The event is deleted from the event store

Column name	Type	Description
EVNTID	varchar(255)	Unique identifier for each event
EVNTSTAT	Integer	Status of the event – NEW(0) – PROCESSED(1) – FAILED(-1)
XID	varchar(255)	Transaction ID used by the adapter
EVNTDATA	varchar(255)	Used to track failed events so that they are not processed again during recoveries

Figure 13-25. Assured event delivery and event source (2 of 2) WB8691.0

Notes:

WebSphere Education IBM

Unit summary

Having completed this unit, you should be able to:

- Collect WebSphere Adapter MustGather data
- Enable confidential tracing for added security
- Specify the log and trace file name for a specific adapter
- Explain what to do when the XAResourceNotAvailableException appears in the SystemOut.log file
- Describe possible causes of the OutOfMemoryError and how to resolve the problem
- Explain some of the encoding configuration problems that you might encounter when using the WebSphere Adapter for Flat Files

Figure 13-26. Unit summary WB8691.0

Notes:

WebSphere Education

Checkpoint questions

1. True or false: WebSphere Adapters support only system or runtime exceptions.

2. Which of the following tools does it filter the log and trace files to isolate the messages and trace information for the adapter?
   a) Problem view
   b) Server Log view
   c) IBM Support Assistant
   d) Log Analyzer

3. True or false: When you see the `OutOfMemoryError` in your log, it implies that the system lacks the resource to handle the incoming events. You need to configure the **poll frequency** and **poll quantity** adapter properties to appropriate values. Also, tune the Java heap size.

Figure 13-27. Checkpoint questions WB8691.0

Notes:

Write down your answers here:

1.

2.

3.

WebSphere Education

Checkpoint answers

1. False. WebSphere Adapters support business faults as well as runtime faults, which can be captured in FFDC.

2. (d) Log Analyzer.

3. True.

Figure 13-28. Checkpoint answers WB8691.0

Notes:

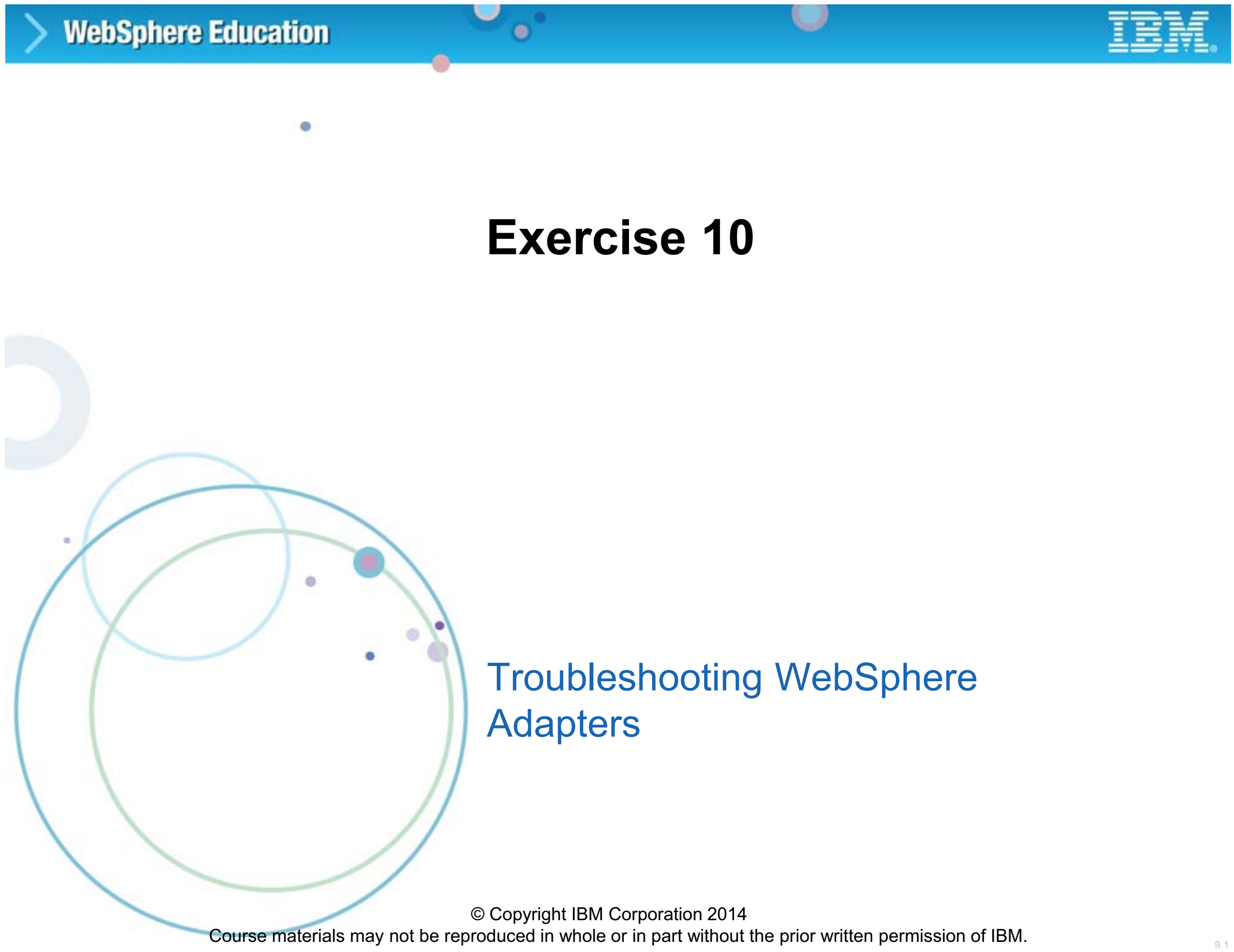


Figure 13-29. Exercise 10 WB8691.0

Notes:

WebSphere Education IBM

Exercise objectives

After completing this exercise, you should be able to:

- Use the IBM Integration Designer test environment to conduct component tests
- Restrict the source of a problem with cross-component trace
- Isolate an adapter problem and test the adapter components
- Examine log messages to pinpoint a problem

Figure 13-30. Exercise objectives WB8691.0

Notes:

Unit 14. Problem prevention and best practices

What this unit is about

In many cases, problem determination takes place after a problem is discovered. However, prudent system administrators or troubleshooters start planning long before a problem occurs. In other words, they prepare the environment so that troubleshooting can be done more quickly and effectively if and when problems occur.

