Redbooks
ibm.com/redbooks

IBM Business Process Manager Operations Guide

Bryan Brown

Karri S Carlson-Neumann

Mark Filley

Weiming Gu

Chris Richardson

Dave Spriet

Shuo Zhang

Cloud

WebSphere

IBM

Redbooks

International Technical Support Organization

IBM Business Process Manager Operations Guide

December 2016

SG24-8356-00

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

First Edition (December 2016)

This edition applies to Version 8, Release 5, Modification 7 of IBM Business Process Manager.

Contents

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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

COPYRIGHT LICENSE:

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

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

AIX®
Bluemix®
DB2®
developerWorks®
FileNet®
IBM®
Passport Advantage®
PureApplication®
Rational®
Redbooks®
Redbooks (logo) ®
WebSphere®
z/OS®

The following terms are trademarks of other companies:

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

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

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

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

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

Preface

This IBM® Redbooks® publication provides operations teams with architectural design patterns and guidelines for the day-to-day challenges that they face when managing their IBM IBM Business Process Manager (BPM) infrastructure. Today, IBM BPM L2 and L3 Support and SWAT teams are constantly advising customers how to deal with the following common challenges:

- Deployment options (on-premises, patterns, cloud, and so on)
- Administration
- DevOps
- Automation
- Performance monitoring and tuning
- Infrastructure management
- Scalability
- High Availability and Data Recovery
- Federation

This publication enables customers to become self-sufficient, promote consistency and accelerate IBM BPM Support engagements.

This IBM Redbooks publication is targeted toward technical professionals (technical support staff, IT Architects, and IT Specialists) who are responsible for meeting day-to-day challenges that they face when they are managing an IBM BPM infrastructure.

Authors

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

Bryan Brown is an advisory engineer in the US with the BPM SWAT team. He has 20 years of experience in the software engineering field and has worked at IBM for 9 years. His area of expertise is the IBM WebSphere® Software Portfolio, focusing on deep technical solutions.

Karri S Carlson-Neumann is a Senior Software Engineer based out of Rochester, Minnesota. Karri has worked for IBM since graduating from the University of Nebraska - Lincoln in 2000. In that time, Karri has worked in Development on WebSphere Application Server Enterprise Edition v4.1, WebSphere Business Integration - Server Foundation v5.1, WebSphere Process Server v6.0, and IBM BPM since v7.5. Although she is most well-known for her work in enterprise production topologies on-premise, Karri's most recent work is in the Cloud, where she has worked Operations for BPM on Cloud since 2014.

Mark Filley is a Senior Software Engineer in Austin, Texas. He has been working with BPM for 8 years and in Support for 6 years. He primarily works on the BPM Standard product. He holds a degree in Mathematics from the University of Texas a Austin. He is also an active member of the online BPM communities.

Weiming Gu is BPM Performance Architect and the Lead of the BPM Performance Team with IBM in Austin, Texas. Before this position, Weiming was a lead Performance Analyst on performance teams of BPM’s predecessor products and other IBM middleware products. Weiming has 20 years of experience working on software and system performance across the stack from OS and Java Virtual Machine to WebSphere and BPM. Weiming holds a PhD degree in Computer Science from Georgia Institute of Technology, Atlanta, Georgia.

Chris Richardson is an Architect for IBM BPM on Cloud with IBM in Austin, Texas. Chris has held several other positions within the IBM BPM organization, including Architect for High Availability and Disaster Recovery and working as a Lead Performance Analyst on the BPM Performance team for many years. Chris has 18 years of experience working with IBM, almost all of that time focusing on Software Systems Performance. Chris holds a B.S. degree in Physics from the University of Texas at Austin and an M.S. degree in Physics from the University of Washington in Seattle.

Dave Spriet is the IBM on Demand Consulting Competency Leader. His role includes shaping the IBM On Demand Consulting practice, culture, and competencies by working with the service offering teams, consultants, clients, product development, and support teams to ensure clients experience the highest quality of service. Dave’s area of expertise is in BPM and IBM Integration Bus where he engages regularly with clients about advising and assisting with all aspects of their IBM BPM and Integration projects, including architecture, infrastructure, and implementation. Dave has worked at IBM for 18 years where he held different roles within product development and lab services. He has spent his entire career working in the Integration and Smarter Process portfolio.

Shuo Zhang (Joshua) joined IBM in 2003. He is the technical lead of BPM migration and upgrade. He was the architect of the BPM application-by-application migration asset. He also worked in different areas, such as WICS globalization, WICS L3 support, BPM administration console development, WICS migration, and WLE development.

This project was led by **Margaret Ticknor**, a IBM Technical Content Services Project Leader in the Raleigh Center. She primarily leads projects about WebSphere products and IBM PureApplication® System. Before joining the IBM Technical Content Services, Margaret worked as an IT specialist in Endicott, NY. Margaret attended the Computer Science program at State University of New York at Binghamton.

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

Andreas Arning
IBM BPM Development

Nathan Dillard
IBM BPM L2 Support Engineer

Kenneth Eitelman
IBM BPM L2 Manager

Jens Engelke
IBM Security Architect

Andre Mueller
IBM BPM L2 Support Engineer

Lynn Trinh
IBM BPM & ODM L2 Manager

Thanks to the following people for their support of this project:

- Deana Coble, IBM Redbooks Technical Writer
- Ann Lund, IBM Redbooks Residency Administrator

Now you can become a published author, too!

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

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

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

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

- Stay current on recent Redbooks publications with RSS Feeds:

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

1

Introduction and basic concepts

Welcome to the IBM Business Process Manager (BPM) Operations IBM Redbooks publication. This book guides you through an analysis of the concepts that are most important to understand when operating an IBM BPM system.

Each of the contributors to this book has extensive experience in developing the IBM BPM product code, consulting IBM BPM customers as they implement business process applications, and supporting those customers as they run those processes in their production environments. Through this book, the authors explore concepts that every user should well-understand when they are operating an IBM BPM environment.

This chapter includes the following topics:

- 1.1, “Why use IBM BPM operations” on page 2
- 1.2, “Basic topology and nomenclature” on page 6
- 1.3, “Special case for IBM BPM Operations: IBM BPM on Cloud” on page 7
- 1.4, “Helpful resources” on page 7

1.1 Why use IBM BPM operations

Investment in any IT operations activity is useful in reducing wasted effort and ensuring appropriate coverage. Business process management projects in general (and IBM BPM systems in particular) typically integrate disparate hardware and software systems. As such, successfully operating an IBM BPM environment benefits from careful planning, organization, and communication.

1.1.1 What is covered in this book

Every IBM BPM environment is unique. Therefore, every IBM BPM operations team assembles a unique runbook that details the procedures to be followed when dealing with the tasks for which it is responsible. This book does not intend to be that runbook. Instead, it is a guide that you can use when you are constructing such a runbook of your own.

The topics that are included in this book are the most universal topics of IBM BPM systems. You can adapt the experience and guidance that is shared within this book to construct procedures that are appropriate to your own IBM BPM installation.

About IBM BPM versions

The IBM BPM product evolved over many years, with capabilities added (and occasionally removed) many times with new operational interfaces or changes to the way administrative tasks are executed. This book is current as of IBM BPM version 8.5.7 and includes detailed descriptions of system capabilities as of that version. It also provides some detailed description of versions 8.5.5 and 8.5.6 where those versions deviate from 8.5.7.

This book provides information that is specific only to earlier versions (for example, 8.5.0, 8.0 and 7.5) on an exception basis or where necessary. Nonetheless, many of the concepts that are described within this book are still relevant to previous versions, although the commands or syntax might be different. In many cases, understanding the contents of this book and consulting the appropriate product version documentation allows you to apply the same principles back to a previous version.

Not included in this book

This book is an operations guide for IBM BPM. It is not an application development guide. Although process application developers should follow many important practices and business process architects should consider many recommendations to improve maintainability and performance of their code, those practices and recommendations are not within the scope of this book.

This book focuses on an IBM BPM operations audience. Although this book does not intend to be a comprehensive guide for operating an IBM Business Monitor environment, many of the topics that are included here also apply to IBM Business Monitor as well because of similarities in architecture and heritage.

Overview of chapters

This publication includes the following chapters:

- Chapter 2, “Application lifecycle maintenance” on page 9

  An Operations view on the Application Development Lifecycle describes critical touch points between application development teams and the operations teams as a process application moves from conception to production release and ongoing maintenance.

- Chapter 3, “Maintaining the IBM BPM system” on page 35

  Describes the recommended practices when the IBM BPM code is upgraded to a new modification or fix pack level or when you are migrating to a new version or release.

- Chapter 4, “Purging and archiving in IBM BPM systems” on page 51

  Describes the places where data can accumulate within an IBM BPM system over time and the tools and practices that application owners and administrators can use to remove the data when it is no longer necessary.

- Chapter 5, “Maintaining IBM BPM-dependent systems” on page 69

  Examines some of the most common extension points for IBM BPM systems, including database and security servers, and the most common requirements when coordinating maintenance activities between these systems and the IBM BPM system.

- Chapter 6, “Problem determination and remediation” on page 87

  Describes the most important system resources to monitor to detect problems that might be occurring within the system.

1.1.2 Roles and responsibilities

In most cases, the IBM BPM operations team is a composite of many teams, each with distinct responsibilities that are working together. The operations staff members also must work closely with several other important teams to get their jobs done. In many ways, the purpose of this book is to define the responsibilities of these extended teams as they relate to IBM BPM operations. These teams are described next.

Key IBM BPM Operations staff

The following key IBM BPM Operations staff are the primary audience of this book:

- The IBM BPM administrator is responsible for lifecycle management of the business processes and the process instances and tasks that are associated with each. This person (or perhaps team) has many responsibilities, including the following tasks:
  - Receiving working code from the application development team and deploying it to the appropriate IBM BPM environments
  - Determining whether anything goes wrong with a process application or process instance and then correcting the appropriate process application or process instance state
  - Maintaining the IBM BPM installation and product version
- The Operating Systems administrator manages the environment that is hosting the IBM BPM system, including topology design, resource allocation, and monitoring and applying operating system updates.
- The Infrastructure and network administrator is responsible for larger connectivity, network addressability, and routing requirements.
- The WebSphere administrator manages resources that are provided by the IBM WebSphere Application Server Network Deployment environment upon which IBM BPM depends, including pooled threads and database connections, data source definitions, and many of the activities important for tracing the IBM BPM systems.

  The WebSphere administrator also plays an important role in managing many of the artifacts that are associated with IBM BPM Advanced content; that is, Business Process Execution Language (BPEL) processes.

- The database administrator is responsible for the care and maintenance of the database infrastructure that the IBM BPM system depends upon, including managing database memory and CPU resources, code versions, configurations, and indexes. Health and stability of the IBM BPM database is critical to the overall operation of the IBM BPM environment; therefore, the wise IBM BPM administrator maintains a positive working relationship with their database administrator.

Other important roles and responsibilities

The following important roles and responsibilities also are featured:

- The executive sponsor is critical to the success of any project because this person owns the budget that makes the work possible and is responsible for ensuring that business value is delivered through automation of business processes. In many cases, IBM BPM solutions are designed to implement business change, so the role of the executive sponsor is important.

  Close collaboration between the operations team and the executive sponsor ensures that infrastructure resource allocation decisions are in line with the priorities of the IBM BPM project as a whole. For example, the delivery of many non-functional requirements (performance, disaster recovery, and so forth) can incur expensive infrastructure costs. Alignment with the executive sponsor is critical to ensure that these non-functional requirements deliver business value that is appropriate to their cost (conversely, clear articulation of the business value frequently helps to justify that cost).

- The process application developer owns responsibility for the design and implementation of the business processes that are hosted by the IBM BPM environment. In most cases, several distinct process application development teams implement applications that run on the extended IBM BPM environment.

  Understanding the implementation of the business process applications helps the IBM BPM operations teams to interpret resource monitoring data and to evaluate configuration properties. A close working relationship with the process application development teams helps the operations team gain this deeper understanding.

- The Business owner (or business focal point) represents the knowledge workers who interact with the business processes daily. In many cases, these workers are the users who contact the IBM BPM operations team if there is a problem with the environment. Understanding the role of the processes within the business and the ways in which the business users interact with the systems provides invaluable insight into the overall behavior of the IBM BPM infrastructure.

Recommended practice: The IBM BPM Center of Excellence

As described in *Business Process Management Design Guide Using IBM Business Process Manager,* SG24-8282, instituting a center of excellence across all IBM BPM projects helps to define goals and vision across the enterprise, discover and prioritize new processes that should be brought under automation, and share infrastructure resources across all process automation projects. This publication is available at this following website:

http://www.redbooks.ibm.com/abstracts/sg248282.html

The Center of Excellence (see Figure 1-1 on page 5) is a team that includes representatives from executive management, the business community that is using the processes, the application development team that is building the processes, and the IT teams that are operating the environments on which the business processes execute.

Each of the teams that are responsible for the success of the project collaborates well with the center of excellence that is defining the priorities and helping to enable effective collaboration. In many cases, consolidation (rather than isolating the operations teams that are supporting the IBM BPM environment) leads to the most efficient execution.

Clearly defining responsibilities is required within the organizational structure to avoid contradictions and unnecessary duplication of work. However, inefficiency and wasted effort is dramatically reduced when the teams that are working together understand each other well and can contribute to some portion of their collaborator's work.

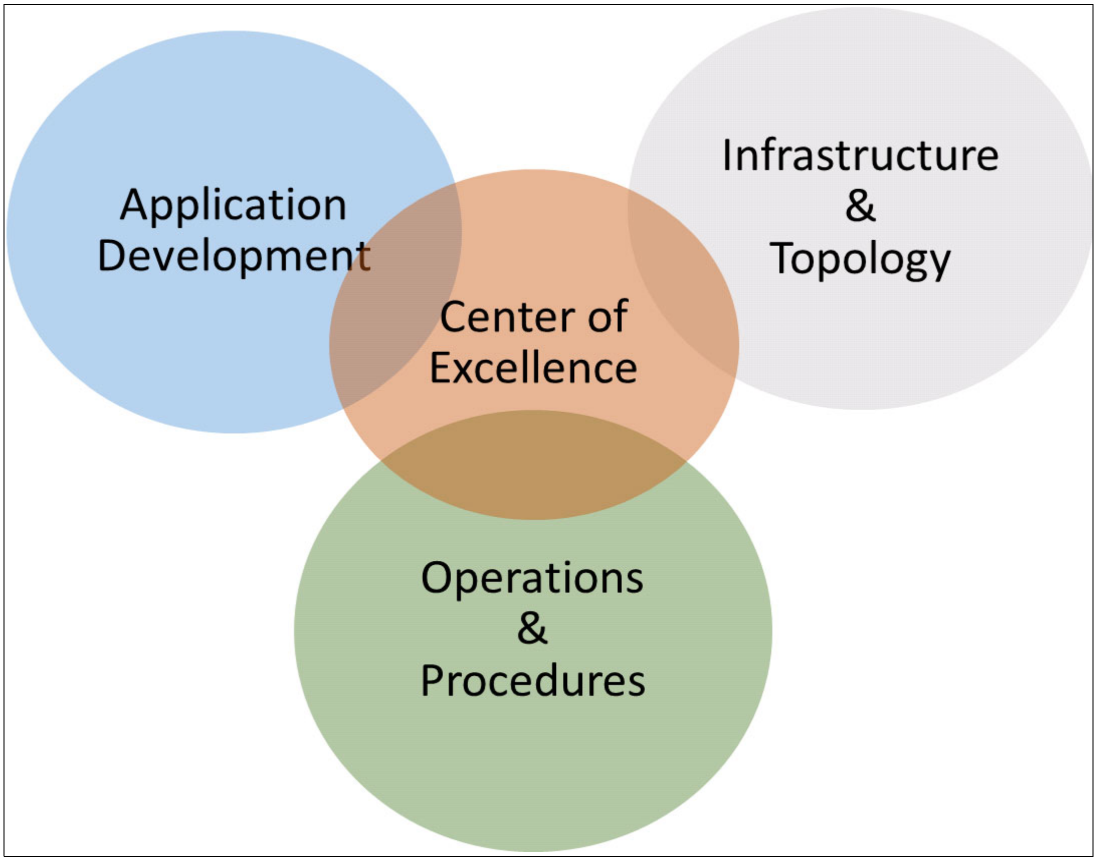


Figure 1-1 Center of Excellence

Discouraged practice: Isolated operations teams

Occasionally, we encounter organizations in which the various operations teams are almost entirely isolated from each other, which leads to inefficient execution. Consider the following examples:

- Debugging an IBM BPM process application frequently requires access to base IBM WebSphere Application Server system resources or tooling (such as enabling trace strings, PMI statistics, or triggering system dumps). When the IBM BPM operations and WebSphere Application Server operations teams work at arm's length, it can be difficult or impossible to trigger diagnostic information at a moment that is most appropriate for the application.
- Gathering diagnostic information from the database system for only a particular moment in the execution of a business process application often requires close collaboration between the IBM BPM administrator and the database administrator (and also, perhaps including the WebSphere administrator).

- Ensuring storage capacity in IBM BPM servers (to hold task search indices or server logs) or database servers (to hold completed business process instances for the duration that is specified by legal requirements) requires cooperation between IBM BPM administrators and infrastructure administrators.

Many more examples are common in IBM BPM installations. The examples that are listed here give you a sense of the types of delivery delays that conflicting objectives (or even simple misunderstanding of project priorities) might cause.

1.2 Basic topology and nomenclature

Throughout this book, we w use some terms to describe the elements of an IBM BPM system and the process applications that run on it. This section defines those terms and places them within the context of the IBM BPM system.

1.2.1 Topology recommendations

IBM BPM recommends a three-cluster topology (Application, remote Messaging, remote Support), as described in Section 3.2 of the IBM BPM v8.5 Deployment Guide. We frequently refer to this topology as a *golden topology*. For more information, see *IBM WebSphere Business Process Management Deployment Guide: Using IBM Business Process Manager V8.5*, SG24-8175, which is available at this website:

http://www.redbooks.ibm.com/abstracts/sg248175.html

This guide is an excellent reference that should be required reading for operations teams that are supporting IBM BPM solutions.

Beyond the basic considerations, Chapter 5, “Maintaining IBM BPM-dependent systems” on page 69 of this book describes many of the interactions between IBM BPM and the systems that form its extended topology (including network and routing, databases, and external service providers).

In many cases, IBM BPM installations make assumptions about services that they “inherit” from the larger infrastructure topology in which they are placed. Services, such as web security, redundant networking, and connectivity with external services (whether within the same datacenter, across data centers, or across infrastructure providers), must be considered. These assumptions often are handled by using corporate policy; however, it is never a good idea to take them for granted without verification.

1.2.2 IBM BPM environments (dev, test, stage, and prod)

Every IBM BPM installation consists of multiple environments, each comprising an IBM BPM cell that is mostly autonomous. In addition to the production environment that executes core business processing, there is a (production class) development environment in which business processes are constructed. The following other pre-production environments also are common:

- A test environment in which basic functional verification occurs.
- A test environment that is dedicated to user acceptance testing.
- A staging environment that is provisioned to match production and is used for load and stress testing.

Discouraged practice: Use of IBM BPM as a system of record

Because IBM BPM stores critical business processing information as metadata that is attached to the business process instances, it is sometimes tempting to use this metadata as the official system of record and retain the primary copy of this information for auditing and historical purposes. However, this practice is discouraged because it overloads the use of the IBM BPM runtime databases, which are optimized for operational activities that are associated directly with helping the business find and complete the most important tasks that are required to do their job.

The accumulation of completed business processing work and executing reports against it (which typically feature different access patterns) degrades the ability of the system to support mainline processing. Therefore, it is better to integrate the business process management system with a dedicated system of record that manages and archives the required primary data for the business and to integrate with a dedicated business monitoring solution for reporting purposes.

1.3 Special case for IBM BPM Operations: IBM BPM on Cloud

For the past several years, IBM offered IBM BPM as a managed and hosted cloud solution. This offering allows subscription-based pricing and separates the infrastructure management tasks (including WebSphere and database administration) from IBM BPM administration. This configuration allows IBM BPM on Cloud customer operations teams to focus their efforts on the IBM BPM administration tasks. For more information about operations within the context of the IBM BPM on Cloud offering, see *Working with IBM BPM Business Process Manager on Cloud for basic daily Operations*, REDP5377, which is available at this website:

http://www.redbooks.ibm.com/abstracts/redp5377.html

For more information about IBM BPM on Cloud (including demonstrations and trials), see this website:

http://www.ibm.com/software/products/en/business-process-manager-cloud

1.4 Helpful resources

As with many complex tasks, many expert resources are available to help you meet your operations goals, whether those goals require short mentoring or full outsourcing of the operations tasks. Many IBM Business Partners are available to help with these tasks and IBM also offers assistance in this area. Contact your IBM account team to find the solution that works best for you. The following resources are available:

- IBM Support:

  https://www.ibm.com/support/entry/portal/support

- IBM Services:

  http://www.ibm.com/services/us/gbs/business-process-management/bpm-consulting/

- Accelerated Value Program (AVP):

  http://www.ibm.com/software/rational/support/tsas/

2

Application lifecycle maintenance

An IBM Business Process Manager (BPM) application consists of a set of Business Process Definitions (BPDs), human or system tasks, services, and so on. IBM BPM application developers are responsible for creating and testing the application code.

The application lifecycle does not end when the code is deployed to the production system and the responsibility is shifted to the operations team. The application must be maintained, updated, and tested over the entire lifespan, from the beginning of development until it is withdrawn from service. This chapter describes the application lifecycle (primarily from the operation's point of view) and emphasizes testing and maintenance.

This chapter includes the following topics:

2.1 Application development versus operations roles

A IBM BPM application has many points of contact with different roles. Application developers develop the code and run unit and functional verification tests to ensure that the solution meets the functional requirement.

Further system integration testing (SIT) and user acceptance testing (UAT) are performed to ensure that the application's behavior at the system level also meets the solution requirement. Non-functional requirements, such as performance, scalability, security, and globalization, also are tested.

It is debatable where performance and stress testing teams (SIT and UAT) belong from the organizational point-of-view. In large enterprises, there often are dedicated testing teams (functional and non-functional). For the purposes of this chapter, we consider these teams a part of the IBM BPM operations.

IBM BPM and WebSphere administrators are responsible for deploying and maintaining IBM BPM applications on the test and production systems. They also are responsible for versioning the performing instance migrations when new snapshots are deployed.

Database administrators (DBAs) also are an important part of the IBM BPM operations. Organizationally, DBAs belong to a different team. However, DBAs must be fully engaged with the rest of the IBM BPM operations to ensure that IBM BPM applications perform well in production.

The application developers and operations teams must work together for successful testing and problem determinations.

2.2 Application lifecycle: Operation's point of view

It is important to ensure that the quality of process applications meet the functional and non-functional requirements of the business. To produce reliable process-based solutions that achieve the business outcomes, you must run functional and non-functional tests. These tests provide confidence that the applications are developed and deployed correctly, that the system is configured properly, and that the system performs and scales well under normal operations or under stressful situations.

A part of quality assurance is to establish a baseline to test against future system changes. These changes are in the form of the following upgrades, changes, and modifications:

- IBM BPM product upgrades
- Server modifications
- Network changes
- Application upgrades
- Application server upgrades
- Database upgrades
- System load dynamics changing over time

It also is important to provide visibility into system scalability to allow for budget planning that is based on anticipated growth in system load, (see Figure 2-1 on page 11).

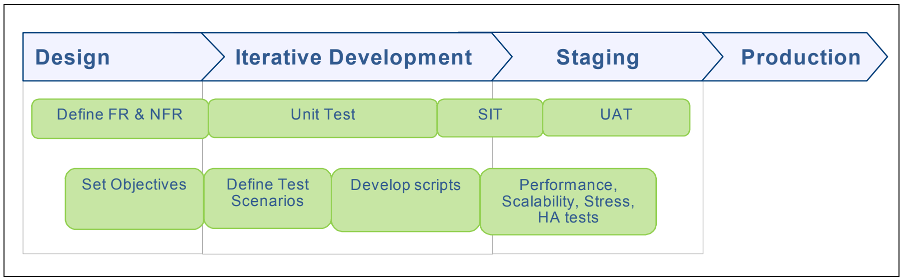


Figure 2-1 Application lifecycle: From design to production

2.2.1 Versioning and continuing testing

New versions of the application can be developed to introduce new capabilities or to address issues, such as simple bugs or improved security. As the owner of operations, you must create policies about application versioning that are based on business requirements, such as deciding whether in-progress process instances must be finished on the current snapshot version or be migrated to the new version. More testing also can be considered when new versions of the application are delivered. The versions can progress from the SIT and UAT testing environment before going into the production environment (see Figure 2-2).

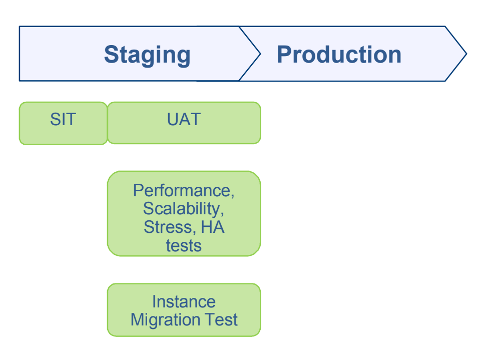


Figure 2-2 Application versioning and testing

The process of testing and versioning should be iterated over the lifespan of the application. Figure 2-2 also shows one iteration of application or system update or upgrade that involves comprehensive testing before deploying the update into the production.

2.2.2 Reviewing and planning

The production system capacity must be planned based on performance testing results. Over time, you must work with business owners to anticipate business growth. For example, if the business anticipates annual growth rate of 10%, and the number of concurrent users is 500, you should expect to support 550 concurrent users within one year. If the current production is expected to support future workload, it must be provisioned based on the projected growth rate.

The operation team should review business needs and system capacity regularly (quarterly or bi-annually). Business requirements change over time. Estimates that are based on previous reviews must be adjusted based on the current state of the business. Future growth must be reevaluated to prepare for the production system capacity.

2.3 Types of tests and their usage

Many terms describe many types of tests. This section defines terminology that is associated with testing and the tests. For example, the term *load test* is often used incorrectly to describe different types of performance testing.

2.3.1 IBM BPM application tests

The following tests are frequently performed for IBM BPM applications:

- Performance test

  A performance test attempts to push the system to the limit to determine the maximum load that can be achieved. The result of performance test is used for capacity planning of a specific environment. For example, if the maximum load of the test is 50 transactions per second, the capacity of the system cannot go beyond the tested throughput.

- Load test

  A load test is similar to the performance test, except that it might not reach the maximum capacity. A load test is to ensure that a specific volume of work (for example, a predefined throughput rate or number of concurrent users) can be sustained on a specified topology while meeting a service level agreement.

- Endurance test

  An endurance test involves testing a system with a significant load over a long period for determining how a system behaves under a period of sustained use. Typically, an endurance test runs for 8 hours or more. The focus of an endurance test is to determine whether flaws in the system appear only gradually over time. For example, a slow memory leak can develop into a serious out-of-memory situation over 48 hours of testing.

- Soak test

  A soak test is another name for an endurance test.

- Stress test

  A stress test involves testing a system with heavy load or high user count. The purpose of this test is to discover how a system behaves under heavier than normal usage. For example, a tax-related system must handle high load just before the tax filing due date, which is April 15 in the United States. A retailer must prepare for the rush of customers and much higher volume during seasonal shopping.

- Failover test

  A failover test is to test how the system behaves when the system or a part of the system fails for whatever reasons. The purpose of the failover test is to identify issues that are associated with system failure and see how the system recovers from normal or abnormal failures. In a topology with high availability, failures in one cluster member should result in the load being routed to the remaining nodes. A single-node failure should not bring down the entire IBM BPM cell. System failures can be simulated through random or specific types of failures.

- Unit test

  A unit test examines the functionality of individual components, such as coaches, coach views, human tasks, and system activities. Unit tests often are performed by the developer to ensure that a function or method does what it is designed to do. For Business Process Model and Notation (BPMN) processes, unit tests can be performed in the process center during the initial development phase. For better testing and to avoid negatively affecting the process center's performance and stability, testing should be run on a process server that is dedicated for testing.

- User Acceptance Test

  User acceptance testing (UAT) is one of the final phases of the IBM BPM solution testing. During UAT, users of the line of the business test the IBM BPM processes and services. This type of testing is to ensure that required tasks in real-world scenarios function according to specifications. UAT is often considered beta testing of the system.

- Integration test

  The integration test focuses on the entire process, including external systems and involves process portal or external user interfaces. Integration testing scenarios must cover all possible paths in the process and consider timers, events, gateways, and so on.

- Instance migration test

  Before a snapshot is deployed into the production environment, an instance migration test is performed. This test ensures that instances of the current snapshot can be migrated to the new snapshot version. It is important that the instance migration test covers problematic cases to ensure that these cases are handled properly. Use the wsadmin `BPMCheckOrphanTokens` command to detect the possibility of orphaned tokens before a new snapshot is deployed and to identify whether to delete or move each token.

  Each process application can use different migration patterns. The migration can drain, which allows the instances to finish on the current snapshot version. An alternative approach is to migrate all process instances to the newly deployed snapshots.

- Browser and mobile test

  Browser testing tests coaches on browsers to be used by users in the production environment. For mobile users, the test is applied to mobile devices, such as smart phones and tablets.

2.3.2 Functional and non-functional tests

Functional testing is a quality assurance process to check whether the process application works as designed. Functional tests use the functional requirements as the exit criteria. For IBM BPM applications, functional tests typically include unit tests, integration tests, user acceptance tests, instance migration tests, globalization tests, mobile, and browser tests.

Non-functional testing is to test a process application for its non-functional requirements, such as performance, scalability, security, and failover. Non-functional tests determine how a system operates, rather than specific behaviors of the system. For IBM BPM applications, non-functional tests typically include performance testing, stress testing, endurance testing, and security testing.

2.3.3 Performance and stress tests

Many people use performance and stress tests interchangeable as though they are the same. Performance and stress tests are different.

Performance testing focuses on testing the most commonly executed path. The test uses metrics, such as throughputs of processes and tasks completed per unit of time, and concurrent users. Performance tests end when the service level agreement (SLA) is violated.

Stress testing focuses on the system surviving under heavy load. In real-world scenarios, a system can be under stressful work load for a short period, often because of bursts of incoming requests, such as batch jobs. If the system survives the burst of activities without falling over, it can recover and continue to work when the volume of workload returns to normal. During stress testing, performance metrics often are not tracked if the system behaves normally. The exit criteria of stress testing is that the system stays alive.

2.3.4 Black-box and white-box testing

Black-box testing is a testing methodology that examines the system without looking in to specific internal components. Black-box testing does not need to know how the system works, and it can be applied to any other tests. Test cases for black-box testing are built around requirements and specifications.

White-box testing scenarios are designed with knowledge of how internal components work. White-box testing can be used in situations where black-box testing is insufficient or not feasible. For example, failover testing is difficult to reproduce in a normal testing environment because it can take too long for failures to occur. For failover testing, failure points can be triggered at specific points in the process applications. For instance, network connections to the backend server are pulled for a period to simulate network failures. White-box testing also can be applied to unit testing, integration, and user acceptance testing.

2.4 Performance testing methodology

Performance testing must be planned and executed from end-to-end. As shown in Figure 2-3 on page 15, performance objectives should be defined at the beginning of the project that is based on non-functional requirements. It is important to clearly define exit criteria for testing to determine whether the IBM BPM application meets the specified non-functional requirements.

