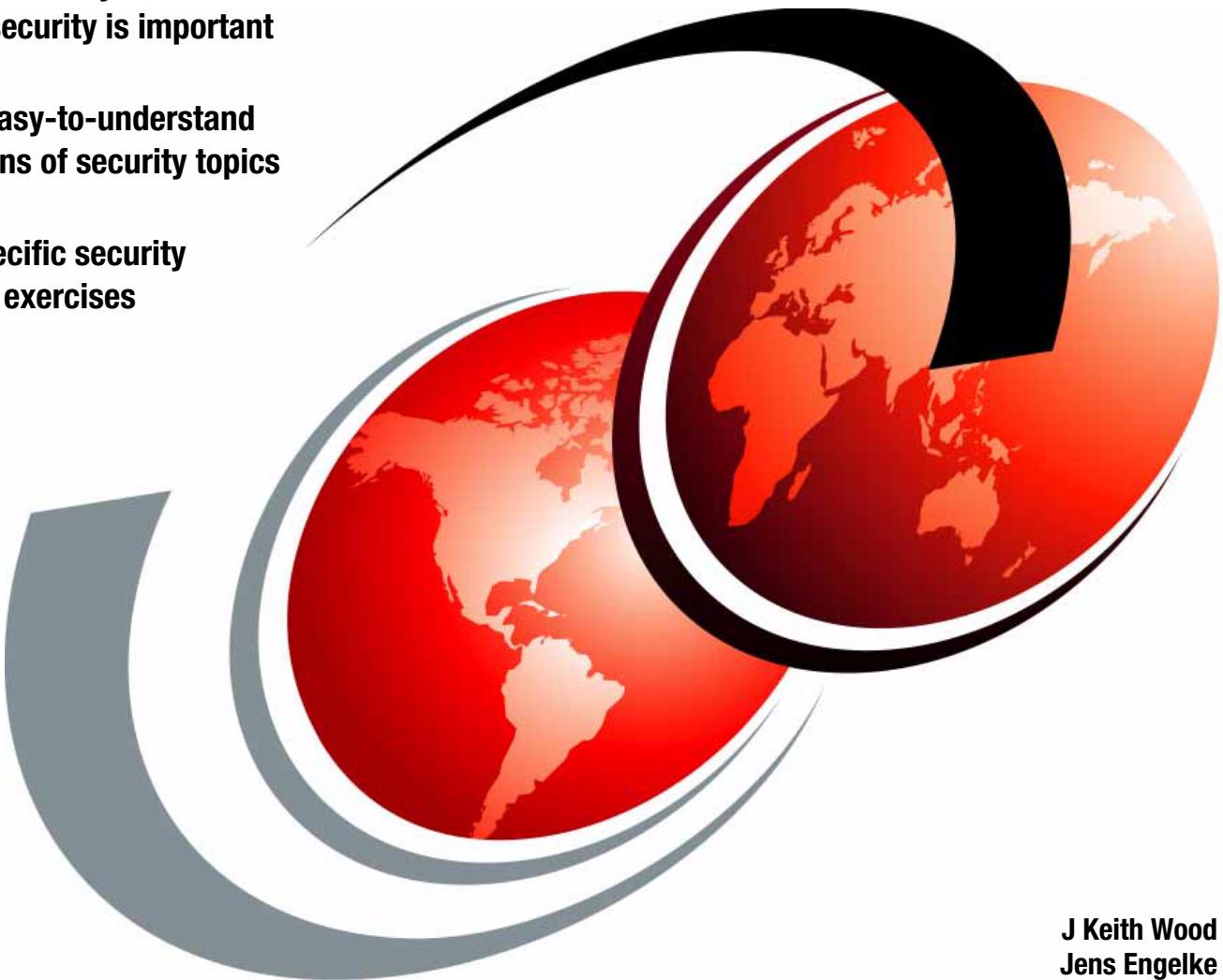


IBM Business Process Manager Security Concepts and Guidance

Demonstrates why Business Process Manager security is important

Includes easy-to-understand explanations of security topics

Covers specific security hardening exercises



J Keith Wood
Jens Engelke

Redbooks



International Technical Support Organization

**IBM Business Process Manager Security: Concepts
and Guidance**

September 2012

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

First Edition (September 2012)

This edition applies to Version 7, Release 5, Modification 1 of IBM Business Process Manager.

© Copyright International Business Machines Corporation 2012. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Noticesv
Trademarksvi
Prefacevii
The team who wrote this bookvii
Now you can become a published author, too!viii
Comments welcomeviii
Stay connected to IBM Redbooksviii
Chapter 1. Why Business Process Manager security is important	1
1.1 BPM is your corporate DNA	2
1.1.1 Business Process Manager users have access	3
1.1.2 Business Process Manager has unique security considerations	4
1.2 Basic concepts	5
1.2.1 Business Process Manager	5
1.2.2 WebSphere Application Server	8
1.2.3 Business Process Manager administration tools	11
1.2.4 Installation options	11
1.3 Encryption, SSL, and certificates	13
1.3.1 Encryption	13
1.3.2 Symmetric and asymmetric keys	14
1.3.3 SSL and digital certificates	16
1.3.4 Certificate authorities	18
Chapter 2. Installation	21
2.1 Business Process Manager and WebSphere Application Server topologies	22
2.1.1 Basic concepts	22
2.1.2 Complex realities	23
2.2 Common security holes	25
2.2.1 Faith in firewalls	25
2.2.2 Failure to use SSL between BPM and database server	26
2.2.3 Failure to encrypt data at rest	26
2.2.4 Failure to use SSL between Process Center and Process Server	32
2.2.5 Overuse of default Business Process Manager accounts	33
2.2.6 Overuse of trust in certificate authorities	36
Chapter 3. Authentication: Who has access	41
3.1 Subjects and Principals	42
3.2 WebSphere user registry	42
3.2.1 Flat-file repositories	44
3.2.2 LDAP repositories	48
3.2.3 Custom software repository	52
3.2.4 Federated repositories	53
3.3 Common security holes	56
3.3.1 Weak password policies	56
3.3.2 Failure to change default passwords	57
3.3.3 Faith in firewalls	58
3.3.4 Insecure LDAP connections	59
3.3.5 Insecure SSO solutions	61

Chapter 4. Authorization: Access to what	65
4.1 Groups versus roles	66
4.2 Grouping mechanisms	66
4.2.1 LDAP groups	67
4.2.2 VMM security groups	70
4.2.3 Process Admin Console and private groups	75
4.2.4 Process Designer swimlanes and participant groups	81
4.2.5 Mapping roles to groups	83
4.2.6 Review and summary	90
4.3 Administrative access	92
4.3.1 Granting access to Process Designer	93
4.3.2 Review and summary	103
4.4 Instance-based authorization	106
4.5 Common security holes	107
4.5.1 Overuse of administrator privileges	107
4.5.2 Failure to map participant groups	108
4.5.3 Overpopulation of groups	110
4.5.4 Overuse of tw_authors, tw_admins	110
4.5.5 Faith in firewalls	111
Chapter 5. Integration: Working with others	113
5.1 Business Process Manager Standard Edition versus Advanced Edition	114
5.2 Business Process Manager Standard Edition outbound web services	114
5.2.1 Using Web Service Integration	115
5.2.2 Using SOAP Integration	127
5.2.3 Using Java Integration	129
5.3 Business Process Manager Standard Edition inbound web services	132
5.3.1 Steps to create inbound web service	133
5.3.2 Review and summary	142
5.3.3 Securing the inbound web service	143
5.4 Business Process Manager Advanced Edition web services options	144
5.5 Common security holes	145
5.5.1 Failure to secure web services passwords	145
5.5.2 Faith in firewalls	146
Related publications	147
IBM Redbooks publications	147
Other publications	147
Help from IBM	148

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

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

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

Any references in this information to non-IBM 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 supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

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

Blueworks Live™

DataPower®

DB2®

Domino®

IBM®

Lombardi Teamworks®

Lotus®

Redbooks®

Redbooks (logo) ®

Teamworks®

Tivoli®

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 information about security concerning an organization's business process management (BPM) program, about common security holes that often occur in this field, and describes techniques for rectifying these holes. This book documents preferred practices and common security hardening exercises that you can use to achieve a reasonably well-secured BPM installation.

Many of the practices described in this book apply equally to generic Java Platform and Enterprise Edition (J2EE) applications, as well as to BPM. However, it focuses on aspects that typically do not receive adequate consideration in actual practice. Also, it addresses equally the BPM Standard and BPM Advanced Editions, although there are topics inherent in BPM Advanced that we considered to be out of scope for this book.

This book is not meant as a technical deep-dive into any one topic, technology, or philosophy. IBM offers a variety of training and consulting services that can help you to understand and evaluate the implications of this book's topic in your own organization.

The team who wrote this book

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

J Keith Wood is a Systems Architect and Security Team Lead in the IBM Business Process Manager services organization, based in Austin, Texas. After 20+ years as a software developer and solutions architect, he now consults with Business Process Management customers to promote more secured installations and environments. Before joining IBM, Keith consulted with a wide variety of clients in the financial services, telecommunications, and retail sectors.

Jens Engelke is a Senior Accredited IT Specialist in the IBM Business Process Manager development organization, Böblingen, Germany. After multiple years of software services in the security and integration area, he now develops new features and functions for the BPM product. Before joining the development organization, Jens worked in the IBM internal IT organization integrating third-party packaged applications in the IBM infrastructure.

Special thanks to:

- ▶ Martin Lansche, Consulting IT Specialist, IBM Application and Integration Middleware Software

This project was managed by:

- ▶ Lisa Dyer, Program Manager, IBM Business Process Management and Decision Management
- ▶ Martin Keen, IBM Redbooks Project Leader

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>



Why Business Process Manager security is important

To provide a common foundation for discussion, and more importantly, to help us build a shared mental framework for understanding the more complex security implications, we will spend a few pages describing basic Business Process Manager and WebSphere® terms and concepts.

This chapter contains the following sections:

- ▶ BPM is your corporate DNA
- ▶ Basic concepts
- ▶ Encryption, SSL, and certificates

1.1 BPM is your corporate DNA

Before we get started with the details, it is worth considering the fundamental need for securing your BPM systems. First of all, we acknowledge that IBM Business Process Manager is based upon Java 2 Enterprise (J2E) technologies, and is delivered largely through HTTP protocols, and so it therefore has the same security requirements as any other J2E enterprise-ready application. Authentication, authorization and protection of sensitive data are all topics that are common to any J2E application, and so many casual observers may stop their inquiry there.

However, Business Process Manager—in many ways—is not just another J2E application. When one looks at an organization's existing software systems, one typically finds applications that are single-purpose built (Figure 1-1). The successful hacking of these applications could certainly expose the process' data, but it is hard to conceive of how such a breach might expose the actual business steps, decision points, or overall operational strategy of a department's business functions.

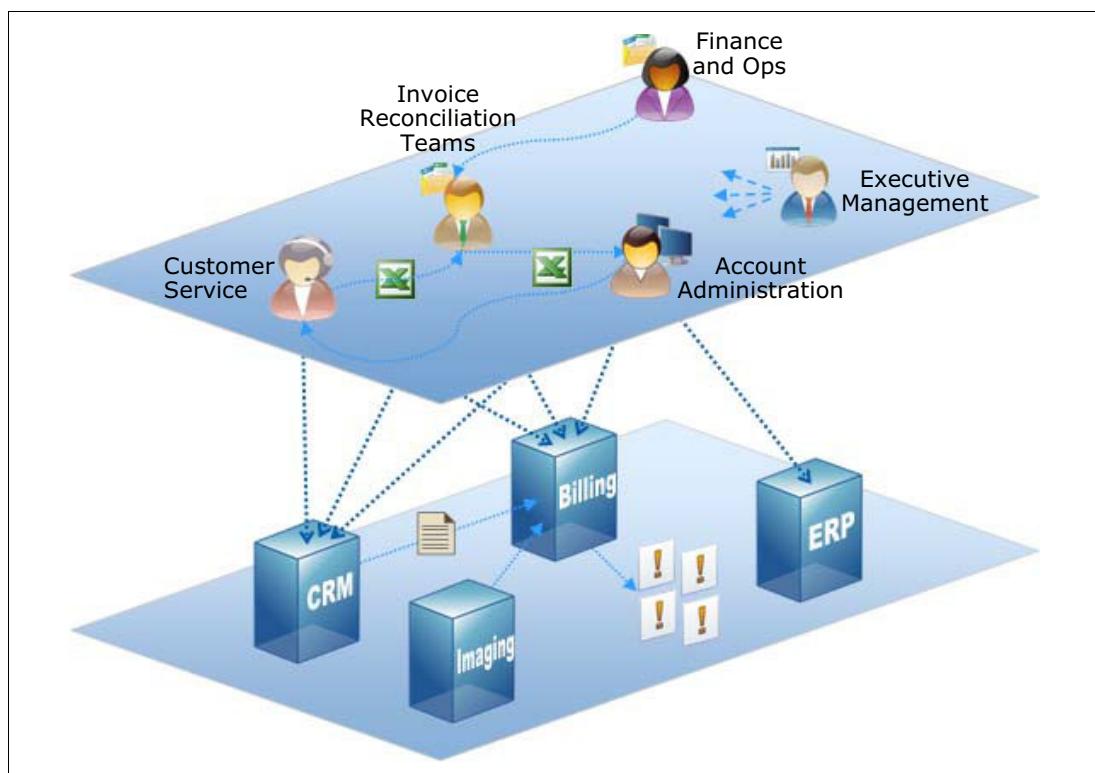


Figure 1-1 Single-purpose applications

Not true of BPM. BPM encapsulates more than just a process' data. BPM process applications capture the very essence and details of a department's way of doing business. IBM Business Process Manager paints easy-to-understand flowcharts of process steps, which employee groups are entitled to execute particular steps, the decision points, and details of how those decisions are evaluated (Figure 1-2 on page 3).

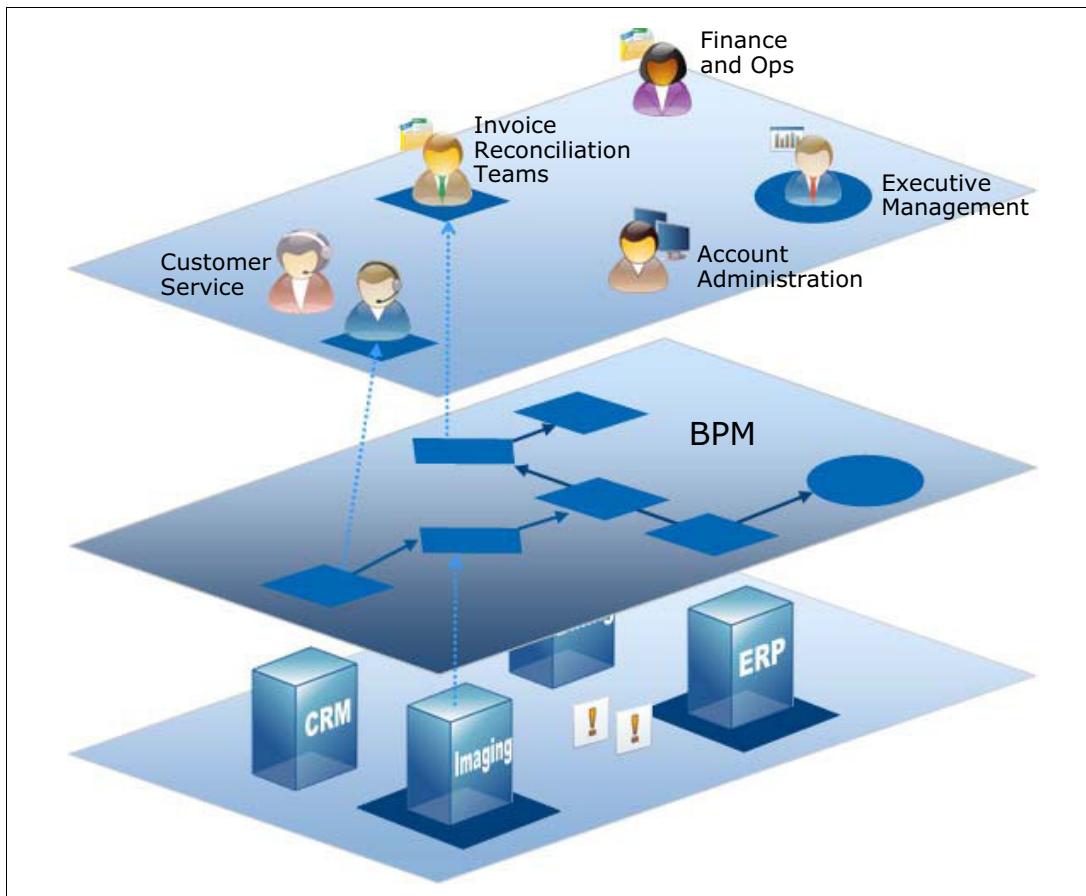


Figure 1-2 Incorporating BPM

As an organization's use of BPM matures from an initial Quick-Win pilot, to an orchestrated collection of BPM projects, the successful deployment and subsequent cost savings associated with these projects most often escalates the adoption of BPM across the enterprise.

Understanding the security implications of Business Process Manager is therefore critical. As more and more of your organization's business processes are detailed in BPM process applications, the "essence and details" of your enterprise expand to include more than just departmental issues—they then potentially represent your enterprise's competitive strategies, industry trade secrets, and, in effect, the very DNA of your business.

All of this information is codified and stored in databases, is communicated across the wire using HTTP protocols, and if not properly secured, would be considered highly visible.

1.1.1 Business Process Manager users have access

Furthermore, consider the universe of users who have access to Business Process Manager. As BPM spreads throughout your enterprise, thousands of users, possibly tens of thousands of users, might be expected to use some aspect of Business Process Manager.

These users have network access as well as valid credentials.

They know which processes, if compromised, would be most disruptive to your business. They know which data, if delivered outside of the corporate firewalls, would be most valuable.

They may also want to bypass security policies for their personal benefit (for example when claiming travel expenses).

The most common security hole we see is an overreliance, or faith, in corporate firewalls. It is very common to hear the phrase “our BPM network is internal, so it is secured.” Yet several sources have stated that the majority of security breaches are instigated from within.

We strongly encourage you to rethink your attitudes towards security. Even if you decide to trust 100% of the employees who have access to your internal networks, can you be certain that all actions they take are consistent with your security policies? What about email or browser exploits which could serve as an entry point to your corporate network? What about “free” software which could include non-trusted code? What about CDs/DVDs/USB drives which your employees might insert into their notebook computers? What about external consultants connecting to your corporate network? Can you ensure that 100% of these devices have been thoroughly scrutinized?

Keep in mind that there is no way that anyone can guarantee a 100% secure environment. Notwithstanding, following the leading practices that are detailed in this book will, at the very least, eliminate the low-hanging fruit and drastically reduce the universe of potential attacks against your Business Process Manager systems.

1.1.2 Business Process Manager has unique security considerations

In addition to these philosophical arguments, there are some practical considerations that need to be addressed with respect to Business Process Manager security. Business Process Manager has a unique hub and spoke deployment model that is built upon the normal J2E deployment model, but extends it in ways that are specific to Business Process Manager (Figure 1-3).

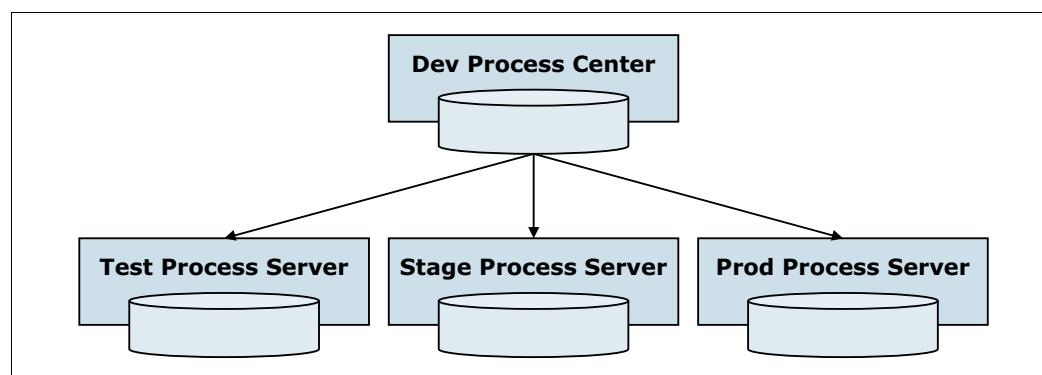


Figure 1-3 Business Process Manager deployment model

Business Process Manager has a unique instance-based authorization model which adds significant functionality beyond the normal J2E authorization techniques. Business Process Manager is delivered atop of WebSphere Application Server, and WebSphere Application Server includes a very sophisticated Virtual Member Manager that allows you to access corporate LDAP (as well as other) user and group repositories. Many applications make use of this group information to perform a high-level authorization, but Business Process Manager's model extends this in ways which need to be understood in order for you to fully evaluate the security implications.

Finally, the ways in which Business Process Manager connects to external systems (web services, message queues, and so on) differs depending upon which Business Process Manager product you are running (Business Process Manager Standard or Advanced), which

version you are running (V7.5, 7.5.1, V8.0) and what type of integration you are discussing (outbound versus inbound web service calls). All of these topics will be addressed in this book.

1.2 Basic concepts

Business Process Manager ships with and is installed upon WebSphere Application Server, a J2E-compliant application server. There are a number of concepts which are native to each of these products that require understanding before we can investigate the security implications of a typical Business Process Manager installation.

Further complications arise when you consider that portions of the Business Process Manager suite came from a product acquired by IBM called Lombardi Teamworks®, and others from what had been called WebSphere Process Server. Although both products shared a common goal of automating and facilitating the management of business processes, they had different focuses. The similarities are now encapsulated in what we call IBM Business Process Manager Standard, and the differences are packaged in IBM Business Process Manager Advanced. Except where noted, the following discussion is applicable to both versions of IBM Business Process Manager.

1.2.1 Business Process Manager

Figure 1-4 shows the main components in Business Process Manager.

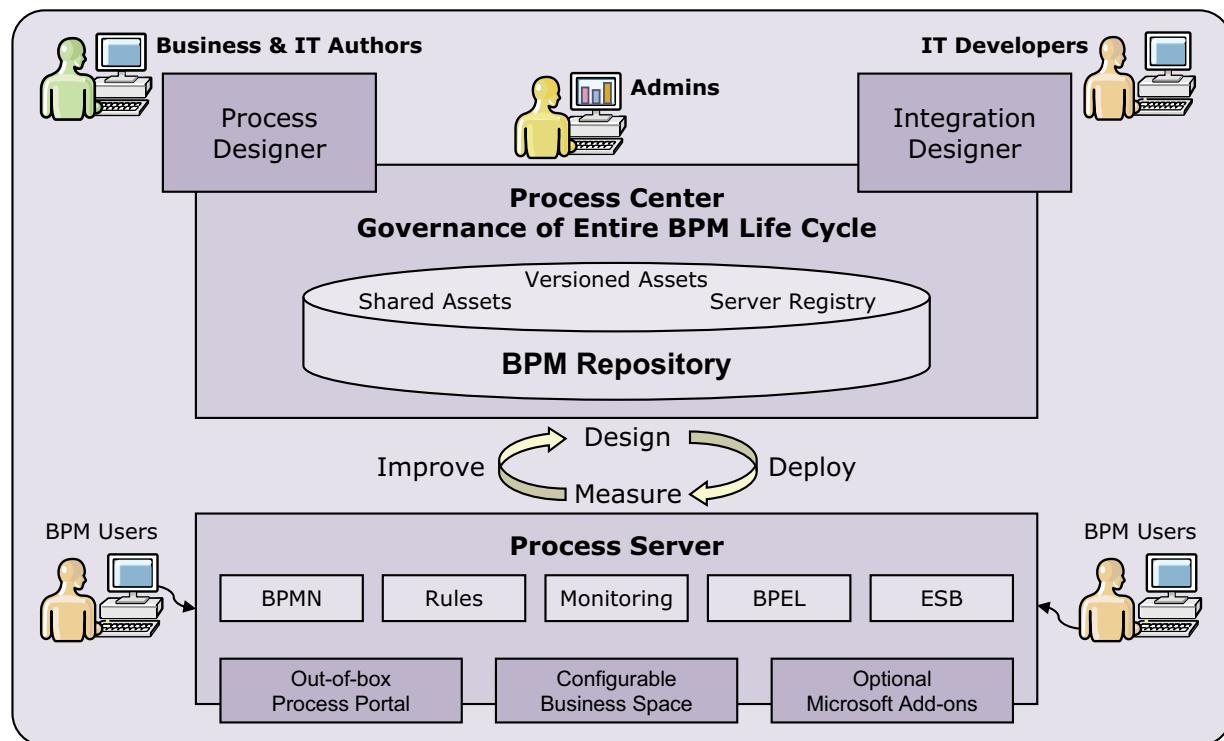


Figure 1-4 Components of Business Process Manager

Process Center

At the heart of Business Process Manager lies the Process Center, the central repository for all Business Process Manager artifacts, and the runtime engine for all process application

development. Your business process authors will connect to the Process Center whenever they wish to create or improve a business process application. (Often, these users are thought of as “developers”, but we will use the term “authors” since many business processes are actively defined and iteratively refined by business subject matter experts who have no formal software development training.)

We define the following naming conventions in order to help distinguish between the different functionality and modes of accessing the Process Center:

- ▶ Process Center will refer to the central repository as described above.
- ▶ /ProcessCenter console (or just /ProcessCenter) will refer to the web-based user interface for accessing certain administrative functions of the Process Center. The rationale for prepending the “/” will be explained when we discuss J2E context roots in the next section.
- ▶ Process Designer’s Process Center console - a second view into the Process Center, similar in functionality to /ProcessCenter, but accessed via the Process Designer development or authoring environment, which will be described next. Since the functionality is similar, we will use /ProcessCenter to refer to both views of the Process Center.

Process Designer

The Process Designer (Figure 1-5 on page 7) is a Windows application that is used by business process authors to create and refine business process applications. Each business process is graphically portrayed as a series of steps (in any combination of human-centric or system-automated tasks).

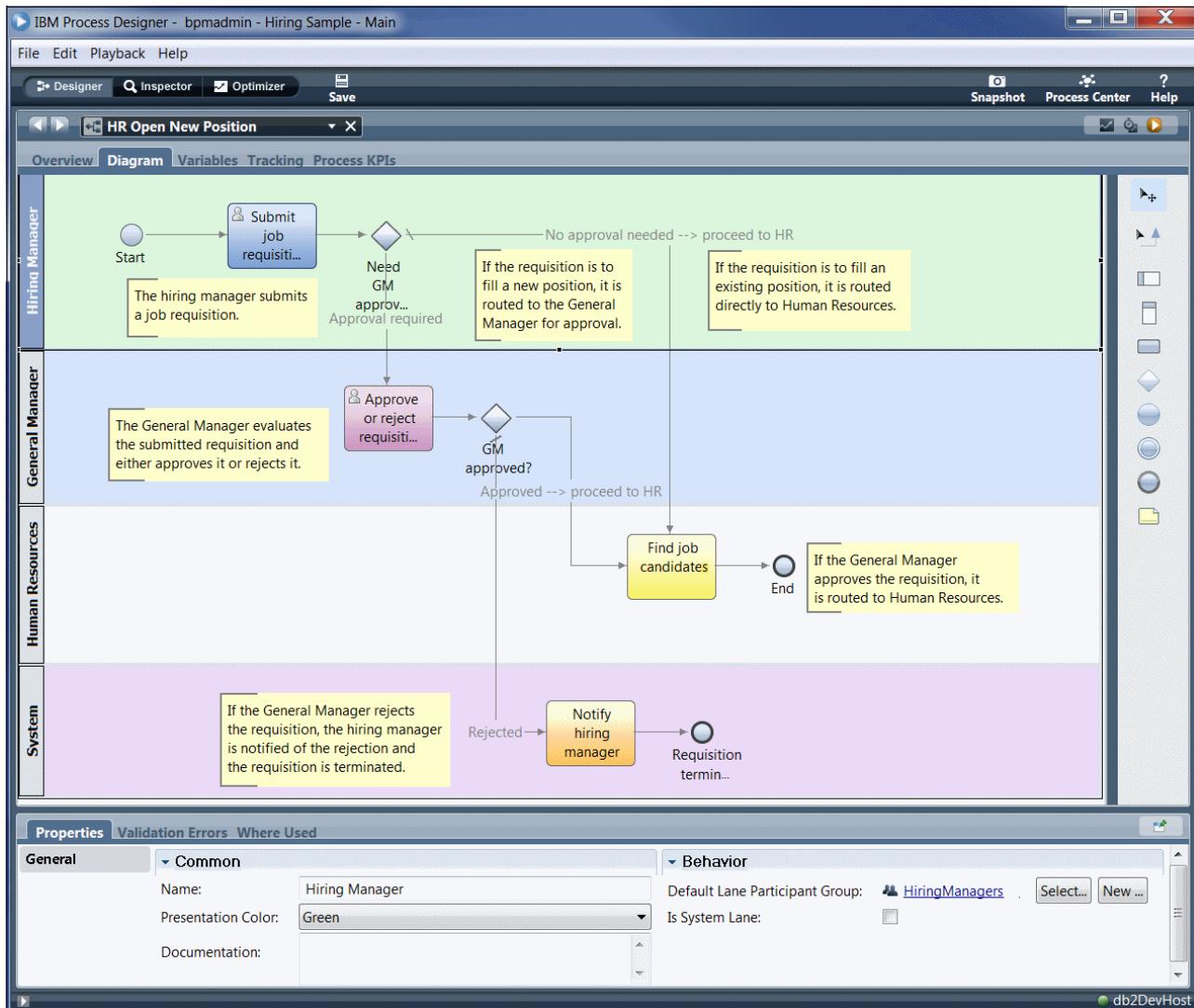


Figure 1-5 Process Designer

In Figure 1-5, you can see that this “HR Open New Position” business process consists of four steps (“Submit job requisition,” “Approve or reject requisition”, “Find job candidates”, and “Notify hiring manager”), distributed over three groups of roles (Hiring Managers, General Managers and Human Resources) plus one system-automated task (System).

These three roles of employees are defined within Process Designer using a Business Process Manager product function called “Participant Groups”. These participant groups can best be thought of as being populated with the employees who are authorized to enact these roles. We will have much more to say on this topic in Chapter 4, “Authorization: Access to what” on page 65.

Note also that in the top right of Figure 1-5, there is a button labeled Process Center. This provides the second view of the /ProcessCenter described in the preceding section.

Integration Designer

Integration Designer is another authoring tool used to define system-to-system interactions such as web services. Whereas Process Designer is used by business process authors, Integration Designer is most typically used by bona-fide software developers. Integration

Designer is a very powerful development environment that allows for the creation of complex, optimized and targeted software modules for the creation or consumption of web services.

Integration Designer is only available in the Business Process Manager Advanced version because the type of code created with it will only run on Business Process Manager Advanced server runtimes.

Process Server

Process Servers are the main runtime engines for business process applications. The business process applications which were developed using the Process Designer, and are stored within the Process Center's repository, are promoted to a Process Server in a runtime deployment environment.

The Process Server is highly optimized (and configurable) to ensure adequate throughput and fail-over for your process applications.

Deployment environments

In almost all corporate environments, there are distinct functional (and often physical) differences between servers used for software development, testing (both by the developers as well as by the users) and their production servers. Business Process Manager supports this common pattern via the mechanism of *deployment environments*. We have already seen that business process applications are developed within the Process Center. The Process Center is also commonly referred to as the Business Process Manager “development” environment.

Once the process application authors are satisfied with their current iteration of the process application, they can take a snapshot of that process application and deploy it to another environment. This next stage is typically called “testing”. Among the benefits of doing this is that you divorce the execution of the process application from the dependencies that may have been present in your company’s development environment—thereby giving you a first glance chance at ensuring that the process application has been bundled up with all the underlying software and toolkits that it needs to execute.

When all parties are convinced that the process application is free of these dependencies and is ready for wider distribution, it is then typically promoted to another environment, called “staging” or “UAT” for User Acceptance Testing. The UAT environment is where a wider universe of users is given access to the process application, and it is their job to ensure that the process application satisfies all business requirements and is bug free.

When the testing is complete, this “staged” application will then be eligible for promotion to the “production” environment—with the complete confidence that the application is ready for use.

1.2.2 WebSphere Application Server

Next, let us take a look at some of the components of WebSphere Application Server, which are the underlying components of Business Process Manager.

Profile

The Profile is a set of .xml files and application components which are used to describe and facilitate an application server where a Business Process Manager instance can be installed. As a first step, the party who installs your Business Process Manager application will gain access to the product installation binaries and go through a few fundamental steps to get the underlying WebSphere Application Server installed.

The next step is to define a profile. A profile is a configuration where you install a common set of WebSphere Application Server binaries (potentially shared among multiple profiles). Included in the profile is the database connectivity information, the account name and password for your Business Process Manager installation, and how each machine to be included in your BPM application environment will be connected in order to provide for high availability and fail-over.

Cell

A cell is the highest level organization unit of collaborating WebSphere Application Server servers. In addition to the servers, which do the actual work, we have a deployment manager (which manages, stores and distributes the configuration of all servers) and node agents (one per node, communicating with each deployment manager to get configuration updates and operations commands such as starting the server).

The cell is the WebSphere Application Server construct that acts as the overall security context for your Business Process Manager installation. It provides for the centralized management of your application, your Business Process Manager users, and for all SSL certificates to be used between the Business Process Manager servers and all other servers. In addition, it is instrumental in ensuring that product components are in sync across all servers in your environment—a step that is particularly important when you need to apply product updates.

In Figure 1-6, the cell is the dotted line that encapsulates all other components.

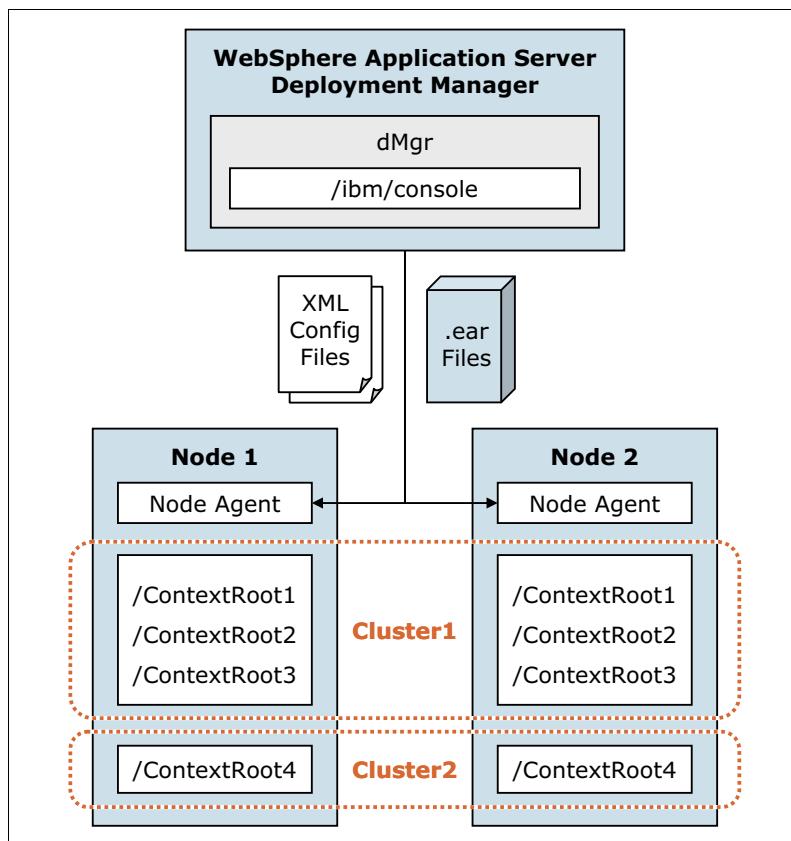


Figure 1-6 WebSphere Application Server cell

Node

The Node is the WebSphere Application Server term for all physical (or virtual) machines in your Business Process Manager environment.

Often, people will call this node a “server”, and a collection of “servers” a cluster. WebSphere Application Server offers a very flexible foundation for fine-tuning and scaling your J2E applications, and you will soon see how the WebSphere use of terms helps us to maintain clarity—even when discussing some very obscure and convoluted configurations!

Application Server

An application server is a Java Virtual Machine (JVM) where business applications are hosted and executed. In Figure 1-6 on page 9, the application servers are the yellow boxes. Each Java application is made visible to web browsers via their “/ContextRoot”. This is just an alias for the address that the users will type into their browsers. The normal incantation for a Business Process Manager product component will be as follows:

`https://bpmhostname:port/ContextRoot`

For convenience throughout this book, we will use the simpler form of “/ContextRoot” when we intend to describe a Business Process Manager product component that is accessible from a web browser—and the `https://bpmhostname:port` will then be assumed. This is particularly important in clustered environments, where many host:port combinations are valid.

As you can see in Figure 1-6 on page 9, an application server can host one or more Java applications (as identified by their /ContextRoot), and any given Java application can be distributed across nodes in a cluster.

Cluster

The cluster is the set of application servers, potentially distributed across multiple nodes, which provide high availability and fail-over.

High-availability refers to WebSphere Application Server’s ability to distribute work load across multiple nodes that have been configured to host a given Java application (or /ContextRoot). If node 1 is busy, new requests can be routed to node 2. If you find that both of your nodes are running at near capacity, it is a relatively easy matter to add a third node, thereby further increasing your system’s availability.

Typically all cluster members are equal (they run the same applications with the same configuration). However, 99Local.xml and related files allow running multiple Business Process Manager servers in the same cluster with different Business Process Manager configurations (for example, running Event Manager on some cluster members, but not on others).

Fail-over refers to WebSphere Application Server’s ability to detect a failure in one cluster member, and reroute all of its existing work to another cluster member on a functioning node. Fail-over is one aspect of a larger topic called “Disaster Recovery,” which is outside of the scope of this book. For more information refer to *IBM WebSphere Deployment and Advanced Configuration*, IBM Press, ISBN 0131468626.

Deployment manager and node agent

The deployment manager and node agent are JVMs that communicate with each other to orchestrate the synchronization of the J2E applications and software components used in the WebSphere Application Server cell. The deployment manager (often referred to as dMgr in

this book) can be configured to run on its own separate node, or on the first node of a cluster. Each node, when added to (or federated into) a cluster, gets a node agent installed.

For more information, refer to *WebSphere Application Server V7.0 Security Guide*, SG24-7660.

1.2.3 Business Process Manager administration tools

Three web applications ship with Business Process Manager, where you will do the majority of your security hardening and ongoing security management:

- ▶ Integrated Solutions Console (/ibm/console) - the main administrative application for the embedded WebSphere Application Server. Using this console, you will perform some generic hardening steps, you will start/stop the Business Process Manager servers within your environment, and you will define the universe of users who will be allowed to log into Business Process Manager.
- ▶ Process Admin Console (/ProcessAdmin) - the main administrative application, which will allow you to associate users with specific roles that have been defined within the Business Process Manager process applications and individual process models. You will be able to define different sets of users for each of your Business Process Manager deployment environments.
- ▶ Process Center Console (/ProcessCenter) - the main administrative application, which will allow you to grant Business Process Manager process analysts, process authors and software developers access to the various Business Process Manager process applications and process models in use at your organization. You will be able to selectively grant certain developers access to process applications by privilege level, process application, and environments.

1.2.4 Installation options

In Figure 1-7 on page 12, we can begin to see how some of the Business Process Manager concepts map into some of the WebSphere Application Server concepts. For example, the Process Center is exposed as a J2E /ContextRoot called /ProcessCenter, running in an application server. Other Business Process Manager product components such as /portal, /ProcessAdmin and /PerformanceAdmin are similarly exposed.

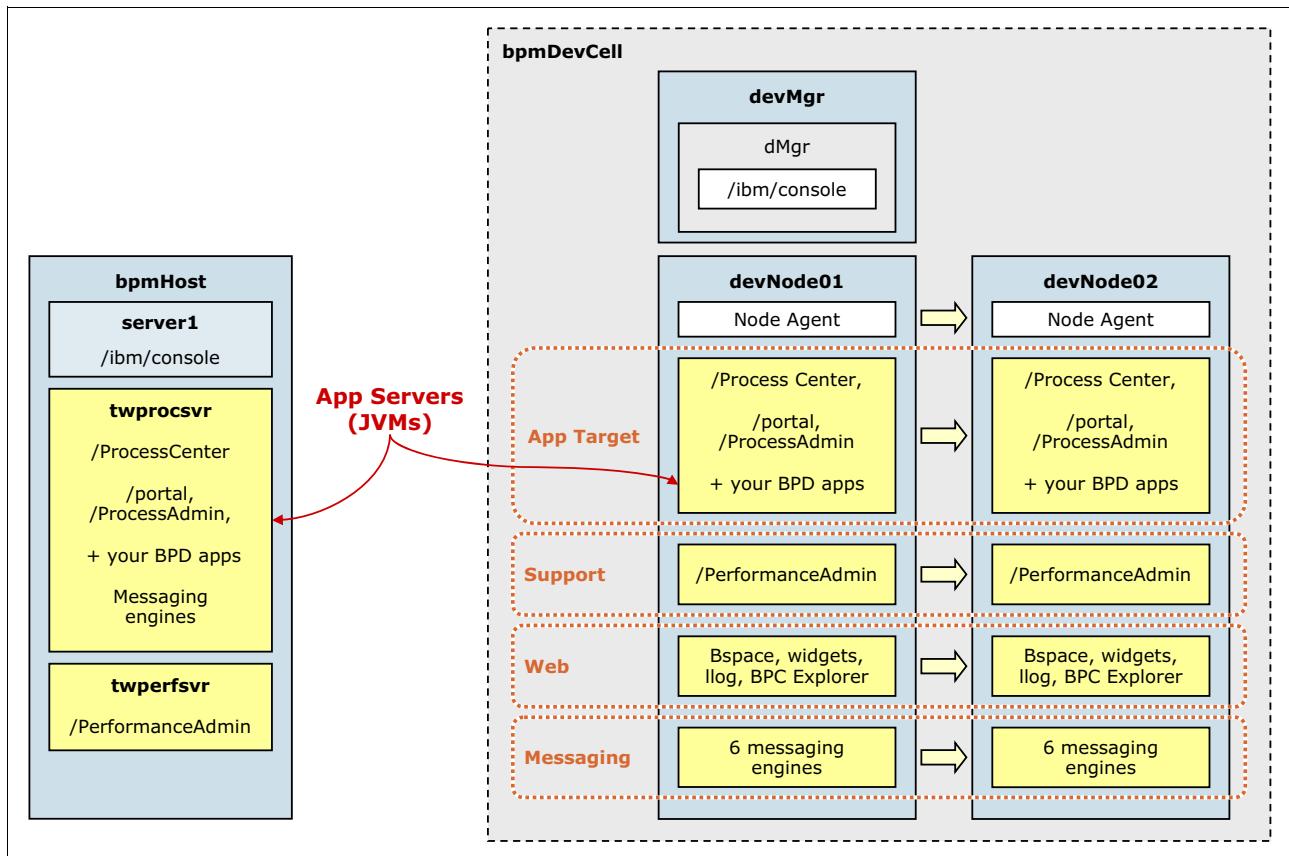


Figure 1-7 Mapping Business Process Manager concepts to WebSphere Application Server

When you install Business Process Manager, you have the choice of installing as Stand Alone or as Network Deployed. The Stand Alone option (pictured on the left in Figure 1-7) is suitable for installation on a single developer's laptop or for some smaller proof of concept type of engagements. There are no clusters.

A Stand Alone topology can even be used for Quick-Win pilots, but often the business value delivered by a Quick-Win pilot is such that the process applications are immediately escalated into a larger, more permanent project, and so we often see even Quick-Win pilots initially delivered as Network Deployments.

The Network Deployment is by far the most common installation option. In Figure 1-7, you can see on the right side that we are depicting a four-cluster topology: as with the Stand Alone configuration, the Performance Data Warehouse functionality is split into its own application server (and exposed as /PerformanceAdmin), but there are three other clusters depicted that host other product components (/portal, /ProcessAdmin, Business Space (BSpace), and so forth).

Furthermore, all four clusters (AppTarget, Support, Web, and Messaging) each span two nodes (which may be either physical boxes or virtual machines). This “spanning” of the application servers over nodes within a cluster is how WebSphere Application Server provides its high availability and fail-over. If Node 1 should become too busy to service new requests, or if Node 1 should fail and become unavailable, then WebSphere Application Server would immediately reroute the traffic to another node within the cluster.

In a Network Deployment, you can specify one, two, three, or four clusters. The choice of topology will depend on a large number of factors. For more information about this topic, refer to:

- ▶ *Business Process Manager V7.5 Production Topologies*, SG24-7976
- ▶ *IBM WebSphere Deployment and Advanced Configuration*, IBM Press, ISBN 0131468626

Despite the fact that this is a large topic, we will still investigate some of the security holes we see that are common to many installations, regardless of topologies, in Chapter 2, “Installation” on page 21.

1.3 Encryption, SSL, and certificates

Encryption, https://, SSL, and digital certificates are so fundamental to the idea of security that it is vitally important that you clearly understand these topics and how they apply to Business Process Manager and WebSphere Application Server.

It is not necessary, however, for you to become an expert in these fields. There is no need for you to be able to explain the mathematical transformations that occur within an AES or Blowfish encryption algorithm. We will keep our discussion to a high-level overview, with the intent to provide you with enough of a foundation to be able to understand these topics when you are exposed to them later in this book, as well as to leverage knowledge of these topics in your organization’s Business Process Manager installation.

1.3.1 Encryption

Encryption is the process of transforming plain text into something that is unreadable unless you possess some secret knowledge.

In early historical uses (1000+ years ago) the processes were very simple, and the secret knowledge needed to decode the message was simply to understand the process and run it in reverse. For example, if you want to conceal the plain text message “IBM”, you might use a simple transposition wherein you shift each letter to the left—for example, you take each letter of the plain text message and replace it with the letters that immediately precede each in the alphabet. Thus, I becomes H, B becomes A, and M becomes L, and the secret plain text message “IBM” becomes the publicly disclosed encrypted message “HAL”. Knowledge of this process is all that is required to reverse and decode the publicly available “HAL” back into the secret message “IBM”. This is known as the Caesar cipher.

Modern processes are far more sophisticated than this example—based upon complex mathematics—but the idea is still the same. There is still a process of transformation (called a cipher), which translates plain text (often also called cleartext) into something unreadable (called ciphertext), and there still exists the requirement of secret knowledge. However, in modern ciphers the process itself is public knowledge, and the secret knowledge has become the input to the cipher (called a key).

Why in the world would you place your faith in a cipher that is common knowledge? The reason is that these algorithms are exceedingly difficult to design securely. “Secret” algorithms, developed in relative isolation, very often contain flaws which become evident only after the algorithm is placed into use. These flaws can then be exploited in order to facilitate the cracking of the cipher. Also, it is far more difficult to keep a cipher’s algorithm secret (since presumably many people will work on an algorithm) than the relatively small input to the cipher (the key). And in the case of a security leak, it is far easier to replace a key (which might take minutes) than an entire algorithm (which could take years). A compromised

key also only puts some data in danger. A compromised algorithm is as bad as it gets. All data ever encrypted with this algorithm becomes public.

In addition, public algorithms are subject to public scrutiny, and over time any potential flaws are very likely to be exposed. For example, the IBM Data Encryption Standard (DES) encryption was in use for over 20 years, subject to plenty of public scrutiny, and with a modern modification (Triple DES), it is still a viable encryption algorithm today.

Modern ciphers are distinguished by two criteria:

- ▶ Type of key used (symmetric private key vs. asymmetric public key)
- ▶ The nature of the input (block ciphers vs. stream ciphers)

1.3.2 Symmetric and asymmetric keys

There are two types of keys: symmetric and asymmetric.

Symmetric key encryption

Symmetric encryption refers to the fact that a single key is used to both encrypt and decrypt a message. The key itself can be anything—a number, a password, a phrase—it just needs to be known to all parties in the communication. The key is used by the cipher algorithm to transform the clear text back and forth into the cipher text (Figure 1-8).

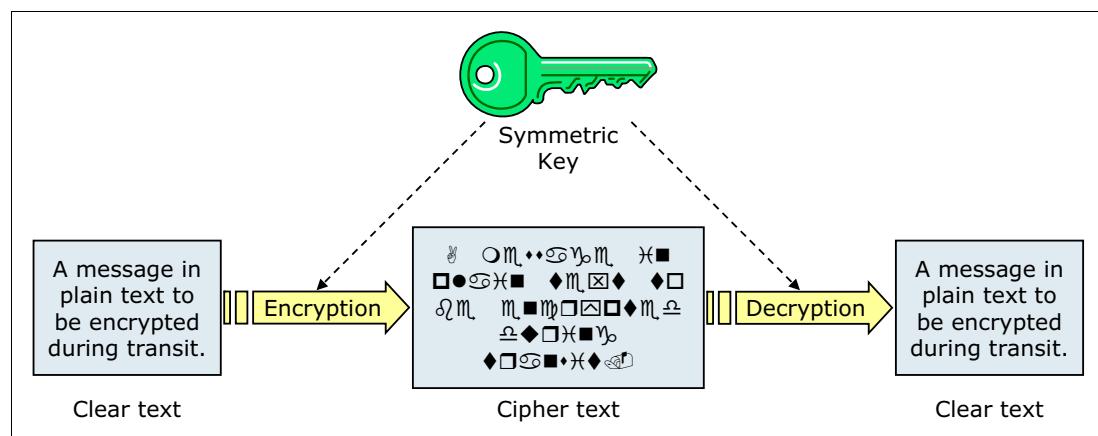


Figure 1-8 Symmetric key encryption

This is loosely analogous to the historic cipher described in the previous section—the secret knowledge was roughly the same for encrypting (shift left one letter) as for decrypting (shift right one letter).

In modern ciphers, this would require that the receiving party has an exact copy of the key that was used to encrypt the original message, as well as knowledge of what cipher algorithm is being used. This can present a number of logistical problems:

- ▶ How do you get that key securely to the receiving party?
- ▶ How can you ensure that the receiving party is the only party in possession of that key?
- ▶ What if you need to share that encrypted message with a large number of parties?
- ▶ What happens if you need to revoke just one party's use of your key?
- ▶ How would you detect if an unauthorized third party has the key as well?
- ▶ If you choose to change the key, how do you ensure that everyone has the latest copy?

On the other hand, symmetric key encryption is relatively fast—thanks to the mathematics behind these algorithms. So, do not give up on symmetric key encryption just yet—we will describe a use for it soon enough.

Asymmetric key encryption

As you might have guessed, asymmetric encryption uses two keys instead of just one: One is for the encrypting of the message (the public key), the other for decrypting the same message (the private key).

The mathematics behind asymmetric encryption is vastly more complex than that for symmetric key encryption. The two keys (one for encrypting, the other for decrypting) are distinct, different keys, and one cannot guess the private key simply by knowing the public key. Notwithstanding, the two keys are related mathematically, and through this mathematical relationship, a message encrypted by one key can only be decrypted by using the other. Asymmetric key encryption is shown in Figure 1-9.

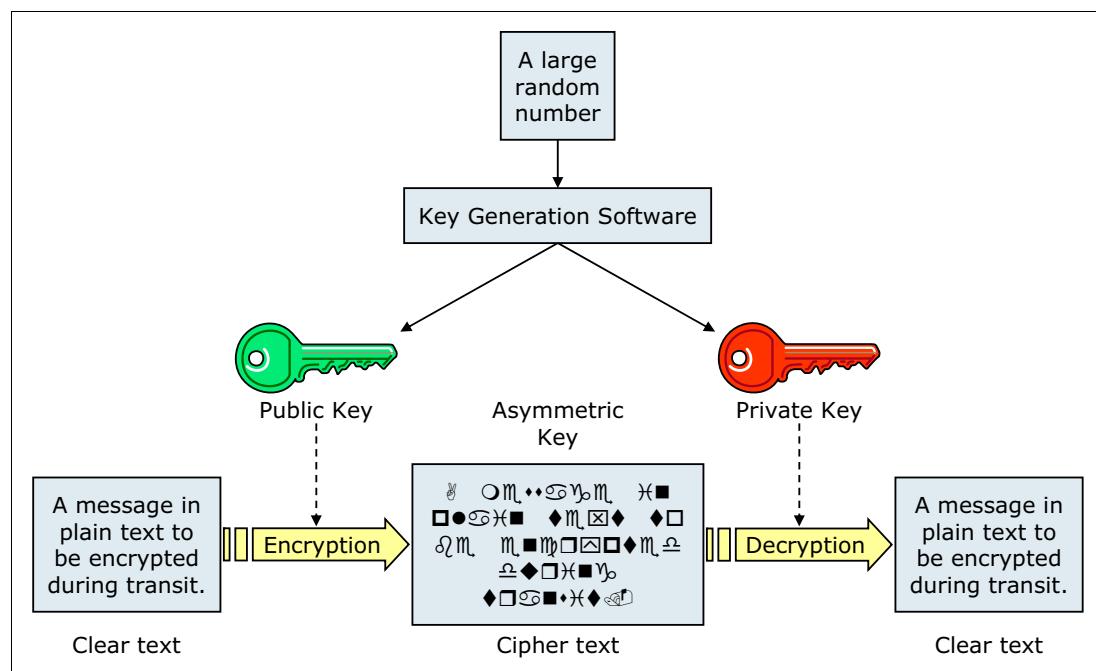


Figure 1-9 Asymmetric key encryption

Even though asymmetric computation is slower than symmetric encryption (thanks to the more complex mathematics used in asymmetric encryption), there are distinct advantages when it comes to logistics. Since knowledge of one cannot be used to discover the other (well, to be exact, it is such a hard problem that most consider it simply computationally infeasible), one of the keys is publicly distributed, and that public key is used to encrypt any message intended for the key holder.