What you should be able to do

After completing this unit, you should be able to:

- Describe best practices for preventing production problems
- Explain how to prepare for problems before they occur
- List some of the diagnostic tools that can help troubleshoot problems

How you will check your progress

- Checkpoint

WebSphere Education

Unit objectives

After completing this unit, you should be able to:
- Describe best practices for preventing production problems
- Explain how to prepare for problems before they occur
- List some of the diagnostic tools that can help troubleshoot problems

Figure 14-1. Unit objectives WB8691.0

Notes:

WebSphere Education

Topics

- Problem prevention strategies
- Problem diagnostics

Figure 14-2. Topics WB8691.0

Notes:

14.1.Problem-prevention strategies

Problem determination does not start with a runtime exception. Problem determination should start well before a problem occurs. If a problem occurs in a production environment, you need to resolve the issue as quickly as possible. You also need to provide a temporal solution for the events that are in progress. Therefore, this topic provides suggested problem-prevention strategies.

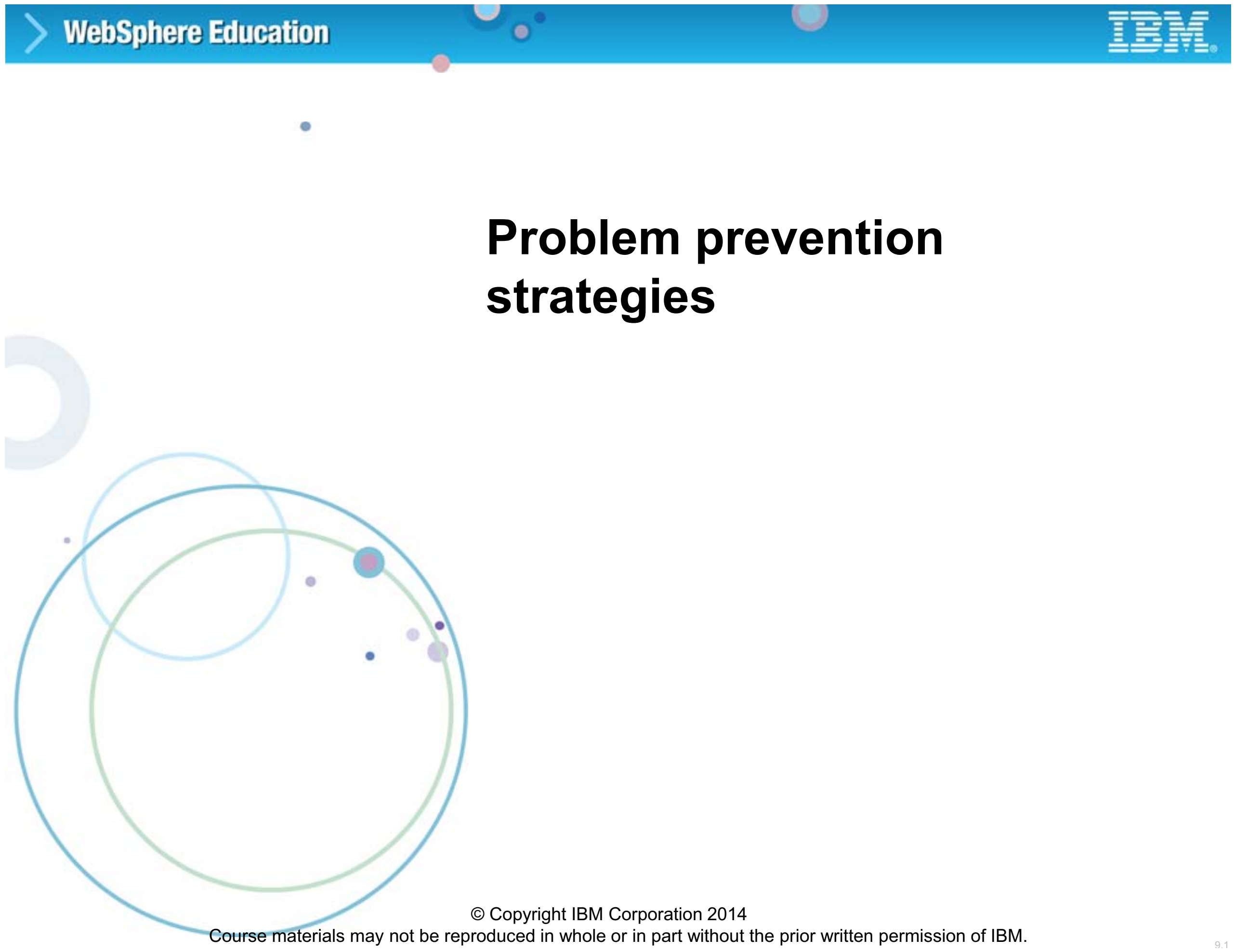


Figure 14-3. Problem prevention strategies WB8691.0

Notes:

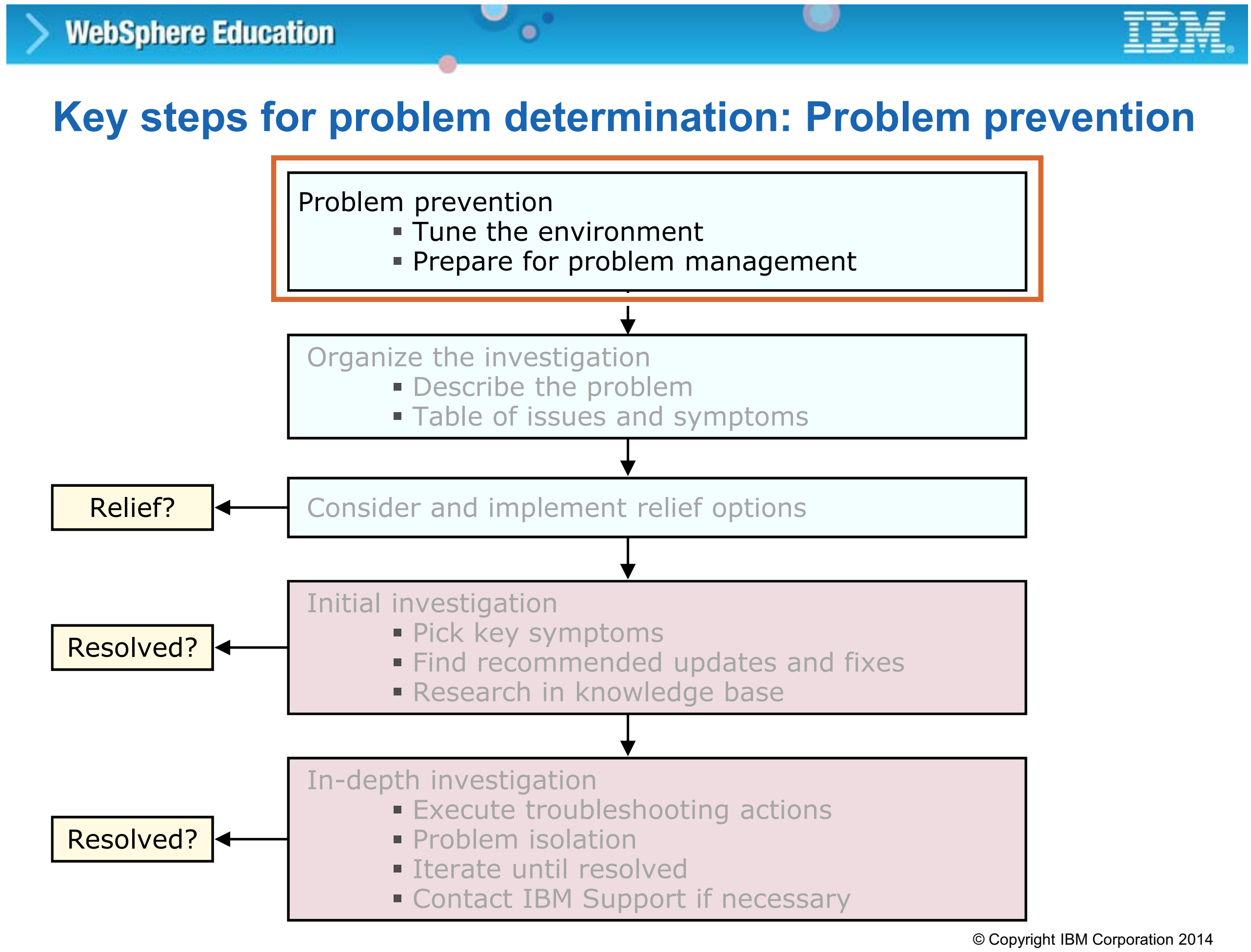


Figure 14-4. Key steps for problem determination: Problem prevention WB8691.0

Notes:

This topic covers problem prevention strategies and tools.

WebSphere Education

Set up monitoring and detect problems

- Passive monitoring versus active monitoring
  - **Passive**: Examine logs, PMI statistics, heap size, verbose GC data
  - **Active**: Send a test transaction from end to end
- Monitor at all levels: Network, operating system, server instance, application, dependent systems
  - **Example:** Database, directory
- Verify that the system is functioning correctly by periodically testing the operation of the system
- Overall **health** or **performance** of the system might degrade slowly over a long period before it finally leads to a serious problem
- The sooner you detect that something is wrong, the more time and opportunity you have to collect diagnostic information for troubleshooting
- Be prepared to actively generate more diagnostic data when a problem occurs

Figure 14-5. Set up monitoring and detect problems WB8691.0

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. This course covers only a few points on this topic.

Passive monitoring can be done at all levels: network, operating system, application server, 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: you periodically test the operation of the entire system from end to end. One technique that you can use is to ping system components, such as one server or one database connection. Another technique is end-to-end pinging: you periodically send an entire “dummy” transaction through the system and verify 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.

Make sure that you:

- Monitor the system at all levels, from the network to applications and dependent systems
- Monitor the main system log files for errors and events
    - **Example**: Detecting application server restarts
- Use monitors and alerts on key system metrics
    - Memory usage
    - Default PMI statistics
    - Performance advisors

Here are some examples of ongoing system “health” monitoring:

- Significant errors in the logs that the various components emit.
- Metrics that each component produces should remain within acceptable norms.
    - For example, operating system processor and memory statistics, IBM Business Process Manager performance metrics, and transaction rates through the application
- Spontaneous appearance of special artifacts that get generated only when a problem occurs, such as Java dumps or heap dumps.
- Periodically send a “ping” through various system components or the application, and verify that it continues to respond as expected.

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 diagnostics:

- Actively trigger various system dumps, if they are not generated automatically (such as Java dump, heap dump, system dump, or other 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, or processor usage.
- Enable and collect information from the IBM Business Process Manager 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.
- Actively test or “ping” various aspects of the system to see how their behavior changes compared to normal conditions. This activity is done to try to isolate the source of the problem in a multicomponent system.

WebSphere Education IBM

Create and maintain a system architecture diagram

- Maintain an architecture diagram that shows all major components of the overall system
  - Machines and software components that operate on these machines
  - How they communicate
  - Main flows for requests that are processed through the system
- Be specific and detailed
  - Indicate software versions, server names, and IP addresses, if possible
- A good up-to-date architecture diagram can help accelerate the troubleshooting process
  - Identify the various points in the system where to find information or clues about the cause of a problem
  - Clearly communicate between the various parties that are involved in the troubleshooting task
  - Answer and verify the ***"what changed recently?"*** question

Figure 14-6. Create and maintain a system architecture diagram WB8691.0

Notes:

The system architecture or topology diagram shows all system components and the main flows between them.

The system architecture diagram can help move the progress of the troubleshooting process by providing information for the following activities:

- 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

An example diagram is on the next page.