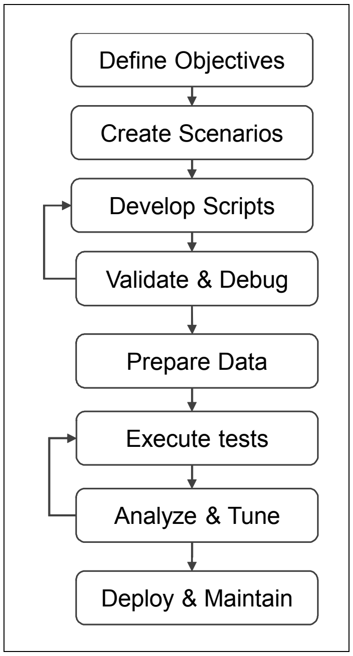


Figure 2-3 Performance Testing Methodology

Many testing scenarios can be defined to evaluate a IBM BPM solution's performance. For most applications, a few key scenarios are sufficient to represent typical application usages by business users.

With scenarios defined, the next step is to develop and debug testing scripts by using your preferred testing tool. Testing tool options can be HP Enterprise's LoadRunner, IBM Rational® Performance Tester, or Apache JMeter. LoadRunner and RPT are licensed application load testing tools, and JMeter is a no-cost open source tool that is distributed under Apache License.

Before testing is performed, a performance testing environment must be set up and testing data must be prepared in the environment. The performance testing environment should be provisioned with the same hardware and software as the intended production environment. A commonly made mistake is that performance testing is conducted on an inadequately provisioned environment (which can be shared with other testing activities).

Prepare the test environment with realistic data, including user and group definitions and their entitlements. Populate the database with active and completed process instances and tasks. Performance results can vary widely, depending on the data that is used in the test.

Ensure that the application code is well-debugged before serious performance testing is run. Buggy application code generates exceptions and errors. Performance testing of buggy code is equivalent to testing how fast errors are generated by the system. It is not a productive use of the testing resources.

Performance testing and tuning is an iterative process. The data-driven performance tuning process starts with measurement. Traces and logs are gathered and analyzed during and after the test. System resources, such as CPU and memory utilization, should be monitored and analyzed. A typical IBM BPM configuration is a multi-tiered environment that consists of the following tiers:

- Browser
- Web server, such as IBM HTTP Server

- Process Server, including Application Target
- Message Engine and support cluster members
- Database server
- Other back end servers

During performance testing, you must monitor and analyze all of these tiers. Bottlenecks in any of these tiers can cause system bottlenecks.

Ideally, tuning changes are made one at a time. Test, analyze, tune a parameter setting, and test again. With too many changes, it can be difficult to determine whether tuning changes help or hurt performance.

During performance testing and tuning, always address the biggest performance bottleneck first. Performance benefits of other tuning changes cannot be clearly visible until the bigger bottleneck is fixed.

2.4.1 Defining objectives

Non-functional requirements are agreements between the business stakeholder and the IT department that is providing the IBM BPM solutions. It is important to engage with business users, business analysts, and executive sponsors. This communication clearly defines the non-functional requirements based on which the performance targets are defined.

User-related targets include the following examples:

- Up to 5,000 business users can access to the system.
- On average, there are 1,000 concurrent users who are using the system at any time (10% of which work on human tasks A, 10% of which work on human task B, and so on).
- From 9 AM - 9:30 AM every day, all 5,000 users log in to the system.
- The response times of task list searches should be less than five seconds.
- On average, each business user works on and completes five tasks per hour.
- Business users often take three minutes to complete each activity.

In addition to user-related non-functional requirements, the following requirements are for how many business processes and tasks are generated:

- There are 10,000 process instances created every day (the peak hour is 1 PM). During this peak hour, 40% of the process instances are generated.
- Each process contains 10 tasks on average. Details about these tasks also are specified.
- Each process takes an average of five days to complete. The completed instances and tasks are kept in the system for two weeks.

Performance targets can be defined based on non-functional requirements. The performance targets should be defined in the following areas:

- Users and groups related targets in terms of concurrent and accessible user counts.
- Throughput-related targets, such as how many process instances and tasks are to be created and completed per unit of time.
- Response time targets as service level agreements.

In addition to business stakeholders, the previous production system is another important source of acquiring non-functional requirements. If the IBM BPM solution is designed to replace a system, you can obtain throughput and usage data. You can use that data to produce concrete performance targets or validate requirements from the business owners.

Testing scenarios

It is important to understand how business users conduct their daily business by using the capabilities that are provided by the IBM BPM application. Their usages can be broken into a few complex end-to-end scenarios or more individual scenarios. Either way, the scenarios must be clearly defined for measuring against previously defined performance objectives.

For example, the following scenarios can be defined for an account opening application at a bank:

- Users log in to the system.
- Bank associates go through steps to collect customer information, enter that information into the system, and then continue the rest of account opening business process.
- Validate customer information and check for potential fraud, and if anomaly is detected, route to human users to perform more checks.
- Branch manager approves account opening requests.

It takes resources and efforts to turn scenarios into test scripts and make them run as performance workloads. For practical reasons, create no more than a dozen scenarios to cover the most commonly used application usage scenarios. Avoid creating too many fine grained scenarios.

2.4.2 Develop testing scripts

Based on defined scenarios, testing scripts must be developed. That process and the tools that are used are described in this section.

Testing tools

To run performance testing, we need tools to generate load to drive the application to do the work it is designed to perform. We must simulate activities from human users or from external services (such as REST calls or UCA messages). The following tools are used for load generation:

- HP LoadRunner
- IBM Rational Performance Tester
- Apache JMeter (open source tool)

Testing script development

You must record scripts by playing back the application through steps that were defined in the testing scenarios. Each tool has its own mechanism for script recording. The script programming languages also differ from one tool to another. Script development is a non-trivial task. It is important to understand the load testing skills of the team that is creating the script development.

After the initial scripts are recorded, you must develop them into testing scripts that implement the scenarios as defined in previous steps. Development includes the following typical activities:

- Organize http activities into transactions and measure each transaction's response time. For example, task list searches and task claims should be grouped into separate transactions and be measured separately.
- Static content, such as images and html code, can be removed or moved outside of measurement transactions. After first use, static contents should be cached in the client browser so that they do not affect runtime performance.

- Different users should be added to the script to simulate multi-user testing. Different user IDs should be used in the testing. Users should be organized into groups, if applicable.
- Input data should be added to the script to simulate different usage patterns.
- Think times and pacing times should be carefully set and added to the script to pace simulated user's activities.

IBM BPM's notification is implemented as CometD long polls, which time out every 30 seconds. When the script is recorded, you can see many such long poll http calls because the recording can take time to complete. In script editing, these long polls should be reduced or removed (this setting is configurable). For more information and product documentation, see the following website:

https://ibm.biz/BdsyUT

Finally, developing testing scripts is an iterative process that requires close cooperation between the test script developer and application code owner. Scripts must be debugged and validated to ensure that they are correctly implemented.

2.4.3 Think and pacing times

At an abstract level, a IBM BPM user often performs the tasks that are shown in Figure 2-4. A user queries for a list of available tasks to work on, chooses one of them to claim, works on the task, and then completes the task. The user then repeats these activities.

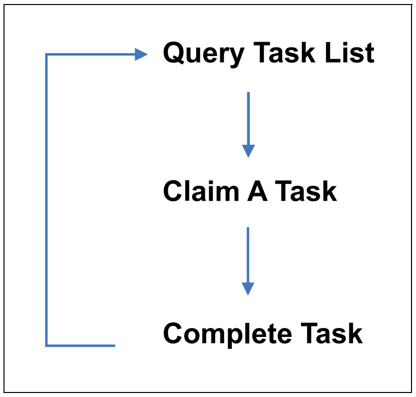


Figure 2-4 Simulated user's activities

Think time is the time that the real user waits between the actions. For example, a user can take 30 seconds between the task list being displayed in the process portal to the time they choose a task on which to work. There also are more fine grained think times between user interface steps that are specific to the application that is tested.

Think times can be expressed as constant times, simple randomized times, or more sophisticated timers. A think time of 5 seconds means that the load generator injects a 5-second sleep time between the steps. With randomized think time, you can specify the minimum and the maximum sleep times with the mean value being the average (if the test runs long enough). For example, a randomized think time of 3 - 7 seconds still means 5 seconds of think time on average over the test run. Randomized think times reflect user behaviors more realistically than constant think times.

Pacing time is the total time between iterations. For example, if a user is expected to finish five tasks per hour, it means that the user takes an average of 20 minutes to complete the round trip of querying, claiming, and completing a task.

JMeter

JMeter does not support pacing time ready for use. However, you can use the cumulative think times to implement pacing times in JMeter.

JMeter's timers have their scope, and they are evaluated before each sampler in which they are found. If there is more than one timer in a scope, all timers are evaluated and the cumulated value of the timers within the same scope is used. For example, a scope has sampler A followed by a timer of 10 seconds and then is followed by sample B with another timer of 30 seconds. The JMeter evaluates the timers before sampler A and uses think times of 40 seconds after sampler A and B. To avoid confusion, avoid having multiple timers within each scope.

Pacing time and throughput time

Pacing times and think times determine how fast simulated users work on the processes and tasks. A few virtual users with no or low think times can put heavy load on the testing system. It is important to set think times and pacing times.

Real human users can take relatively long times to process tasks that are complex, and the process instance can take days, weeks, or even months to complete. For example, a home mortgage application process typically takes several weeks to go through all the steps from the initial application until the closing of the mortgage loan.

For performance testing, it is not realistic to run tests longer than a few hours, at most. It is a usual practice to shorten think times and pacing times to make virtual users work much faster than real users. The number of concurrent *users* in performance testing are equivalent to a higher number of concurrent real human users. When reporting testing results in terms of users, it is important to address the following areas:

- Distinguish between virtual users as compared to real human users
- Explain the relationship between these two metrics based on think time and pacing times

For example, the pacing time of a virtual user's round-trip time is 120 seconds between tasks, and a real human user takes about 20 minutes to finish the job. This means that a virtual user works at 10 times the rate of real human users. In this case, a test with 500 virtual users is equivalent to 5,000 real human users.

2.4.4 Testing data considerations

Before testing is run, the IBM BPM system must be pre-populated with users, groups, and process instances. The data that is used in the test should reflect a realistic production environment. Several considerations regarding the testing data to be used drive performance testing.

Simulated users

Consider the following points when deciding to use simulated users:

- Load testing tools, such as LoadRunner and RPT, are licensed based on the number of virtual users, which often is a limiting factor. Use as many users as possible to simulate real application usage scenarios, especially for test cases, such as logins.
- Use distinct user names, IDs, credentials, entitlements, and so on. Avoid the use of a single or a few user IDs for all simulated virtual users. Reusing the same user rather than unique users might not give accurate performance results.

- Organize users into groups or departments of correct sizes.
- Ensure realistic think times are used between requests, as described in 2.4.3, “Think and pacing times” on page 18.

Input data

The following considerations are related to the input data that is used in tests:

- Use realistic input data in terms of size and data complexity. For example, if your application contains industry standard schemas (such as HIPAA schemas), use them in your test data.
- Use various numbers of fields and nesting levels to cover a range of input data.
- Use various data sizes.

Active and completed instances

Pre-populate the system with a mix of active and completed processes, tasks, and instances. In typical real production environments, it is common for the system to contain work-in-progress business processes. For example, a loan application takes weeks to complete. At any specific time, we should see loan applications at different stages of processing. Consider the following examples when creating your processes for testing:

- Assign tasks to groups and users. Having hundreds or even thousands of users in a single group usually is not realistic. Without proper group structure, the task list can become unrealistically deep, which affects the task search times.
- Start with the same data for each test. This task can be achieved by cleaning up and restoring the test system after each test. Another approach is to always start with an empty system and pre-populate it with the same set of data before each test. If data pre-population takes too much time, the second approach might not be ideal.

Back-end services

Typically, simulated mock services are used in performance and other non-functional testing. Consider the following criteria:

- Ensure that mock services return realistic response data.
- Ensure that the response data is of complexity and size that reflects real production data.
- Notice the response time of the simulated backend services (it should be within SLA for production).

2.4.5 Drain down and steady state testing

There are two popular ways of conducting performance testing: One is to measure the time it takes to address a set of process instances or other inbound requests. The other is to maintain a steady state over the entire testing period. There are advantages and disadvantages to both approaches.

Drain down testing

Drain down testing is easier to set up than steady state performance testing. For drain down testing, you populate the input queue or the database with a many inbound requests. You then run the test and let the system address the requests until all of the requests are processed. For example, you create 10,000 process instances and then test how long it takes for the system to process and complete all these instances.

The use of this approach for testing includes the following drawbacks:

- System behavior can be different, dependent on the state of the processes.
- Be careful about choosing the period to measure performance. For example, most process instance creations occur at the beginning of the test.
- The system can gradually idle at the end of the test run. Much work is completed at the beginning of the test. As time increases, the number of work items becomes smaller, which can make it difficult to calculate proper averages. Figure 2-5 shows an example of drain down testing.

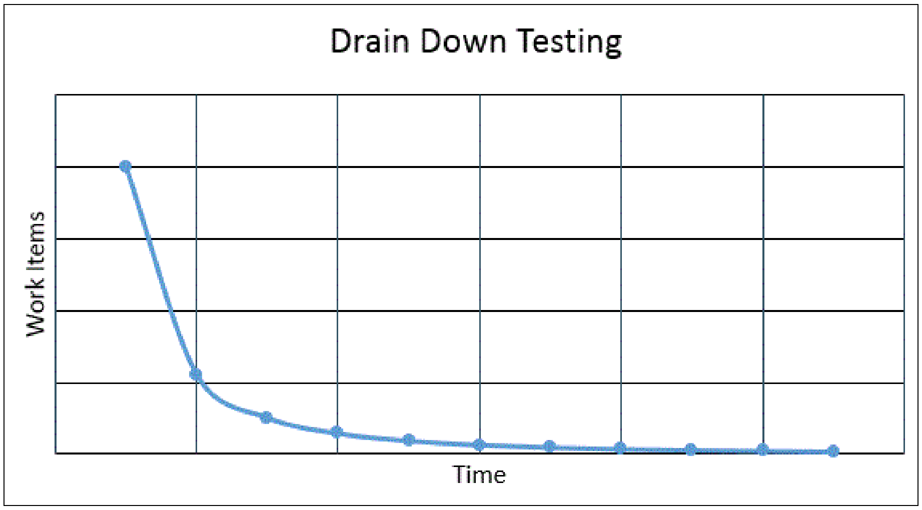


Figure 2-5 Drain down testing

- It is difficult to plan for large-scale tests because all of the process instances are created at the beginning of the test.

Steady state testing

The steady state testing method maintains a consistent flow of work into the system. During the testing, approximately the same number of active process instances and tasks are used as there are at the beginning of the test. You can think of steady state testing as a manufacturer assembly line. New orders come in at one end, and the assembled products exit at the other end. At any time, there should be the same number of work-in-progress products on the assembly line, and workers work at each station focusing on their specific tasks.

Steady state testing requires careful planning. The test driver should control the flow of work into the system by using the following criteria:

- First, pre-populate the system with a mix of process instances at different states.
- Partition users and groups so they complete processes at about the same rate.
- Create process instances and tasks at the same rate as existing instances are completed. The same amount of work should be in the task queues or other internal database tables throughout the test.

Steady state testing often is more realistic than the drain down approach. It is rare for businesses to start from scratch with all work that is created at the beginning. Running steady state performance testing includes the following key advantages:

- It is easy to measure the throughput of the system. Calculate the average instances and tasks that are completed over a period (such as five minutes of steady state run) and you determine the throughput of the system.
- Tests can be run for a short or long period and the results should show consistent throughput.

Steady state testing example

The Human Services Benchmark sample can be found in IBM Bluemix® DevOps Services, which is available at this website:

https://hub.jazz.net/project/ibmbpm/bpm-performance-sample/overview

Documents and sample artifacts for IBM BPM performance testing also are available at this website. You can use the information and adapt or augment it with your own applications and services to fit your specific requirements.

The sample is delivered in IBM BPM V8.5.7 and when the IBM Rational Performance Tester is used. However, the methodology for setting up steady state testing can be applied to other releases and use different load testing tools.

2.4.6 Performance metrics

There are several performance metrics for characterizing the performance of IBM BPM, as described in this section.

Response times

Response times measure how long users wait for their activities, such as refreshing task lists, completing tasks, and selecting the submit buttons. Response times include time that is spent in the browser, in the network for sending requests and receiving responses, and on the server handling the request. In some cases, response times can also include time that is spent in processing the back-end services and database operations.

Response times are often considered SLAs. Business users expect to see response times no greater than 3 seconds.

Response times can be measured as averages or percentiles (typically 90 or 95 percentiles). The measurements also can include both options.

If the measured average response time is significantly higher than the corresponding 90th percentile, it can mean that large outliers are lifting the average. This result can be a cause for further investigation.

Throughput

Throughput defines how many processes, tasks, or other units of works are completed in units of time (such as per second, per hour, or per day).

Consider peaks and averages when throughputs are defined. For example, the throughput of tasks that are created can peak at 10 AM and 1 PM each day. It is important to test the system to handle peak throughput rate, which can be significantly different than the average throughput rate over the entire work-day.

Business users do not always see throughput as an important performance metric (partly because it is not readily visible to many of them). You must identify and connect to the correct business process owner in your organization to obtain the throughput information.

Concurrent users

The number of concurrent users is a metric that readily resonates with business process owners and the executive sponsor. However, this metric also often is a misrepresented metric. Therefore, it is important to report it correctly. The number of simulated virtual users might not be the same as real human users. As described in 2.4.3, “Think and pacing times” on page 18, a simulated user can exert a heavier load on the system than a real human user. Take both think times and pacing times into consideration when calculating supported concurrent users as a performance metric.

Resource utilizations

Performance metrics that track the resource utilizations on the IBM BPM and DB servers also are important for performance testing. The following typical metrics are captured by monitoring tools:

- CPU utilization on both IBM BPM and database servers
- Memory utilization
- I/O and network utilization

2.5 Tuning the system and caches

Optimal tuning is an iterative exercise. It is also a collaboration between Application Development and Operation teams. Tuning continues during application performance testing and after the application is in production. Changes in the use of the production system can require a tuning change of scaling up or scaling down.

The following common usage patterns can require a change in tuning parameters:

- New applications
- Increase in number of concurrent users
- Increase in daily or weekly task and instance generation
- Increase in data size that is sent to IBM BPM (for example, more document attachments or increased JMS messages to be addressed)
- Total number of tasks and instances in the system
- Total number of applications and IBM BPM assets in the applications

For more information, see *IBM BPM Performance Tuning*, SG24-8216-00. The relevant sections are section 2.5 Large Business Objects and Chapter 4: Performance Tuning and configuration. The publication is available at the following website:

http://www.redbooks.ibm.com/abstracts/sg248216.html

Application caches and definitions are documented in the latest release. These settings must be reviewed during load testing scenarios. For more information about the configuration options, see the following documentation:

- https://ibm.biz/BdsMDE
- https://ibm.biz/BdsMDX

Some performance metrics see a positive improvement after data is deleted. With less data in the system, database queries can perform better. For more information, see Chapter 5, “Maintaining IBM BPM-dependent systems” on page 69.

2.6 Application versioning

IBM BPM applications are written with two clients. The IBM BPM Standard (BPMN) is written in the IBM Business Process Execution Language (BPEL) and with the IBM Integration Designer. The applications contain all code, logic, and assets for your business process. This section describes considerations that the Operations team uses to ensure quality delivery of applications and business data.

For more information about deploying applications, see *IBM BPM Adoption: From Project to Program with IBM Business Process Manager*, SG24-7973, which is available at this website:

`http://www.redbooks.ibm.com/abstracts/sg247973.html`

2.6.1 Naming conventions and exports

IBM BPM features two application types: BPEL and BPMN. The IBM Integration Designer creates the BPEL applications; the exports are project interchange (PI) files. BPMN applications are written with the process designer and include process exports (TWX) files.

BPMN Applications that are written for IBM BPM Standard are stored in the Process Center. It is a best practice to use a naming convention for the snapshots and exports. For more information, see the application snapshot naming convention best practice documentation that is available the following websites:

- `https://ibm.biz/BdsMRC`
- `https://ibm.biz/BdsMRQ`

Naming conventions help reduce collisions when snapshots are installed and allow for easier maintenance and troubleshooting when issues occur. For example, if the export features a version and date in the name, it assists with identifying instances and the snapshot on which they were created.

2.6.2 Toolkit dependencies

Toolkits are a way to have common services and assets that are shared by several applications. The sizing of the environment depends on the total number of applications and dependent toolkits that are needed during run time. The cache setting `<branch-context-max-cache-size>` controls the number of snapshots in memory. It is important that applications use the latest toolkits to reduce the number of toolkits versions in production.

For example, suppose that there is a toolkit that is named Common Utilities that contains logging and other services that are shared by the applications. There are 25 applications in production. These applications rely on the toolkit, and there are four active versions in production of the Common Utilities (v1.1, 1.4, 2.0, 2.2). Because this library is used by all applications, it needs four copies in the memory of the Common Utilities. This amount can be reduced to one copy if all applications used the latest version (v2.2).

This example has only one toolkit; however, it is common for an application to have several toolkits and those toolkits to depend on other toolkits. The dependency chain can be long. To reduce the number of active versions of toolkits, the Operations team can work with the Applications team to help progress applications and the toolkit dependencies.

For more information about guidelines for toolkit best practices, see Section 5.5 of *Business Process Management Design Guide: Using IBM Business Process Manager*, SG24-8282, which is available at this website:

http://www.redbooks.ibm.com/abstracts/sg248282.html

2.6.3 Dependent assets

When the IBM BPM applications are written, it is common to use more libraries. These libraries can come from third-party sources or from an approved library of code within the company. These files are added to the IBM BPM application and are contained within the application and thus the export of the application. It is recommended that the added assets are versioned and the versions are included in each IBM BPM application are known.

The following assets are included in an IBM BPM application:

- CSS file
- JavaScript library
- Graphic images
- Compiled Java code
- XML and XSLT files

A list of the third-party dependencies is important for the Operation team. It can be used to get notifications of new updates, such as found security vulnerabilities or critical bug fixes. Collaboration between Applications and Operations teams continues further. For example, within the company, a common library needs to conform to new corporate standards and all dependency assets must be updated by a certain date.

The Process Center is a development environment and repository for your applications. As application milestones are reached, it is recommended exports of the application (PI or TWX) are saved in a company's official code repository. This export is another backup and an official archive of assets.

2.7 Instance migration and application deployment

When an application is ready to deploy to a target runtime environment, there are some considerations for active instances and various recommendations for successful application deployments.

2.7.1 General checklist

When the application has an application to deploy, the Operations team might be responsible for deploying the application. This responsibility depends on your company's separation of duties and into which environment the application is deployed. For example, the production system might be firewalled. Therefore, select individuals and a strict process only are followed to ensure quality and auditing.

It is a reasonable expectation for the Operations team to have a checklist to review with the Application Development Team. The Operations team is responsible for maintaining the system and ensuring the highest operational run time. A best practice is to have a requirements checklist before approving the deployment of an application to a runtime environment.

Table 2-1 lists the example application deployment requirements as a guideline for consistent application deployments. These requirements are checked and verified before code is deployed into the target environment.

Table 2-1 Example application deployment requirements

Activity	Test	Staging	Production
Naming convention followed for snapshot names	X	X	X
Unit testing of Application	X	X	X
User functional		X	X
Load testing: stress			X
Load testing: endurance			X
Team role bindings	X	X	X
Clean up instances before deployment		X	X
Instance migration policy	X	X	X

2.7.2 Offline and online deployments

Depending on the topology of the system and security requirements, deploying an application can be online or offline. The runtime process servers are visible to the Process Center when *online*. In this configuration, applications can be installed with the web-based Process Center console.

Offline means that the runtime system is not directly connected to the Process Center. Deployment to an offline server is done by command line access and requires the export file to be physically moved to the target system. For more information about deploying applications, see the following resources:

- https://ibm.biz/BdsMR7
- https://ibm.biz/BdsMRW
- https://ibm.biz/BdsMFc

Closed production with Process Center

Regarding the production system, some customers have a dedicated Process Center from which to deploy code. This environment is isolated and is effectively an offline server in respect to the primary development Process Center. The production Process Center has an online connection to the production runtime system and can deploy applications in the online fashion. One advantage to this scenario is the Process Designer gives a graphical view of tokens on active process. The requirement for this option lessened because troubleshooting issues in production can be done in IBM BPM 8.5 with the BPM REST UI helper. You can access the console by using the following URL:

```
http://hostName:9080/bpmrest-ui
```

For more information about troubleshooting tips, see Chapter 6, “Problem determination and remediation” on page 87.

2.7.3 Closed and in-flight instances

When deploying an application to a runtime server, consider what happens to an instance on the server. There are three options for addressing instances on the target system that belong to snapshots: leave, migrate, and delete. The delete option is not available in systems that are configured as production. Discuss with the application development team which option is needed for the application version you are installing.

Some applications have small changes and migrating instances is appropriate. Some applications have larger modifications and need multiple running versions of the application. Deleting instances is a common request for a test environment. For load test scenarios that test the same application again but with a new version, it is advised to purge the instance data from the previous test. For information about data clean up, see Chapter 5, “Maintaining IBM BPM-dependent systems” on page 69.

Avoiding orphaned tokens and tasks

For cases where the next application (snapshot) is to migrate instances, ask the Application Team if a migration policy file (BPMN) is needed. If long-running instances are used or the application changed in a significant way, a migration policy file might be needed.

For IBM BPM, the managed policy file is necessary to move tokens to prevent instance failures. For example, if an activity is divided into two new activities, a policy file is needed to move tokens to the appropriate activity. Without the policy file, the failure tokens can become orphaned and instances fail and require manual intervention. For more information, see Chapter 6, “Problem determination and remediation” on page 87. Options to recover can be to move the token, change the instance data, or roll back the snapshot installation.

For more information about commands to help prevent orphaned tokens and tasks, see the following resources:

- BPMN Application:

  https://ibm.biz/BdsMER

- BPEL Application:

  https://ibm.biz/BdsMEF

> **Note:** Back up the production databases before new applications are installed to production. This step is important for your recovery planning. With a backup, the IBM BPM database can be rolled back to the state before the snapshot installation. The rollback option can be used in the following cases to restore production functionality:
>
> - If during the migration of instances, a significant portion of instances were orphaned in an unrecoverable state because an application change was not considered during testing.
> - If ending and deleting unclosed instances occurs before new snapshots are in production, business data can be accidentally deleted.

2.7.4 Multiple active versions of snapshots

There are times in which two or more versions of an application can be active in Production. Discuss with the development team how each of the versions handles new and existing instances. One option is to have existing instances on the old snapshot complete, and allow only the creation of instances on the new snapshot. For more information about these settings, see the following website:

https://ibm.biz/BdsMEX

It is reasonable for the Operations team to require the Development team to perform the following tasks:

- Test various scenarios with data
- Test the migration of instance data
- Create orphan token policy files for changes to applications

Running tests before the new version is in production reduces the risks of having failed instances that need an emergency manual recovery.

Migrating product version by using active instances

When migrating to the next major version of IBM BPM, in-flight instances must be considered. The two options are to migrate to the in-flight instances or perform a drain and fill approach. The drain and fill approach is used when two versions of IBM BPM are active. The new IBM BPM system has two parallel production environments and new instances are on the current version of IBM BPM. The older instances are on the previous version of IBM BPM.

One configuration option with drain and fill is to have two different portals for interaction with users. A second option is to use the federated process server. This server allows for a unified process portal for users. The interaction between the new and the old system is to be transparent to users.

For more information about the federated process server setup and installation, see this website:

https://ibm.biz/BdsME4

In some cases, the drain and fill approach is not appropriate for your application. Some considerations are the length of running instances and how IBM BPM interacts with external systems of record. For example, if the external system of record uses some of IBM BPM’s primary keys (for example, task ID and instance ID), this approach cannot be used. Work with the Application Development team to develop a strategy when migrating to the next version of IBM BPM.

2.8 Capacity planning

Any sizing and capacity planning methodology is an estimate that is based on input variables. It is not exact and it should be validated against the actual variables in your system. There are a few guidelines that are described in this section that are based on the authors’ experience in working with many customer situations.

Capacity planning should be data-driven and based on performance testing of the actual application to be deployed in the production. IBM BPM is a middleware product. The performance of the application and capacity of the system depends on the application’s implementation, choices of system capabilities that are used, and the type of data that is processed. No estimation is better than actual measurement.

With proper performance testing, capacity planning is relatively straightforward. Complete the following steps to guide performance testing:

1. Provision the performance testing environment with a hardware configuration that is similar to the production environment.
2. Run performance tests and gradually increase the load until SLAs are violated.
3. Capture CPU, memory, and other resource utilizations.
4. Use the testing results to project the required hardware configurations for the IBM BPM server and the DB servers.

Consider the following points when extrapolating results:

- Use conservative scaling factors for adding cluster nodes. Horizontal scaling typically starts to decline rapidly when the cell contains too many nodes; for example, six nodes. If you have a large IBM BPM cell, run your application-specific performance testing to ensure that the cell scales well.
- It becomes risky to extrapolate beyond twice of what was measured. Many parameters can affect a IBM BPM solution's scalability.
- All tiers should be considered, including the web tier, database, and backend service providers. IBM BPM is a multi-tiered environment. Any of these tiers can become a bottleneck.
- Consider the capacity of the network and I/O. For example, the storage subsystem of the database server can become bottle necked after its I/O capacity is maxed out. The network between the IBM BPM and the database server can often reach the maximum capacity at a high transaction rate.

Memory considerations

JVM heap sizes should be tuned during performance testing. For the physical memory sizing, we use the following standard formula (rounded up to the next 4 GB boundary). It is the minimum physical memory that is needed for the server:

```
2GB (OS) + for each JVM: (Maximum Heap + 1 GB)
```

For example, if JVM settings for AppTarget, Support, and Message Engine cluster members are 4 GB, 2 GB, and 2GB (respectively), the total physical memory of the IBM BPM server should be at least `2 + (4 + 1) + (2 + 1) + (2 + 1)` rounded to 16 GB.

For database server memory sizing, consult each DB vendor's recommendation. In general, the database server must be provisioned with enough physical memory to accommodate buffer pools and other in-memory cache to support efficient execution of typical database queries.

Disk and file systems

Capacity planning should also consider disk and file system spaces for the IBM BPM server and the DB server.

IBM BPM server storage capacity

The IBM BPM server uses the file system to store log files, Lucene indexes, and other temporary files. A standard is that each server should reserve at least 100 GB free disk space. For example, if an out-of-memory error ever occurs, the JVM can trigger heapdumps of multiple gigabytes, dependent on the size of the Java heap.

The following sample directories include log data:

- `install_root/profiles/profile_name/logs/server_name`
- `install_root/profiles/profile_name/`

For more information about how to gather logs and which directories contain log and temporary information, see the following resources:

- IBM BPM MustGather:

  http://www.ibm.com/support/docview.wss?uid=swg21569731

- Things to know before deleting temporary, cache, and log files in WebSphere Application Server:

  https://ibm.biz/BdXeAn

Database server storage capacity

IBM BPM uses databases to store persistent data, such as process instances, tasks, searchable variables, and deployed artifacts. Many tables are defined in IBM BPM databases and some of these tables require more storage than others. For sizing estimates, we can focus on the storage requirement of these large tables.

The following key variables affect the size of the IBM BPM DB:

- Number of processes: Total number of process instances in the database, including all active and completed (but not yet deleted) processes.
- Number of tasks: Total number of user and system tasks in the database, including all active and completed (but not yet deleted) tasks.
- Number of documents: If embedded ECM is used, you must know how many documents are stored in the database.
- Average size of process instances.
- Average size of tasks.
- Average size of documents.
- Database transaction and archive logs.