The individual or organization wanting to receive encrypted messages begins the process by feeding a large random number into a key-generation software program. This program outputs two keys: one intended for encrypting messages (the public key) and the other intended for decrypting messages (the private key). The individual or organization then widely distributes the public key.

If someone wants to send the key holder a secured message, then they simply need to download the public key and use that to encrypt the message. *Only the holder of the private key will be able to decrypt this message.* Note that this process scales very well—any large

number of people can download the one public key, perform their encryption, and the key holder only needs his one private key to decrypt.

If the sending party desires an encrypted response from the receiving party, then they too must go through the process of generating a public+private key pair for themselves, and the process is used in reverse.

If there is going to be a great deal of communication between these two parties, then the computational overhead of asymmetric encryption can become an issue. What we need is a way to leverage asymmetric encryption's ease of logistics with symmetric encryption's speed. Enter the handshake.

1.3.3 SSL and digital certificates

When we endeavor to secure computer systems, their data, and the networks that carry that data, we are often faced with variations on the themes of logistics and performance that were discussed in the previous two sections. To satisfy these competing goals (simplified logistics without encumbering performance), many web servers use a specific negotiation protocol to establish a secured communications link.

Originally proposed in 1995 by the Internet browser company Netscape, the Secure Sockets Layer (SSL) quickly underwent a series of public reviews and improvements and by 1999 had become the primary mechanism by which electronic commerce is secured on the web today. SSL is so universally accepted that the newest specification of this protocol, the Transport Layer Security (TLS), is still often referred to as SSL.

At its core, the SSL/TLS protocol consists of a series of messages sent between the client (often a user's web browser) and a web server in order to securely exchange the symmetric keys needed to encrypt their subsequent communications. Figure 1-10 on page 17 is a simplified view of this SSL handshake.

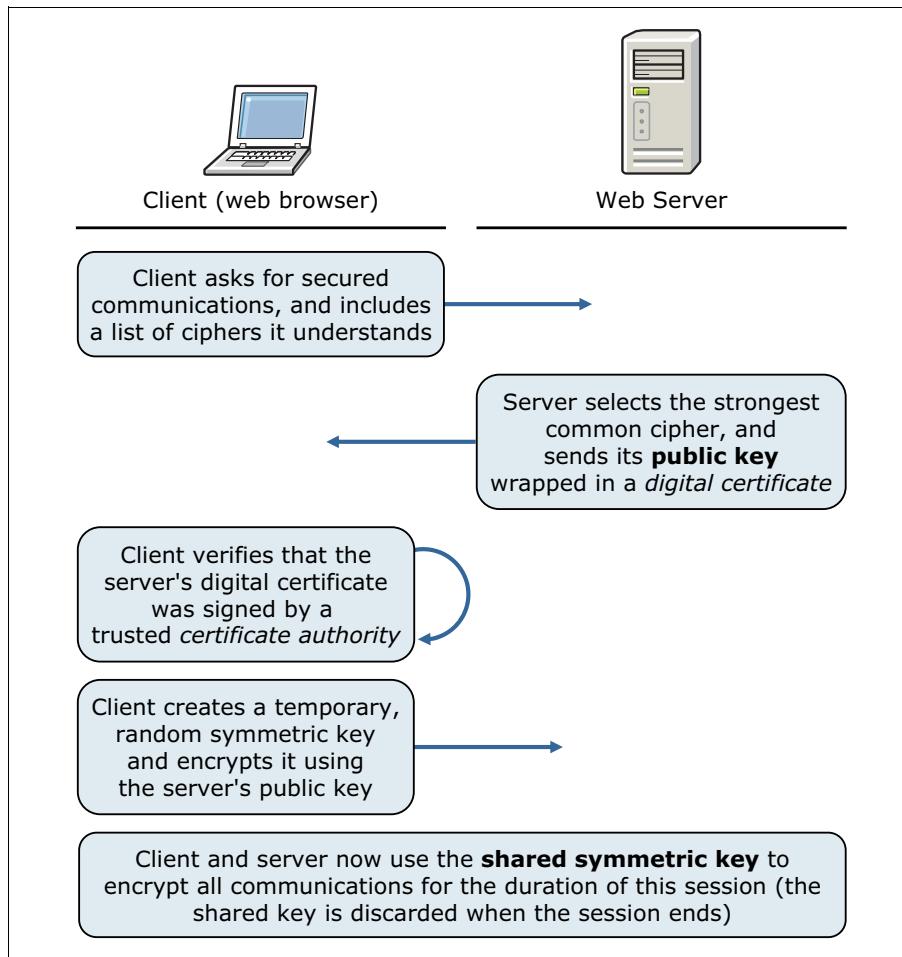


Figure 1-10 SSL handshake

The administrators of the web server had previously created an asymmetric key pair, and had bundled their public key in what is called a digital certificate. This digital certificate asserts that the public key is in fact owned by the web server.

Although modern computers can compute an asymmetric key pair in less than a second, if this process needs to be repeated for each and every SSL client request, the computational overhead will up. By distributing the server's public key and placing the burden of creating and encrypting the shared symmetric key on the client (a one-time per session computation), the entire SSL handshake becomes quite manageable.

Logistics and performance problems solved!

For more information, see: *HTTP Server (powered by Apache) SSL/TLS Cipher List Handshaking*, TIPS-0284

In the following section, we investigate that one message in the middle wherein the client verifies the server's digital certificate against a list of trusted certificate authorities. But before we begin that discussion, make a note of how elegant this SSL handshake solution is. It takes the best of both encryption schemes (asymmetric and symmetric) and delivers a fast, rock solid solution.

1.3.4 Certificate authorities

So, who is the third party (the certificate authority) referred to in Figure 1-10 on page 17? In order to ensure that the public key used in the SSL handshake actually belongs to the web server, all modern browsers take advantage of what is called the Public Key Infrastructure (PKI). The PKI uses a “web of trust” consisting of certificate authorities (CA) whose primary function is to guarantee the authenticity of web servers’ identity.

The certificate authority digitally signs the web server’s certificate, and if the browser holds a certificate from the same CA that signed the web server’s certificate, then the trust is established by basically saying in effect:

“if you trust this web server, then I’ll trust it too.”

Have you ever taken a look at the list of certificate authorities which are supported by your browser? Figure 1-11 is a screen shot from a brand-new installation of one of the most common browsers:

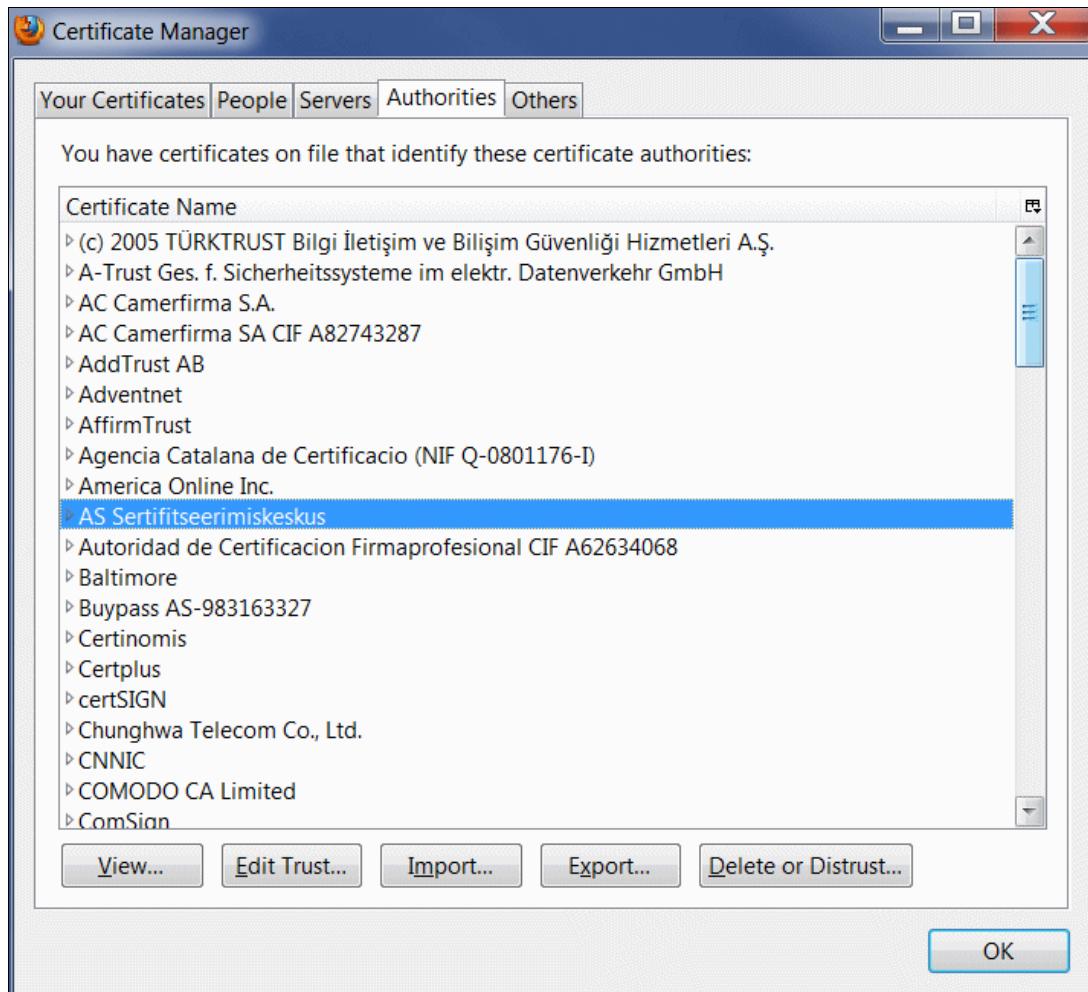


Figure 1-11 Certificate Manager

There are 83 certificate authorities in this list: including CAs from Turkey, Germany, Spain, United States, Estonia, France, the Czech Republic, Romania and China—and that is just the countries represented on the first page of this very large scrolling list.

Have you ever wondered how difficult it might be to get your own certificate? Just perform a web search. You'll see that for just \$12.95, you too can be fully trusted by all known web browsers.

The issue here is simple: if anyone can purchase a certificate from any one of these certificate authorities, and as a result be fully trusted by your browser, then you should look very closely at:

- ▶ The list of CAs that you trust
- ▶ The address of the website to which you are connecting

Suppose you receive an email from an entity proposing to be a well-known financial institution. Suppose that this email is formatted with and includes graphics which look identical to the institution's normal look and feel. You click on a link in the email, and your browser opens to a page with an address of:

<https://well-known-financial-institution.offers.com>

What is important to realize here is that the https:// ensures that you will have an encrypted session—yes—but do you realize that this session is *not* with your well-known financial institution? The encrypted session is with some other website. The administrators for this website have purchased a digital certificate from one of the CAs that is represented in your browser. It could have been from any one of these CAs. If you trust the CA, and the CA sold the website a certificate, then you are trusting the website too. How much trust can \$12.95 buy you?

You can acquire a certificate, which can be used to sign other certificates. Acquiring here could mean purchasing, stealing, or legally mandate issuing. If you have a certificate, trusted by browsers and valid for signing other certificates, you can build a man-in-the-middle solution. You make sure all traffic is routed through your server. If the traffic happens to be HTTPS, you just sign a certificate for the target domain on the fly and present that during the handshake. This is done in corporate environments (the corporate CA is good for signing certificates and is injected in all browsers in the enterprise), in schools and universities and possibly in countries interested in censorship. The browser will then show the correct URL and an icon indicating a trustworthy connection.

While the reasons for, and possible criticism of, the Public Key Infrastructure, are well outside of the scope of this book, it is worth keeping these ideas in mind when we return to this topic in the next chapter. We will be discussing some common security holes we see within Business Process Manager installations—and some are very relevant to this discussion.



Installation

Proper security begins before the software gets installed. How your Business Process Manager servers fit into your enterprise will require careful consideration, with an eye towards security implications at each and every touch point.

- ▶ How many BPM deployment environments will you have?
 - Are each protected by firewalls?
- ▶ Which of your BPM deployment environments will be protected by DMZs?
 - Will they be protected by proxies or dedicated web servers?
- ▶ How will you secure your network traffic?
- ▶ How many certificate authorities will you trust?
- ▶ How will you secure your data at rest?
- ▶ How will you secure web service calls?

In addition to these generic questions, you also have Business Process Manager specific questions:

- ▶ How many user repositories will the Business Process Manager servers access?
- ▶ Does each BPM deployment environment support different user repositories?

Therefore, we shall consider this chapter—the installation of your Business Process Manager software and the decisions surrounding the install—to be one of the first steps in security. This will be the foundation upon which your Business Process Manager installation's security relies.

2.1 Business Process Manager and WebSphere Application Server topologies

Lets start by looking at Business Process Manager topologies, which are built upon WebSphere Application Server Network Deployment.

2.1.1 Basic concepts

In Figure 2-1, we pull together the Business Process Manager and WebSphere Application Server basic concepts and can see how they overlap. There is almost always a one-to-one relationship between a Business Process Manager deployment environment and a WebSphere Application Server cell.

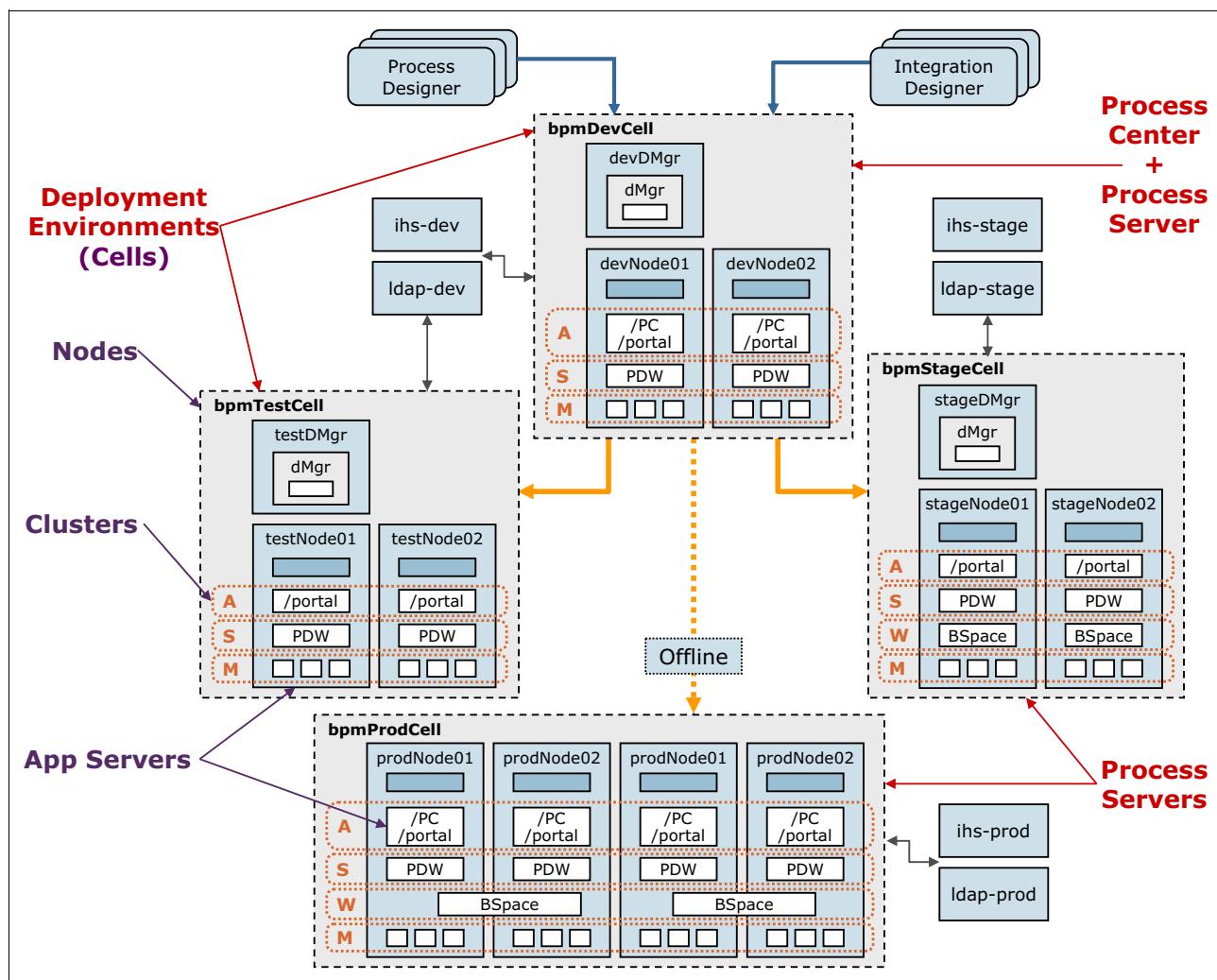


Figure 2-1 Business Process Manager topology basic concepts

In Figure 2-1, we have named the cells according to their role as deployment environments:

- ▶ **bpmDevCell** - Contains the Process Center and central development repository. This deployment environment is often referred to as “development”, “PC”, or “the dev environment”.

- ▶ **bpmTestCell** - A Process Server runtime deployment environment where process application authors can ensure that their process applications are free from any dependencies upon development tools or artifacts. This deployment environment is often referred to as “testing”.
- ▶ **bpmStageCell** - A Process Server runtime deployment environment where process application stakeholders can ensure that the process applications are isolated completely from the development environment. Typically, the staging environments closely mimic the production deployment environment in the hope and expectation that once the staging testing is completed, the process applications will be able to be installed into production environments with no issues. This deployment environment is often referred to as “staging” or “User Acceptance Testing” or, more simply, “UAT”.
- ▶ **bpmProdCell** - A Process Server runtime deployment environment where process applications are deployed for use in the enterprise. Often, these environments are behind strict firewalls and have special deployment considerations. This deployment environment is often referred to as “prod” or “production”.

The distinctions above between test and staging are largely superficial. There is no difference in functionality between them, and often corporations have distinct ideas about how test and staging are treated. Some corporations incorporate yet another “stage” environment called “pre-production”.

By having each deployment environment hosted within a unique WebSphere Application Server cell, the cell’s security configuration can be leveraged and differentiated from that of other deployment environments or cells. For example, you would almost certainly want the staging LDAP server and the production LDAP server completely isolated from those of the development and testing environments.

In addition, in Figure 2-1 we can see that the development and test environments share a common LDAP server (ldap-dev), but that staging and production each have their own LDAP servers (ldap-stage and ldap-prod). Furthermore, both stage and production are 4-cluster topologies, where development and test are only 3-clusters. The production environment also is the only environment where the deployment manager (dMgr) is broken out onto its own server (physical or virtual). This is not a strict requirement, simply an example. Business Process Manager and WebSphere Application Server allow a great deal of flexibility in how you define your BPM environment.

For further information on Business Process Manager topologies, see *IBM Business Process Manager V7.5 Production Topologies*, SG24-7976.

2.1.2 Complex realities

But reality is rarely as pretty as Figure 2-1 would have you believe. The complications multiply when you consider the environment into which Business Process Manager is being deployed. Each Business Process Manager deployment environment will have its own database server, typically a front-end web server, hardware load balancers, Single Sign-On solutions (which may also be hardware), servers that host web services and/or any number of corporate Enterprise Information Systems (EIS) servers. See Figure 2-2 on page 24.

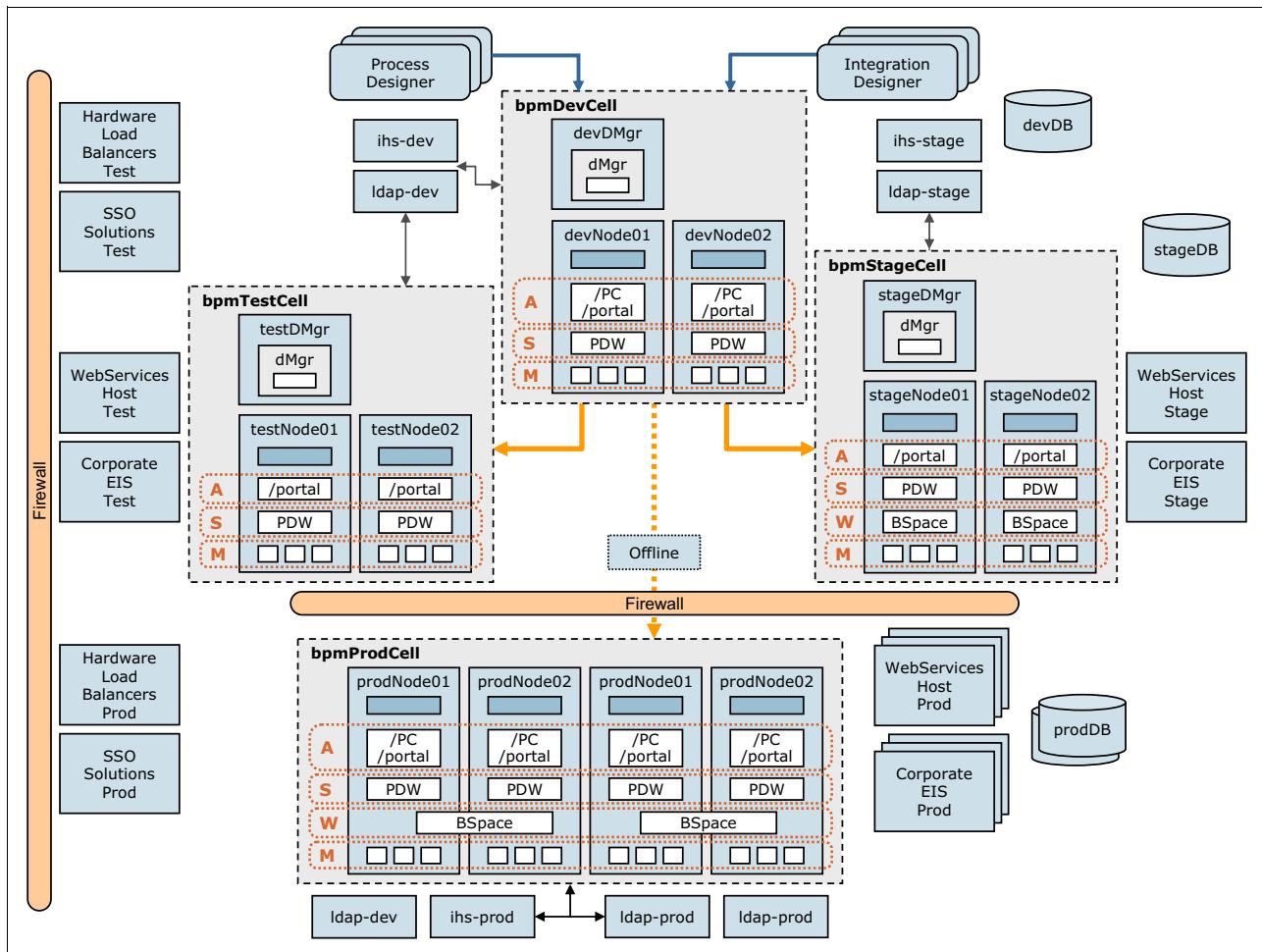


Figure 2-2 Complex realities of a Business Process Manager solution

All of these components talk to each other, and each line of communication introduces another touch point that needs to be scrutinized from a security point of view. Complexity breeds fragility, which in turn breeds opportunity for exploitation.

The antithesis, of course, is simplicity. But how does one achieve simplicity when faced with so many product components, each one speaking to another over different protocols?

Complexity is tamed via decomposition: breaking down the universe of connection points and possible security holes into easily understood components, with an overall framework for management.

Architectural documentation: We strongly advise that your organization invest in architectural documentation that describes the nature of each communication point, together with a discussion of communication options and rationale for choosing one option over another.

Documentation of this nature provides an excellent repository of knowledge and expertise, which can enable all interested parties to understand the security choices that have been employed at each connection point.

It is our hope that as we discuss the various “security holes” that we have seen, that you will return to your organization and seek out how your installation compares on each of these

topics. Furthermore, if you should find that you need to instigate change around any of these topics, that you will commit your rationale to documentation as described above.

2.2 Common security holes

Each chapter in this book describes the common security holes and/or mistakes we have seen in the field. Some of these security holes are actually introduced on purpose as something of a necessary evil, in order to facilitate the installation of the software. Security holes of this nature are easily remedied by the performance of some simple post-installation hardening tasks. We will call these out as appropriate. But more often than not, these security holes are more representative of a misplacement of faith in a single technology.

Still, there is some good news: all of the security holes that we see in common practice have relatively easy solutions. We will, of course, point out the solutions as well as we go through the chapters of this book.

With respect to the topology and installation choices, these are the most common security holes we see:

- ▶ Faith in firewalls
- ▶ Failure to use SSL between BPM and database server
- ▶ Failure to encrypt data at rest
- ▶ Failure to use SSL between Process Center and Process Server
- ▶ Overuse of default Business Process Manager accounts
- ▶ Overuse of trust in certificate authorities

2.2.1 Faith in firewalls

Beyond a doubt, the biggest security hole we see is an overly optimistic belief in the security of a corporate, perimeter-wide firewall. Time and time again we hear the phrase “it is the internal network, so it is secure.” This is a very dangerous and precarious posture to take. It is akin to placing all of your eggs in one basket.

Can you trust with 100% certainty that your firewall vendors will *never* release a software update that has a security hole in it? How often is your laptop’s operating system updated with security fixes?

The simple fact is that many studies, from Gartner, Ponemon, the FBI and others, have shown that:

- ▶ The majority of attacks originate from within.
- ▶ The attackers’ identity is rarely known or discovered.
- ▶ Security breaches are equally likely to be caused by employees as by external agents.
- ▶ An overwhelming majority of attacks are the result of an exploitation of a misconfiguration or failure to follow leading practices.

Security breaches do not have to be the result of malice. They could be the result of simple, honest mistakes. But in the end, it simply does not matter. The security breach occurred, and you have to deal with the consequences.

Notwithstanding, it is still prudent to consider the following question: is it safe to believe that not one of your employees would ever steal data?

The problem we face with “hactivist” groups like Anonymous is that they are, in fact, anonymous. Until an arrest is made, these people remain nameless. Estimates on the

number of arrests (and therefore name disclosures) as a percentage of security attacks are hard to come by, but it is reasonable to assume that more attacks occur than arrests. This leaves us with the realization that *most* security breaches are instigated by individuals who remain anonymous—they could be anyone.

And even if you choose to trust that there is not one employee within your ranks who sympathizes with hactivists, can you ensure that 100% of your employees *always* follow corporate security guidelines? What about email or browser exploits that could serve as an entry point to your corporate network? What about “free” software that could include non-trusted code? What about CDs/DVDs/USB drives which your employees might insert into their laptops—can you ensure that 100% of these devices have been thoroughly scrutinized?

The bottom line on firewall security is this: it is necessary, it is very helpful, but it is *not* a stand-alone solution to enterprise security.

Firewalls are necessary, yes. But sufficient, no.

Our goal is to secure each and every touch point, ensuring that the “low hanging fruit” is eliminated, thereby radically reducing the potential number of attackers by simultaneously increasing the skills required to exploit a security hole.

In this chapter, we look at two specific connection points where failure to use SSL is far easier to exploit than you might imagine:

- ▶ Between Business Process Manager and database server
- ▶ Between Process Center and Process Server

2.2.2 Failure to use SSL between BPM and database server

Everyone recognizes that database user accounts should be password protected. What most fail to recognize is how incredibly easy it is to observe database traffic while it is in transit. If a hacker has the ability to view unencrypted text on its way to the database, they also have the opportunity to store this data in their own systems or to gain knowledge of how your SQL statements are formed, possibly leading to SQL injection attacks.

The solution to this is simple: SSL. The specific steps to ensure SSL between your Business Process Manager and database servers will be, of course, unique to the database vendor you have selected and to your company’s certificate management strategy. Notwithstanding, the concept is simple enough, and should be familiar to your database analysts and administration team.

Tip: We strongly advise SSL/TLS for the communications link between your Business Process Manager servers and your database servers.

2.2.3 Failure to encrypt data at rest

The vast number of security breaches and identity thefts which have been widely publicized on the mass media over the past few years have brought with them a sense of urgency. Government agencies and regulators have pushed for more restrictions and more legislation to govern how data is stored and exchanged, as well as new reporting requirements when security breaches are detected.

Regulatory requirements

As a result of this increased scrutiny, a number of laws and regulations have surfaced that require organizations to enforce a variety of security-related practices. Amongst these are requirements for unique user ID and password usage, protected stored data, extra precautions and preventative measures with respect to data intrusion, more stringent auditing and monitoring of all access to data, regular security systems and processes tests, and most importantly, the storage of data using encrypted software.

This is a global phenomenon. Examples of security legislation include:

- ▶ Payment Card Industry (PCI) Data Security Standard (DSS): This regulation is owned and distributed by the PCI Security Standards Council. It is a collaborative effort between major credit card companies and is designed to protect customers' personal information.
- ▶ California Senate Bill 1386: This is an amendment that provides consumers with notice of security breaches involving compromised personal information.
- ▶ Health Insurance Portability and Accountability Act of 1996 (HIPAA): This act focuses on health care and is designed to protect all forms of personal health information by defending patients' rights to have their health information kept private.
- ▶ Gramm-Leach-Bliley Act of 1999 (GLBA): This act defines that the financial institutions must comply with the privacy provisions that mandate controls over customers' non-political personal information (NPI) with respect to usage, protection, and distribution.
- ▶ Sarbanes-Oxley Act: This act redesigned federal regulation of public company corporate governance and reporting obligations by demanding accountability and assurance of financial reporting by executives, auditors, securities analysts, and legal counsel.
- ▶ Personal Information Protection and Electronic Documents Act (PIPEDA): This act is a Canadian law that incorporates and makes mandatory provisions of the Canadian Standards Association's Model Code for the protection of personal information. PIPEDA contains various provisions to facilitate the use of electronic documents and governs how private sector organizations collect, use, and disclose personal information in the course of commercial business.
- ▶ Data Protection Act (DPA): This is a United Kingdom Act of Parliament that defines a legal basis for the handling in the UK of information relating to living people. It is the main piece of legislation that governs protection of personal data in the UK. Compliance with the act is overseen by an independent government authority, the Office of the Information Commissioner (OIC).

Encryption is a requirement for many of these regulatory compliances. For example, California Senate Bill No.1386 states that any agency, person, or business located in or conducts business in the state of California is required to disclose any breach in security to every resident of California whose information has been compromised. Personal information that triggers this notification requirement is defined as any part of the individual's name combined with any of the following when either the name or this data is not encrypted:

- ▶ Social security number
- ▶ Driver's license number or California ID number
- ▶ Account number, credit or debit card number, when breached in combination with any required security code, access code, or password that would permit access to an individual's financial account

For a business operated in California to comply with this bill, data encryption is required.

Most customers consider having a firewall and database security good enough protection, but what if a hacker gets past those barriers? What if the intrusion is from an authorized user?

What if a mobile computer or backup tape is stolen? Encryption ensures that the data is unreadable.

Common sense requirements

All government regulations and requirements aside, the most powerful argument for encrypting your data is simply this: common sense. A company will spend tens of millions of dollars per year in advertising in the hopes of gaining mind share among potential customers, but typically far less than that on data security.

One data security breach, however, can result in as much (or more) media attention than that same company's entire annual advertising budget. It does not take a stretch to imagine that media and public attention of this nature—highlighting a security breach—will have at least as great an impact on consumer confidence in your business as all of your marketing efforts combined.

If you want to stay out of the security breach headlines, you need to take all elements of security seriously.

Strategies to encrypt data at rest

The following strategies can affect encryption of data at rest:

- ▶ Application specific code
- ▶ Database encryption
- ▶ Operating system and file system encryption

Application layer

Business Process Manager does not provide application-layer support for encryption of data at rest. At least not for the data that Business Process Manager manages itself; that is for the model, state or variable data of a business process. However, it does not restrict you in your ability to encrypt sensitive data yourself. So, if you build an application working with employee or payment related data, you would need to make sure that no sensitive data is ever stored in process or service variables. Instead, you would typically store a pointer to a record in a custom application.

Business Process Manager is highly database-driven, and much of the user-facing code executes within a web browser as JavaScript. Without disparaging JavaScript as a language, it is safe to say that any meaningful encryption algorithm is almost certainly going to overpower the runtime JavaScript engines found within any browser. While it might be possible to overcome some of this by doing the encryption and decryption on the server and then sending in clear text the data to the browser, the fact remains that this is a moot point. Be it because of performance concerns, compatibility constraints, or simple design decisions, Business Process Manager does not offer this strategy for encrypting data at rest for any data elements which are intrinsic to the inner workings of the product. Encryption in the client is very rare. Typically you want the server to work with the data so the server needs the data in clear text.

It might be possible to design your business process applications to encrypt and decrypt data using the extension points provided by Business Process Manager, but again, this is most likely just a theoretical consideration. The computational overhead of doing so within the confines of a runtime JavaScript engine is formidable, but perhaps even more important is that you cannot know with absolute certainty what percentage of your data may be cached somewhere within the Business Process Manager database tables, effectively nullifying your encryption.

Therefore, we shall continue to look at the other strategies for affecting encryption of data at rest.

Database layer

The main advantage of encrypting your data at the database layer is that you have fairly fine-grained control over what data you subject to encryption. We say “subject to” because encryption is a relatively expensive operation, and encrypting every table would be considered by most to be overkill.

Business Process Manager ships with a tool called DbDesignGenerator, which outputs the full set of SQL statements needed to create all of the database tables and indexes that are required for each Business Process Manager product component.

When Business Process Manager is installed into any given deployment environment, the installation process can proceed down one of the following paths:

- ▶ You create the databases and then let the product create the tables.
- ▶ You create the databases and then you run these scripts to create the tables yourself.

If you desire column-level encryption for the tables that are intrinsic to Business Process Manager, then you will need to proceed down the second path.

In addition, you will need to decide upon which tables, even potentially which columns in which tables, you desire to encrypt. Subsequent to that, you will need to manually alter these tables' create() statements according to both your desires as well as the specifics of the database vendor you are using. Also, you need to carefully observe what happens during migrations and updates. Major version changes can introduce database schema changes.

Table-level versus column-level encryption

When discussing database layer encryption, again there are two paths that are possible:

- ▶ Table-level encrypts the contents of an entire table.
- ▶ Column-level encrypts the contents of specified columns only.

Business Process Manager is compatible with the following three database vendor's enterprise databases:

- ▶ The IBM DB2® offers column-level encryption.
- ▶ Microsoft's SQL Server offers table-level encryption.
- ▶ Oracle offers both column-level and tablespace-level encryption.

The main advantage of using column-level encryption is protecting sensitive information or data from attacks against the data within the tables. However, neither scheme can protect the data files stored on the operating system from privileged users who have a significant amount of systems access. In addition, without careful planning, column-level encryption can have a negative impact on the database performance and how data is accessed. For example:

- ▶ Column-level encryption drastically diminishes the effectiveness of indexes, particularly for range queries.
- ▶ Column-level encryption does not prevent users with SYSADM or DBADM database authorities from deleting the data or dropping the tables hosting the encrypted data.
- ▶ Column-level encryption is easily affected by changes to the underlying table, for example, altering the column types and altering objects that reference them such as referential constraints.
- ▶ Column-level encryption is not transparent—in most cases—to the application.
- ▶ Column-level encryption cannot be used to protect against attacks on log files such as audit, import, and export files.
- ▶ Column-level encryption keys are stored in files in the OS and can be potentially compromised, eliminating the benefits of encryption.

From a performance perspective, column-level encryption and decryption reduce the benefits of caching data rows:

- ▶ Column-level encryption adds significant overhead, not because of cryptography but because of the manner in which column-level encryption inserts itself in the database.
- ▶ Column-level encryption reduces the performance benefits of caching data rows, since the rows must first be decrypted.

IBM DB2

IBM DB2 offers column-level encryption, but in order to take advantage of these features, tables in the database need to be created with one of two column types:

```
CREATE TABLE employees (
    id NUMBER(10),
    data CHAR2(50) FOR BIT DATA
)
```

Or:

```
CREATE TABLE employees (
    id NUMBER(10),
    data VARCHAR2(50) FOR BIT DATA
)
```

In addition, the application must also set the password to be used, and it must *explicitly call out* that the value to be inserted be encrypted:

```
SET ENCRYPTION PASSWORD = 'testpwd';
INSERT INTO employees (data) VALUES ENCRYPT('this will be encrypted');
```

This additional lexeme, ENCRYPT, is not universally supported by all database manufacturers.

Because Business Process Manager has been built to run atop a wide variety of database products, it does not include this DB2-specific semantic, and as a result, the DB2 column-level encryption feature is incompatible with Business Process Manager.

For more information, see *DB2 Security and Compliance Solutions for Linux, UNIX and Windows*, SG24-7555.

Microsoft SQL Server

Microsoft SQL Server 2008 Enterprise Edition offers a feature called Transparent Data Encryption (TDE). This feature works at the database level, not at the column level. To enable TDE, you must first create a database master key and a certificate to use as the encryption key:

```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'some password';

CREATE CERTIFICATE tdeDbCert WITH SUBJECT = 'TDE DB Certificate';

CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE tdeDbCert;
```

You can then specify which database is to be encrypted:

```
ALTER DATABASE employees SET ENCRYPTION ON;
```

The Microsoft solution encrypts the database at the page level. Pages in a database are encrypted before they are written to disk and are decrypted when read back into memory. It is important to note, then, that the data is in clear text the entire time it is in memory, and for performance reasons, pages of memory are often written to disk—and along with it the data in clear text. It is also possible that other operating system tasks, such as crash memory dumps, can result in the system memory (and therefore the data in clear text) being written to disk.

Notwithstanding, because Business Process Manager allows you to manually create the tables (which are specified in the results of the DbDesignGenerator tool), the Microsoft mechanism for achieving encryption of data at rest is compatible with Business Process Manager.

For more information, see *Database Encryption in SQL Server 2008*, which is available at:
<http://msdn.microsoft.com/en-us/library/cc278098%28v=sql.100%29.aspx>

Oracle

Oracle 10g r2 introduced a feature also called Transparent Data Encryption (TDE), although with Oracle this implies column-level encryption. This feature does not require any special lexemes within the SQL statement, neither for INSERT() nor SELECT() statements. The only requirement is that the table be created with the lexeme ENCRYPT following the data type of the column to be encrypted:

```
CREATE TABLE employees (
    id NUMBER(10),
    data VARCHAR2(50) ENCRYPT
)
```

Once defined in this way, the SQL statements require no special treatment:

```
INSERT INTO employees VALUES (101, 'this will be encrypted')
```

As with Microsoft SQL Server, you can go through the output of the DbDesignGenerator and select which columns you would like to have encrypted. You then need to manually alter the SQL table create statements to enable TDE. Therefore, Business Process Manager is compatible with the Oracle TDE feature.

For more information, see *Transparent Data Encryption (TDE) in Oracle 10g Database Release 2* at:

<http://www.oracle-base.com/articles/10g/transparent-data-encryption-10gr2.php>

Oracle 11g expands upon this concept, and offers “Tablespace” encryption. The Oracle documentation claims that this allows “encryption of the entire contents of a tablespace, rather than having to configure encryption on a column-by-column basis.”

Tablespaces which are to be encrypted are specified as follows:

```
CREATE TABLESPACE securespace
DATAFILE '/home/user/oradata/secure01.dbf'
SIZE 150M
ENCRYPTION USING '3DES168'
DEFAULT STORAGE(ENCRYPT);
```

The encryption cipher (or algorithm) used is Triple-DES, here specified with a key length of 168 bits.

For more information:

- ▶ *Tablespace Encryption in Oracle 11g Database Release 1*
<http://www.oracle-base.com/articles/11g/tablespace-encryption-11gr1.php>
- ▶ *Securing Stored Data Using Transparent Data Encryption*
http://docs.oracle.com/cd/B28359_01/network.111/b28530/asotrans.htm#ASOAG610

Because no change is needed to either the table create SQL statements or to the columns contained therein, Business Process Manager is compatible with the Oracle Tablespace Encryption feature.

Despite the caveats of performance and other considerations at the beginning of this section, the ability to encrypt your databases may well outweigh the costs mentioned. We have already discussed the benefits of eliminating the “low hanging fruit” in your security strategy, and we have another example of this with respect to encrypting your data at rest. It should be self-evident that those situations where the data is encrypted represent a vastly reduced opportunity for misuse than those where the data is left in clear text.

Operating system layer

Due to the concerns regarding cached tables and poor index performances, we find a large percentage of customers who choose to pursue encrypting their data at the operating system level.

Modern operating systems provide something called Encrypting File System (EFS). EFS enables transparent encryption and decryption of files, folders, or drives using a variety of cryptographic algorithms. Some people have found that they are able to use this feature to encrypt the database files on subdirectories or drives to a performance level which they deem acceptable.

A final word

The choice of which strategy to employ when encrypting your data at rest is nontrivial, and a decision should be made only after careful consideration of all of the options as well as performance impacts.

And above all else, do not keep the encryption keys *anywhere* near the data being encrypted. This is akin to putting bars on your windows, reinforcing door locks, and then leaving the key under the door mat.

2.2.4 Failure to use SSL between Process Center and Process Server

Another common security hole we see, which is related to the over-optimistic faith in firewalls, is a failure to use SSL between the Process Center and the various runtime Process Servers.

During the installation of a Process Server, you (or your IBM consultant) turned to the WebSphere Application Server Integrated Solutions Console and executed a wizard to create the Deployment Environment. This wizard includes a step where the Process Server specifies the host name of the Process Center it will be utilizing as its repository. This step is illustrated in Figure 2-3 on page 33.

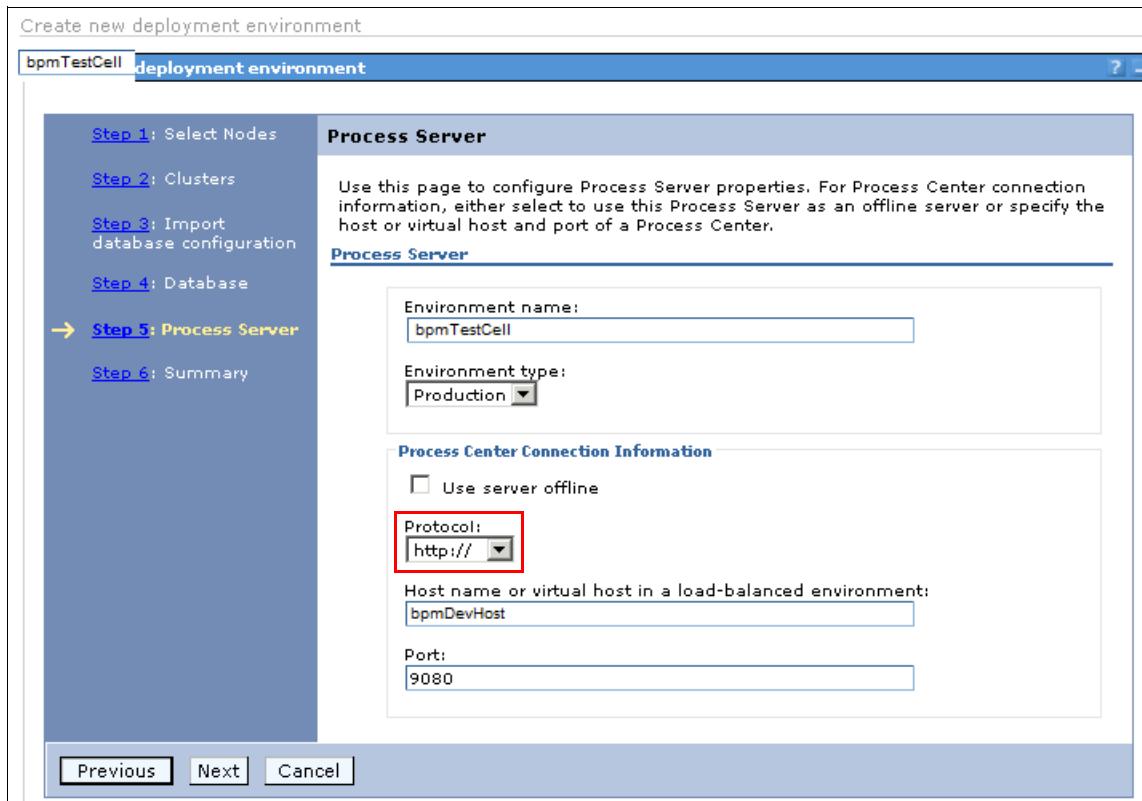


Figure 2-3 Setting the Process Server protocol and host name

Note that the Protocol defaults to http://. During Process Server startup, the runtime environment uses this information to communicate back to the Process Center, notifying the Process Center of the runtime Process Server's availability to receive deployments of process application snapshots. This communication between Process Server and Process Center includes a URL, a user account, and the corresponding password. This is all an attacker needs to know in order to deploy new snapshots of process applications—effectively changing the way you do business. An attacker could also deploy his favorite malware application, which monitors the network, carries out denial of service attacks and spreads other types of malware to other systems in the same network (not only systems your process applications connect to, but basically any systems that can be reached).

If you do not take the extra step to specify the https:// protocol in Figure 2-3, you will be sending your BPM admin account name and password in clear text.

2.2.5 Overuse of default Business Process Manager accounts

During Business Process Manager installation, it is common for us to see one BPM admin account used in every place where an account username and password are created. For example, Figure 2-4 and Figure 2-5 on page 34, and Figure 2-6 on page 35 show the default values for a number of messaging admin accounts.

Create new deployment environment

Create new deployment environment

Step 1: Select Nodes **Step 2:** System REST Service Endpoints **Step 3:** Import database configuration **Step 4:** Database **Step 5:** Security **Step 6:** Business Process Choreographer **Step 7:** Web Application Context Roots **Step 8:** Summary

Security

Edit the user names and passwords for the authentication aliases that are needed by this deployment environment.

Name	User name	Password	Confirm Password
Process Server JMS authentication alias	bpmadmin		
Performance Data Warehouse JMS authentication alias	bpmadmin		
CEI JMS authentication alias	bpmadmin		
SCA authentication alias	bpmadmin		
Business Process Choreographer JMS authentication alias	bpmadmin		

Previous | Next | Cancel

Figure 2-4 Default accounts (1/3)

Security

Role	Use Default	Users	Groups	Description
Administrator	<input type="checkbox"/>	bpmadmin		User names, group names, or both, separated by the " " symbol, for the business flow and human task administrator role. Users who are assigned to this role have all privileges.
Monitor	<input type="checkbox"/>	bpmadmin		User names, group names, or both, separated by the " " symbol, for the business flow and human task monitor role. Users who are assigned to this role can view the properties of all of the business process and task objects.

Figure 2-5 Default accounts (2/3)

Security					
Authentication	Users	Password	Confirm Password	Description	
JMS API Authentication	bpmadmin			Authentication for business flow manager message-driven bean to process asynchronous API calls	
Escalation User Authentication	bpmadmin			Authentication for human task manager message-driven bean to process asynchronous API calls	
Administration job user authentication	bpmadmin			Authentication for the business flow manager and human task manager administrative jobs. This user must be in the Administrator role.	

Figure 2-6 Default accounts (3/3)

The problem here is that we see all too often companies accept the software's default, resulting in a gross over usage of the bpmadmin account. We encourage you to take a step back and create more fine-grained accounts.

Creating accounts: We highly advise (when given the opportunity to do so) that you create account names that closely reflect the roles or responsibilities of that account's intended purpose.

We also suggest (and some standards, such as PCI-DSS, even mandate) that no human administrator ever use an account like bpmAdmin or tw_admin. Every person must have a personal account.

For example, in the final account shown in Figure 2-6, for the "Administration job user authentication" account, the account name, instead of using the default bpmadmin account, should be more reflective of the role that the account is intended to fulfill. Perhaps "jobAdmin" would be a more appropriate account name.

To drive the point home, Figure 2-7 on page 36 depicts a few of the places where the default bpmadmin account is used throughout the Business Process Manager environment. It is, quite naturally, a part of the tw_admins group—the group which should include all users who are granted administrative responsibilities. But it also shows that the account is part of the tw_authors group (software developers and business process application authors), and the process-center-install-group (those who are authorized to deploy snapshot process applications to the various runtime deployment environments).

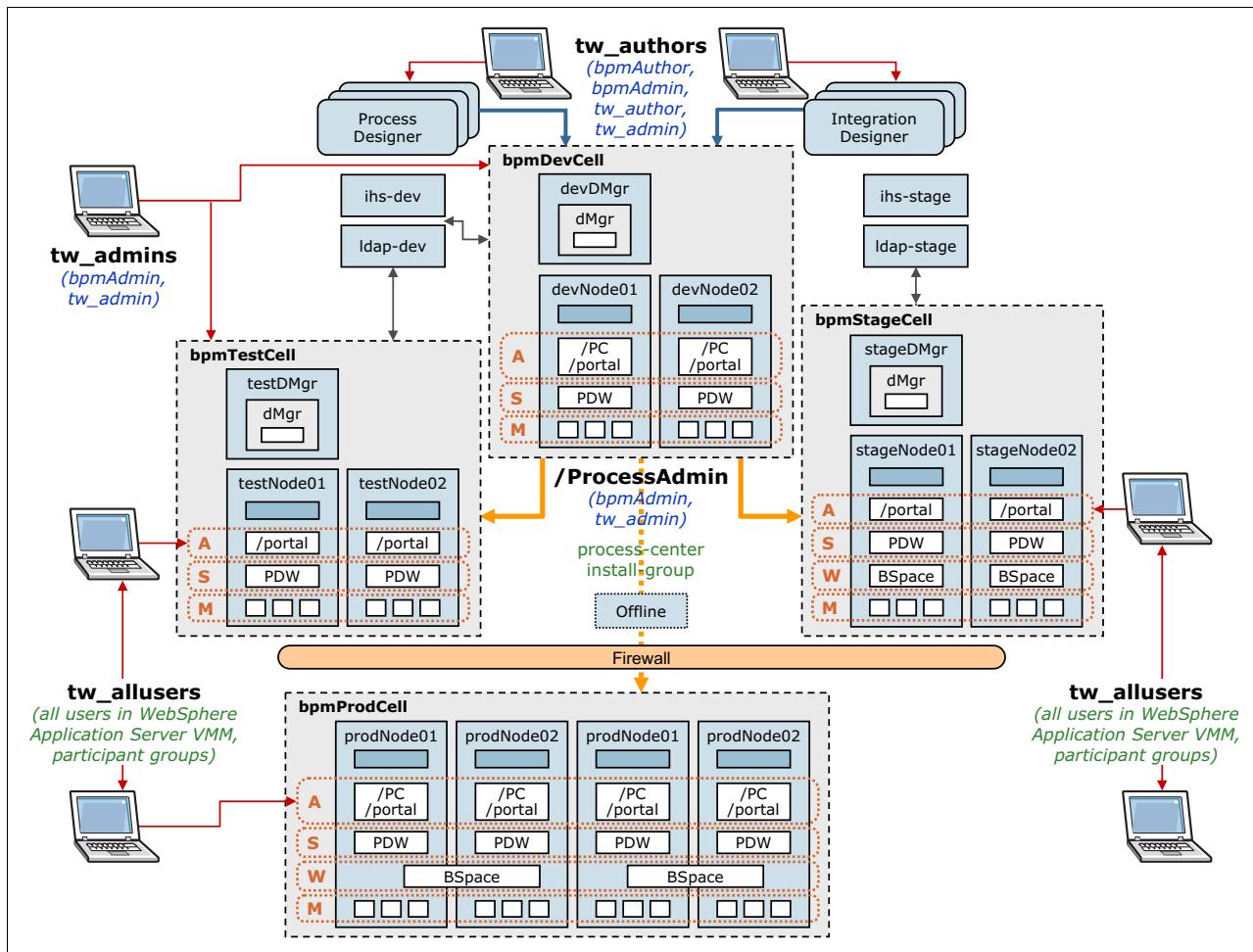


Figure 2-7 User accounts in Business Process Manager

The problem with not adopting the fine-grained approach to account naming as defined above is that a failure to do so promotes a loose attitude towards who gets access to the admin accounts. For example, if a person is given the bpmadmin account simply in order to deploy a snapshot to a runtime Process Server environment, then a consequence of this act is that the same person now has access to just about everything else in the Business Process Manager universe—he/she can view other process application definitions and alter them at will.

2.2.6 Overuse of trust in certificate authorities

Another problem to consider is overuse in the trust of certificate authorities. We look at this from the web browser, WebSphere Application Server, and Java models.

The web browser model

Recall our discussion of certificate authorities (CA) from Chapter 1, “Why Business Process Manager security is important” on page 1. The modern web browser ships with a plethora of certificate authorities. This is because the browser, as a tool, is designed to connect with as many websites as possible. This is most likely not the case with your Business Process Manager installation. Recall the following:

The certificate authority digitally signs the web server's certificate, and if the browser holds a certificate from the same CA which signed the web server's certificate, then the trust is established by basically saying in effect:

"if you trust this web server, then I'll trust it too".

There is no guarantee that certificate authorities fact check the identity of the parties who purchase certificates from them. It is hard to imagine that a \$12.95 certificate is subject to a great deal of scrutiny.