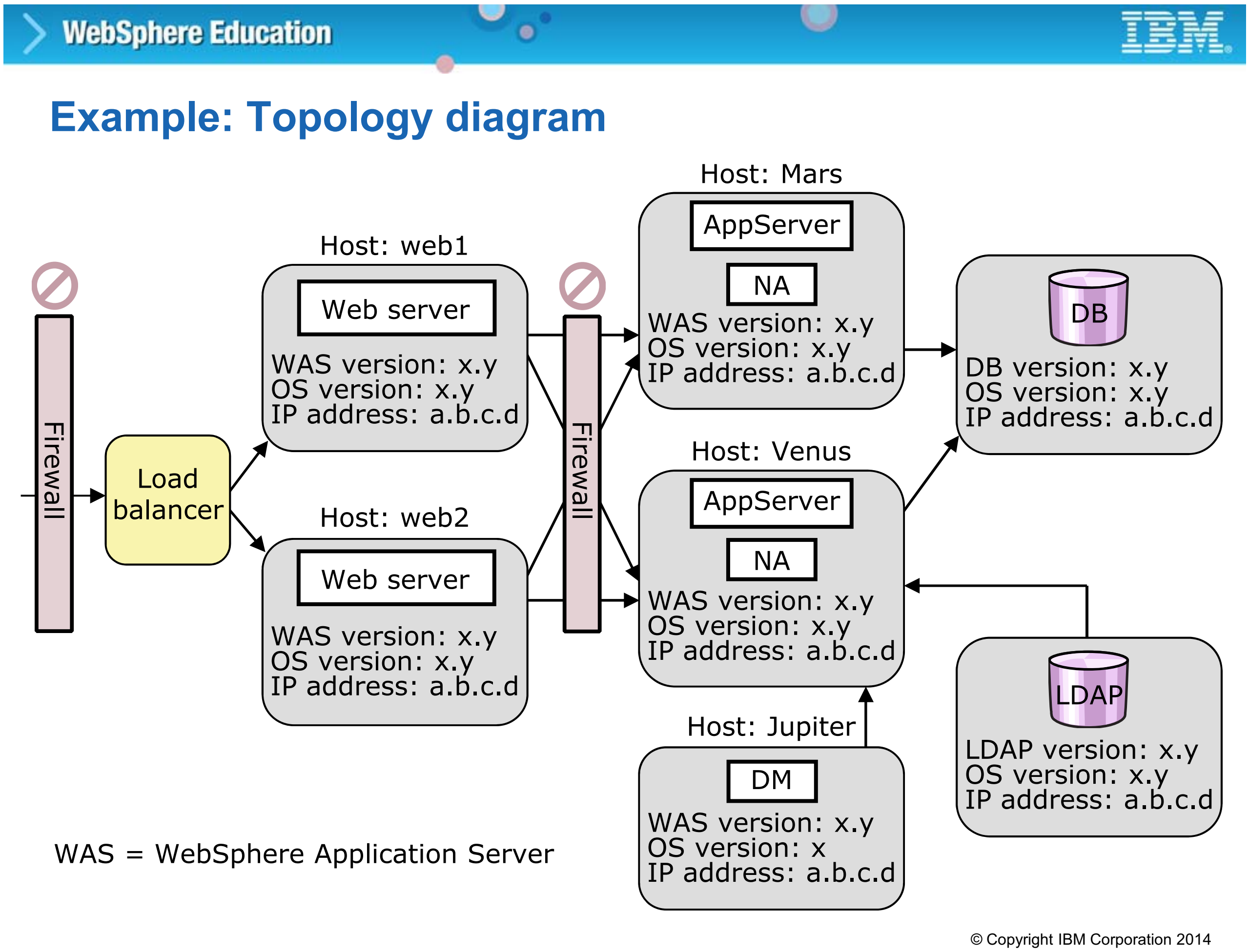


Figure 14-7. Example: Topology diagram WB8691.0

Notes:

The following are some key elements to include in your topology diagram:

- Clearly show all the components in the system and their dependencies, not just the elements that are part of the main flow of processing.
  - For example, administration services and security
- Be specific:
  - Machine names and IP addresses.
  - Software versions of all software that is installed on each machine.
  - Do not forget the network topology and relationships (a separate network topology chart might be useful).

This information helps form the basis for your diagnostic data collection plan.

WebSphere Education

Establish baselines

- To help answer the question, *"What is different now compared to yesterday when the problem was not occurring?"*, actively collect and maintain a baseline

Some questions to answer when creating system 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?

Figure 14-8. Establish baselines WB8691.0

Notes:

Problem determination involves much observation: you observe various aspects of the behavior of the system. Then, you try to determine whether these behaviors are normal or they represent a symptom of a possible problem.

When you establish a system baseline, gather extensive information about the state of the system at a time when the system is operating normally.

Here are some examples of information that you might collect:

- 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 after a problem occurs
- Information about normal transaction rates in the system, response times, and other time-dependent information
- Various operating system 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

In many actual systems, it is not unusual to see various benign “errors” during normal operation. Learn to recognize these benign errors, or better yet, eliminate as many of them from the implementation of the system as possible.

WebSphere Education IBM

Define a diagnostic collection plan

- In advance, decide which diagnostic elements should be captured by default on the first occurrence of any problem
- Your diagnostic collection plan should cover:
  - Collection of the diagnostic artifacts that are always present or automatically generated when a problem occurs
  - Set of specific actions that can be taken to generate more diagnostics
- Make arrangements so that all predetermined diagnostic elements get reliably captured
  - Clearly identify where the data is
  - Establish procedures to collect data and practice these procedures
  - Synchronize the clocks between all servers to facilitate analysis
  - Manage the logs and all other diagnostics artifacts to avoid surprises during normal operation
  - Make sure that you have enough disk space available for diagnostic artifacts
  - Automate as much as possible by providing command scripts that can be invoked to perform a complex set of actions

Figure 14-9. Define a diagnostic collection plan WB8691.0

Notes:

When a problem occurs, there is often confusion along with great pressure to restore the system to normal operation quickly, which can cause mistakes that lead to unnecessary delays. The following steps are critical: make sure that you have an action plan, and make sure that everyone is aware of this action plan.

As with monitoring, gathering data is a trade-off: you want to capture as much as possible, but you do not want to adversely affect the normal operation of the system.

You also want to make sure that you can reliably capture the diagnostic data that you need:

- Can you find where the data is stored?
- Do you have procedures established for collecting the data?
- Do you have enough hard disk space to store the data?

The simplest diagnostic collection plan is in the form of plain, written documentation that lists all the detailed manual steps that must be taken.

To be more effective, try to automate as much of this plan as possible. Provide one or more command scripts that can be invoked to do a complex set of actions, or use more sophisticated system management tools. The various collector tools and scripts now offered as part of IBM Support Assistant can provide a good framework for you to start automating many diagnostic collection tasks.

Also, do not forget the human element:

- Who is in charge, when there is a production problem, of implementing the diagnostic data collection plan and taking recovery actions?
- What groups must be informed or coordinated with?

Here are some tools to consider when you plan for and implement diagnostic data collection.

- Check the default WebSphere logs: The default WebSphere server logs include SystemOut, SystemErr, native_stdout, and native_stderr.
- WebSphere first-failure data capture (FFDC) facility: FFDC provides the instrumentation for exception handlers (catch blocks) to record exceptions that a component produces. FFDC data can be found in the `SystemOut.log` file and in the **FFDC** folder in the server profile `logs` directory.
- HTTP access logs: Increased logging of requests at the HTTP server to show not just a single log entry for each request, but a separate log entry for the start and end of each request.
- JVM verboseGC log: This log is often useful, and usually uses relatively low processing power on a well-tuned system.
- Core memory dumps, JVM Java dumps, heap dumps, and system dumps: Java dumps are typically cheap to produce, and can be enabled for automatic generation. Heap dumps and system dumps can involve significant processor usage, so consider carefully before setting them up to be triggered automatically.
- Enable verbose garbage collection: Verbose garbage collection has a minimal impact on performance; the benefit usually outweighs the cost.
- Minimal monitoring of network health: For example, a moderate level of monitoring performance counters that the WebSphere Application Server Performance Monitoring Infrastructure (PMI) provides.
- Minimal monitoring of operating system resources: For example, minimal WebSphere Application Server tracing to capture one or a few entries only for each transaction (web requests or EJB requests).
- Attempt pings and application checks.
- Application-level tracing and logging, if any.
- Collect specific PMI statistics. When PMI is enabled, the monitoring of individual components can be enabled or disabled dynamically.
- Use IBM Java Health Center to monitor an active JVM: IBM Java Health Center does not use much processing power, and it runs alongside an IBM Java application with a small effect on application performance. It can be downloaded from the following website:
`https://www.ibm.com/developerworks/java/jdk/tools/healthcenter/`

WebSphere Education

Create maintenance plans

- Applying regular maintenance (interim fixes, fix packs) reduces the probability and impact of problems
  - The **maintenance plan** describes how to apply fixes regularly
- In addition to regular scheduled maintenance, you sometimes must apply emergency maintenance to the system, in response to a newly diagnosed problem
  - The **emergency maintenance plan** outlines how to do this task safely and effectively
- Maintenance occurs at all levels
  - Application server and each of the products that are involved in the system
- Keep records of current maintenance levels on the topology diagram
  - Have processes for verifying the maintenance levels regularly
- Consider upgrading to the latest available fix pack during an investigation

Figure 14-10. Create maintenance plans WB8691.0

Notes:

Consider upgrading to the newest available fix pack during an investigation. Individual fixes are meant to be temporary until a fix pack is available. There is considerable risk in using too many individual fixes because it is not possible to test all the possible interactions among individual fixes.