How to obtain values of the list of variables? You can get some of the data points, such as number of processes, tasks, and documents from solution architects and business stakeholders. You can also find the average size information from the database directly. For example, The top nine tables, which are sorted by the size that are related to BPD process instances and tasks, are listed in Table 2-2 on page 31. The data is captured after running performance testing with an internal IBM lab workload that is named BPMBench v8. The top nine tables account for 70% of total IBM BPM database sizes.

Table 2-2 IBM BPM database table sizes

Table name	Total size (MB)	Data size (MB)	Index size (MB)	Row count	Comment
LSW_TASK_EXECUTION_CONTEXT	2,935	2,915	16	372,931	Task related
LSW_BPD_INSTANCE_DATA	868	860	4	100,000	Process related
LSW_TASK	430	213	217	372,931	Task related
LSW_BPD_INSTANCE_VARIABLES	413	287	126	2,000,000	Searchable variables, process related
BPM_TASK_INDEX	75	24	47	372,931	Task related
LSW_BPD_INSTANCE	75	31	40	100,000	Process related
LSW_TASK_NARR	68	38	26	372,931	Task related
LSW_TASK_ADDR	48	21	27	372,931	Task related
LSW_BPD_INSTANCE_VARS_PIVOT	43	39	4	100,000	Process related

For typical BPMN applications, the top-most tables in terms of sizes are almost always related to tasks and process instances. As the tables imply, table LSW_TASK_EXECUTION_CONTEXT stores the execution contexts of all tasks (including all variables that are used in the task) and table LSW_BPD_INSTANCE_DATA, which stores the contexts of BPD process instances.

The amount of DB storage per process and per task can be calculated from the data that is listed in Table 2-2 on page 31. In this example, the DB storage per process instance is 14.3 KB, and the DB storage per task is 9.8 KB. No embedded ECM documents are in the processes. For applications with documents, the corresponding document table can appear at the top of the list.

The formula to estimate the size of IBM BPM database is shown in Example 2-1. The total size is the sum of processes, tasks, and documents divided by the percentage that is accounted for by these tables.

Example 2-1 Formula for estimating size of IBM BPM database

```
Total BPM DB size =
    (num_processes x avg_process_size +
     num_tasks x avg_task_size +
     num_documents x avg_document_size) / %_of_top_tables
```

For example, by using averages that are calculated from Table 2-2 on page 31, the total size of BPMDB is estimated to be the following size for a production system with 1,000,000 process instances and 4,000,000 tasks:

```
(1,000,000 * 14KB + 4,000,000 * 9.8 KB) / %70 = 72.5 GB
```

When requesting storages for the DB, you must take variability into account. Build into estimates a margin for error (minimum of 2X).

Also, the method that is described in this section describes only the estimation of BPMDB. A similar approach can be used to estimate the sizes of PDW and other databases.

For more information about purging and archiving data in IBM BPM, see Chapter 4, “Purging and archiving in IBM BPM systems” on page 51.

Application repository tables

Applications contain the coaches, business process definition (BPDs), managed assets, and other artifacts to run the application. Depending on the size of the applications and the number of snapshots deployed, the table that contains these assets might become large in disk space.

The tables to watch are LSW_SNAPSHOT and LSW_PO_VERSIONS. Customers who migrated from versions where the snapshot delete command was not available (WebSphere Lombardi Edition and IBM BPM 7.5.1) should plan to delete snapshots. For more information, see Chapter 4, “Purging and archiving in IBM BPM systems” on page 51.

Review capacity regularly

Review the resource usages by the IBM BPM and database servers regularly, such as on a quarterly or bi-annual basis. Create a checklist to review and include the following areas:

- IBM BPM server CPU utilization, averages, and peaks
- IBM BPM server memory utilization, averages, and peaks
- IBM BPM server free disk spaces
- DB server CPU utilization, averages, and peaks
- DB server memory utilization, averages, and peaks
- DB server free disk spaces

2.9 Anti-patterns

There are a few common pitfalls that can be navigated or bypassed. To do so, avoid these anti-patterns when you plan and run testing of your IBM BPM solutions. The list that is featured in this section is far from complete. For more information, see other relevant publications and references.

2.9.1 Run testing on the Process Center

The Process Center featured two-fold capabilities. It is a repository server for storing and managing authored artifacts and is a Playback server for playing back newly created artifacts, such as coaches, coach views, tasks, integration services, and BPDs.

As a Playback server, it has all the capabilities of a process server. Performing testing on the Process Center can result in Process Center overload. This situation negatively affects its performance and thus affects other developer’s productivity.

Performance testing results from Process Center also is invalid for the following reasons:

- The Process Center is not provisioned for proper performance testing.
- The application can run a path in Process Center that is different from the application in Process Server. For example, some runtime caches are disabled on the Process Center to enable playing back artifacts under development.

Instead of running tests on the Process Center, set up a dedicated testing Process Server and run tests on it.

2.9.2 Run performance testing on inadequately provisioned environment

For proper performance testing of the end-to-end IBM BPM application, the testing environment should be provisioned and configured in the same way as the production environment. For numerous reasons (cost, poor planning, and so on) some customers run performance testing on a much smaller environment than the target production environment. The results are often extrapolated to estimate whether the production environment can handle the load. Relying on extrapolation includes the following pitfalls:

- Performance results do not always scale linearly.
- Resource bottlenecks, such as DB I/O, can be detected by testing only.

It is important to plan ahead to obtain production-like systems for testing. It takes time and effort to overcome funding obstacles, bureaucracies, skill gaps, and so on.

2.9.3 Run performance testing before application code is debugged

For performance testing, the application code should be reasonably debugged and free of application errors or exceptions. Performance testing of buggy code tests how fast errors and exceptions are generated.

2.9.4 No or low think times between activities

Testing scripts with no or low think times can create high load on the system that is under test. Such testing scenarios are good for generating stressful situations, but they do not reflect realistic human activities.

2.9.5 Use unrealistic data

An unrealistic data anti-pattern includes the following attributes:

- All virtual users use a single user ID.
- Hundreds (or even thousands) of users are grouped into a single group, which results in a deep task list.
- Users are granted with extraordinary entitlements.
- There are too many or too few active instances or tasks.

The use of unrealistic testing data can produce testing results that are confusing and difficult to interpret.

2.9.6 Common mistakes for IBM BPM testing

IBM BPM testing can include the following common mistakes:

- Lack of realistic business requirements
- Focus on rarely used operations
- Measurement period covering only the task or instance creation
- Database server not properly tuned or under provisioned
- Testing on overcommitted virtualized hardware

2.9.7 Application goes into production without load testing

Allowing applications to go into production without any load testing is one of the worst anti-patterns. Without proper load testing, there are high risks that the production environment can encounter in the future. It ultimately costs more than the savings from bypassing load tests.

3

Maintaining the IBM BPM system

This chapter describes how to maintain the IBM Business Process Manager (BPM) System. The following areas must be considered when you move to a higher version:

- Production binary
- Configuration
- Process applications
- Runtime data

A preferred practice is to upgrade or migrate to the latest IBM BPM version as possible. This update allows your organization to use the latest improvements and fixes. The production binary is owned by IBM and is automatically updated by the IBM Installation Manager when you upgrade or migrate to the new IBM BPM version. The configuration and the process applications can be set to default or modified to a specific version during this process. Interaction is required to maintain them to the new version. The runtime data that is generated in daily business must be transformed to the new format during the upgrade or migration.

Each version of the IBM BPM product includes an end of service (EOS) date. To ensure first-class support and better performance and new features, upgrade or migrate the IBM BPM environment before the EOS date. It usually takes months to complete a successful version-to-version migration. If the update is from an older version, the update process might take longer. A year or at least 8 months before EOS date to start a migration project is suggested.

For more information about product versions and EOS dates, see the following website:

http://www.ibm.com/software/support/lifecycle/

This chapter includes the following topics:

3.1 Production migration and upgrade overview

One method for protecting your assets is staying current and using the newest versions of products through migration or an upgrade. The following assets are included in the process:

- Process application
- IBM BPM configuration
- Runtime data (such as process instances and tasks)
- System data (such as users and groups)

3.1.1 Terminology

The following terms are used in this chapter:

- In-place

  This term refers to updating the artifacts and overwriting the original version. If the artifacts are in-place updated, the original version and new version do not coexist. Therefore, you must back up the artifacts in advance if you must roll back the in-place changes.

- Side-by-side

  This term refers to updating the artifacts and writing to another location without overwriting the original version. Typically, the side-by-side updated artifacts do not need to be backed up in advance. The original version and the new version coexist. During a rollback of the changes, the side-by-side method is faster than the in-place method.

- Upgrade

  This term refers to an in-place upgrade of the process server binaries and database. The in-place upgrade is completed without requiring any application migration or runtime database migration. The upgrade procedure is used to stay current with minor changes of different versions. It is also considered to be a lower risk and a faster function than migration.

- Migration

  This term refers to those cases in which server binaries cannot be upgraded and a migration process must be used. The migration procedure manages the major differences between versions, such as when the underlying WebSphere Application Server version is changed and does not support upgrade from the older version. The IBM BPM configurations and the process applications (Advanced edition) are side-by-side migrated from the source version to the target.

  The migration process typically involves converting the database to the new version. The conversion of the database is in-place and reuses the database instances rather than creating a target version database instance. Migrating IBM BPM might require that you upgrade the database instance to a higher version before the IBM BPM migration is run.

- Instance migration

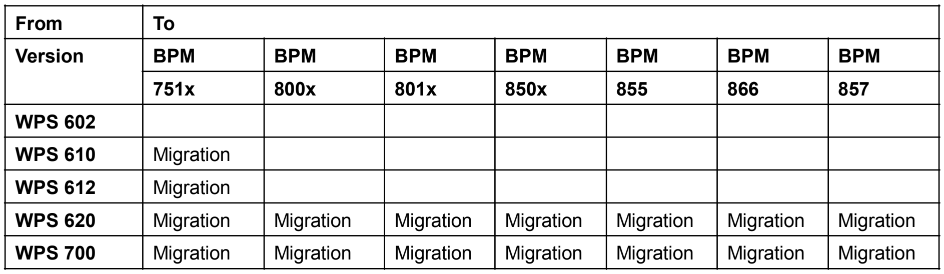
  Instance migration is migrating your process instances from one snapshot to another snapshot without IBM BPM version changes. For more information about instance migration, see 2.7, “Instance migration and application deployment” on page 25.

3.1.2 Migration and upgrade paths overview

In this section, the available migration and upgrade paths are reviewed. In each of the tables, the blank spaces indicate that there is no support.

WebSphere Process Server migration paths

Figure 3-1 shows the WebSphere Process Server for Mulitplatforms migration paths that are supported by each of the IBM BPM versions.

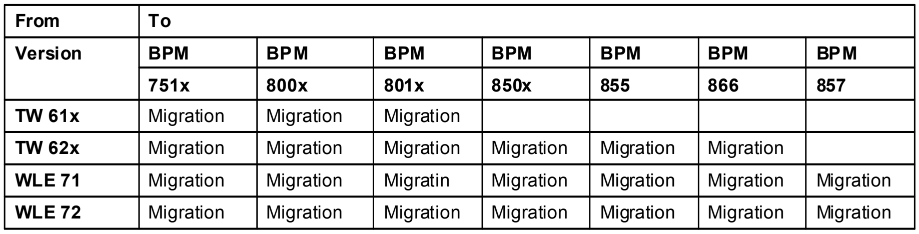


From	**To**						
Version	**BPM**	**BPM**	**BPM**	**BPM**	**BPM**	**BPM**	**BPM**
	751x	**800x**	**801x**	**850x**	**855**	**866**	**857**
WPS 602							
WPS 610	Migration						
WPS 612	Migration						
WPS 620	Migration	Migration	Migration	Migration	Migration	Migration	Migration
WPS 700	Migration	Migration	Migration	Migration	Migration	Migration	Migration

Figure 3-1 WebSphere Process Server for Mulitplatforms migration paths

WebSphere Lombardi Edition migration paths

Figure 3-2 shows the WebSphere Lombardi Edition (WLE) migration paths that are supported by each of the IBM BPM versions.

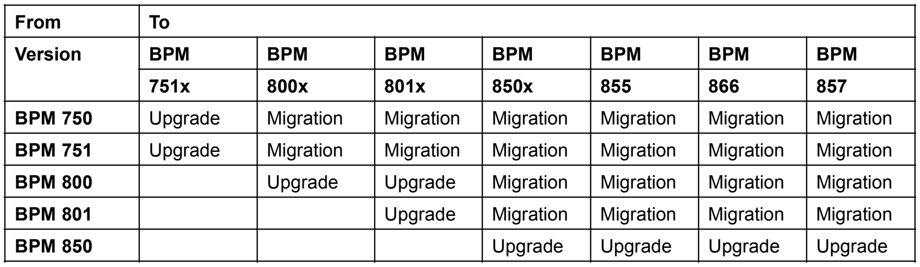


From	**To**						
Version	**BPM**	**BPM**	**BPM**	**BPM**	**BPM**	**BPM**	**BPM**
	751x	**800x**	**801x**	**850x**	**855**	**866**	**857**
TW 61x	Migration	Migration	Migration				
TW 62x	Migration	Migration	Migration	Migration	Migration	Migration	
WLE 71	Migration	Migration	Migratin	Migration	Migration	Migration	Migration
WLE 72	Migration	Migration	Migration	Migration	Migration	Migration	Migration

Figure 3-2 WLE migration paths

IBM BPM upgrade and migration path

Figure 3-3 shows the IBM BPM upgrade and migration paths that are supported by each of the IBM BPM versions.



From	**To**						
Version	**BPM**	**BPM**	**BPM**	**BPM**	**BPM**	**BPM**	**BPM**
	751x	**800x**	**801x**	**850x**	**855**	**866**	**857**
BPM 750	Upgrade	Migration	Migration	Migration	Migration	Migration	Migration
BPM 751	Upgrade	Migration	Migration	Migration	Migration	Migration	Migration
BPM 800		Upgrade	Upgrade	Migration	Migration	Migration	Migration
BPM 801			Upgrade	Migration	Migration	Migration	Migration
BPM 850				Upgrade	Upgrade	Upgrade	Upgrade

Figure 3-3 IBM BPM upgrade and migration paths

3.1.3 Migration approaches

There are different migration approaches that can meet different migration expectations.

Migrating artifacts only

To migrate artifacts, create a parallel target production environment that is configured from scratch differently from the source production environment. After migrating the artifacts, you can modify your applications to use the capabilities that the new version of IBM BPM contains. You can then test and deploy the applications to the parallel target production environment. When the applications are deployed to the target production environment, the applications do not have access to the application data that is stored in the source databases.

Use this approach in the following cases:

- You must keep two systems running in parallel until all processes are completed.
- You are changing database vendors or changing the types of IBM BPM capability (such as from express to standard or from standard to advanced).
- Your business cannot tolerate a production environment downtime window to perform the migration.

The drain approach is an extension of migrating artifacts only. It keeps the source and target IBM BPM server in the production. This approach *drains* instances in the source and starts the new instances in the target. Drain is a powerful approach to solve the zero downtime requirement and is a minimal switch over risk. Because it does not include a switch over by your configuration, the new instances can still be started at the source IBM BPM server if you find anything wrong at the target.

The drain approach includes the following limitations:

- Process histories are not transferred to the new system.
- Parallel production environments require more maintenance and resources.
- Extra effort is needed to federate the portals of two systems or the user must tolerate the two portals.

Process Federation Server (PFS) can be used to federate the task list from different IBM BPM systems with the enablement of responsive portal. Responsive portal is new in IBM BPM v857, and makes work simpler. You can use the responsive portal to get the federated task list from the PFS and then work on these tasks at the same entry point.

Migrating business data and applications

To migrate business data and applications, you export the configuration information from your current environment (the source), modify it for your new environment, and transfer the modified configuration information into your new environment (the target). Then, you point to your databases from your new environment and update the databases.

Alternatively, you can clone your databases and use the cloned databases to test the migration before you migrate your production environment. When all applications are working in the new system, you convert the databases to make them compatible with the new system.

Use the migrating business data and applications approach in the following cases:

- You have long-running process or human task instances that started in the source environment and must complete in the target environment.
- You have product data in queues that were created in the source environment and this data must survive the migration and be managed in the target production environment.

- You want to move your applications to the target version without a dependency on the development tools and the development environment.
- Your business can tolerate a production environment downtime window to perform the migration.

An 18-month entitlement: IBM BPM recognizes that the SWG standard of 90-day dual entitlement might not suffice for large IBM BPM deployments that are performing IBM BPM version-to-version system migrations. If these migrations include long-running processes, you can acquire a pre-approved, 18-month extended migration agreement. For more information about this type of migration, see this website:

https://ibm.biz/BdsmkK

Application-by-application migration

You can migrate the runtime data one application at a time. The database method to use with this process is the side-by-side migration; therefore, the source and target IBM BPM environment can be available at the same time. The application-by-application migration is a service asset, and users should contact the IBM service team before starting this process.

Use this migration approach in the following cases:

- To avoid the risks of migrating all applications to a new version in a single batch.
- To reduce the maintenance window that is planned for the migration.
- To change the IBM BPM database from one vender to IBM DB2® distributed.
- To migrate only the selected data to the target (which avoids a data purge).

Migration decision maker tree

The migration decision tree for reference in selecting the migration approach that is most appropriate for your organizational needs is shown in Figure 3-4.

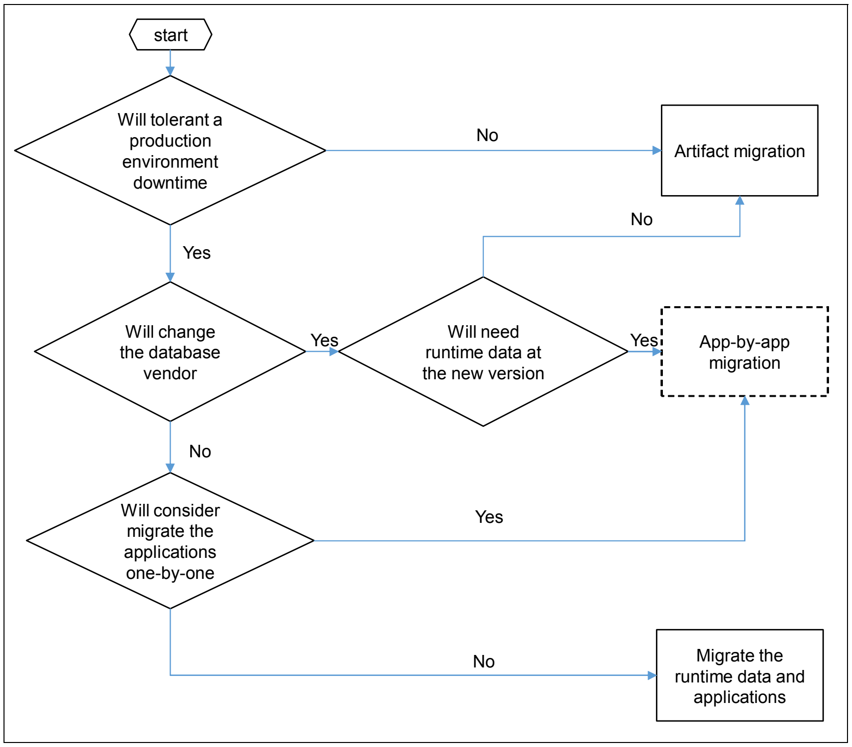

start
Will tolerant a production environment downtime
No
Artifact migration
Yes
No
Will change the database vendor
Yes
Will need runtime data at the new version
Yes
App-by-app migration
No
Will consider migrate the applications one-by-one
Yes
No
Migrate the runtime data and applications

Figure 3-4 MIgration decision tree

Migrating runtime data and applications: To migrate the runtime data and applications, a 48-hour maintenance window often is planned. The migration procedure does not mean running only the migration steps. The following time considerations and procedures must be considered:

- Backing up the IBM BPM environment
- Running the regression testing
- Engaging a fallback plan if any unaccepted regression or failure occurs

If your maintenance window is compressed, consider the artifact migration method. In this method, the source and target IBM BPM environment run in parallel. Then, you drain out the instances at the source before you move to the new version of IBM BPM.

If you do not want a single process application that is run in two IBM BPM environments, you can consider the application-by-application migration method. Although this method still requires a maintenance window, the time is shorter when compared with migrating runtime data and applications.

Application-by-application method: The application-by-application migration method includes the following limitations:

- Only the migration of runtime data of Business Process Model and Notation (BPMN) processes and basic Case Management cases is supported. Business Process Execution Language (BPEL) applications cannot be migrated by using this tool.
- Data in an external IBM FileNet® Content Manager is not migrated.
- Does not support migration of the data at IBM DB2 for z/OS®.
- The source database host machine and the target database host machine must use the same time zone and system time. The IBM BPM source host machine and target host machine also must use the same time zone and system time.
- Contact the IBM service team to determine the support source version and the target version that you need. Updates are ongoing and at the time of this writing, the following paths are supported:
  - IBM BPM V7.5.1.x to IBM BPM V8.5.6.0
  - IBM BPM V8.0.1.x to IBM BPM V8.5.6.0
  - IBM BPM V8.5.6.0 to IBM BPM V8.5.6.0
- This feature is an IBM service asset and support is provided as-is.

3.2 Readiness and planning for migration or upgrade

This section describes migrating runtime data and applications. The methodology that is described is similar for the application-by-application migration or upgrade process. How to plan a successful migration or upgrade project and reduce the risks that are involved also is described in this section.

3.2.1 Verifying your target environment

Verify that your target environment, including hardware, operating systems, and database prerequisites, is a supported operating environment for IBM Business Process Manager v8.5.7.

For more information, see these websites:

- IBM Business Process Manager Advanced system requirements:

  http://www.ibm.com/support/docview.wss?uid=swg27023005
- IBM Business Process Manager Standard system requirements:

  http://www.ibm.com/support/docview.wss?uid=swg27023007

3.2.2 Migration self-evaluation

When you start your migration, you must plan the migration as a project. Migrations are far more complex than simply completing the steps at the IBM Knowledge Center. Planning the migration requires months of work and preparation.

Perform a self-evaluation when you start planning the migration of your IBM BPM environments. This process includes consulting with your business sponsor to understand their requirements and expectations about the migration. The following self-evaluation questions can be used:

- How long of a maintenance window can the business tolerate?
- Are all of the process application sponsors ready for the migration?
- How many process instances in the database must move to the new version?
- How much monitor data (Performance Data Warehouse) in the database needs to move to the new version?
- What is the expected migration finish date, and does this date consider end of service times for IBM BPM versions?

Migrating the artifacts first

Before you attempt to migrate the runtime data, it is important to test if your process applications meet your expectations for running on the new version. If the migration motivations are based on fixes or new features that are provided by the new version, consider verifying that the new version resolved those issues. Consider the following aspects when the applications are tested:

- Compatibility

  Although IBM BPM works to maintain compatibility with an earlier version as much as possible, the compatibility can be affected by the new features that are introduced or a bug fix. Test and verify all the important aspects of the application. Another best practice is to involve the user and acquire feedback about the behavior of the application that uses the new IBM BPM version.
- Performance

  The new IBM BPM version features performance improvements in a major release. However, it is possible that some points of view or specific usage patterns might experience an unexpected performance downgrade when you test the applications. Most of the performance issues can be solved by tuning the performance and other issues might require the IBM support team to be involved.
- User experience

  Version-to-version, the user experience might be changed, especially regarding the GUI. Be prepared to reeducate your users and familiarize them with the new GUI design before starting to use the new version of the IBM BPM.

- Deprecated feature and APIs

  For each of the versions, you can find the deprecated feature or APIs in the IBM Knowledge Center. For more information about IBM BPM v857, see the following websites:

  – Deprecated and removed features:

    `https://ibm.biz/BdsMnV`

  – New features:

    `https://ibm.biz/BdsMnA`

> **Multiple versions:** If your source and target version crosses multiple versions, see the following IBM Knowledge center website:
>
> `https://www.ibm.com/support/knowledgecenter/SSFPJS`
>
> Select your BPM version, then select **What's new in IBM Business Process Manager**. At the bottom of the page, you can refer to the Deprecated and removed features of IBM Business Process Manager.
>
> **Database upgrade:** When you upgrade your database to a higher level, you *must* upgrade your JDBC driver to the matched level.

From version-to-version, the IBM BPM might have some minor behavior changes. For more information about how IBM BPM v857 behavior changes when compared with v856, see the following websites:

– Process Portal:

  `https://ibm.biz/BdsMnL`

– ECM:

  `https://ibm.biz/BdsMnC`

Dry-run migration

As a best practice, dry run the migration before it is used in a production environment. Consider the following points:

- Practice the migration steps and troubleshooting.
- Write your migration cookbook.
- Estimate the maintenance window if you use the production clone as the dry-run environment.
- Create post-migration validation processes for the real data.

The clone is based on the UAT and production environments. If you plan only one environment for the dry run, the production environment is recommended. With a dry run of your production migration, issues can be detected that exist only within a production environment (such as data integrity, special configurations, and corrupted files).

Complete the following steps to clone the IBM BPM environment:

1. Stop the IBM BPM environment.
2. Replicate configuration data by cloning the WebSphere Application Server installation and configuration.

Strategies that accomplish this task by using file-by-file copies of the configuration data (from the source to the target) are called *cloned cell topologies*. The cell in the cloned IBM BPM environment is a copy of the cell that is in the original environment. The cell name, node name, and server names also are the same. The universally unique identifiers (UUIDs) that the WebSphere Application Server and IBM BPM code use to describe all the components of the cell also are identical.

If your transaction log is stored in the file system, it must be copied as a precaution. However, the first step is to stop the IBM BPM environment so that transactions in the IBM BPM system do not hang.

3. Clone the database.

   The DB administrator can clone the database. Different database types might require different steps for cloning. After this step is complete, the administrator created another IBM BPM database instance. Be aware that the new instance is in another host and might be with another port.

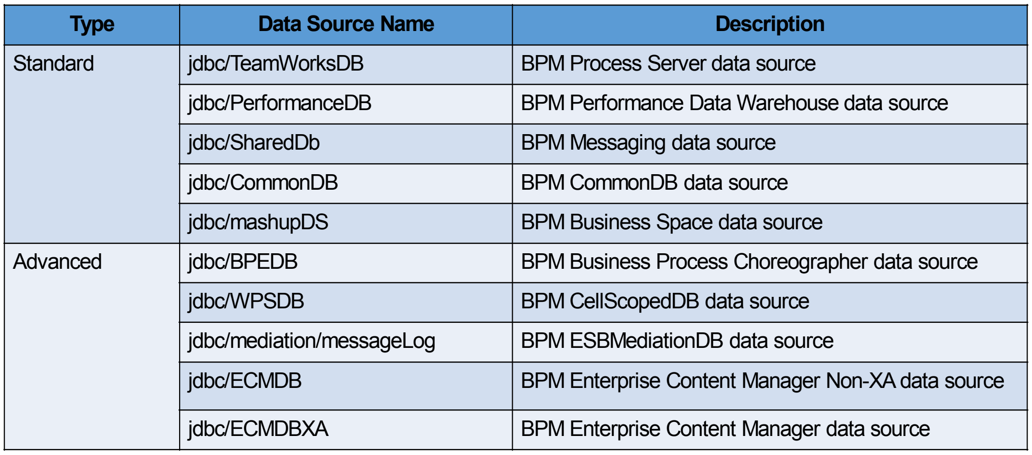
4. Update the IBM BPM data sources in the cloned IBM BPM environment.

   Starting with IBM BPM v856, you can use the **`BPMConfig –upgrade`** command to update the data sources that are used by the IBM BPM system.

   For more information about this process, see the following IBM Knowledge Center website:

   https://ibm.biz/BdsMn6

   If your IBM BPM version is lower than v856, you must update the data sources by using the WebSphere Application Server administration console. The data sources that are shown in Figure 3-5 must be considered.



Type	Data Source Name	Description
Standard	jdbc/TeamWorksDB	BPM Process Server data source
	jdbc/PerformanceDB	BPM Performance Data Warehouse data source
	jdbc/SharedDb	BPM Messaging data source
	jdbc/CommonDB	BPM CommonDB data source
	jdbc/mashupDS	BPM Business Space data source
Advanced	jdbc/BPEDB	BPM Business Process Choreographer data source
	jdbc/WPSDB	BPM CellScopedDB data source
	jdbc/mediation/messageLog	BPM ESBMediationDB data source
	jdbc/ECMDB	BPM Enterprise Content Manager Non-XA data source
	jdbc/ECMDBXA	BPM Enterprise Content Manager data source

Figure 3-5   Data sources of IBM BPM

5. Change the host name and IP address mapping at the hosts.

   If you have a DNS server, the host name in the IBM BPM configuration must point to the original IP address. You must map the host name in the cloned environment to its local IP address by editing the host file.

6. Stop the integration before you start the cloned IBM BPM environment.

   If you integrate with any systems, the cloned IBM BPM server might insert or update data within these systems. In this situation, unexpected results might occur within the application system. To avoid this possibility, ensure that the integration stopped before the cloned IBM BPM environment is started.

Estimate the maintenance window

Estimating the maintenance window is key to planning your migration and affects which migration approach you select. For example, options are available if your business tolerates only a 12-hour maintenance window and the estimated maintenance window exceeds 24 hours. Instead of migrating the runtime data and applications, consider other solutions, such as artifact migration only or application-by-application migration.

The maintenance window includes the following steps:

1. Stopping the IBM BPM environment.
2. Backing up the IBM Installation Manager repository.
3. Backing up the database and the IBM BPM configuration.
4. Running the migration procedure.
5. Testing regression.
6. Rolling back the changes if you experience a failure and cannot resolve it in your remaining time.

The data volume in the IBM BPM database impacts the database backup time and the database upgrade time. If you can purge the data before migration, it can reduce the maintenance window significantly. Refer to Chapter 4, “Purging and archiving in IBM BPM systems” on page 51, for more detailed information.

3.2.3 Quiescing IBM BPM system

Quiescing the IBM BPM system ensures that you can receive a stable status before you migrate or upgrade the data. Consider the following points regarding quiescing the system:

- Do not install any other new process applications to the Process Server (PS).
- Turn off all incoming client requests:
  - Stop user operations.
  - If a front-end HTTP server is used, stop the flow of incoming HTTP traffic to IBM BPM.
  - Stop the portal.
  - Schedule an Event Manager blackout period.
  - For JMS, IBM MQ, IBM MQ/JMS, and web services with SOAP/JMS inbound requests, refer to your operational procedures for more information about how to stop the applications that generate the request messages into queues.
- Allow messages get to a steady state:
  - Wait until the `eventqueue` is empty and check that `eventerrorqueue` is empty (reprocess and delete any errors).
  - Wait for all the following queues to finish processing (which might take a few minutes):
    - DataDefLoaderQueue
    - PostLoadCalculationQueue
    - RepresentationManagerQueue (in WLE 7.x)
  - Check that the errors queues are empty and reprocess or delete those error messages.

  - For IBM BPM Advanced with JMS, IBM MQ binding, you can pause the processing of IBM MQ binding in the JMS messages. For more information about how to pause the messaging, see step 2a at the following website:

    http://www.ibm.com/support/docview.wss?uid=swg21610342

- Stop the Event Manager.
- Stop the Application cluster.
- Stop Messaging and Support clusters.
- Stop Deployment Manager and nodes.

3.3 Upgrading IBM BPM

Although upgrading is simpler than migrating the IBM BPM, upgrading still requires planning. For more information about migration, see “Migrating the artifacts first” on page 41, “Dry-run migration” on page 42, and “Estimate the maintenance window” on page 44.

From version 850 forward, IBM BPM followed an *update only* constraint on every new 8.5 release. Therefore, instead of migrating, you can upgrade your system after that release.

Compared to migration, the upgrade does not move artifacts side by side. Upgrading uses the following actions:

- In-place upgrade of the IBM BPM binaries.
- In-place upgrade of the database schema and transforms the data, if necessary.
- The profile is upgraded when the Dmgr starts.