Tip: We advise that you reduce the number of certificate authorities in use within your organization to just the bare minimum needed.

The WebSphere Application Server model

The WebSphere Application Server certificate management functionality is based upon a well-thought-out strategy and utilizes a minimalist trust model. By default, WebSphere Application Server ships with just two certificates: one for WebSphere DataPower® and one for signing its own certificates. Business Process Manager adds two more for use with Bluworks Live™(Figure 2-8). This self-signing certificate acts as its own certificate authority, and is used by WebSphere Application Server nodes (via the nodeAgent processes) when communicating with the cell's deployment manager.

SSL certificate and key management > Key stores and certificates > CellDefaultTrustStore > Signer certificates				
Manages signer certificates in key stores.				
<input type="checkbox"/> Preferences Add Delete Extract Retrieve from port				
    				
Select	Alias ▾	Issued to ▾	Fingerprint (SHA Digest) ▾	Expiration ▾
You can administer the following resources:				
<input type="checkbox"/>	blueworkslive	CN=www.blueworkslive.com, OU=Application and Integration Middleware, O=INTERNATIONAL BUSINESS MACHINES CORPORATION, L=Austin, ST=Texas, C=US, SERIALNUMBER=DOC:19110616, OID.1.3.6.1.4.1.311.60.2.1.2=New York, OID.1.3.6.1.4.1.311.60.2.1.3=US, OID.2.5.4.15=Private Organization	EB:D1:6E:A0:6F:27:0A:D5:07:9C:77:9F:D5:16:38:8E:FA:42:A6:91	Valid from Oct 30, 2010 to Nov 2, 2012.
<input type="checkbox"/>	datapower	OU=Root CA, O="DataPower Technology, Inc.", C=US	A9:BA:A4:B5:BC:26:2F:5D:2A:80:93:CA:BA:F4:31:05:F2:54:14:17	Valid from Jun 11, 2003 to Jun 6, 2023.
<input type="checkbox"/>	root	CN=db2DevHost, OU=Root Certificate, OU=bpmDevCell, OU=bpmDevHostCellManager01, O=IBM, C=US	9C:0A:2B:0E:6E:3C:71:E1:9A:A0:31:39:BB:C1:66:45:B8:23:D4:BC	Valid from Apr 9, 2012 to Apr 6, 2027.
<input type="checkbox"/>	staticblueworkslive	CN=static.blueworkslive.com, OU=Application and Integration Middleware, O=INTERNATIONAL BUSINESS MACHINES CORPORATION, L=Austin, ST=Texas, C=US, SERIALNUMBER=DOC:19110616, OID.1.3.6.1.4.1.311.60.2.1.2=New York, OID.1.3.6.1.4.1.311.60.2.1.3=US, OID.2.5.4.15=Private Organization	EE:C3:4E:AC:A6:97:9C:7E:16:8B:1B:C1:50:9D:39:9B:D6:37:0A:6E	Valid from Oct 31, 2010 to Nov 1, 2012.
Total 4				

Figure 2-8 WebSphere Application Server signed certificates

If you do not use WebSphere DataPower in your installation, you can safely delete this certificate. If you do not use Blaworks Live, you can delete those two certificates as well. This would leave you with only one certificate. This strategy makes it difficult to impersonate one of the nodes of a WebSphere cell.

Minimize certificates: We advise that you start with a bare minimum of certificates, and add only those that you need for specific purposes.

The Java model

Business Process Manager is a montage of former IBM WebSphere Process Server and Lombardi Teamworks. The Teamworks portion of Business Process Manager is designed to work with a variety of commercially available application servers. One of the consequences of this “support many” strategy is evident in its use of the Java cacerts file for certain security functions.

Similar to the browser’s motivation for supporting a plethora of certificate authorities, the Java JDK cacerts file ships with 77 signer certificates (a certificate representing a certificate authority). See Figure 2-9.

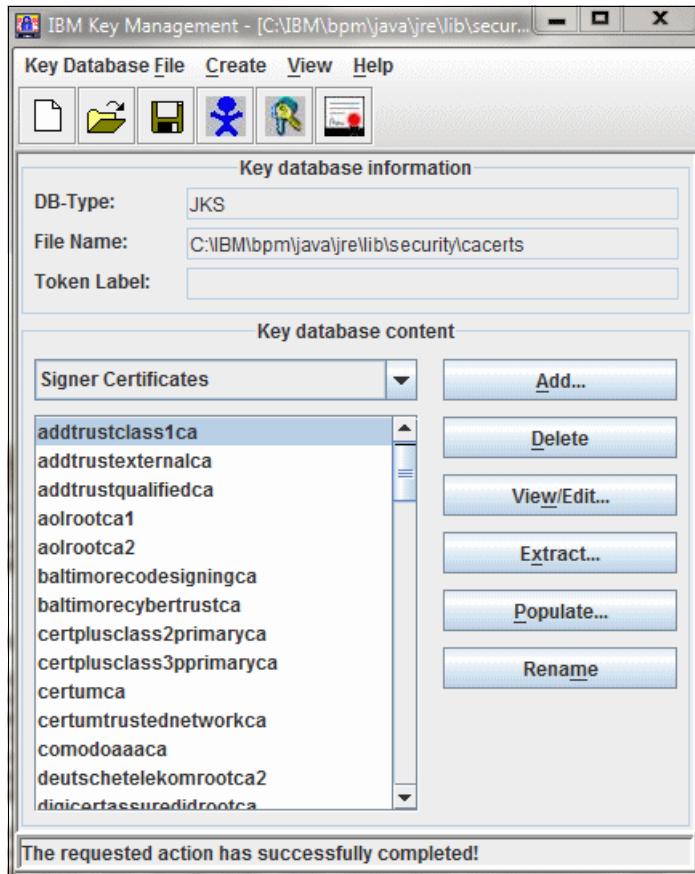


Figure 2-9 Java signed certificates

This is quite the opposite of the “minimalist trust” model espoused by WebSphere Application Server. In effect, for those portions of Business Process Manager which still rely upon cacerts, you are opening up trust to any server that has purchased a certificate from *any one* of these 77 included certificate authorities.

An attack based upon this fact is complex and multi-faceted, but an attack such as this would be made significantly harder if you simply eliminated all of the signer certificates that are not in use by your organization. Again, as mentioned earlier, much of security hardening is the elimination of “low hanging fruit”—and this is a prime example.

In addition, the cacerts file which ships with Business Process Manager has the Java-defined default password—“change it”. Ironically, even though the default password is “change it”, few ever do.

Change the cacerts password: Do change the cacerts password, and remove from it all signer certificates that you do not use. It may appear strange when you first look at it: after changing the cacerts password there is nothing to do in Business Process Manager. You do not need to tell it the new password. The reason is that Business Process Manager uses cacerts as a trust store and therefore only reads public certificates. No password required to do that.

However, do change the password so that nobody else can add a certificate that you do not want to trust. The only time you have to configure the new password in Business Process Manager (using a JVM custom property) is when you have scenarios with client certificate authentication when WebSphere Application Server needs to read a private key out of cacerts. For this type of interaction, the changed password needs to be known in WebSphere Application Server.



Authentication: Who has access

Authentication is the process of proving the identity of the user who (or even another computer system which) is requesting access to software.

The most common type of authentication is, of course, the userid and password. But there are other methods of authenticating to a server. Simply put, here are three categories of authentication:

- ▶ What you know - passwords, session IDs
- ▶ What you have - digital certificates, hardware passcode generators
- ▶ What you are - biometrics such as fingerprints and retinal scans

For each of these types of authentication, one needs to consider the following:

- ▶ Where is the system of record that holds user accounts?
- ▶ How will the authentication information communicate with this system?
- ▶ How will you detect a compromise?
- ▶ How will you revoke access?

In light of recent research showing the high incidence of security attacks that originate from employees or contractors, the choice of how users authenticate to your corporate systems should be evaluated carefully. Often, the choice of how users authenticate to corporate systems is decades old, and it might well be worth reviewing the policies at your organization.

Regardless of which authentication mechanism (or combination of same) your organization uses, the ensuing steps are the same: the user (or other computer system) presents some authentication information, WebSphere Application Server validates that this information is correct and current, and then WebSphere Application Server creates a security context (a subject and principal) on behalf of the user which accompanies all future requests during the course of the session.

After the previous chapter's focus on installation and foundational issues, this chapter shall be considered your second step in securing your Business Process Manager environments. In this chapter we investigate who has access to your Business Process Manager applications.

3.1 Subjects and Principals

The definitions of the terms Subject and Principal appear to vary from one technology and/or author to the next, but for our purposes we can simplify this by asserting the following distinction:

- ▶ A Subject is any actor who is requesting access to some object
- ▶ A Principal is a way of representing who this actor is

So, for example, if you (the Subject) attempt to log into a WebSphere Application Server protected web page (the object), you may type in your user ID and password (a Principal). Or, if your environment has been so configured, your browser may present a Kerberos token (a Principal) which your operating system created upon your behalf when you logged into your laptop. Furthermore, after you have authenticated to this website, the web server may inject a session cookie (a Principal) into your browser which uniquely identifies your identity during the course of your website session. All of these principals are credentials which WebSphere Application Server can use to authenticate you, and they are assembled under a security context (a Subject) which persists over the course of your web session.

In order for the credentials to be authenticated by WebSphere Application Server, WebSphere Application Server must first have a list of users against which these credentials can be compared. At the core of user authentication, then, lies the system of record where the user information is stored.

3.2 WebSphere user registry

Business Process Manager relies entirely upon the WebSphere Application Server mechanism for user authentication. Before an individual can authenticate to WebSphere Application Server, that user must exist within a list which has been configured for this purpose with WebSphere Application Server. This “list” is sometimes called the WebSphere user registry, and at other times, the WebSphere user realm. We use the term *user registry* in this book.

When a user logs in, his/her credentials are presented to WebSphere Application Server, which in turn somehow compares these against the user’s information stored within the user registry. The types of lists that the WebSphere user registry allows can be categorized into the following four types:

- ▶ A flat-file repository
- ▶ A stand-alone LDAP repository
- ▶ A custom software repository
- ▶ Any combination of these, federated together as one

First, a word about repositories and registries. These two terms are often confused. A repository is a concrete set of items (in this case, the “list” of user accounts). A registry is a reference to something (in this case, the repositories within the user registry). It may be helpful to visualize this as an analogy: a library’s card catalog is a registry, the library’s stacks of books is its repository.

We strongly advise using the Federated Repositories user registry. In Federated Repositories you have the ability to connect any mixture of heterogeneous repositories, including flat file and LDAP servers, or multiple LDAP servers.

The following discussion regarding the types of repositories is applicable for each of your Business Process Manager environments (Figure 3-1 on page 43). Once you have decided

upon the type of user registry you will be using, you will need to configure each Business Process Manager environment accordingly using each environment's Integrated Solutions Console (/ibm/console). The following example adds the flat-file repository to Federated Repositories.

The screenshot shows the 'Integrated Solutions Console - Mozilla Firefox: IBM Edition' window. The left sidebar menu includes 'View: All tasks', 'Welcome', 'Guided Activities', 'Servers', 'Applications', 'Services', 'Resources', 'Security' (selected), 'Business Integration Security', 'Global security', 'Security domains', 'Administrative Authorization Groups', 'SSL certificate and key management', 'Security auditing', 'Bus security', and 'JAX-WS and JAX-RPC security runtime'. Other sections like 'Environment', 'Integration Applications', 'System administration', 'Users and Groups', 'Monitoring and Tuning', 'Troubleshooting', 'Service integration', and 'UDDI' are also listed.

The main panel displays 'General Properties' for a realm named 'defaultWIMFileBasedRealm' with the primary administrative user set to 'bpadmin'. Under 'Server user identity', the 'Automatically generated server identity' option is selected. There is a checkbox for 'Ignore case for authorization' which is checked. The 'Repositories in the realm:' section lists four entries: 'o=defaultWIMFileBased...' (InternalFileRepository, File), 'o=region1-prod' (Apache Dir Server, LDAP:CUSTOM), 'o=region2-prod' (Active Directory, LDAP:AD), and 'o=region3-prod' (Tivoli Dir Server, LDAP:IDS). At the bottom are 'Additional Properties' and 'Related Items' sections, along with 'Apply', 'OK', 'Reset', and 'Cancel' buttons.

Below the console window is a diagram illustrating the repository structure across three Business Process Manager environments:

- bpmDevCell:** Contains 'devDMgr' (with 'dMgr') and two nodes, 'devNode01' and 'devNode02'. Each node has a '/PC' directory containing a '/portal' subdirectory, which in turn contains 'PDW' and 'BSpace' components. A dashed orange box labeled 'A S M' highlights this structure.
- bpmTestCell:** Contains 'testDMgr' (with 'dMgr') and two nodes, 'testNode01' and 'testNode02'. Each node has a '/PC' directory containing a '/portal' subdirectory, which in turn contains 'PDW' and 'BSpace' components. A dashed orange box labeled 'A S M' highlights this structure.
- bpmStageCell:** Contains 'stageDMgr' (with 'dMgr') and two nodes, 'stageNode01' and 'stageNode02'. Each node has a '/PC' directory containing a '/portal' subdirectory, which in turn contains 'PDW' and 'BSpace' components. A dashed orange box labeled 'A S W M' highlights this structure.

Figure 3-1 Repositories for each Business Process Manager repository

Each Business Process Manager deployment environment (each WebSphere Application Server cell) will require that you use the Integrated Solutions Console (/ibm/console) in each environment to configure the WebSphere user registry.

3.2.1 Flat-file repositories

Business Process Manager ships and installs with a single federated registry, which is called Federated Repositories. At installation time for Network Deployment environments, it is comprised of just one single flat-file repository, called the “defaultWIMFileBasedRealm”. See Figure 3-2.

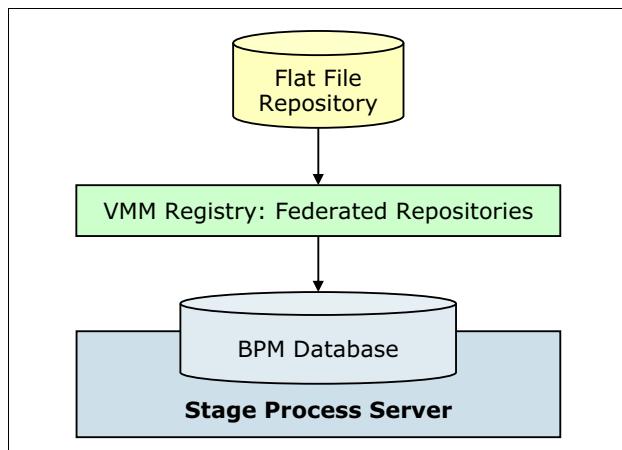


Figure 3-2 Flat-file repository

The flat-file repository was originally provided as a mechanism to facilitate the immediate securing of a WebSphere Application Server environment at install time—by providing user IDs and passwords for administrative accounts and associated functionality. Without such a mechanism, there would be a gap in security before the WebSphere Application Server installation could be integrated into an organization's already existing security infrastructure.

The flat-file repository is not intended to scale beyond 200-300 users, but it does provide usefulness even in large-scale production environments.

Using the Integrated Solutions Console

You can inspect the WebSphere user registry by launching the Integrated Solutions Console. This is also commonly called the “admin console”, and is exposed as a J2E application using the context root /ibm/console. Navigate to **Security → Global security** and you should see the panel shown in Figure 3-3 on page 45.

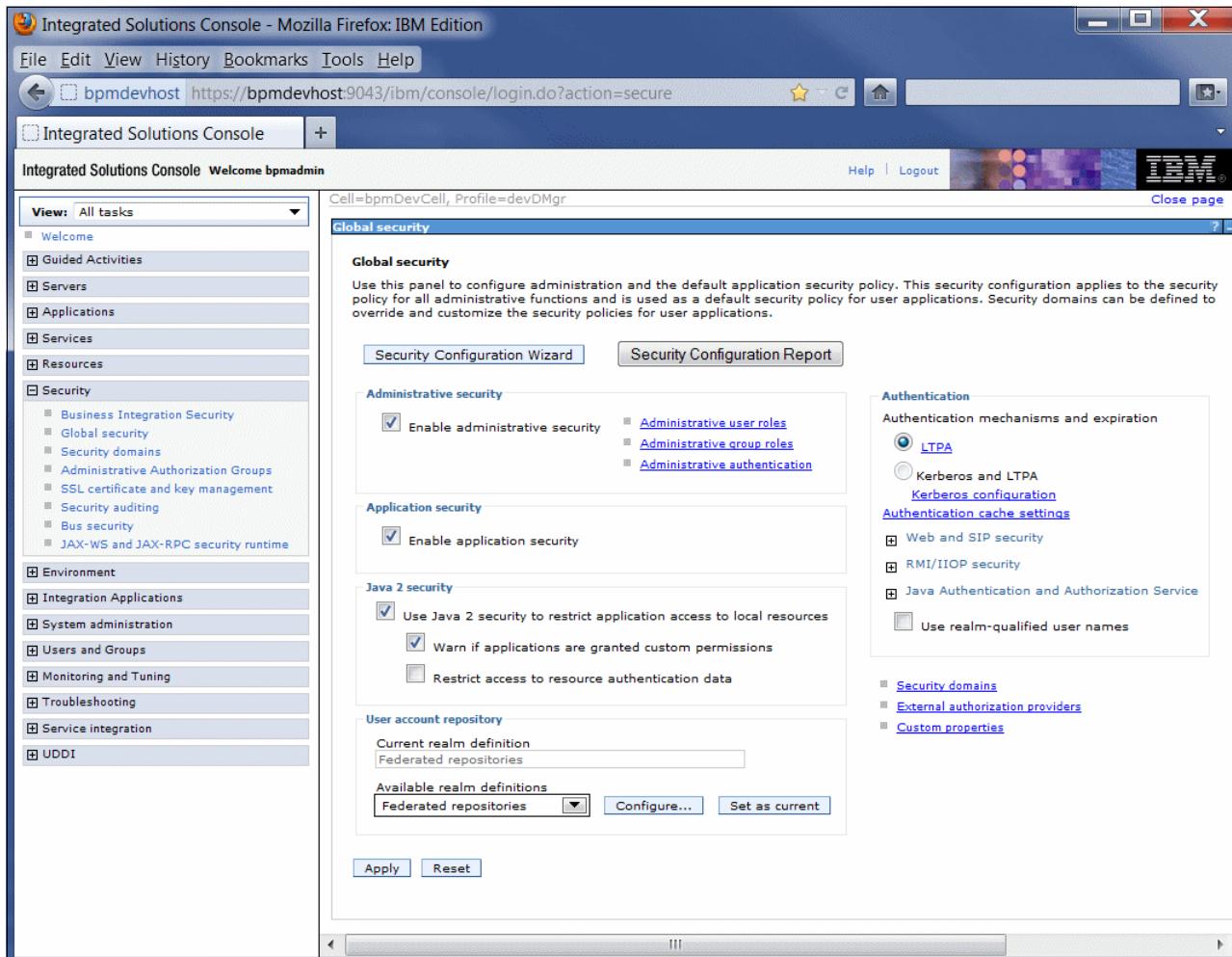


Figure 3-3 Integrated Solutions Console

Clicking **Configure** within the “User account repository” section will bring you to the list of repositories that are members of the Federated repositories. In this example, only the one default repository has been federated (o=defaultWIMFileBasedRealm), as shown in Figure 3-4 on page 46.

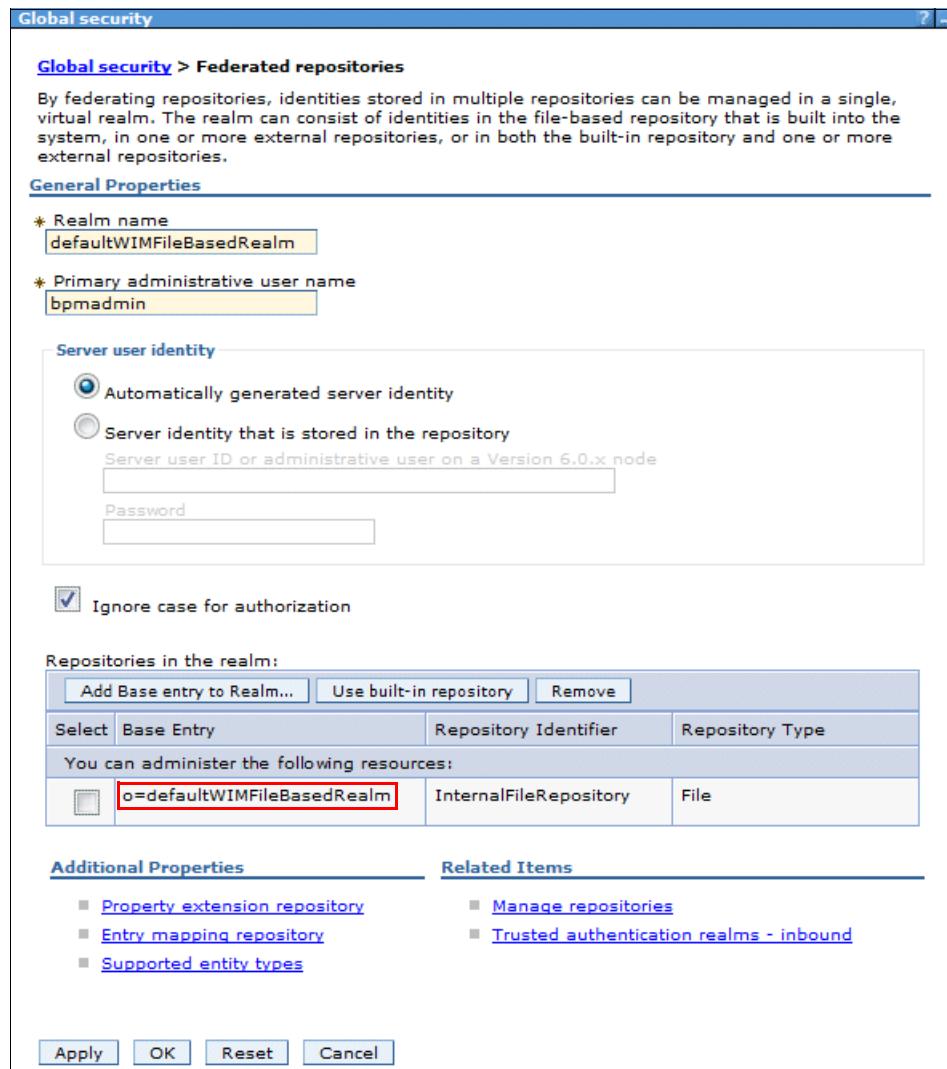


Figure 3-4 Federated repositories

The Integrated Solutions Console does not provide you with an opportunity to change any parameters associated with the flat file. To view the defaultWIMFileBasedRealm flat file repository, you need to view the file that is stored in an XML file at the following location:

<BPM install>/profiles/<ProfileName>/config/cells/<CellName>/fileRegistry.xml

Figure 3-5 on page 47 shows an example fileRegistry.xml file.

```

<?xml version="1.0" encoding="UTF-8"?>
<xml.type:datagraph
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wim="http://www.ibm.com/websphere/wim"
  xmlns:xml.type="commonj.sdo">
  <wim:Root>
    <wim:entities xsi:type="wim:PersonAccount">
      <wim:identifier
        externalId="e13049f2-e174-4658-b6ad-8953a1fd24a7"
        externalName="uid=bpmadmin,o=defaultWIMfileBasedRealm"
        uniqueId="e13049f2-e174-4658-b6ad-8953a1fd24a7"
        uniqueName="uid=bpmadmin,o=defaultWIMfileBasedRealm"/>
      <wim:parent>
        <wim:identifier uniqueName="o=defaultWIMfileBasedRealm"/>
      </wim:parent>
      <wim:createTimestamp>2012-04-10T20:42:35.373Z</wim:createTimestamp>
      <wim:password>U0hBLTE6YTNrMWFsdXM3aGh0OkxaSUFzNVpiTVhEcW5uZHbzUwdLY3hVtwtMMD0NCg==</wim:password>
      <wim:uid>bpmadmin</wim:uid>
      <wim:cn>bpmadmin</wim:cn>
      <wim:sn>bpmadmin</wim:sn>
    </wim:entities>
    <wim:entities xsi:type="wim:PersonAccount">
      <wim:identifier
        externalId="ed490ebe-4f53-4c80-84f0-82bef04eec15"
        externalName="uid=tw_admin,o=defaultWIMfileBasedRealm"
        uniqueId="ed490ebe-4f53-4c80-84f0-82bef04eec15"
        uniqueName="uid=tw_admin,o=defaultWIMfileBasedRealm"/>
      <wim:parent>
        <wim:identifier uniqueName="o=defaultWIMfileBasedRealm"/>
      </wim:parent>
      <wim:createTimestamp>2012-04-10T20:53:25.974Z</wim:createTimestamp>
      <wim:password>U0hBLTE6NDVw0W1zc3ByNWQ3OnRvbGFrMHpxMz1KNG5Hd3pwVU1iU0FHbkNRVT0NCg==</wim:password>
      <wim:uid>tw_admin</wim:uid>
      <wim:cn>tw_admin</wim:cn>
      <wim:sn>tw_admin</wim:sn>
    </wim:entities>
  </wim:Root>
</xml.type:datagraph>

```

Figure 3-5 Example fileRegistry.xml file

Notice that this sample file contains both the Business Process Manager default account that was specified at profile creation (bpmadmin) as well as the default account tw_admin. Also note that the passwords are hashed using SHA-1.

Usefulness of the default flat-file repository

IBM offers a program called a Quick-Win Pilot, where the business value of the Business Process Manager product and IBM methodology are proved within a very short timeframe. The Quick-Win Pilot installations are often isolated, temporary, proof-of-concept installations. They are not intended to become a part of the enterprise's IT infrastructure. It is common to see the flat-file repository used in these situations, and the user data then migrated to the corporate user repository once the business value of the product is proved.

Note: We advise keeping some administrative accounts in the flat-file repository, and the repository in turn federated into the user registry—to be used in the case of an inability to connect to the primary security mechanism.

For example, if the network connection or the primary LDAP server is down, WebSphere Application Server administrators would still be able to log into the Integrated Solutions Console using an administrator account stored in the flat-file repository.

The other advantage is that by using the flat file repository, you have no need of creating tw_admin in your LDAP computation. Imagine a company where all developers have a Business Process Manager system installed on their laptop and connect to corporate LDAP for getting users and groups (and maybe attributes). If tw_admin was in LDAP, everybody

would have the same password for tw_admin—the same one as any production system in the worst case.

3.2.2 LDAP repositories

By far the most common type of user list in use by corporations today is the LDAP repository. It is also commonly called a *directory* of user information. It is often described as a database, but be aware that LDAPs are a specialized type of database with characteristics that set them apart from general purpose relational databases.

Optimized for read-only access

One of the most important specialized characteristics is that an LDAP repository is read much more often than it is updated. Applications and other users might frequently look up an employee's userid, name or phone number, but these values will rarely change for this employee. This fact gives rise to the following generalizations:

- ▶ LDAPs are optimized for read-access.
- ▶ LDAPs store static information.
- ▶ LDAPs do not store application-specific data.

Unlike a general purpose database, which might be used for any number of high-volume high-update applications (such as an airline reservation system), an LDAP server can be optimized for read (and search) access. In fact, it is fairly common for the administrators of the LDAP repository to have strict guidelines over which software applications are even *allowed* to write to the repository.

Often, corporate LDAP systems are considered the System-of-Record for employee data. Because of this, LDAP administrators are typically very careful about what type of data should be stored in the LDAP.

Consider Business Process Manager. Every Business Process Manager user has an inbox that receives business process tasks during the course of a typical work day. But even in an organization whose BPM adoption is mature, where potentially every employee listed in the corporate LDAP might share this same requirement, still it is not a good idea to store the inbox nor process application task lists in the LDAP. Even though they all share the same requirement, the tasks themselves will vary from person to person and will most likely need updating several times per day—violating the very nature of the LDAP concept.

LDAP groups and ACLs

However, many organizations do have grouping information stored in the LDAP repository. These groups can be along geographic, organizational, or business functional lines. Perhaps the groups reflect other business meaningful parameters such as employees who joined as a result of an acquisition. In each of these cases, the presupposition is that the individual employees are not likely to change their group affiliations frequently, and therefore this information is appropriate for an employee System-of-Record such as the LDAP repository.

Furthermore, some LDAP repositories support the notion of an Access Control List (ACL). This is somewhat related to the LDAP groups, and can be used to effectively limit software program access along business lines.

The use of LDAP groups is covered in Chapter 4, “Authorization: Access to what” on page 65.

LDAP Directory Information Tree

LDAP repositories contain a collection of objects that are organized in a tree structure. The LDAP naming model defines how entries are identified and organized. The tree structure is

called the Directory Information Tree (DIT). Entries are arranged in the DIT based upon their Distinguished Name (DN), a unique name that unambiguously identifies one single entry within the LDAP structure. The DN corresponds to the branches of the DIT that lead from the root to the entry in question, with each branch listed and separated by commas. These branches can be representative of geographic, organization, or business functional lines.

These branches can be composed of any number of distinguishing features. The most common “branches” are:

- ▶ dc (domain component)
- ▶ o (organization)
- ▶ ou (organizational unit)
- ▶ cn (common name)
- ▶ uid (unique identifier)

Figure 3-6 shows what an LDAP DIT looks like from the point of view of an LDAP browser. This is a very small organization (called region1) that contains two groups and eight users.

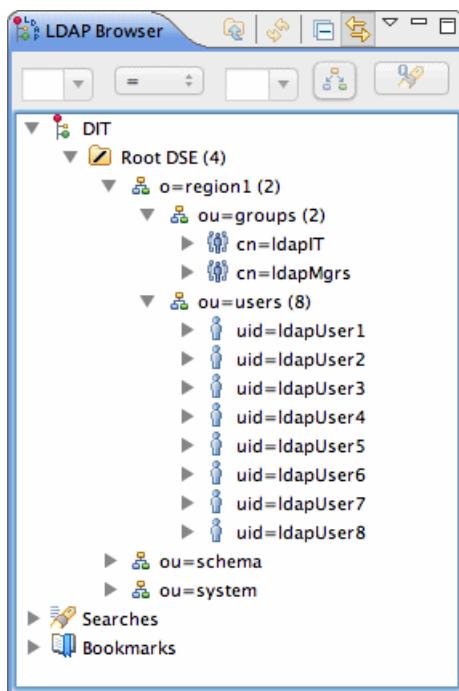


Figure 3-6 LDAP DIT

So, for example, Figure 3-6 shows the following LDAP entries:

- ▶ cn=ldapMgrs, ou=groups, o=region1, dc=companyA, dc=com
- ▶ uid=ldapUser1, ou=users, o=region1, dc=companyA, dc=com

Note that there is no strict rule on how an organization chooses to use these branch names. Some groups may be organizational units (ou) or they may be defined as common names (cn).

Connecting an LDAP to WebSphere Application Server

WebSphere Application Server can connect to any LDAP server that is v3 (RFC 2251) compliant. This currently includes:

- ▶ IBM Tivoli® Directory Server
- ▶ Sun Java Directory Server

- ▶ Lotus® Domino® Enterprise Server
- ▶ Microsoft Windows Active Directory
- ▶ Novell eDirectory
- ▶ IBM z/OS® Integrated Security Services

For any LDAP V3 specification-compliant servers that are not listed above, you will simply need to configure the LDAP server using the WebSphere Application Server “custom” LDAP option. You will most likely need to obtain appropriate filter and object class information from your LDAP vendor.

Most corporations have an LDAP team, with their own administrators and security policies, and so you will almost certainly need to work with them during the task of configuring WebSphere Application Server to work with the LDAP repository. In this section, we look at a simple LDAP instance and take a look at how one would integrate this into WebSphere Application Server.

The first tool you need when working with LDAP repositories is an LDAP browser. There are several that are freely available on the Internet. In this book, we use the Apache Directory Studio. It is written in Java and runs on Windows, Linux and Mac OS X. It is available at:

<http://directory.apache.org/studio>

Figure 3-7 is a screen shot of Apache Directory Studio, after having been connected to an Apache Directory Server LDAP repository.

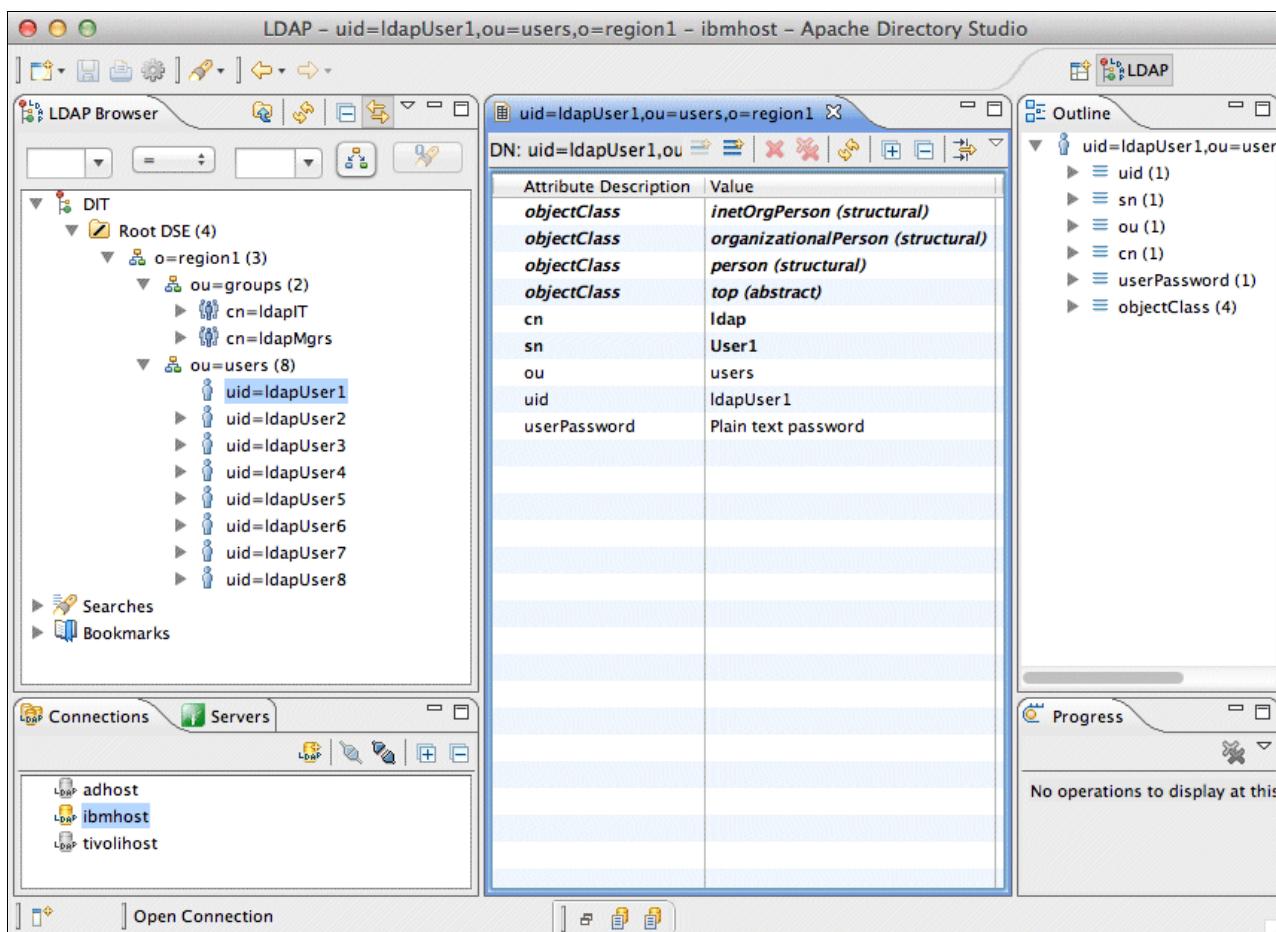


Figure 3-7 Apache Directory Server LDAP repository

The left-most section shows one organization (`o=region1`), with two organizational units underneath (`ou=groups`, `ou=users`). This is a very small, highly simplified LDAP repository, but over the course of this chapter, it will serve to demonstrate some very sophisticated WebSphere Application Server and Business Process Manager capabilities. The middle section shows details about one LDAP entry (`uid=ldapUser1`). The right section shows an outline view of that same entry, which can be very helpful once the LDAP repository structure grows more complex.

Figure 3-8 shows an example Integrated Solutions Console view of a custom LDAP being defined.

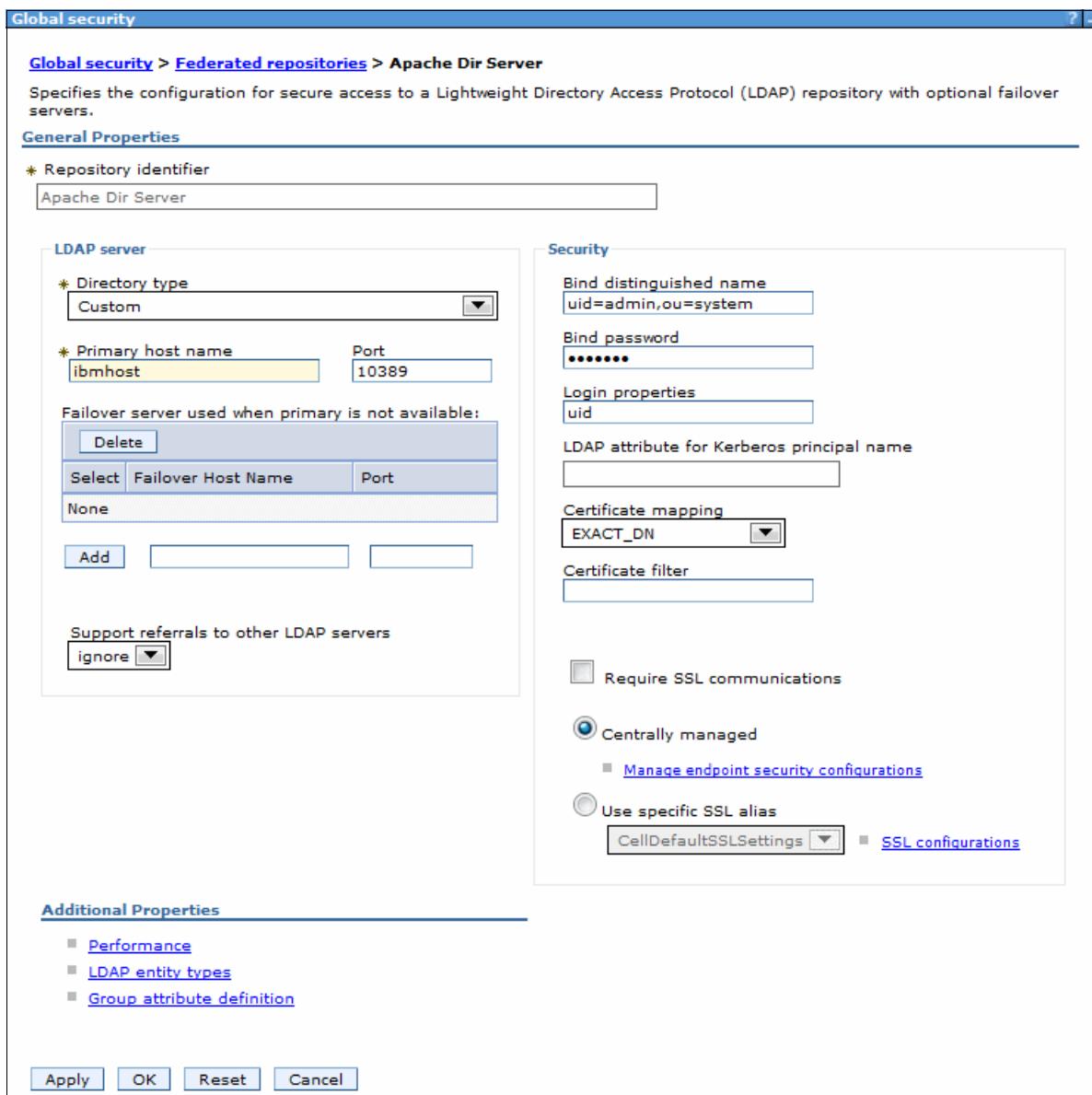


Figure 3-8 Apache Directory Server configuration in the Integrated Solutions Console

Notice that the Directory type is listed as Custom, and that in this example there is a value of "uid=admin,ou=system" for the Bind distinguished name. The Bind distinguished name is an LDAP account that has sufficient authorization (when presented together with the Bind password) to make any query against the LDAP repository. As was mentioned previously,

LDAP administrators are very selective about who is given access to this Bind DN, and so this account is almost always defined as read-only access.

Towards the bottom of Figure 3-8, you can see a link for “LDAP entity types”. Clicking this link will take you to a list of the types of objects which this LDAP server returns when queried (Figure 3-9 on page 52).

The screenshot shows a web-based administrative interface for managing LDAP entity types. At the top, there is a header bar with the URL "Cell=bpmDevCell, Profile=devDMgr". Below this is a navigation bar with the path "Global security > Federated repositories > Apache Dir Server > LDAP entity types". A sub-header below the path reads: "Use this page to list entity types that are supported by the member repositories or to select an entity type to view or change its configuration properties." There is a "Preferences" button with a gear icon. The main content area has two dropdown menus: "Entity Type" and "Object Classes". Below these menus, a section titled "You can administer the following resources:" lists three entries:

Entity Type	Object Classes
Group	groupOfNames
OrgContainer	organization;organizationalUnit;domain;container
PersonAccount	inetOrgPerson

At the bottom of the list, it says "Total 3".

Figure 3-9 LDAP entity types

These are by far the most common three types: an organizational container, groups, and persons. Each LDAP vendor is free to choose different names for these entity types, as well as the object classes that define these types, but the values shown in this example are very common. If you are not using one of the out-of-the-box WebSphere Application Server supported LDAP vendors, then you need to contact your LDAP vendor to verify the correct values to use for these types.

For more information, see *Understanding LDAP - Design and Implementation*, SG24-4986.

For the Apache Directory Server, these default values worked just fine, and so no further changes were necessary.

This discussion of LDAP servers is appropriate to this section on WebSphere Application Server-supported stand-alone LDAP servers, but there are limits to choosing a stand-alone LDAP server as your sole repository. The IBM Virtual Member Manager is a software layer that provides an abstraction above the normal LDAP specification, and provides additional functionality which the Business Process Manager product uses.

Note: We advise federating the LDAP server into the default registry, even if it is a single (or stand-alone, or dedicated) LDAP server.

3.2.3 Custom software repository

Despite the fact that the overwhelming majority of customers use an LDAP repository, situations can exist where a corporation's user and group data resides in other repositories or custom user registries, such as a database, and moving this information to either a local operating system registry or a dedicated LDAP registry implementation might not be feasible. For these situations, WebSphere Application Server security does provide a Service Provider Interface (SPI) that you can implement to interact with your current registry. This custom

registry feature can be developed to support virtually any user registry that is not implemented by WebSphere Application Server.

However, note that developing a custom software registry is decidedly non-trivial, and developing one which is truly secure is even far more complex than that. There are a number of reasons for this, but chief amongst them is that the WebSphere Application Server security mechanisms are initialized and enabled well before other WebSphere Application Server services, such as data sources and enterprise beans. There will therefore be a great deal of software that you will need to write on your own—effectively recreating wheels.

In addition:

- ▶ The SPI must be completely implemented, including error scenarios.
- ▶ You need to consider and implement your own mechanisms for availability and failover.
- ▶ Portions of Business Process Manager require additional security extensions that are above and beyond the WebSphere custom user registry SPI.
- ▶ Any custom registry should undergo extensive and rigorous third party penetration and security vulnerability testing.

Note: We advise that customers find a way to import their custom user and group data into a dedicated LDAP server, perhaps using LDIF or other means, and then federate this single-purpose LDAP server into the WebSphere user registry.

3.2.4 Federated repositories

Business Process Manager's preferred user registry is called Federated Repositories to point out the fact that it is a collection of independent repositories. It is built upon a sophisticated software layer called the Virtual Member Manager (VMM). The VMM allows for tremendous flexibility. Multiple repositories, of various types, can be joined together in the federation, and they act as one (Figure 3-10).

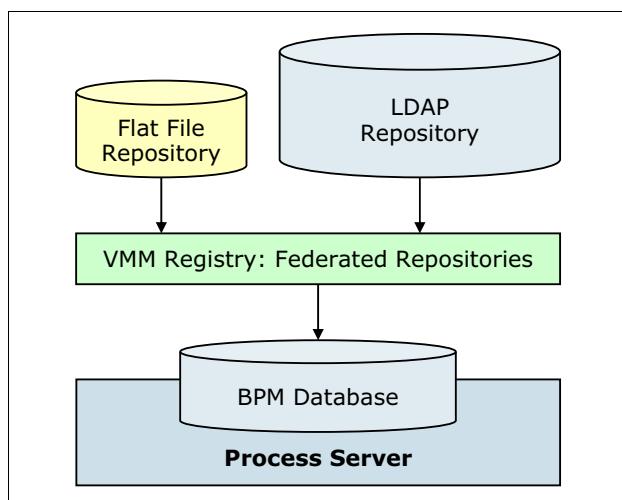


Figure 3-10 Federated user repository

Figure 3-10 depicts a very simple case: one federated registry consisting of one flat-file repository and one LDAP repository. As mentioned before, WebSphere Application Server is compatible with a variety of LDAP vendor's products. Each LDAP vendor may define a different reserved word to access a particular function, but WebSphere Application Server's

VMM can be configured to abstract these differences and therefore work with any LDAP server that is LDAP V3 compliant. WebSphere Application Server also ships with several LDAP templates to facilitate installation.

It is equally common for customers to have more than one LDAP server. This is often the case due to acquisitions or geographical dispersion. The acquired companies may well have a host of software systems that rely upon their existing LDAP structures, and there is always a period of integration where old systems are updated or replaced. WebSphere Application Server allows for this by federating LDAP repositories together. In Figure 3-11, we show an example where the VMM federated registry is composed of a single flat-file repository, one generic LDAP V3 server, plus a series of Microsoft Active Domain repositories arranged in a tree structure. Even more complicated arrangements are possible, but a complete treatment of this topic is not relevant to the purpose of this book. For more information, see *Understanding LDAP – Design and Implementation*, SG24-4986.

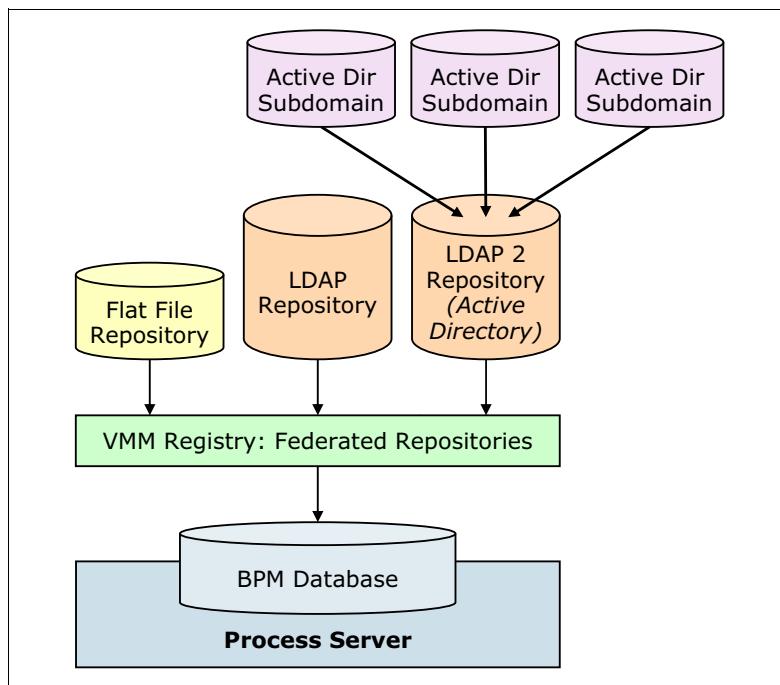


Figure 3-11 VMM registry

When you have multiple repositories federated together, the Integrated Solutions Console will show these as additional rows in the “Repositories in the realm” section (Figure 3-12 on page 55).

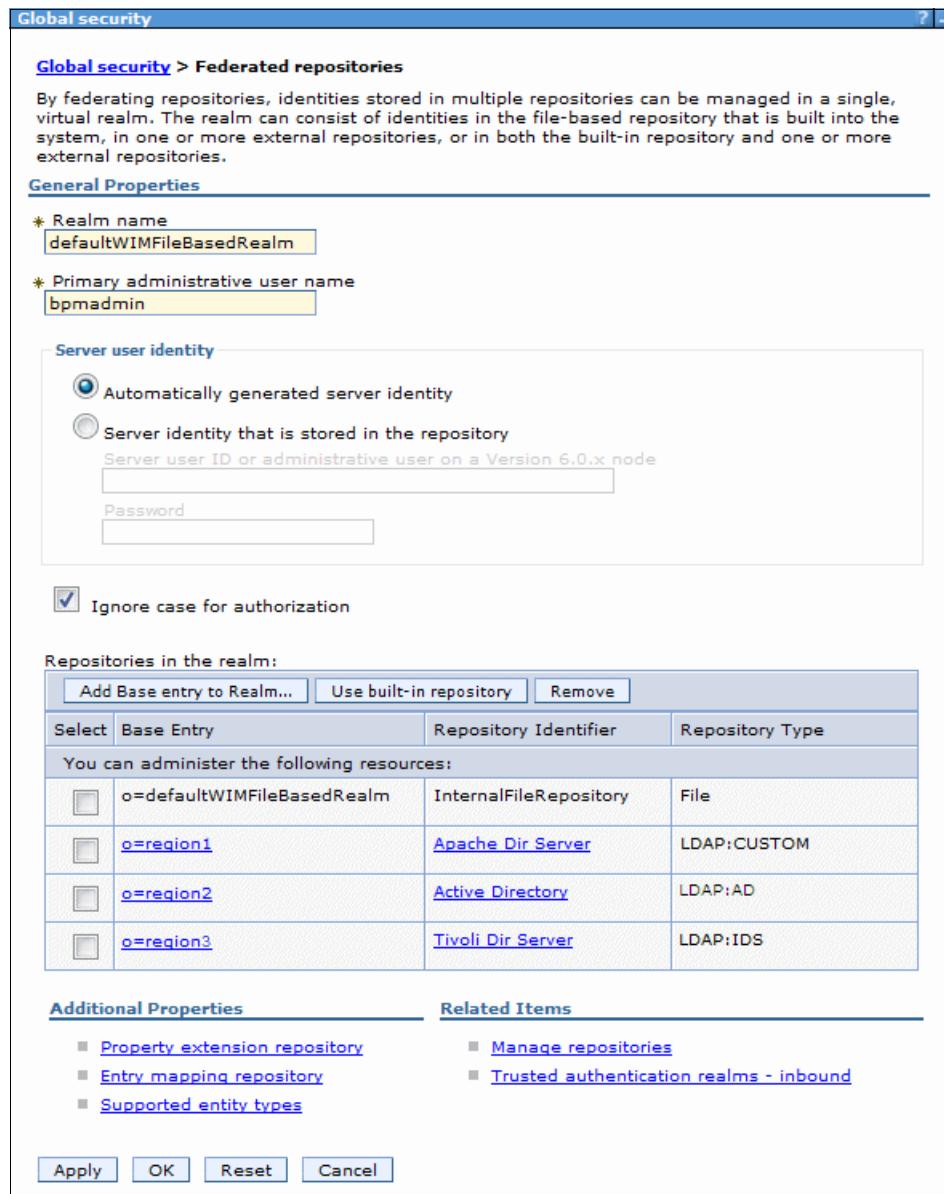


Figure 3-12 Repositories in the realm

In this example window, you can clearly see that the default flat-file repository (`o=defaultWIMFileBasedRealm`), which ships with Business Process Manager, has been federated together with three different LDAP repositories, each from a different vendor. An Apache Directory Server (`o=region1`) has been federated together with a Microsoft Active Directory server (`o=region2`) as well as with an IBM Tivoli Directory Server (`o=region3`).

As you can tell, the WebSphere user registry is a very powerful and flexible mechanism.

Notwithstanding, the one major constraint to using a federated registry is that any given user ID must exist in one and only one repository. The reason is simple: if one user ID were to exist in more than one repository, how would WebSphere Application Server know which account to authenticate—the user ID as stored in repository 1, or the one in repository 2? Although the two accounts may share a user ID, they are distinct, separate accounts. For this reason, any user ID that exists in more than one repository is always denied authentication.

User IDs: We advise that you keep it simple: each user ID exists in one and only one repository.

3.3 Common security holes

In this section we discuss common security holes we see with authentication:

- ▶ Weak password policies
- ▶ Failure to change default passwords
- ▶ Faith in firewalls
- ▶ Insecure LDAP connections
- ▶ Insecure SSO solutions

3.3.1 Weak password policies

When asked to create a password, we humans nearly universally create passwords that are easy to remember. Easy to remember nearly always means easy to guess. Most people will use names of people, pets or locations with which they are familiar. Or perhaps they will use some geometric pattern on the keyboard (like “asdf”). When they do think to include numbers within the password, they will often use two or four digits representing milestone years, birthdates or anniversaries, in some combination with the above.