Because of the complexity of the system and difficulty of reproducing problems and gathering diagnostic information, it is not always practical to determine exactly which fix (APAR) resolved a particular situation.

Overall, it is suggested that you have a strong maintenance strategy.

Use Fix Central to download fixes: `http://www.ibm.com/support/fixcentral`

WebSphere Education

Keep a change log

- The important aspect of most troubleshooting exercises is to find out what is different between a working system and a broken one
- Your change log should:
  - Track all software versions
  - Track all upgrades and software fixes that are applied in every software component in the system
  - Track every configuration change in any component
  - Track any known changes in the pattern of the system usage, such as increase in load, or a change in a client’s service request pattern
- As your system grows and becomes more complex, maintaining an accurate, up-to-date, and global change log can be challenging, but a **change log** can help prevent problems

Figure 14-11. Keep a change log WB8691.0

Notes:

Keeping a rigorous log of all changes that are applied to the system over time can help you determine system differences. 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 change log should at least track all software upgrades and 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. It should also track any known changes in the pattern of usage of the system, such as 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 and understand 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 good practices for managing complex systems to prevent problems, as opposed to troubleshooting them.

WebSphere Education

Keep a change log: Constructing an observation report

- Established format for consistently recording your observations during problem determination
- Maintaining IBM Business Process Manager environment information can help narrow down the possible cause of a problem
  - What changed before the problem started occurring?
  - Installation of fix pack or application updates might introduce an error
- Keep a written document or log book in which you record a ***timeline*** of all major events that occurred during the investigation

Sample observation report format:

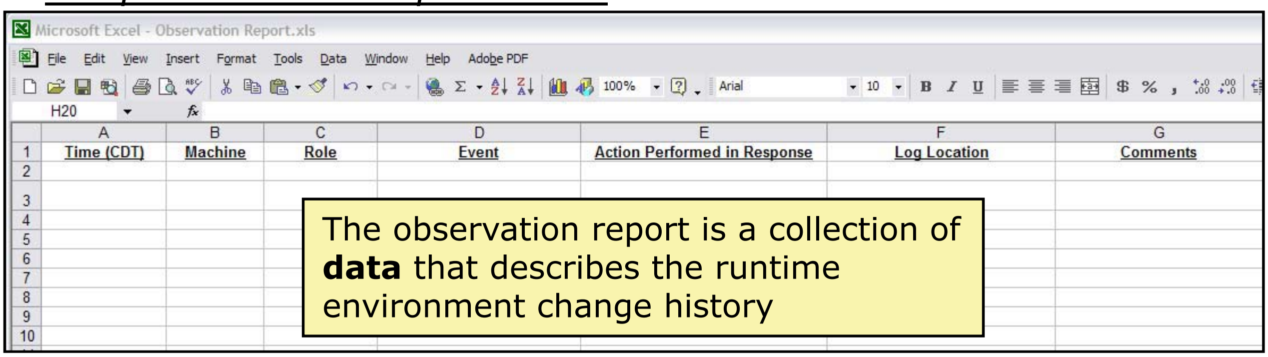


Figure 14-12. Keep a change log: Constructing an observation report WB8691.0

Notes:

The slide shows an example format for an observation report. You can configure the format to meet the needs of your organization.

Create and track an inventory of all problem determination artifacts

- Make an inventory of all the important problem determination artifacts in your system
  - Name of each problem determination artifact, such as file name
  - The purpose of the artifact and typical content and size
  - System architecture diagram helps review the entire system
- Check the live system periodically to verify that all expected log files and other artifacts are still being written as expected
- Ensure that there is enough disk space
- Make sure that these artifacts are not purged too quickly

> - MustGather data is a good example of problem determination artifacts
> - Knowing what they are and how to locate the MustGather data is a good practice

Figure 14-13. Create and track an inventory of all problem determination artifacts WB8691.0

Notes:

WebSphere Education

Watch low-level operating system and network metrics

- When looking for diagnostic information, there is a tendency to focus on the logs, dumps, and other files that are directly associated with the failing component or application
- The underlying hardware, operating system, and network can often provide useful information for tracking down the source of a problem
- System-level metrics:
  – Overall processor and memory usage for the entire machine
  – Processor and memory usage of individual processes that are part of the application
  – Paging and disk I/O activity
  – Rate of network traffic between various components
  – Check for reduction or total loss of network connectivity between various components

Figure 14-14. Watch low-level operating system and network metrics WB8691.0

Notes:

WebSphere Education IBM

Keep what you need

- Periodically purge, archive, or clean up old logs and other data
- In some environments, log files are allowed to grow indefinitely, or old logs and dumps are allowed to accumulate, which can hamper the troubleshooting process
  - Sorting through old information can take considerable effort
  - Diagnostic tools might run much more slowly if they have to transfer a large volume of old, unnecessary information
  - In the worst case, the system could run out of disk space
- Eliminate spurious errors and other “noise” in the logs
  - A large volume of logs that include insignificant error messages during the normal operation makes it difficult to spot unusual errors among all the noise

Figure 14-15. Keep what you need WB8691.0

Notes:

WebSphere Education IBM

Problem prevention best practices

- 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
- 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

Having a good prevention plan saves you time later

Figure 14-16. Problem prevention best practices WB8691.0

Notes:

14.2.Problem diagnostics

This course described how to troubleshoot a runtime problem. To conclude this course, this topic covers steps for diagnosing problems. There are various ways to approach a problem, and numerous tools that you can use. The problem diagnostic steps that are discussed in this topic summarize what you learned throughout the course.