In most cases, the upgrade procedure can be finished in 1 hour. However, regression testing time still needs to be factored into the maintenance window.

3.3.1 Rolling upgrade

You can roll out maintenance incrementally in an IBM BPM installation that consists of a Process Center and multiple Process Servers. This roll out allows for continuous running of production applications during the upgrade and regression test periods.

A rolling upgrade can be performed only when applying fix packs, refresh packs, or interim fixes. It cannot be used for migration between major releases.

Consider the following points regarding rolling upgrades:

- You can use rolling upgrade methodologies to upgrade your IBM BPM systems:
  - In-place Rolling Upgrade
  - Side-by-side Rolling Upgrade
- Do not upgrade the Development environment of your Process Center until you prove that the upgrade is working in your QA environment.
- The version of Process Designer and Integration Designer must match the version of the Process Center.
- The version of Process Center and Process Server must match for the Process Server to be connected as *online*.
- An application that is developed by using an older version of the IBM BPM product can be deployed to run on a newer version of IBM BPM if it is not using features that were removed.

- IBM BPM strives to maintain API compatibility in IBM BPM mod and fix pack releases.
- In IBM BPM 85xx, IBM BPM enabled the support of online rolling upgrades to facilitate the debugging of upgraded applications and allow a gradual roll out of the upgrades:
  - IBM BPM 8500 Process Designer and Integration Designer can connect to IBM BPM 8501 of Process Center and Process Server (for debugging).
  - A IBM BPM 8501 Process Server can connect in *online* mode to IBM BPM 8500 Process Center.

In-place rolling upgrade

In this procedure, you upgrade the Process Server (PS) first. Start with the Test PS, then the Staging PS. Next, the Production PS follows. Finally, you upgrade the Process Center Server (PC), as shown in Figure 3-6.

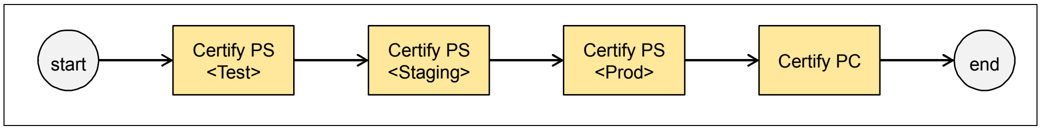


Figure 3-6 Process of in-place rolling upgrade

This process includes following advantages and disadvantages:

- Advantages:
  - Upgrade and certify one Process Server environment at a time in an orderly fashion.
  - Upgrade the PC last as you want to ensure you can continue to provide application fixes to the PS running at the current version.
- Disadvantages:
  - Upgrading the PC last means developers cannot use new features until the entire system is upgraded.
  - Except for 8501, any other upgrade forces the PS to go offline to the previous PC, which requires offline deployment until the PC is upgraded.

Note: Always take a full backup of your environment before the upgrade. The backup ensures the ability to restore any environment during this period in case an upgrade failed and cannot be fixed in the maintenance window.

Side-by-side rolling upgrade

In this procedure, you set up a new *upgraded* PC side by side and then proceed to upgrade each of the PS environments, as shown in Figure 3-7.

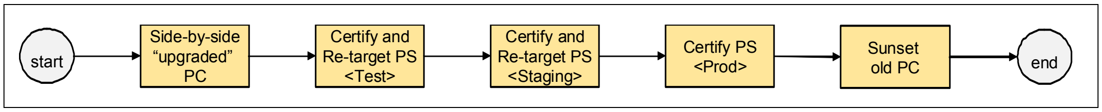

start
Side-by-side "upgraded" PC
Certify and Re-target PS <Test>
Certify and Re-target PS <Staging>
Certify PS <Prod>
Sunset old PC
end

Figure 3-7 Procedure of side by side rolling upgrade

This process includes following advantages and disadvantages:

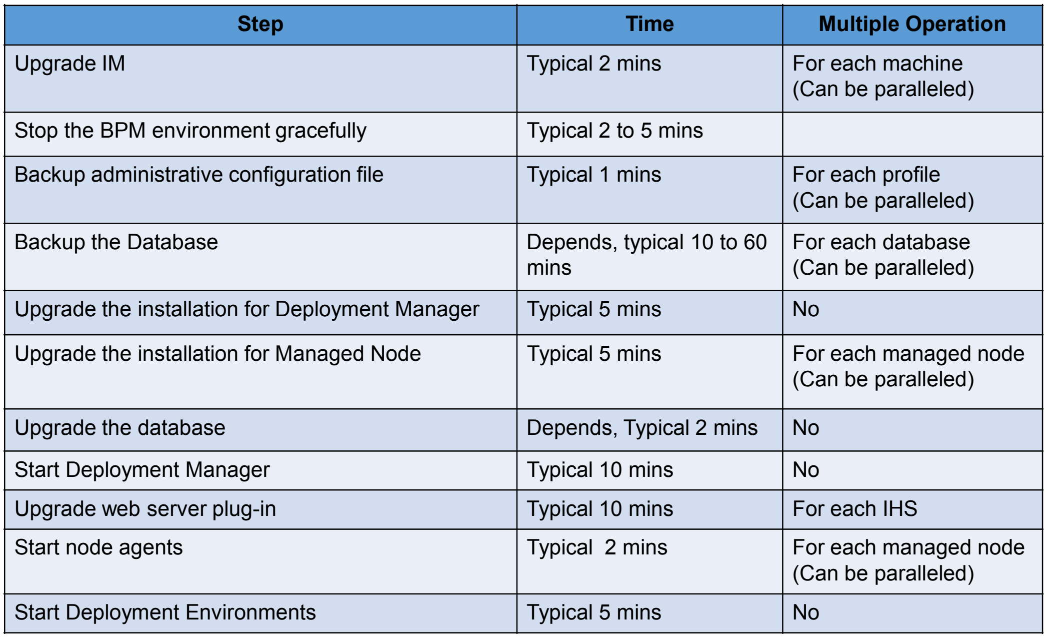
- Advantages:
  - An upgraded PC is available immediately for new development and tests.
  - Customers can certify new product versions on a per-application basis without affecting development and test environments.

- All Process Server environments can maintain online status all of the time.

- Disadvantages:
  - The set-up for a side-by-side upgraded PC requires more IT resources and might incur extra licensing costs.
  - Might require more work if the customers' configuration is not well-documented.
  - Depending on the approach that is used to set up the side-by-side PC, *playback* instances data can be lost.

3.3.2 Estimating the upgrade time

Before planning the upgrade, use Figure 3-8 to estimate the maintenance window. The times that are in the table are based on experimental data.



Step	Time	Multiple Operation
Upgrade IM	Typical 2 mins	For each machine (Can be paralleled)
Stop the BPM environment gracefully	Typical 2 to 5 mins	
Backup administrative configuration file	Typical 1 mins	For each profile (Can be paralleled)
Backup the Database	Depends, typical 10 to 60 mins	For each database (Can be paralleled)
Upgrade the installation for Deployment Manager	Typical 5 mins	No
Upgrade the installation for Managed Node	Typical 5 mins	For each managed node (Can be paralleled)
Upgrade the database	Depends, Typical 2 mins	No
Start Deployment Manager	Typical 10 mins	No
Upgrade web server plug-in	Typical 10 mins	For each IHS
Start node agents	Typical 2 mins	For each managed node (Can be paralleled)
Start Deployment Environments	Typical 5 mins	No

Figure 3-8 Estimate the upgrade time

3.4 Topology refactoring

In this section, the following definitions within topology refactoring are described:

- Number of clusters, such as changing a four-cluster topology to a three-cluster topology
- Number of the nodes in IBM BPM cluster, such as horizontal extension – add node
- Managed node location, such as moving the node from Windows to Linux
- IBM BPM database type, such as changing the database type from SQL server to DB2
- IBM BPM database location, such as moving the database from Windows to UNIX
- IBM BPM edition, such as shifting from Standard to Advanced edition
- Cell location, such as moving the whole cell to different machines

3.4.1 Changing the number of clusters

Each of the IBM BPM versions has a recommended topology. For example, in IBM BPM v857 three clusters is the recommended topology. If you are operating on an older version, you can notice immediate benefits in the use of the recommended topology in your target version. The migration can include the topology change.

3.4.2 Adding a node in a cluster

Notes can be added to increase the performance of the IBM BPM. For example, if only one or two nodes are in the cluster that uses most of the CPU and memory resources in the machine, consider horizontally extending to new hardware by adding nodes.

Complete the following steps to add nodes in a cluster:

1. Add new managed nodes.
   a. Create custom nodes on the new machine by using the Profile Management Tool (PMT)

      Note: Select the **federated** option later in the creation wizard.

   b. Federate the node in the deployment manager. For more information, see the following website:

      https://ibm.biz/BdsMe8
2. Add managed nodes to the cluster:
   a. At the WebSphere Application Server admin console, click **Deployment Environment** → **DE_NAME** → **deployment topology panel**.
   b. Choose the new node in the drop-down node list, and click **Add**.
   c. Edit the number of Application deployment targets (if you have single cluster). The number indicates how many cluster members you must create on the new node.
   d. Click **Save**.

3.4.3 Managing node location

In some instances, you might need to move the IBM BPM node to another powerful machine. This move can be needed because of the limitations of CPU, memory, I/O of the current machine, or changing operating systems (such as from Windows to Linux).

To move the IBM BPM node, follow the steps that are described in 3.4.2, “Adding a node in a cluster” on page 48. However, you want to *remove* (instead of add) the node by using the Deployment Environment.

3.4.4 Changing the IBM BPM database type

If you want to change your database type, use the artifact migration or the application-by-application migration for this process.

The artifact migration can change from or to any database type. However, it cannot maintain your processes at the target database. Therefore, you cannot access the process instances and monitor data in the target IBM BPM system. All runtime data is left in the old system and drains out. For the development PC, migrate individual application artifacts by exporting *active* snapshots (in chronological order) from the source environment to the target environment.

For the applications that must keep the runtime data in the target IBM BPM system, use the application-by-application migration method. This method changes the database type to the DB2 distributed database only. It does *not* support change to other database providers. This process has other limitations. For more information, see section “Application-by-application migration” on page 39.

3.4.5 Changing the IBM BPM database location

You might need to change your IBM BPM database location; for example, when moving to a powerful machine or to another operation system. Contact your DBA or database provider to move the database to another machine.

After the database location is changed, you can reset the data sources for the IBM BPM by using Step 4 that is described in “Update the BPM data sources in the cloned BPM environment” in section “Dry-run migration” on page 42.

3.4.6 Changing IBM BPM edition

IBM BPM support changed the IBM BPM edition from Standard to Advanced. There are no upgrade paths to or from the Express edition or the Advanced - Process Server edition. If you have the express edition and must change to the Standard or Advanced edition, you must use artifact migration. Artifact migration cannot migrate your runtime data.

To upgrade from the Standard edition to the Advanced edition, use the following documents that are available in the IBM Knowledge Center:

- *Upgrading a product installation from IBM BPM Standard to IBM BPM Advanced*:

  https://ibm.biz/BdsMeW

- *Upgrading a deployment environment from IBM BPM Standard to IBM BPM Advanced*:

  https://ibm.biz/BdsMbB

3.4.7 Changing cell location

If one IBM BPM cell must be moved to different hardware, see *Migrating IBM BPM to the same version on new hardware,* which is available at the following website:

https://ibm.biz/BdsMbR

Note: If you are using version v850x, upgrade to the later version first to receive the support that is described in *Migrating IBM BPM to the same version on new hardware*.

However, support for changing the cluster topology is not supported. Single cluster topology cannot add clusters.

4

Purging and archiving in IBM BPM systems

This chapter describes purging and archiving an organization's IBM Business Process Manager (BPM) data. IBM BPM is a stateful product that accumulates data over time. As with any stateful product, it is essential to its ongoing health to have a strategy for purging some of that data occasionally.

In addition to describing the process of purging data from IBM BPM, this chapter also provides information about the Process Center, Process Portal, and Process Federation Server indexers, how to configure the index intervals, and how to manually run the indexers.

> **Important:** Running deletion and cleanup commands for snapshots and instance data are done during the time of least activity. It is important to create a script that contains the logic on your data retention policies and run these scripts regularly.
>
> Problems were reported in the past with the use of the `BPMDeleteSnapshot` and `BPMSnapshotCleanup` commands in IBM BPM. Therefore, check the following Flash Alert to ensure that you have all of the correct fixes for the version of IBM BPM that is being used:
>
> http://www.ibm.com/support/docview.wss?uid=swg21669992

This chapter includes the following topics:

4.1 Data retention policies

If data grows without bounds, it can lead to disk space and performance issues as database queries take ever longer.

Business processes naturally use business data in the form of variables to represent the process state and affect the process flow. However, do not consider this data a system of record (SOR). IBM BPM should always work with external systems of record to access and update data. Use the business data in a process only for that process; do not use that data as an SOR for other purposes.

An effective SOR is designed for transactional, efficient, and safe reading and writing and reporting by one or more users and producers. The business data in a process is merely a temporary stateful representation of that data to affect the process flow and user interactions. Process variables and their underlying IBM BPM persistent data store are not designed to be used outside of that process. Considering these process variables as the single source of truth for that data eventually causes issues and requests for typical transactional create, read, update, and delete access to that data that is not possible or recommended on top of the IBM BPM internal database.

A data retention policy must define policies for purging old snapshots and process instance data.

4.1.1 Snapshot and instance data deletion considerations

Deleting snapshots on the Process Center or Process Server have a different set of preconditions that must be in place for any snapshot that you want to delete. Consider the following points:

- Process Center:
  - You must be a repository administrator.
  - The first snapshot of an application cannot be deleted, even though it might be unnamed or archived. The first snapshot contains original information about the snapshot that is displayed in the history panel in Process Designer.
  - You must archive named snapshots before you delete them.
  - The snapshot must have no running instances.
- Process Server:
  - The snapshot must exist.
  - The snapshot must be inactive.
  - The snapshot must have no running instances.
  - The snapshot must not be deployed.

Note: Process instances cannot run on the snapshot that you want to delete. When a new snapshot is deployed, it is important to always migrate runtime process instances whenever possible.

4.2 Process Center

The Process Center holds projects, which are deployable process applications or reusable toolkits.

4.2.1 Disabling auto-tracking in a BPD

Auto-tracking in IBM BPM is important for many business process definitions (BPDs) because it helps gather, track, and report key business metrics. However, another cost comes with auto-tracking because these events are processed by the Performance Data Warehouse and persisted in the database.

Because of that cost, disable the default auto-tracking capability for BPDs if you do not need to track and report on their business metrics so that you can lower your costs. Also, consider creating tracking groups to track only key business events and then disable auto-tracking. This approach ensures that the events that are persisted are only those events that are required for your business metrics.

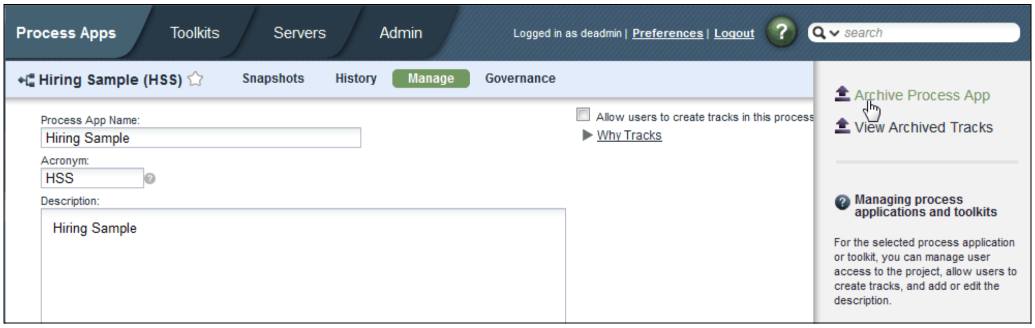
4.2.2 Marking system tasks for deletion when created

For system tasks in your BPDs, select the **Delete task on completion** option in the Implementation tab of the properties in IBM Process Designer. With this option selected, tasks are automatically deleted when they complete, which can save a significant amount of data from being persisted.

For user tasks in your BPDs, make a conscious decision about whether to select the Clean State option in the Implementation tab of the BPD’s properties. By default, this option is not selected. However, selecting this option automatically cleans up the context (such as variables) for the user task when it completes. If the state is not required to be persisted and this option is selected, you can save a significant about of database space and even speed the migration of snapshot instances.

4.2.3 Process applications and toolkits

Process applications and toolkits can be archived from the Manage tab of the Process Center for a project, as shown in Figure 4-1.



Process App Name: Hiring Sample

Acronym: HSS

Description: Hiring Sample

□ Allow users to create tracks in this process

Managing process applications and toolkits

For the selected process application or toolkit, you can manage user access to the project, allow users to create tracks, and add or edit the description.

Figure 4-1 Archiving process applications and toolkits

Archiving a project does not delete it or reclaim its space in the database. Instead, archiving marks the project so it does not show by default in the Process Center user interface. To delete a project, select the option to show the archived projects and then delete the project (the project must be archived first), as shown in Figure 4-2.

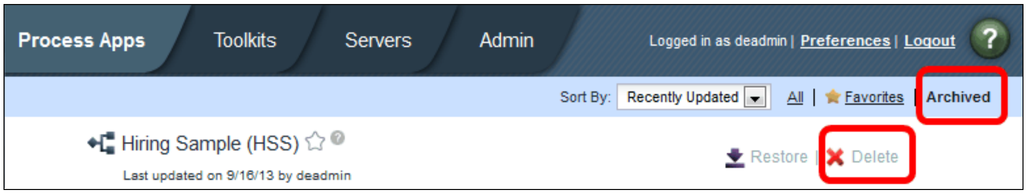

Process Apps
Toolkits
Servers
Admin
Logged in as deadmin | Preferences | Logout
Sort By: Recently Updated
All | Favorites | Archived
Hiring Sample (HSS)
Last updated on 9/16/13 by deadmin
Restore
Delete

Figure 4-2 Deleting archived process applications and toolkits

Deleting a project deletes all snapshots and process instances. It also deletes all IBM BPM Advanced content, such as associated Business Process Execution Language (BPEL) process instances, business-level applications, and enterprise applications. This deletion capability in IBM BPM is available only from the user interface. There is no scripted way to perform this deletion.

4.2.4 Snapshots in Process Center

> **Important**: Do not delete the snapshots until all of the required fixes are applied as directed in the following Flash Alert:
>
> `http://www.ibm.com/support/docview.wss?uid=swg21669992`

When you edit a process application or toolkit by using Process Designer, you are changing a special version or snapshot of it that is named `Current` (or `Tip`). At any point, you can take a new snapshot and assign that snapshot a name. Named snapshots are deployable to Process Server, but other Current versions are not editable.

Whenever you save artifacts in Process Designer, an unnamed snapshot is created in the database. This feature is helpful to enable you to see history, but it is costly because of database growth.

You can archive individually named snapshots instead of archiving the entire project and all of its snapshots. You can perform this archiving process from the Snapshots page of the process application or toolkit by using the drop-down menu for each snapshot. This archiving does not delete the snapshot; instead, the snapshot is marked and hidden.

So, how is a snapshot deleted in Process Center? IBM BPM V8.5.0 introduced the **`BPMSnapshotCleanup`** wsadmin command, with which you can delete b named and unnamed snapshots. The named snapshots first must be archived to delete them.

For more information, see the following IBM Knowledge Center website:

`https://ibm.biz/BdsMVC`

> **Important:** The **`BPMSnapshotCleanup`** wsadmin command also was backported to the v7.5.1.2 Fix Pack, v8.0.1.3 Fix Pack, v8.5.1.2 Fix Pack, and v8.5.6.

4.2.5 Unnamed snapshots in Process Center

The **BPMSnapshotCleanup** wsadmin command can be used to delete unnamed snapshots.

IBM BPM includes the capability to enable automatic clean-up of unnamed snapshots. You can enable this feature by adding the lines that are shown in Example 4-1 to your `100custom.xml` file.

Example 4-1 Automatic Cleanup for Unnamed Snapshots in the Process Center

```
<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 more information, see *Deleting unnamed snapshots, automated from a Process Center server, automated method*, at the following IBM Knowledge Center website:

https://ibm.biz/BdsMVw

Note: Exporting a snapshot to a `.twx` file and reimporting it into a different Process Center is another method that can be used to delete unnamed snapshots. Unnamed snapshots are not exported.

4.2.6 Advanced content in Process Center

Important: Do not delete the snapshots until all of the required fixes are applied, as directed in the following Flash Alert:

http://www.ibm.com/support/docview.wss?uid=swg21669992

If you use IBM BPM Advanced and you have process applications and toolkits that contain advanced content, such as IBM BPEL processes, you must have a strategy for deleting the business-level applications (BLA) and enterprise applications that are created in the Process Center Playback server by the existence of that content.

For every process application or toolkit in Process Center that contains a module or library (directly or inherited through a toolkit), a BLA is created for the current snapshot and for every named snapshot of toolkits and process applications that contain advanced content. The BLA is named <*Acronym*>-<*Snapshot-name*>; for example, `PA1-V1.0`.

Within those BLAs, every module or library produces an enterprise archive (EAR) file that is an asset within it. These BLAs are created on demand when performing a Playback from Process Designer or publishing to Process Center from Integration Designer. It remains until the process application or toolkit is deactivated.

As you create toolkits with advanced content, use those toolkits from process applications, and snapshot the toolkits and process applications, this advanced content can quickly accumulate, which affects server start time, memory consumption, and general performance.

You often need only the BLAs for the current snapshot of a process application. You can delete them from the current snapshot of any toolkits and any named snapshots of either toolkits or process applications. To delete the BLAs, you “undeploy” your snapshot. You see this option on the Snapshots page of Process Center, in the drop-down menu for the Current snapshot if that snapshot’s advanced content is deployed.

For named snapshots, you first use the Deactivate option. Then, you see the Undeploy option if there is deployed advanced content. You can always use the WebSphere administrative console or commands to manually remove the BLAs and EARs. Do not be concerned about deleting something that is needed because these BLAs are re-created on-demand, if needed.

If you are using Advanced Integration services (AIS) that are the bridge between Business Process Model and Notation (BPMN) and BPEL processes, we recommend the façade pattern to reduce the size of the EARs that are involved. For more information about this pattern, see the IBM developerWorks® article, *Implementing the facade pattern using IBM Business Process Manager Advanced V7.5*, which is available at this website:

https://ibm.biz/BdsMJs

4.2.7 Administering the Process Center index

The Process Center index is used to conduct searches on the Process Center repository. The index is automatically created and maintained when the server is started. After the index is created, the system updates the index at regular intervals to reflect any changes that are made to the repository.

You can configure the update interval and index location to better suit your installation. Also, commands are provided with IBM BPM for you to manually re-create or update the index.

Manually re-creating or updating the Process Center index

The Process Center index is in the file system of the machine on which the server is running. You do not need to back up the profile or the database before running the **artifactIndexFullReindex** or **artifactIndexUpdate** commands. You can run the commands when the server is running.

Re-creating the index in a network deployment environment

To re-create the index in a network deployment environment, complete the following steps:

1. From the command prompt, change to the deployment manager profile bin subdirectory where the command is located.
2. Enter the following command:

   ```
   artifactIndexFullReindex -u userId -p password -host hostName -port port
   ```

   where

   hostName is the host name of the application cluster member on the node and port is the server SOAP port.

Updating the index in a network deployment environment

To update the index in a network deployment environment, complete the following steps:

1. From the command prompt, change to the deployment manager profile bin subdirectory where the command is located.
2. Enter the following command:

   ```
   artifactIndexUpdate -u userId -p password -host hostName -port port
   ```

where

hostName is the host name of the application cluster member on the node and port is the server SOAP port.

> **Important:** One index is created for each node. To update the index, you must issue the command for each node. Run the command from the deployment manager and direct it by using the **-host** and **-port** parameter to a cluster member that is running on each node of your installation. This process updates the index for each node; there is no need to run the command locally on the node.

Configuring the Process Center index

You can configure the indexing process to automatically update the index on a preset schedule. You can also set the index location and you can disable indexing altogether. To change the Process Center index update interval or to disable indexing, create or edit the 100Custom.xml file that is in the following directory:

- For a UNIX or Linux operating system:

  ```
  <process_center_profile>/config/cells/<Cell>/nodes/<node>/servers/<server>/proc
  ess-server/config/system
  ```

- For a Windows operating system:

  ```
  <process_center_profile>\config\cells\<Cell>\nodes\<node>\servers\<server>\proc
  ess-server\config\system
  ```

The Process Center index configuration settings that can be updated in the 100Custom.xml file are listed in Table 4-1. If the process instance includes many previous tasks, reindexing these tasks might degrade the system performance.

Table 4-1 XML tags that are used to change the index configuration settings

XML tag	Description
<artifact-index-enabled>	This Boolean value determines whether indexing is enabled. The default value is true; if the index does not exist, it is created. To turn off indexing, change the value to false; if the index does not exist, it is not created.
<artifact-index-update-interval>	This integer value specifies the time between index updates in seconds. Restriction: The minimum update interval is 30 seconds. If you attempt to set a value of less than 30, the index defaults to 30 seconds.

Setting the index location

To change the Process Center index location, log in to the administrative console and select **Environment** → **WebSphere Variables**.

Setting the location for a cluster server

Edit the BPM_SEARCH_ARTIFACT_INDEX variable. For a cluster member, set the scope to the cluster level. The default is $<BPM_SEARCH_ARTIFACT_INDEX_ROOT>/<clusterName>.

Setting the location for a stand-alone server

For a stand-alone server, set the scope to the cell level. The default is `$<BPM_SEARCH_ARTIFACT_INDEX_ROOT>/$<WAS_SERVER_NAME>`. If the variable is not set, the location defaults to `$<USER_INSTALL_ROOT>/searchIndex/artifact/$<WAS_SERVER_NAME>`.

For more information, see *Administering the Process Center index* at the IBM Knowledge Center:

https://ibm.biz/BdsMJP

4.3 Process Server

The Process Server is where you install process application snapshots and where you run processes within the snapshots. You must consider how (in the Process Server in terms of accumulating content) to delete these installed snapshots, and how to delete processes when they complete or end.

4.3.1 Snapshots in Process Server

You install multiple snapshots of the same process application to a specific Process Server. Over time, these snapshots can accumulate and it becomes prudent to delete the snapshots that are no longer used.

For IBM BPM Advanced customers, it is important to remember that if a process application contains any advanced content (such as a module or library from Integration Designer), a business-level application with EARs is created for this process application.

Conceptually, the IBM BPM content is installed into a Process Server after which the IBM BPM Advanced content is deployed, which amounts to generating and installing the BLA and constituent EARs.

Important: You can remove snapshots from Process Server in IBM BPM v8.0.1 or higher only with the introduction of the **`BPMDeleteSnapshot`** wsadmin command.

For the **`BPMDeleteSnapshot`** wsadmin command to work, the following prerequisites must be met:

- The snapshot cannot have any running instances and cannot be the default snapshot. Use the **`BPMShowSnapshot`** command to determine whether either of these statuses is true.
- The snapshot cannot be active. Use the **`BPMDeactivate`** command to deactivate the snapshot. For IBM BPM Advanced processes, you must use the **`BPMStop`** command. These commands prevent new instances of BPMN and BPEL processes from starting and allow instances to quiesce.
- The IBM BPM Advanced content, including the BLA and EARs, must be undeployed by using the **`BPMUndeploy`** command.

  Note: When you successfully delete a snapshot, any BPD instances for it also are deleted.

4.3.2 Instances in Process Server

Two types of instances should be considered: User or human task instances, and process instances for BPMN BPD processes and BPEL processes. Both task and process instances are recorded in the database, even after the task and process complete. Therefore, it is important to think about occasionally purging older instances.

When you delete process instances, task instances also are deleted. Although you can delete BPEL human task instances independently of their processes, BPD user tasks cannot be deleted as such.

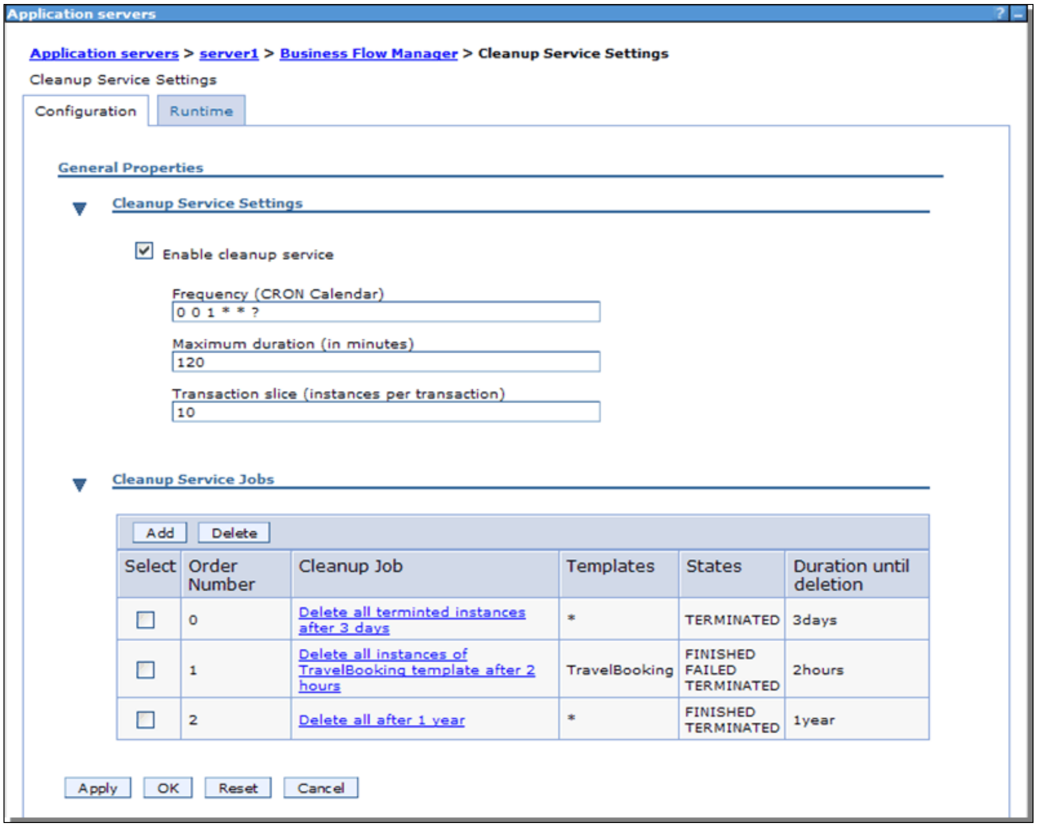
To delete BPD process instances and their associated task instances, you can use the **`BPMProcessInstancesPurge`** wsadmin command. By using this command, you can identify the specific instances to delete, or the date range within which any instances that completed is deleted.

You also identify whether to delete completed, canceled, failed or all types of instances. This command also includes other parameters to specify the maximum duration time and maximum number of instances to delete, which makes it a candidate to run in a regularly scheduled cron job so you can limit its effect on the system and limit its work.

> **Important:** The **`BPMProcessInstancesPurge`** command is not available on IBM BPM 8.5.6 and previous releases. For IBM BPM v8.0.1.x, v8.5.1.x, and v8.5.6.x, you can use the deprecated **`BPMProcessInstancesCleanup`** command to delete BPD process instances. In addition, the **`BPMProcessInstancesCleanup`** does not exist on IBM BPM v7.5.x, but you can use a supplied stored procedure **`LSW_BPD_INSTANCE_DELETE`** to delete explicitly identified process instances.

IBM BPM Advanced provides the following options for deleting Business Process Choreographer human task and process instances:

- When modeling the human tasks and BPEL processes in Integration Designer, specify to automatically delete instances when complete.
- Use the Business Process Choreographer Explorer to delete tasks or processes individually.
- Create a custom utility by using the Business Process Choreographer APIs.
- Use the supplied Jython scripts **`deleteCompletedTaskInstances.py`** or **`deleteCompletedProcessInstances.py`** to delete a batch of task or process instances by state, owning user, or completion date.
- Use the Cleanup Service in the Human Task Manager or Business Flow Manager administrative console pages to schedule jobs to automatically delete task or process instances. You can specify when to run, how long to run, how many instances to delete, and which instances to delete via state and date criteria, as shown in Figure 4-3 on page 60.


Application servers
Application servers > server1 > Business Flow Manager > Cleanup Service Settings
Configuration
Runtime
Add
Delete
Apply
OK
Reset
Cancel

Cleanup Service Settings

General Properties

Cleanup Service Settings

☑ Enable cleanup service

Frequency (CRON Calendar)
0 0 1 * * ?

Maximum duration (in minutes)
120

Transaction slice (instances per transaction)
10

Cleanup Service Jobs

Select	Order Number	Cleanup Job	Templates	States	Duration until deletion
□	0	Delete all terminted instances after 3 days	*	TERMINATED	3days
□	1	Delete all instances of TravelBooking template after 2 hours	TravelBooking	FINISHED FAILED TERMINATED	2hours
□	2	Delete all after 1 year	*	FINISHED TERMINATED	1year

Figure 4-3 Cleanup Service scheduling for BPEL process instance purging

In IBM BPM Advanced, the following items also can be purged:

- IBM BPM process and human task templates that are no longer needed
- Audit log entries
- Failed messages that are in the hold queue
- Unused shared work items

For more information about how to delete these items, see the following resources:

- IBM Knowledge Center topic “Cleanup procedures for Business Process Choreographer”:

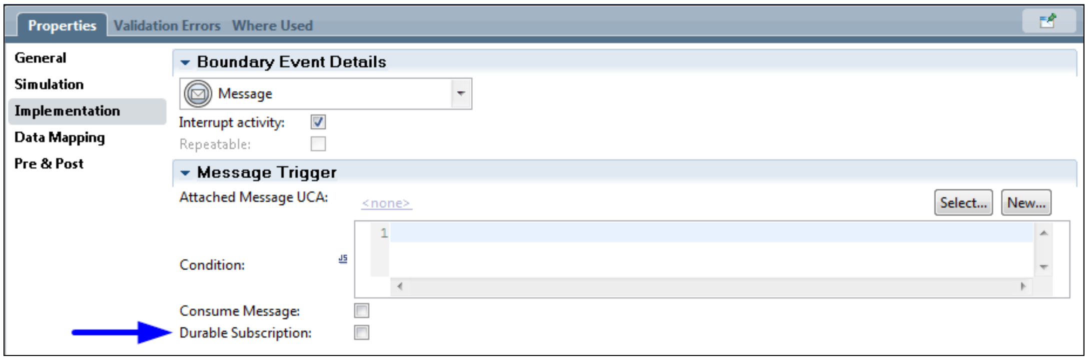
  https://ibm.biz/BdsM77

- IBM developerWorks series *Operating a WebSphere Process Server environment*:

  https://ibm.biz/BdsSds

4.3.3 Durable subscription events in Process Server

Message events in an intermediate message activity of a BPD can be made durable, which you enable by selecting the **Durable Subscription** option in the Properties tab, as shown in Figure 4-4.



Message

Interrupt activity: ☑

Repeatable: □

Attached Message UCA: <none>

Condition:

Consume Message: □

Durable Subscription: □

Figure 4-4 Specifying a durable subscription for intermediate events

If specified in Process Designer, these durable messages accumulate and require occasional clean-up, even if you select the **Consume Message** option. Although no built-in way to perform this clean-up was available in the past, a new **`BPMDeleteDurableMessages`** wsadmin command is now available that takes the following parameters:

- `olderThan`: Only events older than this number of days are deleted.
- `maximumDuration`: The command that is run for this amount of time only.
- `transactionSlide`: Indicates the number of events to delete per transaction.

These parameters are shown in the following example:

```
BPMDeleteDurableMessages {-olderThan 30 -maximumDuration 60 -transactionSlice 100}
```

4.4 Performance Data Warehouse

BPD processes can support tracking, which means events are sent to the Performance Data Warehouse and logged in its database. How many events are sent is determined by whether your BPD includes auto tracking that is enabled, which was the default for new BPDs.

With auto tracking enabled, the Performance Data Warehouse can quickly accumulate much data. Unfortunately, no product was supplied and supported in the past that deletes any of this data. As a result, many customers resorted to custom SQL code that directly accesses the database to delete or move certain data according to age or state criteria.

Alternatively, all Performance Data Warehouse data can be removed by dropping and re-creating the tables. For more information, see the following resources:

- Before V7.5.0: *Using twinit to re-create databases in Teamworks 7.0.x and WebSphere Lombardi Edition*:

  `http://www.ibm.com/support/docview.wss?uid=swg21497392`

- After V7.5.0: *How to clean up the Performance Data Warehouse database and the LSW_PERF_DATA_TRANSFER table for IBM BPM*:

  `http://www.ibm.com/support/docview.wss?uid=swg21612755`

A product is now available that supports pruning selective data from the Performance Data Warehouse. It is included with 8.5.0.1, 8.0.1.2, and 7.5.1.2 fix packs. With these releases, the `perfDWTool` features a new option that is called Prune. By using this command, data that is beyond a specified age in days can be purged, as shown in the following example:

```
perfDWTool.sw –u uid –p pwd –nodeName node prune –daysOld days
```

The use of this command deletes all data older than the days that are specified. This command must run from an active node in the cluster and all members of the cluster should be running. Attempt to run it when the server is least busy. Its operation can be affected by three new settings that you can override in `100custom.xml`, as shown in Example 4-2.

Example 4-2 Performance Data Warehouse pruning

```
prune-batch-size: The number of record to be deleted in a single prune operation.
The default value is 1000.

prune-operation-time-box: The amount of time the operation will run, in seconds.
The default is 10800 or 3 hours.

prune-operation-time-box-retry: The number of times the operation will be tried.
The default is 4 such that it will retry 3 times.
```

4.5 Process Portal index

The Process Portal index allows process participants who are working in Process Portal in non-federated environments to search business processes for instance data. The index also is used to provide data for the charts in the Process Performance and Team Performance dashboards.

> **Important:** The information in this section applies to Heritage Process Portal and Process Portal.

Indexing on an IBM BPM system is enabled by default. Process instances and tasks are indexed according to a time interval that you can specify. To change the indexing behavior, you must edit the `100Custom.xml` configuration file. If a problem occurs with the index, commands are available for updating and rebuilding it.

Tasks and process instances are indexed in the following situations:

- Tasks:
  - A task is assigned.
  - A task is completed and the business data is updated.
  - The due date or at-risk date of a task is changed.
  - The priority of a task is changed.
- Process instances:
  - An instance is started, completed, suspended, resumed, terminated, or restarted.
  - An instance failed.
  - The due date or at-risk date of an instance is changed.

For example, business data for a process instance that exists when a task and its corresponding process instance activity are completed is indexed with the task and instance. Process participants can find the task or instance by searching the instance business data. If a task form consists of several coach views but only one coach view is complete, the updates from this coach view are not searchable until all the coach views in the task form are complete.

By default, the previous tasks in a process instance are not indexed again when later tasks are completed and the business data for the process instance is updated. If you want the updated business data to be searchable from previously completed tasks, change the value of the `<task-index-update-completed-tasks>` configuration setting (see Table 4-2) to true in the `100Custom.xml file`. If the process instance includes many previous tasks, reindexing these tasks might degrade the system performance.

Table 4-2 XML tags that can be used to change the index configuration settings

XML tag	Description
<task-index-enabled>	This Boolean value determines whether indexing is enabled. The default value is true. If the index does not exist, it is created. To turn off indexing, change the value to false. If the index does not exist, it is not created. Restriction: If indexing is turned off, the search field in the Process Portal user interface is hidden. The Process list, the Process Performance dashboard, and the Team Performance dashboard also do not work correctly.
<task-index-update-interval>	This integer value specifies the time between index updates in seconds. The specified interval determines when the state of the instance variables is captured for tasks that completed since the last index update. Only those tasks that are completed during the current interval are searchable with the latest instance data. The default value for the update interval is 5 seconds. The minimum value is 1 second.
<task-index-update-completed-tasks>	This Boolean value controls whether the index is updated for previously completed tasks. The default value is false, which means that only the task that was just completed and open tasks are updated in the index. If you change the value to true, Instance-level updates, such as business data that is updated later in the process, is propagated to completed tasks. Although the system has more work to index completed tasks, searches on completed tasks are based on the final instance business data. Users with many assigned tasks experience improved performance of queries on active process instances because the queries can filter out tasks from completed instances.
<task-index-store-fields>	This Boolean value controls whether the actual field values are stored as separate fields. The default value is false, which means that the actual field values are not stored as separate fields. You might want to change the value to true for debugging purposes because it improves the readability for people and it allows queries by other search tools.

<task-index-work-manager>	This string contains the JNDI name of the work manager that is used by the indexing process to manage the search index. The default value is wm/default, which is the default work manager for WebSphere Application Server. To improve the performance of the index creation, in the administrative console you can create a dedicated work manager with a greater number of available threads. You can then use this tag to switch to the new work manager.
<task-index-include-system-tasks>	This Boolean value controls whether system tasks are indexed. To enable system tasks to be displayed in Gantt charts in Process Portal, ensure that the value of this tag is set to true. If the value of this tag is to false, system tasks are not displayed in Gantt charts.
<process-index-instance-completion-best-effort>	This Boolean value controls whether completion dates are created when instances that are migrated from previous versions of IBM BPM are indexed. The default setting is false. If you change the value to true, the last completion date of the associated tasks is used for the instance completion date. If no associated tasks exist, the last modified time stamp of the instance is used as the completion date.

For more information, see Administering the Process Portal index IBM BPM 8.5.7 at the following IBM Knowledge Center website:

https://ibm.biz/BdsSdA

4.6 Process Federation Server

Federated environments consist of Process Federation Server and multiple federated IBM BPM systems. Use properties in the Process Federation Server `server.xml` configuration file to configure an IBM BPM system for the federated environment and configure and maintain the indexing service for the system.

4.6.1 Modifying configuration properties

You can modify all the general configuration properties in the `ibmPfs_federatedSystem` element for a federated IBM BPM system, except the `index.number_of_shards` property. You also can modify the indexer and retriever implementation properties that are specific to the system type.

4.6.2 Removing a federated IBM BPM system

To remove an IBM BPM system from the federated environment, in the `server.xml` file, remove or comment out the `ibmPfs_federatedSystem` element and the associated indexer and retriever elements for the system. The indexing service on the Process Federation Server node stops and the index is removed from federated queries. The index is still available on the node so that you can re-enable the IBM BPM system later.

To completely remove the index, stop all process federation servers and delete the corresponding index from the following directory on the Process Federation Server node:

```
/elasticsearch/data/federated_cluster/nodes/node_name/indices
```

To keep the federated IBM BPM system configured but temporarily stop the indexing service, remove or comment out the indexer element; for example, the `ibmPfs_bpdIndexer` element.

4.6.3 Rebuilding an index

If problems occur with the Process Federation Server index, you can rebuild it.

Complete the following steps to rebuild an index:

1. Stop all process federation servers.
2. Delete the index from the `/elasticsearch` directory on the file system on all process federation servers.
3. Restart all process federation servers.

The index is rebuilt.

4.6.4 Monitoring and administering an index

`Elasticsearch` provides an HTTP-based REST API to query, monitor, and administer the `Elasticsearch` service and indexes. For production environments, Process Federation Server provides forwarder applications to secure REST requests through an HTTPS secure connection. For more information, see the following resources:

- Enabling monitoring and administration of the `Elasticsearch` service:

  `https://ibm.biz/BdsSd6`

- *Maintaining IBM BPM systems in federated environments IBM BPM 8.5.7*, at the following IBM Knowledge Center website:

  `https://ibm.biz/BdsSxB`

4.7 Other data

Although most data accumulation comes from process, tasks, and events, other areas of accumulation must be considered over time as described in this section.

4.7.1 Document attachments

Heritage Coaches support uploading documents to an internal document store by using APIs and pre-supplied controls. Although these documents are not separately deletable, they are associated with process instances and as such, they are deleted when process instances are deleted. Alternatively, you can write your own service to delete these documents by using the JavaScript method `deleteAllVersions` in TWDocument.

When next generation Coaches were first introduced in V8.0.0 and V8.0.1, no equivalent built-in document attachment store capability was supplied (although there is support for accessing external enterprise content management systems).

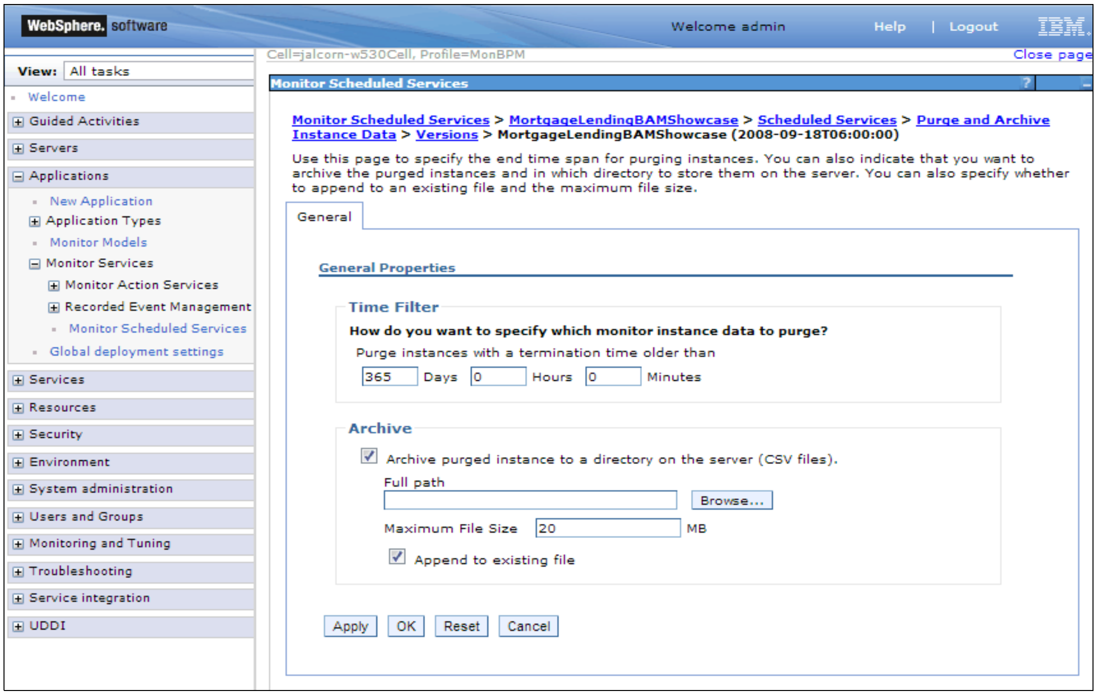
However, this built-in attachment capability was reintroduced in V8.5.0 that uses the same Coach views that are used for external ECM systems. As with heritage Coaches, there is no explicit means of deleting these attachments, but they are deleted when their parent process instances are deleted and can be programmatically purged by using **`deleteAllVersions`**.

4.7.2 Temp directory

During installation and some operations at run time, files are placed in the system temp directory (`%temp%`). These files can accumulate over time, so monitor this directory and develop a policy to occasionally purge it.

4.8 IBM Business Monitor

If you use IBM Business Monitor, occasionally purge data there because it can quickly accumulate. There is mature support in IBM Business Monitor for deleting and archiving data. Business Monitor includes an event purging capability in the administrative console that you configure per monitor model, as shown in Figure 4-5.



Monitor Scheduled Services > MortgageLendingBAMShowcase > Scheduled Services > Purge and Archive Instance Data > Versions > MortgageLendingBAMShowcase (2008-09-18T06:00:00)

Use this page to specify the end time span for purging instances. You can also indicate that you want to archive the purged instances and in which directory to store them on the server. You can also specify whether to append to an existing file and the maximum file size.

General

General Properties

Time Filter

How do you want to specify which monitor instance data to purge?

Purge instances with a termination time older than

365 Days 0 Hours 0 Minutes

Archive

☑ Archive purged instance to a directory on the server (CSV files).

Full path

Maximum File Size 20 MB

☑ Append to existing file

Figure 4-5 Purging Event Console page in IBM Business Monitor

You can specify the age of the instances to be deleted, and optionally identify a directory in which to archive the purged instances as a CSV file.

Important: This function purges terminated instances only; it never removes in-flight instances, regardless of age.

This purging and archiving can be done once, as described in Purging and archiving instance data, which is available at this website:

https://ibm.biz/Bdsmta

Purging also can be scheduled to run regularly, as described in *Schedule purging and archiving instance data*, which is available at this website:

https://ibm.biz/BdsSxf

4.9 IBM BPM and IBM Business Monitor Purging summary

This section highlights key areas of accumulating data that you must be aware of in IBM BPM and Business Process Monitor. It also describes how to purge that data. The data types and options for purging it in various releases are listed in Table 4-3.

Table 4-3 Options for purging various types of data based on release

Data	V7.5.1.x	V8.0.1.x	V8.5.0.x
Process apps and toolkits in Process Center	Archive and delete with Process Center UI	Archive and delete with Process Center UI	Archive and delete with Process Center UI
Snapshots in Process Center	BPMSnapshotCleanup command in V7.5.1.2	BPMSnapshotCleanup command in V8.0.1.2	BPMSnapshotCleanup, plus scheduled service added in V8.5.0.1
Advanced BLA and EARs in Process Center	Undeploy with Process Center UI to remove BLA and EARs	Undeploy with Process Center UI to remove BLA and EARs	Undeploy with Process Center UI to remove BLA and EARs
Snapshots in Process Server	BPMDeleteSnapshot command in V7.5.1.2	BPMDeleteSnapshot command in V8.0.1.0	BPMDeleteSnapshot command in
BPD Process Instances	LSW_BPD_INSTANCE_DELETE stored procedure	BPMProcessInstancesCleanup command in V8.0.1.2	BPMProcessInstancesPurging command in V8.5.7 and BPMProcessInstancesCleanup command, enhanced in V8.5.0.1
BPEL Process Instances	deleteCompletedProcessInstances.py script or Cleanup Service in BFM	deleteCompletedProcessInstances.py script or Cleanup Service in BFM	deleteCompletedProcessInstances.py script or Cleanup Service in BFM
BPD durable events	BPMDeleteDurableMessages command in V7.5.1.2	BPMDeleteDurableMessages command in V8.0.1.2	BPMDeleteDurableMessages
Performance Data Warehouse	prune command in V7.5.1.2	prune command in V8.0.1.2	prune command in V8.5.0.1
Business Monitor	The Purge and Archive Instance Data console action and schedulable service	The Purge and Archive Instance Data console action and schedulable service	The Purge and Archive Instance Data console action and schedulable service

5

Maintaining IBM BPM-dependent systems

This chapter focuses on the performance of and maintaining the underlying infrastructure and dependent systems that are used in IBM Business Process Manager (BPM). The nature of BPM is to regularly and methodically integrate different applications and people into a defined process, which involves integrations to various systems.

The performance and stability of IBM BPM relies on a maintained infrastructure. These integrations can be a few or many systems within or external to the IBM BPM installation.

This chapter includes the following topics:

- 5.1, “Lower stack level products, Java, WebSphere Application Server, and web servers” on page 70
- 5.2, “Security and LDAP connections” on page 72
- 5.3, “External integrations” on page 85

5.1 Lower stack level products, Java, WebSphere Application Server, and web servers

This section describes the underlying stack products on which IBM BPM is installed.

5.1.1 WebSphere Application Server and IBM Java upgrades

IBM BPM is a stack product that relies on IBM Java, IBM WebSphere Application Server, and the OS on which IBM BPM is installed. IBM BPM supports up-level fix packs of IBM Java and IBM WebSphere Application Server.

Starting with WebSphere Application Server version 8 and later, IBM Java upgrades are bundled with the WebSphere Application Server upgrades, which making it a single installation. For more information about the initial versions of WebSphere for each BPM release, see the systems requirements page under “Prerequisites” at the following website:

http://www.ibm.com/support/docview.wss?uid=swg27023007

The Java SDK version for each version of WebSphere is available at the following website:

https://www.ibm.com/support/docview.wss?uid=swg27005002

It is recommended to regularly upgrade to the latest WebSphere fix pack level. If a security vulnerability is found, the announcement and fix is available for all affected product versions. It also is recommended to apply security patches when they are available.

Security bulletins also are sent out by the IBM My Notification system. You can sign up to receive notifications updates for defects, fix packs, product announcements, and security bulletins at the following website:

http://www.ibm.com/software/support/einfo.html

5.1.2 Thread pools and sizings

IBM BPM uses several different thread types. These types feature different tuning needs that are based on your application design. For more information about monitoring systems and load testing optimizations for the thread pools, see *IBM Business Process Manager V8.5Performance Tuning and Best Practices*, SG24-8216, which is available at this website:

http://www.redbooks.ibm.com/abstracts/sg248216.html

All of these thread pools are managed at the WebSphere Application Server level. For more information about a tool that is available to assist with tuning, see the following website:

https://ibm.biz/BdsSHE

Common threads to tune with IBM BPM

The following common threads might need extra tuning:

- JMS threads, which are heavily used in BPEL applications, also are used to send messages to the BPMN event manager and invoke UCAs. For more information about tuning, see this website:

  https://ibm.biz/BdsS42

- ORB, which is used for Process Designer and Process Center communication. This thread also can be used in an integration service for a BPEL application.

- WebContainer, which is used for BPMN coaches, portal, and web interactions and for BPEL human task manager.
- For Work Manager thread, the BPMN event manager uses the `wm/BPMEventManagerWorkManager` thread. For more information about configuring, see this website:

  https://ibm.biz/BdsS4f

5.1.3 HTTP sessions

The primary way business users of IBM connect to IBM BPM is by using a browser. Users log in to a portal to view, claim, and complete tasks. In this section, the configuration that is described applies to the IBM BPM Process Portal and custom portal that is made by the Application Development team.

Session affinity

Session affinity (sometimes referred to as *sticky sessions*) is used at the load balancer in a clustered IBM BPM configuration. It is a required setting for clustered environments. When a user logs in, they are routed to one of the nodes in a clustered configuration. The user works on that server and continues there until their HTTP session is destroyed or times out.

For more information, see this website:

https://ibm.biz/BdsS4M

Session replication

IBM BPM does not support session replication. Session replication is used in some applications. When a node the user is connected to goes offline, the user can continue work without interruption. This continuation is not possible with IBM BPM as the main interaction of data is a coach. Coaches are EJB objects and the data from the users' interface is lost when a server becomes unavailable. The data cannot be retrieved.

> **Note:** When a JVM must be restarted during business hours, remove the server from the load balancer configuration. Taking this precaution prevents new users from being directed to the server that is being restarted. Monitor the load balancer to ensure that sessions are no longer active on that server. Monitoring this activity reduces the number of users who were interrupted from a server restart during the day.

User logout and session expiration

The following sessions are related timeouts in IBM BPM:

- Lightweight Third-Party Authentication

  Lightweight Third Party Authentication (LTPA) is issued by WebSphere for security authentication. The default timeout is 2 hours.
- EJB

  This session is the session bean timeout, which is most commonly connected to Coaches in the BPMN application. The default timeout is 2 hours.
- HTTP

  This session is the relationship between user and web server. The default timeout is 30 minutes.

Consult with your Business and Security teams about standards and expectations for user logout and session timeouts. For example, a requirement might be that a user should log in only from one client IP address at a time. This requirement is not something WebSphere Application Server or IBM BPM can configure. Single sign on (SSO) tools, such as WebSEAL from IBM Security Access Manager, can provide this feature. You might need to consult with your SSO provider for more security options, which can be a business or security requirement.

For more information about how to change the values for these session-related timeouts and technical details, see the following resources:

- https://ibm.biz/BdsSrG
- http://www.ibm.com/support/docview.wss?uid=swg21078845
- https://ibm.biz/BdsSrn
- https://ibm.biz/BdsSre

5.1.4 Operating system maintenance

The operating system in which BPM is installed also must be maintained. This maintenance includes having enough temporary disk space, appropriate swap space, and memory allocation for the IBM BPM server and any supporting applications, such as monitoring tools. Keeping the OS updated also is important for security and other bug fixes. For more information about supported operating systems for your IBM BPM versions, see the following System Requirements website:

http://www.ibm.com/support/docview.wss?uid=swg27023007

5.1.5 Virtualization

Your installation can be installed in a virtual machine (VM), which includes the main IBM BPM installation and the database. Considerations of how many VMs and the distribution of CPU, memory, and IO are needed.

If the system that is controlling the VMs is undersized, you can see CPU starvation or memory swapping. Consult with your virtualization platform and database vendor about how best to monitor and configure a virtual system. A good start is the article *Performance tuning considerations when IBM Business Process Manager (BPM) is running in a virtual machine*, which is available at this website:

http://www.ibm.com/support/docview.wss?uid=swg21574113

5.2 Security and LDAP connections

This section describes configuration options for the Lightweight Directory Access Protocol (LDAP), performance with LDAP, and some IBM BPM authorization. In this section, the term *LDAP* refers to any LDAP directory server that WebSphere supports as described in the following document:

http://www.ibm.com/support/docview.wss?uid=swg27036471

WebSphere Application Server can accommodate more security repositories. This section describes LDAP. The API calls from BPM to WebSphere Application Server behave the same way for all configured security providers in the federated repositories.

Note: The following terms are used in this section:

- Authentication

  Authentication refers to ensuring that the user is who they claim to be. This process is handled by WebSphere Application Server.

- Authorization

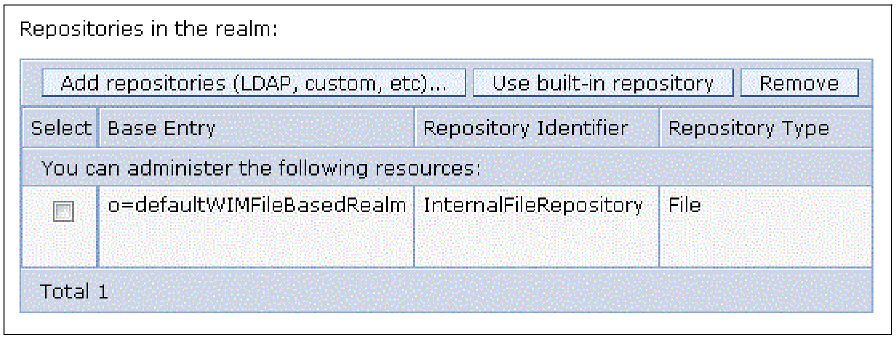
  After the user is authenticated, *authorization* dictates the features, consoles, operations, and data that a user can access. Authorization can be modified and granted through the IBM BPM configuration settings and through application design. Some can be changed at run time, others require a restart after configuration.

5.2.1 Default configuration

The default settings in IBM BPM 8.5 and later include federated repositories as its user registry. The only configured repository in the federation is the WebSphere Application Server internal file registry.

The federation can be easily extended to also include your corporate LDAP server or even a custom adapter to integrate non-LDAP-based user repository. Although other configurations are possible, their use is discouraged because they are uncommon and often include restrictions.

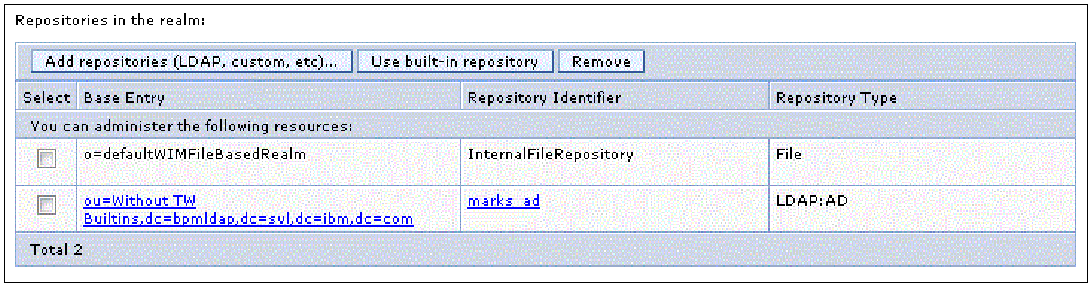
The file registry is the initial configuration. Figure 5-1 shows an example federated configuration that includes the `o=defaultWIMFileBasedRealm` configuration. It is a good practice to change the default realm name to avoid accidentally having SSO enabled between two environments.


Repositories in the realm:
Add repositories (LDAP, custom, etc)...
Use built-in repository
Remove

Select	Base Entry	Repository Identifier	Repository Type
You can administer the following resources:			
□	o=defaultWIMFileBasedRealm	InternalFileRepository	File
Total 1			

Figure 5-1 Federated configuration

Figure 5-2 shows an example of a configuration with a second repository (in this case, a Microsoft Active Directory Server).


Repositories in the realm:
Add repositories (LDAP, custom, etc)...
Use built-in repository
Remove

Select	Base Entry	Repository Identifier	Repository Type
You can administer the following resources:			
□	o=defaultWIMFileBasedRealm	InternalFileRepository	File
□	ou=Without TW Builtins,dc=bpmldap,dc=svl,dc=ibm,dc=com	marks_ad	LDAP:AD
Total 2			

Figure 5-2 Configuration with secondary repository

During the IBM BPM installation process, at least two users are defined: The deployment administrator and the cell administrator. You can provide a dedicated authentication alias and a dedicated user for all BPM security roles while the deployment environment is created. These users are mapped to the WebSphere administrative roles and the IBM BPM security groups.

Extra local users and local security groups can be added manually through the IBM BPM Process Administration console or the WebSphere Application Server console. Scripting options also are available to add users by using wsadmin commands.

The following BPM and WebSphere Application Server user and group security consoles are available:

- Add local users and security groups in WebSphere Administration Console, as shown in Figure 5-3.

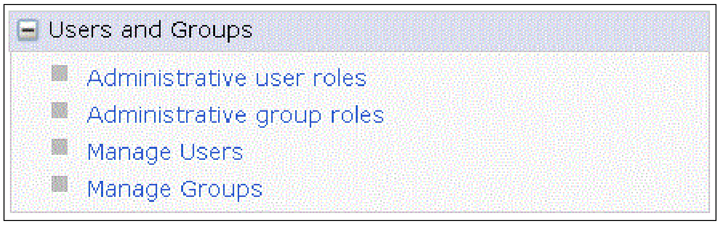


Figure 5-3 WebSphere Application Server console

- Maintain relations with users and internal groups and modify user attributes in the IBM BPM Process Administration console, as shown in Figure 5-4.

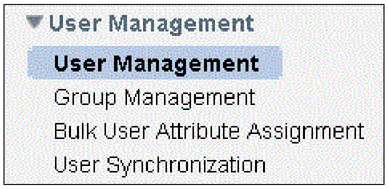


Figure 5-4 IBM BPM Process Administration user console

For more information about WebSphere Application Server wsadmin references (user and group administration commands), see this website:

`https://ibm.biz/BdsSr5`

5.2.2 Default users and groups

During product installation, two users are configured: the deployment administrators and cell administrators. These users configure and run WebSphere Application Server and start the IBM BPM systems. It is not recommended to put these users in LDAP. If the primary IBM BPM administrative users are in LDAP and the LDAP server cannot be reached, you cannot log in to the system to perform operational duties. If the LDAP server is down, administrative login security can be disabled; however, logins can still occur without security credentials, which might not be an allowed security policy.

In IBM BPM v8.5 and later, the document storage is implemented with an embedded IBM FileNet server. The configuration of this component is required, even if your application does not use the internal document storage feature.

For more information about the limitations of the document store that relates to the security configuration for IBM BPM, see this website:

https://ibm.biz/BdsSs7

Some security requirements state that all users must be in an LDAP or other outside repository. If this requirement is applies to you, specific steps are needed to properly configure the EmbeddedECMTechnicialUser for LDAP. For more information, see the following documentation:

https://ibm.biz/BdsSsW

For more information about the default BPM security groups, see the following documentation:

https://ibm.biz/BdsSic

Note: Do not delete the default security groups or users, which are needed for core functionality. If these groups or users are removed or modified, contact IBM BPM Support for assistance. The group `tw_all_users` is a special group; therefore, do not add users or groups to this local security group.

5.2.3 LDAP

This section describes configuring LDAP and provides useful tips.

Preparing to configure LDAP

Before configuring LDAP with IBM BPM, first review with the business which organization units and groups in the company need access to IBM BPM. After you draft a list of users and groups, work with the LDAP administrator to develop an entry point to the LDAP tree.

With the federated repositories, you can have multiple configurations to LDAP. In addition to LDAP tree filtering, groups can be narrowed with a search filter. It is important to limit the scope of LDAP of which IBM BPM can access. The reason is metadata about users and groups (no passwords, only names and membership relations) are stored in the IBM BPM database tables. There is no current functionality to remove metadata that is synced from LDAP. Therefore, we recommend careful planning before LDAP is configured. Each environment features a different base configuration.

Generally, the following user groups need access to IBM BPM:

- Developers
- Operations team
- Business users
- Testers

These users need access to the following types of IBM BPM installations:

- Development
- QA, Test, or Pre-production
- Production

The four groups across these environments can overlap or have the same people, depending on the company structure and separation of duties (see Table 5-1).

Table 5-1 Example user types per environment

Environment	User types
Development (Process Center)	Operations, development
QA, TEST	Operations, development, testers
Production	Operations, business users

Configuring LDAP

To configure an LDAP repository, review the directions that are included in the following WebSphere documentation:

`https://ibm.biz/BdsSir`

Contact the LDAP administrator for more information about the ports, host name, and LDAP filter list that authorizes users for the server that you are configuring. It is recommended to configure LDAP with only the deployment manager (dmgr) running and other servers that are turned off. By following this approach, you can quickly test and determine whether there are any issues with the configuration. You can also check and see whether all expected users and groups are visible in the Users and Groups section of WebSphere. If WebSphere cannot see a user or group, IBM BPM also cannot see the user or group.

Resolve LDAP configuration issues before starting the nodes and BPM JVMs (AppTarget). This issue also is important to not accidentally add groups and users to BPM user and group XREF meta tables. No functionality is available to remove the metadata of users and groups that are synced from LDAP.

Note: Before configuring LDAP, backup the `wimconfig.xml`, `fileRegistry.xml`, and `security.xml` files. These core files are used for configuring security in WebSphere. If the settings are not correct after the deployment manager is restarted, replace the files with your backup version and try the configuration again.

After LDAP is configured, see the IBM BPM administration section of the IBM Knowledge Center for more information about how to grant users authorization to different areas of the product:

`https://ibm.biz/Bdsmt6`

This process includes authorization capabilities in the Process Center Repository, authorization to the Process Admin console, configuration of Teams during run time, and other authorization options.

For more information, see the following documents:

- IBM Business Process Manager security overview:

  `https://ibm.biz/BdsSi2`

- Managing access to the Process Center Repository

  `https://ibm.biz/BdsSiz`

- Configuration properties for Process Portal action policies:

  `https://ibm.biz/BdsSiq`

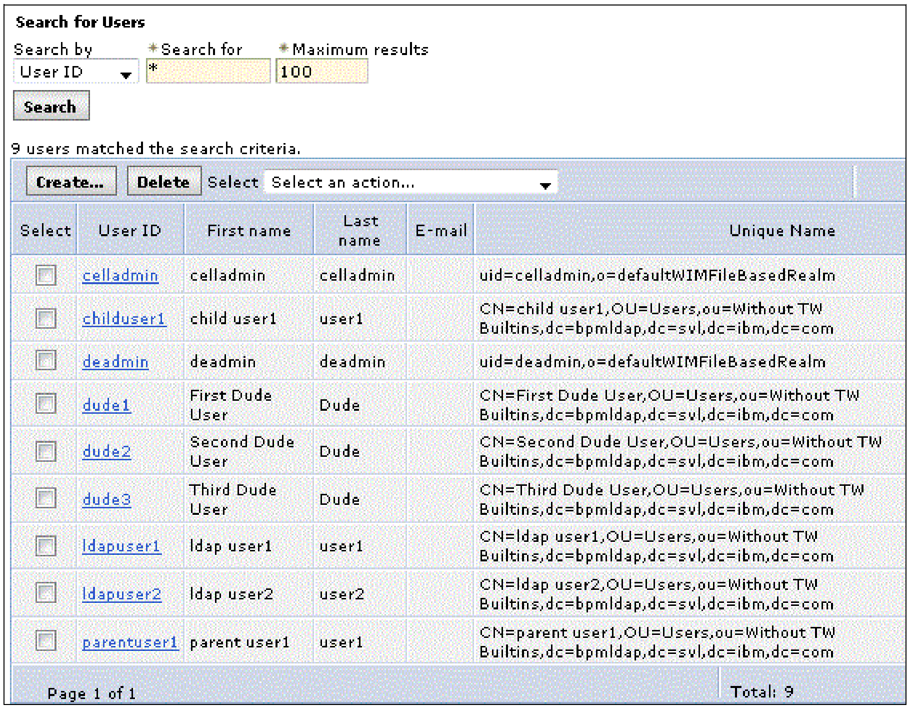
- Modifying the Process Admin Console setting and behavior:

  https://ibm.biz/BdsSif

- Creating a secure environment:

  https://ibm.biz/BdsSiP

After LDAP is configured, you can search for users and groups that are in the federated repositories. Examples of the WebSphere Application Server user and groups section system after LDAP is configured are shown in Figure 5-5 and Figure 5-6. You can see the local file users of `celladmin` and LDAP users, such as `ldapuser1`.



Search for Users

Search by: User ID | * Search for: * | * Maximum results: 100

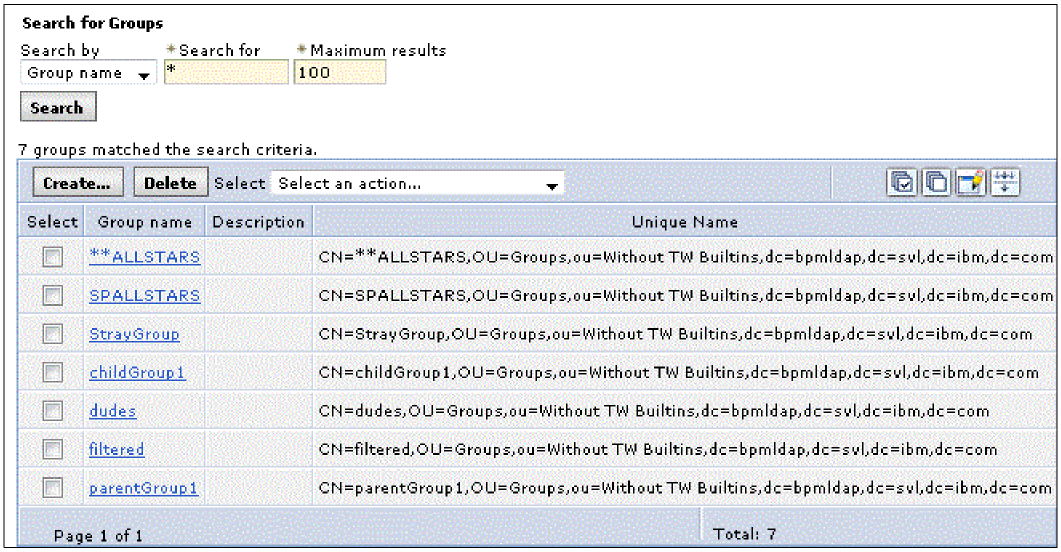
Search

9 users matched the search criteria.

Select	User ID	First name	Last name	E-mail	Unique Name
□	celladmin	celladmin	celladmin		uid=celladmin,o=defaultWIMFileBasedRealm
□	childuser1	child user1	user1		CN=child user1,OU=Users,ou=Without TW Builtins,dc=bpmldap,dc=svl,dc=ibm,dc=com
□	deadmin	deadmin	deadmin		uid=deadmin,o=defaultWIMFileBasedRealm
□	dude1	First Dude User	Dude		CN=First Dude User,OU=Users,ou=Without TW Builtins,dc=bpmldap,dc=svl,dc=ibm,dc=com
□	dude2	Second Dude User	Dude		CN=Second Dude User,OU=Users,ou=Without TW Builtins,dc=bpmldap,dc=svl,dc=ibm,dc=com
□	dude3	Third Dude User	Dude		CN=Third Dude User,OU=Users,ou=Without TW Builtins,dc=bpmldap,dc=svl,dc=ibm,dc=com
□	ldapuser1	ldap user1	user1		CN=ldap user1,OU=Users,ou=Without TW Builtins,dc=bpmldap,dc=svl,dc=ibm,dc=com
□	ldapuser2	ldap user2	user2		CN=ldap user2,OU=Users,ou=Without TW Builtins,dc=bpmldap,dc=svl,dc=ibm,dc=com
□	parentuser1	parent user1	user1		CN=parent user1,OU=Users,ou=Without TW Builtins,dc=bpmldap,dc=svl,dc=ibm,dc=com

Page 1 of 1 | Total: 9

Figure 5-5 List of file registry and LDAP users in WebSphere Administration console



Search for Groups

Search by: Group name | * Search for: * | * Maximum results: 100

Search

7 groups matched the search criteria.

Select	Group name	Description	Unique Name
□	**ALLSTARS		CN=**ALLSTARS,OU=Groups,ou=Without TW Builtins,dc=bpmldap,dc=svl,dc=ibm,dc=com
□	SPALLSTARS		CN=SPALLSTARS,OU=Groups,ou=Without TW Builtins,dc=bpmldap,dc=svl,dc=ibm,dc=com
□	StrayGroup		CN=StrayGroup,OU=Groups,ou=Without TW Builtins,dc=bpmldap,dc=svl,dc=ibm,dc=com
□	childGroup1		CN=childGroup1,OU=Groups,ou=Without TW Builtins,dc=bpmldap,dc=svl,dc=ibm,dc=com
□	dudes		CN=dudes,OU=Groups,ou=Without TW Builtins,dc=bpmldap,dc=svl,dc=ibm,dc=com
□	filtered		CN=filtered,OU=Groups,ou=Without TW Builtins,dc=bpmldap,dc=svl,dc=ibm,dc=com
□	parentGroup1		CN=parentGroup1,OU=Groups,ou=Without TW Builtins,dc=bpmldap,dc=svl,dc=ibm,dc=com

Page 1 of 1 | Total: 7

Figure 5-6 List of LDAP Groups in WebSphere Administration console

Figure 5-7 shows a list of groups in the IBM BPM Process Admin Group Administration panel. LDAP groups cannot be removed, only displayed.

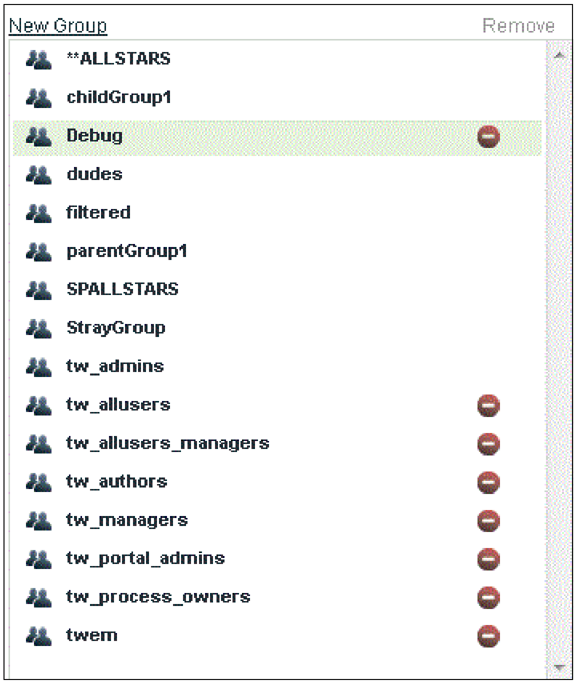
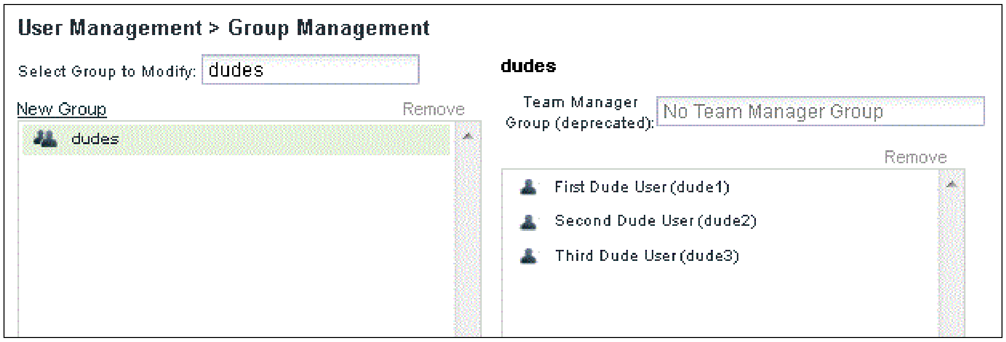


Figure 5-7 List of Groups in BPM Process Admin Console with a wildcard search

Figure 5-8 shows an LDAP group in the IBM BPM Process Admin console. The group membership here is read-only and cannot be changed.



Select Group to Modify: dudes

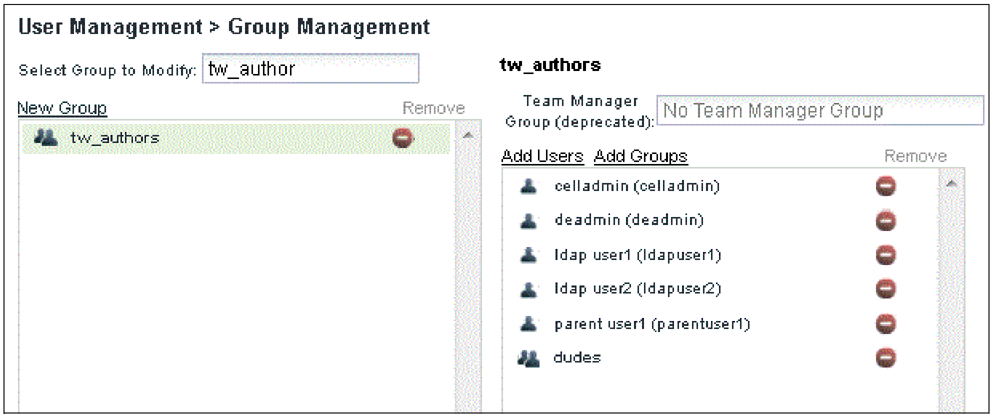
dudes

Team Manager Group (deprecated): No Team Manager Group

First Dude User (dude1)
Second Dude User (dude2)
Third Dude User (dude3)

Figure 5-8 LDAP membership is read only

Figure 5-9 shows the group `tw_author`, which grants privileges to the Process Center repository that is configured with LDAP users and groups.



Select Group to Modify: tw_author

tw_authors

tw_authors

Team Manager Group (deprecated): No Team Manager Group

- celladmin (celladmin)
- deadmin (deadmin)
- ldap user1 (ldapuser1)
- ldap user2 (ldapuser2)
- parent user1 (parentuser1)
- dudes

Figure 5-9 BPM Security group with local users and LDAP members

5.2.4 User and group synchronization

User and group synchronization occurs automatically and can be initialized by using commands. These concepts are important for IBM BPM. The time and length for the automatic synchronization depends on the number of members that are in an individual security group or team and the size of the connected LDAP.

Server start

During the process server or process center (AppTarget JVM) startup, IBM BPM scans and updates new groups that it sees from the federated repositories. If new groups are visible, it adds these groups to the XREF tables and the Group Cache.

The start and end of the synchronization is recorded as an information message in `SystemOut.log`, as shown in Example 5-1.

Example 5-1 SystemOut messages for group sync start and completion