Brute force attacks

Attacks against weak passwords are quick and relatively easy, with tools easily downloaded from the Internet. Brute force refers to “just try” as often as you can. One way to prevent brute force attacks is a lock-out policy (lock a user account after x failed attempts). Other approaches slow attackers down by enforcing x seconds wait between login attempts or using captcha to verify they are interacting with a human user. Brute force attacks often use a dictionary to try common passwords and common patterns (like letters that replace characters) first. Search the web for “password guessing software” for some examples.

Cross-site password theft

To make matters worse, most people will reuse passwords across multiple websites. Websites which offer free access to some desired object (a music file, an ebook, a photo sharing service, and so forth) will typically request that you create an account by providing a userid and password. The administrators of these websites then can (and often do) take this list of freely given user IDs and passwords and attempt to use these same credentials with banks or with other websites. Incredibly easy and surprisingly effective.

We speak of passwords being “stolen”, but unlike a physical device whose absence would be immediately obvious the next time it is needed, a password is still known to the proper owner. It is more as if the password has been “copied” – and detection of a copied password can be very difficult.

Passwords: We advise using authentication mechanisms that are a combination of what you know (passwords) and what you have (some hardware device whose loss would be immediately obvious).

Multi-factor authentication

This combination is called “multi-factor” authentication—your identity is proved based upon more than just one factor (a password).

There are several vendors of products that provide hardware-assisted password generators, and some are tightly integrated into WebSphere Application Server. We encourage you to research this for yourself, or contact your IBM client partner to discuss options.

We are not offering any product recommendations, but an example of such a hardware device is the RSA SecurID Token, which can be integrated into the WebSphere Application Server.

For more information, see:

RSA ClearTrust Web Access Management at:

http://www.rsa.com/products/cleartrust/whitepapers/CTIBM_WP_0403.pdf

3.3.2 Failure to change default passwords

How many times have you had to set up a consumer home wifi router only to discover that it is “secured” by the username + password combination admin + admin? This is actually an improvement. In previous years, the security was turned off by default. At least with this scheme, the device is immediately (albeit loosely) secured. The obvious first step after powering on the device would be to change the admin password. Unfortunately, many home users fail to take this step.

Well, the same is true for Business Process Manager V7.5. These products shipped with eight default accounts (tw_admin, tw_author, and so forth), and each of them had as their passwords their usernames (tw_admin + tw_admin, and so forth).

We have seen in the field that these default passwords are not always changed. This means that anyone who has any familiarity with Business Process Manager would have a reasonable chance of gaining administrative access, based simply upon the knowledge of the factory-default password.

Business Process Manager V7.5.1 improved the situation, in that the main Business Process Manager administrative account name and password are specified at installation time. So even if someone had intimate knowledge of Business Process Manager, they still would have to guess the administrative account name as well as the password. Hopefully, you have taken heed of the previous section’s advice against weak passwords, and so this would be a formidable task.

However, even within Business Process Manager V7.5.1, there is room for password hardening. Business Process Manager still ships with a number of default accounts (the same tw_admin, tw_author, and so forth), and the installation process does not present you with the opportunity to change each account name or password. Instead, the software installer takes the password that was chosen for the Business Process Manager administrative account and reuses it for these other default accounts.

These accounts, which do not have a user interface option for changing the account names, are as follows:

- ▶ tw_admin
- ▶ tw_author
- ▶ tw_portal_admin
- ▶ tw_runtime_server
- ▶ tw_user

- ▶ tw_webservice
- ▶ bpmAuthor

Default accounts: We advise that you remove these default accounts, and instead map actual users in your organization into the groups and roles which these accounts fill by default.

It is far easier to trace a security breach from a specific, named user account, than from a generic account whose password may be known by more than one person.

Important: The sole exception to the above suggestion is that you *do not* remove tw_admin. This account is still being referenced within the Business Process Manager source code directly, and its removal will cause product failures. We suspect that this limitation will be removed in a future version of the product, but as of Business Process Manager V8.0, this is still our advice.

As shown in 2.2.5, “Overuse of default Business Process Manager accounts” on page 33 the Business Process Manager V7.5.1 installation process includes default accounts that are prepopulated with one Business Process Manager administrative account that you specified during profile creation (in our example, it was bpmadmin).

Multiple account names: We advise that you choose different account names for each of these roles, with different strong passwords for each.

3.3.3 Faith in firewalls

A failure to use SSL over all BPM-related communications channels is so common that we have decided to highlight this security hole in each of our chapters. We hope that by demonstrating how security can be compromised from the varying perspectives of each chapter, that our readers will gain an appreciation for how important it is that each communication channel be encrypted.

A simple network protocol analyzer (freely downloadable from the Internet) can be used to eavesdrop on network communications. For the examples in this book, we are using Wireshark (formerly known as Ethereal), which has native versions for both Windows and Mac OS X at:

<http://www.wireshark.org/download.html>

Certainly, many modern network switches incorporate features that analyze the network traffic and make an attack more difficult, forwarding to a computer only those TCP/IP packets that are specifically addressed to that destination computer.

But there are ways around this selective forwarding. Some switches have a monitoring port which allows for a full inspection, and if a hacker is capable of gaining access to that port, then all network traffic would be visible. It may also be possible for a hacker to place a laptop in the middle of a normal network link and inspect (or record) the traffic that passes through it.

The case is even more dramatic for wireless networks. At least with a wired network, a smart switch can send signals down only that wire which connects to the computer for which the traffic is intended. This is not true for wireless signals—the radio waves fall upon everything within their transmit range.

The bottom line, as mentioned in the previous chapter, is to eliminate the “low hanging fruit”. Encrypt the communications channels, and eliminate the possibility of these types of attacks before the opportunity arises.

3.3.4 Insecure LDAP connections

Most people think of LDAP as a server, but in fact LDAP stands for Lightweight Directory Access Protocol. Yes, there are LDAP servers, but it is helpful to keep in mind that it is fundamentally a protocol—which means, in effect, that this is a conversation between two servers.

As another specific example of how an overabundant faith in firewalls can be a security hole, we examine the LDAP connection. Figure 3-13 shows the actual network communications as captured by the freeware network protocol analyzer, Wireshark, during the normal boot-up process of the BPM Process Center.

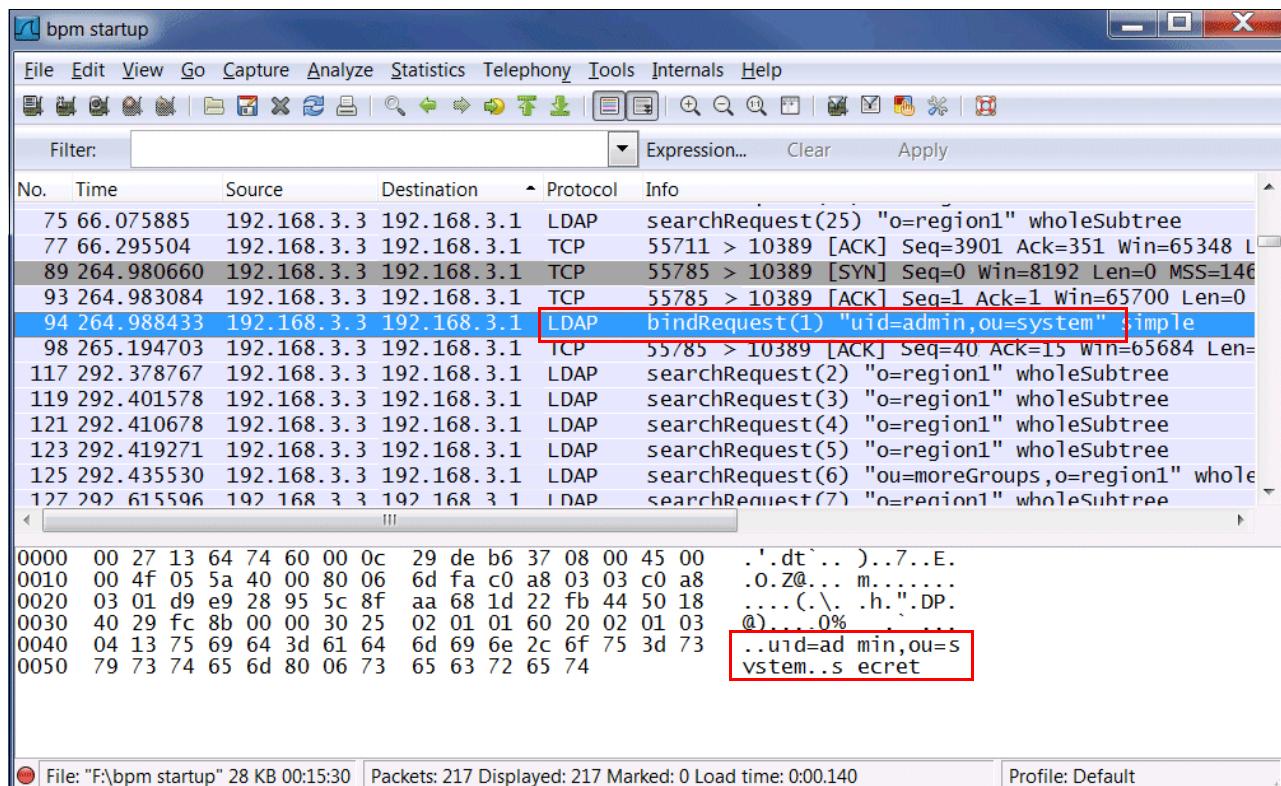


Figure 3-13 Network communications captured by Wireshark¹

If you have never used a network packet analyzer before, all you need to know for our purposes is that each line is a summary of the TCP/IP traffic—you can watch this list in real time—and the lower portion of the panel here is showing the contents of the selected TCP/IP packet.

This highlighted summary clearly shows that this packet is an LDAP “bindRequest” on the account uid=admin, ou=system; in other words, a request to bind the Business Process Manager servers to the LDAP database in order to make a query. If you know the account name and password used to bind, you can browse the LDAP for any information it contains. You can see, in the packet’s content area, that the password for this account is “secret”.

¹ Wireshark (www.wireshark.org)

Yes, it really is that easy.

The bind account name and password are exchanged at each step of the Business Process Manager to LDAP conversation. The WebSphere Deployment Manager and Node Agents, the Business Process Manager application servers (including /ProcessAdmin, /ProcessCenter and /portal), plus the Process Designer, all communicate with the LDAP server and issue this same bindRequest.

Unless you secure your LDAP server using encryption (SSL), you are leaving your corporate LDAP server open to browsing each and every time a Business Process Manager user logs into their /portal Inbox.

Let us take a look at how we got here. After Business Process Manager is installed, during the process of configuring your LDAP server for use with WebSphere's user registry, you see the panel in the Integrated Solutions Console shown in Figure 3-14.

Cell=bpmDevCell, Profile=devDMgr

Global security

Global security > Federated repositories > Apache Dir Server

Specifies the configuration for secure access to a Lightweight Directory Access Protocol (LDAP) repository with optional failover servers.

General Properties

* Repository identifier
Apache Dir Server

LDAP server

* Directory type
Custom

* Primary host name
ibmhost Port
10389

Failover server used when primary is not available:
Delete
Select Failover Host Name Port
None
Add

Support referrals to other LDAP servers
ignore

Security

Bind distinguished name
uid=admin,ou=system

Bind password

Login properties
uid

LDAP attribute for Kerberos principal name

Certificate mapping
EXACT_DN

Certificate filter

Require SSL communications

Centrally managed
[Manage endpoint security configurations](#)

Use specific SSL alias
CellDefaultSSLSettings [SSL configurations](#)

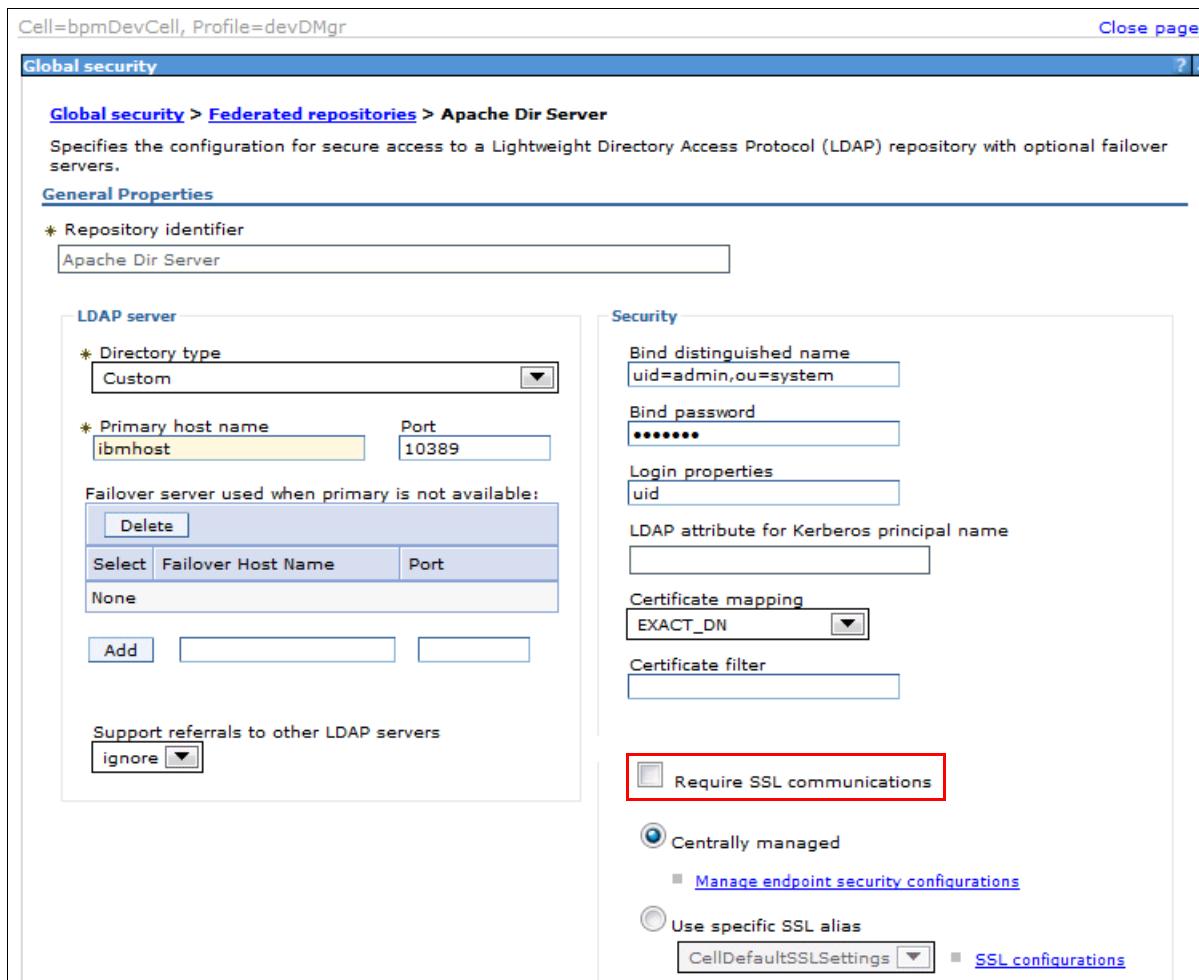


Figure 3-14 Require SSL communications

Notice that there is a checkbox to "Require SSL communications", but that it is not checked by default. In addition to enforcing SSL, you have an option to make this centrally managed or to use a specific SSL alias.

We advise that you:

- ▶ Enforce encryption using SSL over the communications channel between the Business Process Manager servers and your LDAP servers,
- ▶ Be sure to disable non-SSL traffic.
- ▶ Create a specific SSL truststore and alias for the LDAP.

After your LDAP is configured properly, you will see something like Figure 3-15.

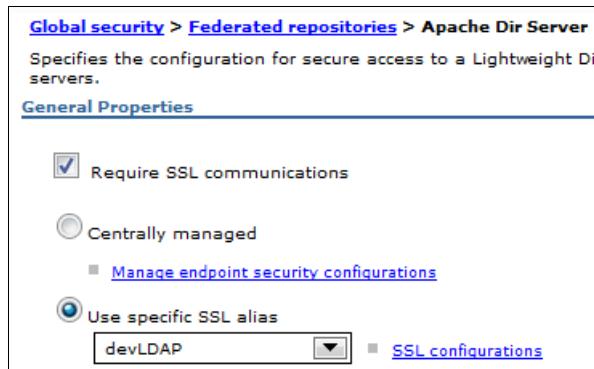


Figure 3-15 LDAP configured

The specific SSL alias has been defined as shown in Figure 3-16.

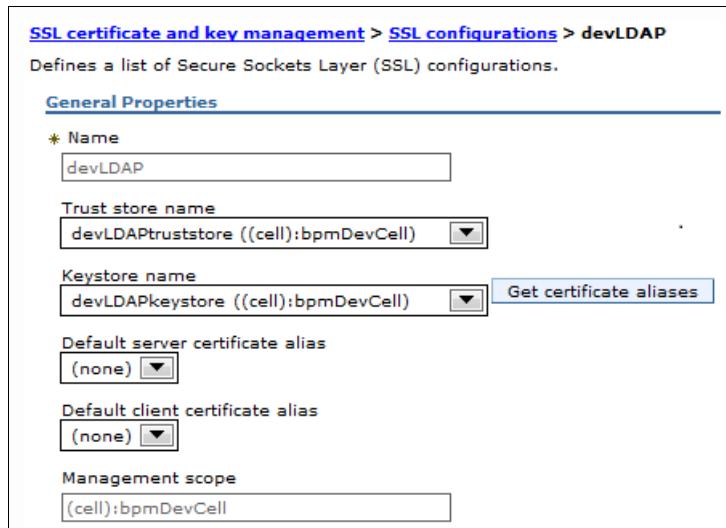


Figure 3-16 SSL alias

For more information, see *WebSphere Application Server V7.0 Security Guide*, SG24-7660.

3.3.5 Insecure SSO solutions

Single Sign-On (SSO) is the ability to share credentials across systems. There are many SSO solutions that can be purchased and integrated into Business Process Manager:

- ▶ Tivoli Access Manager
- ▶ CA SiteMinder
- ▶ Microsoft Windows Integrated Authentication
- ▶ UNIX and Linux Kerberos-based authentication protocols

- ▶ WebSphere custom Trust Association Interceptors

Many SSO technologies rely upon cookies or HTTP headers to carry the user's credentials with each HTTP request. Often, these credentials are encrypted.

Unfortunately, that these credentials are encrypted does not matter—an encrypted header can still be sniffed, copied, and injected into a hacker's browser HTTP requests. The fact that the human hacker cannot read the contents of the encrypted header in no way diminishes the opportunity for attack; he/she will just paste into his/her browser the contents of this encrypted header, thereby impersonating the original SSO credentials.

What must occur for any SSO solution to be considered secure is that the destination server must take the extra steps to ensure that the SSO headers originated from a known good, trusted server. Simple strategies like inspecting the IP address of the HTTP request are not sufficient—IP addresses (as in this case) can be spoofed.

The correct, proper and secure implementation of an SSO solution will employ a mechanism for ensuring that the origination of the SSO credentials is valid, thereby eliminating the type of injection attack described above. This is especially true if you are writing a custom Trust Association Interceptor (TAI) in order to integrate an in-house developed custom authentication system into the WebSphere Application Server security mechanisms.

The Trust Association Interceptor API is shown in Example 3-1.

Example 3-1 Trust Association Interceptor API

```
public class Simple_TAI implements TrustAssociationInterceptor {
    public String getType() {}
    public String getVersion() {}
    public int initialize(Properties props)
        throws WebTrustAssociationFailedException {}
    public void cleanup() {}
    public boolean isTargetInterceptor(HttpServletRequest request)
        throws WebTrustAssociationException {}
    public TAIResult negotiateValidateandEstablishTrust(
        HttpServletRequest request,
        HttpServletResponse response)
        throws WebTrustAssociationFailedException {}
}
```

As you can see, this is deceptively simple. Six methods, two of which are simple get() functions which return (typically) hard-coded strings. Two more are just initialize() and cleanup() functions. This leaves us with only two methods to do the heavy lifting.

It is very common for us to see software developers—under the gun of time constraints and other project pressures—after having gone through whatever hoops were necessary to gain access to the custom authentication system, getting firewall ports opened for their use, having Access Control Lists (ACLs) updated, and so forth (not to mention the software development and debugging process of making the TAI actually work), to celebrate that final change which ultimately allowed their SSO solution to function by simply moving on to the next over-due project task.

It is not enough that your SSO solution works, it needs to work securely.

This is another common security hole that we see: an in-house developed TAI which fails to take the additional step of ensuring that the origination source of the SSO header comes from a trusted source.

Techniques for establishing trust with the origination system include (but are not limited to) the following:

- ▶ Validating the SSL certificate's serial number
- ▶ Validating the SHA1 digest's fingerprint

SSO solution: We advise that you bring in an outside security professional to review your SSO solution to ensure that it meets all leading security practices before you put any such code into use.



Authorization: Access to what

Authorization is the process of ensuring that a user (or other computer system) has permission to perform a given act.

In general, authorization can be enforced in a number of ways, including:

- ▶ Access Control Lists (ACL)
- ▶ LDAP groups
- ▶ Role Based Access Control (RBAC) such as LDAP groups in J2E authorization
- ▶ Attribute Based Access Control (ABAC)

Authorization can be defined for any given application along a continuum of granularity. For example, the following list of theoretical authorizations goes from coarse-grained to fine-grained:

- ▶ Everyone is authorized to view this web page.
- ▶ A user must be logged in to view this web page.
- ▶ A logged in user must also exist within an ACL in order to update this web page.
- ▶ A logged in user must be a member of a specific LDAP group in order to have certain sections of this web page made visible.
- ▶ A logged in user must be known to the underlying application which generates this web page, and before any data is displayed or before any button is drawn on the web page, the application will programmatically check `isUserInRole()` to determine if this user has been granted the role associated with this data and/or button.
- ▶ A logged in Business Process Manager user can only execute this task if they a) are a member of a specific security group, b) were not the last user to touch this task, c) belong to a level of management who has authority to perform this task, and d) the task has not sat unattended for more than four hours.

As you can see, Business Process Manager defines a very fine-grained authorization model, and does so in product-specific terms (for example, you are not authorized to run this task if you were also the last user to do so).

In order to understand its flexibility and power, we need first to consider what is meant by groups, roles, and how the user repository interacts with the Business Process Manager

servers. We see a great deal of confusion over the terms *groups* and *roles*. LDAP groups, VMM Security Groups and Participant Groups—these terms are often used (incorrectly) interchangeably. In order to simplify our understanding of these concepts, we will use a very strict definition for each, and we will introduce them as needed.

In the previous two chapters, we have laid the foundation for ensuring that our Business Process Manager servers are secured and that any user who attempts to log into the Business Process Manager application is verified to be genuinely who they say they are. We now turn our attention to the most feature-rich aspect of the Business Process Manager security stack: authorization. What specific actions is each user allowed?

4.1 Groups versus roles

It will be helpful to establish what we mean when we say groups or roles.

- ▶ Groups are a collection of users, a concrete instantiation.
- ▶ Roles are placeholders for users, to be populated at runtime.

Let us take, for example, a software product which is intended for resell to the general public. It may define different functions for sales managers, support staff and sales people. The software developers cannot know beforehand who is going to purchase this product, and so there is no way for them to know beforehand who these specific people are going to be. Instead, this developer must provide for the roles of sales manager, support staff and sales people, and provide for a way of populating these roles with actual user names after the product is purchased and installed at the various customer locations.

In the case of Business Process Manager, there are various methods of creating and managing groups: actual lists of users, and two methods of populating the roles with these groups.

We shall begin with a look at the mechanisms for creating and managing groups.

4.2 Grouping mechanisms

Business Process Manager authorization begins with an understanding of what users and groups have been defined in the WebSphere user registry. WebSphere Application Server allows a variety of user and group “lists”, collectively called the user registry. By far the most common entity within the user registry is one or more LDAP servers.

Although Business Process Manager can utilize a stand-alone LDAP registry, this book discusses Business Process Manager authorization within the context of a federated user registry, comprised of two LDAP repositories. Much of this authorization discussion would still apply equally to a flat-file repository, a stand-alone LDAP, or a custom user registry, but there are some very significant features that would not apply. We point these out as they occur in the discussion.

Note: We recommend using LDAP repositories federated together within the WebSphere user registry.

4.2.1 LDAP groups

In the preceding chapter, we discussed the “branches” of LDAP trees. Each LDAP vendor, and to a certain extent, each LDAP administrator, is given the flexibility to choose the nature of these branches. These can be considered *organizational units* or they can be groups with *common names*. The specifics of how groups are defined in the LDAP is therefore beyond the scope of this book, but regardless of how they are defined, the BPM product can access them in the same way.

For more information, see *Understanding LDAP – Design and Implementation*, SG24-4986.

Consider the LDAP instance in Figure 4-1.

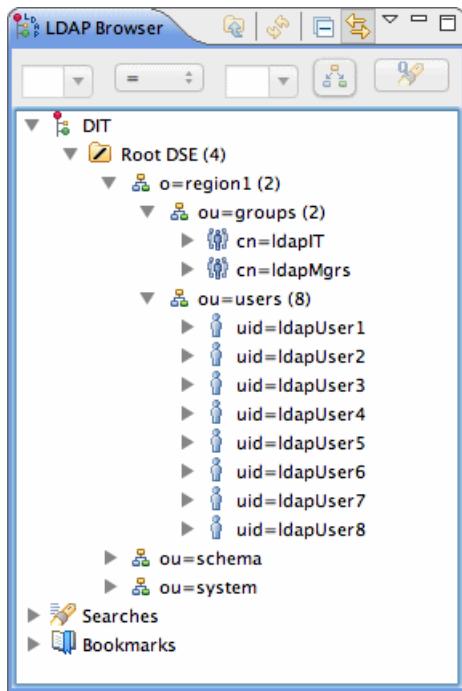


Figure 4-1 LDAP instance

In this structure, we have two groups defined: ldapIT and ldapMgrs. Further inspection of each reveals that both groups have two members each (Figure 4-2 on page 68 and Figure 4-3 on page 69).

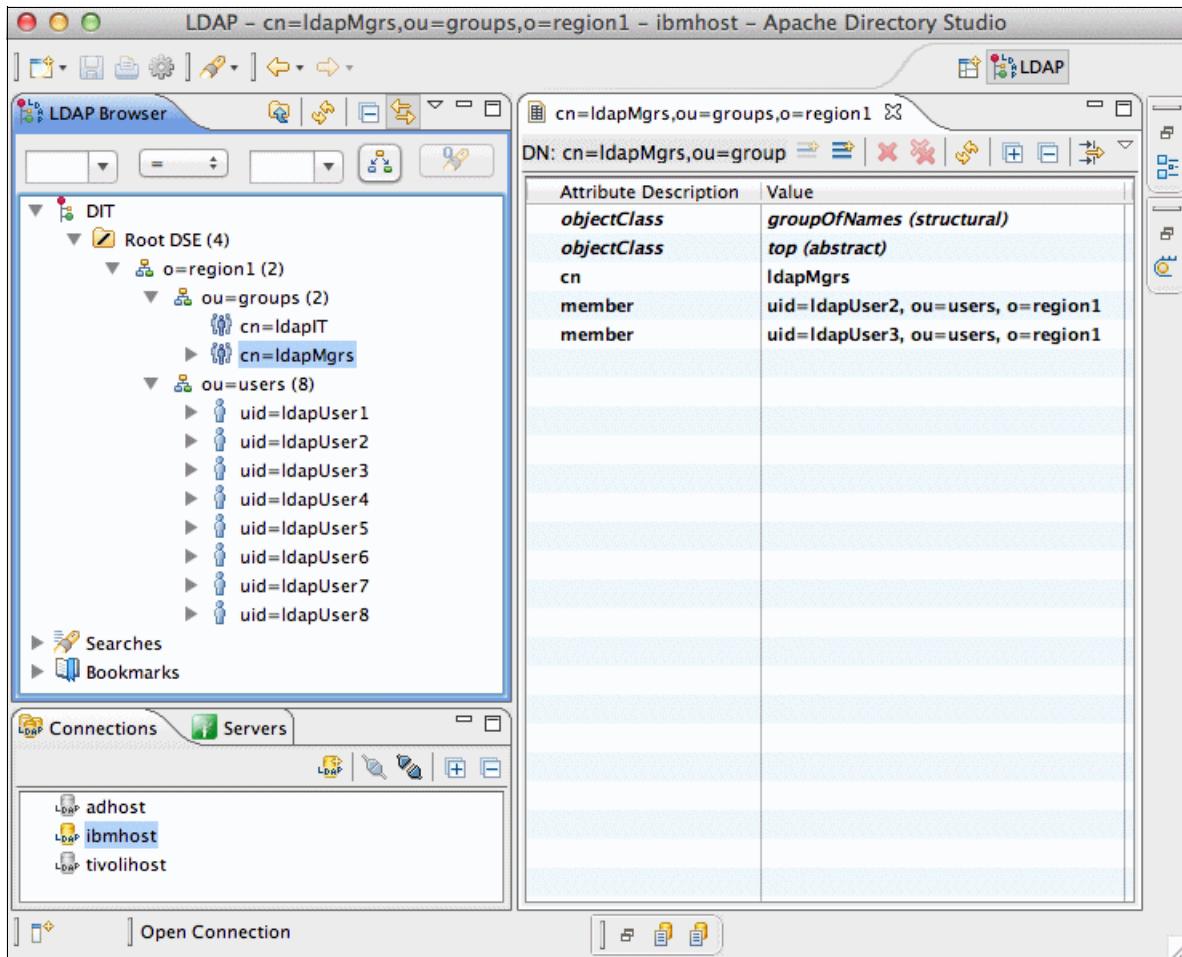


Figure 4-2 *ldapMgrs*

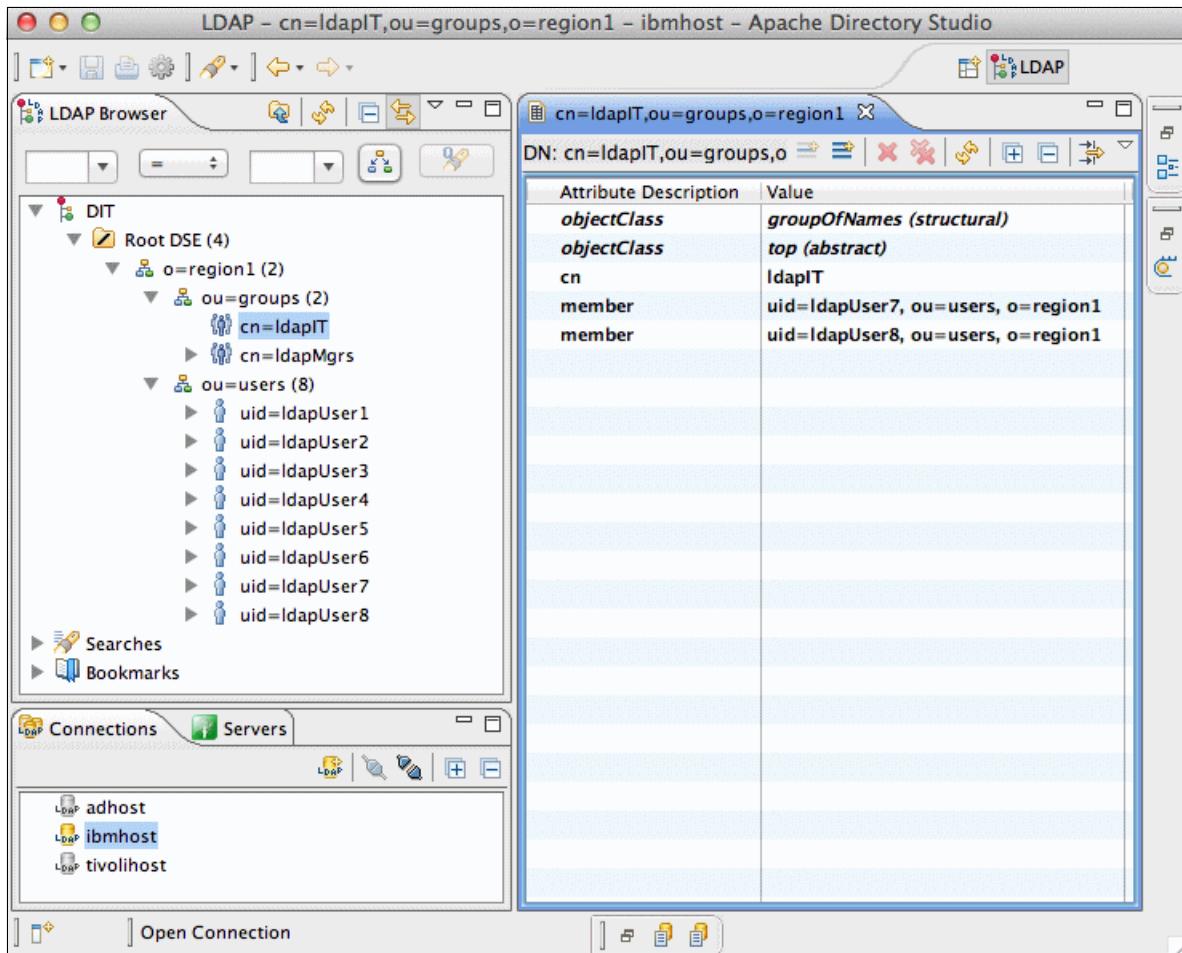


Figure 4-3 *ldapIT*

So to summarize, eight users are defined in region1 LDAP, as well as two groups each containing two members.

This is obviously a very simple example, but it will serve our purposes to demonstrate Business Process Manager's power and flexibility.

In a typical organization, there may be tens of thousands, even hundreds of thousands of users, and thousands of groups. These numbers can grow quite large, and so it is common to see an LDAP vendor restrict the number of users or groups returned by generic queries. For example, Microsoft's Active Directory will return only the first 4,500 users and/or groups, truncating the rest from the results.

Because LDAP products are optimized for read-only and static data, it is very common that the structure and content of the LDAP repository be under the strict control of an LDAP administration team. Furthermore, these groups may have been defined for legacy applications, and it is not uncommon for these legacy applications to no longer be in use, and yet their LDAP groups remain. This can run at odds with the core value proposition of the Business Process Manager product.

One of the most significant benefits of Business Process Manager is its ability to model business processes in order to gain some perspective and, hopefully, to leverage this to effect business process improvement. The entire approach to Business Process Manager process

documentation, process application development, and process improvement is based upon agile principles.

It is quite common for a team of process designers to decide that a process could be improved if the roles were reexamined, possibly creating new roles and combining others. To rely upon the LDAP administration team to make these changes to the corporate LDAP is almost always untenable—untenable from both perspectives: the process designers want immediate turnaround, and the LDAP administrators need to ensure that changes will not break other enterprise applications that rely upon the LDAP structure.

Note: We recommend that the LDAP structure be used when convenient to do so, but look to Business Process Manager under all circumstances where it is not so convenient.

4.2.2 VMM security groups

We have mentioned that the WebSphere Virtual Member Manager (VMM) adds functionality above and beyond what can be achieved using a stand-alone LDAP or a custom user registry. In this section, we investigate one of the VMM's most important features.

For the rest of this book, we use the following LDAP repositories:

- ▶ o=region1 is an Apache Directory Server instance (Figure 4-4).
- ▶ o=region2 is an Active Directory instance (Figure 4-5 on page 71).

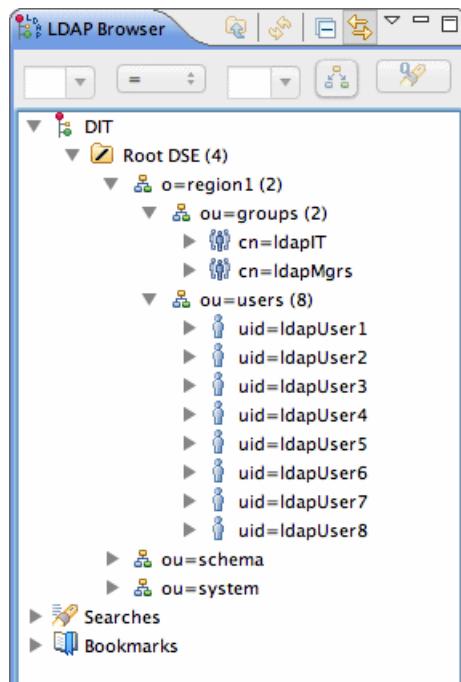


Figure 4-4 Apache Directory Server instance

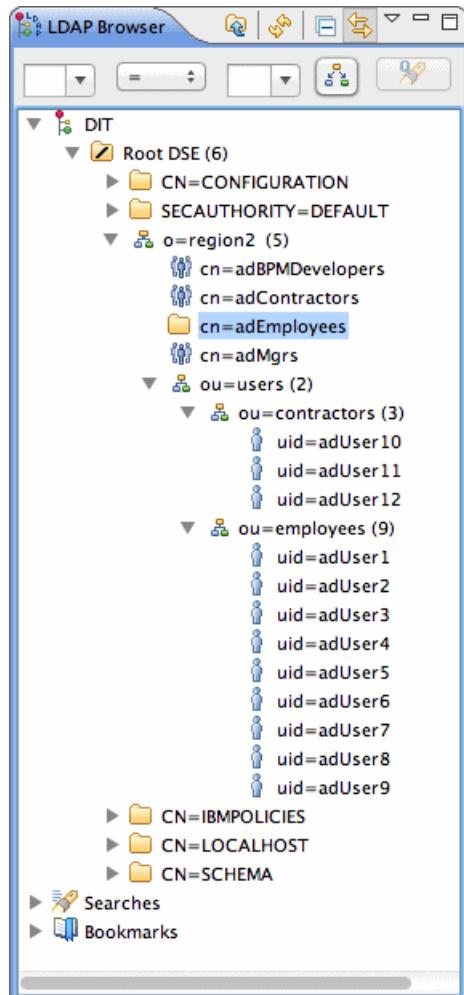


Figure 4-5 Active Directory instance

Even in this simple example, you can see that both LDAP repositories define very different structures. Yes, they are both trees, but the Active Directory repository has nested organizational units to define two categories of users (ou=employees and ou=contractors). This is a completely arbitrary distinction, one which is not in any way related to the fact that it is an Active Directory repository. Notwithstanding, it is very common for us to see multiple LDAP repositories configured with different structures.

The reasons for this are many, but probably the easiest one to grasp is that of a corporate acquisition. For the purpose of maintaining compatibility with existing (legacy) applications, the acquiring company may be resistant (or even incapable) of altering the LDAP structure. There is obviously a period of time following an acquisition when the two company's human resources and IT groups integrate—a perfect time for business process re-engineering.

It is during times and situations such as this that the WebSphere Virtual Member Manager (VMM) really shines. Figure 4-6 on page 72 shows what the Integrated Solutions Console looks like after these two LDAPs have been federated together.

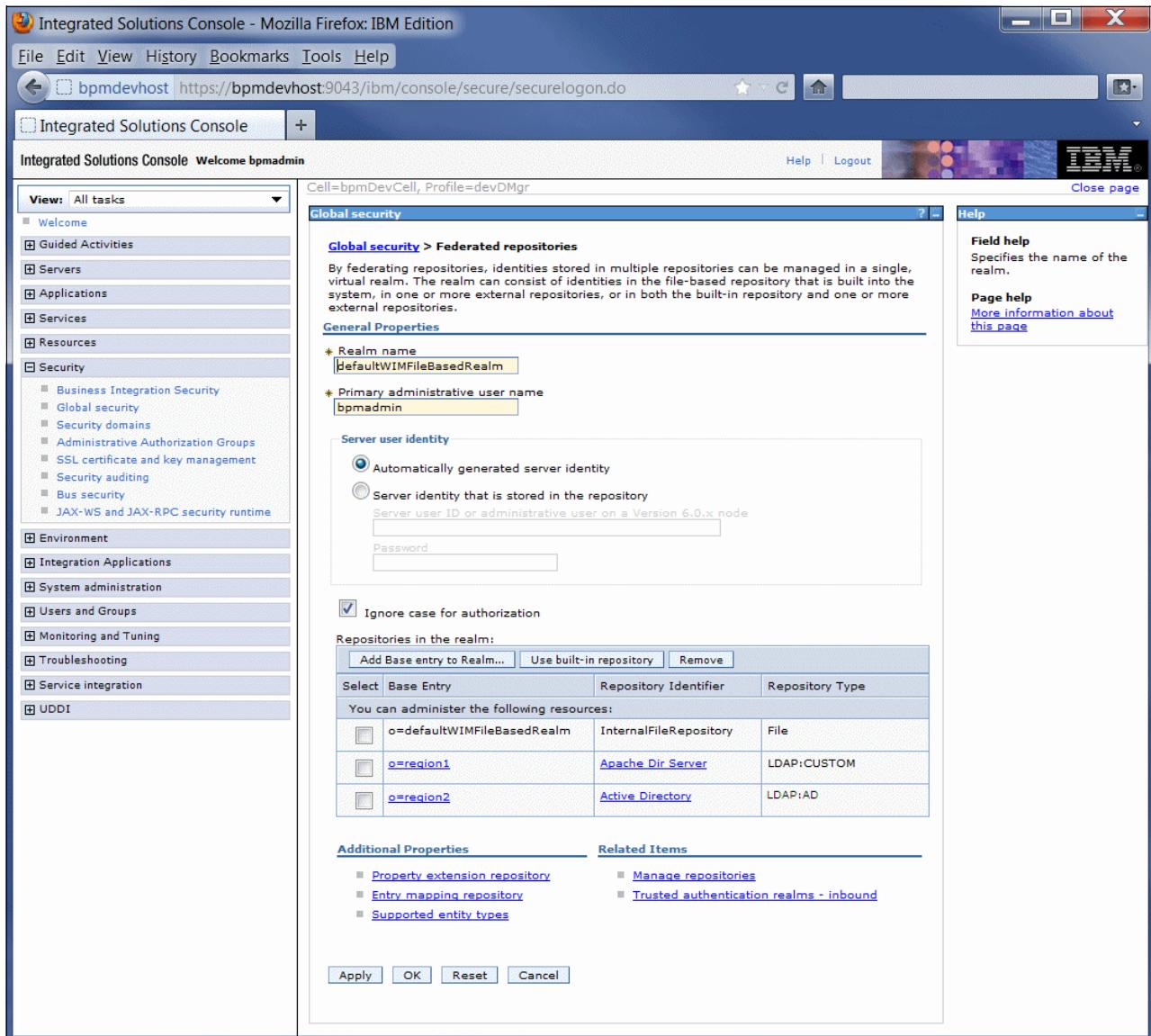


Figure 4-6 Two LDAP servers federated

In the Integrated Solutions Console, navigate to **Users and Groups → Manage Users** and click **Search**. You get a *unified view of all users and groups* in both LDAP repositories. In Figure 4-7 on page 73 and Figure 4-8 on page 74 you can see that we have Apache LDAP users from region1 federated together in the same registry as active directory users from region2.

Manage Users

Manage Users ? □

Search for Users

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

Search

37 users matched the search criteria.

Select	User ID	First name	Last name	E-mail	Unique Name
<input type="checkbox"/>	bpmAuthor	bpmAuthor	bpmAuthor		uid=bpmAuthor,o=defaultWIMFileBasedRealm
<input type="checkbox"/>	bpmadmin	bpmadmin	bpmadmin		uid=bpmadmin,o=defaultWIMFileBasedRealm
<input type="checkbox"/>	bpcMonitor	bpc	Monitor	bpcmonitor	uid=bpcMonitor,o=defaultWIMFileBasedRealm
<input type="checkbox"/>	pbcadmin	bpc	Admin	pbcadmin	uid=pbcadmin,o=defaultWIMFileBasedRealm
<input type="checkbox"/>	tw_admin	tw_admin	tw_admin		uid=tw_admin,o=defaultWIMFileBasedRealm
<input type="checkbox"/>	tw_author	tw_author	tw_autho		uid=tw_admin,o=defaultWIMFileBasedRealm
<input type="checkbox"/>	adUser1	activeDir	Users1		uid=adUser1,ou=employees,ou=users,o=region2
<input type="checkbox"/>	adUser2	activeDir	Users2		uid=adUser2,ou=employees,ou=users,o=region2
<input type="checkbox"/>	adUser3	activeDir	Users3		uid=adUser3,ou=employees,ou=users,o=region2
<input type="checkbox"/>	adUser4	activeDir	Users4		uid=adUser4,ou=employees,ou=users,o=region2
<input type="checkbox"/>	ldapUser1	ldap	User1		uid=ldapUser1,ou=users,o=region1
<input type="checkbox"/>	ldapUser2	ldap	User2		uid=ldapUser2,ou=users,o=region1
<input type="checkbox"/>	ldapUser3	ldap	User3		uid=ldapUser3,ou=users,o=region1
<input type="checkbox"/>	ldapUser4	ldap	User4		uid=ldapUser4,ou=users,o=region1

◀ Page 2 of 3 ▶ 2 Go Total: 37

Figure 4-7 Unified view of all users and groups - users

The screenshot shows a 'Manage Groups' interface with a search bar at the top. The search criteria are set to 'Group name' with a value of '*' and a maximum result count of '100'. A 'Search' button is present. Below the search bar, a message states '6 groups matched the search criteria.' A table follows, with columns: Select, Group name, Description, and Unique Name. The table lists six groups:

Select	Group name	Description	Unique Name
<input type="checkbox"/>	adBPMDevelopers		cn=adBPMDevelopers,o=region2
<input type="checkbox"/>	adContractors		cn=adContractors,o=region2
<input type="checkbox"/>	adEmployees		cn=adEmployees,o=region2
<input type="checkbox"/>	adMgrs		cn=adMgrs,o=region2
<input type="checkbox"/>	ldapIT		cn=ldapIT,ou=groups,o=region1
<input type="checkbox"/>	ldapMgrs		cn=ldapMgrs,ou=groups,o=region1

At the bottom, it says 'Page 1 of 1' and 'Total: 6'.

Figure 4-8 Unified view of all users and groups - groups

Furthermore, navigating into the ldapIT or ldapMgrs groups will show you the members of each that are defined in the LDAP repository (Figure 4-9 and Figure 4-10).

The screenshot shows the 'Manage Groups' interface for the 'ldapMgrs' group. The 'Group name' field is set to 'ldapMgrs'. It displays two members:

Select	ID	Type	Unique Name
<input type="checkbox"/>	ldapUser2		uid=ldapUser2, ou=users, o=region1
<input type="checkbox"/>	ldapUser3		uid=ldapUser3, ou=users, o=region1

At the bottom, it says 'Page 1 of 1' and 'Total: 2'.

Figure 4-9 Members in group ldapMgrs

The screenshot shows the 'Manage Groups' interface for the 'ldapIT' group. The 'Group name' field is set to 'ldapIT'. It displays two members:

Select	ID	Type	Unique Name
<input type="checkbox"/>	ldapUser7		uid=ldapUser7, ou=users, o=region1
<input type="checkbox"/>	ldapUser8		uid=ldapUser8, ou=users, o=region1

At the bottom, it says 'Page 1 of 1' and 'Total: 2'.

Figure 4-10 Members in group ldapIT

Notice that in these panel shots, there are buttons in each named Create, Delete, Add Users, Add Groups and Remove. The ability for these buttons to function will be tied to the read/write privileges of the account you used as the “Bind distinguished name” when configuring the LDAP repository in the Integrated Solutions Console. The LDAP account you specified obviously needs read-access to the LDAP, but it is very uncommon for the LDAP administrator team to give out LDAP accounts that have read/write privileges. In the more common case where the account is read-only, these buttons will have no effect on your LDAP, and the Create button results in a new account being defined within the defaultWIMFilebasedRealm.

The following points are very important to understand:

- ▶ The WebSphere VMM creates a unified view of all users and groups that have been federated into the user registry—quite literally, the union of all users and groups.
- ▶ This is a one-to-one relationship between the VMM security groups and the LDAP security groups.
- ▶ Each user ID can exist in one, and only one, of the repositories within the federation.

Consider the possibility that you may need to define a functional group (for inclusion in one of your Business Process Manager business process applications) which spans these LDAP groups. For example, you may want all adDevelopers from region2 and all ldapIT members from region1 included into one group.

The LDAP protocol does not support the notion of groups spanning repositories, and since the VMM is a one-to-one relationship, neither does it.

Notwithstanding, it is easy to understand how such a group could be of use, especially in the case of corporate acquisitions. Both the acquiring company and the acquired company are certain to have some functional group overlap, and it is easy to see how the new unified corporation would want members from each of these disparate repositories.

4.2.3 Process Admin Console and private groups

There are three main BPM tools for defining and administering your BPM installations:

- ▶ Integrated Solutions Console (/ibm/console)
- ▶ Process Admin Console (/ProcessAdmin)
- ▶ Process Center Console (/ProcessCenter)

We have already seen the Integrated Solutions Console used to a great extent in Chapter 2, “Installation” on page 21 and Chapter 3, “Authentication: Who has access” on page 41 to ensure SSL between Process Center and Process Server, to inspect and change the Business Process Manager default passwords, and to define the WebSphere user registry for use with Business Process Manager authentication.

It is time now to turn to the /ProcessAdmin console. /ProcessAdmin is the main administrative application that is used to identify which users and groups should have access to what processes with each of your Business Process Manager environments.

Business Process Manager ships with a few default groups, as shown in the Process Admin Console (Figure 4-11 on page 76).

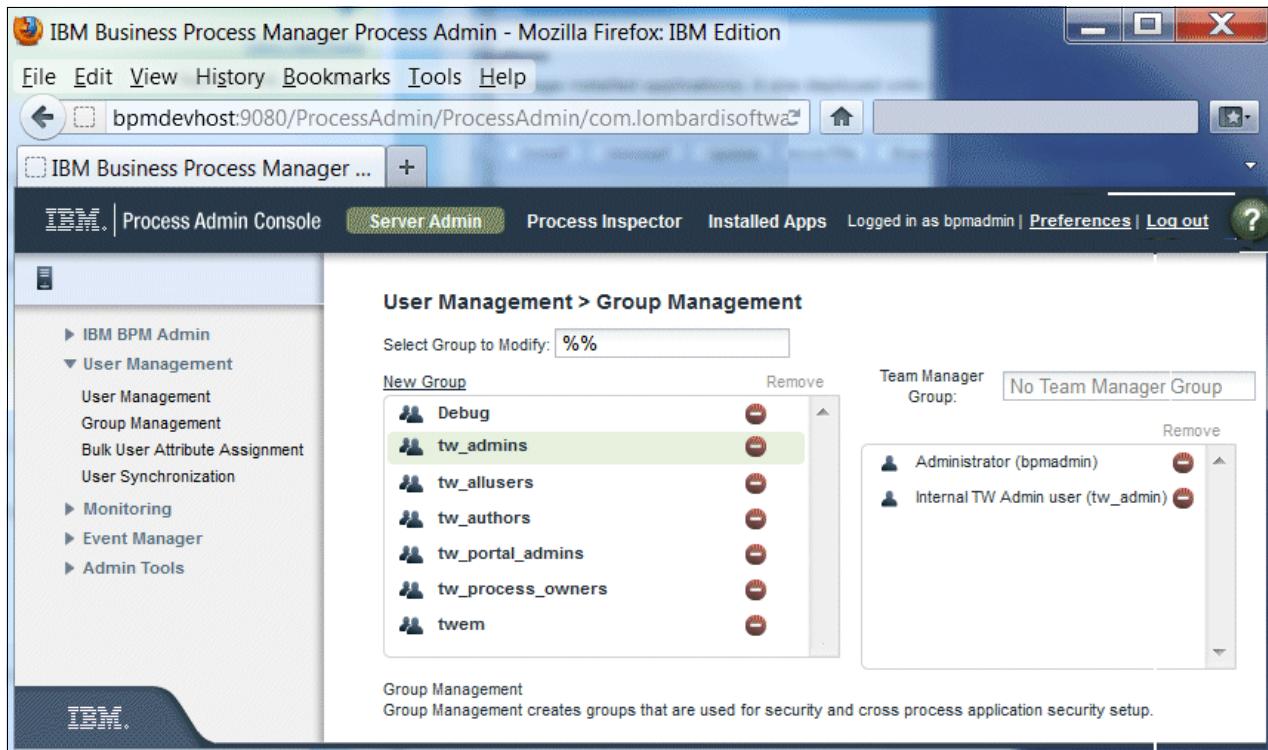


Figure 4-11 Default groups

In Figure 4-11, you can see that the Business Process Manager administrative account (in our example, `bpmadmin`) has been automatically added to the `tw_admins` group.

After the LDAP servers have been federated into the WebSphere VMM, each subsequent restart of the Business Process Manager servers will result in Business Process Manager querying the VMM security groups and comparing these against the groups defined in the Business Process Manager database. If any VMM groups are not present, then they are copied (by reference) from the VMM and inserted into the Business Process Manager database.

These groups will then be visible via the `/ProcessAdmin` (Figure 4-12 on page 77).