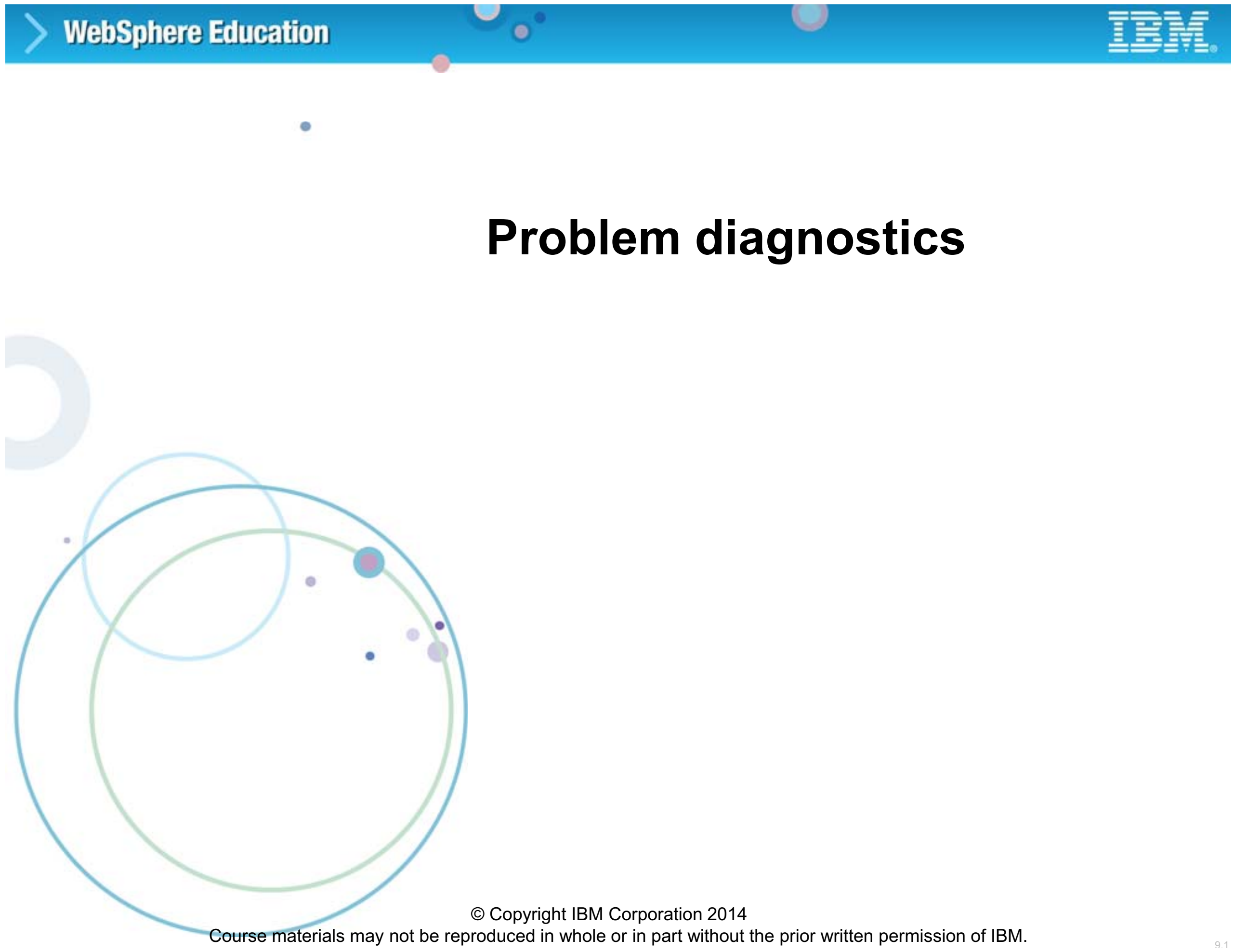


Figure 14-17. Problem diagnostics WB8691.0

Notes:

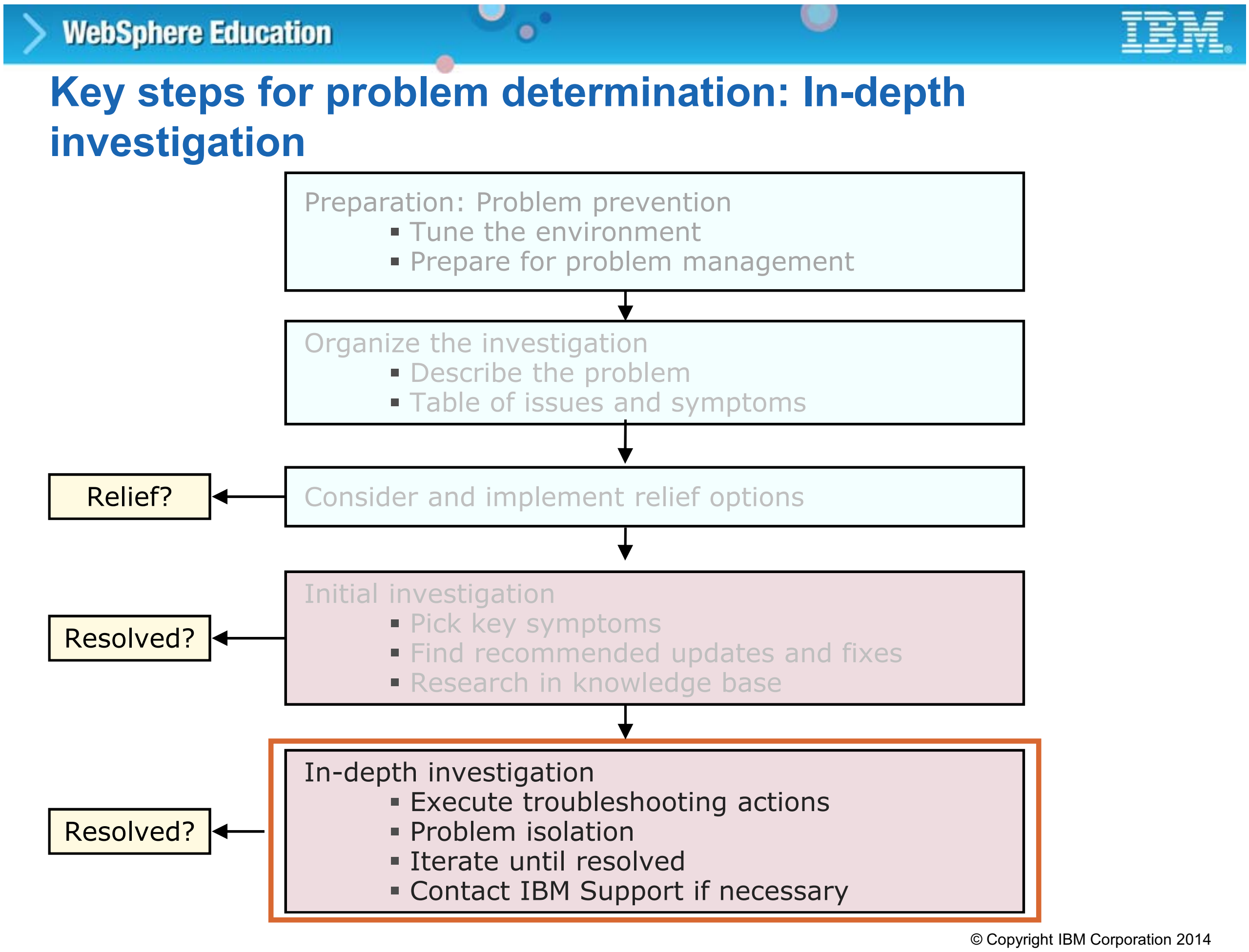


Figure 14-18. Key steps for problem determination: In-depth investigation WB8691.0