```
wle_security  I com.lombardisoftware.server.core.GroupCore getAllGroupsInternal
CWLND0005I - The group replication process has started.
wle_security  I com.lombardisoftware.server.core.GroupCore getAllGroupsInternal
CWLND0006I - The group replication process has completed.
```

During startup, you might see the exception `MaxResultsExceededException: CWWIM1018E`. This exception means that the maximum search value in the LDAP settings was reached. The search limit must be increased or a filter added for groups that are not needed.

For more information, see the following WebSphere Application Server documentation:

https://ibm.biz/BdsSJN

For more information about max search results, see the following resources:

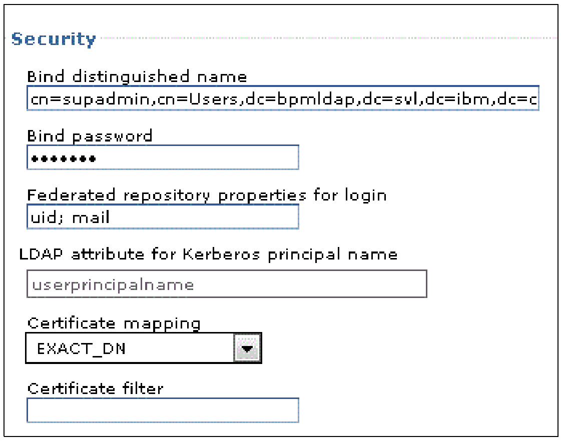
- https://ibm.biz/BdsSAc
- https://ibm.biz/BdsSAx

User login

This section describes configuration and other notes for user login.

LDAP configuration

When you add LDAP to federated repositories, you can specify a login ID. One or more values can be entered so that users can log in with one or more unique attributes. This information is the Federated repository properties for login field setting in the configuration page for LDAP, as shown in Figure 5-10. Typically `uid; mail` is entered, which allows users to log in with their user ID or mail setting. An example of a user ID can be `35121243` and the mail is `userX@companyX.com`.



Security

Bind distinguished name
cn=supadmin,cn=Users,dc=bpmldap,dc=svl,dc=ibm,dc=c

Bind password
•••••••

Federated repository properties for login
uid; mail

LDAP attribute for Kerberos principal name
userprincipalname

Certificate mapping
EXACT_DN

Certificate filter

Figure 5-10 WebSphere Administration Console for configuring LDAP

It is not recommended to use only the mail ID for this configuration because user emails can change. By having the user ID listed first, this entry is entered into the IBM user XREF tables. If a user email changes or has a new alias, such as `userX-2@companyX.com`, they can then log in and still see the same data as the IBM BPM user tables are using the user ID as the primary reference.

Initial and subsequent user login

When a user first logs in to IBM BPM, their primary login ID is added to the database and all group memberships. Subsequent logins refresh only the group membership for the user. Therefore, if user X transferred from Development to Product Management, this relation is reflected in the local database. The user then sees a different set of tasks as their group membership changed and thus their authorization to see tasks.

There are some considerations for login and user lookups. The first is where the group cache is referenced upon login. By default, it accesses updated information from LDAP. This configuration can be changed to use the local cached information in the database. To enable this feature, a `100Custom.xml` setting and a WebSphere Application Server LDAP configuration must be applied. In addition, the use of the wsadmin commands to refresh group and user memberships must be run periodically.

For more information about changing the group, see the following resources:

- `https://ibm.biz/BdsSAJ`
- `https://ibm.biz/BdsSAA`

If the application was created before IBM BPM 8.5, places can exist in which the task assignment is set as an “Expression.” This group dynamic and an example configuration option for this setting is shown in Example 5-2 on page 81.

Example 5-2 Configuration options for controlling when dynamic groups are updated

```
<update-dynamic-groups-on-login>
<update-dynamic-groups-on-task-creation>
<update-dynamic-groups-on-uadchanged>
```

The default values are set to true and are configurable by using a `100Custom.xml file`. The new feature uses Teams with services.

For more information, see the following website:

https://ibm.biz/BdsSAW

Group member cache can be changed to reference the database (see Example 5-3), which reduces calls to LDAP. In environments where the IBM BPM user population LDAP membership is static, changing to the database improves performance.

Example 5-3 Configuration option for group member cache

```
<group-member-cache-source>
```

For more information about the group member cache feature, see this website:

https://ibm.biz/BdsSuD

Installing a snapshot

During the installation of a snapshot, teams that are defined in the application are added to target servers group tables. A unique entry is used for each team entry for each snapshot. After a snapshot is installed, the users and groups that are defined in the team can be changed at run time in the Team Bindings section. Consult with the Application Development and Business teams who should be assigned in each environment in the role bindings page. For more information, see this website:

https://ibm.biz/BdsSuH

Updating users and groups

Several methods are available to update users and their group memberships. The command line scripts are common for the Operations group because they have console access. The JavaScript and REST commands tend to be used in the application. However, the Operations team also might create their own helper applications to facilitate security maintenance.

Command-line scripts

Users and groups can be synced by using the wsadmin commands. These commands allow for synchronization outside of a user login or server startup routine. This configuration can be greatly beneficial to offload the time to create users and groups for large LDAP systems. The group and user functions are helpful for the following cases:

- Adding an application to IBM BPM and extending the number of people who can access IBM BPM. You can synchronize the known LDAP groups that are defined in the Role Bindings page. For applications where tasks are assigned in a round-robin or other dynamic fashion, pre-synchronizing groups and users is helpful.
- Reducing the time for a user's first login to IBM BPM.
- A known LDAP group has high turnover rate. For example, a group for a call center has a 25% change every month. Use the wsadmin command to synchronize members of this group.

- Dynamic groups; for example, there is a dynamic group that is created with the combination of data points (customer type, account value, and geographic region) to determine the best set of people to work on it.

For more information about the commands that are used to synchronize user and groups, see the following resources:

- https://ibm.biz/BdsSuq
- http://www.ibm.com/support/docview.wss?uid=swg1JR48507
- http://www.ibm.com/support/docview.wss?uid=swg1JR48172

JavaScript API

The TWTeam and TWUser objects include methods to find and update other information about the Users and Teams. For more information, see this website:

https://ibm.biz/BdsSuM

REST API methods

The REST API includes a useful set of commands for users and groups. A built-in testing tool is available that lists all REST API methods. This tool is available at this website:

http://server:port/bpmrest-ui

For more information about User, Group, and Team REST functions, see this website:

https://ibm.biz/BdsSuG

5.2.5 IBM BPM database and custom data sources

IBM BPM is backed by a database and many applications feature integrations to outside databases. This section describes maintaining and configuring these systems.

Maintaining the BPM database

From an operations perspective, the database affects the system health most, almost as much as application design and interactions from other integrations. The IBM BPM system is highly reliant on the database for daily work and maintaining performance.

For more information about tips and commands for DB2, see the developer works article series, *IBM Business Process Manager database troubleshooting, Part 1: What you can learn from your IBM DB2 for Linux, UNIX, and Windows database*, which is available at this website:

http://www.ibm.com/developerworks/bpm/bpmjournal/1509_volz1-trs/1509_volz1.html

The same methodology and similar commands can be applied to Oracle and Microsoft SQL Server databases.

Tuning recommendations

IBM BPM uses a relational database to store the application models and business data. The design of the database is not an archival repository. Regular data deletion is necessary for best performance. For more information about purging data, see Chapter 4, “Purging and archiving in IBM BPM systems” on page 51.

IBM BPM inserts and updates data. This configuration is different than that from a database, which is primarily used for read statements (SELECT). Database statements in IBM BPM involve multiple tables and most of these tables involve multiple updates, which affect several tables per transaction. The operations team must work with the DBA to tune the database so that the database is optimized to update and insert large data sets.

Indexes and reorganization

IBM BPM features a set of indexes across many tables. These indexes facilitate finding information quickly. As the number of rows in tables increase through daily use or decreases through deletion, the indexes might need to be reorganized.

As the percentage of rows (which are inserted or deleted from a table) increases since the last index, the greater the need for a reorganization. Maintaining the indexes is an important part of keeping an IBM BPM system healthy. Typically, an IBM BPM production system needs reorganization of tables once a week during a scheduled maintenance window or low-usage period. Monitoring your system over time indicates whether indexes must occur more frequently.

Load test systems that generate large amounts of data and are then deleted should be reorganized between testing scenarios. Monitoring the growth of records in the database can show that certain indexes might need to be run on a more frequent basis (possibly once a day). For more information about how to run the reorganization on indexes, see your database vendor’s documentation.

Adding custom indexes

Creating indexes for the IBM BPM tables often is not needed. If evidence shows that a new index is needed, the issue can be solved by enabling a current feature or applying a fix pack that includes the updated index. One example is the saved search accelerator tables. These changes dramatically increase the speed of finding task records and decrease stress on the `LSW_TASK` table. This feature functions as an index would on the database. It is recommended that all runtime servers have this feature configured. For more information, see this website:

https://ibm.biz/BdsSut

For more information about the use of custom indexes with IBM BPM to add an index to an IBM BPM table, see the following website:

https://ibm.biz/BdsSu5

Query plans and reviewing performance

In addition to indexes, another area to consider is query plans. When a query is sent to the database, the database engine determines a path for the data to take. For simple single selects, few paths are available. For more complex queries that involve multiple joins, conditions, and groupings, different ways are available for the database to internally search for this information.

For queries that do not improve after reindexing and data deletion, the query plan might need to be investigated. For more information about tips and commands for DB2, see the developer works article series, *IBM Business Process Manager database troubleshooting, Part 2: Solve performance issues with IBM DB2 for Linux, UNIX, and Windows examples*, which is available at this website:

http://www.ibm.com/developerworks/bpm/bpmjournal/1509_volz2-trs/1509_volz2.html

Upgrading databases without affecting IBM BPM downtime

To maintain the highest availability for system uptime, the database can be set up to allow rolling upgrades. These upgrades allow you to apply the next database fix pack and keep IBM BPM running. Consult with your DBA and database vendor for proper configuration of this high availability scenario.

JDBC provider and data source configurations

After IBM BPM is installed, the JDBC driver must be updated. Custom data sources that are created for the application also might need tuning.

JDBC driver

After the IBM BPM system is installed, upgrade the JDBC drivers to the version that matches your database. IBM BPM requires Type 4 JDBC driver. It is a best practice to mach the JDBC driver version to the version of the database, including the IBM BPM product data sources and any added custom data sources. An example of `SystemOut.log` that shows the named data source and associated JDBC driver version is shown in Example 5-4.

Example 5-4 Example SystemOut message with JDBC Version from DB2 without the box drivers

```
DSRA8225I: DataSource JNDI name : jdbc/TeamWorksDB
DSRA8203I: Database product name : DB2/NT64
DSRA8204I: Database product version : SQL10053
DSRA8205I: JDBC driver name  : IBM Data Server Driver for JDBC and SQLJ
DSRA8206I: JDBC driver version  : 4.11.69
DSRA8218I: JDBC driver specification level  : 4.0
DSRA8212I: DataStoreHelper name is:
com.ibm.websphere.rsadapter.DB2UniversalDataStoreHelper@67c85533.
DSRA8208I: JDBC driver type  : 4
```

For more information, see the following resources:

- IBM DB2 JDBC Driver download list:

  http://www.ibm.com/support/docview.wss?uid=swg21363866

- For more information about Microsoft SQL Server download list:

  https://msdn.microsoft.com/en-us/library/mt484311%28v=sql.1%29.aspx

- Oracle Database download list:

  http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html

Data source properties

Common tuning for data source connection pool properties is connection timeout, max connections, and unused timeout. For the IBM BPM product default data sources, these properties should not need to be tuned. This issue is true for almost all default values in IBM BPM 8.5.0.1 and later.

Custom data sources that are added for the applications must be configured and tuned. During a load test, monitor the custom data sources and determine how often connections were released and if the pool was ever exhausted. For more information about configuring a JDBC provider and data source in WebSphere for your custom applications, see this website:

https://ibm.biz/BdsSLj

5.3 External integrations

Various configuration tuning in the IBM BPM installation were described in this chapter. The IBM BPM applications connect to outside systems with the various connections that were described.

During initial configuration and troubleshooting, it is a good idea for the Operations team to have a list of all dependent systems and a way to identify host names or IP addresses. Having a list or an internal tool to find the owners of the system reduces time and cycles to find the owners who can help resolve issues.

6

Problem determination and remediation

This chapter describes problem determination, including a methodology of finding and isolating problems, anticipating problems, and tips to resolve Business Process Model and Notation (BPMN) and Business Process Execution Language (BPEL) problems.

Monitoring and problem determination are closely related operations. Alerts and data that are collected during monitoring assist in problem determination when an incident occurs. By its nature, an IBM Business Process Manager (BPM) solution relies on many components, applications, and systems that are working in a harmonious orchestrated fashion.

At times, one system or component can behave unpredictably. This behavior can affect a specific component or possibly the entire system. Therefore, problem determination is an important aspect in maintaining an IBM BPM solution.

This chapter includes the following topics:

6.1 Problem determination

Resolving problems is a collaborative and iterative approach. The basic steps for resolving all problems are the same. For each problem, a document is needed to keep the documentation of the research and resolution streamlined. Problem determination includes the following overall steps:

1. Identify the problem.
2. Gather information.
3. Understand the problem.
4. Search for a solution in the knowledge base by using keywords and symptoms. If a solution cannot be found, research and determine the solution.
5. Test the solution.
6. Analyze the solution.
7. Resolve the problem.

It is important to remember that resolving problem is an iterative approach. During each cycle, a new problem might be identified, a suspect area eliminated, or the time span of a data collection might not contain an instance of the problem.

After applying and observing the solution, the problem might not be resolved or only part of the reported problem is resolved. Research continues and returns to the gathering information phase so that the problem can be captured and identified.

For more information about outlining problem determination and the various aspects of the process, see Chapter 2 of *Problem Determination Across Multiple WebSphere Products AIX Platform*, SG24-7043, which is available at this website:

http://www.redbooks.ibm.com/redbooks/pdfs/sg247043.pdf

This Redbooks publication includes a thoughtful and holistic approach to resolving problems. It is a great resource for those users who are new to problem determination and is a useful review for seasoned users.

6.1.1 Resources for finding solutions to problems

After a log message or symptom is identified, it is possible it was reported before. Take the relevant text and messages and description and search for them on the Internet. It might be a reported defect with a code fix or a configuration change to resolve the problem. The solution might also be in a forum or blog post.

The following resources are available for solutions:

- dW Answers BPM Space (includes a question and answer forum and Good Practices and Examples section):

  https://developer.ibm.com/answers/smartspace/bpm/

- Developer Works forum, open forum:

  https://ibm.biz/BdsSUd

- Dr. Debug, which includes helpful tips for resolving problems with IBM BPM:

  https://ibm.biz/BdsSUD

- IBM Support TV, which includes detailed videos about specific data collection or configuration options for IBM products:

  `https://www.youtube.com/channel/UC9K3qy72l9D27vPeKWoUq7Q`
- IBM Technical Support:

  `https://www.ibm.com/support/entry/myportal/support`
- IBM Services:

  `https://ibm.biz/BdsSUF`
- IBM On Demand Consulting:

  `http://www.ibm.com/developerworks/middleware/services/odc/`
- Accelerated Value Program (AVP):

  `http://www.ibm.com/software/rational/support/tsas/`

6.1.2 Problem determination example

The first step in determining a problem is to identify and describe the problem. Initial steps focus on who, what, when, where, and why. For each of these areas of focus, the breadth of impact also is assessed (all, some, or none). In this section, we describe two examples of this process.

Performance

When a user accesses the Human resource (HR) application, the response time for the forms and navigation is too slow.

Questions with possible answers:

- Who is affected by the problem?

  Managers.
- When did the problem occur?

  Yesterday.
- What are the symptoms of the problem?

  The monthly review chart now takes 5 minutes to complete. It used to take 30 seconds.
- Where in the application does the problem occur?

  The managers use a custom dashboard that is named “Monthly HR Review”.
- Why is this causing a problem?

  The average manager has 25 employees. This process is taking a significant amount of time for users and delaying business work. From the operations side, CPU increased 25% more than normal.

Exception

Consider the exceptions that appear in the `SystemOut.log`, as shown in Example 6-1.

Example 6-1 Exceptions in SystemOut.log