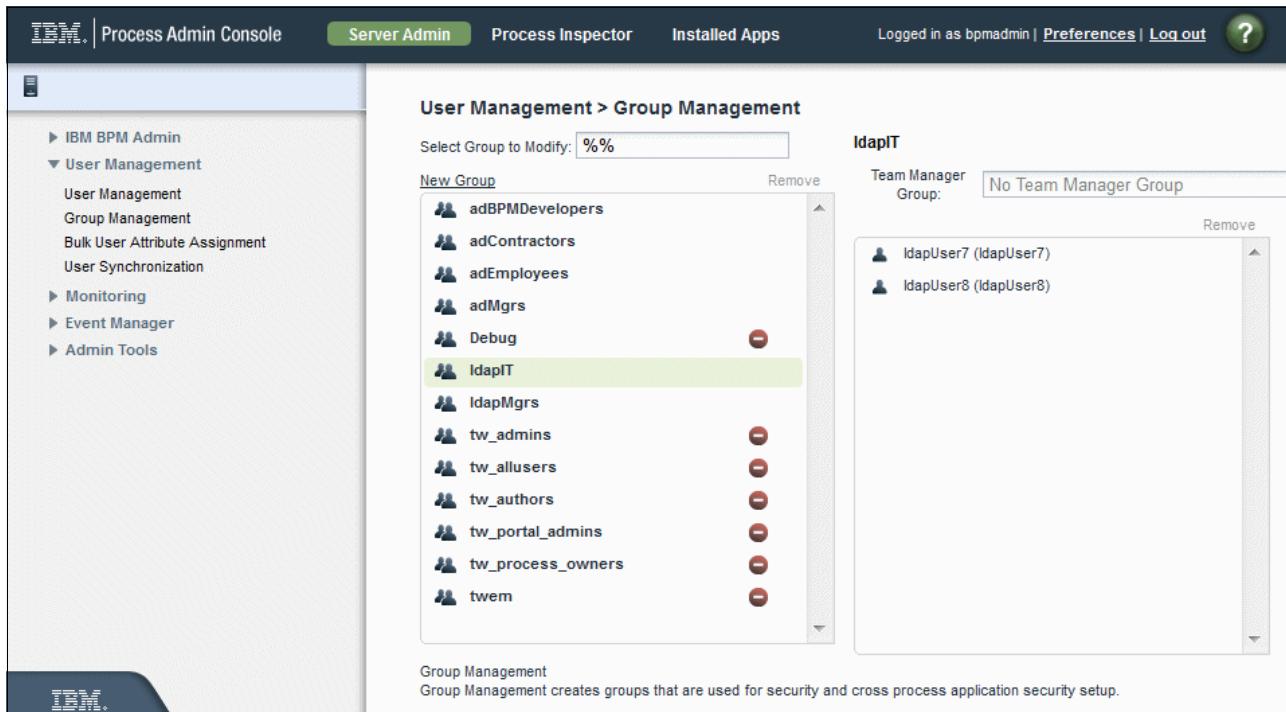


Figure 4-12 Groups in the Process Admin Console

Notice how none of the six groups that are now a part of the Business Process Manager database (adBPMDevelopers, adContractors, adEmployees, adMgrs, IdapIT, and IdapMgrs) have the red delete icon next to them. The groups that are brought forward into Business Process Manager by way of their existence in the VMM are not editable. You cannot delete them, nor can you add or remove members from them. This is in keeping with the normal practice of your LDAP Bind distinguished name being given read-only access.

Here is another very important point that is worth considering with respect to these LDAP defined, VMM present groups: they may have nothing to do with your current business processes. As was mentioned earlier, they may have been an artifact of legacy applications that are no longer in use. They may have members that have moved on to other roles in the company, or they may include members who are not even employed by the company any longer.

Note: We recommend that you ensure that your groups are tightly coupled to the business process functions.

This means that these groups should be precisely defined such that only those users who should be authorized to perform these functions appear in these groups.

Using the /ProcessAdmin, new groups can be added to the Business Process Manager database. Although the product never “officially” gives these groups a name, for our purposes these will be called Business Process Manager private *groups*. We term them private because groups added here are not visible upstream to either the WebSphere VMM nor any repositories federated therein. Conversely, however, these Business Process Manager private groups are capable of adding users and groups from the VMM—making them immediately useful.

Clicking the **New Group** link at the top of the center column in Figure 4-12, you will see the dialog box shown in Figure 4-13 on page 78.



Figure 4-13 Create Group dialog

You simply give the new group a name, document its use in the Description field, and you are ready to start adding users from the VMM.

This sounds simple enough—and it is—but do not let its simplicity obscure the following two very important points:

- ▶ Private groups can span VMM repositories.
- ▶ Private groups support nested groups.

Private groups can span VMM repositories

We begin by adding a few members to our new devAuthors group (Figure 4-14).

The screenshot shows the IBM BPM Admin interface. The left sidebar has 'User Management' expanded, with 'Group Management' selected. The main area shows the 'User Management > Group Management' page. A modal dialog titled 'Add User and Groups' is open, showing a list of users and groups under 'Results:' with 'ldap' typed into the search bar. The 'Add Selected' button is at the bottom right of the modal.

Figure 4-14 Adding members to a group

And we now have the definition shown in Figure 4-15 on page 79.

User Management > Group Management

Select Group to Modify: %%

New Group Remove

- adBPMDevelopers
- adContractors
- adEmployees
- adMgrs
- Debug
- devAuthors**
- IdapIT
- IdapMgrs
- tw_admins
- tw_allusers
- tw_authors
- tw_portal_admins
- tw_process_owners

Team Manager Group: No Team Manager Group

Add Members Remove

- adUser12 (adUser12)
- adBPMDevelopers
- IdapIT

Group Management
Group Management creates groups that are used for security and cross process application security setup.

Figure 4-15 Group management

We have created a new group named “devAuthors” and have specified the following members:

- ▶ adUser12 (an individual from the Active Directory region2 repository)
- ▶ adBPMDevelopers (a group from the Active Directory region2 repository)
- ▶ IdapIT (a group from the Apache Directory Server region1 repository)

Private groups support nested groups

Now, we add this new group, devAuthors, to the BPM default group tw_authors (Figure 4-16 on page 80).

Figure 4-16 Adding devAuthors to tw_authors

We have the following result: tw_authors includes devAuthors, which in turn includes adBPMDevelopers and IdapIT. This is represented in Figure 4-17.

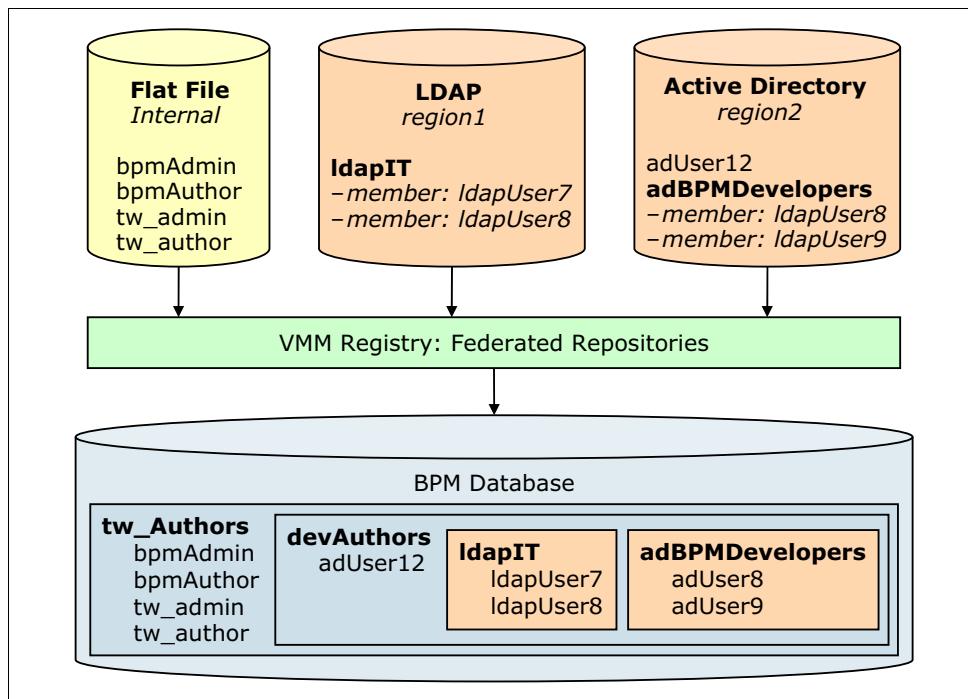


Figure 4-17 Updated groups

From a business process management point of view, you can very easily create groups which are defined entirely based upon the needs of the business process. Yes, you can draw from existing LDAP users and groups, but you are not constrained by the LDAP structures, nor are you constrained by the logistics of which repository a given user may belong to.

This is the very essence of authorization: ensuring that only users who should have permission to participate in a business process have that access.

So far, we have been looking at how to get your users into manageable groups that will be tied to process applications. But what about the process applications themselves? How does one specify the roles that these users should be playing? How does one tie the abstract to the concrete?

To answer these questions, we need to leave the /ProcessAdmin for a moment and turn our attention to the Process Designer.

4.2.4 Process Designer swimlanes and participant groups

The Process Designer is a Business Process Manager product component that allows business process application authors to model their processes at a high level, and yet it also allows for them to be drilled down into and described at a highly detailed level. Business processes are graphically defined using horizontal swimlanes to depict different classes of actors, and flowchart-like diagrams to depict the various process steps (Figure 4-18 on page 82).

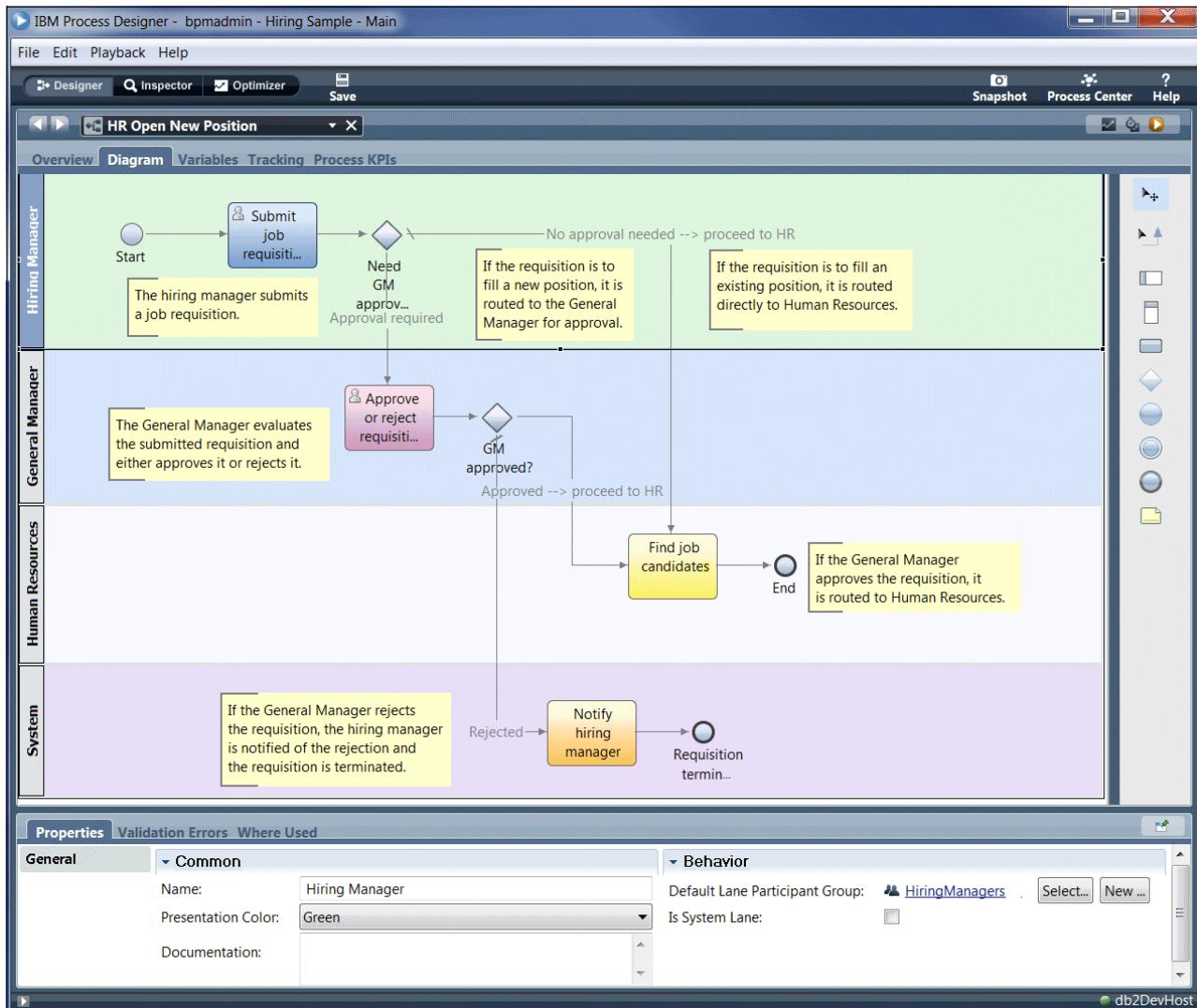


Figure 4-18 Process Designer

In the example in Figure 4-18, the process application is called Hiring Sample (it ships with the Business Process Manager product), and the business process definition is called HR Open New Position. There are three swimlanes: one each for Hiring Managers, General Managers, and Human Resources. These three swimlanes map directly to a Business Process Manager concept called a Participant Group.

During the act of process modeling, these swimlanes are created by simply dragging a swimlane icon from the toolbar (shown on the right of Figure 4-18) into the process application (the main part of the window). Give it a name, and you have just created a role. To create the participant group, the Business Process Manager author of this process definition simply clicked **New**, as shown in the bottom right of Figure 4-18, and was presented with the dialog box shown in Figure 4-19 on page 83.

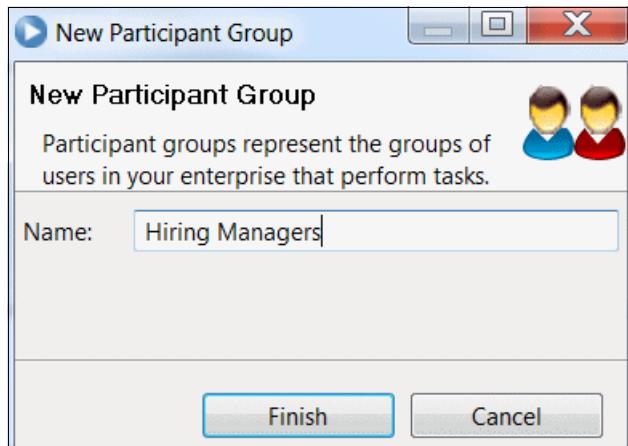


Figure 4-19 New Participant Group dialog

You just created a placeholder, to be filled in at runtime, of those users who should be given permissions to perform whatever actions appear in the Hiring Managers swimlane.

So, for example, Hiring Managers are permitted to “submit job requisition”, but nothing else. Only General Managers are allowed to “approve or reject requisitions”, and only in the case where the job requests are approved, are the Human Resources allowed to “find job candidates.”

Notice that this process is easy to understand, and close to the language that the business stakeholders are likely to be using. In fact, in keeping with the Business Process Manager agile methodology, it is almost certainly the exact language that the business stakeholders are using.

The process of defining authorization in Business Process Manager has abstracted the entire concepts of LDAP groups, Access Control Lists, and J2E programmatic authorization into swimlanes and flowchart icons.

From a business security point of view, this is useful. Only those users who actually have the job responsibilities depicted in a swimlane will be given authorization to execute the steps therein. And perhaps even more importantly, the business authors are given a tool that allows them to define—at a perfectly-grained level—the names of the roles which need to be populated with users and groups.

4.2.5 Mapping roles to groups

Notice too, however, that these swimlanes have nothing to do with the LDAP, VMM, or Private groups. The final step we need to undertake is to map these participant groups (roles, really) to groups that are visible in the /ProcessAdmin.

After the participant groups (roles) are defined in the Process Designer, we can return to the /ProcessAdmin and navigate to **Installed Apps → Role Bindings** (Figure 4-20 on page 84).

The screenshot shows the 'Role Bindings' tab for the 'Hiring Sample (HSS) - 7.5.1 (Active)' process. The interface includes a top navigation bar with links for Process Admin Console, Server Admin, Process Inspector, Installed Apps, Preferences, and Log out. Below the navigation is a toolbar with icons for Exposing, Role Bindings (which is selected), and Environment Vars. The main content area displays four participant groups:

- All Users**: Members: 1, Tasks: 0. Includes a link to 'Add Users and Groups' and a small icon showing a user named 'tw_allusers'.
- GeneralManagers**: Members: 0, Tasks: 0. Includes a link to 'Add Users and Groups'.
- HiringManagers**: Members: 0, Tasks: 0. Includes a link to 'Add Users and Groups'.
- HRManagers**: Members: 0, Tasks: 0. Includes a link to 'Add Users and Groups'.

A sidebar on the right contains links for application management: Deactivate Application, Migrate Inflight Data, Sync Settings, and Update Tracking Definitions. A detailed description of Role Bindings is also provided in the sidebar.

Figure 4-20 Role bindings

Here we can clearly see the participant groups (roles) that were defined in the Process Designer, and we can also see that at this point, there are no users mapped to these participant groups (roles).

The Add Users and Groups links next to each of the Participant Groups will take you to the panel shown in Figure 4-21.

The dialog box is titled 'Add people to GeneralManagers Role'. It features a search bar with the placeholder 'Search For Name:' and the input 'Idap'. To the right of the search bar is a note: 'Start typing to view matching results'. Below the search bar is a section labeled 'Results:' containing a list of users and groups:

- IdapIT
- IdapMgrs
- IdapUser1 (IdapUser1)
- IdapUser2 (IdapUser2)
- IdapUser3 (IdapUser3)
- IdapUser4 (IdapUser4)
- IdapUser5 (IdapUser5)

On the right side of the results list are several small square checkboxes. At the bottom of the dialog is a descriptive text block:

You can add users or groups available in the current environment to each participant group. (The users in the environment on which this snapshot was initially developed might be different from the users in your test or production environment.) To add users or groups, enter part of a user or group name in the Retrieve field, click the check box for each user and group that you want, and then click the Add button.

At the bottom right is a blue 'Add Selected' button.

Figure 4-21 Add users and groups

This dialog box provides nearly the exact same functionality we saw earlier when adding VMM or LDAP users and groups to Business Process Manager private groups. Business Process Manager allows you to select from the entire universe of VMM or LDAP users and groups, as well as the Business Process Manager private groups when mapping to participant group roles.

Create fine-grained BPM private groups

We have a small problem. It was mentioned earlier that the LDAP (and therefore VMM) security groups may have nothing to do with your business processes. Here we have a good example of that. Our participant groups are called GeneralManagers, HiringManagers, and HR Managers. There is no logical counterpart to these fine-grained roles within our LDAP security groups.

Yes, we do have groups called `ldapMgrs` and another called `adMgrs`, and yes, we can investigate their contents to ensure that they contain exactly who we need in our groups. But it should be obvious in this example that the two LDAP groups cannot possibly map cleanly to the three participant groups that are defined in this process application.

These LDAP groups are too coarse-grained. We certainly do not want to summarily grant all managers within either region the same rights to each of the three swimlanes—we require precision in our group membership. Without this precision, we will be granting authorization to these LDAP group members to execute swimlane steps that are inappropriate. There is no justification for placing an HR Manager's task in a General Manager's inbox.

There is another consideration as well: we most likely do not have read-write access to the LDAP, so we cannot just go and change the LDAP group membership to reflect the needs of our process application. Nor would we want to. We also have no idea where else within the organization these LDAP groups are being used. Any change that our application might require could easily break someone else's application.

We therefore need to create Business Process Manager private groups that contain the users who will be mapped to these roles. The process is simple: an administrator launches `/ProcessAdmin`, navigates to **User Management → Group Management**, and clicks **New Group**. This opens the Create Group dialog (Figure 4-22).

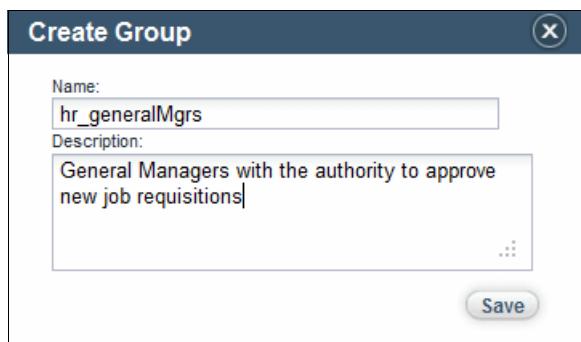


Figure 4-22 Create Group dialog

After creating the group, highlight the newly created group (Figure 4-23 on page 86), and click **Add Members**.

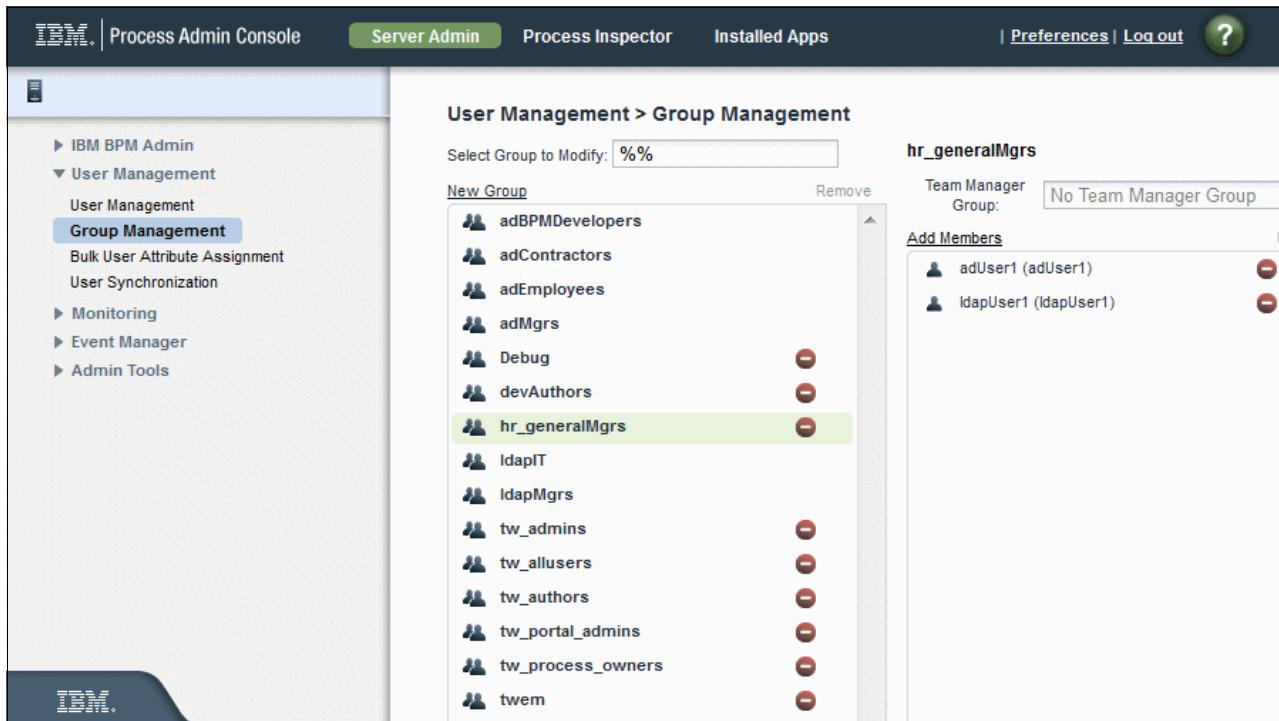


Figure 4-23 Adding members to a group

Repeat the process in order to create the following three groups with the associated members:

- ▶ hr_generalMgrs - adUser1, ldapUser1
- ▶ hr_hiringMgrs - adUser2, ldapUser2
- ▶ hr_hrMgrs - adUser3, ldapUser3

Map participant group roles to private groups

Returning now to the mapping of participant groups to BPM private groups, we can now see the appropriate mapping (Figure 4-24 on page 87).

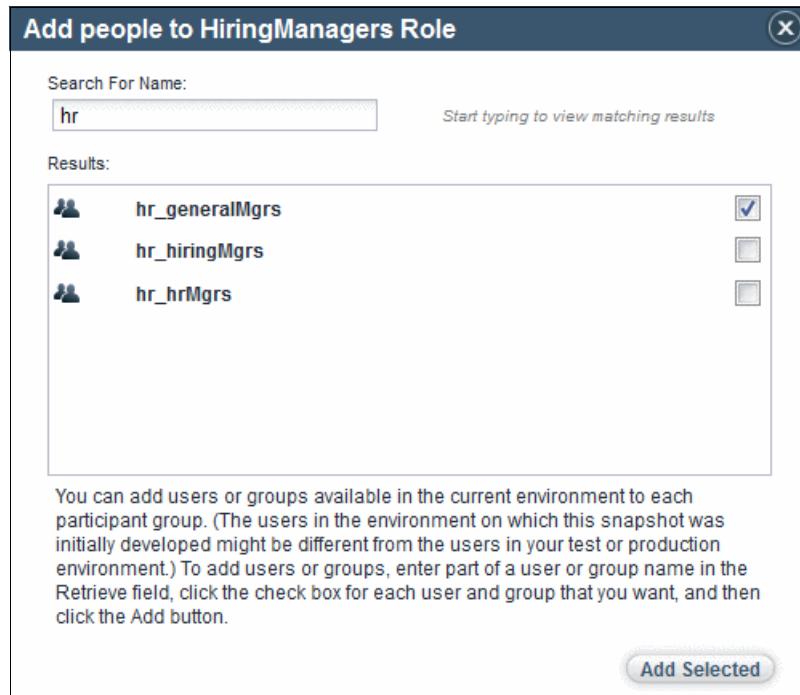


Figure 4-24 Add people to HiringManagers Role

After all three groups have been mapped, you should see the mappings shown in Figure 4-25.

The screenshot shows the "Role Bindings" tab in the Process Admin Console for the "Hiring Sample (HSS) - 7.5.1 (Active)" process. There are four entries:

- All Users:** Members: 1, Tasks: 0. Mapped to [tw_allusers](#).
- GeneralManagers:** Members: 1, Tasks: 0. Mapped to [hr_generalMgrs](#).
- HiringManagers:** Members: 1, Tasks: 0. Mapped to [hr_hiringMgrs](#).
- HRManagers:** Members: 1, Tasks: 0. Mapped to [hr_hrMgrs](#).

Figure 4-25 Role bindings for Hiring Sample

Almost perfect. There is still a mapping from All Users to `tw_allusers`, an automatically created and maintained BPM private group that represents each and every user who is represented in the VMM registry.

There was no requirement in the business process definition for “all users,” nor was there a swimlane for it. So why is it there? It is a convenience for the development process, but it has

no place once the process definition is complete. So we can safely remove the tw_allusers from our role bindings mapping. Figure 4-26 shows the final product.

Participant Group	Members: Tasks:	Add Users and Groups
All Users	0: 0	Add Users and Groups
GeneralManagers	1: 0	Add Users and Groups
HiringManagers	1: 0	Add Users and Groups
HRManagers	1: 0	Add Users and Groups

Figure 4-26 Final mappings

Seed development environment mapping

There is one last step that is required. Because these participant groups were created before we had BPM private groups to fill them, Business Process Manager cannot know beforehand which users should be allowed to receive tasks from each swimlane. Unfortunately, Business Process Manager in this case automatically defaults the participant groups to include tw_allusers. Therefore, we must go back into Process Designer and clean up this artifact. In the bottom right corner of the Process Designer, you will see a link to Hiring Managers (Figure 4-27).

Figure 4-27 HiringManagers link

Clicking this link will take you to a panel that is functionally similar to the /ProcessAdmin, where you can select users and groups to map to this participant group role (Figure 4-28 on page 89).

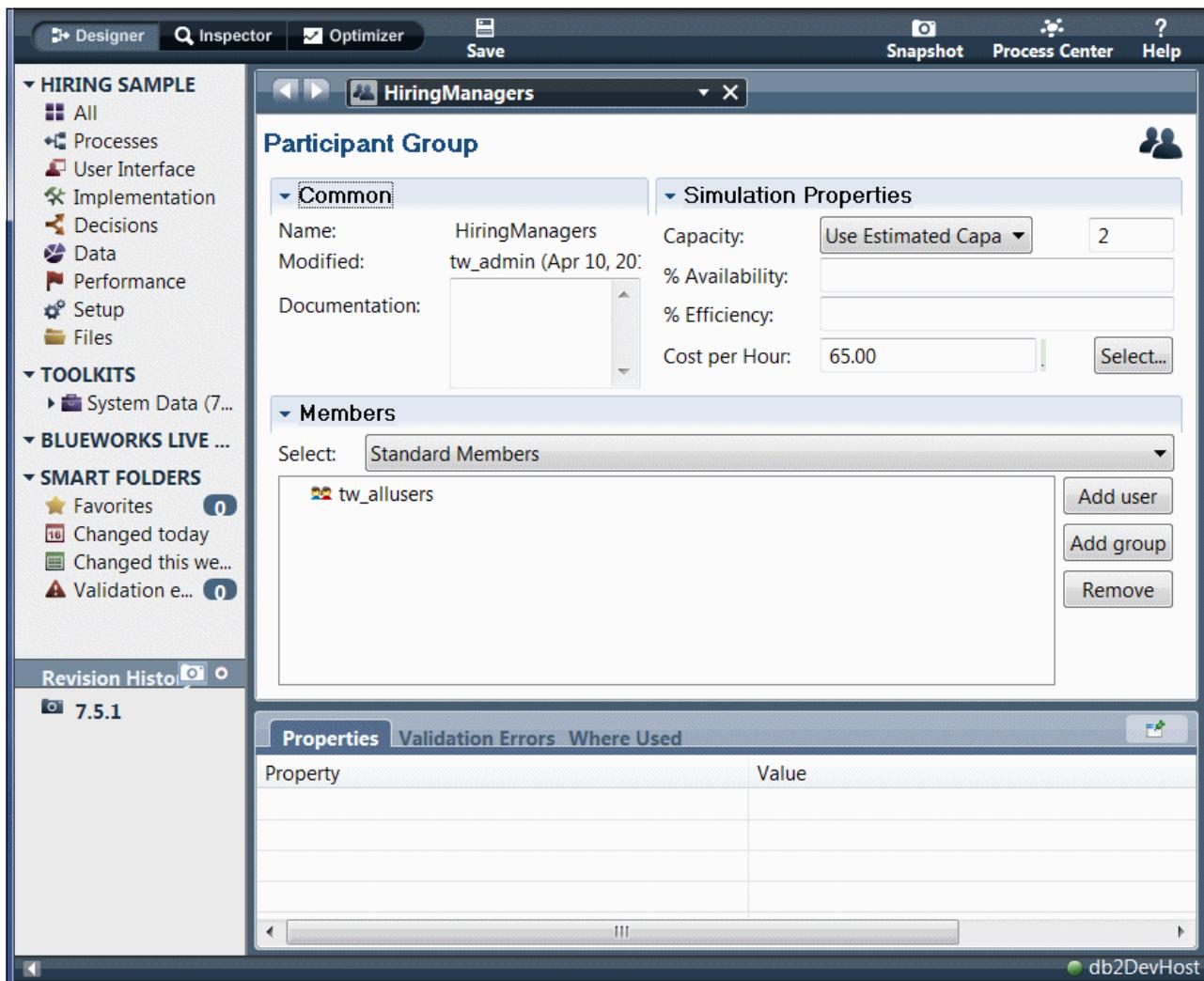


Figure 4-28 Participant group

You can see that the default is for **tw_allusers**, and we want to remove that group by clicking **Remove**. Proceeding to click **Add group** will allow you to specify **hr_hiringMgrs**, the Business Process Manager private group that we created earlier.

Once this is done, the participant group panel will look like Figure 4-29 on page 90.

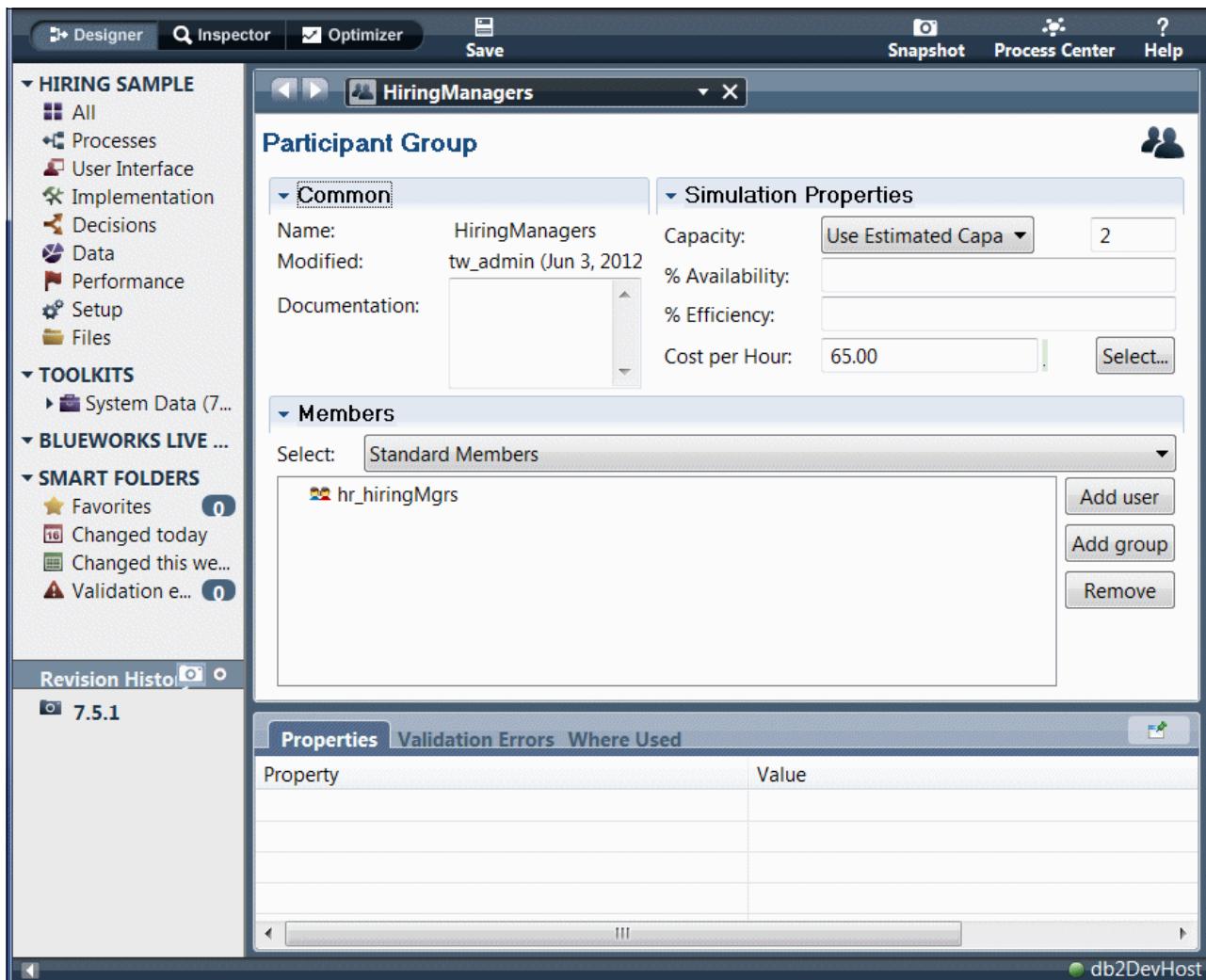


Figure 4-29 Participant group showing hr_hiringMgrs as a member

Repeat this process for General Managers and HR Managers.

Much of this functionality is hidden behind links. It could be easy to miss this default behavior. If you do not have a rigorous review process before process applications are promoted to the various environments, you could easily end up with a business process application containing a highly sensitive process definition, open and available to tw_allusers.

4.2.6 Review and summary

That is quite a lot of information to digest. Remember, complexity is the enemy of security, and so we will take a moment to review Business Process Manager's grouping mechanisms. Refer to Figure 4-30 on page 91.

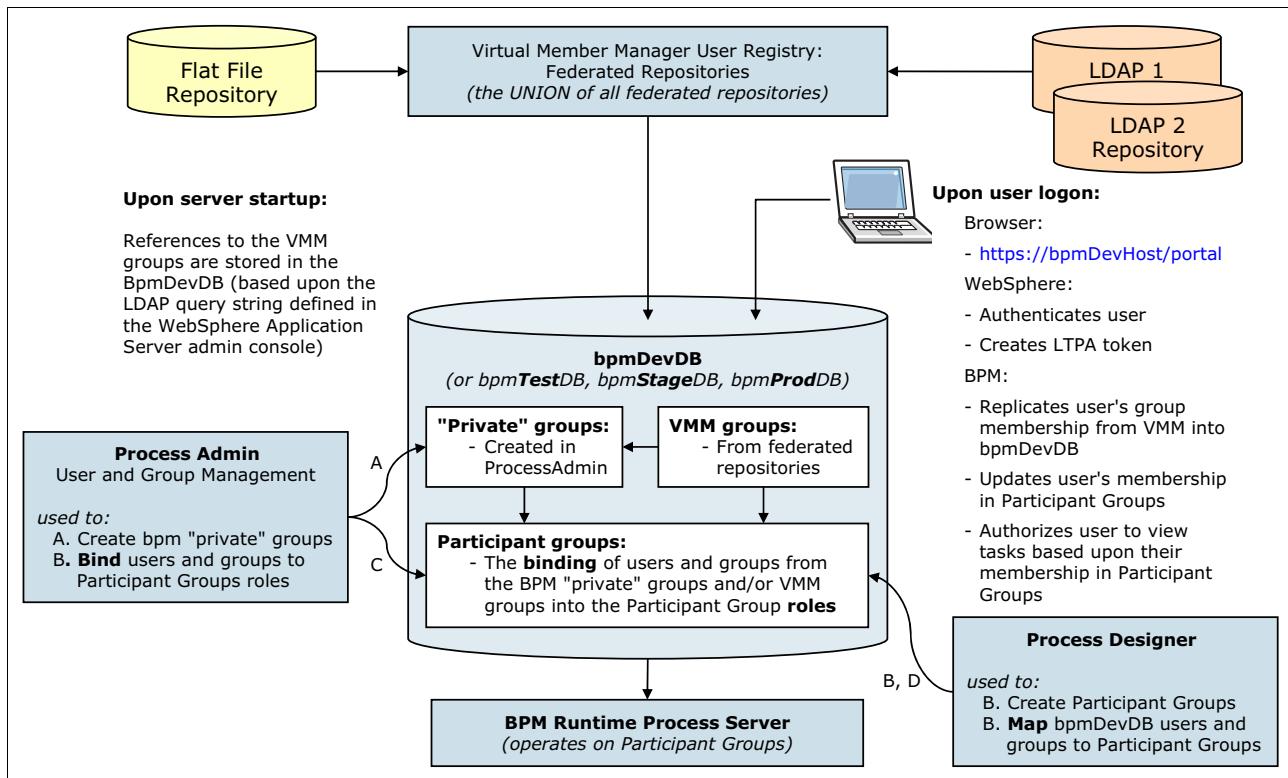


Figure 4-30 Review and summary

1. The WebSphere VMM user registry is the union of all repositories that have been federated together within it.
2. Upon Business Process Manager server startup, references to each VMM defined group are created in the Business Process Manager databases.
3. /ProcessAdmin is used to create Business Process Manager private groups, populated with users and groups from the VMM registry (step A).
4. Process Designer is used to create swimlanes and participant groups (roles) that are directly relevant to the process definition (step B).
5. /ProcessAdmin is used to bind the Business Process Manager private groups to the participant groups (roles) (step C).
6. Process Designer is used to seed the Business Process Manager participant groups with the users from the BPM private groups (step D).
7. The Process Server runtime environments execute based upon the users who have been seeded in the participant groups.

In addition, in parallel, there is the sequence of events that occur upon a Business Process Manager user logging in:

1. The Business Process Manager user points his/her browser to a Business Process Manager /ContextRoot and logs in (or, alternatively, is redirected to a Business Process Manager /ContextRoot as a result of an SSO solution).
2. WebSphere Application Server security mechanisms authenticate the user and create a LTPA token representing the user's credentials.

3. The Business Process Manager servers replicate all of the users' current VMM group membership into Business Process Manager participant groups, and authorize him/her to view tasks based upon their membership in these participant groups.

One of this chapter's most important messages is this: use the Business Process Manager private groups mechanism to allow the abstraction of the physical user repositories (complete with their potentially incompatible, vendor-specific details), into a space where business users can access and organize them in ways that are specifically targeted at the business process.

One of BPM's most salient value propositions is the enablement of business process *improvement*. This flexible and powerful abstraction is core to separating a business process' logic from its underlying logistics.

4.3 Administrative access

Heretofore in this chapter, we have been discussing Business Process Manager user access to certain predefined Business Process Manager process applications and the tasks therein. This is only one half of the authorization story. We also need to consider those users who *create* the Business Process Manager process applications.

Returning once again to the three main BPM tools for defining and administering your BPM installations:

- ▶ Integrated Solutions Console (/ibm/console)
- ▶ Process Admin Console (/ProcessAdmin)
- ▶ Process Center Console (/ProcessCenter)

We are now ready to introduce the /ProcessCenter web application. /ProcessCenter is the main administrative application which will allow you to grant the rights to author and deploy your Business Process Manager process applications to your Process Server runtime environments.

Keep in mind the following naming conventions in order to help distinguish the different functionality and modes of accessing the Process Center:

- ▶ Process Center refers to the Business Process Manager development environment, its Business Process Manager repository, and runtime server.
- ▶ /ProcessCenter console (or just /ProcessCenter) refers to the web-based user interface for accessing certain administrative functions of the Process Center. These will be discussed in this section.
- ▶ Process Designer's Process Center console, a second view into the Process Center, similar in functionality to /ProcessCenter, but accessed via the Process Designer development and authoring environment. We use /ProcessCenter to refer to both of these views.

With administrative rights to the /ProcessCenter, one can:

- ▶ Create process applications and toolkits.
- ▶ In turn grant administrative access to /ProcessCenter to other users and groups.
- ▶ Specify which Business Process Manager authors and developers have access to the Process Designer.
- ▶ Specify which users have the authority to deploy process applications.

The steps for achieving these /ProcessCenter functions are fairly obscure, and as a result, we see many organizations just using super groups to get around some of the subtleties. Our goal for this section is to demystify this process so that you can instigate a rational, fine-grained approach to managing this very powerful resource.

4.3.1 Granting access to Process Designer

Access to /ProcessCenter can be granted with or without administrative rights. In order for BPM authors or developers to use Process Designer:

- ▶ Access to /ProcessCenter is required.
- ▶ Administrative rights are not necessary, nor recommended.

Non-administrative rights allow BPM authors and developers to log into /ProcessCenter, log into Process Designer, and participate in the business process definition activities.

Administrative rights, on the other hand, come with extensive powers that are not necessary for most BPM authors or developers.

It makes sense that access to the Process Designer is granted and revoked from the /ProcessCenter, since the Process Center is the repository of all business process definitions (BPD) and artifacts that are created by the Process Designer.

It does not, however, make sense that access to Process Designer is granted from the /ProcessCenter's Admin tab, and that the Admin check box is laid out right there. It is easy to assume that, since we are adding users to the Admin tab, that we should grant them Admin rights. Unfortunately, that is the way the product is currently designed and so we have to be doubly certain to not enable the Admin checkbox.

Tip: Grant Admin rights to /ProcessCenter to few users.

Launch /ProcessCenter, and click the **Admin** tab. Business Process Manager ships with two private groups that (by default) have access to /ProcessCenter: tw_admins and tw_authors (Figure 4-31 on page 94).

The screenshot shows the 'Manage Users' section of the IBM BPM Admin interface. On the left, there is a list of users and groups under the heading 'Admin'. Two groups are listed: 'tw_admins' and 'tw_authors'. To the right of the list is a vertical sidebar with a green header 'Add Users/Groups' and a blue sub-header 'Admin'. The sidebar contains a brief description: 'From this page, you can perform administrative tasks such as granting access to the Process Center repository.' At the bottom right of the main area is the 'IBM | Process Center' logo.

Figure 4-31 Groups with access to /ProcessCenter

Note that to fully investigate the list of users who have access to /ProcessCenter and to Process Designer, it is necessary to use the **/ProcessAdmin → User Management → Group Management** function. Only by clicking each of these groups, as well as all nested groups contained therein, can one see all of the users who belong to these two groups.

For example, in a previous section we created a group called devAuthors and, in turn, added this to the tw_authors group (Figure 4-32).

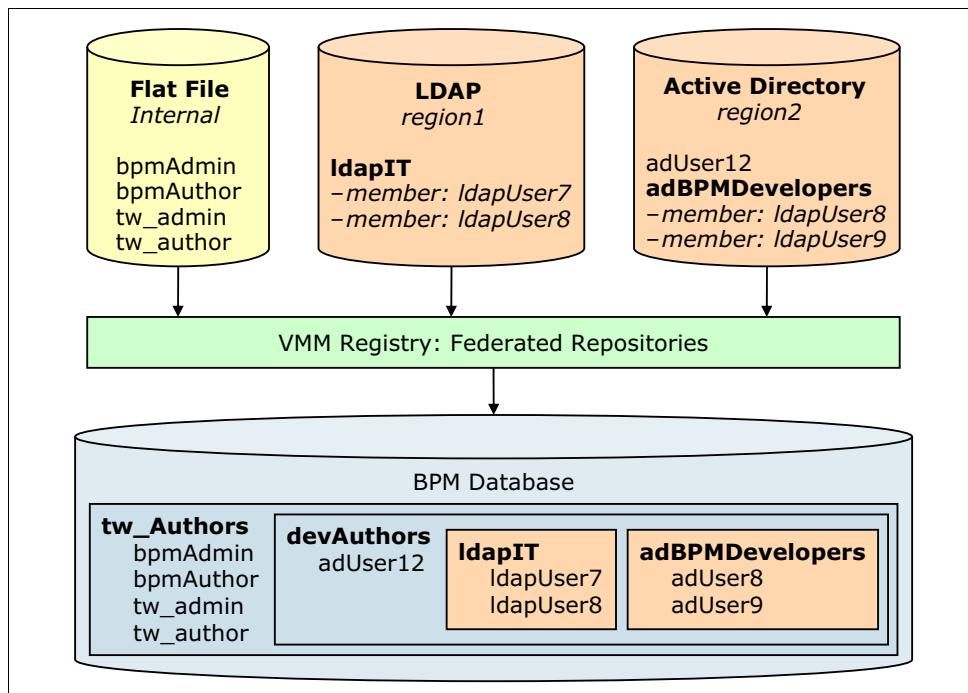


Figure 4-32 devAuthors group

As a result of that activity, the following users already have access to Process Designer due to their inclusion in the tw_authors group, and the inclusion of that group in /ProcessCenter:

- ▶ bpmAdmin
- ▶ bpmAuthor
- ▶ tw_admin
- ▶ tw_author
- ▶ adUser12
- ▶ ldapUser7
- ▶ ldapUser8
- ▶ adUser8
- ▶ adUser9

Adding users to either of these groups accomplishes the following:

- ▶ They gain access to the Process Designer. If a user is not in this list, they can still download the Process Designer to their laptop, but when they run Process Designer, they will find that they are not able to log in.
- ▶ They gain the ability to create process applications and toolkits.
- ▶ They gain read-only visibility to the System Data Toolkit.
- ▶ They gain read-only access to the Servers tab in /ProcessCenter.

Notice how this is another example of a common security hole we see: this could easily result in an overuse of default accounts. Do *not* simply add users to the tw_admins group. More on this towards the end of this chapter when we discuss common security holes we see with respect to authorization.

When a user or group is first added to the /ProcessCenter, they will most likely have no process applications visible. If they do have some, it is because they had been explicitly added to that process application's "Manage" feature by the process application's creator or other /ProcessCenter administrator.

Creating a new process application

But what users *will* have, out of the box as a new /ProcessCenter member, will be the ability to create their own process applications. As a result of having defined devAuthors and adding it to the tw_authors group, the user adUser8 can log into /ProcessCenter and create a new process application (Figure 4-33).

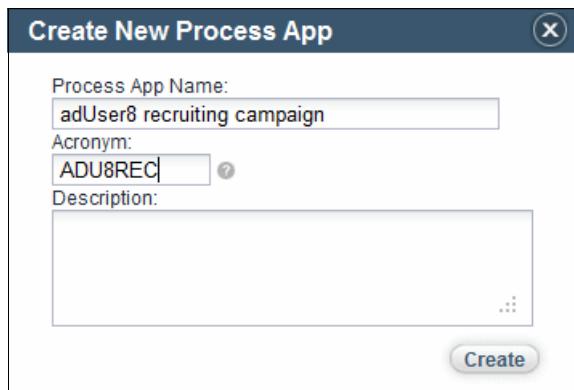


Figure 4-33 Create New Process App dialog

This will result in the panel shown in Figure 4-34 on page 96, showing that the Process App has been created.

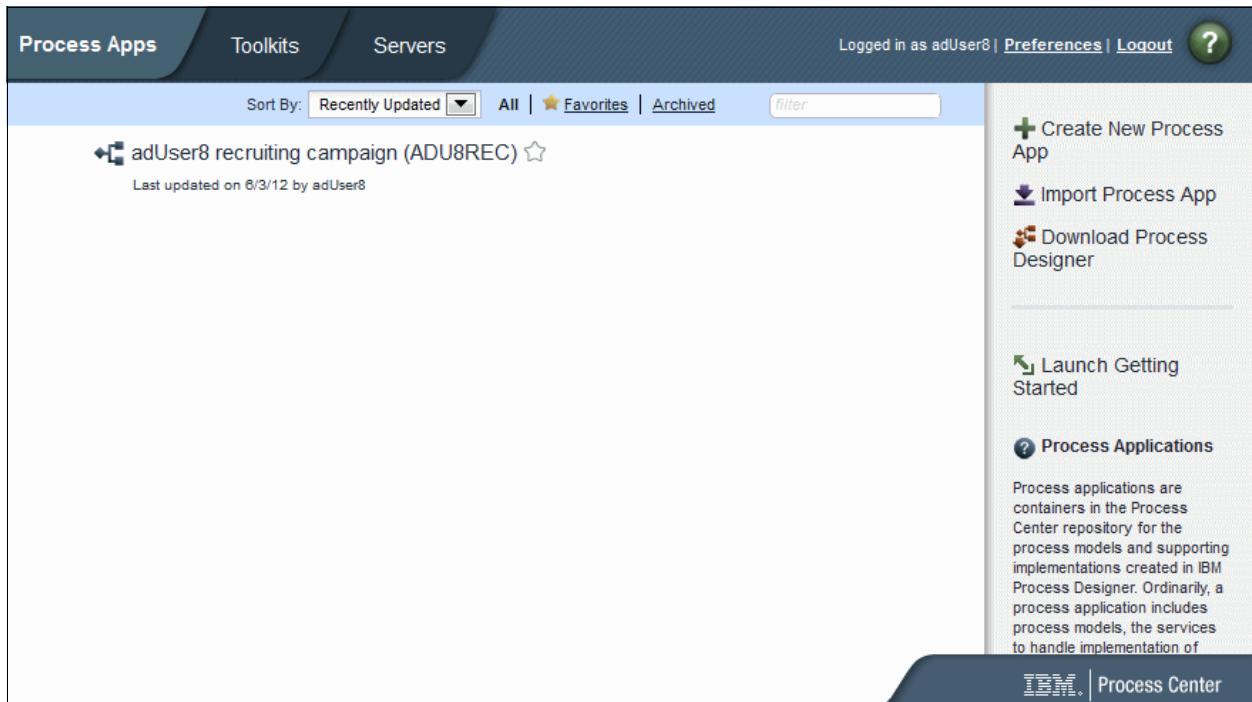


Figure 4-34 New process created

Clicking the new **adUser8 recruiting campaign** process application link, and then the **Manage** tab shows us that adUser8 is the only user who is currently authorized to work on this process application (Figure 4-35).

This screenshot shows the 'Manage' tab for the 'adUser8 recruiting campaign (ADU8REC)' process application. The left side displays basic details like the process app name, acronym, and description. The right side provides management options such as archiving or viewing tracks. A large central area is titled 'Manage Access to Process Library' and lists the user 'adUser8 (adUser8)' with checkboxes for Read, Write, and Admin permissions, all of which are checked. A 'Remove' button is also present.

Figure 4-35 adUser8 authorized to work on the process application

You will also notice that adUser8 has Read, Write and Admin privileges to this process application. This makes sense, that he would have all privileges as the process application creator. The implications of these three security levels are as follows:

- ▶ Read - Users with read access can view the process application in Process Designer, as well as any library items or toolkits that are included in the process application, but no edits are allowed.
- ▶ Write - Users with write access can open the process application in Process Designer and make changes, including creating and deleting business process definitions (BPDs), library items, or toolkits. In addition, they can create snapshots, which are used during the deployment process from Process Center to a Process Server runtime environment.
- ▶ Admin - Users with Admin access inherit all the rights of users with write access, and in addition they can use /ProcessCenter to add other users to this list, granting them read, write or admin access. In addition, process application admin users can create, edit, and delete tracks, which are subdivisions within a process application that are based upon team tasks or process application versions.

The adUser8 account, with its Admin rights to this process application, can add any other user to this managed list of users who have access to the process library. This does not imply, however, that any users added will automatically be granted access to /ProcessCenter:

- ▶ If a user is granted non-admin access to /ProcessCenter, but no access to this specific process application, they will not be able to view or modify this app.
- ▶ If a user is granted access to this process application, but not access to /ProcessCenter, they will not be able to log into either /ProcessCenter or Process Designer, and will therefore not be able to view nor modify this app.

Both permissions are required: access to /ProcessCenter, as well as rights to manage this process application. Note that both of these permissions can be read-only in order for a user to view the application.