Notes:

WebSphere Education IBM

Steps for problem diagnostics

1. Describe the problem
2. Go through the IBM Business Process Manager troubleshooting checklist
3. Collect data to help troubleshoot the problem
   - Collect MustGather data
   - Enable necessary traces
   - Enable cross-component trace
4. Examine the problem by using the logging and tracing data
   - Use the tools that are available through IBM Support Assistant
   - Use IBM Integration Designer tools such as the Server Logs viewer
5. Isolate the component that is throwing the error and debug

Figure 14-19. Steps for problem diagnostics WB8691.0

Notes:

WebSphere Education IBM

Tools that can help troubleshoot the problem (1 of 2)

- IBM Business Process Manager provides the following tools:
  - Failed event manager
  - Business Process Choreographer Explorer
  - Cross-component trace
- IBM Support Assistant provides:
  - Garbage Collection and Memory Visualizer (GCMV)
  - Health Center
  - Interactive Diagnostic Data Explorer (IDDE)
  - MemoryAnalyzer
  - Pattern Modeling and Analysis Tool for Java Garbage Collector
  - Thread and Monitor Dump Analyzer for Java
  - WebSphere Cross Component Trace Logviewer
  - Other tools available

Figure 14-20. Tools that can help troubleshoot the problem (1 of 2) WB8691.0

Notes:

IBM Support Assistant tools

- Garbage Collection and Memory Visualizer (GCMV): Analyzes and visualizes verbose garbage collection (GC) logs
- Health Center: Monitors the status of a running IBM Java virtual machine (JVM)
- Interactive Diagnostic Data Explorer (IDDE): Analyzes artifacts that IBM Java virtual machines (JVMs) produce
- MemoryAnalyzer: Analyzes system dumps and Java heap dumps
- Pattern Modeling and Analysis Tool for Java Garbage Collector: Analyzes verbose GC logs
- Thread and Monitor Dump Analyzer for Java: Analyzes javacore files
- WebSphere Cross Component Trace Logviewer: Helps you view augmented cross-component trace correlation log records

WebSphere Education

Tools that can help troubleshoot the problem (2 of 2)

- Separate products:
  - IBM Tivoli Composite Application Manager for Application Diagnostics: Monitors web and application servers to provide in-depth diagnostics, including garbage collection analysis and method tracing
  - IBM Tivoli Composite Application Manager for Applications: Monitoring, management, capacity planning, and historical analysis for applications
  - Hardware monitoring software
  - Database monitoring software

Figure 14-21. Tools that can help troubleshoot the problem (2 of 2) WB8691.0

Notes:

WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Describe best practices for preventing production problems
- Explain how to prepare for problems before they occur
- List some of the diagnostic tools that can help troubleshoot problems

Figure 14-22. Unit summary WB8691.0

Notes:

WebSphere Education IBM

Checkpoint questions

1. What type of information should your change log contain?
   a) Track all upgrades and software fixes that are applied in every software component in the system
   b) Track every configuration change in any component
   c) Track any known changes in the pattern of the system usage
   d) All of the above

2. True or false: The overall performance of the system might degrade slowly over a long period before it leads to a serious problem. Therefore, you should establish a diagnostic collection plan.

Figure 14-23. Checkpoint questions WB8691.0

Notes:

Write down your answers here:

1.

2.

WebSphere Education IBM

Checkpoint answers

1. (d) All of the above.

2. False. A diagnostic collection plan establishes a set of actions to take when a problem arises. In order to detect problems early, check the live system periodically to verify that all expected log files and other artifacts are still being written as expected.

Figure 14-24. Checkpoint answers WB8691.0

Notes:

Unit 15. 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 IBM

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 15-1. Unit objectives WB8691.0

Notes:

WebSphere Education IBM

Course learning objectives (1 of 2)

After completing this course, you should be able to:

- Describe key problem determination steps
- Describe error prevention strategies
- Identify and collect MustGather data for IBM Business Process Manager
- Use the server logs viewer tool to read logging and tracing files
- Query failed events through Failed Event Manager and resubmit the events after the problem is resolved

Figure 15-2. Course learning objectives (1 of 2) WB8691.0

Notes:

WebSphere Education

Course learning objectives (2 of 2)

After completing this course, you should be able to:

- Describe the exception types from the SCA programming model: service business exceptions and service runtime exceptions
- Examine the trace log to follow the navigation steps inside the business process engine
- Isolate and troubleshoot WebSphere Adapter-related runtime problems
- Explain how to troubleshoot Advanced Integration services (AIS)
- Use Business Process Choreographer Explorer to examine problems with running business process instances

Figure 15-3. Course learning objectives (2 of 2) WB8691.0

Notes:

WebSphere Education IBM

To learn more on the subject

- IBM Training website:
  `www.ibm.com/training`

- Business Process Manager V8.5 documentation on IBM Knowledge Center:
  `www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/ditamaps/ic-homepage-bpm.html`

- IBM Software Support Portal:
  `www.ibm.com/software/support`

- IBM developerWorks (forums, articles):
  `www.ibm.com/developerworks`

Figure 15-4. To learn more on the subject WB8691.0

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 15-5. Unit summary WB8691.0

Notes:

Appendix A. List of abbreviations

A

ACL access control list
ACORD Association for Cooperative Operations Research and Development
AFC Adapter Foundation Classes
AIS Advanced Integration service
AIX Advanced IBM UNIX
Ajax Asynchronous JavaScript and XML
Ant Another Neat Tool
APAR authorized program analysis report
API application programming interface
ARM Application Response Measurement
ASBO application-specific business object
ASCII American Standard Code for Information Interchange
AVS Account Verification Skeleton
AWR Automatic Workload Repository
AWT Abstract Window Toolkit

B

B2B business-to-business
BAL Business Action Language
BAM business activity monitoring
BFM business flow management
BFM Business Flow Manager
BG business graph
BI business integration
BLA business level application
BO business object
BPC Business Process Choreographer
BPD business process definition
BPE business process engine
BPEDB Business Process Choreographer database
BPEL Business Process Execution Language
BPEL4WS Business Process Execution Language for Web Services
BPELJ Business Process Execution Language for Java
BPM business process management
BPMN Business Process Modeling Notation
BRM business rules manager
BSS business support systems

C

C++ C object-oriented programming language
CCI common client interface
CD compact disc
CEI Common Event Infrastructure

CICS Customer Information Control System
CMP container-managed persistent
COBOL Common Business Oriented Language
CORBA Common Object Request Broker Architecture
CPU central processing unit
CRM customer relationship management
CRON Chronograph
CSV comma-separated values
CVS Concurrent Versions System
CW conditioned wait

D

DAAPI data access API
DAS data access service
DB database
DB2 Database 2
DE deployment environment
DESPI Data Exchange Service Provider Interface
DOP defect-oriented problems
DOS Disk Operating System
DTFJ Diagnostic Toolkit and Framework for Java

E

EAI Enterprise Application Integration
EAR enterprise archive
ECI External Call Interface
ECS event correlation sphere
EE Enterprise Edition
EIS enterprise information system
EJB Enterprise JavaBeans
EM event manager
EMD Enterprise Metadata Discovery
EMF Eclipse Modeling Framework
EPV exposed process value
ERC edition revision code
ERP enterprise resource planning
ES event sequencing
ESB enterprise service bus
ESD Enterprise Service Discovery

F

FAQ frequently asked questions
FFDC first-failure data capture
FIFO first-in first-out
FTP File Transfer Protocol

G

GB gigabyte
GBO generic business object
GC garbage collector
GCMV Garbage Collection and Memory Visualizer
GM general manager
GMT Greenwich mean time
GPF general protection fault
GUI graphical user interface

H

HATS Host Access Transformation Service
HL7 Health Level 7
HP Hewlett Packard
HPEL High Performance Extensible Logging
HP-UX Hewlett Packard UNIX
HR human resources
HTM Human Task Manager
HTML Hypertext Markup Language
HTTP Hypertext Transfer Protocol
HTTPS Hypertext Transfer Protocol Secure

I

ID identification
IDDE Interactive Diagnostic Data Explorer
IDE integration development environment
IETF Internet Engineering Task Force
IID instance ID
IIOP Internet Inter-ORB Protocol
IMAP Internet Message Access Protocol
IMS Information Management System
IMS TM Information Management System Transaction Manager
INS Interoperable Naming Service
I/O input/output
IP Internet Protocol
ISMP InstallShield MultiPlatform
IT information technology
IVT Installation Verification Tool

J

J2C J2EE Connector architecture
J2CA J2EE Connector Architecture
J2EE Java 2 Platform, Enterprise Edition
Jacl Java Command Language
JAR Java archive
JAXB Java Architecture for XML Binding

JAX-RPC Java API for XML-based RPC
JAX-WS Java API for XML Web Services
JCA Java EE Connector Architecture
JCL Java class library
JDBC Java Database Connectivity
JDK Java development kit
JDT Java development tools
JIT just-in-time
JMS Java Message Service
JMX Java Management Extensions
JNDI Java Naming and Directory Interface
JNI Java Native Interface
JNLP Java Network Launching Protocol
JPA Java Persistence API
JRE Java runtime environment
JSF JavaServer Faces
JSON JavaScript Object Notation
JSP JavaServer Pages
JSR Java Specification Request
JVM Java virtual machine
JVMPI Java Virtual Machine Profiler Interface
JVMTI Java Virtual Machine Tool Interface

K

KPI key performance indicator

L

LDAP Lightweight Directory Access Protocol
LIFO last-in first-out

M

MBean managed bean
MDB message-driven bean
MIME Multipurpose Internet Mail Extensions
MQ Message Queue
MQMD MQ Message Descriptor
MSO mark stack overflow
MSSQL Microsoft Structured Query Language
MTOM Message Transmission Optimization Mechanism

N

NCSA National Center for Supercomputing Applications
ND Network Deployment
NDOP non-defect-oriented problems
NUMA nonuniform memory architecture

O

OAGIS Open Applications Group Integration Specification
OASIS Organization for the Advancement of Structured Information Standards
OLTP online transaction processing
OOM out-of-memory
ORB Object Request Broker
OS operating system
OSAR Optical Storage and Retrieval
OSGi Open Service Gateway initiative
OSOA open service-oriented architecture
OSS operations support systems

P

PC personal computer
PDF Portable Document Format
PHD portable heap dump
PI Project Interchange
PID Process Identifier
PMAT Pattern Modeling and Analysis Tool
PMI Performance Monitoring Infrastructure
PMR problem management record
PMR program maintenance request
POJO plain old Java object
POP3 Post Office Protocol

Q

QA quality assurance
QoS quality of service

R

RAM random access memory
RAR resource adapter archive
RAS Reusable Asset Specification
RC return code
RDBMS relational database management system
REST Representational State Transfer
RFC request for comments
RMI Remote Method Invocation
RMI/IIOP Remote Method Invocation over Internet InterORB Protocol
RPC Remote Procedure Call
RPG Report Program Generator
RSS Really Simple Syndication
RUP Rational Unified Process

S

SACL State Adaptive Choreography Language

SAP Systems Applications and Products (data processing)

SCA Service Component Architecture

SCDL Service Component Definition Language

SDK software development kit

SDO Service Data Object

SIB service integration bus

SIBus service integration bus

SLA service level agreement

SMB small and medium-sized business

SMO service message object

SMP symmetric multiprocessing

SMTP Simple Mail Transfer Protocol

SNA Systems Network Architecture

SNMP Simple Network Management Protocol

SOA service-oriented architecture

SOAP a lightweight, XML-based protocol for exchanging information in a decentralized, distributed environment. Usage note: SOAP is not an acronym; it is a word (formerly an acronym for Simple Object Access Protocol)

SOI service-oriented integration

SPI service provider interface

SQL Structured Query Language

SR service request

SSL Secure Sockets Layer

SSO single sign-on

STW stop the world

SVG Scalable Vector Graphics

SwaRef SOAP with attachments, referenced

T

TCL Tool Command Language

TCP Transmission Control Protocol

TCP/IP Transmission Control Protocol/Internet Protocol

TIBCO The Information Bus Company

TLS Transport Layer Security

TM Transaction Manager

TMDA Thread and Monitor Dump Analyzer

U

UCA undercover agent

UDB Universal Database

UDDI Universal Description, Discovery, and Integration

UDP User Datagram Protocol

UI user interface

UML Unified Modeling Language

UNIX Uniplexed Information and Computing System

UOW unit of work

URI Uniform Resource Identifier

URL Uniform Resource Locator

UTC Coordinated Universal Time
UTC Universal Test Client
UTE unit test environment
UTF Unicode Transformation Format
UTF-8 Unicode Transformation Format (8-bit)

V

VM virtual machine
VMM virtual member manager

W

W3C World Wide Web Consortium
WAIT Whole-system Analysis of Idle Time
WAN wide area network
WAR web archive
WebDAV Web-based Distributed Authoring and Versioning
WLM workload management
WS web services
WS-BPEL Web Services Business Process Execution Language
WSDL Web Services Description Language
WSIL Web Service Inspection Language
WS-N Web Services Notification
WS-Security Web Services Security
WWW World Wide Web

X

XA Extended Architecture
XCT cross-component tracing
XID transaction ID
XMI XML metadata interchange
XML Extensible Markup Language
XML4J XML Parser for Java
XOP XML-binary Optimized Packaging
XPath XML Path Language
XSD XML Schema Definition
XSL Extensible Stylesheet Language
XSLT Extensible Stylesheet Language Transformation

Z

z/OS 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:
- **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 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/`