```
wle          E   CWLLG0391E: An update operation failed.
com.lombardisoftware.core.TeamWorksRuntimeException: String "NYC, NY" cannot be
converted to "Integer" type. The value is not integer.
at
com.lombardisoftware.core.helpers.STConversionHelperImpl.doIntegerConversionIfPoss
ible(STConversionHelperImpl.java:256)
at
com.lombardisoftware.core.helpers.STConversionHelperImpl.toSTSupportedObject(STCon
versionHelperImpl.java:98)
at
com.lombardisoftware.core.helpers.SymbolTableObjectHelperRegistry.toSTSupportedObj
ect(SymbolTableObjectHelperRegistry.java:66)
at com.lombardisoftware.server.core.SymbolTable.set(SymbolTable.java:413)
at
com.lombardisoftware.core.script.js.AbstractTWSymbolTableScriptable.put(AbstractTW
SymbolTableScriptable.java:176)
at org.mozilla.javascript.ScriptableObject.putProperty(ScriptableObject.java:1729)
at org.mozilla.javascript.ScriptRuntime.setObjectProp(ScriptRuntime.java:1557)
at org.mozilla.javascript.ScriptRuntime.setObjectProp(ScriptRuntime.java:1547)
at org.mozilla.javascript.Interpreter.interpretLoop(Interpreter.java:3036)
at org.mozilla.javascript.Interpreter.interpret(Interpreter.java:2487)
at org.mozilla.javascript.InterpretedFunction.call(InterpretedFunction.java:164)

wle          E   CWLLG0594E: An exception occurred while initializing the
runnable thread. Error:
com.lombardisoftware.component.common.workflow.WorkflowProcessItemException:
Runtime error in script ("Process: 'Enter Address' ProcessItem: 'initialize' Type:
'ITEM'" 0:0).Internal Script error:
com.lombardisoftware.core.TeamWorksRuntimeException: String "NYC, NY" cannot be
converted to "Integer" type. The value is not integer
```

Questions:

- How often is this issue occurring?
- Did it happen after a new code release?
- Do all users report a problem or only a few?
- What do Internet search results show for the codes CWLLG0391E and CWLLG0594E?
- Does the development team know where “Enter Address” is in the application? If not, try the TWX Usage tool (see 6.2.13, “TWXUsage tool” on page 98).

These questions are the types of questions IBM Support asks. Only after the problem is sufficiently identified with how the problem affects the business can a meaningful dialogue begin that assists with resolving the problem.

6.1.3 Working with IBM BPM Support

At times, you might need to contact IBM Support to resolve a problem. This section describes working with IBM BPM Support.

Support Guide Handbook

The Support Guide Handbook outlines the roles and responsibilities of IBM and customers when engaging Support. It outlines what Support can provide and escalation procedures if needed and is available at this website:

http://www.ibm.com/support/customercare/sas/f/handbook/home.html

The handbook also is available for download from this FTP site:

ftp://ftp.software.ibm.com/software/server/handbook/webhndbk.pdf

Service Request tool

From the following primary electronic support portal, you can find information about preventing problems, downloading fixes, and engaging IBM Support:

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

The Service Request (SR) tool is the primary electronic method to open and update problem management records (PMRs). Through the following website, you can update tickets and send data to IBM. You can also send data to IBM by using FTP for larger files or batches of files:

http://www.ibm.com/support/docview.wss?uid=swg21153852

To access the SR tool, you must be a named user for your company. Each IBM Customer has a site technical contact who can add, approve, and remove users. These users can create, update, and view PMRs. They can also download fixes and product upgrades for your licensed software through IBM Passport Advantage®.

For more information about Site Technical Contact, see this website:

http://www.ibm.com/software/passportadvantage/site_technical_contact.html

If you do not know your Site Technical Contact or are having problems accessing the SR Tool, contact SR Help Team at the following website:

https://www.ibm.com/sr/help/stc.html

Passport Advantage and Fix Central

Passport Advantage is the site where you can download product installation binaries for your licensed software. For more information, see this website:

https://www.ibm.com/software/passportadvantage/

Fix packs and interim fixes (iFix) are available through the following IBM Fix Central website:

https://www.ibm.com/support/fixcentral/

Preventive maintenance

Chapter 5, “Maintaining IBM BPM-dependent systems” on page 69 and Chapter 4, “Purging and archiving in IBM BPM systems” on page 51 describe methods to monitor and prevent problems. This chapter describes problem determination. Some problems might be the result of product defects, which are then published and rolled into future releases.

It is recommended that upgrades are applied regularly. Work with your business and application teams on an upgrade schedule. Cumulative fixes will be released starting with IBM BPM 8.5.6.0. For more information, see the following TechNote:

http://www.ibm.com/support/docview.wss?uid=swg21964995

Working with Support on an Active PMR

A named contact is available for each PMR. It is important that this person have accurate phone and email contact information. You should regularly check your SR profile to have the current phone and email numbers.

Support personnel often must call directly to discuss the problem. A short phone call can help establish the problem description. If more than one person must access to the ticket, work with your Site Technical Contact to ensure they can view the PMR through the SR Tool.

When first opening a PMR, provide as much detail and data as possible with information you know and researched about the problem. Provide all of the data that you reviewed, which can include `SystemOut.log` files, trace files, heap dumps, or other data from your environment. If you have not collected data that is not provided, provide the basic IBM BPM MustGather, which includes a `.zip` file of the logs directory from each profile. For more information, see this website:

http://www.ibm.com/support/docview.wss?uid=swg21569731

It is best to collect data when a problem occurs. Collection this information might require coordination between several people and systems. After the first round of problem determination, a configuration change, application code change, code fix (APAR or Fix Pack), or other setting might be recommend. Then, the solution must be tested. A second round of data collection might be needed to confirm that the problem is no longer present. This process is an iterative and systematic approach to help resolve the problem.

6.2 Data collection methods and analysis tools

Some amount of data must be collected to resolve problems. Data collection during an instance of the problem occurring is essential. If data cannot be collected, symptoms and secondary data is needed.

Data from different sources at the same time also might be needed. Each of the collection points has specific information about the problem. There are unique markers between data sets that complete the picture of the system and the problem record.

The tools and examples that are described in this section are all public tools and resources that are available to help resolve problems. IBM Support uses these same tools.

6.2.1 Data gathering

After a problem is identified and research begins, data collection and then analysis of the data is next. The first place to start is with messages that the users see and then information in the `SystemOut.log` files. After the component area is identified, a trace on that area might be needed to gather more information. For more information, see the following IBM BPM MustGather web page:

http://www.ibm.com/support/docview.wss?uid=swg21569731

This page includes information about where logs are located and links to specific component areas.

Consider the following example scenario:

- Problem: Managers who use the “Submit HR Request” coach experience wait times of over 1 minute. Last week, the wait was only 5 seconds.

- Data Collection: A REST API trace, a web browser network trace, and a set of java thread dumps.
- Data Analysis: The slow response that was measured in the web browser network trace captures the URL, which is taking longer to respond than expected. This URL can be found in the REST API Trace. The thread ID in the trace can be correlated in the thread dumps to find what method in the API call is being used.
- Conclusion: Data shows the `/rest/bpm/wle/v1/exposed` REST API call is taking longer than other calls in the web traffic. A new set of applications and features were released last week and managers have access to more of these features than standard users. A web search shows APAR JR48570 as a good match for this problem. After applying the fix from IBM Fix Central, the problems is resolved. The IBM BPM Operations team makes a note of this defect and plans to upgrade to next fix pack, which includes this APAR.

6.2.2 Reading a log file

Using the standard and trace log files are essential for problem determination. Key elements in the file help you find key elements, which help find solutions. WebSphere Application Server logging features six elements in each line of logging. The log format is shown in Example 6-2.

Example 6-2 WebSphere Application Server log format

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

For more information about the format, see the following documentation for WebSphere Application Server:

https://ibm.biz/BdsvZj

An IBM BPM trace logging example is shown in Example 6-3. User, groups, and instance IDs were recorded in a few short lines.

Example 6-3 IBM BPM Trace from a task assignment

```
[2/2/16 9:12:45:830 PST] 000049bf WSUserRegistr 1
com.lombardisoftware.userorg.WSUserRegistryModule getIdFromPrincipalName
getIdFromPrincipalName() principalName=deadmin
[2/2/16 9:12:45:830 PST] 000049bf UserCore      >
com.lombardisoftware.server.core.UserCore getOrCreateUserInfoObject ENTRY User
name = deadmin, isCheckForUpdates = false
[2/2/16 9:12:45:830 PST] 000049bf UserCore      1
com.lombardisoftware.server.core.UserCore getOrCreateUserInfoObject User name =
deadmin, Cache data = [id=User.9 fullname=deadmin username=deadmin data={Full
Name=deadmin}]
[2/2/16 9:12:45:830 PST] 000049bf UserCore      <
com.lombardisoftware.server.core.UserCore getOrCreateUserInfoObject RETURN
[id=User.9 fullname=deadmin username=deadmin data={Full Name=deadmin}]
[2/2/16 9:12:45:845 PST] 000049bf NotifyCacheUp 1
com.lombardisoftware.server.NotifyCacheUpdateManager objectSaved objectSave()
called.
[2/2/16 9:12:45:845 PST] 000049bf Instrumentati 3   Instrumentation period 6
ended.
[2/2/16 9:12:45:845 PST] 000049bf PersistenceSe <
com.lombardisoftware.server.ejb.persistence.PersistenceServicesCore Save PO RETURN
```

```
[2/2/16 9:12:45:845 PST] 000049bf CaseHelper    >
com.lombardisoftware.bpd.runtime.engine.CaseHelper onTaskCreated ENTRY
BPDInstance.107 UserGroup.1713 null
```

From the logging, we see that these activities are all on thread `000049bf`. It is useful to find activities that are all in sequence as the thread does work. Log items are written as they are received and different threads can be next to each other.

One tip in troubleshooting is to find all logs for a specific work thread. This information can be found by using a find all or grep-like command on a text file, or by using the pull thread command from the IBM Support Assistant Tool. For more information about the syntax for these commands, see this website:

https://ibm.biz/Bdsm69

In Example 6-3 on page 93, all of the work that is done for this case is on a single thread, and these calls are synchronous. Some traces and analysis are asynchronous, or possibly parallel synchronous and are across different WebSphere thread IDs.

In cases of multiple threads, a different identifier is used to link data. To link threads, an instance ID, task ID, or other identifier can be used to link work across threads. In IBM BPM Standard, a component that is multi-threaded is under cover agents (UCA). For more information about UCA, see this website:

https://ibm.biz/Bdsm6Q

For more information about reading WebSphere Application Server-based products, see the following resources. The examples and methodology that is described here also applies to IBM BPM trace and SystemOut log files. Refer to these related documents for reviewing log files and correlation between other data sources:

- *Mapping Underlying Java Thread Identifiers to those in Logging and Trace*:

  https://www.ibm.com/support/docview.wss?uid=swg21418557

- *The Support Authority: Interpreting a WebSphere Application Server trace file*:

  https://ibm.biz/BdsvZp

6.2.3 Thread dump

A thread dump (sometimes referred to a *java core* that is not to be confused with a system core) is a full listing of all active threads in the JVM and the method call stack of those threads. It also shows the current list of classes in memory, information about the size of the JVM heap, and some information about the native memory that is used by the JVM.

A set of thread dumps over a short period with a small interval in between can show what threads are doing work and if any are being blocked. Thread dumps are primarily used for finding blocked or hung threads and evaluating performance problems.

The following methods can be used to generate thread dumps:

- The whole-system analysis of idle time (WAIT) tool (for more information, see 6.2.4, “WAIT tool”).
- WebSphere Application Server administrative console:

  https://ibm.biz/BdsvZ5

- wsadmin scripting:

  https://ibm.biz/BdsvZN

- Operating system commands:

  http://www.ibm.com/support/docview.wss?uid=swg21138203

The following related tools are available for Java core analysis:

- Thread and Monitor Dump Analysis for Java tool:

  https://ibm.biz/BdsvZ7
- Walkthrough of using the Thread and monitoring Dump analysis tool:

  https://ibm.biz/BdsvYc

6.2.4 WAIT tool

The WAIT tool is a convenient and compact way to generate thread dumps and collect information about system performance in a simple script. It can be configured to generate data at specific intervals. After data collection, it can be uploaded for analysis in an interactive visual report with drill-down charts for each thread dump and specific threads. For more information, see this website:

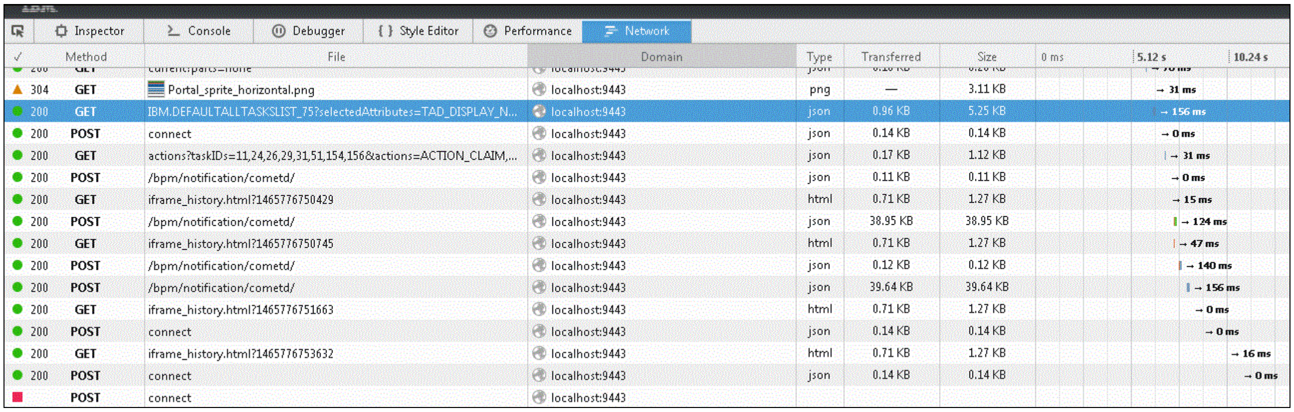
https://wait.ibm.com/

6.2.5 Web browser trace

When working with issues that are reported by users, information about the HTTP calls to and from user to the IBM BPM server are helpful. The assets, URLs, time spent on network, HTTP response codes, size of each call, and more information is shown in a web browser. Each browser includes its own set of built-in tools to view browser data and traffic. Third-party tools, such as Fiddler and Firebug, also are resources for capturing this data. For more information about collecting data and the available tools, see the following video at IBM Support TV:

https://www.youtube.com/watch?v=MQcUkus1Ndk

A sample web traffic trace is shown in Figure 6-1.



✓	Method	File	Domain	Type	Transferred	Size	0 ms	5.12 s	10.24 s
● 200	GET	current?parts=none	localhost:9443	json	[illegible]	[illegible]		→ 78 ms	
▲ 304	GET	Portal_sprite_horizontal.png	localhost:9443	png	—	3.11 KB		→ 31 ms	
● 200	GET	IBM.DEFAULTALLTASKSLIST_75?selectedAttributes=TAD_DISPLAY_N...	localhost:9443	json	0.96 KB	5.25 KB		→ 156 ms	
● 200	POST	connect	localhost:9443	json	0.14 KB	0.14 KB		→ 0 ms	
● 200	GET	actions?taskIDs=11,24,26,29,31,51,154,156&actions=ACTION_CLAIM,...	localhost:9443	json	0.17 KB	1.12 KB		→ 31 ms	
● 200	POST	/bpm/notification/cometd/	localhost:9443	json	0.11 KB	0.11 KB		→ 0 ms	
● 200	GET	iframe_history.html?1465776750429	localhost:9443	html	0.71 KB	1.27 KB		→ 15 ms	
● 200	POST	/bpm/notification/cometd/	localhost:9443	json	38.95 KB	38.95 KB		→ 124 ms	
● 200	GET	iframe_history.html?1465776750745	localhost:9443	html	0.71 KB	1.27 KB		→ 47 ms	
● 200	POST	/bpm/notification/cometd/	localhost:9443	json	0.12 KB	0.12 KB		→ 140 ms	
● 200	POST	/bpm/notification/cometd/	localhost:9443	json	39.64 KB	39.64 KB		→ 156 ms	
● 200	GET	iframe_history.html?1465776751663	localhost:9443	html	0.71 KB	1.27 KB		→ 0 ms	
● 200	POST	connect	localhost:9443	json	0.14 KB	0.14 KB		→ 0 ms	
● 200	GET	iframe_history.html?1465776753632	localhost:9443	html	0.71 KB	1.27 KB			→ 16 ms
● 200	POST	connect	localhost:9443	json	0.14 KB	0.14 KB			→ 0 ms
■	POST	connect	localhost:9443						

Figure 6-1 HTTP network traffic

6.2.6 SystemOut.log files

The `SystemOut.log` files contains information, warnings, and exception messages from the WebSphere Application Server, IBM BPM, and possibly custom application logging. This captured information is used for initial problem determination to find exceptions and warnings.

For more information about log message types, see the following WebSphere Application Server product documentation:

https://ibm.biz/BdsvYT

6.2.7 Trace files

Trace files are verbose debug logging for a specific component area. Multiple component areas can be combined in a trace to correlate different components to see timing and individual methods and variables; for example, a database trace and a BPD engine trace. The Trace Request Analysis tool can be used to find trace gap and method duration in trace files. For more information, see this website:

https://ibm.biz/BdR7vQ

Some requested traces can be across different component areas to capture all areas where the problem occurs. The general trace for debugging application deployments has five different logging areas, as shown in the following example:

```
*=info: WLE.*=all: WAS.clientinfopluslogging=all: com.ibm.bpm.fds.*=all:
ProcessApplicationLifecycle=all
```

For more information, see this website:

http://www.ibm.com/support/docview.wss?uid=swg21618716

6.2.8 Operating system diagnostics tools

The operating system **TOP**, **TOPAS**, **vmstat**, **nmom**, and **ps** commands report back the current memory, CPU, and processes that are running. The commands that are available depend on the operating system level that is used and the installed system administrator tools and packages.

By using the High CPU MustGather with thread dumps, a correlation between high CPU on the machine can be linked to a specific thread in the JVM. This feature is helpful in isolating problems.

For more information, see the following IBM Support MustGather Documents:

- IBM AIX®:

  http://www.ibm.com/support/docview.wss?uid=swg21052641
- Linux:

  http://www.ibm.com/support/docview.wss?uid=swg21115785
- Windows:

  http://www.ibm.com/support/docview.wss?uid=swg21111364
- Performance tool nmon (AIX and Linux)

  https://www.ibm.com/developerworks/aix/library/au-analyze_aix/

6.2.9 Heap dump

A heap dump is a snapshot of the memory of a JVM. The snapshot contains information about the Java objects and classes in the heap at the moment the snapshot is triggered. A heap dump can be generated automatically from an OutOfMemory event.

Two heap dumps that are generated from the same JVM (without restart) can be used to determine whether an unnatural increase in some collection of objects (a memory leak) occurred. Heap dumps also can be used to find suspiciously large collections of objects or small collections of large objects.

For more information about generating heap dumps, see the following resources:

- WebSphere Application Server administrative console:

  https://ibm.biz/BdsvZ5

- wsadmin script:

  https://ibm.biz/Bdsv2M

- Operating system commands:

  http://www.ibm.com/support/docview.wss?uid=swg21138203

- Memory Analyzer Version 1.4 tool:

  http://www.ibm.com/developerworks/java/jdk/tools/memoryanalyzer/

6.2.10 Core dump

A core dump or a crash dump is a memory snapshot of a running JVM process. A core dump can be automatically created by the operating system when a fatal or unhanded error (for example, signal or system exception) occurs. You can configure a core dump to be automatically triggered by a JVM. A core dump is similar to a heap dump, but it contains the objects and classes in the heap and the data that they contain. These dumps often are large (more than 10 GB) and require a large amount of disk space to generate.

For more information about IBM DTFJ with the Eclipse Memory Analyzer Tool, see this website:

http://www.ibm.com/developerworks/java/jdk/tools/mat.html

For more information about how to use the Memory Analysis tool, see this website:

https://ibm.biz/Bdsv2n

6.2.11 Verbose garbage collection

Verbose garbage collection (GC) refers to the verbose logging of garbage collection statistics for a JVM, which includes various metrics about how the Java GC algorithm is performing. This feature is useful to find memory leaks and memory usage over time. We recommend enabling verbose GC in production so that the system can be monitored over time.

For more information about how to enable verbose GC, see this website:

http://www.ibm.com/support/docview.wss?uid=swg21114927

For more information about available tools, see the following resources:

- IBM Pattern Modeling and Analysis Tool for Java Garbage Collector:

  https://ibm.biz/Bdsv23

- Garbage Collection and Memory Visualizer:

  http://www.ibm.com/developerworks/java/jdk/tools/gcmv/

6.2.12 Instrumentation within BPMN

Instrumentation within BPMN records the time to run a call stack within the BPMN code paths. This information is useful for performance debugging. For more information about how to enable and decode instrumentation files, see this website:

http://www.ibm.com/support/docview.wss?uid=swg21613989

6.2.13 TWXUsage tool

This tool is useful for finding a specific asset or text string in an application export. Some data in `SystemOut.log` or `trace.log` does not contain the asset or step name, only the UUID that is generated for the asset. This tool searches through the application and all dependent toolkits for the asset. A log file statement that contains a UUID and can then be searched with the TWX Usage tool is shown in Example 6-4.

Example 6-4 Log message with UUID string

```
CWLLG2229E: An exception occurred in an EJB call. Error:
null with ID TWClass.3527c918-7794-43fa-b38e-121125c4480
```

For more information about the TWXUsage tool, see this website:

https://hub.jazz.net/project/lashower/bpm-TWXUsage/overview

6.2.14 Wombat application analysis

The Wombat project is a tool that is used to analyze BPMN and BPEL applications. After the application is uploaded, a report is generated that includes application statistics and checks for possible concerns and anti-patterns in the application. For more information, see this website:

https://ibm.biz/Bdsvqu

This project is an ongoing project and not all practices are scanned by the tool. Cases, such as business logic, might loop and cannot be determined by this tool.

For more information about downloading the Analyzer tool, see this website (IBM ID is required):

https://wombat.mybluemix.net/

6.3 BPMN component problem determination

In this section, examples of sample output for different component areas are presented, including cache size, instance and task identifiers, and references to sample solution sets.

While the operations team is investigating issues in the IBM BPM environments, the conclusion and recommendation might need to be sent back to the application development team. Some problems can be reduced or corrected with application design changes.

6.3.1 Managed asset class loader

Managed assets include the JavaScript, Java archive files, CSS, and other files that are added to the IBM BPM application during development. These assets are used to support the IBM BPM application.

To maintain versioning across applications and snapshots, a specific class loader was created. The class loader has two settings, one related to the number of active snapshots, and the second setting to the number of total class files in the system.

> **Note:** The internal configuration of the managed asset class loader changed in IBM BPM 8.5.0.1. The configuration values in your system before 8.5.0.1 must be increased after migrating to a version after 8.5.0.1 The new values can be 5 - 10 times larger than your previous setting.

During run time, a trace string can be set to help determine the current size. The output from the managed asset class loader that is shown in Example 6-5 is printed to the trace file. This output can be used for diagnostic purposes and to determine whether the maximum sizes must be increased. Starting in IBM BPM 8.5.7.0, the following trace string outputs the managed asset class loader sizes:

```
com.ibm.ws.classloader.*=all
```

Example 6-5 Diagnostic information about Managed Asset Class Loader size

```
com.lombardisoftware.server.core.ManagedAssetClassLoader
initial cache size={0}
com.lombardisoftware.server.core.ManagedAssetClassLoader
current classloaderMap size={1}
com.lombardisoftware.server.core.ManagedAssetClassLoader
initial resourceMapSize={2}
com.lombardisoftware.server.core.ManagedAssetClassLoader
current resourceMap.size()={3}
com.lombardisoftware.server.core.ManagedAssetClassLoader
current locationMap.size()={4}
```

When a specific asset is called, it is loaded into the cache and outputs the current `<classloader-resource-map-size>` size, as shown in Example 6-6.

Example 6-6 Trace entry with size of cache with the UUID of the asset loaded

```

wle          1 com.lombardisoftware.client.persistence.ManagedAsset getAssetData
Reading AssetData from cache: bcf85643-2095-4c84-9e1b-00921c053fa3. Cache size is
3
```

The current size should not regularly exceed the maximum size. Change the configuration to ensure that the application is not exceeding the cache size. Having this setting undersized is known to have significant performance problems.

For more information about managed asset tuning, see this website:

https://ibm.biz/BdsvfZ

6.3.2 Repository assets, snapshot, and branch size

The configuration snapshot and branch sizes are related to the number of applications and the active snapshots in the system. IBM BPM systems that feature many applications or many versions of the same application might need to increase the default values. If multiple versions of an application are present, it is a best practice to deactivate the snapshot when there are no more active instances and delete the snapshot.

When `com.lombardisoftware.server.ejb.persistence.versioning.BranchManager=all` trace is enabled, the trace file includes the message “`branches.size is 7`” when a new snapshot is loaded in memory.

Another trace message “`size is: 8. maxSize is: 7`” shows the current and maximum branch manager cache size. When the maximum size is regularly exceeded, the cache must be increased to avoid reloading commonly used snapshots. The branch size is seven, as shown in Example 6-7.

Example 6-7 Trace entry with size of branch manager

```
wle_versionin 1
com.lombardisoftware.server.ejb.persistence.versioning.BranchManager getContext
return branchContext
com.lombardisoftware.server.ejb.persistence.versioning.BranchContextImpl@84d8ee5c.
branches.size is 7
```

For a Process Center, the delta and concrete snapshots sizes are logged. The delta of four and concrete size of 1414 are two other parameters that can be configured, as shown in Example 6-8.

Example 6-8 Trace entry with concrete snapshot size for Process Center

```
wle_versionin 1
com.lombardisoftware.server.ejb.persistence.versioning.SnapshotContextHelper
isDeltaContextPreferred current context snapshot =
Snapshot.6cd8561c-66d2-4c3f-bb98-2bc0c3f5057cdelta size= 4 vs concrete size=1414
```

Persistent objects are the main assets for the application, including coaches, BPDs, users, and favorites. Depending on the number of assets in total across all applications, the default values might need to be increased. The cache for these objects is `<default-versioned-po-cache-size>`.

For more information about PO factory and repository configuration settings, see this website:

https://ibm.biz/BdsvfY

6.3.3 Event manager

The event manager is the engine that moves BPMN tasks from one step to the next. It also helps coordinate the load between nodes in the cluster. For more information about event manager and its tuning, see the following resources:

- Documentation on settings:

  https://ibm.biz/BdsvfS

- *IBM BPM Performance Tuning*, SG24-8216-00:

  http://www.redbooks.ibm.com/abstracts/sg248216.html

- Common problems and solutions with event manager:

  https://ibm.biz/Bdsvfm

Event manager history

The event manager (EM) is the core BPMN engine. It schedules and executes the work to be done in the system. The history of the EM can be recorded. Data analysis of this information can show patterns of when the system is used and which applications are used more often than others. For more information about EM history for analysis, see the following resources:

- https://ibm.biz/Bdsvfn
- https://ibm.biz/Bdsvfe

IBM Support might ask you to provide the following information to assist with problem determination:

- The TeamWorksConfiguration.running.xml file
- Export of LSW_EM_TASK and LSW_EM_INSTANCE tables

6.3.4 Instance and tasks

Troubleshooting tasks and instances feature several paths. A few cases are described in this section.

Finding a specific task or instance in the log files involves looking for entries in the log that is marked with specific text. Samples of regular expressions to find these entries in the log are shown in Example 6-9, Example 6-10, Example 6-11, Example 6-12 on page 102, Example 6-13 on page 102 and Example 6-14 on page 102. Different areas of the application report unique IDs in different formats.

Example 6-9 Find BPD instance ID: Accessing instance data

```
[11/17/15 10:19:24:380 PST] 0000e8d5 BPDInstanceDA >
com.lombardisoftware.bpd.runtime.engine.BPDInstanceDAO resumeInstance ENTRY id =
BpmnInstanceId(509)

regular expression: BpmnInstanceId[(]\d
```

Example 6-10 Find BPD Instance ID: User accessing task

```
[11/17/15 10:19:24:411 PST] 0000e8d5 BPDInstanceDA <
com.lombardisoftware.bpd.runtime.engine.BPDInstanceDAO resumeInstance RETURN
bpdInstanceId(false) = BPDInstance.509, instanceName(false) = IBM L2 - simple
enter data:509, bpdRef(false) = Ref[/BPD.23471e4f-c849-4a29-8595-2631d81f7900],
cachedBpdVersionId(false) = 0dc9c137-2c0a-49a2-bdaa-17b1ab86356d,

regular expression BPDInstance\.\d
```

Example 6-11 Find BPD Instance ID with a Task ID: Updating portal task index

```
[11/17/15 10:19:28:982 PST] 00000170 ProcessIndexB 1
com.ibm.bpm.search.process.spi.impl.ProcessIndexBatchHelper updateTaskIndexBatch
There are 1 for processing: [{taskId=565, instanceId=509}]

regular expressions: instanceId\=\d taskId\=\d
```

Example 6-12 Find BPMN Task ID: Accessing task data

```
[2/2/16 9:12:51:789 PST] 000049bf ExecutionCont 1
com.lombardisoftware.component.common.workflow.ExecutionContextHelper
getExecutionContextOfTask BEGIN: loadExecutionContext(), taskId = Task.107

[2/2/16 9:12:59:295 PST] 0000eb55 Engine        1
com.lombardisoftware.server.scheduler.Engine execute  Executing TASK_ID = 164

regular expression taskid = Task\.\d and Executing TASK_ID \= \d
```

Example 6-13 Finding User IDs and Names

```
[2/2/16 9:12:45:830 PST] 000049bf UserCore      <
com.lombardisoftware.server.core.UserCore getOrCreateUserInfoObject RETURN
[id=User.9 fullname=deadmin username=deadmin data={Full Name=deadmin}]

regular expresssion User\.\d and username\=.