We will now abandon this newly created process application, and instead turn our attention to the sample BPM process application which ships with Business Process Manager: Hiring Sample, since it already includes process definitions that we can leverage to further demonstrate the authorization capabilities of Business Process Manager.

Fine-grained project teams

A typical BPM project consists of a small number of business subject matter experts (SMEs), BPM process analysts, project managers and software developers. How you structure the access rights for these individuals may change over time, but in any event we recommend this approach: create BPM “private” groups for each of your BPM projects, rather than trying to add all users directly to the tw_authors or tw_admins groups.

The example that we use in this book follows a process application called Hiring Sample, and the specific business process definition (BPD) is called HR Open New Position. Business Process Manager imposes no limits with respect to which logical level you choose to create your ‘private’ groups – you can choose either the process application level Hiring Sample, or you can choose the process definition level.

Tip: Specify the absolute minimum number of individuals necessary to get the job done.

This is just one example of what we call fine-grained authorization. Typically, people think of fine-grained authorization with respect to the actions that a BPM user can take, but it applies equally well to the BPM authors. Each organization will undoubtedly have differing levels of

concern regarding which of its developers have access or visibility to applications in which they have not participated. We are recommending that SMEs, process analysts, authors and developers be given access to process apps on a need-to-access basis.

Granting access to team members

For the purposes of this book and this example, we will choose to create /ProcessAdmin and Process Designer groups at the level of the process application. We are assuming that any SME, analyst, author, or developer who is privy to the inner workings of any of the Hiring Sample processes is equally likely to be privy to all other details within this process application. This may not be true for your specific process application, so we encourage you to think this through with great attention to detail.

For the purposes of this example, consider the following users and their respective roles in the Hiring Sample process application:

- ▶ adUser1- The BPM Hiring Sample product manager
- ▶ adUser8 - A business process subject matter expert (SME) with many computer skills. He is familiar with the HR Hiring processes, and has built a number of spreadsheet macros to facilitate his work.
- ▶ IdapUser8 - A software developer

The read, write, and admin access requirements for these three users are not the same. Clearly, adUser8 and IdapUser8 will require the ability to create process definitions and make changes to the process application. For these two users (plus any future authors and developers who may join the team), we create a group called hr_hiringSample_rw, where the rw is our example's code for read-write.

Alternatively, adUser1, the project manager, will need to view the business process definitions (BPDs), but will rely upon the two authors to make any changes. Therefore, adUser1 only will require read-only access. Since there is only one user in this read-only category, it is tempting to just add him/her directly to this list. That is OK, but our recommendation is to create a hr_hiringSample_ro (ro for read-only), and to add adUser1 to that group. In this simple and small example project, it is not as important, but it is a good idea to get in the habit of using groups instead of listing individuals. This practice facilitates the sharing of tasks as your team grows in number.

Create BPM private groups for the project team

We begin with /ProcessAdmin, where we define the hr_hiringSample_rw group (Figure 4-36 on page 99).

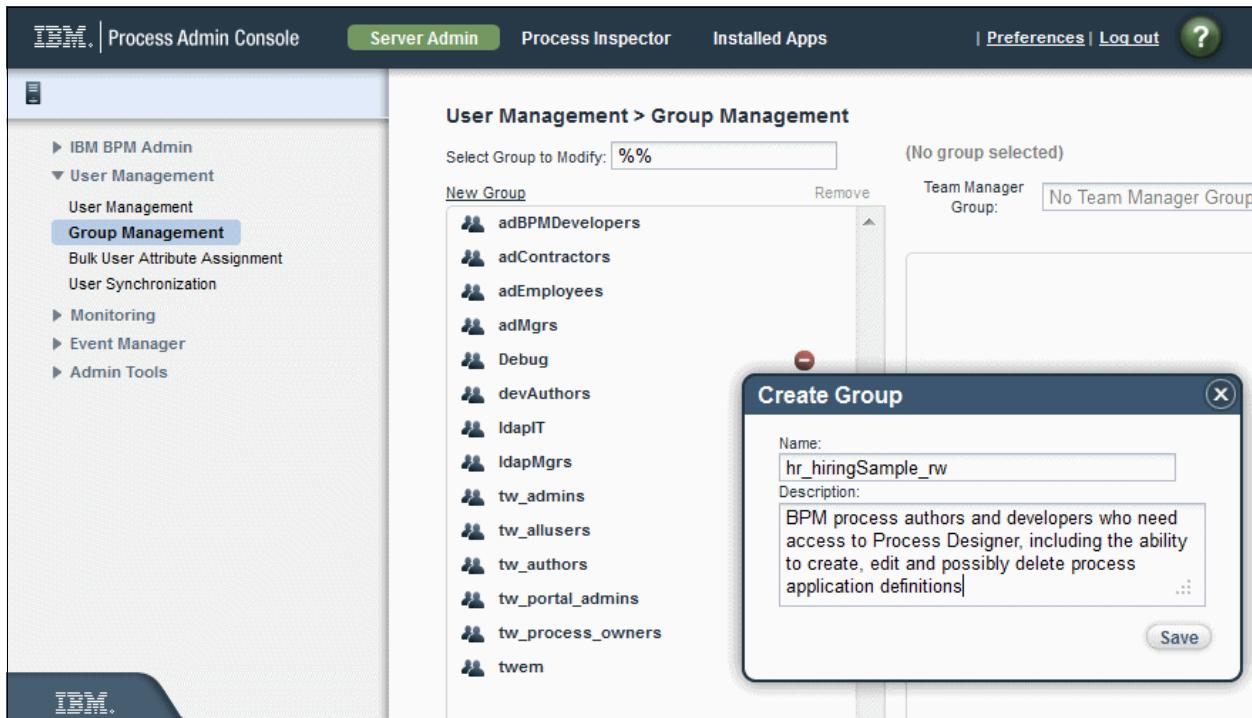


Figure 4-36 Create Group dialog

Next we add the members (Figure 4-37).

Figure 4-37 Add members to the group

Repeat the process to create hr_hiringSample_ro and include just adUser1.

Grant /ProcessCenter access to the project team

The Hiring Sample process application is owned by its creator, tw_admin (Figure 4-38).

The screenshot shows the IBM BPM Process Center interface. At the top, there are tabs for 'Process Apps', 'Toolkits', 'Servers', and 'Admin'. The 'Admin' tab is selected. In the top right, it says 'Logged in as bpmadmin | Preferences | Logout'. Below the tabs, the process application 'Hiring Sample (HSS)' is selected. The 'Manage' tab is active. On the left, there are fields for 'Process App Name' (Hiring Sample), 'Acronym' (HSS), and 'Description' (Hiring Sample). To the right of these fields is a checkbox for 'Allow users to create tracks in this process app' with a link to 'Why Tracks'. On the right side of the main content area, there is a sidebar with the following text:
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.
Below this, there are two links: 'Archive Process App' and 'View Archived Tracks'. At the bottom right of the main content area, it says 'IBM. | Process Center'.

Figure 4-38 tw_admin as process owner

In practice, you will typically have one of your team members create a process application, and that user will be listed here instead of the “Internal TW Admin user” account. In some cases you may have a “repository admin” creating the process applications and nobody from the team ever gets administration rights on the process applications. This would prevent authors from granting access.

In fact, the appearance of the Internal TW Admin user account in this sample is completely redundant. Since the tw_admin account already has full admin access to the /ProcessCenter, anyone who knows the password to this account can perform all operations—regardless of whether they are listed in this panel.

Keep your /ProcessCenter administrative accounts close. Add users to the tw_admins group on an absolute minimum basis, for they have visibility to each and every process application on the Business Process Manager server.

To make this application conform to our leading practice recommendations, we remove the tw_admin account from this list, and instead we promote just a few members of our team to take administrative control of this process application.

Begin by clicking the red “-” icon next to the tw_admin account to remove this account from our list. Next, click **Add Users/Groups** and specify our two hr_hiringSample groups (Figure 4-39 on page 101).



Figure 4-39 Add User and Groups dialog

To make sure that you grant Write access to the group hr_hiringSample_hr, click the **Write** check box as shown in Figure 4-40.

Manage Access to Process Library				
Add Users/Groups				
Read	Write	Admin		Remove
<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>	hr_hiringSample_ro	<input type="button" value="–"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="radio"/>	hr_hiringSample_rw	<input type="button" value="–"/>

Figure 4-40 Grant Write permission

Now, at this point, both adUser8 and IdapUser8 are ready to begin development, because both of these users have already been granted access to /ProcessCenter (and more to the point, to Process Designer) by virtue of their membership in devAuthors. Either can log into Process Designer, and they are both presented with the Hiring Sample process application (Figure 4-41).

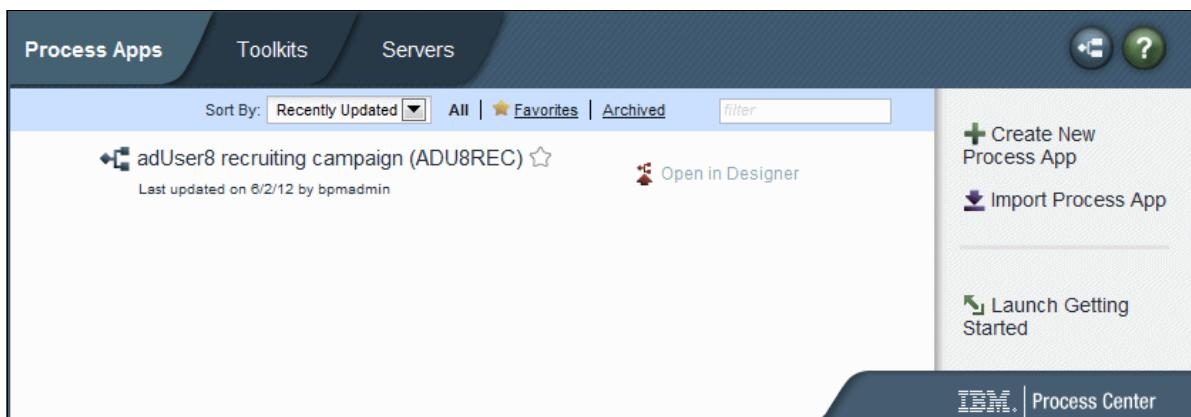


Figure 4-41 Logged in to Process Designer

However, adUser1 has not yet been given access to /ProcessCenter (Figure 4-42).

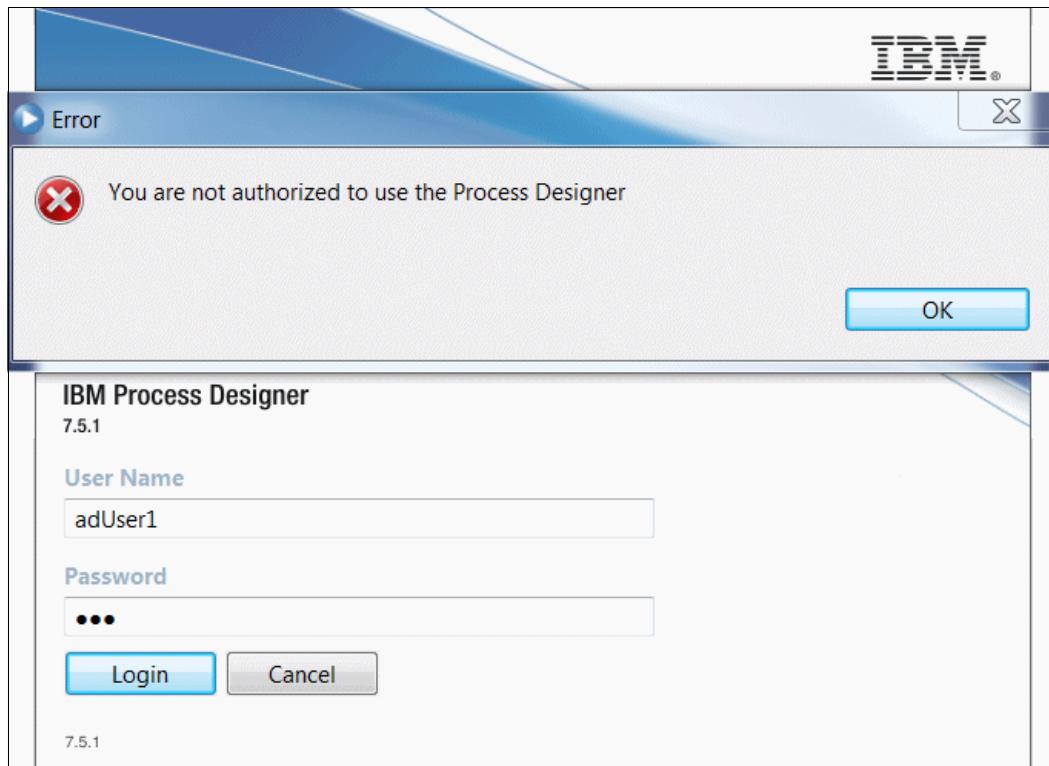


Figure 4-42 Not authorized to use Process Designer

So, for that, we need someone with full administrative rights to the /ProcessCenter to grant adUser1 rights to Process Designer. We need to add our project team's groups to the **/ProcessCenter → Admin tab** (Figure 4-43 on page 103).

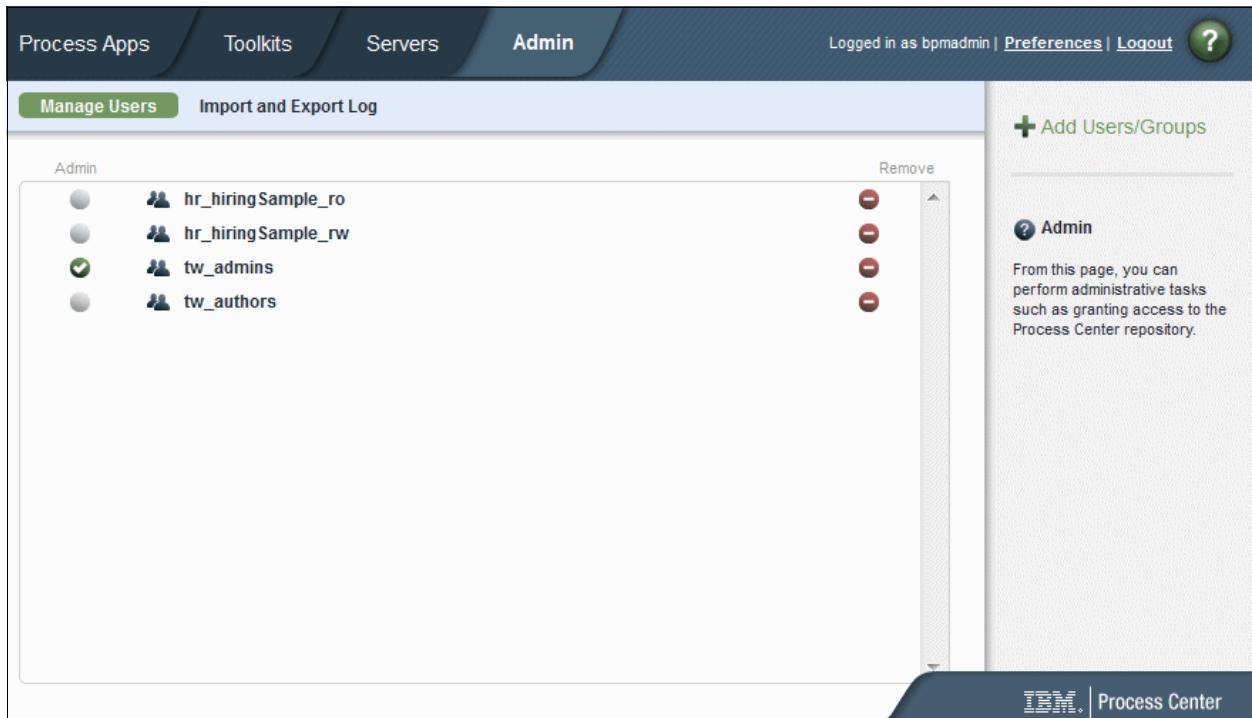


Figure 4-43 Admin tab

Notice that adUser8 and ldapUser8 were already granted access by virtue of their membership in devAuthors, because tw_authors includes devAuthors. Their inclusion in hr_hiringSample_rw here is redundant. Notwithstanding, as your BPM project adoption grows from the typical Quick-Win Pilot project to a more encompassing program and ultimately to enterprise-wide adoption, it is extremely important that you maintain a fine-grained management of where users appear in this /ProcessCenter Admin tab. We would therefore recommend that you now go and remove them from the tw_authors group.

Tip: Resist the temptation to add users to any all-encompassing groups (such as tw_authors or tw_admins) and instead create smaller groups that are directly related to specific requirements.

4.3.2 Review and summary

As it was with the section on BPM grouping mechanisms, there is a lot of information to assimilate with respect to BPM administrative access, so we will take another moment to summarize.

The first distinction we would like to point out is that the /ProcessCenter, Process Designer, and Integration Designer are all intended to act upon the Process Center (Figure 4-44 on page 104).

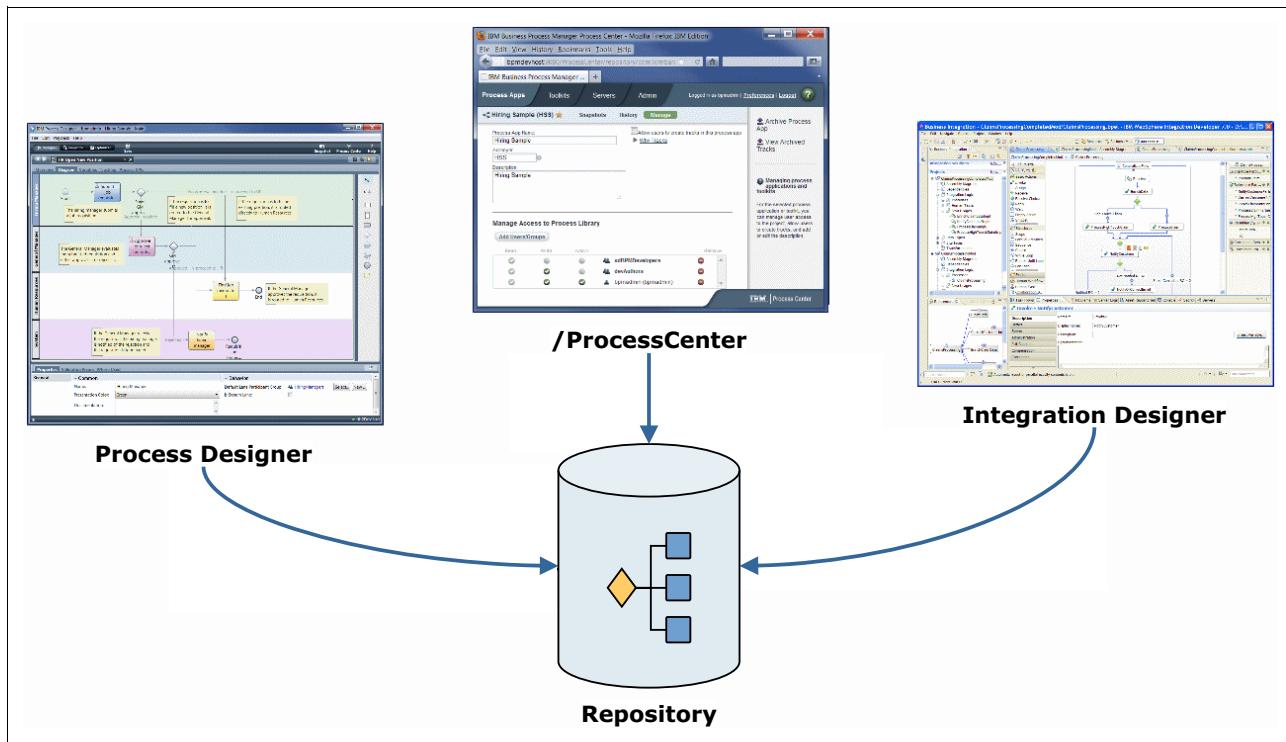


Figure 4-44 Access to the common repository

Similarly, the **/ProcessAdmin** is intended to operate on each environment's Process Server runtime environments (Figure 4-45 on page 105).

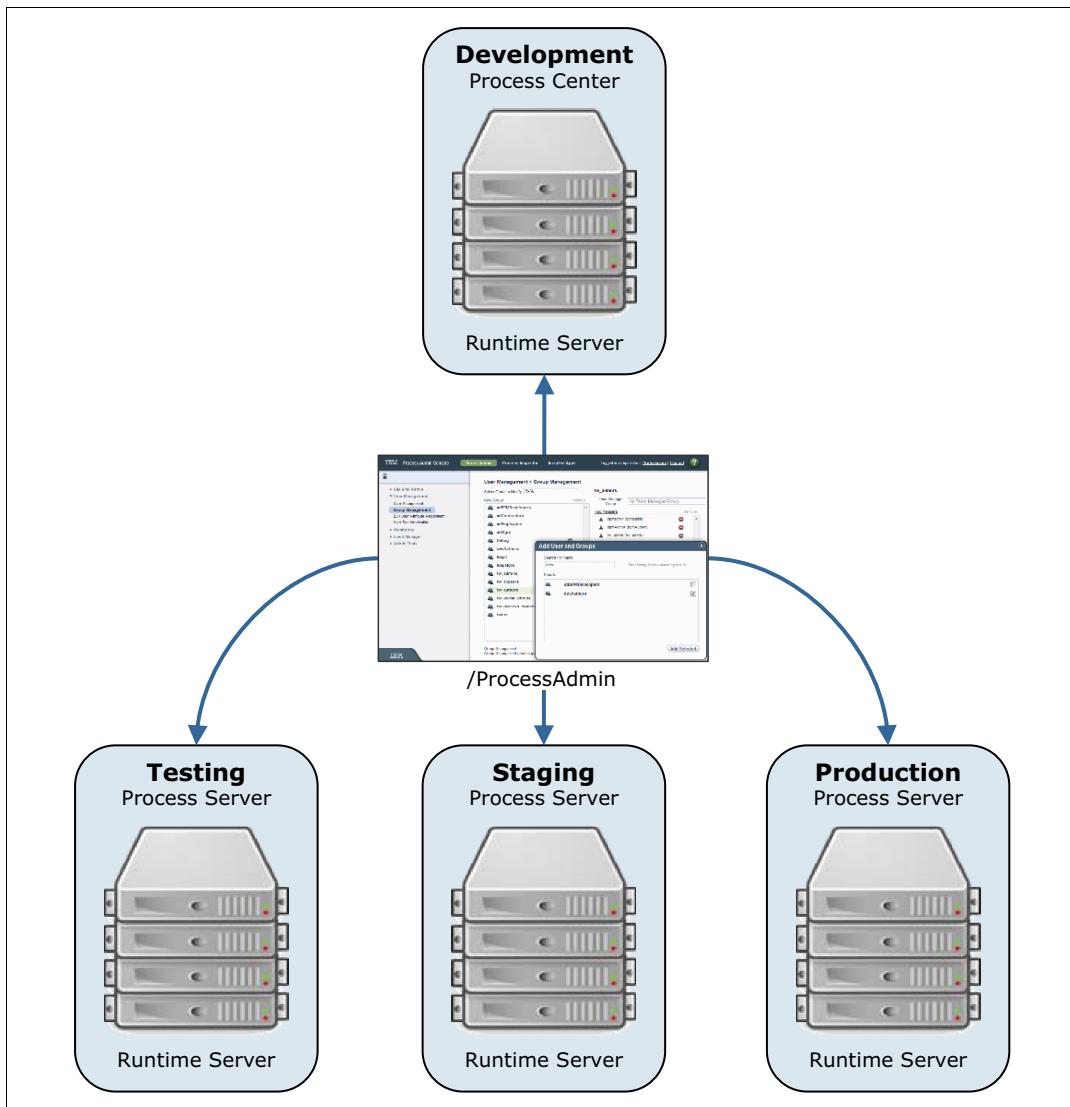


Figure 4-45 /ProcessAdmin

Now, having said that, there is still a great deal of interplay between these Business Process Manager product components when you need to put them into practical use.

The steps to giving BPM authors and developers access to Process Designer are as follows:

1. Use /ProcessAdmin to create project team Business Process Manager private groups. Create separate groups for read-only as well as read-write access. Consider the possibility that you may not need admin access.
2. Add only those users to each of these two groups who have a demonstrated need to access the Process Designer in the read-only or read-write capacity. For example, do not put read-only users in the read-write groups.
3. Use /ProcessCenter to add these two groups (or three, if you decide you need project-level administrative representation) to the **/ProcessCenter → Admin** tab. This will grant both groups access to Process Designer. Do not enable the Admin checkbox in this tab.
4. Create the process application by navigating to **/ProcessCenter → Process Apps** and clicking **+ Create New Process App**.

5. Add these project team groups to the process application by navigating to the **/ProcessCenter → Process Apps → Manage** tab and clicking **Add Users/Groups**. This will enable the users to view and/or edit the process app. Consider removing the tw_admin account, which has Admin access.
6. Be sure to give write access to the read-write group.

4.4 Instance-based authorization

It has already been mentioned that Business Process Manager offers a very fine-grained, instance-based authorization mechanism. There are three levels of this fine-grained authorization:

- ▶ Level 1 - The use of swimlanes and participant groups
- ▶ Level 2 - The use of routing policies
- ▶ Level 3 - The ability to selectively hide UI elements (widgets) based upon runtime criteria

We have already covered Level 1 in the previous section. Level 2, the use of routing policies, uses the Business Process Manager runtime engine to allow for a series of complex, real-time decisions to affect an in-flight task's destination (Figure 4-46).

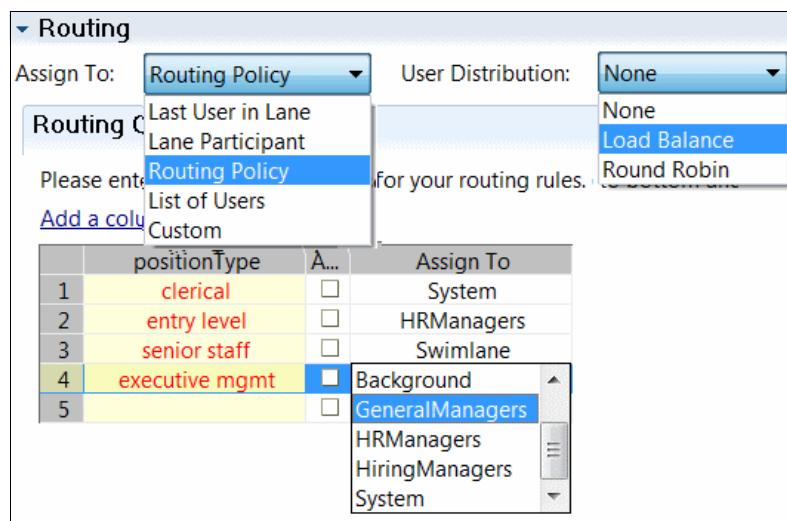


Figure 4-46 Routing

For example, a task can be rerouted to the last person (within the given swimlane) to have touched this task, which can be helpful in establishing continuity. Or it can be routed to the next available member of a given swimlane. It can be routed to one of a given list of specified users, or it can be routed based upon the outcome of a custom JavaScript that is evaluated at run time.

With respect to the third level of instance-based authorization, the capability to selectively hide user interface elements deserves special consideration (Figure 4-47 on page 107).

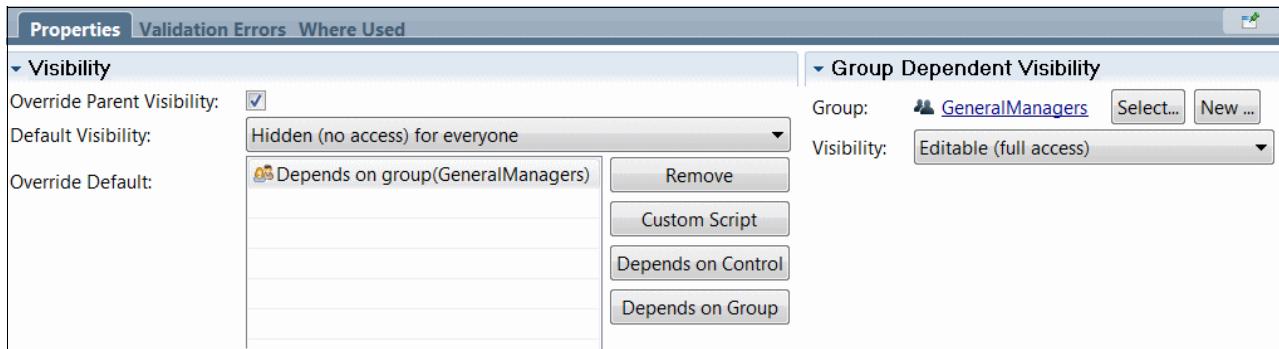


Figure 4-47 Hiding user interface elements

This feature truly does hide the information (using dynamic HTML), it does not remove it. As a result, it is possible that some data might be held within UI widgets and therefore be visible to anyone with enough browser skills to navigate and browse the document object model. Furthermore, if the network protocol is not secured, then this opens the door to users who are not even sent the task.

Tip: Use SSL/TLS between Business Process Manager and all users.

In another type of authorization, a BPD can be exposed to start. This refers to showing it as a startable item in the Process Portal (the same setting exists for Human Service, which can be added to Process Portal and Process Admin Console). Expose Business Data is for Exposed Process Variables (EPVs).

4.5 Common security holes

In this section we describe security holes that are most common in business process management programs.

4.5.1 Overuse of administrator privileges

Notice in the /ProcessAdmin Admin tab list of users and groups, the column to the far left entitled Admin, which has a grey or green check box, as shown in (Figure 4-48 on page 108). By default, it is *not* enabled, and you should be careful about changing that default.

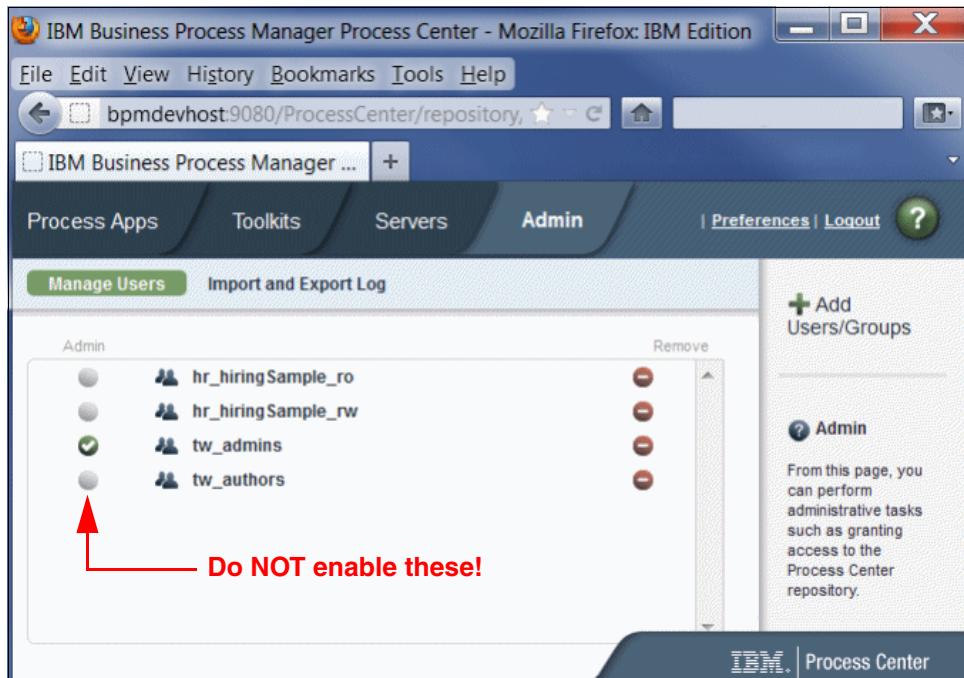


Figure 4-48 Admin check box

This little check box can be deceiving. There are a lot of implicit permissions granted when you enable this check box. Enable this option only for a few trusted users. Enabling this option grants the user or group the following permissions:

- ▶ They can read and edit *any* process application.
- ▶ They can create snapshots of *any* process application.
- ▶ They can deploy snapshots of *any* process application to the runtime Process Servers in all of the Business Process Manager deployment environments (test, stage, and production).

But far more importantly, they can grant /ProcessCenter Admin rights to *any* Business Process Manager user and to any process application.

This option is easily overlooked, especially given that the option is placed on the Admin tab for the /ProcessCenter. What is more natural than granting Admin access to users who appear on the Admin tab? It is another unfortunate choice of labels, so let us be clear:

- ▶ The appearance of a user or group in this list simply grants that user or group access to Process Designer.
- ▶ The enabled check box next to their name grants them super-user status.

Enable this option sparingly.

4.5.2 Failure to map participant groups

When a new swimlane is introduced to a Business Process Manager process definition, the default participant group defaults to All Users. Notice the bottom of the Process Designer panel has a Properties view (Figure 4-49 on page 109). With a swimlane selected, the Behavior section shows the Default Lane Participant Group for this swimlane.

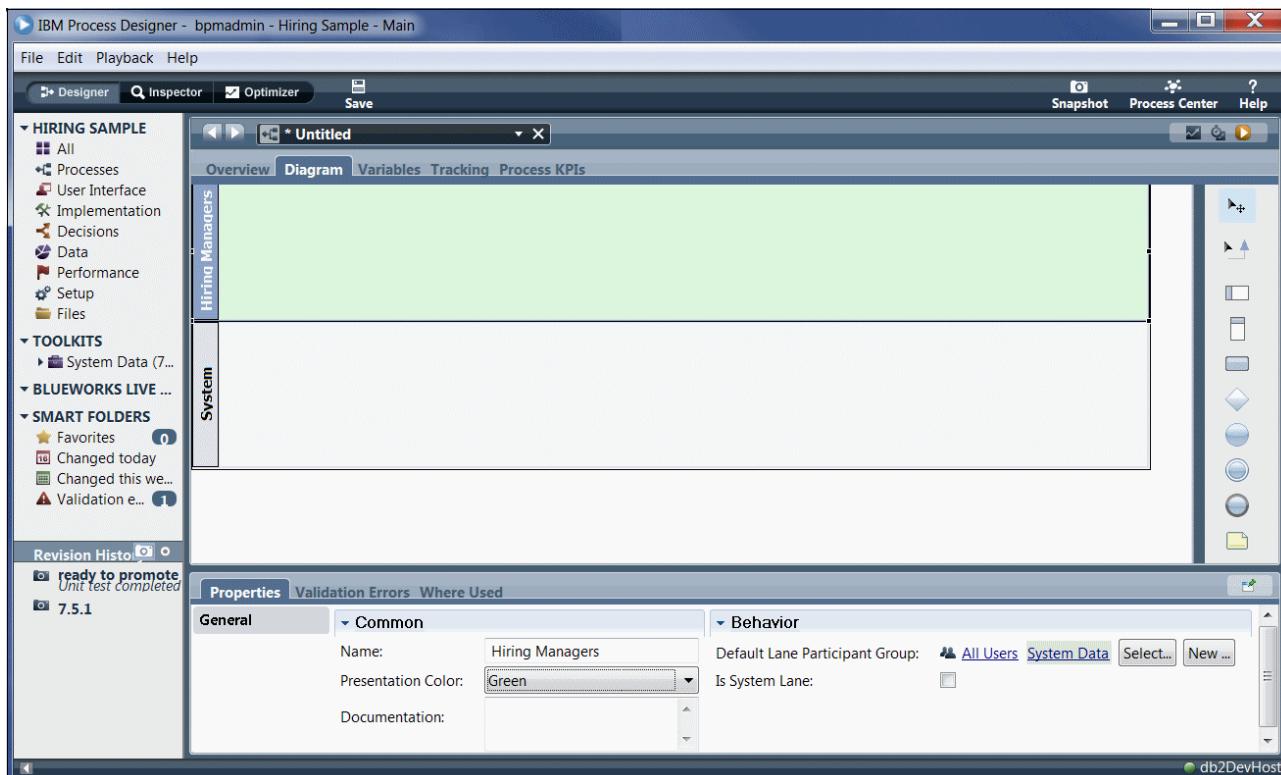


Figure 4-49 Process Designer Properties view

Now, if the process definition modeling occurs in advance of the creation of the Business Process Manager private groups—which is common—then there may be no private group to map to this newly created swimlane and participant group. For purposes of modeling, this is not a problem. Additional swimlanes can be created, new participant groups associated with them, and no thought necessarily need be given to the actual users who will be performing in these roles.

However, one of the main tenets of the Business Process Manager agile methodology is that the process applications are “played back” frequently in order to demonstrate that the process is being modeled correctly. In order for the playbacks to proceed, we need some concrete users for these roles.

This presents us with a bit of a chicken-and-egg problem. The business process diagram has defined a participant group (a role) and we need members from the Business Process Manager database to populate this role, but which private, VMM, or LDAP security group best matches this participant group (role)?

It is too easy to just leave the default of All Users in place, or to use already-existing LDAP groups and call it a job well done. After all, the business process now has concrete users defined and mapped to the process application’s roles—and the playback will proceed just fine.

The security hole occurs when the project team forgets to go back and define this mapping. One has to specifically go looking for mismatches of this nature by actively reviewing the behind-the-scenes association of participant group to Business Process Manager private group to ensure that the mapping is as intended.

This is an important point: if tw_allusers is allowed to stay, then you are effectively, completely, turning off authorization for all tasks in this swimlane. If you are doing this for one swimlane,

the chances are that you are doing the same for most, if not all, swimlanes in the process application.

Tip: Use a rigorous review process to ensure that each and every swimlane or participant group is mapped to a Business Process Manager private group that includes only those users who should have authority to execute the steps within the swimlane.

4.5.3 Overpopulation of groups

Perhaps even harder to diagnose will be those cases where the swimlanes and participant groups have been mapped to Business Process Manager private groups, but perhaps these private groups will be close-but-not-quite exactly what the process definition requires.

As you could see in our example earlier in the chapter, we created separate Business Process Manager private groups for each function we required: there were two groups for the developers, and three groups for the users. Some of the LDAP users were actually included in both camps: adUser1 was not only a hr_hiringSample_ro member, but also a hr_generalMgr member. There is nothing wrong with that. The problem occurs when organizations combine all of these functions, perhaps using a single generic hr_mgrs private group—knowing that adUser1 belongs in both anyway.

The security hole occurs when another hr_mgr is added to the group. Perhaps this new manager belongs in the user-facing generalMgr group, but not in the developer-facing read-only group. It simply is not safe to assume that whoever is asked to add this new manager into the generic hr_mgrs group is going to be familiar enough with the application's design to catch this. The result may be that you have granted this new manager authorization to do something that a more careful review would have prevented.

Tip: Use fine-grained groups for each functional role that your process application can conceive.

This requires the up-front discipline to create appropriately themed private groups, as well as the ongoing maintenance and scrutiny to ensure that these processes are followed without fail.

4.5.4 Overuse of tw_authors, tw_admins

The process of ensuring that an author or developer has adequate authorization to create and deploy applications is rather a daunting task. As a result, we see many organizations simply adding their authors and developers into the default groups of tw_authors or tw_admins.

What you need to understand is that membership in these all-encompassing groups grants these accounts super user status—and visibility to all process applications that are installed in your environment. This is almost universally undesirable. Access to /ProcessCenter and Process Designer should be granted in small, highly related chunks. Create project team groups as we have shown in this book (hr_hiringSample_ro and hr_hiringSample_rw), which closely reflect the roles that these authors or developers play in the processes being modeled.

4.5.5 Faith in firewalls

And, of course, there's the ever-present faith in firewalls.

Do not underestimate the amount of information that can be gathered by a curious, motivated, or perhaps mischievous user. If a user can sniff the network traffic, then they can analyze it. If they can analyze it, they can spoof it. It is a short path from unencrypted network traffic to unauthorized access.

Specifically, given Business Process Manager's ability to perform instance-based authorization based upon runtime criteria, it is certainly conceivable that someone might be able to sniff an in-flight process and alter its authorization criteria.

Encrypt all communications links between the following components:

- ▶ Business Process Manager and LDAP
- ▶ Business Process Manager and database
- ▶ Business Process Manager and web or proxy servers
- ▶ Business Process Manager and any web services hosts
- ▶ Process Center and Process Server
- ▶ Process Center and Process Designer
- ▶ Process Center and Integration Designer
- ▶ Process Servers and users



Integration: Working with others

An important aspect in Business Process Manager security is integration. In this chapter we focus on integration in the context of Business Process Manager Standard and Advanced Edition.

5.1 Business Process Manager Standard Edition versus Advanced Edition

One of the most significant differences between Business Process Manager Standard Edition and Business Process Manager Advanced Edition lies in their handling of integrations in general and web services specifically.

With respect to Business Process Manager Standard Edition, we look at web services from the point of view of two separate use cases:

- ▶ Outbound: Business Process Manager processes calling external web services
- ▶ Inbound: Business Process Manager processes exposed as web services for others to consume

It is necessary to look at these separately because the mechanism underlying these two use cases differs significantly. Business Process Manager 7.5.1 Standard and earlier versions, including WebSphere Lombardi Edition, base their web services implementations upon the Apache foundation's AXIS 1.3 toolkit (earlier versions were based on 1.2), which is an implementation of the draft W3 specification SOAP protocol.

Note, however, that with Business Process Manager V8.0 Standard, there is some movement away from the older Apache AXIS foundation. For outbound web service calls only, in Business Process Manager V8.0, the web services implementations are based on JAX-WS 2.2. For Business Process Manager Standard Edition V8.0 this move brings no additional functionality, but it does represent the first steps towards the long-term goal of replacing this foundation with the newer JAX-WS standards.

Business Process Manager Standard Edition V7.5, on the other hand, is more feature-rich with respect to all things related to web services. Business Process Manager Advanced is:

- ▶ Based upon JAX-WS 2.1 (JAX-WS 2.2 for Business Process Manager V8.0)
- ▶ Uses Policy Sets and Bindings
- ▶ Includes Integration Designer
- ▶ Ships with a fully licensed copy of the WebSphere Enterprise Service Bus

The greatest difference between Business Process Manager Standard's foundation of AXIS and Business Process Manager Advanced's foundation of JAX-WS is support for the most popular WS-* specifications. There are a host of APIs contained in WS-* that provide for a greatly enhanced security model than that available to the AXIS implementation.

5.2 Business Process Manager Standard Edition outbound web services

In order for a Business Process Manager Standard Edition business process application to call an outbound web service, an *Integration Service* needs to be built and populated with either a Web Services or a Java integration (in the case of Business Process Manager Standard Edition), or an *Advanced Integration Service* (in the case of Business Process Manager Advanced Edition).

5.2.1 Using Web Service Integration

Including an outbound web service call in a Business Process Manager Standard Edition V7.5 business process definition (BPD) can be quite simple.

If the Web Services Description Language (WSDL) document, which is associated with the foreign web service, is well-constructed (and to some degree, not overly complex), then all that is required to utilize this web service in a Business Process Manager process is to drag a *Web Service Integration* widget from the right side of the Process Designer's design view (Figure 5-1).

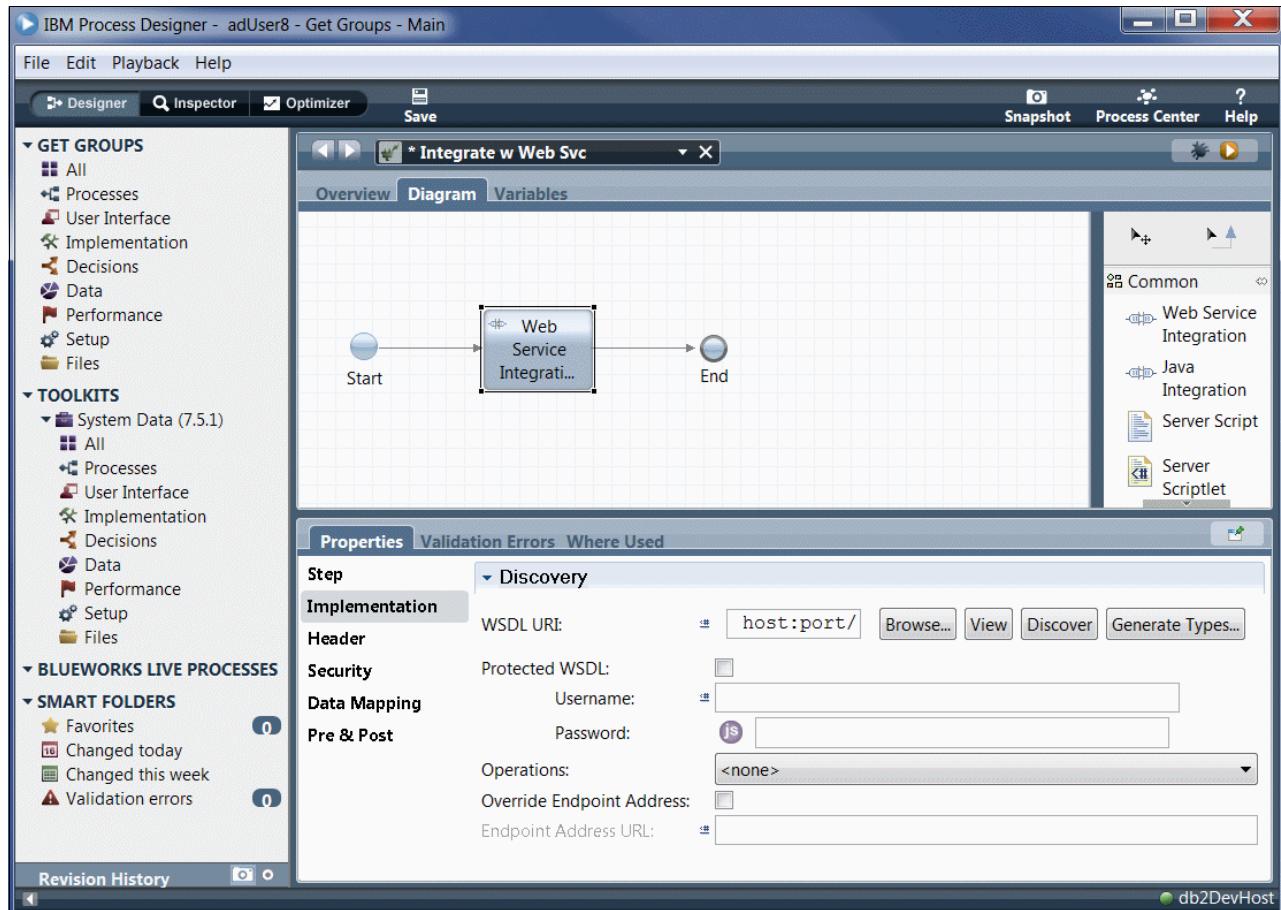


Figure 5-1 Web Service Integration widget

Discovering the WSDL

WSDL is an XML document that describes the web service's operations and data input and output formats and requirements. In the Implementation section of the properties tab (below the main diagram area), you will find a few buttons that facilitate the integration of the foreign web service into the Business Process Manager process application. You can:

- ▶ *Browse* either a WebSphere Service Registry and Repository or a UDDI Registry
- ▶ *View* the contents of the specified WSDL
- ▶ *Discover* the operations defined in the WSDL
- ▶ *Generate Types* based upon a selected web service operation

Regardless of the method used to select an operation, if the web service is considered a protected asset by the web server that is hosting the web service, then you will need to supply a username and password.

At this point, we are only talking about requiring the username and password during the discovery and type generation process, but it is still important to note that these credentials are passed via *HTTP Basic Authentication*.

Clicking **View** enables us to see the full text of the WSDL. Selecting **Generate Types** will parse the WSDL document and automatically:

- ▶ Populate the Operations pull-down menu item in the above panel.
- ▶ Create any input and output data types that are required for any of the WSDL-defined operations.

For purposes of illustration, we will be utilizing a Web Service Integration, which integrates with the Business Process Manager Web API. This is a set of web services published by Business Process Manager that give developers access to the Business Process Manager engine in order to be used by external applications. Clearly, an API of this sort would be considered a *protected asset*, and as such, we will need to specify a username and password in order to discover the WSDL.

The Business Process Manager Web API is accessed via:

- ▶ WSDL URI:
[https://bpmdchost:\[port\]/webapi/services/WebAPIService?WSDL](https://bpmdchost:[port]/webapi/services/WebAPIService?WSDL)
- ▶ Protected WSDL: enabled
- ▶ Username: yourBPMadminUserId
- ▶ Password: yourBPMadminPswd

We use an HTTPS URL because we send credentials when fetching the WSDL. The results of selecting **Generate Types** can be seen immediately in the Operations pull-down menu of the Implementation tab (Figure 5-2 on page 117).

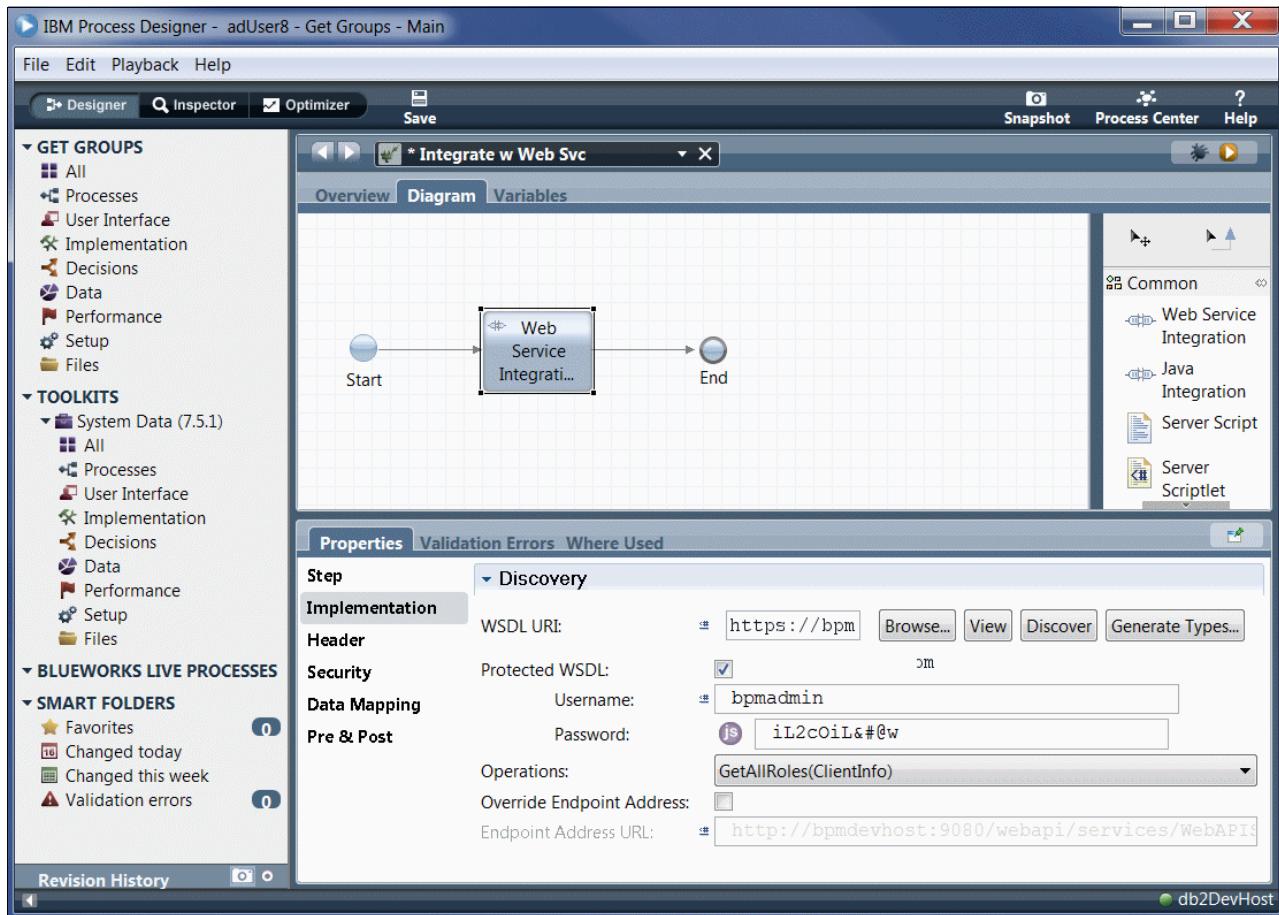


Figure 5-2 Implementation tab

Here you can see that the WSDL-defined operation `GetAllRoles()` has been selected.

Mapping to BPM variables

As a part of the Generate Types discovery process, the WSDL is parsed and all data types represented within are added to the process application's list of available data types along with the default types (String, Integer, Date, Document, and so on). Figure 5-3 shows the portions of the WSDL document that have to do with the `GetAllRoles()` operation:

The screenshot shows a window titled "TextPad - C:\temp\webAPI extract.wsdl *". The menu bar includes File, Edit, Search, View, Tools, Macros, Configure, Window, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and Find. A search bar at the top right says "Find incrementally" with a "Match case" checkbox. The main area displays the WSDL code for a web service named "webAPI extract.wsdl". The code defines an operation "GetAllRoles" with a request message "GetAllRolesRequest" and a response message "GetAllRolesResponse". The response message has a part "parameters" of type "tns:ArrayOfRole". The operation also has documentation and input/output messages. Red boxes highlight the "parameters" part of the response message and the "ClientInfo" header part of the request message.

```
<?xml version="1.0"?>
<wsdl:definitions name="webAPI extract" targetNamespace="http://lombardisoftware.com/webapi" xmlns:tns="http://lombardisoftware.com/webapi" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <xsd:schema targetNamespace="http://lombardisoftware.com/webapi">
      <xsd:complexType name="ArrayOfRole">
        <xsd:sequence>
          <xsd:element name="Role" type="tns:Role" maxOccurs="unbounded" minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="Role">
        <xsd:sequence>
          <xsd:element name="id" type="xsd:int" />
          <xsd:element name="name" type="xsd:string" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="GetAllRolesRequest">
    <wsdl:part element="tns:GetAllRoles" name="parameters" />
  </wsdl:message>
  <wsdl:message name="GetAllRolesResponse">
    <wsdl:part element="tns:GetAllRolesResponse" name="parameters" />
  </wsdl:message>
  <wsdl:operation name="GetAllRoles">
    <wsdl:documentation>Get all roles in the system @return The roles</wsdl:documentation>
    <wsdl:input message="tns: GetAllRolesRequest" name="GetAllRolesRequest" />
    <wsdl:output message="tns: GetAllRolesResponse" name="GetAllRolesResponse" />
  </wsdl:operation>
  <wsdl:operation name="GetAllRoles">
    <soap:operation soapAction="http://webapi.lombardisoftware.com/GetAllRoles" />
    <wsdl:input name="GetAllRolesRequest">
      <soap:body use="literal" />
      <soap:header message="tns:ClientInfo" part="ClientInfo" use="literal" />
    </wsdl:input>
    <wsdl:output name="GetAllRolesResponse">
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:definitions>
```

Figure 5-3 WSDL for GetAllRoles

You can see that there are two types of messages (a request and a response), and two data types (the request requires a soap header of type ClientInfo, and the response returns an array of Role objects).

Figure 5-4 on page 119 shows the Web Service Integration Variables tab showing how these two automatically-generated data types can be used.

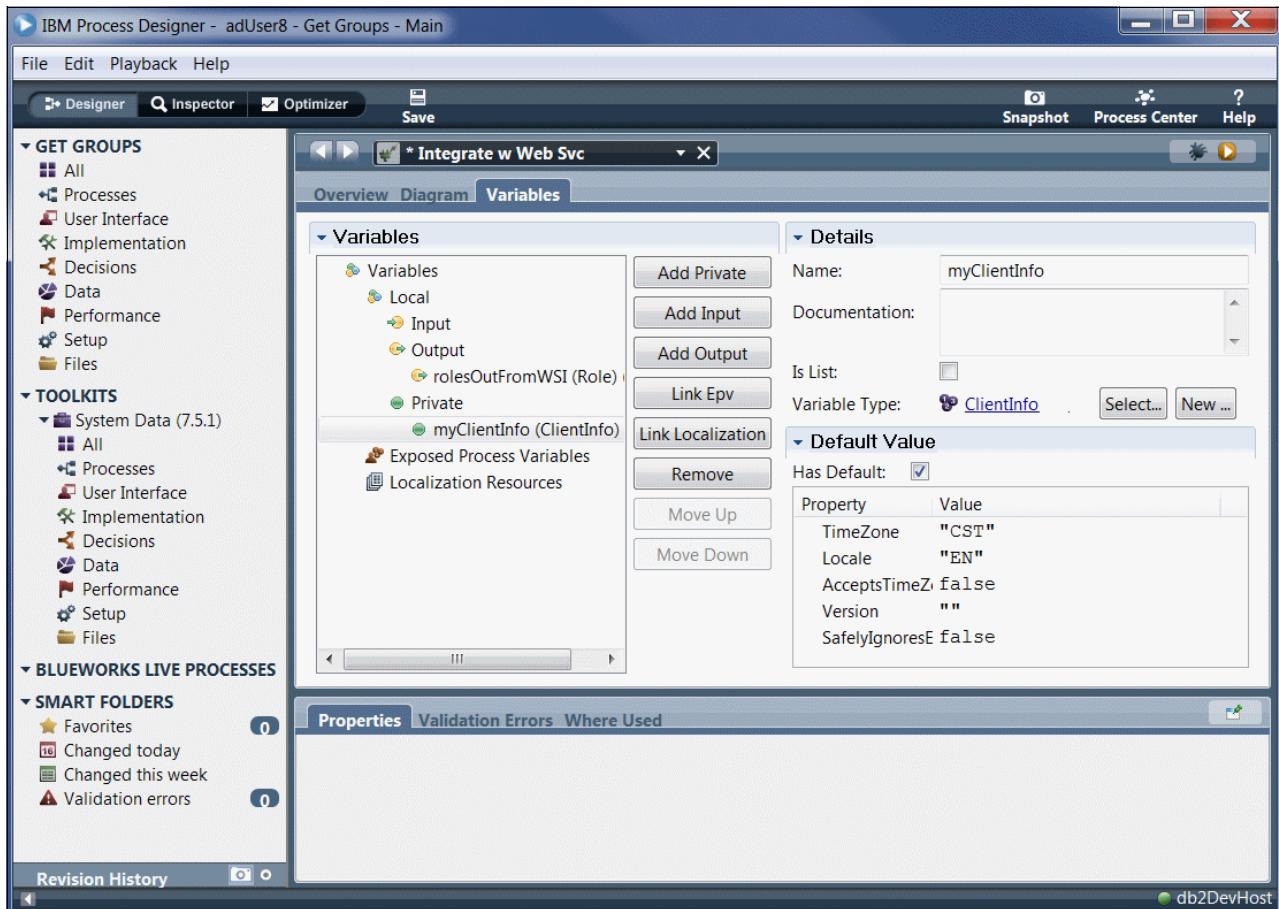


Figure 5-4 Variables tab

Ensuring security during web service execution

In “Discovering the WSDL” on page 115, we specified a username and password which were required because the WSDL is considered a protected asset. It will come as no surprise that the web services described therein are also considered protected. Therefore, in order to invoke the web service request, we once again need to supply credentials (Figure 5-5 on page 120).

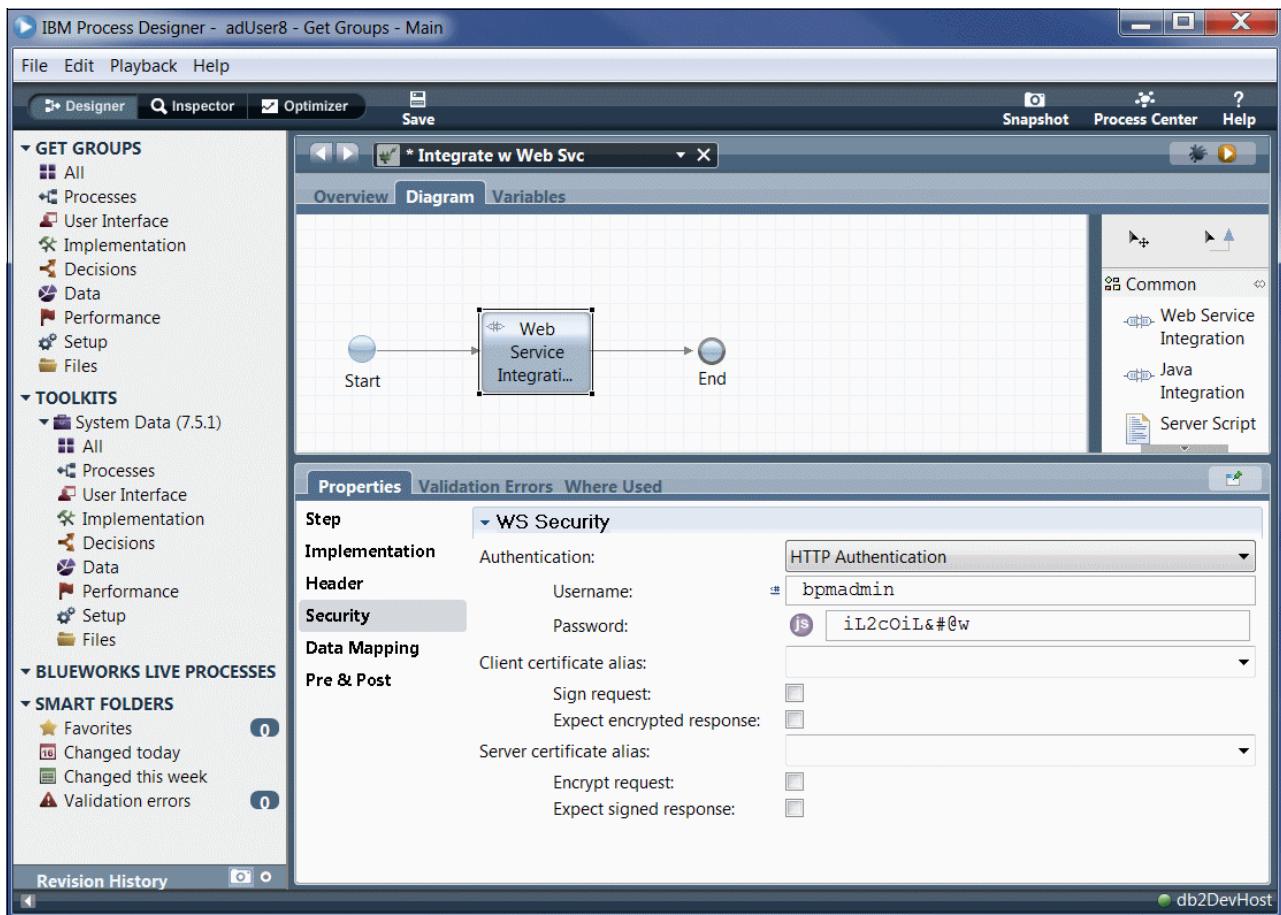


Figure 5-5 Authentication credentials

Entering the username and password in this way ensures that the process application will be able to access the web services running on the host as specified.

Different credentials for each environment

However, using just one username and password combination for each host running web services within all of your Business Process Manager environments is not recommended. In fact, in your organization it is most likely not permitted. It is far more likely that each environment will have a different user ID and/or password. One can understand why an organization would not want developers to have access to web services running in production.

Therefore, we need a way to specify different usernames and passwords for each environment, and probably different URLs too (a production system will not share the same host and port with a test system).

Business Process Manager includes a feature called *environment variables*, which are useful for this. Each of the parameters we specified when discovering the WSDL is a potential candidate for Business Process Manager environment variables (Table 5-1).

Table 5-1 Environment variable candidates

Parameter	Description
WSDL URI	At a minimum, the host name will be different.
User name	The user name can be shared across hosts.
Password	The password <i>should not</i> be shared.

The environment variables are found in the process application's settings panel under the Environment tab (Figure 5-6).

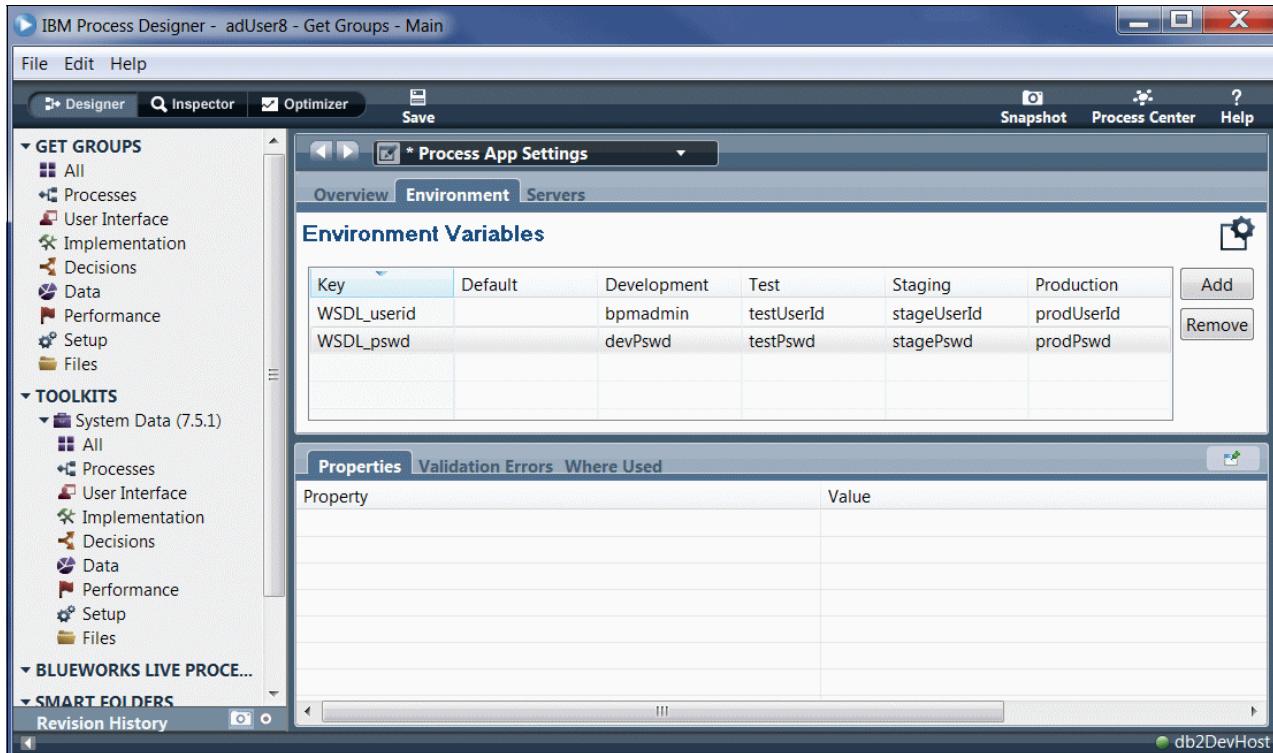


Figure 5-6 Environment tab

The Web Services Integration component would then be updated (Figure 5-7 on page 122).

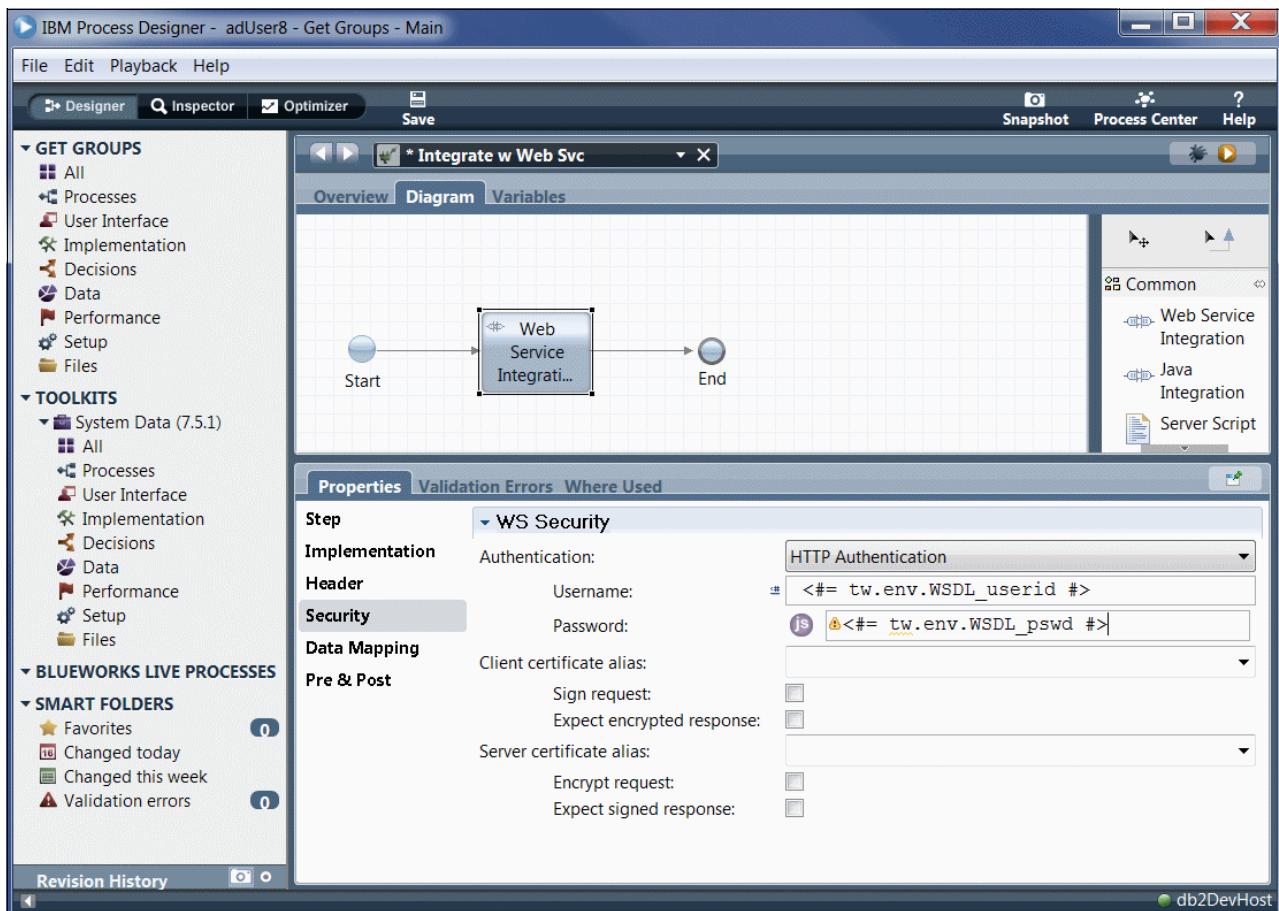


Figure 5-7 Web Services Integration component updated

Notice how the hard-coded values are now replaced with JavaScript expressions. The `tw` is a legacy naming convention due to the previous name of Business Process Manager (Teamworks). The `env` specifies that we want to access the environment name space. The `WSDL_userid` and `WSDL_pswd` are the names of the environment variables that were specified in the far left column (labeled `Key`) of the process applications settings panel.

Security considerations using the Web Service Integrations

Consider the following security considerations for Web Service Integration widgets:

- ▶ Unencrypted environment variables
- ▶ HTTP Basic Authentication
- ▶ Other security options

Unencrypted environment variables

There is a problem in the preceding section. The environment variables are all visible to anyone who has *even just read-only access* to the process application. As you will recall, that is each and every business process author, developer and project manager who is associated with this process application.

In order to overcome this situation, we suggest that you encrypt the values of the passwords *before you enter them* into the process application's settings panel. This needs to be done outside of the product, unless you have already gone through the process of creating an encryption process for use in Business Process Manager. For the purposes of this book, we

are assuming that you have not created an encryption process for use in Business Process Manager.

The problem persists, however. Although the passwords are encrypted, they are not directly usable. The server that is hosting the web services is unlikely to know which encryption scheme you used to create the passwords, and so we need a way to decrypt the passwords before the credentials are put into action.

The following discussion is one possible workaround to the above problem. There are countless others, and certainly ones which are more secure than what we present, but we discuss this topic using a very simple encryption algorithm in order to illustrate the technique.

The only changes that need to be made to the process application are:

1. Encrypt web service passwords externally, and update the settings page (Figure 5-8).

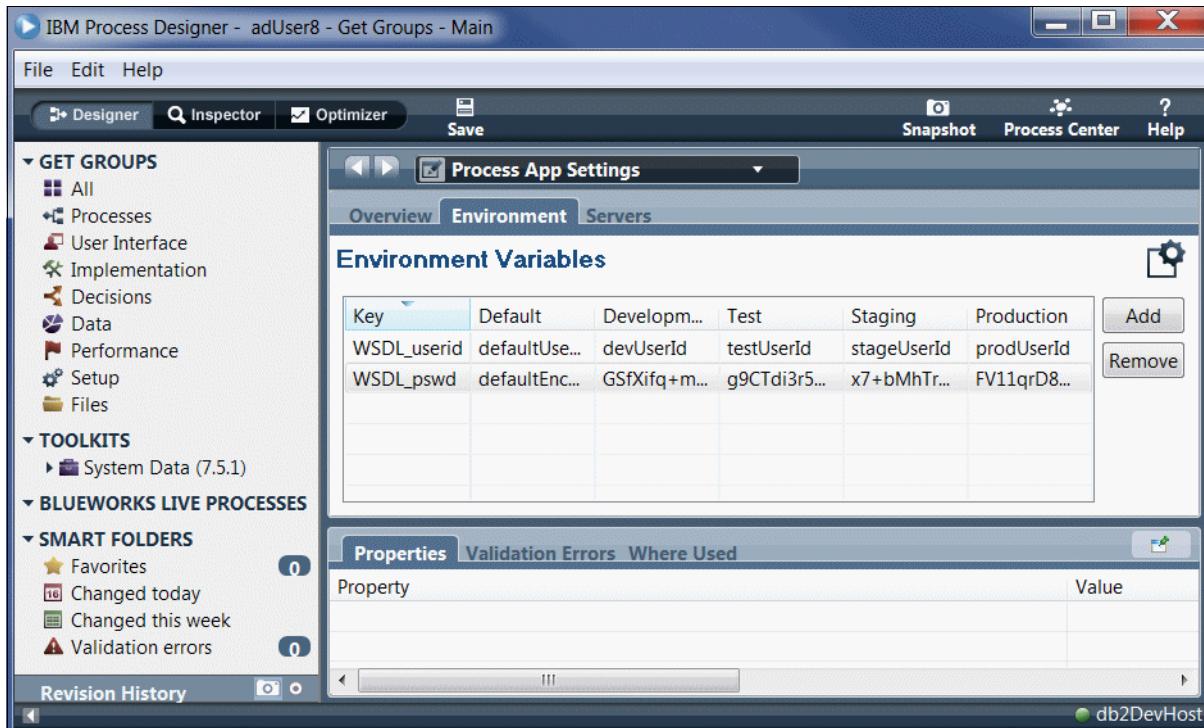


Figure 5-8 Update settings page

2. Create a JavaScript file that defines a decryption function (Figure 5-9 on page 124).

```

function decrypt(input) {
    var cipher = Packages.com.custom.DecryptPassword;
    return cipher.decrypt(input);
}

```

Figure 5-9 JavaScript file

3. Add the JavaScript file to the process application as a Server File (Figure 5-10).

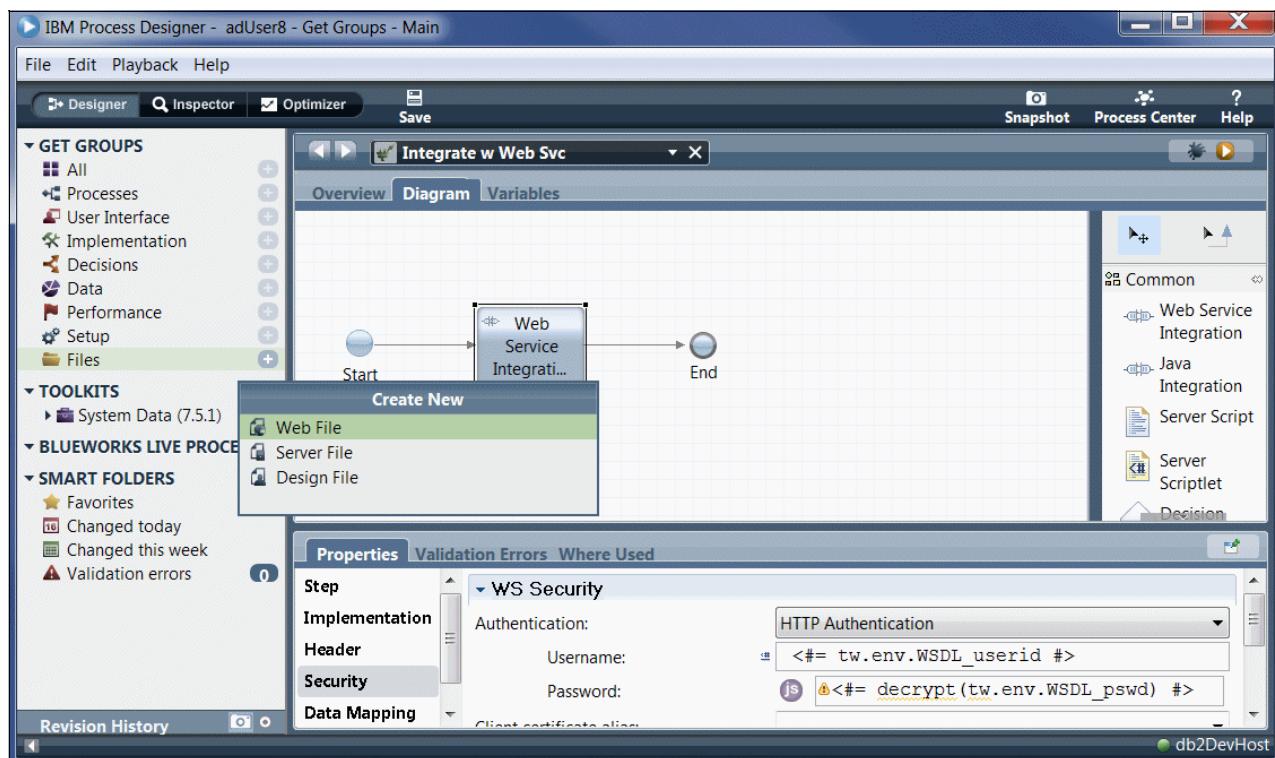


Figure 5-10 Adding a Server File

4. Include a call to the decrypt function (Figure 5-11 on page 125).

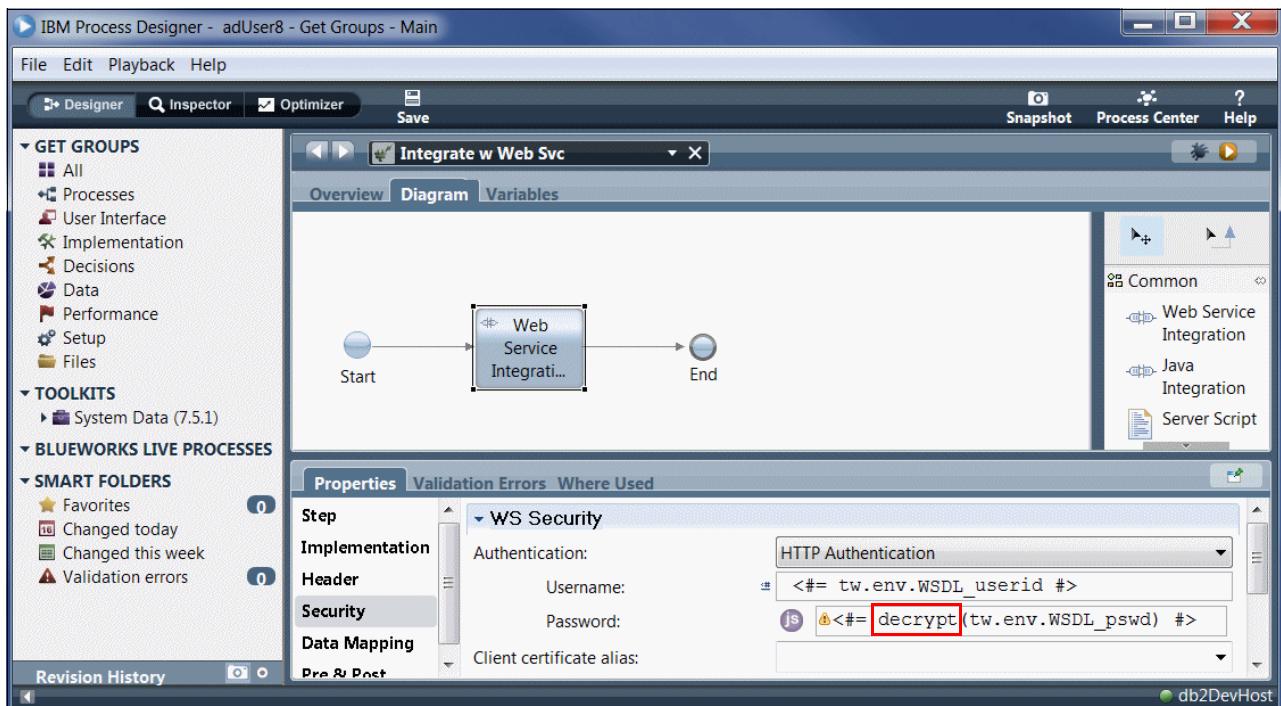


Figure 5-11 Call the decrypt function

With these changes, Business Process Manager (at runtime) will now execute the JavaScript to retrieve both the username and password from the environment variables (as is appropriate for the environment in which the process application is running), and in addition, will pass the password encrypted value into the `decrypt()` function. This function was defined to return a decrypted value, and this result is what is ultimately sent to the web service as the authentication password.

Notice, however, that this scheme is only as secure as the access to the `cipher.js` JavaScript file which contains the `cipher` function. If a developer can gain access to this file, then it is a simple matter to invoke the function with any one of these (visible) environment variables.

Clearly, protecting access to the function is outside of the scope of this book, but it is worth noting here nonetheless. For an even more security conscious solution, you would need to come up with an encryption key management concept too.

HTTP Basic Authentication

Despite our efforts to encrypt the environment variables, author a `decrypt()` function and get all of this installed, this solution must *still* be considered insecure. This is due to the fact that after the Business Process Manager application decrypts the password, it is then sent over the network via HTTP Basic Authentication.

HTTP Basic Authentication is a specification for how web browsers provide a username and password to the server when making a request. Although the most common use is for web browsers, other client applications can use the technique too, as is the case for Business Process Manager calling a remote web service. The specification itself is fairly simple: for a username of `Alladin` and password `open sesame`, the two are separated by a colon (`:`) and concatenated as `Alladin:open sesame`. The result is then Base64 encoded, which renders the combination unreadable to human eyes:

```
QWxhZGRpbjpvcGVuIHNlc2FtZQ==
```

However, there are two problems here. First is that this encoding *is not an encryption scheme*—and therefore cannot be considered secure. Base64 decoders are easily searched online, and within seconds the password can be known.

And second, HTTP Basic Authentication itself *has nothing to do with encryption*. The transport of HTTP Basic Authentication headers over HTTP is as wide open as you can get. SSL is required in order to consider this action secured.

Important: We recommend SSL between the Business Process Manager servers and each and every server that is hosting web services in your environment.

Other security options

In addition to HTTP Basic Authentication, the Business Process Manager Web Service Integration offers the following options for securing access to outbound web services:

- ▶ OASIS Username Token (with password in plain text).)
- ▶ XML Signature (via X.509 certificates)
- ▶ Encryption

These options are specified in the same Sec Web Service Integration properties tab (Figure 5-12).

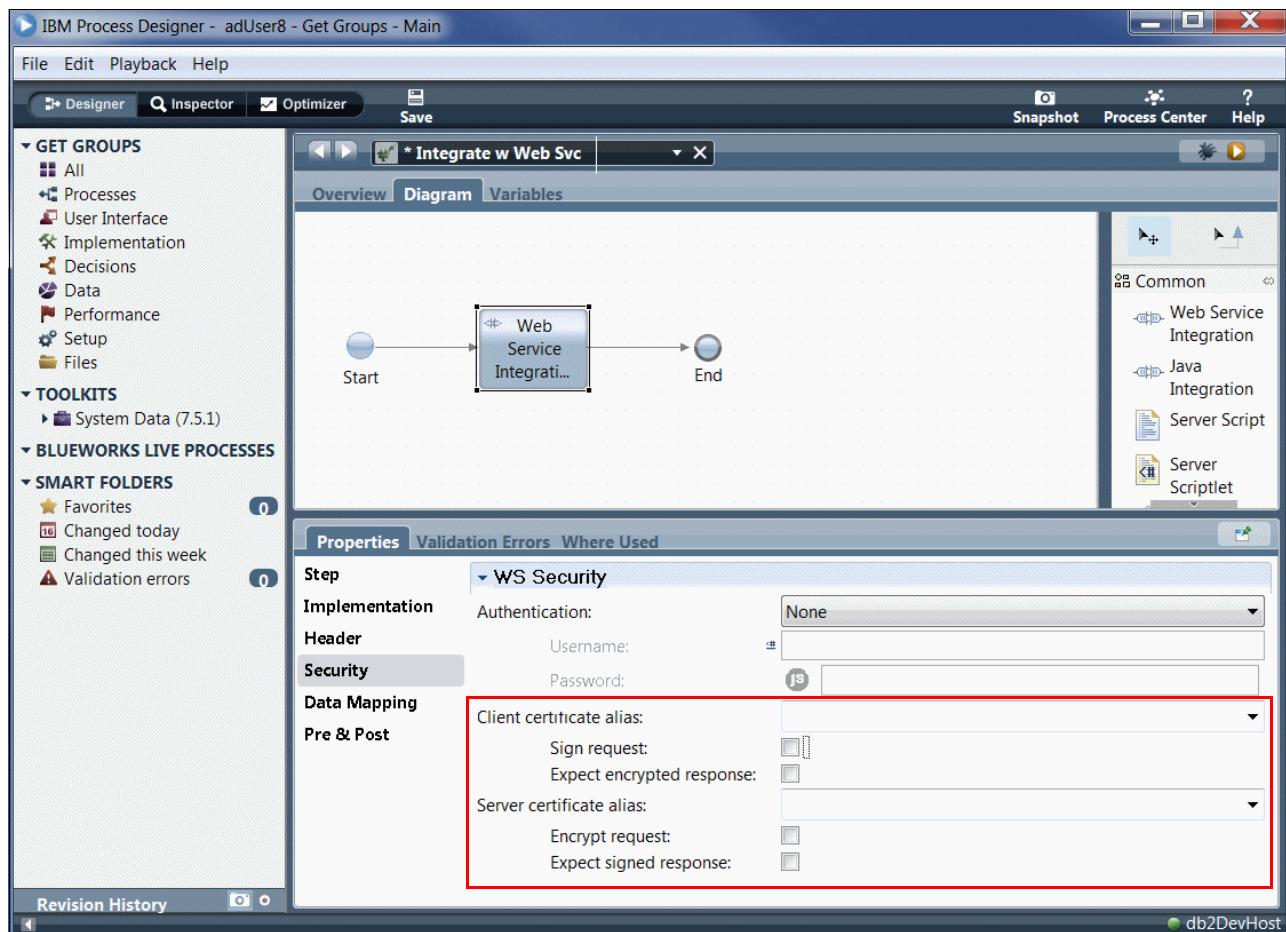


Figure 5-12 Security section

5.2.2 Using SOAP Integration

The Business Process Manager Web Services Integration component cannot be used universally. There are occasions when the WSDL is not well formed, occasions when you want to pass different SOAP headers, and even some occasions when it is just too complex for the underlying software to parse. Under these circumstances, you may find that the Web Services Integration component cannot be used. If you run into problems with that technique, then Business Process Manager Standard Edition offers a second, more tolerant but more generic and therefore more manual option: the SOAP Integration toolkit implementation.

The SOAP Integration implementation is found in the Implementations group of the System Data Toolkit.

The basic idea behind a SOAP Integration is that you begin with a web services browsing tool (such as SoapUI) and formulate a request that successfully invokes the web service. The data from the SoapUI request is then copied into the properties section of the SOAP Integration.

Next, as before, create a new Integration Service. Expand the System Data Toolkit and find under the Implementation group an Integration Service called Call WebService via SOAP, and drag this into the BPD diagram (Figure 5-13).

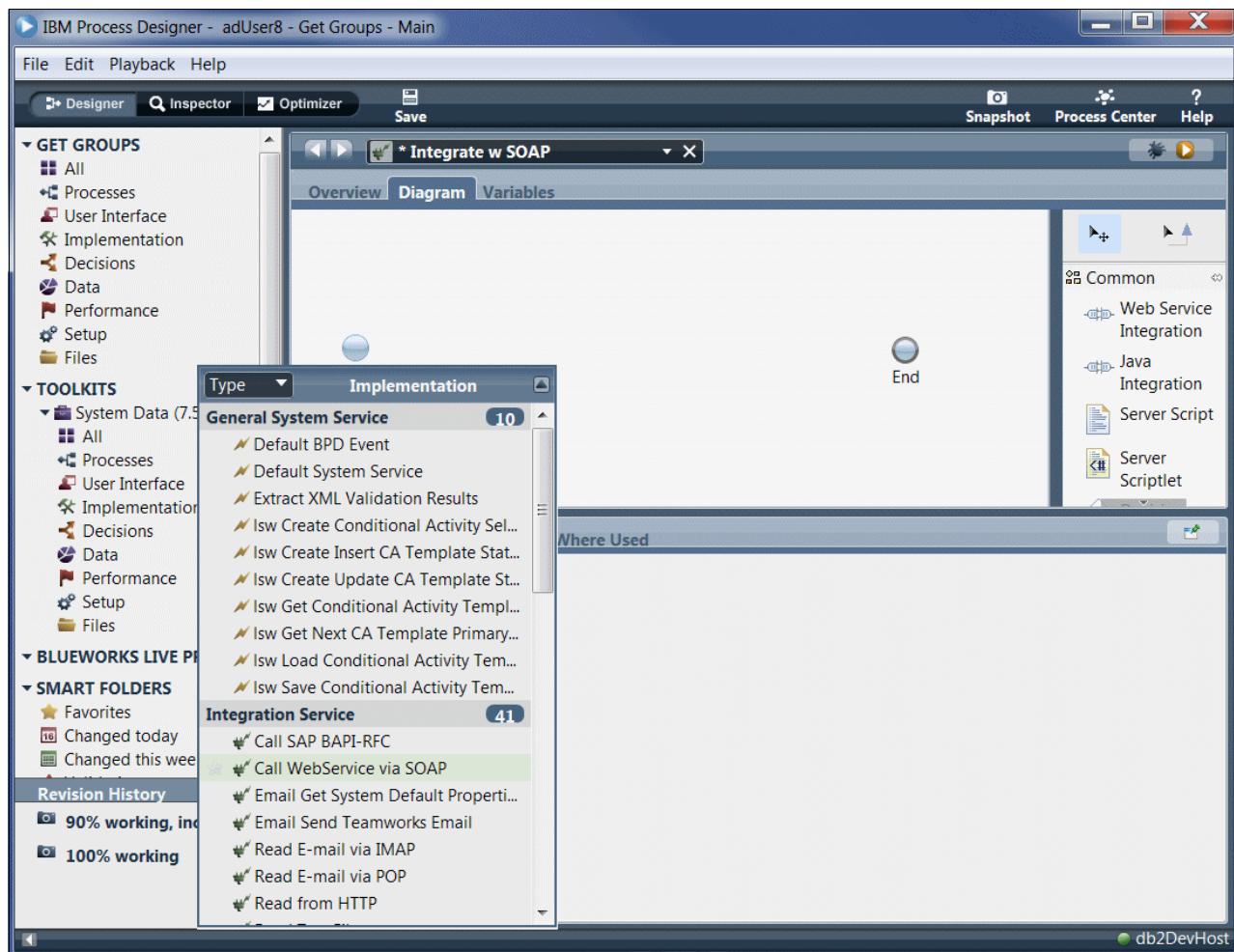


Figure 5-13 New Integration Service

You will also need a *Server Scriptlet* to contain the XML of the web service request that you created for use in SoapUI, and two private variables to specify the XML request and response XML elements (Figure 5-14).

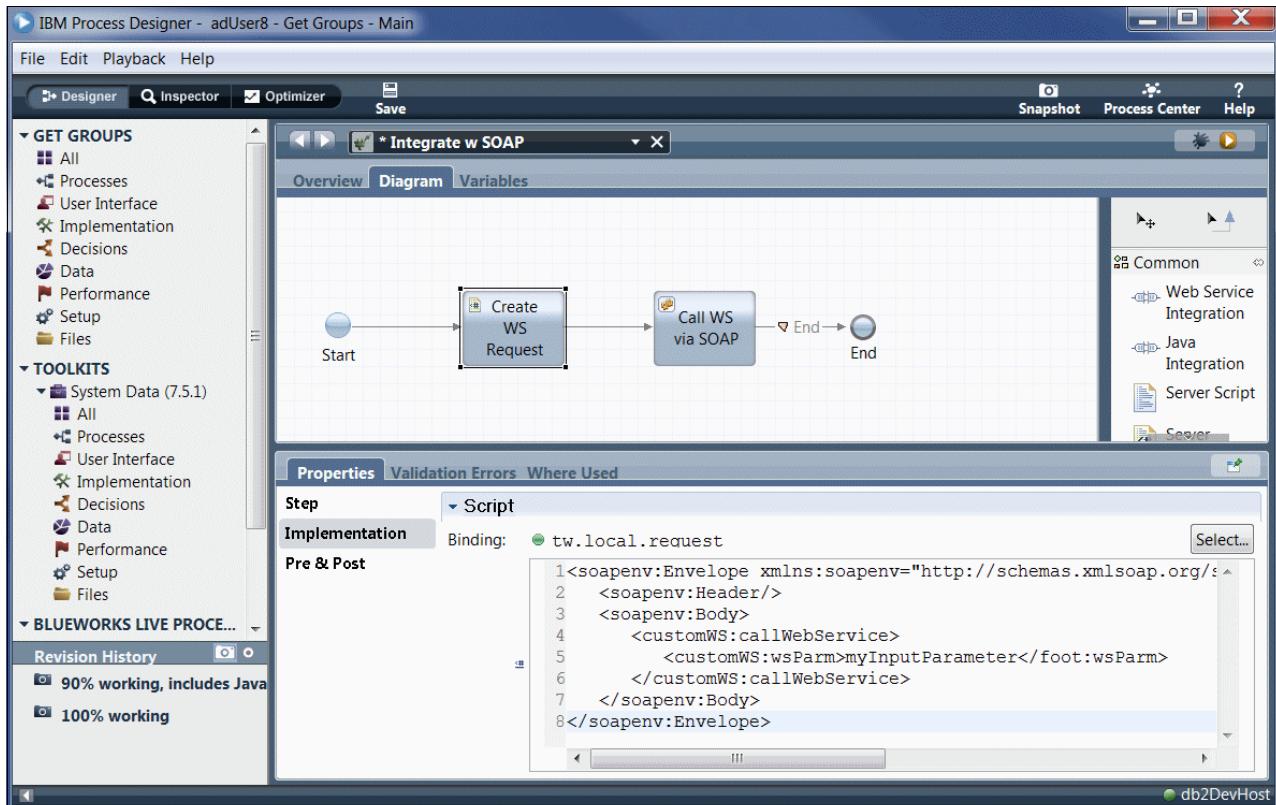


Figure 5-14 Server Scriptlet

You need to read through the WSDL to discover the values that will be used in the SOAP Implementation's data mapping section, as well as map the web service's output to the response XMLElement you created (Figure 5-15 on page 129).

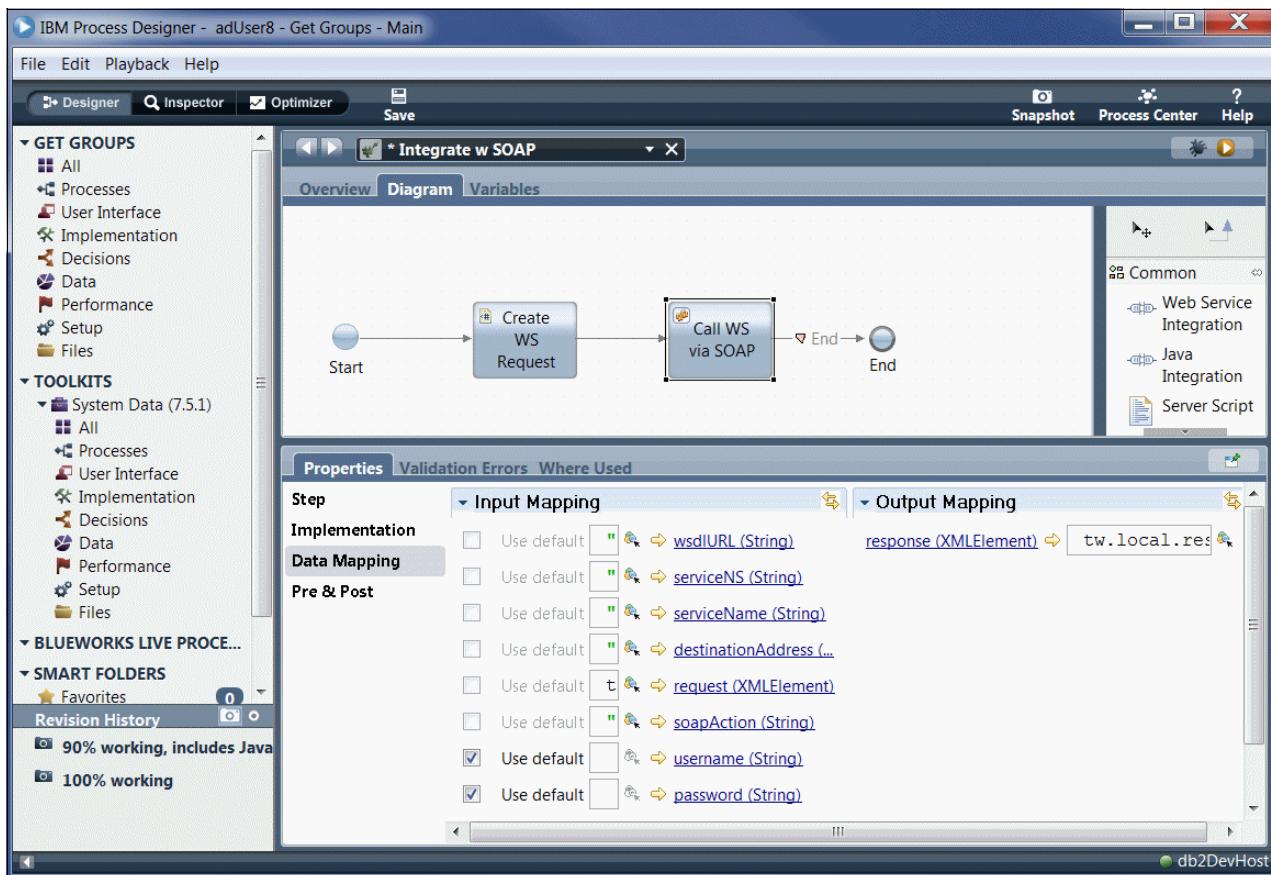


Figure 5-15 Data Mapping

If the WSDL and web service are protected, there is an input mapping for username and password as shown in Figure 5-15. You can use a portion of the previous technique and use JavaScript to decrypt the password (and even passwords that are specific to each environment).

To expand upon the range of security options available, you can move beyond the JavaScript technique and move into a full-blown Java implementation.

5.2.3 Using Java Integration

The basic idea behind the Java Integration component is that you require access to methods defined in a Java object in order to complete a business process definition (BPD) step.

In our previous two examples, we focused exclusively on accessing a web service. The Java Integration component is more versatile, allowing you to access any Java object from within a BPD, feeding it input from the BPD and extracting from it its outputs. This technique could be used to access virtually any Java-based security mechanism in order to feed passwords (or any other data, for that matter) to your web services. You could even include a Java class that was designed to communicate directly with web services, bypassing both of the previous two techniques.

The first step is, of course, to write the Java. Once it is ready to deploy, package your classes in a .jar file, and add them to the business process application as a *Server File*, just as we did with the encrypt.js file in the previous example (Figure 5-16 on page 130).

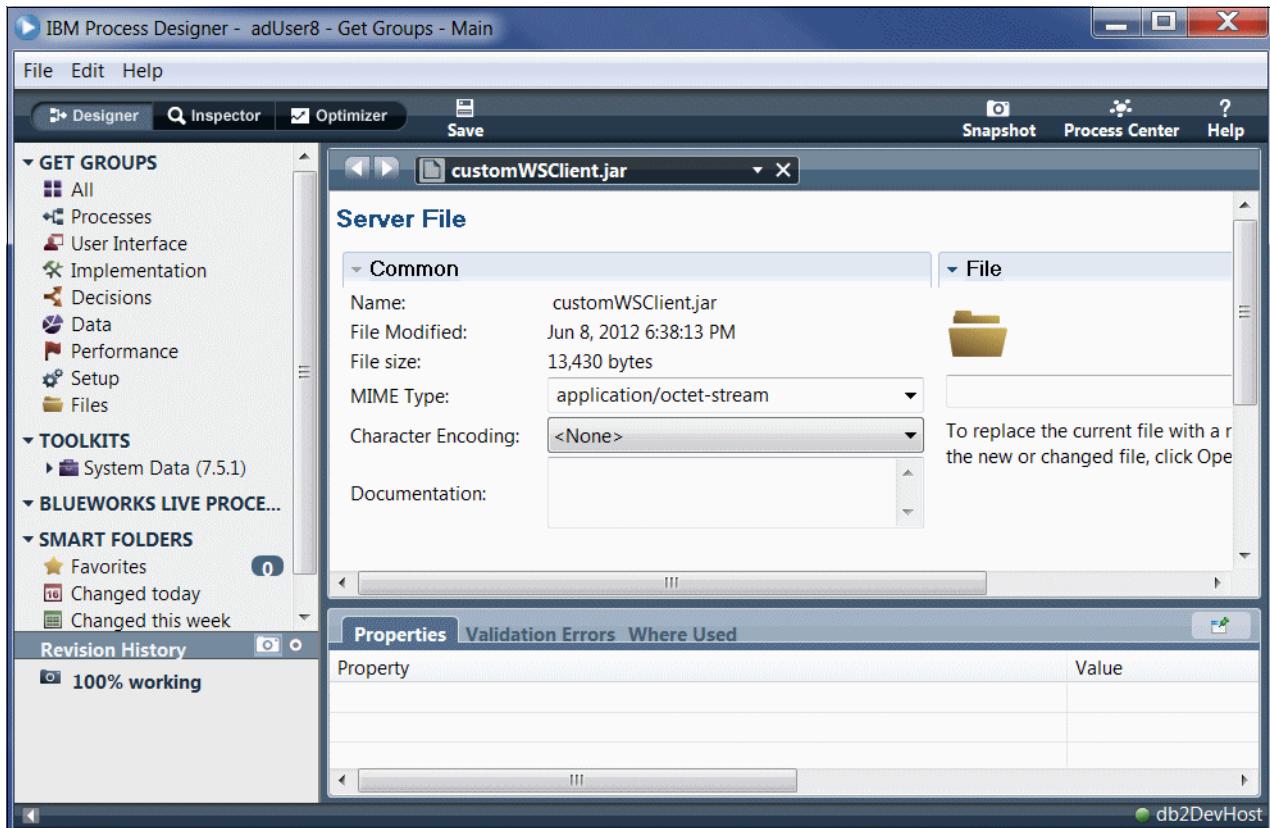


Figure 5-16 New Server File

You then create a new Integration Service (as we did before in preparation to work with the Web Service Integration) but this time drag in a Java Integration component (Figure 5-17 on page 131).

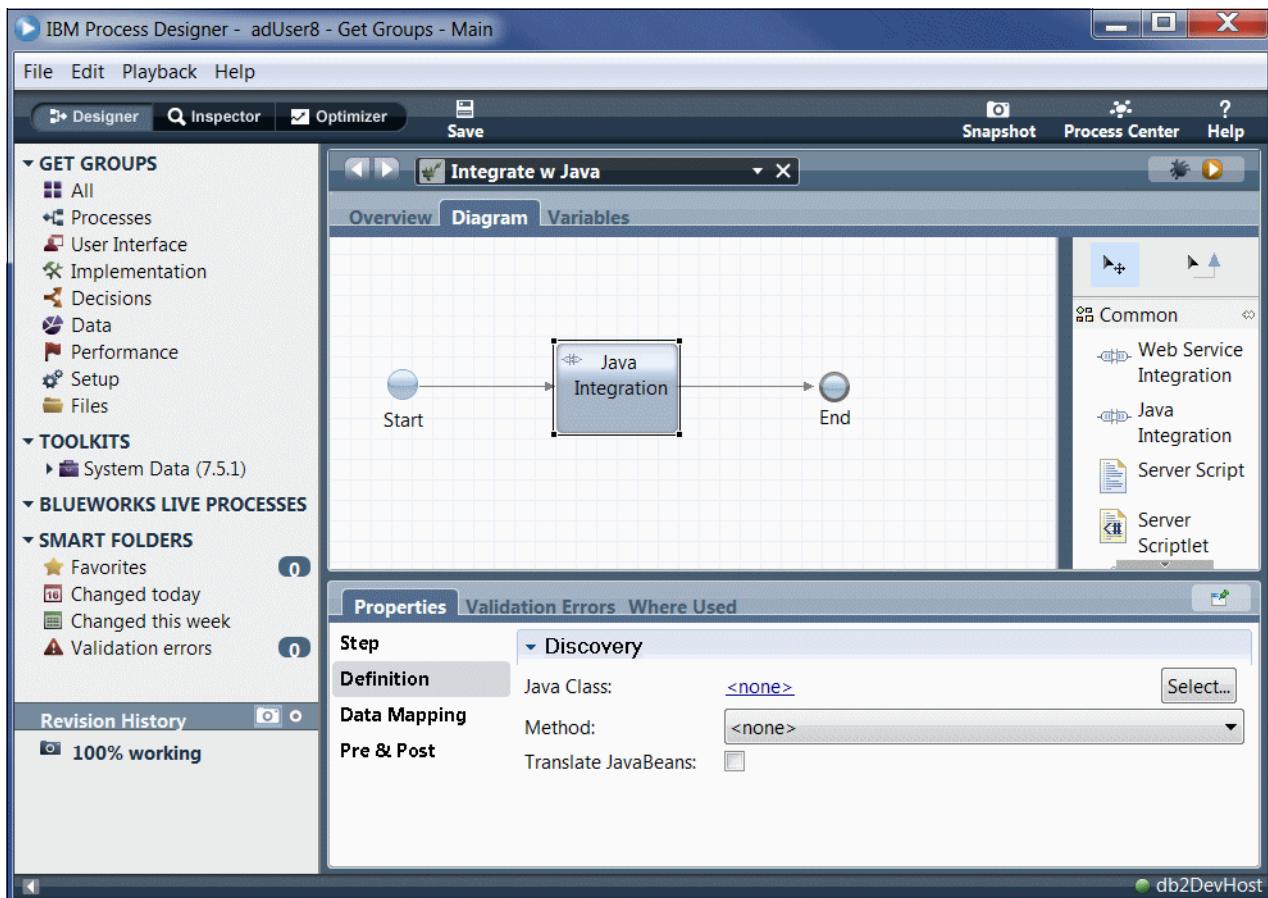


Figure 5-17 Java Integration component

Just as with the Web Services Integration, we have a Discovery section (Figure 5-18 on page 132). Click **Select**.