[2/2/16 9:12:51:797 PST] 000049bf TaskCore      >
com.lombardisoftware.server.core.TaskCore canCollaborate ENTRY taskId = Task.107,
userId = 1

regular expression: userId \= \d

[2/2/16 9:12:59:296 PST] 0000eb55 UserCore      >
com.lombardisoftware.server.core.UserCore getOrCreateUserInfoObject ENTRY User
name = deadmin, isCheckForUpdates = false

regular expression: User name = .
```

Example 6-14 Event Manager scheduling with instance ID

```
[2/2/16 9:12:53:988 PST] 000001c1 TaskManager   >
com.lombardisoftware.server.scheduler.TaskManager scheduleTasks ENTRY
[com.lombardisoftware.server.scheduler.ScheduledTask@5fa6de05
 - scheduledTime: 1454433173986 = Tue Feb 02 09:12:53 PST 2016
 - repeatString: null
 - execClassName:
com.lombardisoftware.bpd.runtime.engine.quartz.DbNotificationBpdTask
 - arguments: 107;110
 - description: Notify BPD 107 of notification
 - queueId: -101
 - blackoutCalendarId: 1
 - blackoutBehavior: BlackoutBehavior(2, EXECUTE_ONE_AFTER_BLACKOUT)
 - inClosingTransaction: false
 - idInDB: null
 - reuseStrategy: 2
 - keywords: List start
   - bpd.guid.guid:df35784cf6a4aae8:61feb82d:1527afab2ff:312b
   - bpd.instance.107
   - bpd.group.BPDNotification
    List end] true

regular expression: bpd\.instance\.\d and Notify BPD \d
```

BPD and service looping

The best defense for preventing looping is to review any place that uses a loop and enabling the loop detection configuration. It is recommended to enable the loop detection to prevent a run away service from disrupting the system. The loop detection can be set to interrupt and fail the instance on which the looping is occurring.

The warning and termination is output that is to `SystemOut.log` for easy monitoring in your system. For more information about how to configure and log message format, see the IBM Knowledge Center. Services that exceed the loop detection time should be sent back to the Application Development team for review. For more information, see this website:

https://ibm.biz/BdsmUj

For more information about troubleshooting, see this website:

http://www.ibm.com/support/docview.wss?uid=swg21622584

Updating instance data manually

There are times where the task data must be updated to a correct value to recover from a failed instance. Several methods are available to correct the values in the execution context. The common method is the REST API method `setData`. For more information, see this website:

https://ibm.biz/BdsvMA

Handling failed instances

Instances fail for several reasons. Data that supplied in the previous step is missing or corrupted, an instance was migrated and is orphaned, an outside integration service failed to complete (which failed the instance), and others. Some recovery methods are described next.

Resume

First, attempt to resume the instance by using the REST API or Process Designer. Often, if a time-out occurred or an integration was unavailable, resuming the instance allows it to proceed. For more information about resuming an instance with the REST API, see this website:

https://ibm.biz/BdsvMu

Move token REST API

For instances that fail migrating in-flight instances to a new snapshot, the BPMN tokens might need to be moved. Consult with your development team about where tokens should be moved to recover the instance. For more information about moving tokens, see this website:

http://www.ibm.com/developerworks/bpm/bpmjournal/1409_ahuja/1409_ahuja.html

To reduce issues with migrating in-flight instances, work with your development team to ensure adequate coverage in the testing cycles. For more information about handling in-flight instances, see this website:

https://ibm.biz/BdsvSX

Delete and re-create

Some IBM BPM applications are designed so that if there is a problem with an instance, it can easily be re-created with the current data and the process resumes. This process is done by using an operations administrative tool that is created by the Application Operations team.

6.3.5 Under Cover Agents

Failures that are related to Under Cover Agents (UCAs) relate to application development and expectations of how and when a message is expected to be received and used. For more information about resolving these errors and known defects, see the TechNote *Common steps to resolving problems with undercover agents in IBM Business Process Manager (BPM)*, which is available at this website:

http://www.ibm.com/support/docview.wss?uid=swg27041883

Failed UCA entries

By default, a UCA attempts to run five times. If the last attempt fails, the UCA record is stored as a February 1, 2099 date in the event manager. These attempts can be manually replayed at a later time. At times, events cannot be successfully executed and fail. Manual deletion might be necessary. For more information, see this website:

https://ibm.biz/BdsvSi

6.4 Operations with Process Admin console

The Process Admin console features the following useful tools for use by the IBM BPM Operations team:

- Process monitor
- Event manager
- Instrumentation
- Creating alerts

These features are described next.

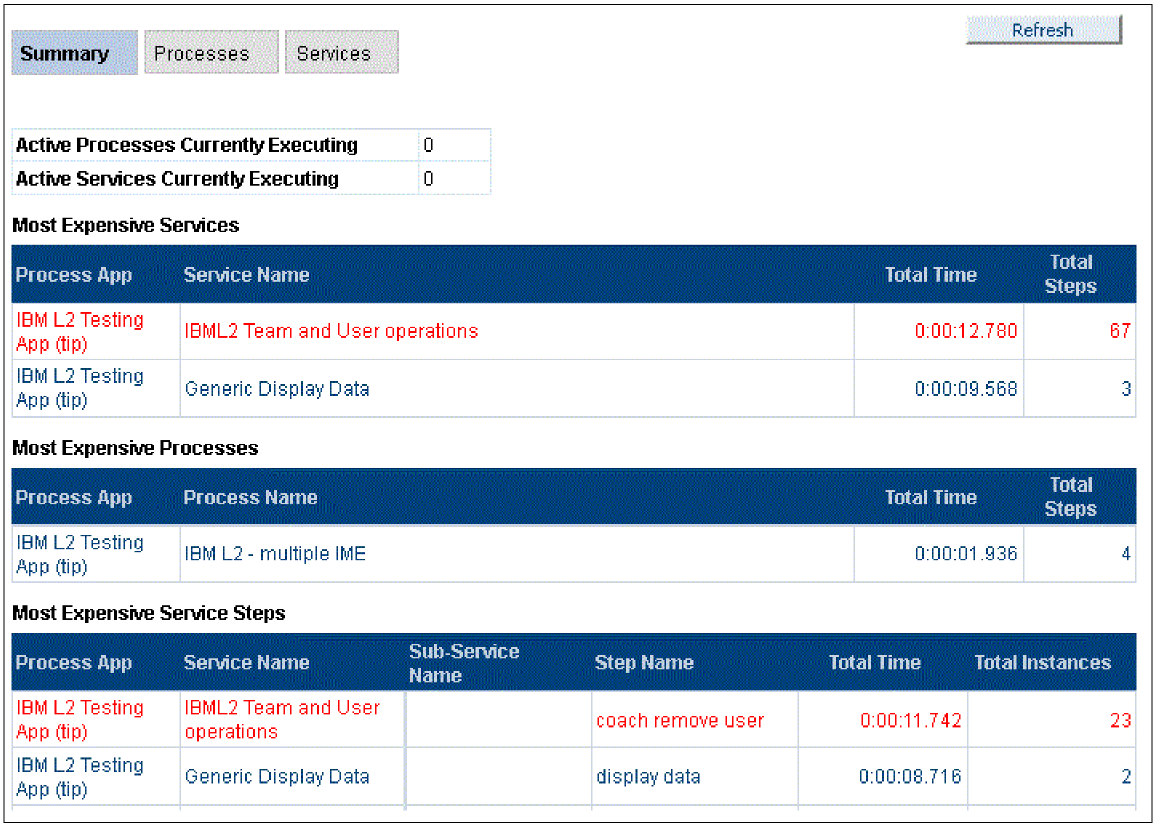
6.4.1 Process monitor

This dashboard displays the top running services and processes by using two metrics: Used are time and number of steps. Time refers to how long a specific BPD or service is running. Steps are the number of iterations the same process or service ran, as listed in Table 6-1.

Table 6-1 Problem determination with Process Monitor

Service or process	Low step count	High Step Count
Low Execution Time	Normally running application code.	A loop exists within the application code. Review FOR, WHILE, and diagram flow loops.
High Execution Time	Activity might be processing large amounts of data. Discuss with Application Development team to determine whether this activity is normal. Activity might be hung waiting on a response from an outside system.	Combination of application looping and outside connections hanging or not being responsive preventing the activity to complete.

Items that are in red Figure 6-2 are considered high relative to the other services and processes running.



Active Processes Currently Executing	0
Active Services Currently Executing	0

Most Expensive Services

Process App	Service Name	Total Time	Total Steps
IBM L2 Testing App (tip)	IBML2 Team and User operations	0:00:12.780	67
IBM L2 Testing App (tip)	Generic Display Data	0:00:09.568	3

Most Expensive Processes

Process App	Process Name	Total Time	Total Steps
IBM L2 Testing App (tip)	IBM L2 - multiple IME	0:00:01.936	4

Most Expensive Service Steps

Process App	Service Name	Sub-Service Name	Step Name	Total Time	Total Instances
IBM L2 Testing App (tip)	IBML2 Team and User operations		coach remove user	0:00:11.742	23
IBM L2 Testing App (tip)	Generic Display Data		display data	0:00:08.716	2

Figure 6-2 Process monitor

If the system has higher than expected CPU or memory consumption, check the process monitor. Code that was running for a longer than normal time can be the cause of the issue. You can drill down into these elements to gain more information about the step name, and possibly the instance. Traces also can be enabled to find the specific element and a root cause. The long-running or much executed code then can be sent back to the application team for review.

The process monitor can be monitored by using a ProcessMonitor MBean. For more information, see this website:

`https://ibm.biz/BdsvSP`

Most services and processes can be halted from the Process Admin console window. Open the process monitor and click the **Services** tab. Then, click the **Stop Service** button. An attempt is made to cancel the runaway service or process. Stand-alone services that are exposed and not tied to a process instance cannot be stopped. The only way to resolve a task-less service is to restart the JVMs on which the problem is occurring. For more information about recovery options, see this website:

`http://www.ibm.com/support/docview.wss?uid=swg21622584`

6.4.2 Event Manager

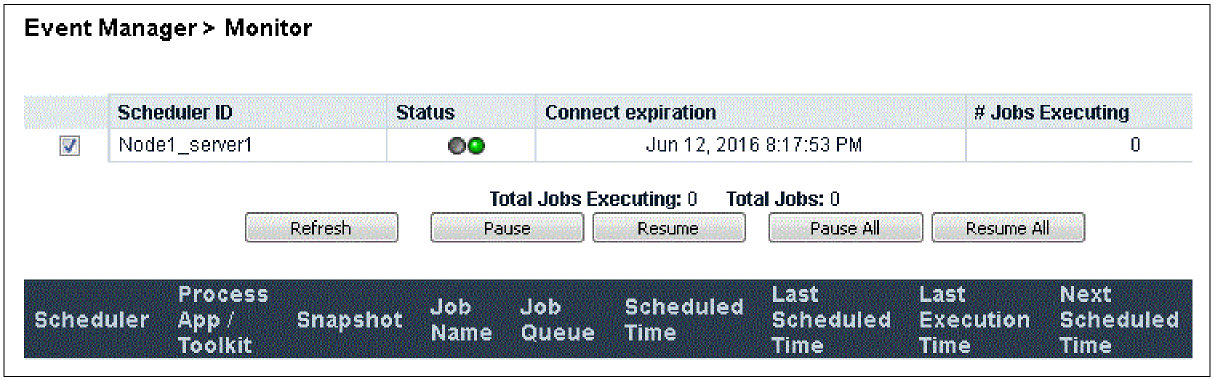
The Event Manager window shows all available event managers in the configuration, the currently running tasks, and the pending tasks. This information can be useful to see whether all nodes in the cluster are online and processing. You can also pause individual event managers for troubleshooting, if needed.

Another important area is the blackout settings. These settings are used to schedule the event manager to stop at a specified time, which is useful for maintenance periods and installing new snapshots.

For more information about the event manager and blackout periods, see the following product documentation:

https://ibm.biz/BdsvSM

The Event Manager with no scheduled work and only one node configured is shown in Figure 6-3.



Event Manager > Monitor

	Scheduler ID	Status	Connect expiration	# Jobs Executing
☑	Node1_server1	●●	Jun 12, 2016 8:17:53 PM	0

Total Jobs Executing: 0 **Total Jobs:** 0

Scheduler	Process App / Toolkit	Snapshot	Job Name	Job Queue	Scheduled Time	Last Scheduled Time	Last Execution Time	Next Scheduled Time

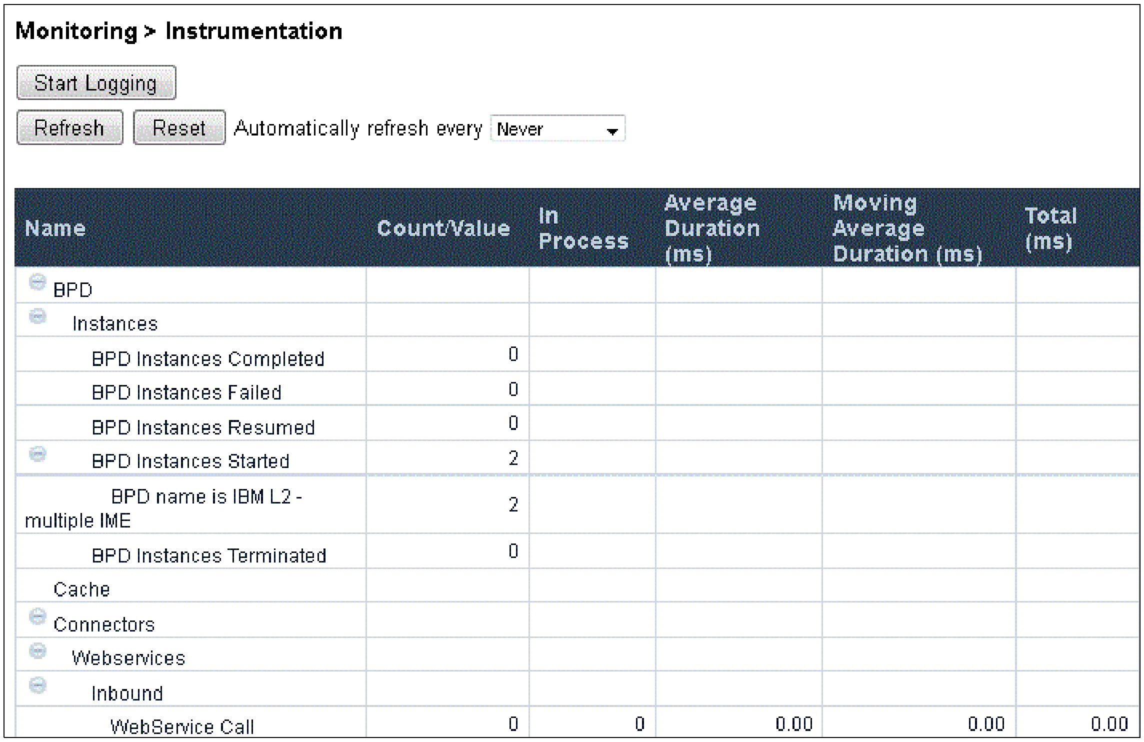
Figure 6-3 Event Manager

6.4.3 Instrumentation

The instrumentation window is a running metric of various caches and objects. It is reset after a process server (AppTarget JVM) restart. One useful metric is the cache misses. This metric helps with tuning the IBM BPM caches. Too many misses cause extra strain on the system.

The output of this page can be saved as XML. The logging of instrumentation produces a data file and can be extracted for analysis, as described in 6.2.12, “Instrumentation within BPMN” on page 98. The ratio of misses-to-hits should be less than 5%. If the ratio is greater, tuning of the IBM BPM caches is needed.

Figure 6-4 on page 107 shows a small sample of the instrumentation page (more metrics are available).



Monitoring > Instrumentation

Automatically refresh every Never

Name	Count/Value	In Process	Average Duration (ms)	Moving Average Duration (ms)	Total (ms)
⊖ BPD					
⊖ Instances					
BPD Instances Completed	0				
BPD Instances Failed	0				
BPD Instances Resumed	0				
⊖ BPD Instances Started	2				
BPD name is IBM L2 - multiple IME	2				
BPD Instances Terminated	0				
Cache					
⊖ Connectors					
⊖ Webservices					
⊖ Inbound					
WebService Call	0	0	0.00	0.00	0.00

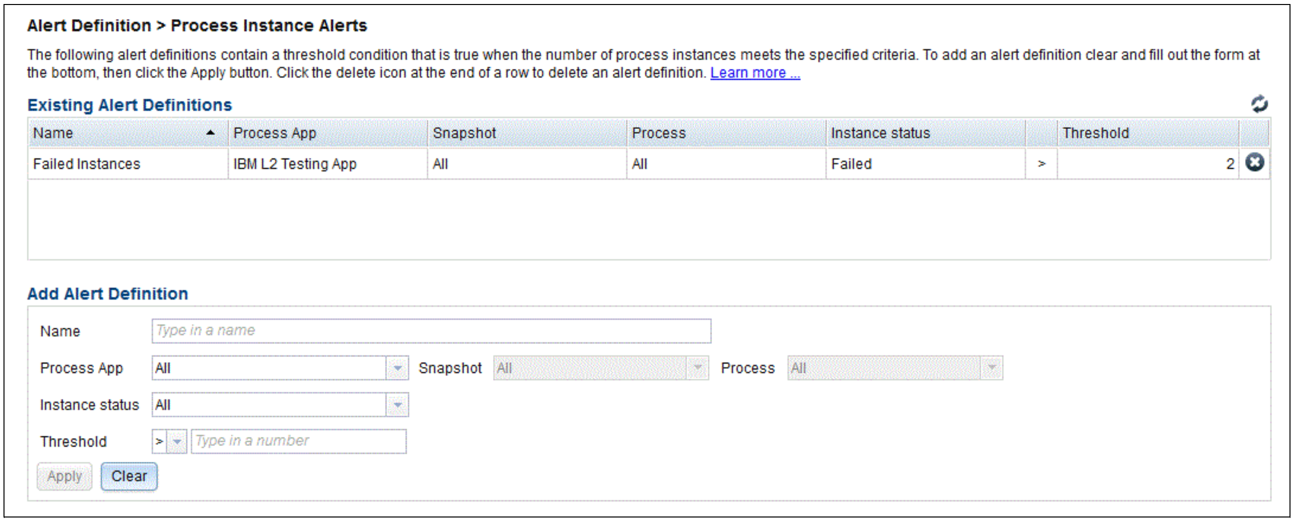
Figure 6-4 Sample instrumentation page

6.4.4 Creating alerts

Instance and task alerts are a new feature in IBM BPM 8.5.7.0. This feature allows the Operations team to monitor information about specific applications or processes in the system. For more information and how to configure this feature, see the following product documentation:

https://ibm.biz/BdsvSG

An example of an instance alert is shown in Figure 6-5. It is configured for failed instances of all processes in a specific application with a threshold of two.



Alert Definition > Process Instance Alerts

The following alert definitions contain a threshold condition that is true when the number of process instances meets the specified criteria. To add an alert definition clear and fill out the form at the bottom, then click the Apply button. Click the delete icon at the end of a row to delete an alert definition. Learn more ...

Existing Alert Definitions

Name ▲	Process App	Snapshot	Process	Instance status		Threshold	
Failed Instances	IBM L2 Testing App	All	All	Failed	>	2	⊗

Add Alert Definition

Name: Type in a name

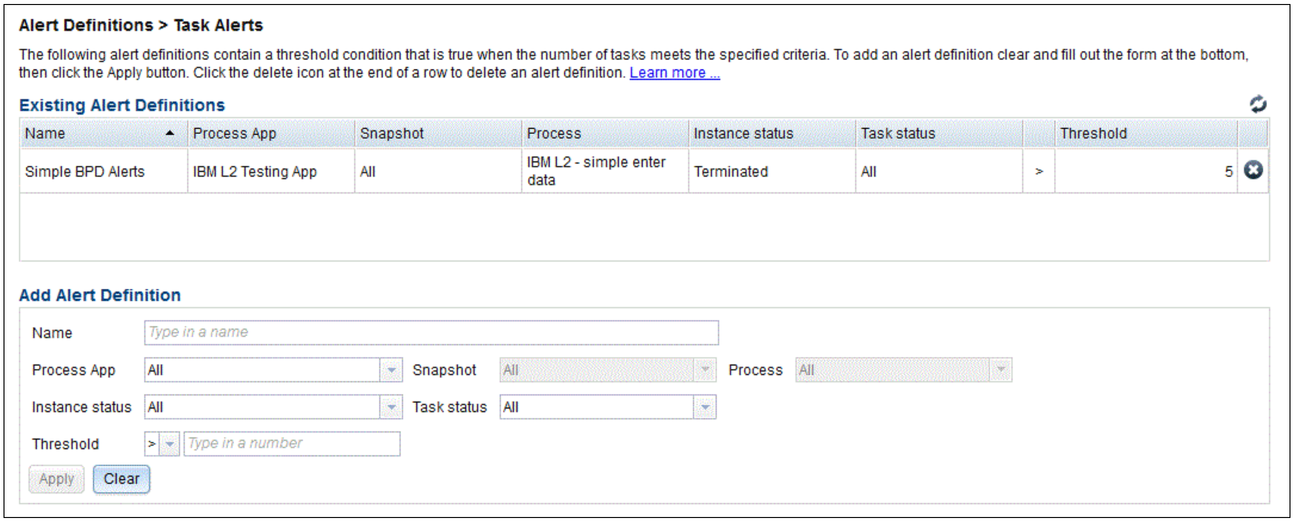
Process App: All   Snapshot: All   Process: All

Instance status: All

Threshold: >   Type in a number

Figure 6-5 Sample instant alert

An example task alert is shown in Figure 6-6. This alert is triggered if a specific process is ended more than five times.


Alert Definitions > Task Alerts

The following alert definitions contain a threshold condition that is true when the number of tasks meets the specified criteria. To add an alert definition clear and fill out the form at the bottom, then click the Apply button. Click the delete icon at the end of a row to delete an alert definition. Learn more ...

Existing Alert Definitions

Name ▲	Process App	Snapshot	Process	Instance status	Task status		Threshold	
Simple BPD Alerts	IBM L2 Testing App	All	IBM L2 - simple enter data	Terminated	All	>	5	⊗

Add Alert Definition

Name: Type in a name

Process App: All — Snapshot: All — Process: All

Instance status: All — Task status: All

Threshold: > — Type in a number

Figure 6-6 Sample task alert

6.5 Recovery modes and transactions

Recovery modes and transactions are described in this section.

6.5.1 Starting the server in recovery mode

When a WebSphere Application Server with active transactions in progress starts after a failure, the transaction service uses recovery logs to complete a recovery process. These logs, which each transactional resource maintains, are used to recover any in-doubt transactions and return the system to a self-consistent state. This recovery process begins when all of the necessary subsystems within the WebSphere Application Server are available during a server startup.

If the application server is not restarted in recovery mode, the application server can start accepting new work when the server is ready, which might occur before the recovery work is complete. This configuration might be acceptable in many cases. However, you might want to ensure that the recovery is performed and completed as seemlessly and quickly as possible after an outage runs. In these cases, you should first start your WebSphere Application Servers in recovery mode.

6.5.2 Resolving in-doubt transactions

Transactions can become stuck in the in-doubt state indefinitely because of an exceptional circumstance. When a transaction becomes in-doubt, it must be committed or rolled back so that normal processing by the affected transactional systems can continue.

To resolve the in-doubt transaction, use the WebSphere Application Server admin console Transaction Management panels by clicking **Servers** → **Application servers** → **Select a server** → **Container Services** → **Transaction Service**. Then, click **Runtime tab** → **Imported prepared transactions** → **Review**.

If the transaction identity is displayed in the resulting panel, you can commit or roll back the transaction. Choose the option to commit or roll back the transaction as required to resolve the in-doubt transaction.

6.6 BPEL problem determination

BPEL problem determination is described in this section.

6.6.1 Failed events

If an event fails in IBM BPM, it is persisted by the *Failed Event Subsystem*, which is also known as the *Recovery Subsystem*. You can use the Failed Event Manager to monitor failed events by using a GUI in the WebSphere administrative console, or through the command line by using wsadmin commands.

The following types of failed events can occur:

- Service Component Architecture (SCA) events
- WebSphere MQ events
- Java Message Service (JMS) events
- Business Process Choreographer events
- Business Flow Manager hold queue events

SCA failed events

In the context of SCA, a failed event is a request or response message that was received by an Advanced Integration service application and could not be delivered.

Java Message Service failed events

In the context of Java Message Service (JMS), a failed event is generated when a JMS data binding or function selector operation fails, or when an exception occurs in a downstream component and is thrown back to the JMS export or import. The Recovery service handles failed events from JMS data bindings and IBM MQ JMS data bindings.

WebSphere MQ failed events

In the context of 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.

Business Process Choreographer stopped, ended, and failed events

In the context of Business Process Choreographer, exceptions can occur that cause an activity to stop or the process instance to fail. A failed event is generated when a long-running BPEL process fails and one of the following events occurs:

- The process instance enters the Failed or Terminated state.
- An activity enters the Stopped state.

Business Flow Manager Hold queue events

You can use the failed event manager to manage navigation messages that are stored in the Business Flow Manager Hold queue. A navigation message might be stored in the hold queue if one of the following conditions exists:

- An infrastructure, such as a database, was unavailable.
- The message is damaged.

6.6.2 Security considerations for the recovery subsystem

It is important to understand how role-based access and user identity affect the recovery subsystem.

Role-based access for the Failed Event Manager

The Failed Event Manager uses role-based access control for the failed event's data and tasks. Only the administrator and operator roles are authorized to perform tasks within the Failed Event Manager. Users that are logged in as a user by using the administrator or operator role can view all data that is associated with failed events and perform all tasks.

Event identity and user permissions

A failed event encapsulates information about the user who originated the request. If a failed event is resubmitted, its identity information is updated to reflect the user who resubmitted the event. The administrator or operator role user must be given permissions to the downstream components that are required to process the event.

6.6.3 Hold and retention queue

If there are problems processing messages that are internal to the business process engine, these messages are moved to a retention queue. The system attempts to continue to process messages. If a subsequent message is successfully processed, the messages on the retention queue are submitted for reprocessing. If the same message is placed on the retention queue five times, it is then placed in the hold queue.

Replaying messages from the retention and hold queues

The administering messages on the retention queue, hold queue, or both and replaying these messages can be part of a recovery procedure. Checking and replaying messages can be done by using scripting or the administrative console.

6.6.4 Event sequencing

Event sequencing is described in this section.

Lock table

For the Event Sequencing Component to control the flow of messages, it must maintain a list of the messages that are locked. To do maintain this list, IBM BPM uses the `PERSISTENTLOCK` table that in the common database.

esAdmin tool

The IBM BPM administrator can use the esAdmin tool to maintain the locks that are active within the system. The tool allows the user to list locks by module or component, or it can list all locks in the system. The tool also allows users to unlock a message or delete messages from the lock queue. For more information, see this website:

https://ibm.biz/BdsvS7

Troubleshooting

You can troubleshoot event sequencing by adding the following line to the Runtime Diagnostic Trace Service:

```
com.ibm.wbiserver.sequencing.*=all
```

This line outputs to the `trace.log` the work that is done by the event sequencing components. It also shows work that is run by these components and any failures that result.

As more messages are received, the event sequencing component attempts to acquire a lock for each additional message so that it can be processed. However, the lock cannot be obtained because the previous message is yet completed.

These messages are saved for processing in the order they were received after the current message is processed. By using the esAdmin tool, you see these pending messages for that lock ID, module, and component.

For more information, see this website:

https://ibm.biz/Bdsvvx

6.7 Preparing for fire drills

After the IBM BPM system is live and deployed to production, instances can occur in which the system is down or unresponsive or data is missing. Before final production launch, the Operations team needs a fire drill handbook that must be practiced. Administrator contacts and escalation procedures for the core parts of the IBM BPM system must be available. Phone and pager lists for weekend and off-shift support also must be reviewed.

6.7.1 Key contacts

Have known document or processes in place to contact the individuals that are responsible for each environment. This contact information might include email, phone numbers, or other methods. Depending on your company roles and responsibilities, this point of contact can be one or more people.

The main contacts are the following roles when a problem occurs:

- IBM BPM administrator
- WebSphere administrator
- OS and server administrator
- Database administrators

Secondary component contacts might be needed to determine why a resource that IBM BPM is integrated with is not responding and is affecting IBM BPM. The Application team might be needed to determine how best to recover failed instances or answer questions about the application design.

The following Secondary Component Contacts are needed:

- Lead IBM BPM Developer
- Network administrator
- Secondary dependent systems administrator: LDAP, systems of record, and various integrated systems

6.7.2 Emergency actions

You might need to perform specific tasks during an outage. Practice the following exercises so that you are ready when a problem occurs:

- Review internal policies and procedures for making emergency changes to a production system.
- Set trace parameters in the WebSphere console and send a `.zip` file of files to IBM for analysis:
  - IBM BPM MustGather, with sub component areas:

    `http://www.ibm.com/support/docview.wss?uid=swg21569731`
  - Sending Data to IBM Support:

    `http://www.ibm.com/support/docview.wss?uid=swg21153852`
- Delete instances and tasks by using wsadmin scripts. For more information, see 4.3.2, "Instances in Process Server" on page 59.
- Be familiar with running **`SELECT`**, **`UPDATE`**, and **`DELETE SQL`** commands.
- Move BPMN tokens. For more information, see "Move token REST API" on page 103.
- Recover the failed event manager.
- Change IBM BPM configuration files. For more information, see this website:

  `https://ibm.biz/Bdsvvs`
- Install a single iFix. For more information, see this website:

  `https://ibm.biz/Bdsvvi`
- Modify a load balancer in a maintenance mode so users are not directed to IBM BPM.
- Roll back to a known full backup.

Related publications

The publications that are listed in this section are considered particularly suitable for a more detailed discussion of the topics that are covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide more information about the topics in this document. Note that some publications that are referenced in this list might be available in softcopy only:

- *IBM BPM Performance Tuning*, SG24-8216
- *Scaling BPM Adoption From Project to Program with IBM Business Process Manager*, SG24-7973
- *Business Process Management Design Guide: Using IBM Business Process Manager*, SG24-8282
- *IBM Business Process Manager V8.5 Performance Tuning and Best Practice*, SG24-8216
- *Problem Determination Across Multiple WebSphere Products AIX Platform*, SG24-7043

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, draft, and other materials at the following website:

ibm.com/redbooks

Online resources

The following websites also are relevant as further information sources:

- Accelerated Value Program (AVP):

  http://www.ibm.com/software/rational/support/tsas/
- IBM BPM Knowledge Center:

  http://www.ibm.com/support/knowledgecenter/en/SSFPJS
- IBM developerWorks article about BPM Answers and Questions:

  https://ibm.biz/BdsMRQ
- IBM developerWorks articles:

  http://www.ibm.com/developerworks/
- IBM BPM MustGather:

  http://www.ibm.com/support/docview.wss?uid=swg21569731
- IBM WebSphere systems requirements for each IBM BPM release:

  http://www.ibm.com/support/docview.wss?uid=swg27023007
- Security bulletins:

  http://www.ibm.com/software/support/einfo.html

- System requirements page for the supported operating systems for your BPM versions:

  http://www.ibm.com/support/docview.wss?uid=swg27023007
- IBM Business Process Manager running in a virtual machine:

  http://www.ibm.com/support/docview.wss?uid=swg21574113
- LDAP directory server that IBM WebSphere supports:

  http://www.ibm.com/support/docview.wss?uid=swg27036471
- IBM DB2 JDBC Drivers:

  http://www.ibm.com/support/docview.wss?uid=swg21363866
- Microsoft SQL Server:

  https://msdn.microsoft.com/en-us/library/mt484311%28v=sql.1%29.aspx
- Oracle Database download list:

  http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html
- Dr. Debug, helpful tips to resolving problems with BPM:

  https://ibm.biz/BdsSUD
- IBM Support TV how-to videos about specific data collection or configuration options for IBM Products:

  https://www.youtube.com/channel/UC9K3qy72l9D27vPeKWoUq7Q
- IBM Technical Support:

  https://www-947.ibm.com/support/entry/myportal/support
- IBM Services:

  https://ibm.biz/BdsSUF
- IBM On Demand Consulting:

  http://www.ibm.com/developerworks/middleware/services/odc/
- IBM Business Process Manager security overview:

  https://ibm.biz/BdsSi2
- Managing access to the Process Center Repository:

  https://ibm.biz/BdsSiz
- Configuration properties for Process Portal action policies:

  https://ibm.biz/BdsSiq
- Modifying the Process Admin Console setting and behavior:

  https://ibm.biz/BdsSif
- Creating a secure environment:

  https://ibm.biz/BdsSiP

Help from IBM

IBM Support and downloads:

ibm.com/support

IBM Global Services:

ibm.com/services

Redbooks

IBM Business Process Manager Operations Guide

(0.2”spine)
0.17”<->0.473”
90<->249 pages

SG24-8356-00

ISBN 0738442291

Printed in U.S.A.