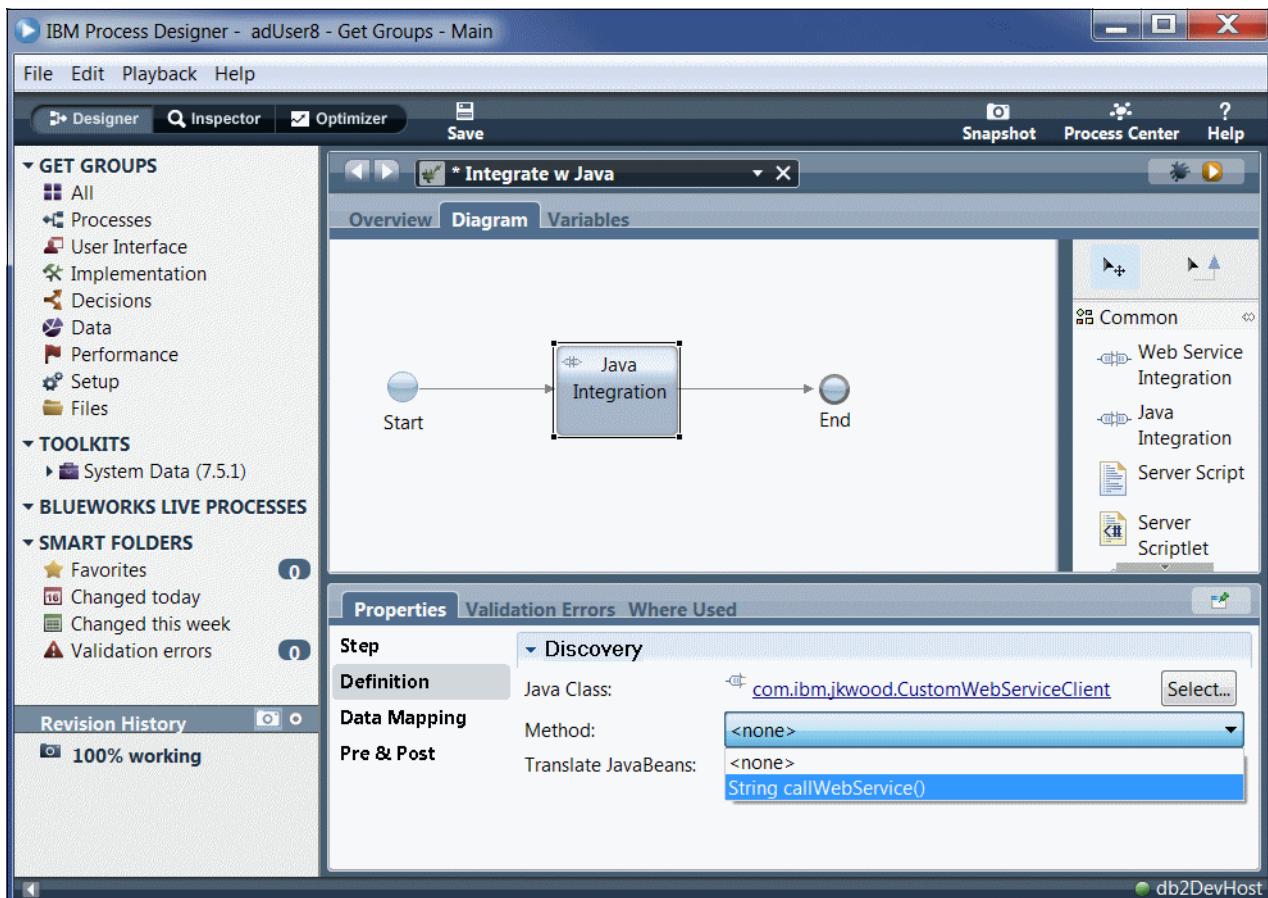


Figure 5-18 Discovery section

You can see that we have selected the class (`CustomWebServiceClient`) and are in the process of selecting the one method defined therein (`callWebService`).

We could just as easily have defined a Java class called `CustomEnvironmentPasswordCip` defined therein called `decrypt`—which could have been used in either of the first two examples to provide password encryption.

That is all there is to specifying the use of a plain old Java object for use in a Business Process Manager process application. As you can see, there are no Business Process Manager application-defined mechanisms for specifying, supporting, or enforcing security. All of this will be the responsibility of the software developer.

5.3 Business Process Manager Standard Edition inbound web services

The security considerations when developing a Business Process Manager Standard Edition inbound web service are even simpler than the case for outbound services. You have only one option with Business Process Manager Standard Edition: HTTP Basic Authentication.

This is due to the fact that Business Process Manager Standard Edition is based upon the Apache Axis 1.3 (Axis 1.2 for earlier versions, including WebSphere Lombardi Edition). There is an acknowledgement that this web services framework needs to be replaced with a more

flexible mechanism, and we have already seen the beginnings of this movement with Business Process Manager V8.0, in that V8 has replaced the *outbound* web services stack with the newer, more powerful JAX-WS 2.2.

Unfortunately, the steps to create an inbound web service are not as simple as your options for securing them. Several components need to be in place in order to be able to publish the web service.

5.3.1 Steps to create inbound web service

Several steps are necessary to expose a Business Process Manager process application as a callable web service:

1. Create a business process definition (BPD) that includes an Incoming Message Event.
2. Create a General System Service which will hold any required input variables (these will be mapped to web service parameters).
3. Create an Under Cover Agent to hold the General System Service, and schedule this agent as type On Event, and configure the BPD's incoming message event to trigger the under cover agent.
4. Create an Integration Service to invoke the under cover agent.
5. Create a Web Service implementation that will also define the WSDL for this web service.

Create the BPD

The first step is to add an *Incoming Message Event* to a business process diagram (BPD). We create one from scratch (Figure 5-19).

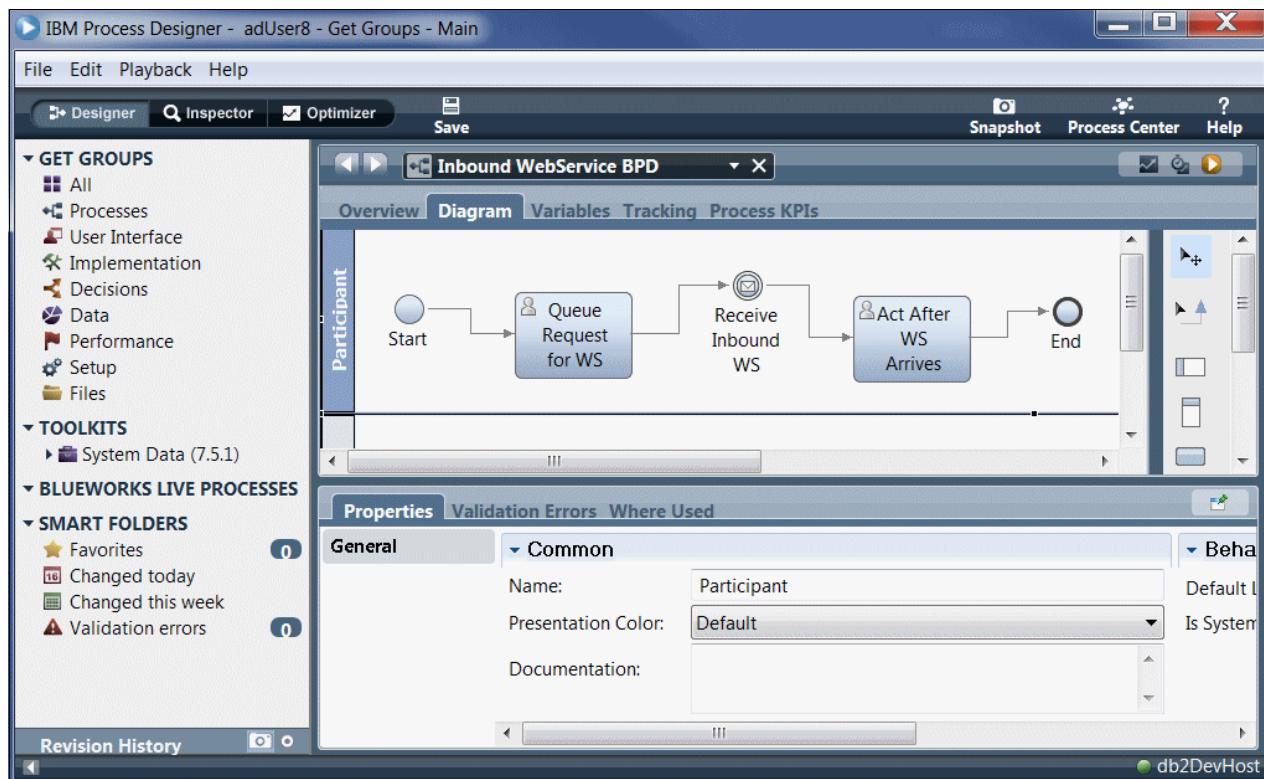


Figure 5-19 Incoming Message Event

Here you can see that our example BPD has two steps:

- ▶ Queue Request for WS
- ▶ Act After WS Arrives

The basic idea for our example is that we have a BPD which requires some setup (accomplished in the Queue Request for WS step), and then at some later point an external web service will invoke the remainder of this BPD in order to perform the Act After WS Arrives step.

Create General System Service

We next need to create the General System Service which will hold a few variables. Since our web service is invoked from within the flow of the BPD, at a minimum it will need to hold our in-flight BPD instanceId or any other correlation data, so that the incoming web service will know which BPD it should be invoking (Figure 5-20).

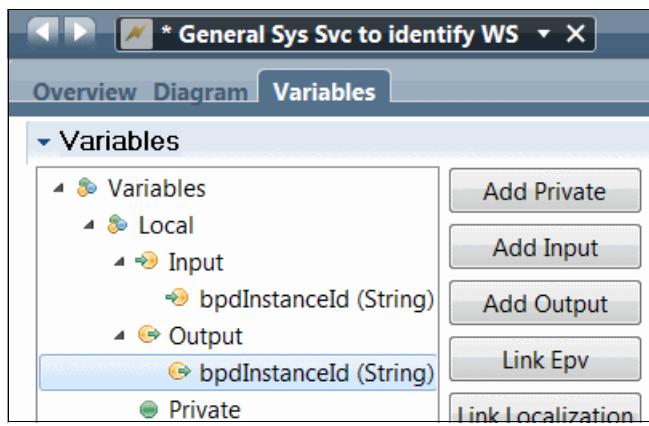


Figure 5-20 Variables view

Create the UCA

1. The Under Cover Agent (UCA) is mapped to the General System Service in order to facilitate the continuity of variables, and it serves as an entry point to the BPD when the web service is called (Figure 5-21 on page 135).

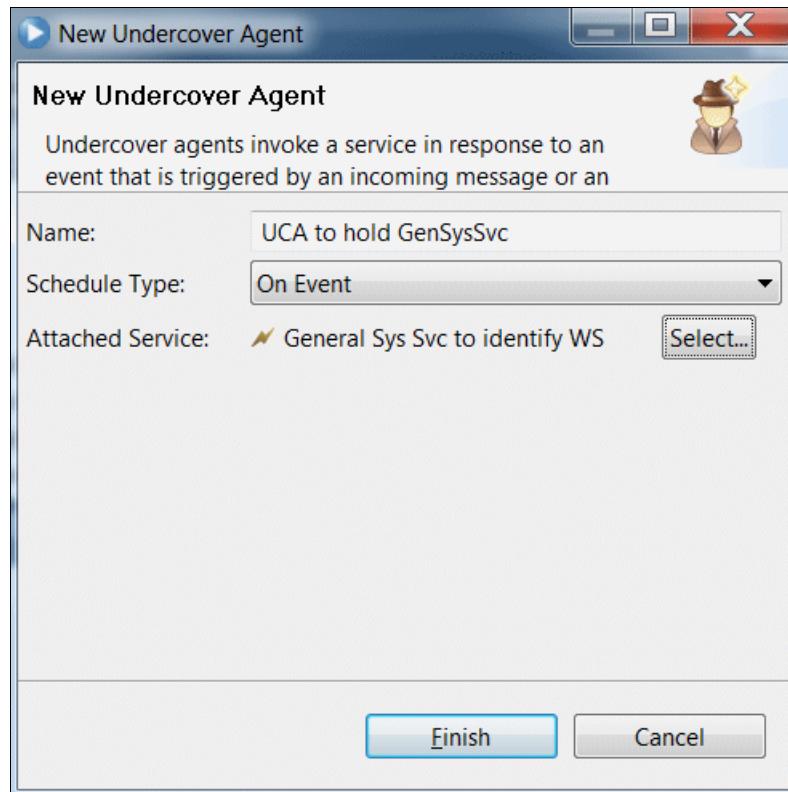


Figure 5-21 New Undercover Agent

2. We need to map the parameters (in our example, we are just concerned with the in-flight BPD instanceId). See Figure 5-22 on page 136.

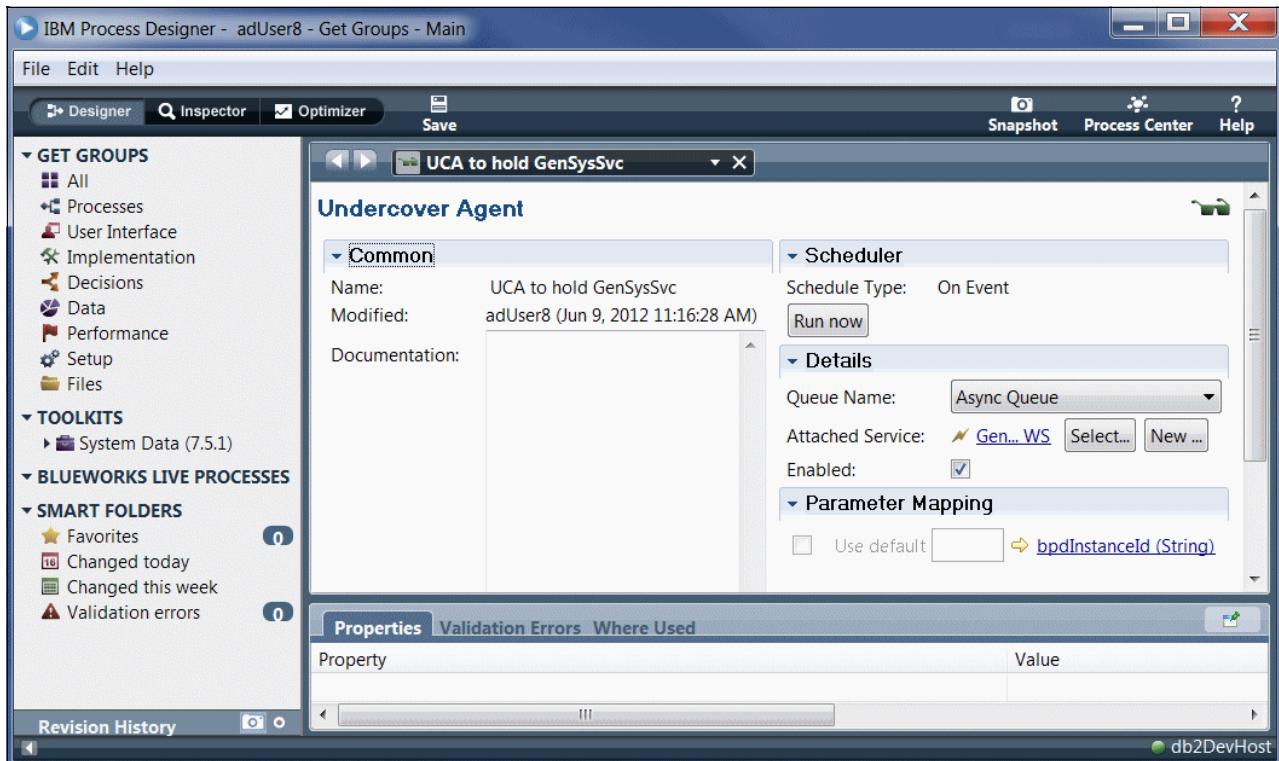


Figure 5-22 Map parameters

3. We also need to configure the BPD to be triggered by the UCA (Figure 5-23 on page 137).

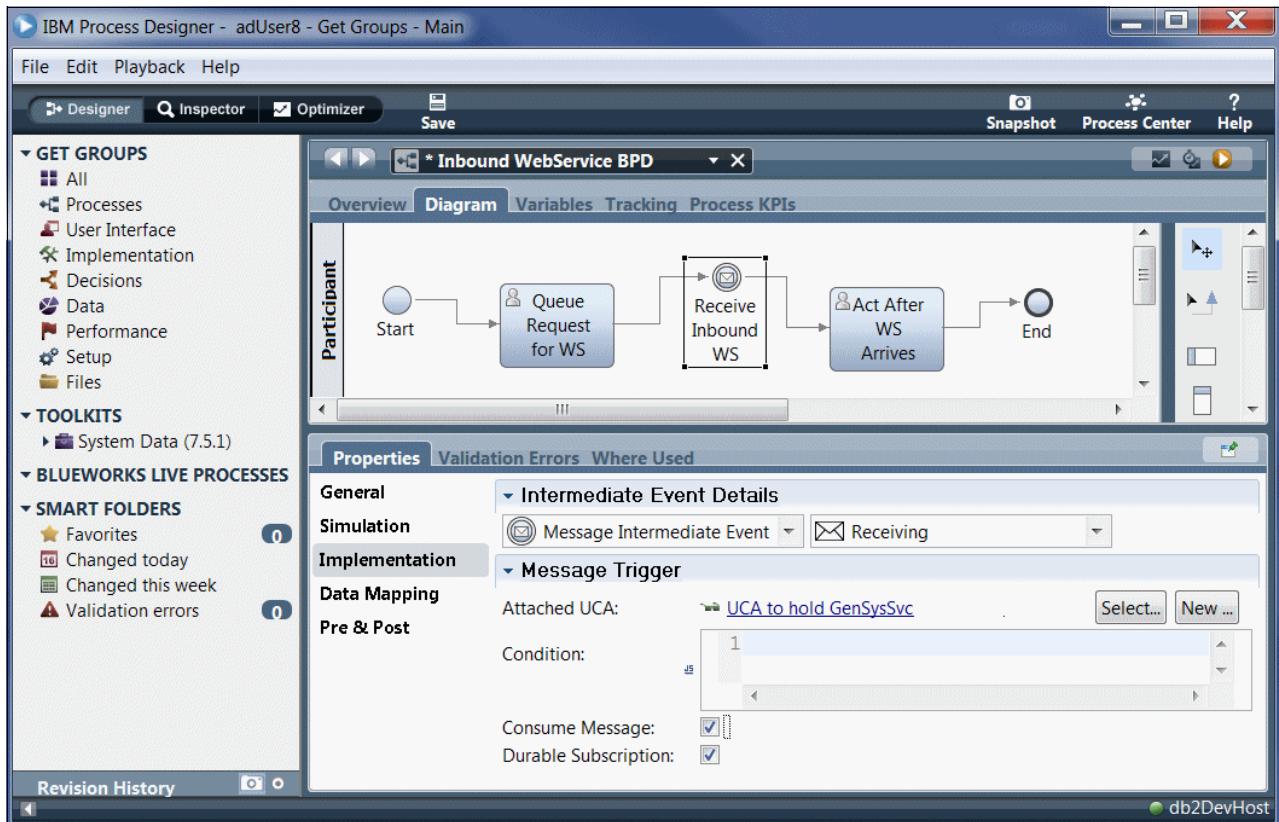


Figure 5-23 Message Trigger

4. And finally we need to map the in-flight BPD instanceId to the General System Service's variable (Figure 5-24 on page 138).

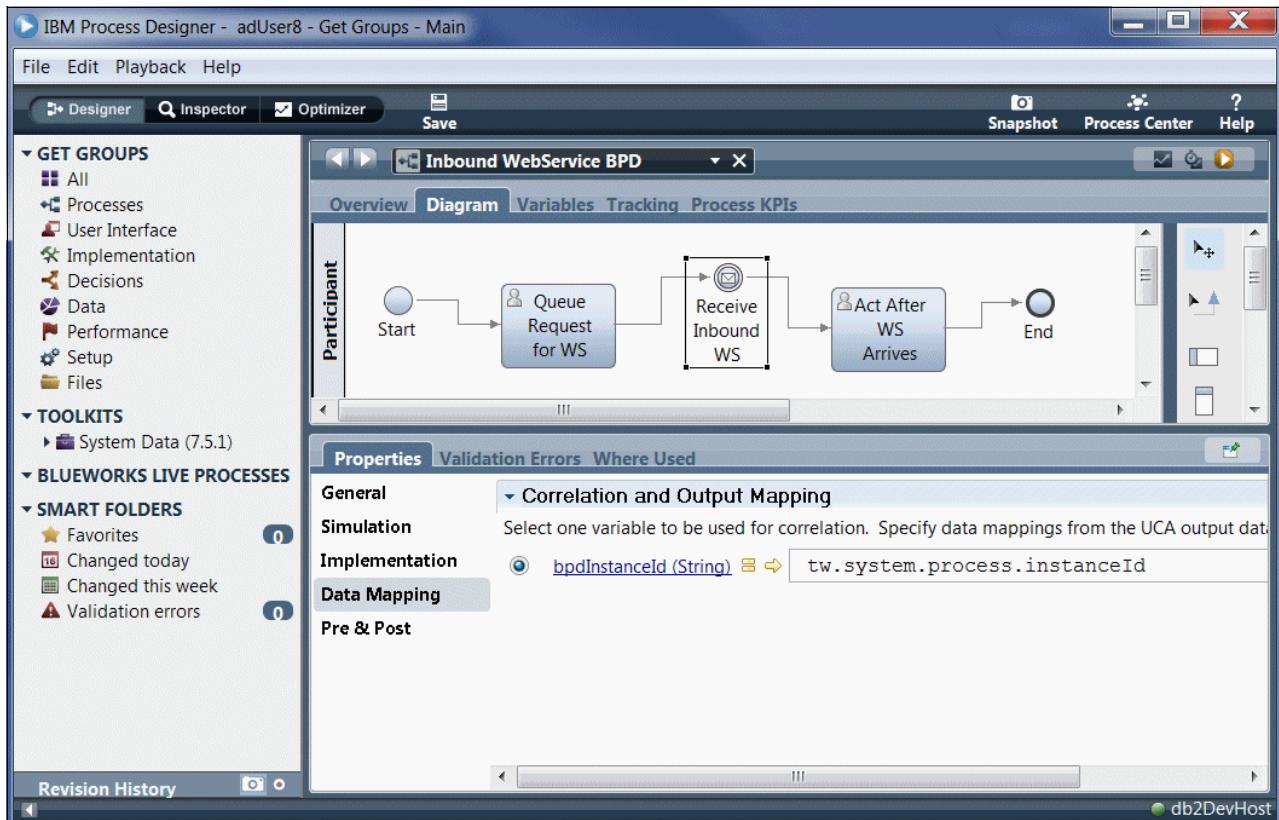


Figure 5-24 Correlation and Output Mapping

Create an Integration Service

1. The Web Service Integration will be used to invoke the UCA from the Apache Axis web services software stack (Figure 5-25 on page 139).

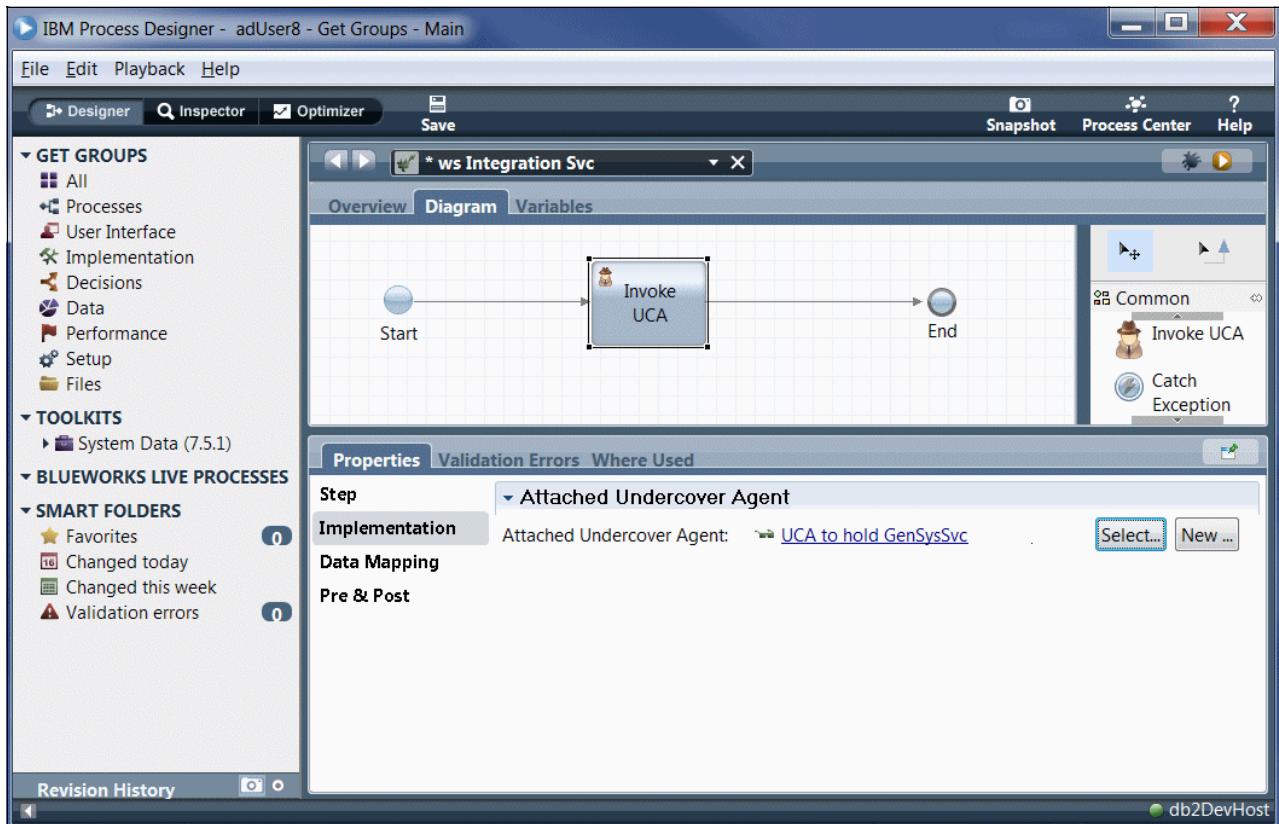


Figure 5-25 Integration Service

2. We need to create an input variable that will map to the UCA's variable (Figure 5-26 on page 140).

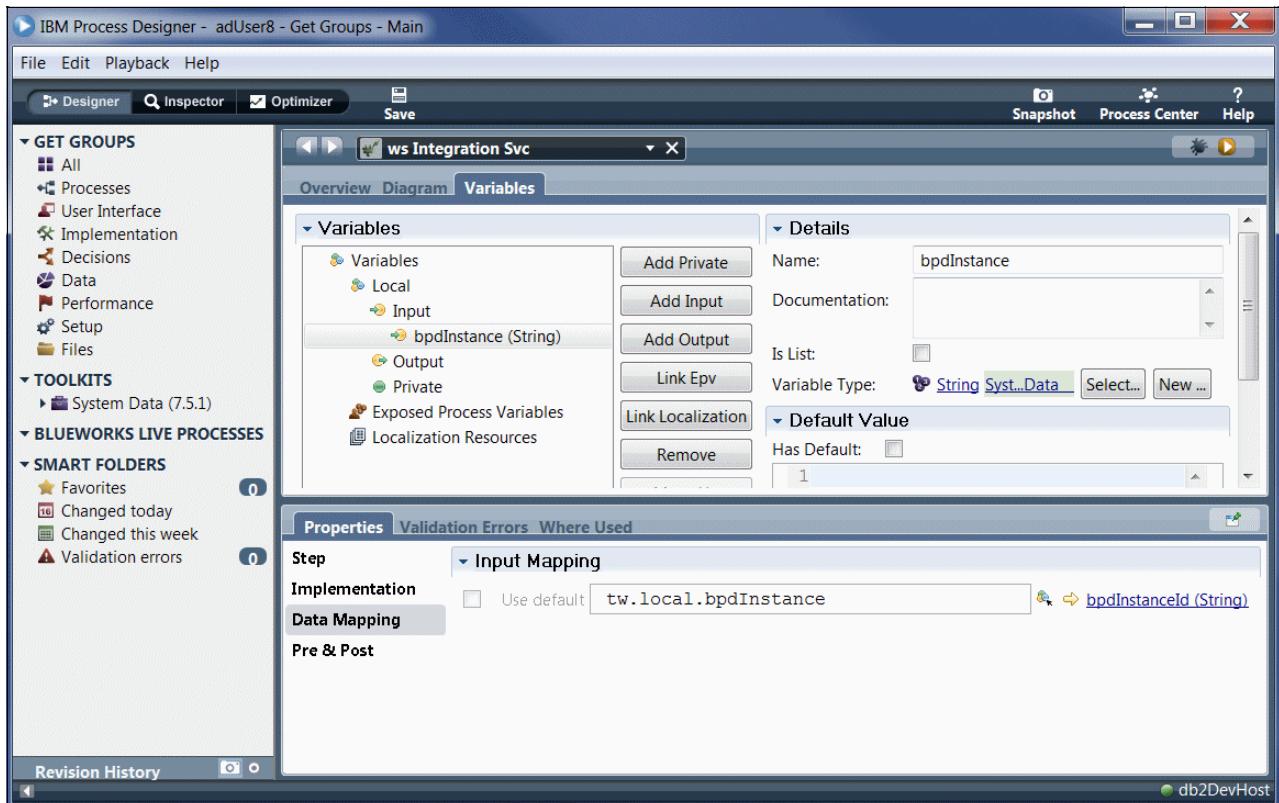


Figure 5-26 New input variable

Create a Web Service Implementation

1. The final step is to create a Web Service Implementation from the Implementations group (Figure 5-27).



Figure 5-27 Create a new web service

2. This brings up the panel shown in Figure 5-28 on page 141.

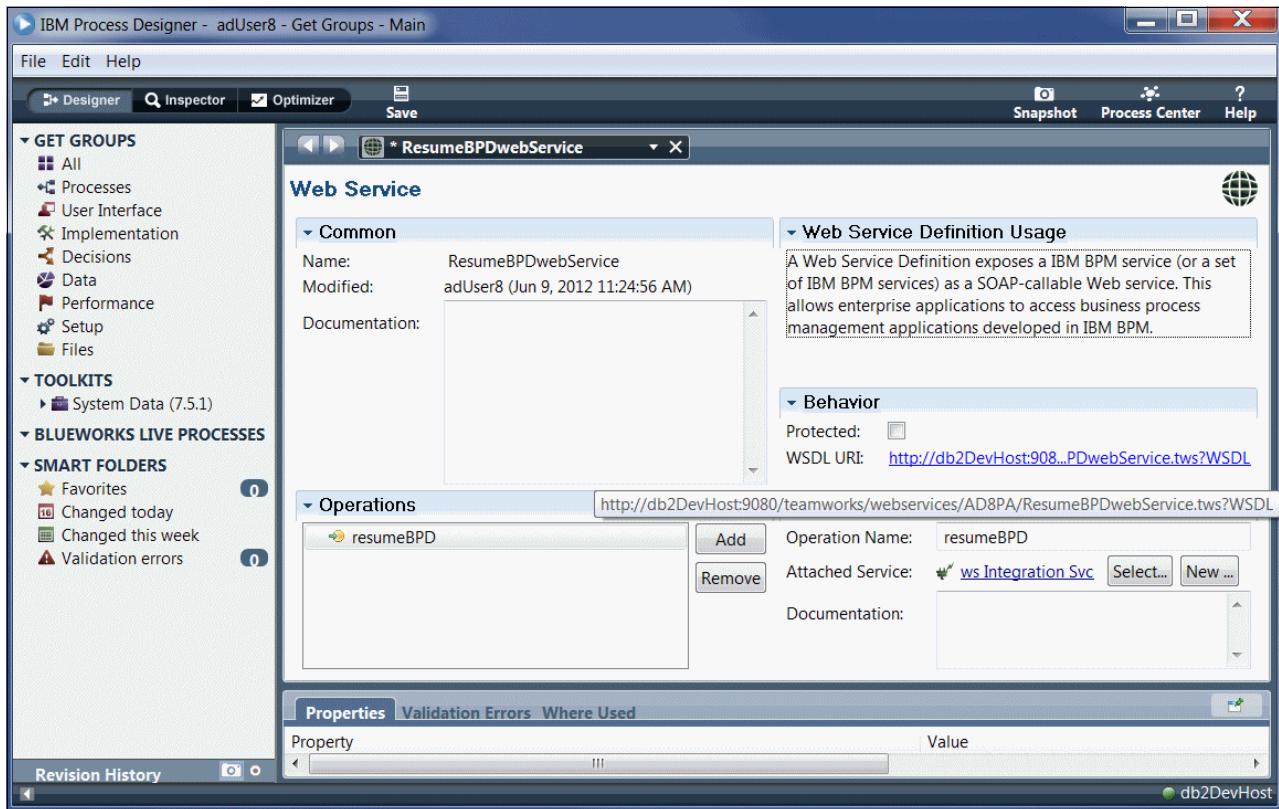
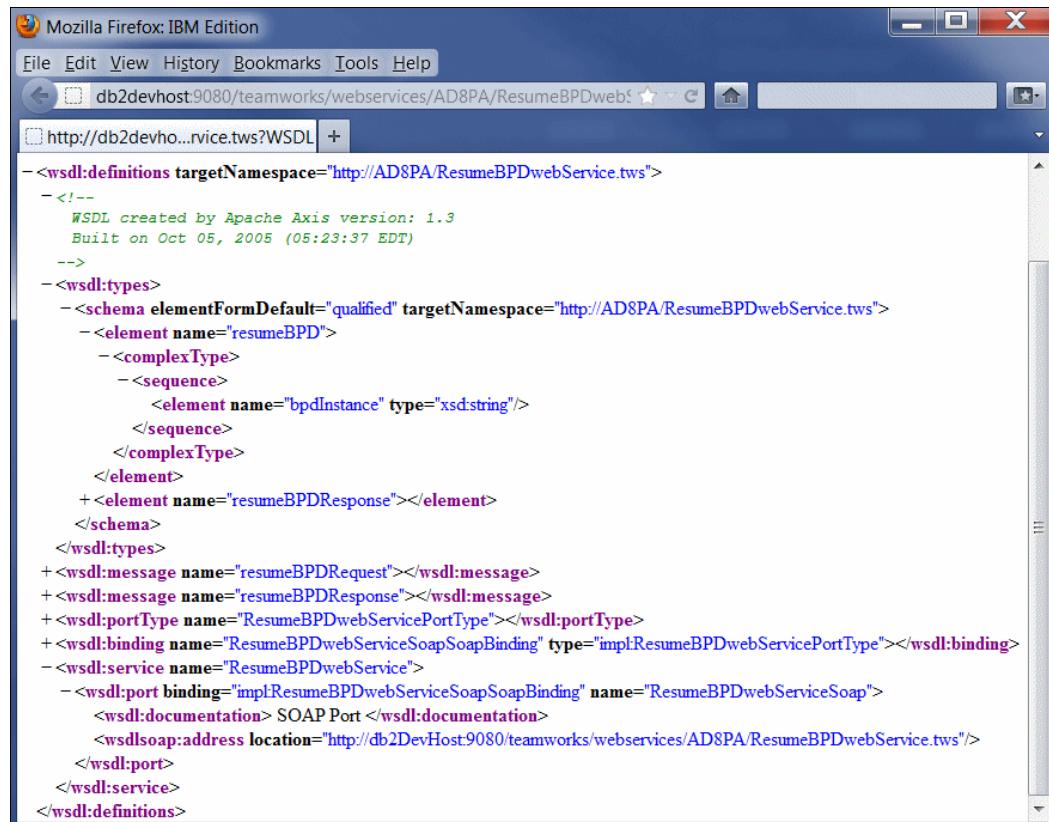


Figure 5-28 Web Service

3. We click **Add**, name the operation resumeBPD, and then just click **Select** and choose the integration service. From these two simple operations, BPM is able to build the WSDL (Figure 5-29 on page 142).



The screenshot shows the Mozilla Firefox browser window with the title "Mozilla Firefox: IBM Edition". The address bar displays "db2devhost:9080/teamworks/webservices/AD8PA/ResumeBPDwebService.tws?WSDL". The main content area shows the WSDL XML code for the service. The XML includes definitions for types, messages, port types, bindings, and a service. It specifies the target namespace as "http://AD8PA/ResumeBPDwebService.tws" and uses Apache Axis version 1.3, built on Oct 05, 2005 (05:23:37 EDT). The code defines a schema for "resumeBPD" with a complex type "bpdInstance" of type "xsd:string". It also defines a message "resumeBPDRRequest" and a response "resumeBPDRResponse". The port type "ResumeBPDwebServicePortType" is defined, along with a binding "ResumeBPDwebServiceSoapBinding" and a service "ResumeBPDwebService". The binding specifies a port "implResumeBPDwebServiceSoapBinding" with a documentation entry for a SOAP port at the specified URL.

```
<?xml version="1.0"?>
<wsdl:definitions targetNamespace="http://AD8PA/ResumeBPDwebService.tws">
    <!--
        WSDL created by Apache Axis version: 1.3
        Built on Oct 05, 2005 (05:23:37 EDT)
    -->
    <wsdl:types>
        <schema elementFormDefault="qualified" targetNamespace="http://AD8PA/ResumeBPDwebService.tws">
            <element name="resumeBPD">
                <complexType>
                    <sequence>
                        <element name="bpdInstance" type="xsd:string"/>
                    </sequence>
                </complexType>
            </element>
            <element name="resumeBPDRResponse"></element>
        </schema>
    </wsdl:types>
    <wsdl:message name="resumeBPDRRequest"></wsdl:message>
    <wsdl:message name="resumeBPDRResponse"></wsdl:message>
    <wsdl:portType name="ResumeBPDwebServicePortType"></wsdl:portType>
    <wsdl:binding name="ResumeBPDwebServiceSoapBinding" type="implResumeBPDwebServicePortType"></wsdl:binding>
    <wsdl:service name="ResumeBPDwebService">
        <wsdl:port binding="implResumeBPDwebServiceSoapBinding" name="ResumeBPDwebServiceSoap">
            <wsdl:documentation> SOAP Port </wsdl:documentation>
            <wsdlsoap:address location="http://db2DevHost:9080/teamworks/webservices/AD8PA/ResumeBPDwebService.tws"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

Figure 5-29 WSDL

5.3.2 Review and summary

The creation of an inbound web service can seem disjointed and can seem to include a lot of moving parts. Let us take a step back and look at the whole process in diagram form (Figure 5-30 on page 143).

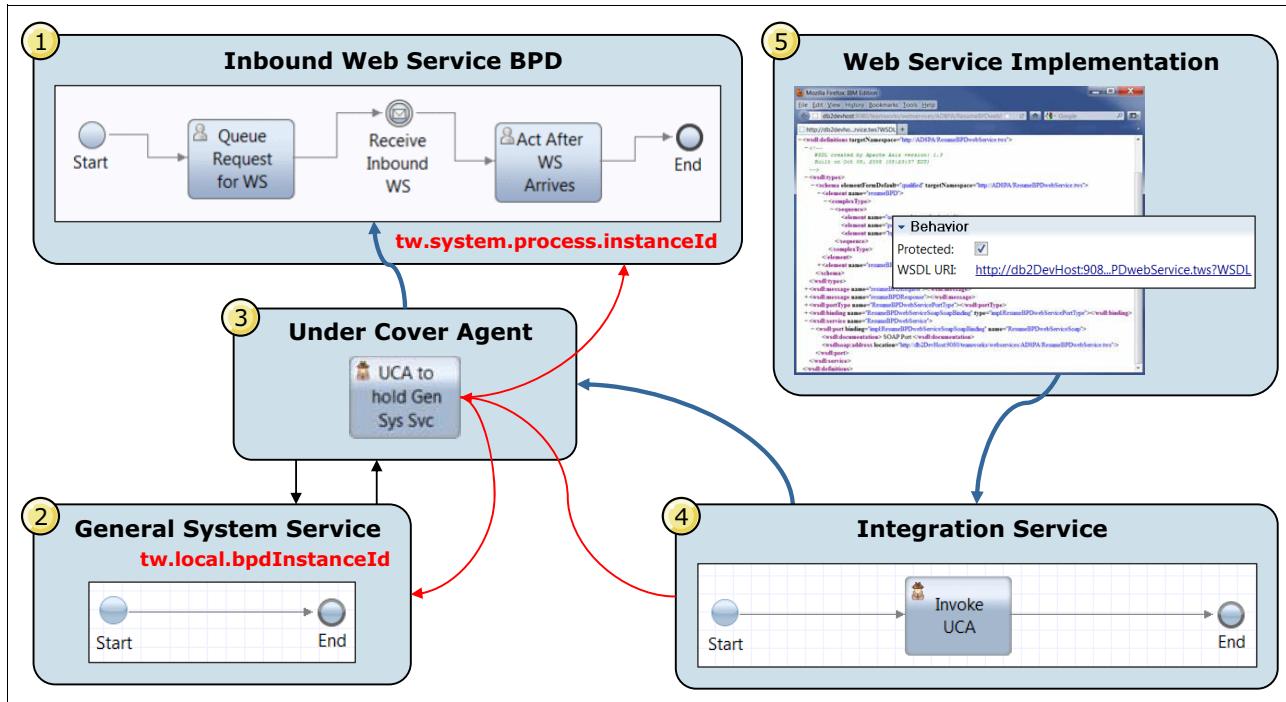


Figure 5-30 Full process

The steps to create run counter-clockwise from 1 to 5. Interestingly, the steps to invoke run exactly the opposite—from 5 to 1. As you can see from Figure 5-30, the Under Cover Agent can be thought of as the piece that holds it all together.

5.3.3 Securing the inbound web service

Compared to building the inbound web service, securing the service is easy. Simply select the **Protected** check box in the web services implementation panel (Figure 5-31).



Figure 5-31 Protected check box

Save the implementation, and the business parameters are added to the WSDL (Figure 5-32 on page 144).

The screenshot shows the Mozilla Firefox browser window with the title "Mozilla Firefox: IBM Edition". The address bar displays the URL "http://db2devhost:9080/teamworks/webservices/AD8PA/ResumeBPDwebService.tws?WSDL". The main content area of the browser shows the XML structure of the WSDL document. The XML code defines a service named "ResumeBPDwebService" with a port type "ResumeBPDwebServicePortType" and a binding "ResumeBPDwebServiceSoapBinding". It includes message definitions for "resumeBPDRRequest" and "resumeBPDRResponse", and describes the SOAP port with its location at "http://db2DevHost9080/teamworks/webservices/AD8PA/ResumeBPDwebService.tws". The XML uses color coding for different XML elements and attributes.

Figure 5-32 WSDL

Of course, this simplicity comes at a cost—this is just HTTP Basic Authentication. As we have discussed previously, this is neither an encryption algorithm nor secure.

Important: We suggest running all web services integrations (both inbound and outbound) over SSL/TLS.

5.4 Business Process Manager Advanced Edition web services options

The story concerning web services is very different between the two versions of Business Process Manager, Standard and Advanced. In fact, support for web services is one of the main points of distinction between the two products. Business Process Manager Advanced Edition is more feature rich with respect to all things related to web services. Business Process Manager Advanced Edition:

- ▶ Is based upon JAX-WS 2.1 (JAX-WS 2.2 for BPM 8.0)
- ▶ Uses Policy Sets and Bindings
- ▶ Includes Integration Designer
- ▶ Ships with a fully licensed copy of the WebSphere Enterprise Service Bus

The greatest difference between Business Process Manager Standard Edition's foundation of AXIS and Business Process Manager Advanced Edition's foundation of JAX-WS is support

for the most popular WS-* specifications. There are a host of APIs contained in WS-* that provide for a greatly enhanced security model compared to the AXIS implementation (Figure 5-33).

Specification or API	BPM 7.5 Adv	BPM 7.5 Std
Java API for XML Web Services	JAX-WS 2.1	N/A
Canonical XML	Canonical XML 1.0	N/A
Decryption Transform for XML Signature	Decryption Transform for XML Signature	N/A
Exclusive XML Canonicalization	Exclusive XML Canonicalization v1.0	N/A
OASIS WS-Security: SOAP Message Security	WS-Security v1.1	WS-Security v1.0
OASIS WS-Security: Kerberos Token Profile	Kerberos Token Profile 1.1	N/A
OASIS WS-Security: SAML Token Profile	SAML v1.1 and 2.0 assertions	N/A
OASIS WS-Security: Username Token Profile	Username Token Profile 1.1	Username Token Profile 1.0
OASIS WS-Security: X.509 Token Profile	X.509 Token Profile 1.1	N/A
WS-I Basic Security Profile	WS-I Basic Security Profile 1.1	N/A
WS-I Reliable Secure Profile	WS-I Reliable Secure Profile 1.0 (draft)	N/A
Web Services Secure Conversation	WS-SecureConversation 1.3	N/A
OASIS Web Services Trust	WS-Trust 1.3	N/A
XML Signature Syntax and Processing	XML Signature Syntax and Processing	XML Signature Syntax and Processing
XML Encryption Syntax and Processing	XML Encryption Syntax and Processing	XML Encryption Syntax and Processing

Figure 5-33 Specifications for Business Process Manager Standard and Advanced Editions

Business Process Manager Advanced Edition is built upon an earlier IBM business process management tool, IBM WebSphere Process Server. This product has been in the IBM family for longer than the Lombardi product has, and as a result, it has benefited from a greater number of security reviews. In addition, the security aspects of WebSphere Process Server are the subject of a great number of IBM authored books, papers and articles.

We are therefore not going to try to recreate these valuable works in this book. Instead we have chosen to focus on those security elements that are common to both versions of the current Business Process Manager product, or are specific to Business Process Manager Standard Edition.

5.5 Common security holes

In this section we discuss the failure to secure web services passwords and faith in firewalls.

5.5.1 Failure to secure web services passwords

As was discussed in “Security considerations using the Web Service Integrations” on page 122, any user who has access to a process application in /ProcessCenter—even just read-only access—has full visibility to the values of the web services passwords. It does not matter if they are hard-coded into the Properties tab (under Implementation or Security) or if they are referenced from environment variables. In each of these cases, they are completely visible to anyone who cares to look.

Important: We advise that these web services password values be stored in the process application in an encrypted form, and that access to the keys and cipher functions be highly restricted.

Recall from our discussion of encryption, SSL and certificates: it matters not who is in possession of your encrypted passwords, as long as they do not have access to the secret knowledge that generates and/or decrypts them.

5.5.2 Faith in firewalls

Here it is, once again, our most common security hole by far. In the context of this chapter, a “faith in firewalls” can lead to a huge security hole because of the nature of Business Process Manager Standard Edition’s reliance upon HTTP Basic Authentication for web service credentials.

To illustrate this point, consider Figure 5-34.

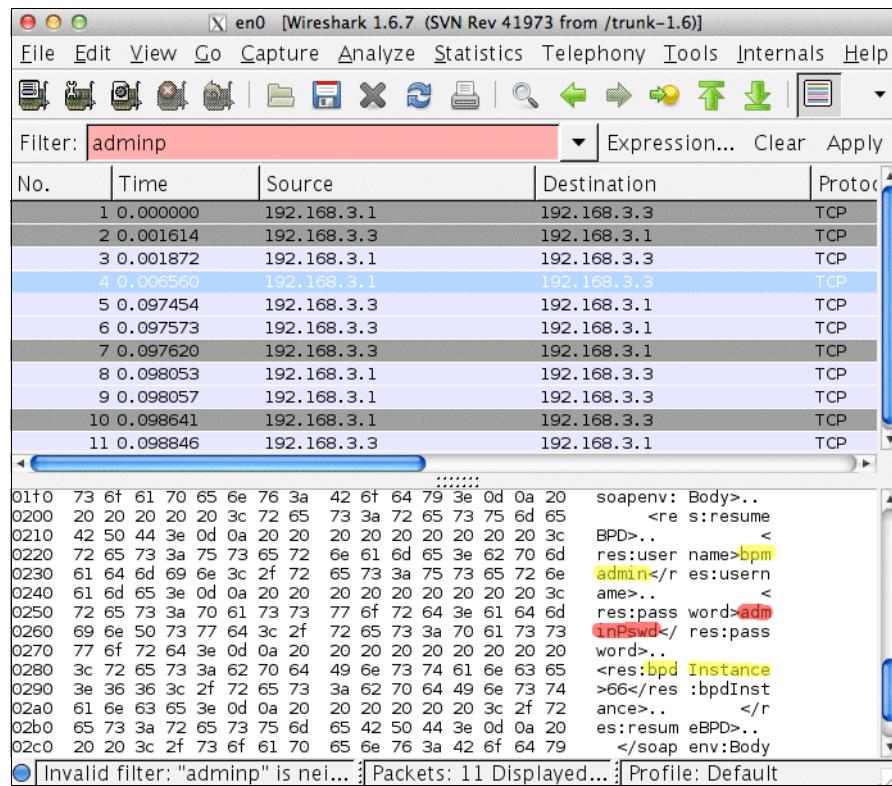


Figure 5-34 Network flow¹

Figure 5-34 shows how clearly visible the admin password is. This capture was made possible by a network protocol analyzer that is easy to download from the Internet.

Important: We suggest that all communications between Business Process Manager and each server that hosts web services be encrypted using SSL/TLS.

¹ Wireshark (www.wireshark.org)

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks publications

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- ▶ *IBM z/OS V1R10 Communications Server TCP/IP Implementation Volume 1: Base Functions, Connectivity, and Routing*, SG24-7696
<http://www.redbooks.ibm.com/redbooks/pdfs/sg247976.pdf>
- ▶ *WebSphere Application Server V7.0 Security Guide*, SG24-7660
<http://www.redbooks.ibm.com/redbooks/pdfs/sg247660.pdf>
- ▶ *HTTP Server (powered by Apache) SSL/TLS Cipher List Handshaking*, TIPS0284
<http://www.redbooks.ibm.com/abstracts/tips0284.html>
- ▶ *DB2 Security and Compliance Solutions for Linux, UNIX, and Windows*, SG24-7555
<http://www.redbooks.ibm.com/abstracts/sg247555.html>
- ▶ *Understanding LDAP - Design and Implementation*, SG24-4986
<http://www.redbooks.ibm.com/redbooks/pdfs/sg244986.pdf>
- ▶ *WebSphere Application Server V7.0 Security Guide*, SG24-7660
<http://www.redbooks.ibm.com/redbooks/pdfs/sg247660.pdf>

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- ▶ *IBM WebSphere Deployment and Advanced Configuration*, IBM Press, ISBN 0131468626
- ▶ *Preparing to Fail: Practical IBM WebSphere Application Server High Availability*, Impact 2012 presentation
- ▶ Database Encryption in SQL Server 2008
<http://msdn.microsoft.com/en-us/library/cc278098%28v=sql.100%29.aspx>
- ▶ Transparent Data Encryption (TDE) in Oracle 10g Database Release 2
<http://www.oracle-base.com/articles/10g/transparent-data-encryption-10gr2.php>
- ▶ Tablespace Encryption in Oracle 11g Database Release 1
<http://www.oracle-base.com/articles/11g/tablespace-encryption-11gr1.php>

- ▶ Securing Stored Data Using Transparent Data Encryption
http://docs.oracle.com/cd/B28359_01/network.111/b28530/asotrans.htm#ASOAG610
- ▶ RSA ClearTrust Web Access Management
http://www.rsa.com/products/cleartrust/whitepapers/CTIBM_WP_0403.pdf

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



IBM Business Process Manager Security: Concepts and Guidance

(0.2"spine)
0.17" <-> 0.473"
90<->249 pages



IBM Business Process Manager Security Concepts and Guidance



Demonstrates why Business Process Manager security is important

Includes easy-to-understand explanations of security topics

Covers specific security hardening exercises

This IBM Redbooks publication provides information about security concerning an organization's business process management (BPM) program, about common security holes that often occur in this field, and describes techniques for rectifying these holes. This book documents preferred practices and common security hardening exercises that you can use to achieve a reasonably well-secured BPM installation.

Many of the practices described in this book apply equally to generic Java Platform and Enterprise Edition (J2EE) applications, as well as to BPM. However, it focuses on aspects that typically do not receive adequate consideration in actual practice. Also, it addresses equally the BPM Standard and BPM Advanced Editions, although there are topics inherent in BPM Advanced that we considered to be out of scope for this book.

This book is not meant as a technical deep-dive into any one topic, technology, or philosophy. IBM offers a variety of training and consulting services that can help you to understand and evaluate the implications of this book's topic in your own organization.

**INTERNATIONAL
TECHNICAL
SUPPORT
ORGANIZATION**

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks